

Confidence Intervals in General Regression Models that Utilize Uncertain Prior Information

Submitted by

Kapuruge Nishika Oshadini Ranathunga

Bachelor of Science (Honours) in Statistics

A thesis submitted in total fulfilment

of the requirements for the degree of

Doctor of Philosophy

College of Science, Health and Engineering

School of Engineering and Mathematical Sciences

La Trobe University

Victoria Australia

February 2021

Contents

Notation	xviii
Summary	xxxvii
Acknowledgements	xxxix
Statement of Authorship	xl
1 Introduction	1
1.1 Literature review	1
1.2 Thesis overview	8
1.2.1 Outline of the contributions due solely to the author . .	13
2 Confidence intervals in general regression models that utilize uncertain prior information	20
2.1 Introduction	20
2.2 The confidence interval that utilizes uncertain prior informa- tion in linear regression with known error variance	22
2.2.1 Likelihood-based approach for the confidence interval $CI(b_\rho, s_\rho)$	24
2.3 Asymptotic results for a general regression model, without a scale parameter	25
2.4 Analogues of I and $CI(b_\rho, s_\rho)$ based on Wald statistics	27
2.5 Analogues of I and $CI(b_\rho, s_\rho)$ based on likelihood functions . .	29

2.5.1	Profile likelihood confidence interval for θ	29
2.5.2	Likelihood-based analogue of $CI(b_\rho, s_\rho)$	30
2.6	Assessment of the coverage probability of a confidence interval	31
2.6.1	Definition of the <i>local minimum coverage probability</i> , for given $\tilde{\beta}$	32
2.6.2	Data-based choice of $\tilde{\beta}$	33
2.7	Definition of the <i>local scaled expected length</i> of a confidence interval	34
2.8	Application to quantal bioassays	35
2.9	Numerical illustration: quantal bioassay of Morphine and Ami- done.	37
2.9.1	Application of the procedure described in subsection 2.6.2	38
2.9.2	Choice of \mathbf{u}	40
2.10	Numerical Illustration: Monte Carlo simulation estimation of the <i>local coverage probabilities</i> and <i>scaled expected lengths</i> of $I_L(\mathbf{y}^*; \mathbf{c})$ and $ACI_L(\mathbf{y}^*)$	41
2.10.1	Monte Carlo simulation estimation of the <i>local cover- age probabilities</i>	42
2.10.2	Monte Carlo simulation estimation of the <i>local scaled expected length</i> of $ACI_L(\mathbf{y}^*)$	44
2.10.3	Large sample approximation to the <i>local coverage prob- ability</i> and scaled expected length of the confidence interval $ACI_L(\mathbf{y}^*)$	45
2.11	Conclusion	46
3	Computation of the expected value of a function of a chi- distributed random variable	48
3.1	Introduction	48

3.2	Integral of interest	50
3.3	Properties of the trapezoidal rule found using the Fourier transform of the integrand	52
3.4	Application of the transformation (2.6) of Mori (1988), fol- lowed by the application of the trapezoidal rule	53
3.4.1	Transformation (2.6) of Mori (1988)	53
3.4.2	Assumption FT and exponential convergence	54
3.4.3	A simple and effective procedure for evaluating the in- tegral (3.6)	57
3.4.4	An exponentially convergent procedure for evaluating the integral (3.6)	58
3.5	Numerical results for the transformation (2.6) of Mori (1988), followed by the application of the trapezoidal rule	61
3.6	Comparison with three other methods of numerical integration	63
3.6.1	Gauss Legendre quadrature	63
3.6.2	Generalized Gauss Laguerre quadrature	65
3.6.3	Inverse cdf method, followed by Gauss Legendre quadra- ture	71
3.7	Application to the computation of the coverage probability and scaled expected volume of non-standard confidence regions	73
3.8	Remarks	76
3.9	Conclusion	76
4	R programs for the computation of the Kabaila & Giri (2009a) confidence interval	78
4.1	Introduction	78
4.2	The confidence interval of Kabaila & Giri (2009a)	80
4.2.1	Choice of d	81
4.2.2	Parametric form for the functions b and s	82

4.2.3	Choice of the knots x_1, \dots, x_q	82
4.2.4	Description of the nonlinear constrained optimization that needs to be carried out to compute the Kabaila & Giri (2009a) confidence interval for given λ	83
4.3	New method for evaluating the computationally convenient formula of Kabaila & Giri (2009a) for the coverage probability of $\text{CI}(b, s)$	85
4.3.1	Examination of $\text{IICP}(x, w, \gamma, \rho)$	87
4.3.2	Evaluation of the outer integral of the double integral term in the expression of $\text{CP}(\gamma; b, s)$	88
4.4	Computationally convenient formula for the scaled expected length of $\text{CI}(b, s)$	89
4.4.1	First definition of scaled expected length and the ob- jective function based on it	90
4.4.2	Second definition of scaled expected length and the objective function based on it	91
4.5	Choice of some of the computational parameters of the new confidence interval and the method used for its computation	93
4.5.1	Choosing the value of d and the set Γ_{grid}	93
4.5.2	Choosing the values of l and n.ints	94
4.5.3	Choosing the number of outer integrand evaluations, the value of ϵ and the number of Gauss Legendre quadra- ture nodes	96
4.5.4	Comparison with some of the past results obtained by Kabaila and Giri	96
4.6	Computation of the Kabaila & Giri (2009a) confidence interval for the standard choice of λ	98
4.6.1	Standard choice of λ for $m > 2$	99

4.6.2	Standard choice of λ for $m \in \{1, 2\}$	100
4.7	Comparison of the numerical results for the two definitions of scaled expected length for the standard choice of λ	100
4.7.1	Comparison for $m > 2$	101
4.7.2	Comparison for $m \in \{1, 2\}$	103
4.8	Conclusion	105
5	On adaptive Gauss-Hermite quadrature for estimation in generalized linear mixed models	107
5.1	Introduction	107
5.2	The logistic regression with random intercept model and its log-likelihood function	109
5.3	Adaptive Gauss-Hermite quadrature	110
5.4	The teratology data and importance sampling	112
5.4.1	Description of the dataset and the maximum likelihood estimates	112
5.4.2	The performance of importance sampling	113
5.5	The performance of adaptive Gauss-Hermite quadrature for cluster 29 of the teratology data	115
5.6	Remark	118
5.7	Conclusion	119
A	Proofs and R programs for Chapter 2	120
A.1	Derivation of the expressions for $r_1(\theta')$ and r_2 given in subsec- tion 2.2.1	120
A.2	Numerical results for the <i>local coverage probability</i> and <i>local scaled expected length</i> of the confidence interval $\text{ACI}_W(\hat{\beta}^*)$	122

A.3	Expressions for the coverage probabilities of $I_L(\mathbf{y}; \mathbf{c})$ and $ACI_L(\mathbf{y})$ that do not require the computation of the endpoints of these confidence intervals	125
A.4	Evidence in favour of the correctness of Assumption A	126
A.5	Fast evaluation of the functions b_ρ and s_ρ for any given ρ . . .	127
A.6	Computation of the SRLR statistics	129
A.7	Computation of \tilde{c}	130
A.7.1	Monte Carlo simulation estimation of the <i>minimum</i> <i>coverage probabilities</i> of $ACI_W(\hat{\beta}^*)$ and $I_W(\mathbf{y}^*; c)$. . .	130
A.7.2	Use of a fitted straight line in c to compute \tilde{c}	131
A.8	The distribution of q_L^*	131
A.9	R programs for computing the coverage probability and the scaled expected length of the likelihood-based confidence in- tervals.	133
A.9.1	Computation of the γ^* values	133
A.9.2	Coverage probability of the likelihood-based confidence intervals, $I_L(\mathbf{y}^*; 0.05)$ and $ACI_L(\mathbf{y}^*)$	136
A.9.3	Scaled expected length of the likelihood-based confi- dence interval $ACI_L(\mathbf{y}^*)$	142
A.9.4	Grewal (1952) data set used in the R programs	148
B	Proofs and R programs for Chapter 3	149
B.1	Application of the Poisson summation formula to understand- ing the accuracy of the trapezoidal rule	149
B.2	Application of the transformation (2.6) of Mori (1988)	151
B.3	Proof of Theorem 3.4.1	152
B.4	An expression for a polynomial p of degree u using the bino- mial theorem	160
B.5	Computation of $F_\nu^{-1}(y)$	161

B.6	Proof of the result (3.19)	162
B.7	R programs for the computation of the integral (3.1)	164
B.7.1	The transformation (2.6) of Mori (1988), followed by the application of the trapezoidal rule	164
B.7.2	Gauss Legendre quadrature method	169
B.7.3	Generalized Gauss Laguerre quadrature method	171
B.7.4	Inverse cdf method followed by Gauss Legendre quadrature	173
C	Proofs and R programs for Chapter 4	175
C.1	Computation of the first definition of scaled expected length $\text{SEL}_1(\gamma; s)$	175
C.1.1	Computation of the inner integral, $\text{ISEL}(w, \gamma)$, of the double integral term in the expression of $\text{SEL}_1(\gamma; s)$	175
C.1.2	Computation of the outer integral of the double inte- gral term in the expression of $\text{SEL}_1(\gamma; s)$	176
C.1.3	Computation of $E(W)$	177
C.1.4	Evaluation of the objective function $\text{OBJ}_1(\gamma; s)$	178
C.2	Computation of the second definition of scaled expected length $\text{SEL}_2(\gamma; s)$	178
C.2.1	Computationally convenient formula for the second def- inition of scaled expected length $\text{SEL}_2(\gamma; s)$	179
C.2.2	Computationally convenient formula for the objective function $\text{OBJ}_2(\gamma; s)$	179
C.2.3	Computation of the outer integral of the double inte- gral term in the expression of $\text{SEL}_2(\gamma; s)$	181
C.2.4	Evaluation of the objective function $\text{OBJ}_2(\gamma; s)$	182
C.3	Exploration of the six cases given in Table 4.1	183
C.3.1	Numerical results for a given value of λ for $m = 1$	187

C.4	Choosing the number of outer integrand evaluations, the value of ϵ and the number of Gauss Legendre quadrature nodes . . .	189
C.4.1	Choice of N, ϵ and the number of Gauss Legendre quadrature nodes to compute the functions $b_{1\lambda}$ and $s_{1\lambda}$	190
C.4.2	Choice of N, ϵ and the number of Gauss Legendre quadrature nodes to compute the graphs of the coverage probability and the squared scaled expected length	192
C.5	Differences between the graphs of the coverage probabilities and the differences between the graphs of the squared scaled expected lengths	193
C.6	Comparison with some of the past results obtained by Kabaila and Giri	197
C.7	R programs for the computation of the Kabaila & Giri (2009a) confidence interval for the standard choice of λ	201
C.7.1	R programs for the computation of the vector $(b_1(d/6), \dots, b_1(5d/6), s_1(0), \dots, s_1(5d/6))$	202
C.7.2	R program for the computation of the graphs of the functions b_1 and s_1	225
C.7.3	R program for the computation of the graph of the coverage probability $CP(\gamma; b_1, s_1)$	226
C.7.4	R programs for the computation of the graph of the first definition of scaled expected length $SEL_1(\gamma; s_1)$. .	228
C.7.5	R programs for the computation of the graph of the second definition of scaled expected length $SEL_2(\gamma; s_2)$	233
D	Proofs and R programs for Chapter 5	236
D.1	Derivation of the mode and the variance of the posterior distribution	236

D.2	An expression for $c(\theta, \sigma)$ needed for the application of Gauss-Hermite quadrature	237
D.3	An expression for $r(t; \theta, \sigma)$	238
D.4	Examination of the possible values for c_3	238
D.5	Computation of the maximum likelihood estimates of the parameters of the logistic regression model with random intercept	240
D.6	Proof of Theorem 5.5.1	241
D.7	Evaluation of $k(z)$ for $c_3 > 0$	245
D.8	R programs for the evaluation of the performance of importance sampling	246
D.8.1	R programs for the computation of the maximum likelihood estimates $\hat{\beta}_1$, $\hat{\beta}_2$ and $\hat{\sigma}$	246
D.8.2	R programs for the computation of $\log r(t; \theta, \sigma)$	249
Bibliography		252

List of Figures

2.1	Approximate 95% confidence intervals for the coverage probabilities of the confidence intervals $I_L(\mathbf{y}^*; 0.05)$ (top panel) and $ACI_L(\mathbf{y}^*)$ (bottom panel), both with nominal coverage 0.95, for $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$	43
2.2	The left panel presents approximate 95% confidence intervals for the <i>local scaled expected length</i> of the confidence interval $ACI_L(\mathbf{y}^*)$, with nominal coverage 0.95, evaluated at $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$. The right panel is the graph of $SEL(\gamma^*; \rho(\tilde{\boldsymbol{\beta}}))$ for the confidence interval $CI(b_{\rho(\tilde{\boldsymbol{\beta}})}, s_{\rho(\tilde{\boldsymbol{\beta}})})$, found using the R package <code>ciuupi</code>	45
2.3	Graphs of the coverage probability $CP(\gamma^*; \rho(\tilde{\boldsymbol{\beta}}))$ and the scaled expected length $SEL(\gamma^*; \rho(\tilde{\boldsymbol{\beta}}))$ for the confidence interval $CI(b_{\rho(\tilde{\boldsymbol{\beta}})}, s_{\rho(\tilde{\boldsymbol{\beta}})})$, found using the R package <code>ciuupi</code>	46
3.1	Plot of the function $\psi_\nu(y)$ as a function of y for $\nu = 1, 2, 3, 10$ and 100	54
3.2	Graph of $\psi_\nu(y_u + t)$ as a function of t for $\nu = 10$ and $h = 0.01$	56
3.3	Graphs of $a_{\nu, \alpha}(x)$ as a function of x for $\alpha = 0.05$ and $\nu = 1, 2$ and ∞	61
3.4	Graphs of $d_\nu(y)$ as a function of y for $\nu = 1, 2, 3$ and 10 and $\alpha = 0.10, 0.05$ and 0.02	67

3.5	The top and bottom panels are scatterplots of the (y_j, w_j) 's for $(\nu, m) = (1, 65)$ and $(\nu, m) = (2, 33)$, respectively	70
3.6	Graphs of $b_\nu(z)$ as a function of z for $\nu = 1, 3, 10$ and 100 and $\alpha = 0.10, 0.05$ and 0.02	72
4.1	Graphs of d and extra as a functions of m	94
4.2	Graphs of the functions b_1 and s_1 (left panel) and b_2 and s_2 (right panel) for $m = 76$, $\alpha = 0.05$, $\rho = -0.707$, $d = 6.1$, n.ints = 6 and $l = 60$. The standard choice of λ is equal to 0.104 for the both confidence intervals $\text{CI}(b_1, s_1)$ and $\text{CI}(b_2, s_2)$	102
4.3	Graphs of the coverage probability and the squared scaled expected length of the confidence intervals $\text{CI}(b_1, s_1)$ (left panel) and $\text{CI}(b_2, s_2)$ (right panel) for $m = 76$, $\alpha = 0.05$, $\rho = -0.707$, $d = 6.1$, n.ints = 6 and $l = 60$. The standard choice of λ is equal to 0.104 for the both confidence intervals $\text{CI}(b_1, s_1)$ and $\text{CI}(b_2, s_2)$	103
4.4	Graphs of the functions b_1 and s_1 (left panel) and b_2 and s_2 (right panel) for $m = 2$, $\alpha = 0.05$, $\rho = -0.5$, $d = 10.1$, n.ints = 6 and $l = 60$. The standard choice of λ s is equal to 0.023 and 0.020 for the confidence intervals $\text{CI}(b_1, s_1)$ and $\text{CI}(b_2, s_2)$, respectively.	104
4.5	Graphs of the coverage probability and the squared scaled expected length of the confidence intervals $\text{CI}(b_1, s_1)$ (left panel) and $\text{CI}(b_2, s_2)$ (right panel) for $m = 2$, $\alpha = 0.05$, $\rho = -0.5$, $d = 10.1$, n.ints = 6 and $l = 60$. The standard choice of λ s is equal to 0.023 and 0.020 for the confidence intervals $\text{CI}(b_1, s_1)$ and $\text{CI}(b_2, s_2)$, respectively.	105

5.1	Graphs of $\log r(t; \theta, \sigma)$ as a function of t for $\sigma = 1.3$, $\theta = 2.6$ for a control litter and $\theta = 1.5$ for a treatment litter. The top two graphs are for control litters 12 and 15. The bottom two graphs are for treatment litters 29 and 32.	115
5.2	Graph of the function $k(z) - q_m(z) = h(\mu + \sqrt{2}\tau z; \theta, \sigma) - q_m(z)$, considered as a function of z , for litter 29 for adaptive Gauss-Hermite quadrature with $m = 5$ nodes. The smallest and largest nodes z_1 and z_m , respectively, are shown.	117
A.1	Approximate 95% confidence intervals for the coverage probabilities of the confidence intervals $I_W(\mathbf{y}^*; 0.05)$ (top panel) and $ACI_W(\hat{\beta}^*)$ (bottom panel), both with nominal coverage 0.95, for $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$	123
A.2	Plot of the estimated minimum coverage probability of $I_W(\mathbf{y}^*; c)$ for $c = 0.03, 0.05$ and 0.07 . The red straight line is the least squares fit to this data.	124
A.3	The left panel presents approximate 95% confidence intervals for the <i>local scaled expected length</i> of the confidence interval $ACI_W(\hat{\beta}^*)$, which has nominal coverage 0.95, for $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$. The right panel is the graph of $SEL(\gamma^*; \tilde{\beta})$ for the confidence interval $CI(b_{\rho(\tilde{\beta})}, s_{\rho(\tilde{\beta})})$, found using the R package <code>ciuupi</code>	125
C.1	Graphs of the function $b_{1\lambda}$ for $m = 2$, $\alpha = 0.05$, $\lambda = 0.15$, $\rho = -0.5$, $N = N_1 = 17$, $\epsilon = 10^{-7}$ and Gauss Legendre quadrature with 10 nodes for the cases (a) - (f) in Table 4.1. .	184
C.2	Graphs of the function $s_{1\lambda}$ for $m = 2$, $\alpha = 0.05$, $\lambda = 0.15$, $\rho = -0.5$, $N = N_1 = 17$, $\epsilon = 10^{-7}$ and Gauss Legendre quadrature with 10 nodes for the cases (a) - (f) in Table 4.1. .	185

C.3	Graphs of the coverage probability of the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for $m = 2$, $\alpha = 0.05$, $\lambda = 0.15$, $\rho = -0.5$, $N_1 = 33$, $\epsilon = 10^{-10}$ and Gauss Legendre quadrature with 20 nodes for the cases (a) - (f) in Table 4.1.	186
C.4	Graphs of the squared scaled expected length of the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for $m = 2$, $\alpha = 0.05$, $\lambda = 0.15$, $\rho = -0.5$, $N_1 = 33$, $\epsilon = 10^{-10}$ and Gauss Legendre quadrature with 20 nodes for the cases (a) - (f) in Table 4.1.	187
C.5	Graphs of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length of the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for (a) $l = 60$, $d = 20$, n.ints =6. Note that $m = 1$, $\alpha = 0.05$, $\rho = 0.4$ and $\lambda = 0.2$	188
C.6	Graphs of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length of the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for (a) $l = 60$, $d = 20$, n.ints =7. Note that $m = 1$, $\alpha = 0.05$, $\rho = 0.4$ and $\lambda = 0.2$	189
C.7	Plots of the differences, (1) - (2), between the graphs of the coverage probabilities for the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for the six cases (a) - (f) in Table 4.1. Note that $m = 2$, $\alpha = 0.05$, $\rho = -0.5$, $\lambda = 0.15$	195
C.8	Plots of the differences, (1) - (2), between the graphs of the squared scaled expected lengths for the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for the six cases (a) - (f) in Table 4.1. Note that $m = 2$, $\alpha = 0.05$, $\rho = -0.5$, $\lambda = 0.15$	196
C.9	Graphs of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length of the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for the example (a). In this example, $m = 76$, $\alpha = 0.05$, $\rho = -0.707$, $\lambda = 0.2$, $d = 6$ and n.ints = 6.	198

C.10	Graphs of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length of the confidence interval $\text{CI}(b_{1\lambda}, s_{1\lambda})$ for the example (b). In this example, $m = 2$, $\alpha = 0.05$, $\rho = -0.5$, $\lambda = 0.15$, $d = 14$ and n.ints = 7.	199
C.11	Graphs of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length of the confidence interval $\text{CI}(b_{1\lambda}, s_{1\lambda})$ for the example (c). In this example, $m = 1$, $\alpha = 0.05$, $\rho = 0.4$, $\lambda = 0.2$, $d = 30$ and n.ints = 6.	200

List of Tables

2.1	Quantal bioassay of Morphine and Amidone	37
2.2	The values of γ^* and τ^* and the percentage of simulation runs (row marked %) for which the length of the profile likelihood confidence interval is greater than 1000 for $z = 60$ and $M =$ 5000 simulation runs.	41
3.1	The approximation error for the simple procedure described in subsection 3.4.3, with $\epsilon = 10^{-17}$ and stopped after the compu- tation of the approximation for $n = 65$ for $\nu = 1$ and $n = 33$ for $\nu = 2, 3, 4, 5, 10, 100$ and 1000. Here $\alpha = 0.10, 0.05$ and 0.02 and $\nu = 1, 2, 3, 4, 5, 10, 100$ and 1000. We interpret an entry 0 in this table as $ \text{approximation error} < 1.11 \times 10^{-16}$. .	62
3.2	The approximation error for Gauss Legendre quadrature ap- plied to the evaluation of (3.13), for $\alpha = 0.10, 0.05$ and 0.02 and $\nu = 1, 2, 3, 4, 5, 10, 100$ and 1000. The number of nodes m is 65 for $\nu = 1$ and 33 for $\nu = 2, 3, 4, 5, 10, 100$ and 1000. . . .	65
3.3	The approximation error for Generalized Gauss Laguerre quadra- ture for $\alpha = 0.10, 0.05$ and 0.02 and $\nu = 1, 2, 3, 4, 5, 6, 10, 100$ and 300. The number of nodes m is 65 for $\nu = 1$ and 33 for $\nu = 2, 3, 4, 5, 6, 10, 100$ and 300.	68

3.4	The approximation error for the inverse cdf method, using Gauss Legendre quadrature, for $\alpha = 0.10, 0.05$ and 0.02 and $\nu = 1, 2, 3, 4, 5, 6, 10, 100$ and 1000 . The number of nodes m is 65 for $\nu = 1$ and 33 for $\nu = 2, 3, 4, 5, 6, 10, 100$ and 1000	73
4.1	The values of $(\text{SEL}_1(0; s_{1\lambda}))^2$, $\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda})\right)^2$ and the time needed to compute the vector $(b_{1\lambda}(x_2), \dots, b_{1\lambda}(x_{q-1}), s_{1\lambda}(x_1), \dots, s_{1\lambda}(x_{q-1}))$ for $m = 2$, $\alpha = 0.05$, $\rho = -0.5$ and $\lambda = 0.15$. . .	95
4.2	Performance of the confidence interval $\text{CI}(b_1, s_1)$ for $m = 76$, $\alpha = 0.05$, $\rho = -0.707$, $d = 6.1$, <code>n.ints</code> = 6 and $l = 60$ for $\lambda \in \{0.2, 0.105\}$	99
4.3	Performance of the confidence interval $\text{CI}(b_1, s_1)$ for $m = 2$, $\alpha = 0.05$, $\rho = -0.5$, $d = 10.1$, <code>n.ints</code> = 6 and $l = 60$ for $\lambda \in \{0.15, 0.023\}$	100
5.1	Teratology data set of Weil (1970). This data lists the number of rat pups in 16 control litters that survived and the the number of rat pups in 16 treated litters that survived.	113
5.2	Values of $10^8 \times e_m$ for cluster 29, where z_1 and z_m are the smallest and largest nodes for Gaussian-Hermite with m nodes. Here z_ℓ and z_u are the lower and upper limits such that $z_\ell \leq z_1$ and $z_u \geq z_m$	118
A.1	Some descriptive statistics for $q_L^*(1), \dots, q_L^*(M)$ for $\gamma^* \in \{-2.5, -1.5, -0.5, 0.5, 1.5, 2.5\}$ where the number of simulation runs, for each value of γ^* , is $M = 10,000$	132
A.2	Grewal1952.csv data set used in the R programs	148

C.1	The values of $(\text{SEL}_1(0; s_{1\lambda}))^2$, $\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda})\right)^2$ and the times needed to compute the vector $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$ for $N \in \{5, 9, 17, 33\}$ for Gauss Legendre quadrature with 5 nodes and $\epsilon = 10^{-7}$.	191
C.2	The values of $(\text{SEL}_1(0; s_{1\lambda}))^2$, $\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda})\right)^2$ and the times needed to compute the vector $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$ for $N \in \{5, 9, 17, 33\}$ for Gauss Legendre quadrature with 10 nodes and $\epsilon = 10^{-7}$.	192

Notation

Notation used in Chapter 2

Latin Alphabet

a	centre of the interval $[a - w, a + w]$
\mathbf{a}	a specified nonzero p -vector
A	a compound used in a quantal bioassay
$\text{avar}(\hat{\theta}; \boldsymbol{\beta})$	the asymptotic variance of $\hat{\theta}$
$\text{avar}(\hat{\tau}; \boldsymbol{\beta})$	the asymptotic variance of $\hat{\tau}$
$\text{acov}(\hat{\theta}, \hat{\tau}; \boldsymbol{\beta})$	the asymptotic covariance of $\hat{\theta}$ and $\hat{\tau}$
$\text{ACI}_L(\mathbf{y})$	a likelihood-based analogue of $\text{CI}(b_\rho, s_\rho)$ which has lower endpoint $\inf(\mathcal{S}_A(\mathbf{y}))$ and upper endpoint $\sup(\mathcal{S}_A(\mathbf{y}))$
$\text{ACI}_W(\hat{\boldsymbol{\beta}})$	the analogue of $\text{CI}(b_\rho, s_\rho)$, based on Wald test statistics
b	$b : \mathbb{R} \rightarrow \mathbb{R}$ is an odd continuous function, where $b(x) = 0$ for all $ x \geq 6$
b_ρ	the function b computed using the R package <code>ciuupi</code> , for given $1 - \alpha$ and ρ
\mathcal{B}	$\boldsymbol{\beta}$ belongs to the open set \mathcal{B}
B	a compound used in a quantal bioassay
c	a specified value where $0 < c < 1/2$
\tilde{c}	a value of c where the <i>local minimum coverage probabilities</i> of $\text{ACI}_W(\hat{\boldsymbol{\beta}}^*)$ and $\text{I}_W(\mathbf{y}^*; \tilde{c})$ or the <i>local minimum coverage probabilities</i> of $\text{ACI}_L(\mathbf{y}^*)$ and $\text{I}_L(\mathbf{y}^*; \tilde{c})$ are the same

\mathbf{c}	a specified nonzero p -vector
$\text{CI}(b_\rho, s_\rho)$	the confidence interval for θ computed by the <code>ciuupi</code> package, with minimum coverage $1 - \alpha$, that utilizes the uncertain prior information that $\tau = t$
$CP(\gamma; \rho)$	coverage probability of the confidence interval $\text{CI}(b_\rho, s_\rho)$
$CP(\gamma^*, \rho(\tilde{\boldsymbol{\beta}}))$	coverage probability of the confidence interval $\text{CI}(b_{\rho(\tilde{\boldsymbol{\beta}})}, s_{\rho(\tilde{\boldsymbol{\beta}})})$
$\text{cov}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\tau}})$	the covariance of $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\tau}}$, given by $\sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{c}$
d	any dose level of compound A
d'	any dose level of compound B
d_i	i 'th dose level where $i = 1, \dots, m$
ED_z	the log-dose x of compound A for which the probability of response S for a randomly chosen individual is $z/100$
ED'_z	the log-dose x' of compound B for which the probability of response S for a randomly chosen individual is $z/100$
$E_{\boldsymbol{\beta}^*}(q_L^*)$	the <i>local scaled expected length</i> of $\text{ACI}_L(\mathbf{y}^*)$
g	$g : \mathbb{R}^p \rightarrow \mathbb{R}$ is a sufficiently smooth function
h	$h : \mathbb{R}^p \rightarrow \mathbb{R}$ is a sufficiently smooth function
i	index for the dose level
I	usual $1 - \alpha$ confidence interval for θ in the context of the linear regression model with known error variance
$\text{I}_L(\mathbf{y}; \alpha)$	the likelihood-based analogue of the confidence interval I , with nominal coverage $1 - \alpha$, which has lower endpoint $\inf(\mathcal{S}_{\text{PL}}(\mathbf{y}))$ and upper endpoint $\sup(\mathcal{S}_{\text{PL}}(\mathbf{y}))$
$\text{I}_W(\mathbf{y}; \alpha)$	the Wald-based analogue of the confidence interval I , with nominal coverage $1 - \alpha$
$I(\boldsymbol{\beta})$	the Fisher information matrix
k	a summation index

$l(\theta, \tau)$	log-likelihood function for a linear regression model, with known error variance
$\ell(\boldsymbol{\beta} \mathbf{y})$	log-likelihood function for a general regression model, without a scale parameter
L	the particular case that the regression model is linear and has independent and identically normally distributed random errors with known error variance
m	number of dose levels
M	number of simulation runs
M'	initial number of simulation runs
n	number of responses
n_i	number of individuals given dose d_i of compound A
n'_i	number of individuals given dose d_i of compound B
N_i	number of individuals in the i 'th group for $i = 1, \dots, n$
p	number of unknown parameters
p	the probability of response S for a randomly chosen individual from the population given dose level d of compound A
p'	the probability of response S for a randomly chosen individual from the population given dose level d' of compound B
q_L^*	defined to be the length of $\text{ACI}_L(\mathbf{y}^*)$ divided by the length of $\text{I}_L(\mathbf{y}^*; \tilde{c})$ computed from the same data
$q_L^*(k)$	an observation of q_L^* generated on the k th simulation run
$r_1(\theta')$	the SRLR statistic, $\text{sign}(\hat{\theta} - \theta') \sqrt{2 \left(l(\hat{\theta}, \hat{\tau}) - l(\theta', \hat{\tau}_{\theta'}) \right)}$
$r_1(\theta' \mathbf{y})$	the SRLR statistic, $\text{sign}(\hat{\theta} - \theta') \sqrt{2 \left(\ell(\hat{\boldsymbol{\beta}} \mathbf{y}) - \ell(\hat{\boldsymbol{\beta}}(\theta'; \theta) \mathbf{y}) \right)}$
r_2	the SRLR statistic, $\text{sign}(\hat{\tau} - t) \sqrt{2 \left(l(\hat{\theta}, \hat{\tau}) - l(\hat{\theta}_t, t) \right)}$
$r_2(\mathbf{y})$	the SRLR statistic, $\text{sign}(\hat{\tau} - t) \sqrt{2 \left(\ell(\hat{\boldsymbol{\beta}} \mathbf{y}) - \ell(\hat{\boldsymbol{\beta}}(t; \tau) \mathbf{y}) \right)}$

r_i	the number of individuals with response S out of n_i given dose d_i of compound A for $i = 1, \dots, m$
r_i'	the number of individuals with response S out of n_i' given dose d_i of compound B for $i = 1, \dots, m$
s	$s : \mathbb{R} \rightarrow \mathbb{R}$ is an even continuous function, where $s(x) = z_{1-\alpha/2}$ for all $ x \geq 6$
s_ρ	the function s computed using the R package <code>ciuupi</code> , for given $1 - \alpha$ and ρ
S	a possible value of a dichotomous response, whose possible values are S and not- S
$SEL(\gamma; \rho)$	scaled expected length of the confidence interval $CI(b_\rho, s_\rho)$
$SEL(\gamma^*, \rho(\tilde{\beta}))$	scaled expected length of the confidence interval $CI(b_{\rho(\tilde{\beta})}, s_{\rho(\tilde{\beta})})$
SRLR	stands for the signed root likelihood ratio test statistic
$\mathcal{S}_{PL}(\mathbf{y})$	the likelihood based confidence set for θ , with nominal coverage $1 - c$, analogous to I
$\mathcal{S}_A(\mathbf{y})$	the likelihood based confidence set for θ , with nominal coverage $1 - \alpha$, analogous to $CI(b_\rho, s_\rho)$
t	a specified number
u	a specified number which satisfies $1 \leq u \leq 10$
$\text{var}(\hat{\theta})$	variance of $\hat{\theta}$, given by $\sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a}$
$\text{var}(\hat{\tau})$	variance of $\hat{\tau}$, given by $\sigma^2 \mathbf{c}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{c}$
v_θ	compact notation for $\text{var}(\hat{\theta})$
v_τ	compact notation for $\text{var}(\hat{\tau})$
w	half width of the interval $[a - w, a + w]$ ($w > 0$)
x	$\log_{10}(d)$
x'	$\log_{10}(d')$

\mathbf{x}_i	a vector of explanatory variables of dimension p for $i = 1, \dots, n$
\mathbf{X}	a known $n \times p$ matrix with linearly independent columns
y_i	i 'th independent response variable which has pmf or pdf $f_i(y \mathbf{x}_i, \boldsymbol{\beta})$ evaluated at y
\mathbf{y}	a random n -vector of responses
\mathbf{y}^*	the response vector for the simulated data
$\mathcal{Y}_{\boldsymbol{\beta}}$	a set of values of \mathbf{y}
z	a specified number where $0 < z < 100$
z_p	p 'th quantile of the standard normal distribution, defined by $P(Z \leq z_p) = p$ for $Z \sim N(0, 1)$

Greek Alphabet

α	$1 - \alpha$ is the nominal coverage
β_1	intercept parameter in the Morphine model
β_2	slope parameter in the Morphine model
β_3	intercept parameter in the Amidone model
β_4	slope parameter in the Amidone model
$\boldsymbol{\beta}$	an unknown parameter p -vector
$\hat{\boldsymbol{\beta}}$	the maximum likelihood estimator of $\boldsymbol{\beta}$
$\boldsymbol{\beta}^*$	defined to be $\tilde{\boldsymbol{\beta}} + \kappa (\partial h(\tilde{\boldsymbol{\beta}})/\partial \boldsymbol{\beta})^\top$
$\hat{\boldsymbol{\beta}}^*$	the maximum likelihood estimator of $\boldsymbol{\beta}^*$
$\hat{\boldsymbol{\beta}}(\theta'; \theta)$	maximises $\ell(\boldsymbol{\beta} \mathbf{y})$ with respect to $\boldsymbol{\beta}$, subject to the constraint that $g(\boldsymbol{\beta}) = \theta'$
$\hat{\boldsymbol{\beta}}(t; \tau)$	maximises $\ell(\boldsymbol{\beta} \mathbf{y})$ with respect to $\boldsymbol{\beta}$, subject to the constraint that $h(\boldsymbol{\beta}) = t$
$\tilde{\boldsymbol{\beta}}$	a given value that satisfies $h(\tilde{\boldsymbol{\beta}}) = t$
δ	a positive number equal to $u/5$
δ_c	a judiciously-chosen small positive number
$\boldsymbol{\varepsilon}$	random error vector with $N(\mathbf{0}, \sigma^2 \mathbf{I})$ distribution
γ	defined to be $(\tau - t)/(\text{var}(\hat{\tau}))^{1/2}$
γ_a	a specified value in the interval $[-u, u]$
γ^*	defined to be $\tau^*/(\text{avar}(\hat{\tau}; \tilde{\boldsymbol{\beta}}))^{1/2}$
κ	defined to be $(\text{avar}(\hat{\tau}; \tilde{\boldsymbol{\beta}}))^{1/2} / \ \partial h(\tilde{\boldsymbol{\beta}})/\partial \boldsymbol{\beta}\ ^2 \gamma_a$
λ	a fixed positive number
ψ_i	the i 'th probability of response where $y_i \sim \text{Binomial}(N_i, \psi_i)$ for $i = 1, \dots, n$
ρ	the correlation between $\hat{\theta}$ and $\hat{\tau}$
$\rho(\boldsymbol{\beta})$	defined to be $\text{acov}(\hat{\theta}, \hat{\tau}; \boldsymbol{\beta}) / \left(\text{avar}(\hat{\theta}; \boldsymbol{\beta}) \text{avar}(\hat{\tau}; \boldsymbol{\beta}) \right)^{1/2}$
σ^2	known error variance

τ	scalar parameter, distinct from θ , given by $h(\boldsymbol{\beta})$
$\widehat{\tau}$	maximum likelihood estimator of τ
$\widehat{\tau}_\theta$	the value of τ that maximizes $l(\theta, \tau)$ with respect to τ , for given θ
τ^*	given by $h(\boldsymbol{\beta}^*)$
$\widehat{\tau}^*$	maximum likelihood estimator of τ^*
θ	scalar parameter of interest, given by $g(\boldsymbol{\beta})$
θ'	a specified value of θ
$\widehat{\theta}$	maximum likelihood estimator of θ
$\widehat{\theta}_t$	the value of θ that maximizes $l(\theta, \tau)$ with respect to θ , for $\tau = t$
$\widehat{\theta}_l$	lower endpoint of the confidence interval $I_L(\mathbf{y}; \alpha)$
$\widehat{\theta}_u$	upper endpoint of the confidence interval $I_L(\mathbf{y}; \alpha)$
$\widetilde{\theta}_l$	lower endpoint of the confidence interval $ACI_L(\mathbf{y})$
$\widetilde{\theta}_u$	upper endpoint of the confidence interval $ACI_L(\mathbf{y})$
θ^*	given by $g(\boldsymbol{\beta}^*)$
$\widehat{\theta}^*$	maximum likelihood estimator of θ^*

Notation used in Chapter 3

Latin Alphabet

a	a smooth bounded real-valued function
a_0	constant term in the polynomial $p(y)$
$a_{\nu,\alpha}(x)$	defined to be $2\Phi(t_{\nu,1-\alpha/2}x) - 1$
$a(y)$	defined to be $\lambda(y/c(\kappa, \xi))$
b	a small positive integer multiple of h_0
$b_\nu(z)$	defined to be $a(F_\nu^{-1}((z+1)/2))/2$
c_{FT}	a positive number used in the description of the Assumption FT
c_1, c_2, c_3	positive numbers that appear in the description of <i>double exponential</i> decay
c_4	a positive number used in the description of the Assumption FT
c_T	defined to be $9h_0/(10\log_e(2))$
$c(y)$	defined to be $y^{(\nu/2)-1}\exp(-y)$, where $y = \nu x^2/2$
$c(\kappa, \xi)$	defined to be $(\kappa/(\kappa + \xi))^{1/2}$
d	defined to be nh
d_0	the initial value of d
$d_\nu(y)$	defined to be $a((2y/\nu)^{1/2})$
f_ν	pdf of a random variable with the same distribution as $R/\nu^{1/2}$, where $R \sim \chi_\nu$
F_ν	the cdf corresponding to the pdf f_ν
g	a real-valued absolutely integrable function
$g_\nu(y)$	defined to be $a(x(y))\psi_\nu(y)$
$g_{\nu,\alpha}(y)$	defined to be $a_{\nu,\alpha}(x(y))\psi_\nu(y)$
G	the Fourier transform of g
G_ν	the Fourier transform of $g_\nu(y)$

h	the step length for the trapezoidal rule
h_0	initial value of h
i	defined to be $\sqrt{-1}$
k	k is the iteration number of the exponentially convergent procedure
$k_\nu(s)$	defined to be $a\left(x\left(\left((y_u - y_l)s\right)/2 + (y_l + y_u)/2\right)\right) \times$ $\psi_\nu\left(\left((y_u - y_l)s\right)/2 + (y_l + y_u)/2\right)$
m	denotes the number of quadrature nodes for either Gauss Legendre or Gauss Laguerre quadrature
M	lower endpoint of the finite sum approximation, based on the trapezoidal rule, to the integrand $g(y)$
n	number of evaluations of the integrand $a(x(y))\psi_\nu(y)$
n_0	initial value of n
N	upper endpoint of the finite sum approximation, based on the trapezoidal rule, to the integrand $g(y)$
p	a polynomial of degree u
Q_ν	denotes the χ_ν^2 cdf
R	a random variable with χ_ν distribution
t	a real-valued variable, where $t \in (-\infty, y_\nu^*]$
s	defined to be $(2/(y_u - y_l)) (y - ((y_l + y_u)/2))$
$t_{\nu,a}$	a 'th quantile of the t_ν distribution, defined by $P(T \leq t_{\nu,a}) = a$ for $T \sim t_\nu$
$u_\nu(y, d)$	defined to be $Q_\nu\left(\nu x^2(y)\right) + 1 - Q_\nu\left(\nu x^2(y + d)\right)$
w_j	j 'th weight of the Generalized Gauss Laguerre quadrature
\tilde{w}_j	j 'th weight of the Gauss Legendre quadrature
$x(y)$	defined to be $\exp(y/2 - e^{-y})$
y_j	j 'th node of the Generalized Gauss Laguerre quadrature

y_ℓ	the first evaluation of the integrand $a(x(y))\psi_\nu(y)$ is at y_ℓ
$y_{\ell 0}$	initial value of y_ℓ
y_u	is set equal to $y_\ell + d$
y_{u0}	is set equal to $y_{\ell 0} + d_0$
y_ν^*	the value of y at which $\psi_\nu(y)$ is maximized
z	is set equal to $2y - 1$, where $y = F_\nu(x)$
z_j	j 'th node of the Gauss Legendre quadrature

Greek Alphabet

α	$1 - \alpha$ is the nominal coverage probability
δ	a small positive number, where $\delta \in [0, h)$
ϵ	a desired upper bound on the absolute value of the error of approximation
κ	a positive integer, which arises in the integral of interest in the context of non-standard confidence regions
λ	a smooth bounded real-valued function, where $\lambda : [0, \infty) \rightarrow \mathbb{R}$
ν	a positive integer equal to $\kappa + \xi$
ω	angular frequency
$\psi_\nu(y)$	defined to be $f_\nu(x(y)) dx(y)/dy$
Φ	denotes the $N(0, 1)$ cdf
τ_ν	defined to be $\nu^{\nu/2} / (\Gamma(\nu/2) 2^{(\nu/2)-1})$
ξ	a positive integer, which arises in the integral of interest in the context of non-standard confidence regions

Notation used in Chapter 4

Latin Alphabet

\mathbf{a}	a specified nonzero p -vector, which defines the parameter of interest θ , where $\theta = \mathbf{a}^\top \boldsymbol{\beta}$
b	$b : \mathbb{R} \rightarrow \mathbb{R}$ is an odd continuous function, where $b(x) = 0$ for all $ x \geq d$
b_1	the Kabaila & Giri (2009a) confidence interval for the first definition of the scaled expected length and the standard choice of λ is $\text{CI}(b_1, s_1)$
$b_{1\lambda}$	the Kabaila & Giri (2009a) confidence interval for the first definition of the scaled expected length and a given value of λ is $\text{CI}(b_{1\lambda}, s_{1\lambda})$
b_2	the Kabaila & Giri (2009a) confidence interval for the second definition of the scaled expected length and the standard choice of λ is $\text{CI}(b_2, s_2)$
$b_{2\lambda}$	the Kabaila & Giri (2009a) confidence interval for the second definition of the scaled expected length and a given value of λ is $\text{CI}(b_{2\lambda}, s_{2\lambda})$
\mathbf{c}	a specified nonzero p -vector which defines the parameter τ , where $\tau = \mathbf{c}^\top \boldsymbol{\beta}$
$c(\kappa, \xi)$	defined to be $(\kappa/(\kappa + \xi))^{1/2}$
$\text{CI}(b, s)$	the confidence interval $[\hat{\theta} - v_\theta^{1/2} \hat{\sigma} b(\hat{\gamma}) \pm v_\theta^{1/2} \hat{\sigma} s(\hat{\gamma})]$
$\text{CI}(b_1, s_1)$	the Kabaila & Giri (2009a) confidence interval for the first definition of the scaled expected length and the standard choice of λ
$\text{CI}(b_2, s_2)$	the Kabaila & Giri (2009a) confidence interval for the second definition of the scaled expected length and the standard choice of λ

$\text{CI}(b_{1\lambda}, s_{1\lambda})$	the Kabaila & Giri (2009a) confidence interval for the first definition of the scaled expected length and a given value of λ
$\text{CI}(b_{2\lambda}, s_{2\lambda})$	the Kabaila & Giri (2009a) confidence interval for the second definition of the scaled expected length and a given value of λ
$\text{CP}(\gamma; b, s)$	coverage probability of the confidence interval $\text{CI}(b, s)$
d	a sufficiently large positive number, eventually set equal to $6 t_{m,p}/z_p$
$E(W)$	expected value of W
$f_W(w)$	pdf of W , evaluated at w , where $W = \widehat{\sigma}/\sigma$
$g(z)$	defined to be $\exp(z/2 - e^{-z})$
h	step length in the expression for the approximation to the coverage probability
h_1	step length in the expression for the approximation to first definition of the scaled expected length
h_2	step length in the expression for the approximation to second definition of the scaled expected length
$\text{ICP}(w, \gamma, \rho)$	defined to be $\int_0^d \text{IICP}(x, w, \gamma, \rho) dx$
$\text{IICP}(x, w, \gamma, \rho)$	defined to be $(k(x, w, \gamma, \rho) - k^\dagger(x, w, \gamma, \rho)) \phi(wx - \gamma) + (k(-x, w, \gamma, \rho) - k^\dagger(-x, w, \gamma, \rho)) \phi(wx + \gamma)$
$\text{IOBJ}_1(x; s, \lambda)$	defined to be $(s(x) - t_{m,1-\alpha/2}) \times (\lambda + (1/\sqrt{2\pi})(m/(x^2 + m))^{(m/2)+1})$
$\text{IOBJ}_2(x; s, \lambda)$	defined to be $(s(x) - t_{m,1-\alpha/2}) \times (\lambda + (1/\sqrt{2\pi})(m/(x^2 + m))^{(m+1)/2})$
$\text{ISEL}(w, \gamma)$	defined to be $\int_{-d}^d (s(x) - t_{m,1-\alpha/2}) \phi(wx - \gamma) dx$
$k(x, w, \gamma, \rho)$	defined to be $\Psi\left(w(b(x) - s(x)), w(b(x) + s(x)); \rho(wx - \gamma), 1 - \rho^2)\right)$

$k^\dagger(x, w, \gamma, \rho)$	defined to be $\Psi\left(-t_{m,1-\alpha/2} w, t_{m,1-\alpha/2} w; \rho(wx - \gamma), 1 - \rho^2\right)$
l	length of the interval $[0, d]$, where $l \in \{60, 120\}$
m	defined to be $n - p$
n	length of the response vector \mathbf{y}
N	number of outer integrand evaluations in the approximation to the coverage probability
N_1	number of outer integrand evaluations in the approximation to the first definition of the scaled expected length
N_2	number of outer integrand evaluations in the approximation to the second definition of the scaled expected length
$\text{OBJ}_1(s; \lambda)$	the objective function based on $\text{SEL}_1(\gamma; s)$
$\text{OBJ}_2(s; \lambda)$	the objective function based on $\text{SEL}_2(\gamma; s)$
p	length of the regression parameter vector $\boldsymbol{\beta}$
q	number of knots, where $0 = x_1 < x_2 < \dots < x_q = d$
Q	a random variable with a χ_m^2 distribution
s	$s : \mathbb{R} \rightarrow [0, \infty)$ is an even continuous function where $s(x) = t_{m,1-\alpha/2}$ for all $ x \geq d$
s_1	the Kabaila & Giri (2009a) confidence interval for the first definition of the scaled expected length and the standard choice of λ is $\text{CI}(b_1, s_1)$
$s_{1\lambda}$	the Kabaila & Giri (2009a) confidence interval for the first definition of the scaled expected length and a given value of λ is $\text{CI}(b_{1\lambda}, s_{1\lambda})$
s_2	the Kabaila & Giri (2009a) confidence interval for the second definition of the scaled expected length and the standard choice of λ is $\text{CI}(b_2, s_2)$

$s_{2\lambda}$	the Kabaila & Giri (2009a) confidence interval for the second definition of the scaled expected length and a given value of λ is $\text{CI}(b_{2\lambda}, s_{2\lambda})$
$\text{SEL}_1(\gamma; s)$	defined to be the scaled expected length of $\text{CI}(b, s)$, according to the first definition of the scaled expected length
$\text{SEL}_2(\gamma; s)$	defined to be the scaled expected length of $\text{CI}(b, s)$, according to the second definition of the scaled expected length
t	a specified number
$t_{m,1-\alpha/2}$	$\alpha/2$ 'th quantile of the t_m distribution, defined by $P(T \leq t_{m,\alpha/2}) = \alpha/2$ for $T \sim t_m$
v_θ	compact notation for $\text{var}(\hat{\theta})/\sigma^2$
v_τ	compact notation for $\text{var}(\hat{\tau})/\sigma^2$
W	random variable equal to $\hat{\sigma}/\sigma$, which has the same distribution as $\sqrt{Q/m}$
x	a real-valued variable, where $x \in [-d, d]$
\mathbf{X}	a known $n \times p$ matrix
y	defined to be $c(m, 1) w$
\tilde{y}	defined to be $c(m, 2) w$
\mathbf{y}	response vector
z_ℓ	location of the first evaluation of the integrand, $\text{ICP}(g(z)/c(m, 1), \gamma, \rho) \psi_{m+1}(z)$
\tilde{z}_ℓ	location of the first evaluation of the integrand, $\text{ISEL}(g(z)/c(m, 2), \gamma) \psi_{m+2}(z)$
$\tilde{\tilde{z}}_\ell$	location of the first evaluation of the integrand, $\text{ISEL}(g(z)/c(m, 1), \gamma) \psi_{m+1}(z)$
z_p	p 'th quantile of the standard normal distribution, defined by $P(Z \leq z_p) = p$ where $Z \sim N(0, 1)$

Greek Alphabet

α	the desired minimum coverage probability of the confidence interval is $1 - \alpha$
β	an unknown parameter p -vector
$\hat{\beta}$	the least squares estimator of β
ϵ	a desired upper bound on the absolute value of the error of the approximation
ε	random error vector with $N(\mathbf{0}, \sigma^2 \mathbf{I})$ distribution
γ	defined to be $(\tau - t)/(\sigma v_\tau^{1/2})$
$\hat{\gamma}$	least squares estimator of γ
Γ_{grid}	the set of γ values used in the implementation of the coverage constraint
γ_{max}	the maximum possible value in the set Γ_{grid}
κ	a positive integer variable
λ	positive tuning parameter, which specifies the weight given to the uncertain prior information
ϕ	pdf of $N(0, 1)$
$\psi_{m+1}(z)$	defined to be $f_{m+1}(g(z))dg(z)/dz$
$\Psi(x, y; \mu, v)$	defined to be $P(x \leq Z \leq y)$ for $Z \sim N(\mu, v)$
ρ	the correlation between $\hat{\theta}$ and $\hat{\tau}$
σ^2	unknown error variance
$\hat{\sigma}^2$	least squares estimator of σ^2
τ	a parameter, distinct from θ , where $\tau = \mathbf{c}^\top \beta$
$\hat{\tau}$	least squares estimator of τ
θ	parameter of interest, where $\theta = \mathbf{a}^\top \beta$
$\hat{\theta}$	least squares estimator of θ
ξ	an integer variable belonging to the set $\{1, 2\}$

Notation used in Chapter 5

Latin Alphabet

a_ℓ	defined to be $(1/\sqrt{\pi}) \int_{-\infty}^{z_\ell} (k(z) - q_m(z)) \exp(-z^2) dz$
a_u	defined to be $(1/\sqrt{\pi}) \int_{z_u}^{\infty} (k(z) - q_m(z)) \exp(-z^2) dz$
c_0	defined to be $(\tau/\sqrt{2\pi}\sigma^2)^{1/2}$
c_1	defined to be $\theta y + \mu^2/(4\tau^2)$
c_2	defined to be $y - \mu/(2\tau^2)$
c_3	defined to be $1/(4\tau^2) - 1/(2\sigma^2)$
$c(\theta, \sigma)$	defined to be $\int_{-\infty}^{\infty} g(t; \theta, \sigma) \phi(t; 0, \sigma^2) dt$
$c_m(\theta, \sigma)$	m -node adaptive Gauss-Hermite quadrature approximation to $c(\theta, \sigma)$
$\tilde{c}_M(\theta, \sigma)$	importance sampling estimator of $c(\theta, \sigma)$
e_m	minimized value of $\max_{z \in [z_\ell, z_u]} k(z) - p_m(z) $
$g(t; \theta, \sigma)$	defined to be $\exp((\theta + t)y) / \left(1 + \exp(\theta + t)\right)^J$
$h(t; \theta, \sigma)$	defined to be $(g(t; \theta, \sigma) \phi(t; 0, \sigma^2)) / \phi(t; \mu, \tau^2)$
i	cluster index
J	size of a given cluster
$k(z)$	defined to be $h(\mu + \sqrt{2}\tau z; \theta, \sigma)$
m	number of Gauss-Hermite quadrature nodes
M	number of simulation runs
N	number of clusters
$p_m(z)$	a polynomial of degree $2m - 1$
$q_m(z)$	polynomial $p_m(z)$ of degree $2m - 1$ that minimizes $\max_{z \in [z_\ell, z_u]} k(z) - p_m(z) $
$r(t; \theta, \sigma)$	defined to be $h(t; \theta, \sigma) (\phi(t; \mu, \tau^2))^{1/2}$
\mathbf{v}	an N -vector, where $\mathbf{v} = (v_1, \dots, v_N)$
w_i	i 'th Gauss-Hermite quadrature weight, where $i = 1, \dots, m$

- x covariate for a given cluster
- x_i covariate for the i 'th cluster, for $i = 1, \dots, N$
- y observed response for a given cluster
- y_i observed response for the i 'th cluster, where $y_i \sim \text{Binomial}(J_i, \pi_i)$
for $i = 1, \dots, N$
- z defined to be $(t - \mu)/(\sqrt{2}\tau)$
- z_i i 'th Gauss-Hermite quadrature node, where $i = 1, \dots, m$
- z_ℓ a Gauss-Hermite quadrature node, where $z_\ell \leq z_1$
- z_u a Gauss-Hermite quadrature node, where $z_u \geq z_m$

Greek Alphabet

β_1	intercept parameter in the binomial logistic regression model
$\hat{\beta}_1$	maximum likelihood estimate of β_1
β_2	coefficient of the covariate in the binomial logistic regression model
$\hat{\beta}_2$	maximum likelihood estimate of β_2
$\boldsymbol{\eta}$	a random N -vector, where $\boldsymbol{\eta} = (\eta_1, \dots, \eta_N)$ and η_i 's are independent and identically $N(0, \sigma^2)$ distributed
μ	mode of $g(t; \theta, \sigma) \phi(t; 0, \sigma^2)$ considered as a function of t
π_i	binomial probability for the i 'th cluster
$\phi(t; \mu, \sigma^2)$	an approximation to $g(t; \theta, \sigma) \phi(t; 0, \sigma^2)$
σ	standard deviation of the distribution of η_i
$\hat{\sigma}$	maximum likelihood estimate of σ
$\tilde{\sigma}^2$	defined to be $\int_{-\infty}^{\infty} (g(t; \theta, \sigma) \phi(t; 0, \sigma^2))^2 / \phi(t; \mu, \tau^2) dt - c^2(\theta, \sigma)$
τ^2	defined to be $[-\partial^2 \log (g(t; \theta, \sigma) \phi(t; 0, \sigma^2)) / \partial t^2]^{-1}$
θ	defined to be $\beta_1 + \beta_2 x$

Summary

We consider a general regression model, without a scale parameter. We construct a confidence interval for a scalar parameter of interest that utilizes the uncertain prior information that a distinct scalar parameter takes a specified value. This confidence interval has good coverage properties. It also has scaled expected length, where the scaling is with respect to the usual confidence interval, that is (a) substantially less than 1 when the prior information is correct, (b) has a maximum value that is not too large and (c) is close to 1 when the data and prior information are highly discordant. We use the R package `ciuupi`, which computes a confidence interval that utilizes uncertain prior information in a linear regression model with known error variance, to build this confidence interval.

We also solve the problem of numerically evaluating the expected value of a smooth bounded function of a chi-distributed random variable, divided by the square root of the number of degrees of freedom, using Mori's transformation followed by the trapezoidal rule, which is exponentially convergent for suitable integrands. This problem arises in simultaneous inference, selection and ranking of populations, the evaluation of multivariate t probabilities and the assessment of the coverage and expected volume properties of non-standard confidence regions.

We apply this solution in the new R package `ciuupi2` that computes the [Kabaila & Giri \(2009a\)](#) confidence interval, which utilizes uncertain prior information in a linear regression model with unknown error variance. Pre-

vious computations of this interval used **MATLAB** programs that were time-consuming to run. By writing these programs in **R**, the computation time is greatly reduced and they become freely available. We also assess a new definition of scaled expected length.

Finally, we compare the computations of the log-likelihood function for generalized linear mixed models using (a) adaptive Gauss-Hermite quadrature and (b) importance sampling, where both methods share the same initial step.

Acknowledgements

Foremost, I would like to express my heartiest gratitude to my principal supervisor, Assoc. Prof. Paul Kabaila, for his precious assistance, support and the guidance offered me in order to achieve aims and objectives throughout the process of my PhD study. And also I would like to thank my co-supervisor Dr. David Farchione for the assistance to make my efforts a success.

Besides my supervisors, I would like to thank the chair of my research progress panel, Dr. Hien Nguyen and the Assoc. Head of School of Engineering and Mathematical Sciences, Prof. Luke Prendergast for their insightful comments and encouragement, throughout this period. I am grateful to acknowledge all the staff members at Department of Mathematics and Statistics for providing me support and valuable information from various perspectives. This work was supported by a La Trobe University Full Fee Research Scholarship, a La Trobe University Graduate Research Scholarship and an Australian Government Research Training Program Scholarship.

To conclude, I am very thankful for my parents, my brother and my friends for their unconditional support and understanding in many ways in this very intense four year period of time. Specially, I am very grateful to my caring and loving husband, Ganganath, for his enormous support given me in every possible way.

Statement of Authorship

Except where reference is made in the text of the thesis, this thesis contains no material published elsewhere or extracted in whole or in part from a thesis submitted for the award of any other degree or diploma.

No other person's work has been used without due acknowledgement in the main text of the thesis.

This thesis has not been submitted for the award of any degree or diploma in any other tertiary institution.

Nishika Oshadini Ranathunga Kapuruge

11 February 2021

Chapter 1

Introduction

1.1 Literature review

Uncertain prior information about the values of the parameters of a model may result from previous experience with similar data sets and/or expert view and/or scientific concepts. For example, it is widely believed, see [Mead \(1988\)](#) and [Wu & Hamada \(2000\)](#), that for a factorial design the higher the order of an interaction term the more likely it is that it differs negligibly from zero. In a linear regression setting it is commonly believed that higher order interaction terms are likely to be zero. In this setting, [Pratt \(1965\)](#), [Cohen \(1974\)](#) and [Leamer \(1978\)](#) suggested that a preliminary test for an interaction term being zero seems to be motivated by a desire to utilize uncertain prior information that this term is zero.

The utilization of such prior information in frequentist inference was initiated by [Hodges & Lehmann \(1952\)](#). These authors considered frequentist inference in which they sought “to utilize the available information”, while providing “a safeguard in case this information is not correct”. They used a decision-theoretic methodology that was extended by [Kempthorne \(1983, 1987, 1988\)](#) to more complicated uncertain prior information scenarios.

To say that a frequentist confidence region for the (vector) parameter

of interest utilizes the uncertain prior information it must, at a minimum, satisfy the following two requirements. It must have excellent coverage properties. It must also have a scaled expected volume, where the scaling is with respect to the usual confidence region with the same nominal coverage, that is substantially less than one when the prior information is correct.

The econometrician [Leamer \(1978\)](#) pointed out that the preliminary data-based model selection may be motivated by a desire to utilize uncertain prior information in subsequent inference. However, the confidence intervals constructed after data-based model selection may have very unsatisfactory coverage properties and so fail to utilize the uncertain prior information. The literature on the coverage properties of such post-model-selection confidence intervals is reviewed by [Kabaila \(2009\)](#). Frequentist model averaged confidence intervals ([Buckland *et al.*, 1997](#), [Fletcher & Turek, 2011](#), [Turek & Fletcher, 2012](#), [Efron, 2014](#)) may, using an extension of the thinking of [Leamer \(1978\)](#) and [Kabaila \(2009\)](#) on post-model-selection inference, also be viewed as attempts to utilize uncertain prior information. [Fletcher \(2018\)](#) presents numerous examples of the application of frequentist model averaged confidence intervals to real-life data.

[Kabaila *et al.* \(2016\)](#) examined the performances of the model averaged profile likelihood confidence interval (MPI), due to [Fletcher & Turek \(2011\)](#), and the model averaged tail area (MATA) confidence interval, due to [Turek & Fletcher \(2012\)](#), using a testbed scenario consisting of two nested linear regression models. We refer to the two models in this testbed scenario as the full model and the simpler model. They showed that in some cases, MPI intervals have poorer performance than the post-model-selection confidence interval with same nominal coverage probability. They also showed that the MATA confidence interval performs “better than the MPI and post-model-selection confidence intervals”. However, their overall finding was that it

seemed to be difficult to find cases in which the MATA confidence interval performed substantially better than the standard confidence interval (based on the full model) with the same nominal coverage probability. The numerical performance results presented by [Kabaila *et al.* \(2016\)](#) are for the case of model weights based on the Akaike Information Criterion (AIC). [Kabaila *et al.* \(2017\)](#) examined the performance of MATA intervals using the same testbed scenario of two nested linear regression models for a family of data-based weights on the two models. This family is indexed by a scalar tuning parameter that defines the data-based weight given to the simpler model. They find that the MATA interval perform best when this tuning parameter corresponds to the minimum possible weight on the simpler model. [Kabaila \(2018\)](#) assessed MATA interval for a large number of linear models and for a variety of data-based model weights. He provided an easily-computed upper bound on the minimum coverage probability of this interval. He used this upper bound to argue against the use of weights based on the Bayesian Information Criterion (BIC).

Consider the same testbed scenario consisting of two nested linear regression models, with error variance assumed to be known. Also, consider a preliminary test of the null hypothesis that the simpler model is correct. In this scenario, [Kabaila & Wijethunga \(2019a\)](#) assessed the performance of a confidence interval centred on the bootstrap smoothed estimator which has a width proportional to the estimate of the delta method approximation, derived by [Efron \(2014\)](#), to the standard deviation of this estimator. They also consider a confidence interval centred on the bootstrap smoothed estimator, but with width proportional to the estimate of the actual standard deviation of the bootstrap smoothed estimator. They assessed the performance of these intervals in terms of coverage probability and scaled expected length, where scaling is with respect to the usual confidence interval based on the full model

with the same minimum coverage. When considering coverage properties, both of these intervals dominate the post-model-selection confidence interval, with the same nominal coverage. However, these two intervals do not have the desired scaled expected length properties. [Kabaila & Wijethunga \(2019b\)](#) extended the assessment of [Kabaila & Wijethunga \(2019a\)](#) to the same testbed situation, but for the case of unknown error variance. They considered only the confidence interval centred on the bootstrap smoothed estimator, with width proportional to the estimate of the delta method approximation to the standard deviation of this estimator. They observed that for the variance unknown case, the scaled expected length shows some appealing features when the degrees of freedom for the estimation of the error variance is small.

[Kabaila *et al.* \(2020\)](#) explored confidence intervals centred on the frequentist model averaged estimator, proposed by [Buckland *et al.* \(1997\)](#), using the testbed scenario of two nested linear regression models, with unknown error variance. They considered two forms of standard error of this estimator from [Buckland *et al.* \(1997\)](#) and [Burnham & Anderson \(2002\)](#). They also consider four procedures for obtaining the half-width of the confidence interval from the chosen standard error. Consequently, they consider eight model averaged confidence intervals. They derived computationally convenient exact formulas for the coverage probability and scaled expected length of each of these eight confidence intervals. Numerical evaluations show that these eight confidence intervals do not have satisfactory finite sample performance. This fact, combined with the large sample result of [Hjort & Claeskens \(2003\)](#), shows that these confidence intervals do not have satisfactory performances in general. To summarise, the performance of frequentist model averaged confidence intervals in terms of the utilization of uncertain prior information (through favourable scaled expected length properties) is very patchy.

Suppose that the parameter of interest is a scalar, there are no nuisance parameters and there is uncertain prior information about this parameter of interest. [Pratt \(1961, 1963\)](#) devised an elegant method, which makes clever use of the Neyman-Pearson lemma, for the construction of a confidence interval for this parameter that utilizes the uncertain prior information. Now suppose that the parameter of interest is a scalar, there is a pivotal quantity for this parameter and there is uncertain prior information about this parameter of interest. The easily-understood “tail method” ([Stein, 1962](#), [Bartholomew, 1971](#), [Puza & O’Neill, 2006b](#), [Puza & O’Neill 2006a](#), [Puza & Yang, 2016](#), [Yang & Puza, 2020](#)) can be used to construct confidence intervals that utilize the uncertain prior information.

[Puza & O’Neill \(2006b\)](#) applied the “tail method” to obtain confidence intervals for the normal mean (for both known and unknown variance) and the binomial success probability that have smaller expected length than the usual confidence intervals in specified parts of the parameter space. This decrease in expected length comes at the cost of an increase in expected length in other parts of the parameter space. [Puza & O’Neill \(2006a\)](#) applied the “tail method” to a randomized confidence interval for the binomial success probability. [Puza & Yang \(2016\)](#) used the “tail method” to construct frequentist confidence intervals for the mean of the exponential distribution. Using a similar approach, [Yang & Puza \(2020\)](#) constructed frequentist confidence intervals for the parameter of the geometric distribution. Both of these papers show that these confidence intervals outperform the usual confidence intervals in some parts of the parameter space. A novel extension of the “tail method” was put forward by [Yu & Hoff \(2018\)](#) and [Hoff & Yu \(2019\)](#). [Yu & Hoff \(2018\)](#) used this novel method for the construction of confidence intervals for the treatment means in a one-way layout that utilize the uncertain prior information that these treatment means are the same.

Henceforth, to qualify as a genuine confidence region that utilizes the uncertain prior information this region must satisfy the following additional requirement. The scaled expected volume (a) must have a maximum value that is not too large and (b) be close to 1 when the data and the prior information are in conflict. Such regions include (a) confidence regions for the multivariate normal mean that dominate the usual confidence region ([Casella & Hwang, 1987, 2012](#)) and (b) confidence intervals for the normal variance that dominate the usual confidence interval ([Maata & Casella, 1990](#)).

[Farchione & Kabaila \(2008\)](#) constructed frequentist confidence intervals for the normal mean that utilize the uncertain prior information that this mean takes a specified value. They considered both the known and unknown variance cases. These intervals have a scaled expected length that is substantially less than 1 when the prior information is correct and have a scaled expected length with maximum value that is not too large. In addition, these intervals reduce to the usual confidence interval when the data greatly disagree with the prior information. [Kabaila & Giri \(2009a\)](#) considered a linear regression model, with unknown error variance, and a parameter of interest that is a specified linear combination of the regression parameters. They supposed that there is uncertain prior information that a different linear combination of the regression parameters takes a specified value. They showed how to construct a confidence interval for the scalar parameter of interest that utilizes this uncertain prior information. They illustrated the application of this confidence interval to the case of data from a 2×2 factorial experiment with more than one replicate, the parameter of interest a specified simple effect and the uncertain prior information that the two-factor interaction is zero. [Kabaila & Giri \(2014\)](#) considered data from a 2×2 factorial experiment and constructed simultaneous frequentist confidence intervals for the four population cell means that utilized the same uncertain prior infor-

mation. [Kabaila & Tissera \(2014\)](#) considered a linear regression model, with unknown error variance, parameter of interest a specified linear combination of the regression parameters and uncertain prior information that a vector of regression parameters, with dimension two or greater, took the value zero. They showed how to construct a confidence interval for this parameter of interest that utilizes this uncertain prior information. [Abeysekera & Kabaila \(2017\)](#) assessed the coverage and scaled expected volume properties of recentered confidence spheres for the mean of a multivariate normal distribution with a diagonal covariance matrix whose diagonal elements take the same value. They considered both the case that this common value is known and that it is unknown. They supposed that the uncertain prior information is that the mean of this distribution is zero. By carefully choosing the data-based radius and centre of these confidence spheres they were able to obtain confidence spheres for the mean that utilize this uncertain prior information.

The confidence regions that utilize uncertain prior information put forward by [Farchione & Kabaila \(2008\)](#), [Kabaila & Giri \(2009a\)](#), [Kabaila & Giri \(2014\)](#), [Kabaila & Tissera \(2014\)](#) and [Abeysekera & Kabaila \(2017\)](#) were all found using numerical nonlinear constrained optimizations implemented in **MATLAB** programs. This made these computations unavailable to statisticians who do not have a **MATLAB** licence. Also, these **MATLAB** programs were slow because they used the **MATLAB** function `dblquad` for two-dimensional numerical integration. This function has since been superseded in **MATLAB** by a much more efficient numerical integration function `integral2`. To make these confidence regions more widely accessible to statisticians it is important to implement these methods, where possible, in **R** programs. [Mainzer & Kabaila \(2019\)](#) considered a linear regression model, with known error variance, that the parameter of interest is a given linear combination of the regression parameters and the uncertain prior information that a distinct

linear combination of these parameters takes a specified value. This is the same scenario as that considered by [Kabaila & Giri \(2009a\)](#), except that the error variance is assumed known. The case where the error variance is assumed known can be regarded as an approximation to the case where the error variance must be estimated, and the number of degrees of freedom for its estimation is large. [Mainzer & Kabaila \(2019\)](#) describe their R package `ciuupi` which computes the confidence interval for this parameter of interest that utilizes this uncertain prior information.

Further developments of the numerical nonlinear constrained optimization approach to the construction of confidence intervals that utilize uncertain prior information will be in the following three areas. Firstly, finding further examples where uncertain prior information can very reasonably be assumed to exist. Secondly, expanding the range of models and types of uncertain prior information for which confidence intervals that utilize uncertain prior information can be constructed. Thirdly, the implementation of the computations of these confidence intervals in R, to make them widely accessible. In this thesis, we make contributions to each of these three areas.

1.2 Thesis overview

This thesis consists of five chapters. The first chapter presents the introduction and the remaining four chapters are described as follows.

Chapter 2

The most significant contributions of this thesis is in Chapter 2. In this chapter, we consider a general regression model without a scale parameter. An example for a general regression model, without a scale parameter is a generalized linear model with binomial responses and canonical link function. We suppose that we have uncertain prior information that a specified scalar

parameter takes a given value. Our aim is to construct a confidence interval for a distinct scalar parameter that utilizes this uncertain prior information. The asymptotic joint distribution of the maximum likelihood estimators of these two parameters is similar to the joint distribution of these estimators in the particular case of a linear regression with normally distributed errors having known variance. This similarity is used to construct a new confidence interval for the parameter of interest, using the output from the R package `ciuppi` described by [Mainzer & Kabaila \(2019\)](#). This confidence interval has good coverage properties and has a scaled expected length that (a) is substantially less than one when the prior information is correct, (b) has a maximum value that is not too large and (c) approaches one when the data and the prior information are highly contradictory.

An important practical application of the new confidence interval described in Chapter 2 is to a quantal bioassay carried out to compare two similar compounds. [Finney \(1955\)](#), [Morgan \(1992\)](#) and [Robertson *et al.* \(2017\)](#) noted that such bioassays have broad applications in pharmacological and toxicological studies. In these types of assays, we commonly have uncertain prior information that the hypothesis of “parallelism” holds. For those cases, a confidence interval that utilizes this uncertain prior information, which we construct in our body of work, is an appealing option.

[Finney \(1964\)](#) emphasised the importance of pre-existing knowledge of prior information when designing parallel line assays when he stated that “Without a *priori* knowledge that the effective stimuli in the standard and test preparations are qualitatively identical, the employment of 4-point designs is sheer obscurantism”. [Finney \(1955\)](#) stated that “Unless previous experience of an assay technique gives very strong reasons for believing that the assumptions of linearity and parallelism are correct, 4-point assays provide inadequate evidence for testing conditions that are essential to the validity

of the analysis.” [Khan \(2003\)](#) and [Khan \(2008\)](#) considered the point estimation of the parameters in two normal straight line models in the presence of uncertain prior information that the slopes of these straight lines are the same i.e. that the hypothesis of “parallelism” holds.

Chapter 3

In Chapter 3, we consider the computation of the expected value of a smooth bounded function of a chi-distributed random variable, divided by the square root of the number of degrees of freedom. This type of computation occurs in the evaluation of the coverage probabilities and scaled expected volumes of post-model-selection confidence intervals, frequentist model-averaged confidence intervals and other non-standard confidence regions ([Farchione & Kabaila, 2008](#), [Kabaila & Giri, 2009a](#), [Kabaila & Giri, 2009b](#), [Kabaila & Farchione, 2012](#), [Kabaila & Giri, 2013](#), [Kabaila & Tissera, 2014](#), [Kabaila *et al.*, 2016](#), [Kabaila *et al.*, 2017](#), [Abeysekera & Kabaila, 2017](#) and [Kabaila, 2018](#)). In all of these papers, this evaluation has previously been carried out by first truncating the integral (the truncation error is easily bounded) and then applying an adaptive numerical integration method. We seek a more efficient method for this evaluation. The new and more efficient method that we describe in Chapter 3 is applied in Chapter 4 of this thesis to the computation of the [Kabaila & Giri \(2009a\)](#) confidence interval that utilizes uncertain prior information.

We use the transformation (2.6) of [Mori \(1988\)](#), followed by the application of the trapezoidal rule. As noted by [Trefethen & Weideman \(2014\)](#), the trapezoidal rule has the remarkable property that, for suitable integrands, it is exponentially convergent. The trapezoidal rule has the advantage that it leads to a nested sequence of quadrature rules that can be used for estimation of the approximation error.

We describe the properties of the trapezoidal rule using the Fourier trans-

form of the integrand and the Poisson summation formula. The application of the trapezoidal rule requires the approximation of an infinite sum by a finite sum. We provide a new easily computed upper bound on the error of this approximation. We describe two procedures for the computation of the integral of interest. The first of these is a simple procedure that is expected to have good properties in practice. The second of these procedures is a slight modification of this procedure which is proved to be exponentially convergent. We also compare the performance of this simple procedure with three other methods: (a) Gauss Legendre quadrature, (b) Generalized Gauss Laguerre quadrature and (c) the inverse cdf method, using the simple test scenario that consists of evaluating a known univariate t probability.

Chapter 4

[Kabaila & Giri \(2009a\)](#) considered a linear regression model, with unknown error variance. They constructed a frequentist confidence interval for a specified linear combination of regression parameters that utilizes the uncertain prior information of a different linear combination of the regression parameters is equal to a specified value.

This confidence interval was previously implemented using **MATLAB** programs, which were time-consuming to run and which cannot be used without a **MATLAB** licence. In [Chapter 4](#), we describe several computational innovations that we have used to write the **R** programs for more efficient computation of this confidence interval. These **R** programs have been placed in the **R** package `ciuupi2`. The confidence intervals and their properties previously calculated using **MATLAB** programs were re-calculated using `ciuupi2` and were in good agreement with those previously calculated results. It is planned to make the **R** package `ciuupi2` available on CRAN after submission of the thesis for examination. In [Chapter 4](#) we also carry out an important theoretical and computational analysis of a second (new) definition of the

scaled expected length.

The computation of the [Kabaila & Giri \(2009a\)](#) confidence interval requires evaluating double integrals in the expressions for the coverage probability and the scaled expected length. The inner integrals of these double integrals have some special features that lead us first to divide them into a sum of integrals and then apply Gauss Legendre quadrature to each integral in this sum. [Mainzer & Kabaila \(2019\)](#) used Gauss Legendre quadrature with five nodes for a similar, but more straightforward, computation. We use the new method described in Chapter 3 to approximate the outer integrals of these double integrals. Computation of the [Kabaila & Giri \(2009a\)](#) confidence interval also requires numerical nonlinear constrained optimization. [Kabaila & Giri \(2009a\)](#) used the MATLAB function `fmincon` to perform this computation using the sequential quadratic programming option. We carry out sequential quadratic programming in R using the function `slsqp` in the `nloptr` package ([Johnson \(2014\)](#)).

Chapter 5

As we have seen from the work in Chapter 3, the efficient numerical evaluation of integrals requires a careful analysis. In Chapter 5 of this thesis, we explore the numerical evaluation of integrals in the context of the computation of the log-likelihood function for generalized linear mixed models.

Adaptive Gauss-Hermite quadrature can be used for the computation of the log-likelihood function for generalized linear mixed models. For Gauss-Hermite quadrature, [Liu & Pierce \(1994\)](#) proposed a method to transform the variable of integration in such a way that the integrand is sampled at relatively important values. [Pinheiro & Bates \(1995\)](#) referred to this method put forward by [Liu & Pierce \(1994\)](#) as ‘adaptive’ Gauss-Hermite quadrature. This method has found applications in the computation of the log-likelihood function for generalized linear mixed models ([Lesaffre & Spiessens, 2001](#),

Demidenko, 2004, Hedeker & Gibbons, 2006, Tuerlinckx *et al.* , 2006, Rabe-Hesketh & Skrondal, 2008, Kim *et al.* , 2013 and Chang & Hoaglin, 2017).

The basic first step in this method is to multiply and divide the integrand of interest by a carefully chosen probability density function. The same first step is used for the computation of this log-likelihood function using simulations that employ importance sampling. As is well-known, importance sampling needs to be applied with extreme care to be successful (Robert & Casella, 2004 and Owen, 2013). We compare these two methods by considering in detail a single cluster from a well-known teratology data set that is modelled using a logistic regression with random intercept. We show that while importance sampling fails for this computation, adaptive Gauss-Hermite quadrature does not. We derive a new upper bound on the error of approximation of adaptive Gauss-Hermite quadrature. Using this new upper bound, we show that the feature of this problem that makes importance sampling fail is useful in disclosing why adaptive Gauss-Hermite quadrature succeeds.

1.2.1 Outline of the contributions due solely to the author

Chapter 2

The work described in this chapter appears in Kabaila & Ranathunga (2020) and has been submitted for publication.

Section 2.4: The standard confidence interval and the confidence interval that utilizes uncertain prior information are defined in Section 2.2 in the context of a linear regression model with known error variance. In Section 2.4, we derive analogues of these confidence intervals, based on Wald statistics, in a general regression model without a scale parameter.

Subsection 2.9.1: A local asymptotic framework is described in subsection 2.6.1. In this framework the regression parameter vector lies on a straight line segment centred at a particular value. A procedure for a data-based choice of this particular value is described in subsection 2.6.2. In subsection 2.9.1, we apply this procedure in the context of the quantal bioassay dataset from Grewal (1952) for the comparison of the analgesic properties of Morphine and Amidone in mice.

Subsection 2.9.2: Choice of the length of the straight line segment of regression parameter vector values that specifies the local asymptotic framework. This choice is based on an assessment, using Monte Carlo simulation, of values of this parameter vector for which the profile likelihood confidence interval is sometimes extremely long.

Subsections 2.10.1 - 2.10.2: Numerical evaluation, using Monte Carlo simulations, of the *local coverage probability* and the *local scaled expected length* of the likelihood-based confidence interval that utilizes uncertain prior information in a general regression model without a scale parameter.

Appendix A.2: Numerical evaluation, using Monte Carlo simulations, of the *local coverage probability* and the *local scaled expected length* of the Wald-based confidence interval that utilizes uncertain prior information in a general regression model without a scale parameter.

Appendix A.5: The two functions used in the specification of the confidence interval that utilizes uncertain prior information, in the context of a linear regression model with known error variance, are computed using the R package `ciuupi`. Using values computed using this package, we set up a *look-up table* for the fast evaluation of these two functions.

Subsection 2.10.3: Numerical evaluation of the large sample approximation to the *local coverage probability* and the *local scaled expected length* of the

likelihood-based confidence interval that utilizes uncertain prior information, for a general regression model without a scale parameter.

Appendix A.8: The ratio of the length of the likelihood-based confidence interval that utilizes uncertain prior information to the length of the profile likelihood confidence interval computed from the same data, for a general regression model without a scale parameter, is defined in subsection 2.10.2. In Appendix A.8, we analyse the probability distribution of this ratio using Monte Carlo simulations.

Appendix A.9: R programs for the computation of the coverage probabilities of the profile likelihood confidence interval and the likelihood-based confidence interval that utilizes the uncertain prior information and the scaled expected length of the latter confidence interval, for a general regression model without a scale parameter.

Chapter 3

The work described in this chapter appears in Kabaila & Ranathunga (2021).

Subsection 3.4.1 and Appendix B.2: Application of the transformation (2.6) of Mori (1988) to the integral of interest.

Subsection 3.4.2: Proof of Lemma 3.4.1, an upper bound on the *trimming error*, using the *integral test* for series convergence.

Section 3.5: A simple and effective procedure, similar to that described by Mori (1988), for the evaluation of the integral of interest is described in subsection 3.4.3. In Section 3.5, we apply this method to evaluate the integral of interest for the simple test scenario that consists of evaluating a known univariate t probability.

Subsection 3.6.1: Derivation of formulas to evaluate the integral of interest using the Gauss-Legendre quadrature method. Using R programs, calculation

of the approximation error resulting from this procedure and summarising the results. This is in the context of the simple test scenario that consists of evaluating a known univariate t probability.

Subsection 3.6.2 and Appendix B.4: Derivation of formulas to evaluate the integral of interest using the Generalized Gauss Laguerre quadrature method. Using R programs, calculation of the approximation error resulting from this procedure and summarising the results. This is in the context of the simple test scenario that consists of evaluating a known univariate t probability.

Subsection 3.6.3 and Appendix B.5: Derivation of formulas to evaluate the integral of interest using the inverse cdf method followed by the Gauss Legendre quadrature. Using R programs, calculation of the approximation error resulting from this procedure and summarising the results. This is in the context of the simple test scenario that consists of evaluating a known univariate t probability.

Appendix C.7: R programs for the computation of the integral of interest by applying the methods (a) the transformation (2.6) of [Mori \(1988\)](#), followed by the application of the trapezoidal rule, (b) Gauss Legendre quadrature, (c) Generalized Gauss Laguerre quadrature and (d) the inverse cdf method followed by Gauss Legendre quadrature.

Chapter 4

Subsection 4.3.2 and Appendix C.1.2: Application of the new method described in Chapter 3 to the evaluation of the outer integrals of the double integrals in the formulas for the coverage probability and the scaled expected length of the [Kabaila & Giri \(2009a\)](#) confidence interval.

Subsection 4.4.2, Appendix C.2.1 and Appendix C.2.2: Derivation of computationally convenient formulas for the second (new) definition of the

scaled expected length and the resulting objective function.

Appendix C.2.3 and Appendix C.2.4: Application of the new method described in Chapter 3 to the evaluation of the outer integral of the double integral term in the formula for the scaled expected length. This is in the context of the second definition of the scaled expected length.

Subsection 4.5.2, Appendix C.3 and Appendix C.3.1: The computation of the Kabaila & Giri (2009a) confidence interval that utilizes uncertain prior information requires the choice of a number of computationally-related quantities. In most cases these quantities must be chosen to be sufficiently large so that the performance of this confidence interval is not degraded. On the other hand, values of these quantities that are too large can lead to excessive computation times and, potentially, unstable computed results. Formulas for the choice of some of these quantities are presented in subsection 4.5.1. In subsection 4.5.2, Appendix C.3 and Appendix C.3.1, we choose the remaining computationally-related quantities, based on numerical evidence.

Subsection 4.5.3 and Appendices C.4 - C.5: Computational analysis for choosing the number of outer integrand evaluations and the number of Gauss Legendre quadrature nodes.

Subsection 4.5.4 and Appendix C.6: Numerical comparison of the results obtained using the new R package `ciupl2` with some of the past results obtained by Kabaila & Giri (2009a), Kabaila & Giri (2013) and Giri (2008) for a given choice of the tuning parameter λ , which specifies the weight given to the uncertain prior information, using MATLAB programs.

Subsections 4.6.1 - 4.6.2: Evaluate the efficiency of the Kabaila & Giri (2009a) confidence interval for the “standard” choice of the tuning parameter λ , using the new R package `ciupl2`.

Section 4.7: A numerical comparison of the two definitions of scaled ex-

pected length in terms of the resulting confidence intervals, for the “standard” choice of the tuning parameter λ , that utilize uncertain prior information and their coverage and scaled expected lengths properties.

Appendix C.7: R programs for the efficient computation of the [Kabaila & Giri \(2009a\)](#) confidence interval.

Chapter 5

The work described in this chapter appears in [Kabaila & Ranathunga \(2019\)](#).

Section 5.2 and Appendix D.5: Derivation of an expression for the log-likelihood function of a logistic regression model with random intercept.

Appendix D.1: Details of a particular step in the derivation of the adaptive Gauss-Hermite quadrature formula, using Theorem 5.1 of [Carlin & Louis \(1996\)](#); [Abeysekara \(2014\)](#), are provided.

Section 5.3 and Appendix D.2: Further details of the derivation of an expression for the adaptive Gauss-Hermite quadrature approach for the computation of the log-likelihood function of the logistic regression model with random intercept.

Section 5.4: For a logistic regression model with random intercept, the log-likelihood function is the sum of the logarithms of similar terms, one term per cluster. We evaluate the performance of importance sampling for the computation of a specified term (corresponding to a specified cluster) for the teratology dataset of [Weil \(1970\)](#).

Section 5.5: For the same situation as that considered in Section 5.4, we evaluate the performance of the adaptive Gauss-Hermite quadrature using a program written in Maple.

Appendix D.8: R programs for the computation of the maximum likelihood estimates of the parameters of the logistic regression model with random

intercept. R programs for the evaluation of the performance of importance sampling.

Chapter 2

Confidence intervals in general regression models that utilize uncertain prior information

2.1 Introduction

In this chapter, we consider a general regression model that does not include a scale parameter. We construct a confidence interval for a scalar parameter of interest θ that utilizes the uncertain prior information that a distinct scalar parameter τ takes the specified value t .

The asymptotic joint distribution of the maximum likelihood estimators of θ and τ is similar to the joint distribution of these estimators in the particular case of a linear regression with normally distributed errors having known variance. In the latter case, the R package `ciuupi` is used to compute a confidence interval for θ that utilizes the uncertain prior information that $\tau = t$. We use this confidence interval to construct a new confidence interval for θ that utilizes the uncertain prior information in the case of a general regression model without a scale parameter.

In Section [2.2](#), we describe this confidence interval (constructed using the

package `ciuupi`) in detail. In Section 2.3, we derive the asymptotic distribution of the maximum likelihood estimators θ and τ in the case of a general regression model, without a scale parameter. In Sections 2.4 and 2.5, we express the confidence interval for θ that utilizes the uncertain prior information in the case of a general regression model without a scale parameter in terms of Wald statistics and signed root likelihood ratio (SRLR) statistics, respectively.

We also define the *local coverage probability* (Section 2.6) and the *local scaled expected length* (Section 2.7) of a confidence interval for θ using a *local asymptotic framework*, similar to the “local misspecification framework” of Hjort & Claeskens (2003) and Claeskens & Hjort (2008) which was used by Kabaila & Kuveke (2019). The confidence interval that we construct in this chapter has the following desirable characteristics.

- (1) It has endpoints that are smooth functions of the data.
- (2) It has *local coverage probability* that is close to $1 - \alpha$.
- (3) It has *local scaled expected length* that (a) is substantially less than 1 when the prior information that $\tau = t$ is correct, (b) has a maximum value that is not too large and (c) approaches 1 for large $|\tau - t|$.

In Section 2.8, we provide a detailed description of a quantal bioassay designed to compare two similar compounds. In this context, we have uncertain prior information that the hypothesis of “parallelism” holds, which can be expressed in the form τ takes the specified value t . In Sections 2.9 and 2.10, we provide extensive numerical results that illustrate the attractive properties of the confidence interval that we construct in this chapter. The work described in this chapter appears in Kabaila & Ranathunga (2020) and has been submitted for publication.

2.2 The confidence interval that utilizes uncertain prior information in linear regression with known error variance

Consider the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where \mathbf{y} is a random n -vector of responses, \mathbf{X} is a known $n \times p$ matrix with linearly independent columns, $\boldsymbol{\beta}$ is an unknown parameter p -vector and $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$, where σ^2 is known. Suppose that the parameter of interest is $\theta = \mathbf{a}^\top \boldsymbol{\beta}$, where \mathbf{a} is a specified nonzero p -vector. Let $\tau = \mathbf{c}^\top \boldsymbol{\beta}$, where \mathbf{c} is a specified nonzero p -vector that is linearly independent of \mathbf{a} . Suppose that we have uncertain prior information that $\tau = t$, where t is a specified number (commonly $t = 0$). Our aim is to construct a confidence interval for θ , with minimum coverage probability $1 - \alpha$, that utilizes this uncertain prior information.

Let $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$, the least squares estimator of $\boldsymbol{\beta}$. Then $\hat{\theta} = \mathbf{a}^\top \hat{\boldsymbol{\beta}}$ and $\hat{\tau} = \mathbf{c}^\top \hat{\boldsymbol{\beta}}$ are the least squares estimators of θ and τ , respectively. Note that $\text{var}(\hat{\theta}) = \sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a}$, $\text{var}(\hat{\tau}) = \sigma^2 \mathbf{c}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{c}$ and $\text{cov}(\hat{\theta}, \hat{\tau}) = \sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{c}$, which are known quantities. Hence

$$\rho = \text{corr}(\hat{\theta}, \hat{\tau}) = \frac{\text{cov}(\hat{\theta}, \hat{\tau})}{(\text{var}(\hat{\theta}) \text{var}(\hat{\tau}))^{1/2}} \quad (2.1)$$

is also known.

Our first step in the description of the CI for θ that utilizes the uncertain prior information is to reduce the data to $(\hat{\theta}, \hat{\tau})$. A justification for this data reduction is provided by the change of parametrization described in Section 4 of the Supplementary Material for [Kabaila *et al.* \(2016\)](#) with $t = 0$. Observe

that

$$\begin{bmatrix} \hat{\theta} \\ \hat{\tau} \end{bmatrix} \sim N \left(\begin{bmatrix} \theta \\ \tau \end{bmatrix}, \begin{bmatrix} \text{var}(\hat{\theta}) & \text{cov}(\hat{\theta}, \hat{\tau}) \\ \text{cov}(\hat{\theta}, \hat{\tau}) & \text{var}(\hat{\tau}) \end{bmatrix} \right). \quad (2.2)$$

Let $[a \pm w]$ denote the interval $[a - w, a + w]$ ($w > 0$). The usual $1 - \alpha$ confidence interval for θ is

$$I = \left[\hat{\theta} \pm z_{1-\alpha/2} (\text{var}(\hat{\theta}))^{1/2} \right], \quad (2.3)$$

where the quantile z_p is defined by $P(Z \leq z_p) = p$ for $Z \sim N(0, 1)$.

The confidence interval for θ computed by the R package `ciuupi`, with minimum coverage probability $1 - \alpha$, that utilizes the uncertain prior information that $\tau = t$, has the form

$$CI(b, s) = \left[\hat{\theta} - (\text{var}(\hat{\theta}))^{1/2} b \left(\frac{\hat{\tau} - t}{(\text{var}(\hat{\tau}))^{1/2}} \right) \pm (\text{var}(\hat{\theta}))^{1/2} s \left(\frac{\hat{\tau} - t}{(\text{var}(\hat{\tau}))^{1/2}} \right) \right], \quad (2.4)$$

where $b : \mathbb{R} \rightarrow \mathbb{R}$ is an odd continuous function and $s : \mathbb{R} \rightarrow \mathbb{R}$ is an even continuous function. In addition, $b(x) = 0$ and $s(x) = z_{1-\alpha/2}$ for all $|x| \geq 6$. Define the scaled expected length of $CI(b, s)$ to be

$$E(\text{length of } CI(b, s)) / (\text{length of } I).$$

The R package `ciuupi` computes the functions b and s such that $CI(b, s)$ has the following properties. It has minimum coverage probability $1 - \alpha$ and scaled expected length for $\tau = t$ that is as small as possible, subject to an upper bound on its maximum value. [Mainzer & Kabaila \(2019\)](#) describe this computation in full detail. The functions b and s that are computed by `ciuupi` are determined by ρ and $1 - \alpha$. We denote them by b_ρ and s_ρ , respectively, so that the confidence interval computed by the R package `ciuupi` is $CI(b_\rho, s_\rho)$.

Let $\gamma = (\tau - t) / (\text{var}(\hat{\tau}))^{1/2}$. The coverage probability of $CI(b_\rho, s_\rho)$ is a

function of γ , for given ρ . We denote this function by $CP(\gamma; \rho)$. Note that $CP(\gamma; \rho)$ is an even function of γ for every given ρ and an even function of ρ for every given γ . For later reference, we make the very simple observation that the scaled expected length of $CI(b_\rho, s_\rho)$ is

$$\frac{E(\text{length of } CI(b_\rho, s_\rho))}{(\text{length of } I)} = E \left(\frac{\text{length of } CI(b_\rho, s_\rho)}{\text{length of } I \text{ computed from the same data}} \right). \quad (2.5)$$

The scaled expected length of $CI(b_\rho, s_\rho)$ is a function of γ , for given ρ . We denote this function by $SEL(\gamma; \rho)$. Note that $SEL(\gamma; \rho)$ is an even function of γ for every given ρ and an even function of ρ for every given γ .

2.2.1 Likelihood-based approach for the confidence interval $CI(b_\rho, s_\rho)$

The results in this subsection are due to Paul Kabaila. For later reference, we note that the confidence interval $CI(b_\rho, s_\rho)$ can be expressed in terms of likelihood functions as follows. Let $l(\theta, \tau)$ denote the log-likelihood function based on $(\hat{\theta}, \hat{\tau})$. The distribution of $(\hat{\theta}, \hat{\tau})$ is given by (2.2). Let $\hat{\tau}_\theta$ denote the value of τ that maximizes $l(\theta, \tau)$ with respect to τ , for given θ . Now define the SRLR statistic

$$r_1(\theta') = \text{sign}(\hat{\theta} - \theta') \sqrt{2 \left(l(\hat{\theta}, \hat{\tau}) - l(\theta', \hat{\tau}_{\theta'}) \right)}.$$

Let $\hat{\theta}_t$ denote the value of θ that maximizes $l(\theta, \tau)$ with respect to θ , for $\tau = t$. Now define the SRLR statistic

$$r_2 = \text{sign}(\hat{\tau} - t) \sqrt{2 \left(l(\hat{\theta}, \hat{\tau}) - l(\hat{\theta}_t, t) \right)}.$$

For notational convenience, let $v_\theta = \text{var}(\hat{\theta})$ and $v_\tau = \text{var}(\hat{\tau})$, so that

$$\begin{bmatrix} \hat{\theta} \\ \hat{\tau} \end{bmatrix} \sim N \left(\begin{bmatrix} \theta \\ \tau \end{bmatrix}, \begin{bmatrix} v_\theta & \rho v_\theta^{1/2} v_\tau^{1/2} \\ \rho v_\theta^{1/2} v_\tau^{1/2} & v_\tau \end{bmatrix} \right) \quad (2.6)$$

and

$$\text{CI}(b_\rho, s_\rho) = \left[\hat{\theta} - v_\theta^{1/2} b_\rho \left(\frac{\hat{\tau} - t}{v_\tau^{1/2}} \right) \pm v_\theta^{1/2} s_\rho \left(\frac{\hat{\tau} - t}{v_\tau^{1/2}} \right) \right].$$

Consider testing $H_0 : \theta = \theta'$ against $H_A : \theta \neq \theta'$. Suppose we accept H_0 if and only if

$$\theta' \in \left[\hat{\theta} - v_\theta^{1/2} b_\rho \left(\frac{\hat{\tau} - t}{v_\tau^{1/2}} \right) \pm v_\theta^{1/2} s_\rho \left(\frac{\hat{\tau} - t}{v_\tau^{1/2}} \right) \right].$$

Clearly, $\text{CI}(b_\rho, s_\rho)$ is equal to

$$\left\{ \theta' \in \mathbb{R} : b_\rho \left(\frac{\hat{\tau} - t}{v_\tau^{1/2}} \right) - s_\rho \left(\frac{\hat{\tau} - t}{v_\tau^{1/2}} \right) \leq \frac{\hat{\theta} - \theta'}{v_\theta^{1/2}} \leq b_\rho \left(\frac{\hat{\tau} - t}{v_\tau^{1/2}} \right) + s_\rho \left(\frac{\hat{\tau} - t}{v_\tau^{1/2}} \right) \right\}.$$

Furthermore, as shown in Appendix A.1,

$$\frac{\hat{\theta} - \theta'}{v_\theta^{1/2}} = r_1(\theta') \quad \text{and} \quad \frac{\hat{\tau} - t}{v_\tau^{1/2}} = r_2.$$

Hence the confidence interval $\text{CI}(b_\rho, s_\rho)$ is given by

$$\left\{ \theta' \in \mathbb{R} : b_\rho(r_2) - s_\rho(r_2) \leq r_1(\theta') \leq b_\rho(r_2) + s_\rho(r_2) \right\}. \quad (2.7)$$

2.3 Asymptotic results for a general regression model, without a scale parameter

In this section, we consider a general regression model without a scale parameter. Using the well-known asymptotic distribution of the maximum likelihood estimator, we derive an asymptotic distribution that is analogous to the distribution (2.2), which is for the linear regression model considered

in the previous section.

We consider a general regression model with response vector $\mathbf{y} = (y_1, \dots, y_n)$, where (y_1, \dots, y_n) denotes a column n -vector. The random variables y_1, \dots, y_n are independent and y_i has pmf or pdf (as appropriate) $f_i(y | \mathbf{x}_i, \boldsymbol{\beta})$, where $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)$ is an unknown parameter p -vector, which belongs to the open set \mathcal{B} , and \mathbf{x}_i a vector of explanatory variables of given dimension ($i = 1, \dots, n$). Suppose that the scalar parameter of interest $\theta = g(\boldsymbol{\beta})$, where $g : \mathbb{R}^p \rightarrow \mathbb{R}$ is a sufficiently smooth function. Also suppose that the parameter $\tau = h(\boldsymbol{\beta})$, where $h : \mathbb{R}^p \rightarrow \mathbb{R}$ is a sufficiently smooth function. Let $\partial g(\boldsymbol{\beta})/\partial \boldsymbol{\beta}$ denote the row p -vector with i th component $\partial g(\boldsymbol{\beta})/\partial \beta_i$ ($i = 1, \dots, p$). Suppose that $\partial g(\boldsymbol{\beta})/\partial \boldsymbol{\beta}$ and $\partial h(\boldsymbol{\beta})/\partial \boldsymbol{\beta}$ are linearly independent p -vectors, for all $\boldsymbol{\beta} \in \mathcal{B}$. Finally, suppose that we have uncertain prior information that $\tau = t$, where t is a specified number.

Let $I(\boldsymbol{\beta})$ denote the Fisher information matrix. In other words, $I(\boldsymbol{\beta})$ is the $p \times p$ matrix with (i, j) th element

$$-\sum_{i=1}^n E \left(\frac{\partial^2 \log f_i(y_i | \mathbf{x}_i; \boldsymbol{\beta})}{\partial \beta_i \partial \beta_j} \right).$$

We suppose that $I(\boldsymbol{\beta})$ is nonsingular for all $\boldsymbol{\beta} \in \mathcal{B}$. For convenience, we do not make the dependence of this matrix on n explicit in the notation. We also suppose that $n^{-1}I(\boldsymbol{\beta})$ converges to a finite nonsingular matrix as $n \rightarrow \infty$, for each $\boldsymbol{\beta} \in \mathcal{B}$.

Denote the maximum likelihood estimator of $\boldsymbol{\beta}$ by $\hat{\boldsymbol{\beta}}$. Under the appropriate regularity conditions,

$$n^{1/2}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \stackrel{approx}{\sim} N \left(\mathbf{0}, \left(n^{-1}I(\boldsymbol{\beta}) \right)^{-1} \right), \quad (2.8)$$

for large n . We use the following shorthand for this large sample distribution

$$\hat{\boldsymbol{\beta}} \stackrel{asympt}{\sim} N \left(\boldsymbol{\beta}, \left(I(\boldsymbol{\beta}) \right)^{-1} \right). \quad (2.9)$$

Let $\hat{\theta} = g(\hat{\beta})$ and $\hat{\tau} = h(\hat{\beta})$ denote the maximum likelihood estimators of θ and τ , respectively. Similarly to Section 2.2, our first step in the description of the CI for θ that utilizes the uncertain prior information is to reduce the data to $(\hat{\theta}, \hat{\tau})$.

By the mean value theorem,

$$\hat{\theta} - \theta \approx \frac{\partial g(\beta)}{\partial \beta} (\hat{\beta} - \beta) \quad \text{and} \quad \hat{\tau} - \tau \approx \frac{\partial h(\beta)}{\partial \beta} (\hat{\beta} - \beta).$$

Therefore

$$\begin{bmatrix} \hat{\theta} \\ \hat{\tau} \end{bmatrix} \stackrel{asympt}{\sim} N \left(\begin{bmatrix} \theta \\ \tau \end{bmatrix}, \begin{bmatrix} \text{avar}(\hat{\theta}; \beta) & \text{acov}(\hat{\theta}, \hat{\tau}; \beta) \\ \text{acov}(\hat{\theta}, \hat{\tau}; \beta) & \text{avar}(\hat{\tau}; \beta) \end{bmatrix} \right), \quad (2.10)$$

where $\text{avar}(\hat{\theta}; \beta)$ denotes the asymptotic variance of $\hat{\theta}$, $\text{acov}(\hat{\theta}, \hat{\tau}; \beta)$ denotes the asymptotic covariance of $\hat{\theta}$ and $\hat{\tau}$,

$$\begin{aligned} \text{avar}(\hat{\theta}; \beta) &= \frac{\partial g(\beta)}{\partial \beta} (I(\beta))^{-1} \left(\frac{\partial g(\beta)}{\partial \beta} \right)^{\top}, \\ \text{avar}(\hat{\tau}; \beta) &= \frac{\partial h(\beta)}{\partial \beta} (I(\beta))^{-1} \left(\frac{\partial h(\beta)}{\partial \beta} \right)^{\top}, \\ \text{and} \quad \text{acov}(\hat{\theta}, \hat{\tau}; \beta) &= \frac{\partial g(\beta)}{\partial \beta} (I(\beta))^{-1} \left(\frac{\partial h(\beta)}{\partial \beta} \right)^{\top}. \end{aligned}$$

Similarly to (2.1), let

$$\rho(\beta) = \frac{\text{acov}(\hat{\theta}, \hat{\tau}; \beta)}{\left(\text{avar}(\hat{\theta}; \beta) \text{avar}(\hat{\tau}; \beta) \right)^{1/2}}. \quad (2.11)$$

2.4 Analogues of I and CI(b_{ρ}, s_{ρ}) based on Wald statistics

In this section, we describe analogues of the confidence intervals I and CI(b_{ρ}, s_{ρ}) based on Wald statistics. The analogue of the confidence interval I, given by

(2.3) and based on the assumption that $(\hat{\theta} - \theta)/(\text{avar}(\hat{\theta}; \hat{\beta}))^{1/2}$ has approximately an $N(0, 1)$ distribution, is $I_W(\mathbf{y}; \alpha)$, where

$$I_W(\mathbf{y}; c) = \left[\hat{\theta} \pm z_{1-c/2} \left(\text{avar}(\hat{\theta}; \hat{\beta}) \right)^{1/2} \right],$$

with $0 < c < 1/2$.

Let $\text{ACI}_W(\beta)$ denote the interval

$$\left[\hat{\theta} - (\text{avar}(\hat{\theta}; \beta))^{1/2} b_{\rho(\beta)} \left(\frac{\hat{\tau} - t}{(\text{avar}(\hat{\tau}; \beta))^{1/2}} \right) \right. \\ \left. \pm (\text{avar}(\hat{\theta}; \beta))^{1/2} s_{\rho(\beta)} \left(\frac{\hat{\tau} - t}{(\text{avar}(\hat{\tau}; \beta))^{1/2}} \right) \right],$$

where the functions $b_{\rho(\beta)}$ and $s_{\rho(\beta)}$ are the functions b and s , respectively, computed using the R package `ciuupi`, with the desired minimum coverage probability $1 - \alpha$ and $\rho = \rho(\beta)$. We now apply the plug-in principle to obtain the confidence interval $\text{ACI}_W(\hat{\beta})$ for θ . This confidence interval is given by

$$\left[\hat{\theta} - (\text{avar}(\hat{\theta}; \hat{\beta}))^{1/2} b_{\rho(\hat{\beta})} \left(\frac{\hat{\tau} - t}{(\text{avar}(\hat{\tau}; \hat{\beta}))^{1/2}} \right) \right. \\ \left. \pm (\text{avar}(\hat{\theta}; \hat{\beta}))^{1/2} s_{\rho(\hat{\beta})} \left(\frac{\hat{\tau} - t}{(\text{avar}(\hat{\tau}; \hat{\beta}))^{1/2}} \right) \right],$$

Note that $(\hat{\tau} - t)/(\text{avar}(\hat{\tau}; \hat{\beta}))^{1/2}$ is the Wald test statistic for testing the null hypothesis $H_0 : \tau = t$ against the alternative hypothesis $H_A : \tau \neq t$. The similarity between the bivariate normal distribution (2.2) and asymptotic bivariate normal distribution (2.10) suggests that $\text{ACI}_W(\hat{\beta})$ will have coverage probability approximately equal to $1 - \alpha$ and the desired expected length properties. To summarize, the analogues of the confidence intervals I and $\text{CI}(b_\rho, s_\rho)$, based on Wald statistics, are $I_W(\mathbf{y}; \alpha)$ and $\text{ACI}_W(\hat{\beta})$, respectively.

2.5 Analogues of I and $\text{CI}(b_\rho, s_\rho)$ based on likelihood functions

The results in this section are due to Paul Kabaila. There is some evidence that likelihood-based methods lead to better hypothesis tests and confidence intervals than Wald based methods, see e.g. [Meeker & Escobar \(1995\)](#), [Cox \(2006, p.117–118\)](#), [Pawitan \(2000\)](#), [Young & Smith \(2005, p.137\)](#) and [Pawitan \(2013\)](#). For this reason, in this section, we describe analogues of I and $\text{CI}(b_\rho, s_\rho)$ based on likelihood functions.

2.5.1 Profile likelihood confidence interval for θ

The profile likelihood confidence interval for θ , with nominal coverage $1 - \alpha$, is the likelihood-based analogue of the confidence interval I. Let $\ell(\boldsymbol{\beta} | \mathbf{y})$ denote the log-likelihood function for the general regression model, without a scale parameter, described in Section 2.3. To compute the profile likelihood confidence interval for θ , with nominal coverage $1 - \alpha$, we invert a family of hypothesis tests. We test the null hypothesis $H_0 : \theta = \theta'$ against the alternative hypothesis $H_A : \theta \neq \theta'$ using the signed root likelihood ratio test (SRLR) statistic

$$r_1(\theta' | \mathbf{y}) = \text{sign}(\hat{\theta} - \theta') \sqrt{2 \left(\ell(\hat{\boldsymbol{\beta}} | \mathbf{y}) - \ell(\hat{\boldsymbol{\beta}}(\theta'; \theta) | \mathbf{y}) \right)}, \quad (2.12)$$

where $\hat{\boldsymbol{\beta}}(\theta'; \theta)$ maximises $\ell(\boldsymbol{\beta} | \mathbf{y})$ with respect to $\boldsymbol{\beta}$, subject to the constraint that $g(\boldsymbol{\beta}) = \theta'$. Suppose that we accept H_0 if and only if $-z_{1-c/2} \leq r_1(\theta' | \mathbf{y}) \leq z_{1-c/2}$, where $0 < c < 1/2$. The confidence set, with nominal coverage $1 - c$ and found by inverting the family of hypothesis tests obtained as θ' varies over $\theta' \in \mathbb{R}$, is

$$\mathcal{S}_{\text{PL}}(\mathbf{y}) = \left\{ \theta' \in \mathbb{R} : -z_{1-c/2} \leq r_1(\theta' | \mathbf{y}) \leq z_{1-c/2} \right\}. \quad (2.13)$$

Define the profile likelihood confidence interval, with nominal coverage $1 - \alpha$, as follows. This confidence interval, denoted by $I_L(\mathbf{y}; c)$, has lower endpoint $\inf(\mathcal{S}_{PL}(\mathbf{y}))$ and upper endpoint $\sup(\mathcal{S}_{PL}(\mathbf{y}))$. When $r_1(\theta' | \mathbf{y})$ is a decreasing function of θ' , $I_L(\mathbf{y}; c) = [\hat{\theta}_l, \hat{\theta}_u]$, where $\hat{\theta}_l$ and $\hat{\theta}_u$ are the solutions for θ' of

$$r_1(\theta' | \mathbf{y}) = z_{1-c/2} \quad \text{and} \quad r_1(\theta' | \mathbf{y}) = -z_{1-c/2}, \quad (2.14)$$

respectively. Thus the coverage probability of $I_L(\mathbf{y}; c)$ is

$$P\left(\theta' \in \mathbb{R} : \hat{\theta}_l \leq \theta \leq \hat{\theta}_u\right) = P\left(\theta' \in \mathbb{R} : -z_{1-c/2} \leq r_1(\theta' | \mathbf{y}) \leq z_{1-c/2}\right).$$

To summarize, the analogue of the confidence interval I, based on likelihood functions, is $I_L(\mathbf{y}; \alpha)$.

2.5.2 Likelihood-based analogue of CI(b_ρ, s_ρ)

The SRLR test statistic for testing $H_0 : \tau = t$ against the alternative hypothesis $H_A : \tau \neq t$ is

$$r_2(\mathbf{y}) = \text{sign}(\hat{\tau} - t) \sqrt{2 \left(\ell(\hat{\beta} | \mathbf{y}) - \ell(\hat{\beta}(t; \tau) | \mathbf{y}) \right)},$$

where $\hat{\beta}(t; \tau)$ maximises $\ell(\beta | \mathbf{y})$ with respect to β , subject to the constraint that $h(\beta) = t$. The likelihood-based confidence set for θ , with nominal coverage $1 - \alpha$, that is analogous to (2.7) is

$$\begin{aligned} \mathcal{S}_A(\mathbf{y}) = \left\{ \theta' \in \mathbb{R} : b_{\rho(\hat{\beta})}(r_2(\mathbf{y})) - s_{\rho(\hat{\beta})}(r_2(\mathbf{y})) \leq r_1(\theta' | \mathbf{y}) \right. \\ \left. \leq b_{\rho(\hat{\beta})}(r_2(\mathbf{y})) + s_{\rho(\hat{\beta})}(r_2(\mathbf{y})) \right\}. \end{aligned} \quad (2.15)$$

Define the confidence interval $ACI_L(\mathbf{y})$, with nominal coverage $1 - \alpha$, as follows. This confidence interval has lower endpoint $\inf(\mathcal{S}_A(\mathbf{y}))$ and upper endpoint $\sup(\mathcal{S}_A(\mathbf{y}))$. When $r_1(\theta' | \mathbf{y})$ is a decreasing function of θ' , $ACI_L(\mathbf{y}) = [\tilde{\theta}_l, \tilde{\theta}_u]$, where $\tilde{\theta}_l$ and $\tilde{\theta}_u$ are the solutions for θ' of

$$\begin{aligned}
r_1(\theta' | \mathbf{y}) &= b_{\rho(\hat{\beta})}(r_2(\mathbf{y})) + s_{\rho(\hat{\beta})}(r_2(\mathbf{y})) \quad \text{and} \\
r_1(\theta' | \mathbf{y}) &= b_{\rho(\hat{\beta})}(r_2(\mathbf{y})) - s_{\rho(\hat{\beta})}(r_2(\mathbf{y})),
\end{aligned} \tag{2.16}$$

respectively. Thus the coverage probability of $\text{ACI}_L(\mathbf{y})$ is

$$\begin{aligned}
&P\left(\theta' \in \mathbb{R} : \tilde{\theta}_l \leq \theta \leq \tilde{\theta}_u\right) \\
&= P\left(\theta' \in \mathbb{R} : b_{\rho(\hat{\beta})}(r_2(\mathbf{y})) - s_{\rho(\hat{\beta})}(r_2(\mathbf{y})) \leq r_1(\theta' | \mathbf{y}) \right. \\
&\quad \left. \leq b_{\rho(\hat{\beta})}(r_2(\mathbf{y})) + s_{\rho(\hat{\beta})}(r_2(\mathbf{y}))\right).
\end{aligned}$$

To summarize, the analogue of the confidence interval $\text{CI}(b_\rho, s_\rho)$, based on likelihood functions, is $\text{ACI}_L(\mathbf{y})$.

2.6 Assessment of the coverage probability of a confidence interval

The results in this section are due to Paul Kabaila. How should we assess the coverage probability of the confidence interval $\text{ACI}_W(\hat{\beta})$, which has nominal coverage $1 - \alpha$? For the sake of concreteness, suppose that the response $y_i \sim \text{Binomial}(N_i, \psi_i)$, with N_i given ($i = 1, \dots, n$). Let $\text{logit}(x) = \log(x/(1 - x))$, for $0 < x < 1$. Also suppose that $\text{logit}(\psi_i) = \sum_{j=1}^p x_{ij}\beta_j$, where the x_{ij} are explanatory variables taking positive values and β_1, \dots, β_p are unknown parameters. The coverage probability $P_\beta(\theta \in \text{ACI}_W(\hat{\beta}))$ will take values far below $1 - \alpha$ for extreme values of β , such as when β_1, \dots, β_p all have the same sign and $|\beta_1|, \dots, |\beta_p|$ are all large. In fact, the infimum over $\beta \in \mathbb{R}^p$ of $P_\beta(\theta \in \text{ACI}_W(\hat{\beta}))$ is 0. We expect that such extreme values of β are unlikely to occur in practice, so that this assessment of the coverage probability of $\text{ACI}_W(\hat{\beta})$ is unduly conservative.

2.6.1 Definition of the *local minimum coverage probability*, for given $\tilde{\boldsymbol{\beta}}$

Let $\tilde{\boldsymbol{\beta}}$ be a given value that satisfies $h(\tilde{\boldsymbol{\beta}}) = t$. We deal with the choice of $\tilde{\boldsymbol{\beta}}$ in the next subsubsection. Consider the straight line consisting of the values of $\boldsymbol{\beta}$ satisfying

$$\boldsymbol{\beta} = \tilde{\boldsymbol{\beta}} + \kappa (\partial h(\tilde{\boldsymbol{\beta}})/\partial \boldsymbol{\beta})^\top, \quad (2.17)$$

where $\partial h(\tilde{\boldsymbol{\beta}})/\partial \boldsymbol{\beta}$ denotes the row vector with i th element $\partial h(\tilde{\boldsymbol{\beta}})/\partial \beta_i$ and $\kappa \in \mathbb{R}$. Let $\|\cdot\|$ denote the Euclidean norm. For given small $\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|$, $|\tau - t| = |h(\boldsymbol{\beta}) - h(\tilde{\boldsymbol{\beta}})|$ is maximized by choosing $\boldsymbol{\beta}$ to satisfy (2.17).

We will assess the coverage probability of $\text{ACI}_W(\hat{\boldsymbol{\beta}})$ for values of $\boldsymbol{\beta}$ satisfying (2.17) and for

$$\kappa = \frac{(\text{avar}(\hat{\tau}; \tilde{\boldsymbol{\beta}}))^{1/2}}{\|\partial h(\tilde{\boldsymbol{\beta}})/\partial \boldsymbol{\beta}\|^2} \gamma_a, \quad \text{where } \gamma_a \in [-u, u] \quad (2.18)$$

and the chosen value of u satisfies $1 \leq u \leq 10$. For the numerical illustration presented in Sections 2.9 and 2.10, we have chosen $u = 2.5$. For the straight line segment of values of $\boldsymbol{\beta}$ satisfying (2.17) and (2.18),

$$\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\| = \frac{(\text{avar}(\hat{\tau}; \tilde{\boldsymbol{\beta}}))^{1/2}}{\|\partial h(\tilde{\boldsymbol{\beta}})/\partial \boldsymbol{\beta}\|} |\gamma_a| \leq \frac{(\text{avar}(\hat{\tau}; \tilde{\boldsymbol{\beta}}))^{1/2}}{\|\partial h(\tilde{\boldsymbol{\beta}})/\partial \boldsymbol{\beta}\|} 10.$$

Note that, from the distribution (2.8) for large n , $I(\boldsymbol{\beta})$ increases with n and so $(I(\boldsymbol{\beta}))^{-1}$ decreases with n . It follows that $(\text{avar}(\hat{\tau}; \tilde{\boldsymbol{\beta}}))^{1/2}$ converges to 0, as $n \rightarrow \infty$. Thus the supremum, over the values of $\boldsymbol{\beta}$ satisfying (2.17) and (2.18), of $\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|$ converges to 0, as $n \rightarrow \infty$. Let $\gamma = (\tau - t)/(\text{avar}(\hat{\tau}; \tilde{\boldsymbol{\beta}}))^{1/2}$. Recall that $\tau = h(\boldsymbol{\beta})$ and $\hat{\tau} = h(\hat{\boldsymbol{\beta}})$. We apply the first order Taylor series expansion for $h(\boldsymbol{\beta})$ of $\boldsymbol{\beta}$ about $\tilde{\boldsymbol{\beta}}$. Thus $\gamma - \gamma_a \rightarrow 0$, as $n \rightarrow \infty$.

For given $\tilde{\boldsymbol{\beta}}$ and for $\boldsymbol{\beta}$ satisfying (2.17) and (2.18), the coverage probability $P_{\boldsymbol{\beta}}(\theta \in \text{ACI}_W(\hat{\boldsymbol{\beta}}))$ is a function of the scalar parameter $\gamma_a \in [-u, u]$. For

given $\tilde{\beta}$, we define the *local minimum coverage probability* of the confidence interval $\text{ACI}_W(\hat{\beta})$ to be the minimum over the set of β satisfying (2.17) and (2.18) of $P_{\beta}(\theta \in \text{ACI}_W(\hat{\beta}))$.

2.6.2 Data-based choice of $\tilde{\beta}$

Consider a given data set that is assumed to be correctly modelled by the model described at the start of Section 2.3. As before, let $\hat{\beta}$ denote the maximum likelihood estimate of β . We choose $\tilde{\beta}$ to be the value of β that (a) satisfies $h(\beta) = t$ and (b) minimizes $\|\tilde{\beta} - \hat{\beta}\|$. This ensures that the $\tilde{\beta}$ is a realistic value.

Let

$$\beta^* = \tilde{\beta} + \kappa (\partial h(\tilde{\beta}) / \partial \beta)^\top, \quad (2.19)$$

where κ satisfies (2.18). Now let $\theta^* = g(\beta^*)$ and let $\hat{\beta}^*$ denote the maximum likelihood estimator of β^* . We assess the coverage probability $P_{\beta^*}(\theta^* \in \text{ACI}_W(\hat{\beta}^*))$ of the confidence interval $\text{ACI}_W(\hat{\beta}^*)$, when the true parameter value is set to β^* , by Monte Carlo simulation for each value in an equally-spaced grid of values of $\gamma_a \in [-u, u]$. These simulation results can then be used to estimate the *local minimum coverage probability* of the confidence interval $\text{ACI}_W(\hat{\beta}^*)$, which is the minimum over the set of β^* satisfying (2.19), where κ satisfies (2.18), of $P_{\beta^*}(\theta^* \in \text{ACI}_W(\hat{\beta}^*))$.

Let $\hat{\theta}^* = g(\hat{\beta}^*)$, $\tau^* = h(\beta^*)$ and $\hat{\tau}^* = h(\hat{\beta}^*)$. It follows from the asymptotic distribution (2.10) and Slutsky's theorem that a large sample approximation to the distribution of $(\hat{\theta}^*, \hat{\tau}^*)$ is

$$N \left(\begin{bmatrix} \theta^* \\ \tau^* \end{bmatrix}, \begin{bmatrix} \text{avar}(\hat{\theta}; \tilde{\beta}) & \text{acov}(\hat{\theta}, \hat{\tau}; \tilde{\beta}) \\ \text{acov}(\hat{\theta}, \hat{\tau}; \tilde{\beta}) & \text{avar}(\hat{\tau}; \tilde{\beta}) \end{bmatrix} \right). \quad (2.20)$$

This distribution is obtained when we set $\text{var}(\hat{\theta})$, $\text{cov}(\hat{\theta}, \hat{\tau})$ and $\text{var}(\hat{\tau})$ equal

to $\text{avar}(\hat{\theta}; \tilde{\beta})$, $\text{acov}(\hat{\theta}, \hat{\tau}; \tilde{\beta})$ and $\text{avar}(\hat{\tau}; \tilde{\beta})$, respectively, in (2.2). Consequently, a large sample approximation to the coverage probability $P_{\beta^*}(\theta^* \in \text{ACI}_W(\hat{\beta}^*))$ is given by $CP(\gamma^*, \rho(\tilde{\beta}))$, the coverage probability of the confidence interval $\text{CI}(b_{\rho(\tilde{\beta})}, s_{\rho(\tilde{\beta})})$ computed using `ciuupi`.

2.7 Definition of the *local scaled expected length* of a confidence interval

We consider the local parametric framework described in subsection 2.6.2. It is within this framework that we define the *local scaled expected length* of the confidence interval $\text{ACI}_W(\hat{\beta}^*)$, which has nominal coverage $1 - \alpha$. This definition is an analogue of the definition of the scaled expected length of the confidence interval $\text{CI}(b_\rho, s_\rho)$, as given by the right-hand side of (2.5). The definition of the scaled expected length, as given by the right-hand side of (2.5), is reasonable since the minimum coverage probabilities of $\text{CI}(b_\rho, s_\rho)$ and I are the same. However, the *local minimum coverage probabilities* of $\text{ACI}_W(\hat{\beta}^*)$ and $I_W(\mathbf{y}^*; \alpha)$ may not be the same. Therefore we define the *local scaled expected length* of $\text{ACI}_W(\hat{\beta}^*)$ to be

$$E_{\beta^*} \left(\frac{\text{length of } \text{ACI}_W(\hat{\beta}^*)}{\text{length of } I_W(\mathbf{y}^*; \tilde{c}) \text{ computed from the same data}} \right),$$

where \tilde{c} is such that the *local minimum coverage probabilities* of $\text{ACI}_W(\hat{\beta}^*)$ and $I_W(\mathbf{y}^*; \tilde{c})$ are the same.

Consider the argument given in the last paragraph of subsection 2.6.2. This argument implies that a large sample approximation to the *local scaled expected length* of $\text{ACI}_W(\hat{\beta}^*)$ is given by $SEL(\gamma^*, \rho(\tilde{\beta}))$, the scaled expected length of the confidence interval $\text{CI}(b_{\rho(\tilde{\beta})}, s_{\rho(\tilde{\beta})})$ computed using the R package `ciuupi`.

2.8 Application to quantal bioassays

We consider a quantal bioassay carried out to compare two similar compounds, labelled A and B. This comparison is with respect to a specified dichotomous response, labelled S and not-S, for individuals that belong to a specified large homogeneous population. Let d_1, \dots, d_m denote given dose levels. Now let $x_i = \log_{10}(d_i)$ for $i = 1, \dots, m$. Suppose that n_1, \dots, n_m and n'_1, \dots, n'_m are given positive integers.

One half of the experiment consists of carrying out the following steps for each $i = 1, \dots, m$. Suppose that n_i individuals are chosen at random from the population and given dose d_i of compound A. Let r_i denote the number of these individuals with response S. The other half of the experiment consists of carrying out the following steps for each $i = 1, \dots, m$. Suppose that n'_i individuals are chosen at random from the population and given dose d_i of compound B. Let r'_i denote the number of these individuals with response S.

We will use the following logistic regression models. Suppose that $r_1, \dots, r_m, r'_1, \dots, r'_m$ are independent. Also suppose that $r_i \sim \text{Binomial}(n_i, p_i)$ and $r'_i \sim \text{Binomial}(n'_i, p'_i)$ for $i = 1, \dots, m$. Let $\text{logit}(p) = \log(p/(1-p))$ for $0 < p < 1$. Suppose that for any dose level d of compound A, the probability p of response S for a randomly chosen individual from the population is given by $\text{logit}(p) = \beta_1 + \beta_2 x$, where $x = \log_{10}(d)$. This implies that

$$\text{logit}(p_i) = \beta_1 + \beta_2 x_i \quad \text{for } i = 1, \dots, m. \quad (2.21)$$

Also suppose that for any dose level d' of compound B, the probability p' of response S for a randomly chosen individual from the population is given by $\text{logit}(p') = \beta_3 + \beta_4 x'$, where $x' = \log_{10}(d')$. This implies that

$$\text{logit}(p'_i) = \beta_3 + \beta_4 x_i \quad \text{for } i = 1, \dots, m, \quad (2.22)$$

Let $\mathbf{y} = (r_1, \dots, r_m, r'_1, \dots, r'_m)$ and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_4)$, so that this is a model

of the type described in Section 2.3.

Let ED_z denote the log-dose x of compound A for which the probability of response S for a randomly chosen individual from the population is $z/100$. Also let ED'_z denote the log-dose x' of compound B for which the probability of response S for a randomly chosen individual from the population is $z/100$. Suppose that the parameter of interest $\theta = ED_z - ED'_z$, for some given z ($0 < z < 100$).

Consider the case that for all possible dose levels d of compound A, the probability p of response S for a randomly chosen individual from the population is the same as that for a dose level $d' = \lambda d$ of compound B, for some fixed $\lambda > 0$. Therefore, the log-dose $x = \log_{10}(d)$ of compound A leads to the same probability p of response S for a randomly chosen individual from the population as the log-dose $\log_{10}(\lambda d) = \log_{10}(\lambda) + \log_{10}(d)$ of compound B. Hence $\text{logit}(p) = \beta_1 + \beta_2 x$ and

$$\text{logit}(p) = \text{logit}(p') = \beta_3 + \beta_4 (\log_{10}(\lambda) + \log_{10}(d)) = (\beta_3 + \beta_4 \log_{10}(\lambda)) + \beta_4 x.$$

Therefore $\beta_2 = \beta_4$, so that the straight lines $\beta_1 + \beta_2 x$ and $(\beta_3 + \beta_4 \log_{10}(\lambda)) + \beta_4 x$ are parallel. This condition of “parallelism”, i.e. that $\beta_2 = \beta_4$, greatly simplifies the statistical analysis.

We consider the case that, although the compounds A and B are thought *a priori* to be sufficiently similar that the hypothesis of “parallelism” is highly plausible, we are not certain that this hypothesis holds. In other words, suppose that we have uncertain prior information that the hypothesis of “parallelism” holds.

2.9 Numerical illustration: quantal bioassay of Morphine and Amidone.

We illustrate the properties of the confidence interval $\text{ACI}_L(\mathbf{y})$, which utilizes the uncertain prior information that the hypothesis of “parallelism” holds, using data from [Grewal \(1952\)](#). This data was collected to compare the analgesic properties of Morphine and Amidone (also known as Methadone) in mice. A total of 616 homogeneous mice were randomly allocated to the groups shown in Table 2.1. In this table x_i denotes \log_{10} of the dose and n_i and n'_i denote the number of mice give this log-dose of Morphine and Amidone, respectively. The experimenter recorded the number of shocks that could be applied to the tail of the mouse before it squeaked. If the number of shocks was four or more then the mouse was taken to give response S. In Table 2.1, r_i denotes the number of mice (out of n_i mice) with response S for the log-dose x_i of Morphine. Similarly, r'_i denotes the number of mice (out of n'_i mice) with response S for the log-dose x_i of Amidone.

Table 2.1: Quantal bioassay of Morphine and Amidone

\log_{10} dose	Morphine		Amidone	
x_i	n_i	r_i	n'_i	r'_i
0.18	103	19	60	14
0.48	120	53	110	54
0.78	123	83	100	81

Suppose that the parameter of interest is $\theta = \text{ED}_z - \text{ED}'_z$, where ED_z and ED'_z are the log-doses of Morphine and Amidone, respectively, for which the probability of response S for a randomly chosen mouse is $z/100$. Also suppose that our aim is to find a confidence interval for θ with minimum

coverage probability 0.95.

Morphine and Amidone both belong to the family of drugs known as opioids. Opioids act on the brain in a particular way that can provide pain relief. Because Morphine and Amidone are both opioids, the hypothesis of “parallelism” is highly plausible. However, we are not certain that this hypothesis holds. In other words, we have uncertain prior information that the hypothesis of “parallelism” holds.

The models that we use for the Morphine and Amidone data are (2.21) and (2.22), respectively, with $m = 3$, $n_1 = 103$, $n_2 = 120$, $n_3 = 123$, $n'_1 = 60$, $n'_2 = 110$ and $n'_3 = 100$ with

$$\text{ED}_z = \frac{1}{\beta_2} \left(\text{logit} \left(\frac{z}{100} \right) - \beta_1 \right) \quad \text{and} \quad \text{ED}'_z = \frac{1}{\beta_4} \left(\text{logit} \left(\frac{z}{100} \right) - \beta_3 \right).$$

The parameter of interest is equal to

$$\theta = g(\boldsymbol{\beta}) = \frac{1}{\beta_2} \left(\text{logit} \left(\frac{z}{100} \right) - \beta_1 \right) - \frac{1}{\beta_4} \left(\text{logit} \left(\frac{z}{100} \right) - \beta_3 \right).$$

Henceforth, we consider that case that $z = 60$. Let $\tau = h(\boldsymbol{\beta}) = \beta_2 - \beta_4$. The uncertain prior information is that $\tau = 0$.

2.9.1 Application of the procedure described in subsection 2.6.2

We apply the procedure described in subsection 2.6.2 to the Morphine/Amidone data. For this data, the maximum likelihood estimate $\hat{\boldsymbol{\beta}} = (-2.0652, 3.6418, -2.0968, 4.4581)$. We find that $\tilde{\boldsymbol{\beta}} = (\tilde{\beta}_1, \dots, \tilde{\beta}_p)$ is given by

$$\tilde{\beta}_1 = \hat{\beta}_1, \quad \tilde{\beta}_2 = \frac{\hat{\beta}_2 + \hat{\beta}_4}{2}, \quad \tilde{\beta}_3 = \hat{\beta}_3, \quad \tilde{\beta}_4 = \frac{\hat{\beta}_2 + \hat{\beta}_4}{2}.$$

The data for Morphine and Amidone come from independent experiments, so that the estimators $(\hat{\beta}_1, \hat{\beta}_2)$ and $(\hat{\beta}_3, \hat{\beta}_4)$ are independent. As

a result of this, the inverse of the Fisher information matrix, $\left(I(\boldsymbol{\beta})\right)^{-1}$, is block diagonal. Using the expression for the Fisher information matrix, in the context of a logistic regression model, given on page 116 of [McCullagh & Nelder \(1989\)](#) we find that

$$\left(I(\tilde{\boldsymbol{\beta}})\right)^{-1} = \left[\begin{array}{cc|cc} 0.086304 & -0.142229 & 0 & 0 \\ -0.142229 & 0.280902 & 0 & 0 \\ \hline 0 & 0 & 0.132504 & -0.216338 \\ 0 & 0 & -0.216338 & 0.408074 \end{array} \right].$$

Also note that,

$$\frac{\partial g(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \left[\frac{-1}{\beta_2} \quad \frac{\beta_1 - \text{logit}(z/100)}{\beta_2^2} \quad \frac{1}{\beta_4} \quad \frac{\text{logit}(z/100) - \beta_3}{\beta_4^2} \right]$$

and

$$\frac{\partial h(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \begin{bmatrix} 0 & 1 & 0 & -1 \end{bmatrix}.$$

Therefore

$$\begin{aligned} \text{avar}(\hat{\theta}; \tilde{\boldsymbol{\beta}}) &= \frac{\partial g(\tilde{\boldsymbol{\beta}})}{\partial \boldsymbol{\beta}} \left(I(\tilde{\boldsymbol{\beta}})\right)^{-1} \left(\frac{\partial g(\tilde{\boldsymbol{\beta}})}{\partial \boldsymbol{\beta}}\right)^{\top} = 0.002333, \\ \text{avar}(\hat{\tau}; \tilde{\boldsymbol{\beta}}) &= \frac{\partial h(\tilde{\boldsymbol{\beta}})}{\partial \boldsymbol{\beta}} \left(I(\tilde{\boldsymbol{\beta}})\right)^{-1} \left(\frac{\partial h(\tilde{\boldsymbol{\beta}})}{\partial \boldsymbol{\beta}}\right)^{\top} = 0.688976, \\ \text{acov}(\hat{\theta}, \hat{\tau}; \tilde{\boldsymbol{\beta}}) &= \frac{\partial g(\tilde{\boldsymbol{\beta}})}{\partial \boldsymbol{\beta}} \left(I(\tilde{\boldsymbol{\beta}})\right)^{-1} \left(\frac{\partial h(\tilde{\boldsymbol{\beta}})}{\partial \boldsymbol{\beta}}\right)^{\top} = -0.01603, \end{aligned}$$

so that $\rho(\tilde{\boldsymbol{\beta}}) = -0.399855$.

We also find that

$$\boldsymbol{\beta}^* = \tilde{\boldsymbol{\beta}} + \frac{(\text{avar}(\hat{\tau}; \tilde{\boldsymbol{\beta}}))^{1/2}}{2} [0 \ 1 \ 0 \ -1]^{\top} \gamma_a.$$

Let $\tau^* = h(\boldsymbol{\beta}^*) = \beta_2^* - \beta_4^*$. Now let $\gamma^* = \tau^* / (\text{avar}(\hat{\tau}; \tilde{\boldsymbol{\beta}}))^{1/2}$. It follows that

$\gamma^* = \gamma_a$, so that $\beta^* = (\beta_1^*, \dots, \beta_p^*)$ is given by

$$\beta_1^* = \hat{\beta}_1, \beta_2^* = \frac{\hat{\beta}_2 + \hat{\beta}_4}{2} + \frac{\tau^*}{2}, \beta_3^* = \hat{\beta}_3, \beta_4^* = \frac{\hat{\beta}_2 + \hat{\beta}_4}{2} - \frac{\tau^*}{2}, \quad (2.23)$$

where $\tau^* = (\text{avar}(\hat{\tau}; \tilde{\beta}))^{1/2} \gamma^*$ and $\gamma^* \in [-u, u]$. We deal with the choice of u in the next subsection.

2.9.2 Choice of u

For $\beta_4^* = 0$ it is impossible to determine ED_z for any $0 < z < 100$. Furthermore, values of $\beta_4^* < 0$ seem impossible. Therefore, $\beta_4^* = 0$ is a boundary point for impossible values of β_4^* . Note that $\beta_4^* = 0$ when $\tau^* = \hat{\beta}_2 + \hat{\beta}_4$. Similarly, for $\beta_2^* = 0$ it is impossible to determine ED_z for any $0 < z < 100$. Furthermore, values of $\beta_2^* < 0$ seem impossible. Therefore, $\beta_2^* = 0$ is a boundary value for impossible values of β_2^* . Note that $\beta_2^* = 0$ when $\tau^* = -(\hat{\beta}_2 + \hat{\beta}_4)$. Therefore, u must be less than $\hat{\beta}_2 + \hat{\beta}_4 = 8.0999$.

In fact, u must be a good deal less than $\hat{\beta}_2 + \hat{\beta}_4$ for the profile likelihood confidence interval for θ^* , with nominal coverage 0.95, not to have extremely large lengths for a substantial proportion of samples. This is evident from Table 2.2 which shows the values of τ^* and γ^* and the percentage of simulation runs for which the length of the profile likelihood confidence interval is greater than 1000 for $z = 60$ and $M = 5000$ simulation runs. We have therefore chosen $u = 2.5$, so that we restrict attention to $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$. Note that $\gamma^* = 2.5$ corresponds to $\tau^* = 2.075$. To get a sense of the difference in slopes that this allows, consider the following.

- (i) Suppose that $\gamma^* = 0$, so that $\tau^* = 0$ and the hypothesis of “parallelism” is satisfied. In this case, $\beta_2^* = \beta_4^* = 4.0499$. Consequently, $ED_{60} = 0.6101$ and $ED'_{60} = 0.6178$, so that $\theta^* = -0.0078$.
- (ii) Suppose that $\gamma^* = 2.5$, so that $\tau^* = 2.075$. In this case, $\beta_2^* = 5.0874$,

$\beta_4^* = 3.0124$. Consequently, $\text{ED}_{60} = 0.4856$ and $\text{ED}'_{60} = 0.8306$, so that $\theta^* = -0.345$.

(iii) Suppose that $\gamma^* = -2.5$, so that $\tau^* = -2.075$. In this case, $\beta_2^* = 3.0124$, $\beta_4^* = 5.0874$. Consequently, $\text{ED}_{60} = 0.8202$ and $\text{ED}'_{60} = 0.4918$, so that $\theta^* = 0.3283$.

Table 2.2: The values of γ^* and τ^* and the percentage of simulation runs (row marked %) for which the length of the profile likelihood confidence interval is greater than 1000 for $z = 60$ and $M = 5000$ simulation runs.

γ^*	-5	-4.5	-4	-3.5	-3		-2.5	-2	-1.5	-1	-0.5
τ^*	-4.15	-3.74	-3.32	-2.91	-2.49		-2.08	-1.66	-1.25	-0.83	-0.42
%	2.10	1.08	0.16	0.02	0.01		0.00	0.00	0.00	0.00	0.00

γ^*	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
τ^*	0.00	0.42	0.83	1.25	1.66	2.08	2.49	2.91	3.32	3.74	4.15
%	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.53	1.50	3.92	7.51

2.10 Numerical Illustration: Monte Carlo simulation estimation of the *local coverage probabilities* and *scaled expected lengths* of $\mathbf{I}_L(\mathbf{y}^*; c)$ and $\text{ACI}_L(\mathbf{y}^*)$

Suppose that β^* is given by (2.23), where γ^* (and therefore τ^*) is specified. Replace β by β^* in the models (2.21) and (2.22) that we use for the Morphine and Amidone data, respectively. For these models, $k = 3$, $n_1 = 103$, $n_2 = 120$, $n_3 = 123$, $n'_1 = 60$, $n'_2 = 110$ and $n'_3 = 100$. Let \mathbf{y}^* denote the response vector.

Our Monte Carlo simulation results show that the local coverage and scaled expected length properties of $\text{ACI}_L(\mathbf{y}^*)$ are superior to these properties for $\text{ACI}_W(\hat{\boldsymbol{\beta}}^*)$. Consequently, the description of these properties for $\text{ACI}_W(\hat{\boldsymbol{\beta}}^*)$ have been relegated to Appendix A.2. For the remainder of this chapter, we deal only with these properties for $\text{ACI}_L(\mathbf{y}^*)$.

2.10.1 Monte Carlo simulation estimation of the *local coverage probabilities*

The Monte Carlo simulation estimation of the *local coverage probabilities* of $I_L(\mathbf{y}^*; c)$, for given c , and $\text{ACI}_L(\mathbf{y}^*)$ are very similar. Let $\theta^* = g(\boldsymbol{\beta}^*)$. We carry out M independent simulation runs. The k th simulation run generates an observation of \mathbf{y}^* . We make Assumption A (stated in Appendix A.3) with $\boldsymbol{\beta}$ and \mathbf{y} replaced by $\boldsymbol{\beta}^*$ and \mathbf{y}^* , respectively. The Monte Carlo simulation results reported in Appendix A.4 provide evidence in favour of the correctness of this assumption.

We estimate the coverage probability $P_{\boldsymbol{\beta}^*}(\theta^* \in I_L(\mathbf{y}^*; c))$, for given c , as follows. On the k th simulation run we record $\mathbf{1}(-z_{1-\alpha/2} \leq r_1(\theta^* | \mathbf{y}^*) \leq z_{1-\alpha/2})$. Using the recorded results for the M simulation runs, we estimate this coverage probability and the standard error of this estimate in the obvious way.

We estimate the coverage probability $P_{\boldsymbol{\beta}^*}(\theta^* \in \text{ACI}_L(\mathbf{y}^*))$ as follows. On the k th simulation run we record

$$\mathbf{1}\left(b_{\rho(\hat{\boldsymbol{\beta}}^*)}(r_2(\mathbf{y}^*)) - s_{\rho(\hat{\boldsymbol{\beta}}^*)}(r_2(\mathbf{y}^*)) \leq r_1(\theta^* | \mathbf{y}^*) \leq b_{\rho(\hat{\boldsymbol{\beta}}^*)}(r_2(\mathbf{y}^*)) + s_{\rho(\hat{\boldsymbol{\beta}}^*)}(r_2(\mathbf{y}^*))\right). \quad (2.24)$$

Using the recorded results for the M simulation runs, we estimate this coverage probability and the standard error of this estimate in the obvious way. In Appendix A.5, we describe the use of a *look-up table* for the fast evaluation

of the functions b_ρ and s_ρ for any given ρ . In Appendix A.6, we describe the computations of $r_1(\theta^* | \mathbf{y}^*)$ and $r_2(\mathbf{y}^*)$ in R.

The top panel of Figure 2.1 presents approximate 95% confidence intervals for the coverage probability of the confidence interval $I_L(\mathbf{y}^*; 0.05)$, which has nominal coverage 0.95, evaluated at $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$. The bottom panel of this figure presents approximate 95% confidence intervals for the coverage probability of the confidence interval $ACI_L(\mathbf{y}^*)$, which has nominal coverage 0.95, evaluated on the same set of values of γ^* . For both of these panels, the number of simulation runs $M = 40,000$. These figures show that both $I_L(\mathbf{y}^*; 0.05)$ and $ACI_L(\mathbf{y}^*)$ have good local coverage properties as the estimated coverage probabilities for the values of γ^* considered are all very close to 0.95.

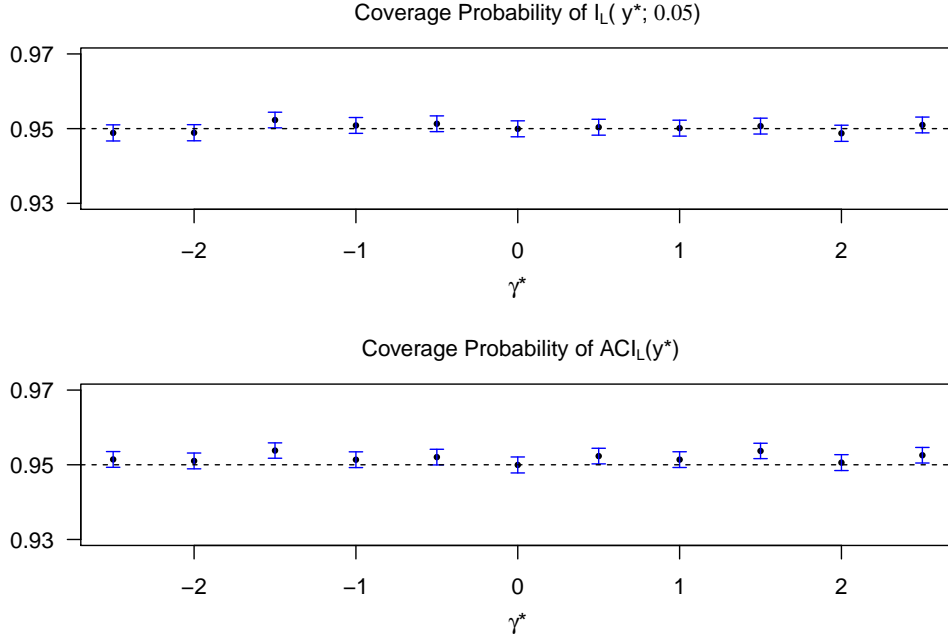


Figure 2.1: Approximate 95% confidence intervals for the coverage probabilities of the confidence intervals $I_L(\mathbf{y}^*; 0.05)$ (top panel) and $ACI_L(\mathbf{y}^*)$ (bottom panel), both with nominal coverage 0.95, for $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$.

2.10.2 Monte Carlo simulation estimation of the *local scaled expected length* of $\text{ACI}_L(\mathbf{y}^*)$

Let

$$q_L^* = \frac{\text{length of } \text{ACI}_L(\mathbf{y}^*)}{\text{length of } I_L(\mathbf{y}^*; \tilde{c}) \text{ computed from the same data}},$$

where \tilde{c} is such that the *local minimum coverage probabilities* of $\text{ACI}_L(\mathbf{y}^*)$ and $I_L(\mathbf{y}^*; \tilde{c})$ are the same. The *local scaled expected length* of $\text{ACI}_W(\hat{\beta}^*)$ was defined in Section 2.7. The *local scaled expected length* of $\text{ACI}_L(\mathbf{y}^*)$ is similarly defined to be $E_{\beta^*}(q_L^*)$.

We computed \tilde{c} using the method described in subsection A.7.1 of Appendix A.7, with $M' = 10,000$. We then used Monte Carlo simulation to estimate $E_{\beta^*}(q_L^*)$ as follows. We carry out M independent simulation runs. On the k th simulation run we generate an observation $q_L^*(k)$ of q_L^* . We estimate $E_{\beta^*}(q_L^*)$ by $\sum_{k=1}^M q_L^*(k)/M$. The analysis of the distribution of q_L^* , given in Appendix A.8, shows that this distribution does not have any long or heavy tails.

The left panel of Figure 2.2 presents approximate 95% confidence intervals for the *local scaled expected length* of the confidence interval $\text{ACI}_L(\mathbf{y}^*)$, with nominal coverage 0.95, evaluated at each $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$ using $M = 40,000$ simulation runs. These approximate 95% confidence intervals were found using the simplifying approximation that \tilde{c} is computed without error. This panel shows that the confidence interval $\text{ACI}_L(\mathbf{y}^*)$ utilizes the uncertain prior information that $\tau^* = 0$.

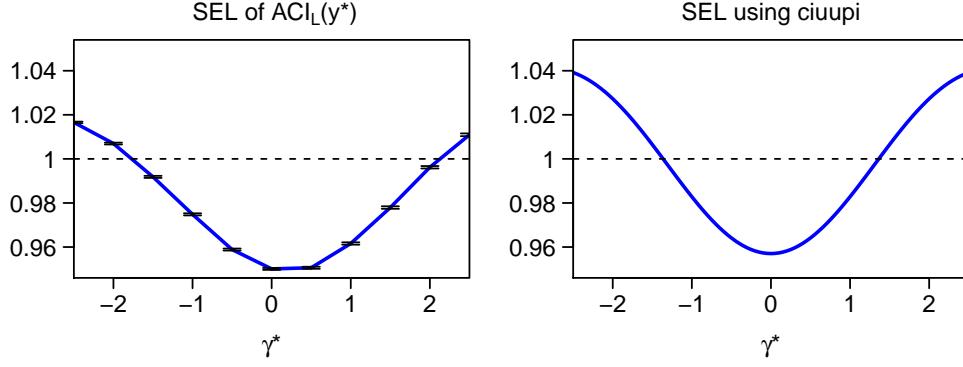


Figure 2.2: The left panel presents approximate 95% confidence intervals for the *local scaled expected length* of the confidence interval $\text{ACI}_L(\mathbf{y}^*)$, with nominal coverage 0.95, evaluated at $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$. The right panel is the graph of $SEL(\gamma^*; \rho(\tilde{\boldsymbol{\beta}}))$ for the confidence interval $\text{CI}(b_{\rho(\tilde{\boldsymbol{\beta}})}, s_{\rho(\tilde{\boldsymbol{\beta}})})$, found using the R package `ciuupi`.

2.10.3 Large sample approximation to the *local coverage probability* and scaled expected length of the confidence interval $\text{ACI}_L(\mathbf{y}^*)$

Application of the large sample approximations given in subsection 2.6.2 and Section 2.7 to the Morphine/Amidone data gives the following results. The coverage probability and the scaled expected length of the confidence interval $\text{ACI}_L(\mathbf{y}^*)$ are approximated by the coverage probability $CP(\gamma^*; \tilde{\boldsymbol{\beta}})$ and the scaled expected length $SEL(\gamma^*; \tilde{\boldsymbol{\beta}})$ of the confidence interval $\text{CI}(b_{\rho(\tilde{\boldsymbol{\beta}})}, s_{\rho(\tilde{\boldsymbol{\beta}})})$ computed using the R package `ciuupi`. Recall that $\rho(\tilde{\boldsymbol{\beta}}) = -0.399855$ for the Morphine/Amidone data,

Suppose that $1 - \alpha = 0.95$. Graphs of the coverage probability $CP(\gamma^*; \tilde{\boldsymbol{\beta}})$ and the scaled expected length $SEL(\gamma^*; \tilde{\boldsymbol{\beta}})$, considered as functions of $|\gamma^*| \in [0, 10]$, are shown in Figure 2.3. The right panel of Figure 2.2 is a graph of

$SEL(\gamma^*; \rho(\tilde{\beta}))$, considered as a function of $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$. The left and right hand panels of Figure 2.2 show very similar qualitative features.

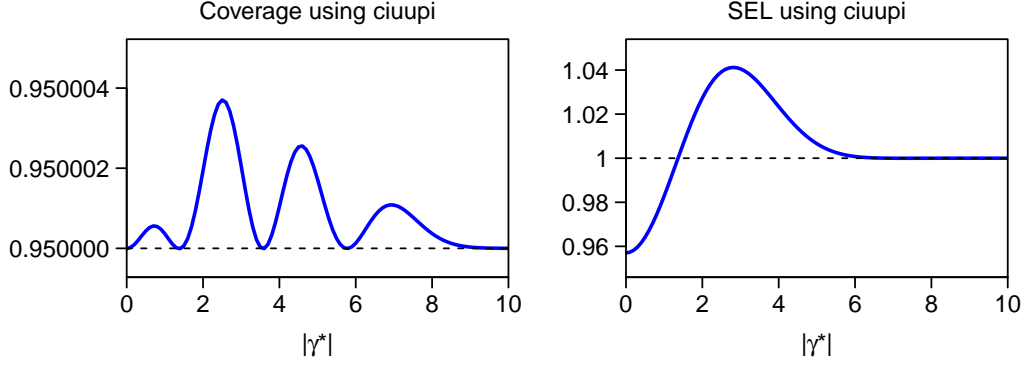


Figure 2.3: Graphs of the coverage probability $CP(\gamma^*; \rho(\tilde{\beta}))$ and the scaled expected length $SEL(\gamma^*; \rho(\tilde{\beta}))$ for the confidence interval $CI(b_{\rho(\tilde{\beta})}, s_{\rho(\tilde{\beta})})$, found using the R package `ciuupi`.

2.11 Conclusion

We constructed a confidence interval for a scalar parameter of interest θ that utilizes the uncertain prior information that a distinct scalar parameter τ takes the specified value t for a general regression model, without a scale parameter. To construct this confidence interval, we consider the similarity between the asymptotic joint distribution of the maximum likelihood estimators of θ and τ and the joint distribution of these estimators in the particular case of a linear regression with normally distributed errors having known variance.

We expressed this similarity both in terms of Wald statistics and signed root likelihood ratio (SRLR) statistics. As evidenced by the graphs of the coverage probability and scaled expected length in Section 2.10 and Appendix A.2, we observed that expressing this similarity in terms of SRLR statistics

leads to the confidence interval in the general regression context having better performance than when we express this similarity in terms of Wald statistics. The coverage probability plots of the likelihood-based confidence intervals $I_L(\mathbf{y}^*; 0.05)$ and $ACI_L(\mathbf{y}^*)$ both showed good local coverage properties and the likelihood-based confidence interval $ACI_L(\mathbf{y}^*)$ utilized the uncertain prior information that $\tau^* = 0$.

Chapter 3

Computation of the expected value of a function of a chi-distributed random variable

3.1 Introduction

In this chapter, we consider the numerical evaluation of the expected value of a smooth bounded function of a chi-distributed random variable, divided by the square root of the number of degrees of freedom. This computation, described in detail in Section 3.2, arises in the assessment of the coverage probability and expected volume properties of some non-standard confidence regions (Farchione & Kabaila, 2008, Kabaila & Giri, 2009a, Kabaila & Giri, 2009b, Kabaila & Farchione, 2012, Kabaila & Giri, 2013, Kabaila & Tissera, 2014, Kabaila *et al.*, 2016, Kabaila *et al.*, 2017, Abeysekera & Kabaila, 2017 and Kabaila, 2018). It also arises in simultaneous statistical inference and the selection and ranking of populations (Miller, 1981, Hochberg & Tamhane, 1987, Gupta & Panchapakesan, 2002) and in the evaluation of central and non-central (Kshirsagar definition, which is consistent with expression (1.4) of Genz & Bretz, 2009) multivariate t probabilities (Dunnett & Sobel, 1955,

Dunnett, 1989, Genz & Bretz, 2009). The new computational method that we describe in the present chapter is applied in Chapter 4 to the computation of the Kabaila & Giri (2009a) confidence interval that utilizes uncertain prior information.

For the evaluation of the integral of interest, we initially seek out an appropriate transformation of the variable of integration. As noted by Schwartz (1969), “The real artistry of numerical integration lies in learning to make changes of the variable” appropriate for the problem at hand and that “this must be studied separately for every problem.” The literature on various initial changes of the variable of integration for the purpose of efficient numerical integration is vast, with early references including Davis & Rabinowitz (1984), Sag & Szekeres (1964), and Imhof (1963). Some simple illustrations of the power of appropriate changes in the variable of integration before numerical integration were provided by Avery & Soler (1988).

We carry out the numerical evaluation of interest using a transformation put forward by Mori (1988), followed by the application of the trapezoidal rule. This transformation belongs to a family of transformations proposed and investigated by Takahasi & Mori (1973), Mori (1985), and others. The trapezoidal rule has the advantage that it leads to a nested sequence of quadrature rules that can be used for estimation of the approximation error. It has the remarkable property that, for suitable integrands, it is exponentially convergent (Trefethen & Weideman, 2014). There are several well-known explanations for this remarkable property, including the Euler-Maclaurin summation formula and Fourier transform methods.

We describe the properties of the trapezoidal rule using the Fourier transform of the integrand and the Poisson summation formula in Section 3.3. In Section 3.4, we describe in detail the application of the transformation (2.6) of Mori (1988) followed by the application of the trapezoidal rule. We also

describe a method of carrying out the required ‘trimming’ of the infinite sum approximation to the integral that leads to an easily-computed upper bound on the resulting *trimming error*. In subsection 3.4.3, we describe a simple and effective procedure, similar to that described by Mori (1988), for the evaluation of the integral of interest. In subsection 3.4.4, we describe an extension of this procedure that we prove to be exponentially convergent under the appropriate regularity condition.

In Section 3.5, we provide numerical results for the method described in subsection 3.4.3, using the simple test scenario that consists of evaluating a known univariate t probability. In Section 3.6, we compare the performance of the method described in subsection 3.4.3 with the three methods (a) Gauss Legendre quadrature, (b) Generalized Gauss Laguerre quadrature and (c) inverse cdf method followed by Gauss Legendre quadrature. The purpose of this comparison is to illustrate the factors that can lead to a relatively poor performance of these three methods.

Finally, in Section 3.7, we discuss the application of the procedures described in Section 3.4 to the computation of the coverage probability and scaled expected volume of some non-standard confidence regions. The computations for this chapter were carried out using the R computer language. The work described in this chapter appears in Kabaila & Ranathunga (2021).

3.2 Integral of interest

We consider the problem of finding an accurate and efficient method of numerically computing an integral of the form

$$\int_0^\infty a(x) f_\nu(x) dx, \quad (3.1)$$

where a is a smooth bounded real-valued function, ν is a positive integer and f_ν is the probability density function (pdf) of a random variable with

the same distribution as $R/\nu^{1/2}$, where R has a χ_ν distribution (i.e. R^2 has a χ_ν^2 distribution). Note that $(3.1) = E(a(R/\nu^{1/2}))$, which is the expected value of a smooth bounded function of $R/\nu^{1/2}$. We suppose that a computer program for the accurate and efficient evaluation of $a(x)$, for any given $x > 0$, is either already available or can be easily written. In other words, our focus is solely on the numerical evaluation of the integral (3.1).

The pdf f_ν is given by

$$f_\nu(x) = \begin{cases} \tau_\nu x^{\nu-1} \exp(-\nu x^2/2) & \text{for } x > 0 \\ 0 & \text{otherwise,} \end{cases}$$

where

$$\tau_\nu = \frac{\nu^{\nu/2}}{\Gamma(\nu/2) 2^{(\nu/2)-1}}.$$

Our search for a better method for the evaluation of an integral of the form (3.1) has led us to seek out an appropriate transformation of the variable of integration, followed by the application trapezoidal rule over the real line. The trapezoidal rule also has the great advantage that it can be used to create a nested sequence of quadrature rules, used for the estimation of the approximation error, so that previous evaluations of the function a are not wasted. For our purposes, the best description of the properties of the trapezoidal rule is found using the Fourier transform of the integrand and the Poisson summation formula. We describe this well-known description in the next section.

3.3 Properties of the trapezoidal rule found using the Fourier transform of the integrand

In this section we recall the Poisson summation formula and its well-known use in assessing the error of the trapezoidal rule. Suppose that we wish to evaluate

$$\int_{-\infty}^{\infty} g(y) dy, \quad (3.2)$$

where g is a real-valued absolutely integrable function. Let G denote that Fourier transform of g . This transform is defined by

$$G(\omega) = \int_{-\infty}^{\infty} g(y) \exp(-i \omega y) dy,$$

where $i = \sqrt{-1}$ and the angular frequency $\omega \in \mathbb{R}$. Since g is real-valued, $G(\omega)$ is an even function of ω (see e.g. p.11 of [Papoulis 1962](#)). As shown in [Appendix B.1](#), it follows from the Poisson summation formula (see e.g. p.47 of [Papoulis 1962](#)) that

$$\left| h \sum_{j=-\infty}^{\infty} g(jh + \delta) - \int_{-\infty}^{\infty} g(y) dy \right| \leq 2 \sum_{j=1}^{\infty} \left| G\left(\frac{2\pi j}{h}\right) \right|, \quad (3.3)$$

for all $\delta \in [0, h)$. The left-hand side is the *discretization error*. This error is small when $|G(\omega)|$ decays rapidly as $\omega \rightarrow \infty$ and h is sufficiently small.

We approximate the infinite sum

$$h \sum_{j=-\infty}^{\infty} g(jh + \delta) \quad (3.4)$$

by the finite sum

$$h \sum_{j=M}^N g(jh + \delta), \quad (3.5)$$

for appropriately chosen integers M and N ($M < N$). The “trapezoidal rule” approximation to (3.2) is (3.5). The absolute value of the difference (3.5) – (3.4) is called the *trimming error*. For (3.5) to be a good approximation to (3.2), we require that both the *discretization error* and the *trimming error* are small.

3.4 Application of the transformation (2.6) of Mori (1988), followed by the application of the trapezoidal rule

In this section, we apply the transformation (2.6) of Mori (1988) to the integral (3.1), followed by the application of the trapezoidal rule. We also describe a method of carrying out the required ‘trimming’ of the infinite sum approximation to the integral that leads to an easily-computed upper bound on the resulting *trimming error*. Throughout this section we suppose that ν is given.

3.4.1 Transformation (2.6) of Mori (1988)

To evaluate (3.1), we first apply the transformation (2.6) of Mori (1988), namely

$$x(y) = \exp\left(\frac{1}{2}y - e^{-y}\right),$$

so that

$$\frac{dx(y)}{dy} = \exp\left(\frac{1}{2}y - e^{-y}\right) \left(\frac{1}{2} + e^{-y}\right)$$

and

$$\int_0^\infty a(x) f_\nu(x) dx = \int_{-\infty}^\infty a(x(y)) \psi_\nu(y) dy, \quad (3.6)$$

where

$$\psi_\nu(y) = f_\nu(x(y)) \frac{dx(y)}{dy}.$$

The formula for the function $\psi_\nu(y)$ is given in Appendix B.2.

3.4.2 Assumption FT and exponential convergence

As noted by Mori (1985) this transformation leads to $\psi_\nu(y)$ having *double exponential* decay as $y \rightarrow \pm\infty$, i.e. there exist positive numbers c_1 , c_2 and c_3 such that

$$|\psi_\nu(y)| \sim c_1 \exp(-c_2 \exp(c_3|y|)), \quad y \rightarrow \pm\infty. \quad (3.7)$$

This implies that $g_\nu(y) = a(x(y)) \psi_\nu(y)$ also has *double exponential* decay as $y \rightarrow \pm\infty$. Computational results show that the function ψ_ν is unimodal for all positive integers ν . Figure 3.1 shows the distribution of the function $\psi_\nu(y)$ as a function of y for $\nu = 1, 2, 3, 10$ and 100.

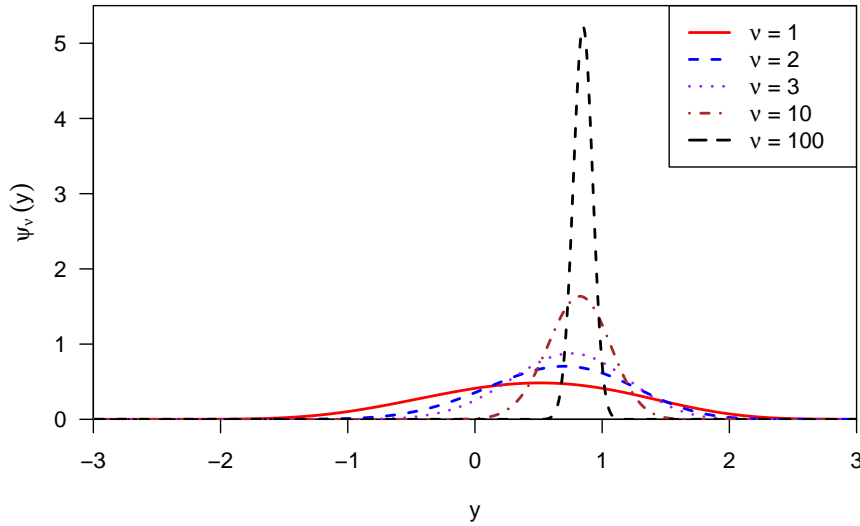


Figure 3.1: Plot of the function $\psi_\nu(y)$ as a function of y for $\nu = 1, 2, 3, 10$ and 100

Let y_ν^* denote the value of y at which $\psi_\nu(y)$ is maximized. The value of y_ν^* is roughly 0.85 for all positive integers ν (as evidenced by Figure 3.1). We

suppose, without loss of generality, that $|a(x)| \leq 1$ for all $x \in \mathbb{R}$.

Let G_ν denote the Fourier transform of $g_\nu(y)$. We now introduce the following assumption.

Assumption FT: There exist positive numbers c_4 and c_{FT} such that

$$|G_\nu(\omega)| \leq c_4 \exp(-c_{FT}|\omega|)$$

for all $\omega \in \mathbb{R}$. In other words, $G_\nu(\omega)$ has *single exponential* decay as $\omega \rightarrow \pm\infty$.

Theorem 5.1 of [Trefethen & Weideman \(2014\)](#) provides conditions on the function $g_\nu(y)$ that imply that this assumption holds.

We will approximate (3.6) by

$$h \sum_{j=0}^{n-1} a(x(y_\ell + hj)) \psi_\nu(y_\ell + hj), \quad (3.8)$$

where n denotes the number of evaluations of the integrand $a(x(y))\psi_\nu(y)$, h denotes the step length and the first evaluation of this integrand is at y_ℓ . Let $d = nh$. Of course, our aim is to choose (n, h, y_ℓ) such that (3.8) provides an efficient and accurate approximation.

We will use the following result which provides an easily computed upper bound on the *trimming error*. This result is due to Paul Kabaila.

Lemma 3.4.1. *Suppose that $y_\ell < y_\nu^*$ and that $y_\ell + d > y_\nu^*$. Then, when we approximate (3.6) by (3.8), the trimming error is bounded above by $u_\nu(y_\ell, d)$, where*

$$u_\nu(y, d) = Q_\nu(\nu x^2(y)) + 1 - Q_\nu(\nu x^2(y + d))$$

and Q_ν denotes the χ_ν^2 cdf.

Proof. Suppose that $y_\ell < y_\nu^*$ and that $y_\ell + d > y_\nu^*$. The *trimming error* is

$$\left| h \sum_{j=-\infty}^{-1} a(x(y_\ell + hj)) \psi_\nu(y_\ell + hj) + h \sum_{j=n}^{\infty} a(x(y_\ell + hj)) \psi_\nu(y_\ell + hj) \right|. \quad (3.9)$$

The *trimming error* is bounded above by

$$h \sum_{j=-\infty}^{-1} \psi_{\nu}(y_{\ell} + hj) + h \sum_{j=n}^{\infty} \psi_{\nu}(y_{\ell} + hj),$$

since, for all positive integers ν , $\psi_{\nu}(y) > 0$ for all $y \in \mathbb{R}$. Observe that

$$h \sum_{j=n}^{\infty} \psi_{\nu}(y_{\ell} + hj) = h \sum_{j=1}^{\infty} \psi_{\nu}(y_u + hj),$$

where $y_u = y_{\ell} + d$.

Figure 3.2 shows the plot of $\psi_{\nu}(y_u + t)$ as a function of t for $\nu = 10$ and $h = 0.01$.

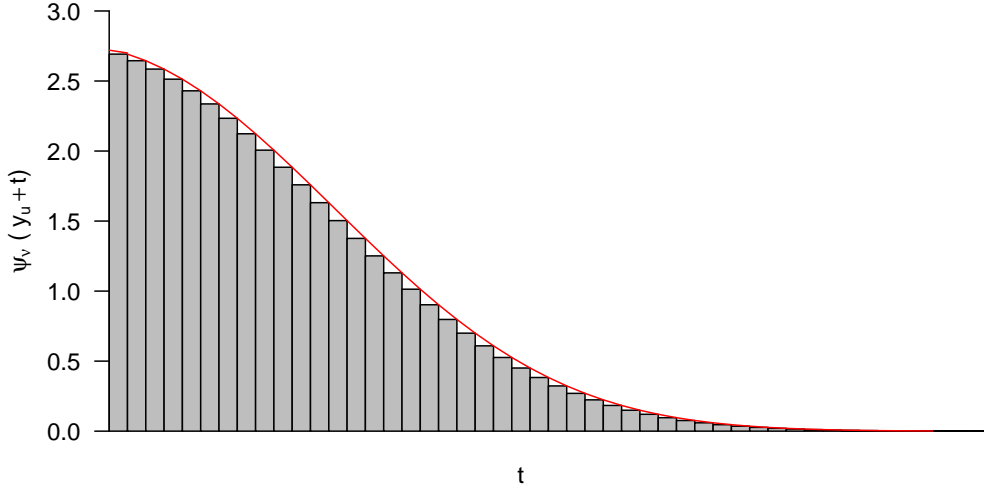


Figure 3.2: Graph of $\psi_{\nu}(y_u + t)$ as a function of t for $\nu = 10$ and $h = 0.01$.

We now use the same reasoning as for the *integral test* for series convergence. Since $\psi_{\nu}(y_u + t)$ is a decreasing function of $t \geq y_{\nu}^*$ (as shown in Figure 3.2),

$$\begin{aligned} h \sum_{j=1}^{\infty} \psi_{\nu}(y_u + hj) &\leq \int_{y_u}^{\infty} \psi_{\nu}(t) dt = \int_{y_u}^{\infty} f_{\nu}(x(y)) \frac{dx(y)}{dy} dy \\ &= P(R > \nu^{1/2} x(y_u)), \\ &= 1 - Q_{\nu}(\nu x^2(y_u)) \end{aligned}$$

where Q_{ν} denotes the χ_{ν}^2 cdf.

Similarly, since $\psi_\nu(y_\ell + t)$ is an increasing function of $t \in (-\infty, y_\nu^*]$,

$$h \sum_{j=-\infty}^{-1} \psi_\nu(y_\ell + hj) \leq \int_{-\infty}^{y_\ell} \psi_\nu(t) dt = Q_\nu(\nu x^2(y_\ell)).$$

Therefore (3.9) is bounded above by $u_\nu(y_\ell, d)$. \square

3.4.3 A simple and effective procedure for evaluating the integral (3.6)

Suppose that we are given the value $\epsilon > 0$ of a desired upper bound on the absolute value of the approximation error that we will develop. We now describe a simple and effective procedure for evaluating the integral (3.6), to roughly this accuracy, that leads to a nested sequence of quadrature rules. This procedure, which is similar to that described by Mori (1988, pp.370–371), consists of the following steps:

Step 1: Choose y_ℓ and d and an initial value of n

The upper bound (3.3) on the *discretization error* suggests that, for a given value of the upper bound on the *trimming error*, as given in Lemma 3.4.1, it makes sense to minimize h . This provides the motivation for the following choice of d . Choose d such that

$$\min_y u_\nu(y, d) = 10^{-3} \epsilon.$$

Choose y_ℓ to be the value of y minimizing $u_\nu(y, d)$. This will ensure that the magnitude of the approximation error will be dominated by the *discretization error*. This is not as wasteful of evaluations of the integrand $g_\nu(y)$ as it might seem at first since $g_\nu(y)$ has *double exponential* decay as $y \rightarrow \pm\infty$. We have chosen the initial value of n to be 5. Proceed to the next step.

Step 2: For given (n, h, y_ℓ) , evaluate the approximation (3.8)

Evaluate the approximation (3.8) and store the result. Using the stored

values of the approximations decide whether or not to stop the procedure. Because the magnitude of the approximation error is dominated by the *discretization error*, this stopping rule can depend simply on estimating the *discretization errors*, as in the procedure described by Mori (1988, pp.370–371). Proceed to the next step.

Step 3: Halve h and go back to the previous step

3.4.4 An exponentially convergent procedure for evaluating the integral (3.6)

While the procedure described in the previous subsection is simple to program and effective (as evidenced by the numerical results presented in Section 3.5), it does not lead to exponential convergence. We now describe a procedure that results in a nested sequence of quadrature rules that, under Assumption FT, is exponentially convergent. The fact that $g_\nu(y)$ has *double exponential* decay as $y \rightarrow \pm\infty$, whereas its Fourier transform $G_\nu(\omega)$ has only *single exponential* decay as $\omega \rightarrow \pm\infty$, implies that, at each iteration, d should be increased at a slower rate than $1/h$. By adding a given positive number $2b$ to d and halving h at each iteration, we obtain exponential convergence. For simplicity of exposition, we have not included a stopping rule in the description of this procedure.

Step 1: An initial choice of a reasonable value of (y_ℓ, n, d)

Choose an initial value of n , which we denote by n_0 . The initial value of h , denoted by h_0 , is the initial value of d (to be specified shortly) divided by n_0 . We choose b to be some small positive integer multiple of h_0 . For the sake of concreteness, we have chosen $b = h_0$. The initial value of d is such that

$$\min_y u_\nu(y, d) \text{ is equal to some specified small positive number.}$$

The initial value of y_ℓ , denoted by $y_{\ell 0}$, is the value of y minimizing $u_\nu(y, d)$

for the chosen initial value of d , denoted by d_0 . Let $y_{u0} = y_{\ell 0} + d_0$. Proceed to the next step.

Step 2: For given (y_ℓ, n, d) , evaluate the approximation (3.8)

Evaluate the approximation (3.8) and store the result. Proceed to the next step.

Step 3: Add $2b$ to d , halve h and choose the new value of y_ℓ

Add $2b$ to d and halve h . Choose the new value of y_ℓ to be $y_\ell - b$. It will be convenient for the proof of exponential convergence to define the iteration number k by $h = h_0/2^k$. Go back to the previous step.

The following theorem and its proof, given in Appendix B.3, is due to Paul Kabaila. This theorem states that under Assumption FT this procedure is exponentially convergent. The type of convergence described in this theorem is consistent with that other *double exponential* types of quadrature formulas (Mori & Sugihara, 2001).

Theorem 3.4.1. *Suppose that Assumption FT holds. Then the magnitude of the approximation error is, for all sufficiently large iteration numbers k , bounded above by*

$$\begin{aligned} & \frac{10 \tau_\nu}{9\nu} \left(\exp \left(-\frac{\nu}{2} \exp \left(\frac{9 y_{u0}}{10} \right) 2^{c_T k} \right) + \exp \left(-\nu \exp \left(-\frac{9 y_{\ell 0}}{10} \right) 2^{c_T k} \right) \right) \\ & + 2 c_4 \exp \left(-\left(\frac{2 \pi c_{FT}}{h_0} \right) 2^k \right), \end{aligned}$$

where $c_T = 9 h_0 / (10 \log_e(2))$. Since, at iteration number k , $n = (n_0 + 2k) 2^k$, the magnitude of the approximation error converges exponentially to 0 as $n \rightarrow \infty$.

Proof. Suppose that Assumption FT holds. By the proof of Lemma 3.4.1, the *trimming error* for iteration number k , is bounded above by

$$\int_{y_{u0}+kh_0}^{\infty} \psi_{\nu}(t) dt + \int_{-\infty}^{y_{\ell 0}-kh_0} \psi_{\nu}(t) dt. \quad (3.10)$$

As proved in part (a) and part (b) of Appendix B.3, there exist $t_1 < \infty$ and $t_2 > -\infty$ such that

$$\psi_{\nu}(t) \leq \frac{\tau_{\nu}}{2} \exp \left(-\frac{\nu}{2} \exp \left(\frac{9}{10} t \right) \right) \quad \text{for all } t \geq t_1$$

and

$$\psi_{\nu}(t) \leq \tau_{\nu} \exp \left(-\nu \exp \left(-\frac{9}{10} t \right) \right) \quad \text{for all } t \leq t_2.$$

It follows from this that

$$\int_y^{\infty} \psi_{\nu}(t) dt \leq \frac{10 \tau_{\nu}}{9 \nu} \exp \left(-\frac{\nu}{2} \exp \left(\frac{9}{10} y \right) \right) \quad \text{for all } y \geq t_1$$

and

$$\int_{-\infty}^y \psi_{\nu}(t) dt \leq \frac{10 \tau_{\nu}}{9 \nu} \exp \left(-\nu \exp \left(-\frac{9}{10} y \right) \right) \quad \text{for all } y \leq t_2$$

as shown in part (c) of Appendix B.3.

Therefore, from part (d) of Appendix B.3, for all sufficiently large iteration numbers k , (3.10) is bounded above by

$$\frac{10 \tau_{\nu}}{9 \nu} \left(\exp \left(-\frac{\nu}{2} \exp \left(\frac{9 y_{u0}}{10} \right) 2^{c_T k} \right) + \exp \left(-\nu \exp \left(-\frac{9 y_{\ell 0}}{10} \right) 2^{c_T k} \right) \right),$$

where $c_T = 9h_0/(10 \log_e(2))$.

According to part (e) of Appendix B.3, it follows from the upper bound (3.3) on the *discretization error* and Assumption FT that, for all sufficiently large iteration numbers k , the *discretization error* is bounded above by

$$3 c_4 \exp \left(-\left(\frac{2 \pi c_{FT}}{h_0} \right) 2^k \right).$$

□

3.5 Numerical results for the transformation (2.6) of Mori (1988), followed by the application of the trapezoidal rule

In this section, we use the simple test scenario that consists of evaluating a known univariate t probability.

Through the consideration of the coverage probability of a $1-\alpha$ t-interval, it may be shown that

$$1 - \alpha = \int_0^\infty a_{\nu,\alpha}(x) f_\nu(x) dx, \quad (3.11)$$

where

$$a_{\nu,\alpha}(x) = 2 \Phi(t_{\nu,1-\alpha/2} x) - 1,$$

with Φ the $N(0,1)$ cdf and the quantile $t_{\nu,a}$ defined by $P(T \leq t_{\nu,a}) = a$ for $T \sim t_\nu$. Figure 3.3 provides an illustration of the fact that the $a_{\nu,\alpha}(x)$'s are smooth bounded functions of x for the values of α considered and all positive integers ν . This figure presents graphs of $a_{\nu,\alpha}(x)$ as a function of x for $\alpha = 0.05$ and $\nu = 1, 2$ and ∞ . The graph labeled $\nu = \infty$ refers to the case that $t_{\nu,1-\alpha/2}$ is replaced by its limit, as $\nu \rightarrow \infty$.

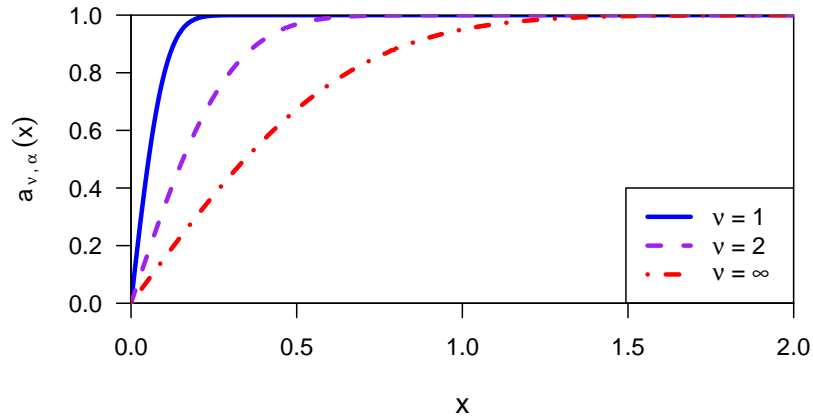


Figure 3.3: Graphs of $a_{\nu,\alpha}(x)$ as a function of x for $\alpha = 0.05$ and $\nu = 1, 2$ and ∞

Apply the transformation (2.6) of Mori (1988), so that

$$\int_0^\infty a_{\nu,\alpha}(x) f_\nu(x) dx = \int_{-\infty}^\infty g_{\nu,\alpha}(y) dy, \quad (3.12)$$

where $g_{\nu,\alpha}(y) = a_{\nu,\alpha}(x(y)) \psi_\nu(y)$.

We apply the simple procedure described in subsection 3.4.3, with $\epsilon = 10^{-17}$ and stopped after the computation of the approximation for $n = 65$ for $\nu = 1$ and $n = 33$ for $\nu = 2, 3, 4, 5, 10, 100$ and 1000 . The approximation error is defined to be this approximation minus $1 - \alpha$. Table 3.1 presents the approximation error for $\alpha = 0.10, 0.05$ and 0.02 and $\nu = 1, 2, 3, 4, 5, 10, 100$ and 1000 . Due to the finite precision of our computations in \mathbb{R} , we interpret an entry 0 in this table as $|\text{approximation error}| < 1.11 \times 10^{-16}$.

Table 3.1: The approximation error for the simple procedure described in subsection 3.4.3, with $\epsilon = 10^{-17}$ and stopped after the computation of the approximation for $n = 65$ for $\nu = 1$ and $n = 33$ for $\nu = 2, 3, 4, 5, 10, 100$ and 1000 . Here $\alpha = 0.10, 0.05$ and 0.02 and $\nu = 1, 2, 3, 4, 5, 10, 100$ and 1000 . We interpret an entry 0 in this table as $|\text{approximation error}| < 1.11 \times 10^{-16}$.

α	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$
0.10	-1.11×10^{-16}	-2.22×10^{-16}	0	-2.22×10^{-16}	-1.11×10^{-16}
0.05	9.66×10^{-15}	2.10×10^{-14}	-1.11×10^{-16}	-1.11×10^{-16}	-1.11×10^{-16}
0.02	-1.23×10^{-12}	5.82×10^{-11}	9.99×10^{-16}	-1.11×10^{-16}	0
α	$\nu = 10$	$\nu = 100$	$\nu = 1000$		
0.10	2.00×10^{-15}	2.78×10^{-14}	-2.37×10^{-13}		
0.05	2.11×10^{-15}	2.94×10^{-14}	-2.49×10^{-13}		
0.02	2.22×10^{-15}	3.03×10^{-14}	-2.57×10^{-13}		

3.6 Comparison with three other methods of numerical integration

In this section, we compare the performance of the method applied in the previous section with three other methods (a) Gauss Legendre quadrature, (b) Generalized Gauss Laguerre quadrature and (c) inverse cdf method followed by the Gauss Legendre quadrature. We consider the same simple test scenario, described in the previous section, that consists of evaluating a known univariate t probability. The purpose of this comparison is to illustrate the factors that may lead to a relatively poor performance of these three methods.

3.6.1 Gauss Legendre quadrature

Once having the truncation parameters (n, h, y_l) from Step 1 in subsection 3.4.3, one can simply use the Gauss-Legendre quadrature (instead of the application of the trapezoidal rule) to evaluate the truncated integral

$$\int_{y_l}^{y_u} a_{\nu, \alpha}(x(y)) \psi_{\nu}(y) dy. \quad (3.13)$$

This application is made in the usual way by first carrying out a straight line transformation of the interval $[y_l, y_u]$ to $[-1, 1]$. We first change the variable of integration in (3.13) to

$$s = \frac{2}{y_u - y_l} \left(y - \left(\frac{y_l + y_u}{2} \right) \right)$$

so that $ds = 2 dy / (y_u - y_l)$. Now

$$(3.13) = \frac{y_u - y_l}{2} \int_{-1}^1 a \left(x \left(\frac{(y_u - y_l)}{2} s + \frac{(y_l + y_u)}{2} \right) \right) \times$$

$$\begin{aligned}
& \psi_\nu \left(\frac{(y_u - y_l)}{2} s + \frac{(y_l + y_u)}{2} \right) ds \\
&= \frac{y_u - y_l}{2} \int_{-1}^1 k_\nu(s) ds,
\end{aligned} \tag{3.14}$$

where

$$k_\nu(s) = a \left(x \left(\frac{(y_u - y_l)}{2} s + \frac{(y_l + y_u)}{2} \right) \right) \psi_\nu \left(\frac{(y_u - y_l)}{2} s + \frac{(y_l + y_u)}{2} \right).$$

We then approximate (3.14), using Gauss Legendre quadrature with m nodes, by

$$\frac{y_u - y_l}{2} \sum_{j=1}^m \tilde{w}_j k_\nu(s_j) \tag{3.15}$$

for the appropriately chosen \tilde{w}_j 's (which are all positive) and s_j 's ($-1 < s_1 < \dots < s_m < 1$). We define the approximation error to be (3.15) minus $1 - \alpha$.

The resulting approximation errors are shown in Table 3.2. The magnitudes of these approximation errors are all larger than the magnitudes of the corresponding approximation errors for the trapezoidal rule, with the same number of integrand evaluations, reported in Table 3.1. In other words, for the same number of evaluations of the integrand, the trapezoidal rule outperforms Gauss-Legendre quadrature applied to the evaluation of (3.13), in terms of magnitude of approximation error. This result may be explained by the fact that the Gauss-Legendre quadrature nodes, which lie in the interval $[-1, 1]$, cluster near the values -1 and 1 , where the transformed integrand takes values very close to zero. This clustering also leads to Gauss-Legendre quadrature nodes not far from 0 , where the transformed integrand differs most from zero, being more widely spaced than for the trapezoidal rule, with the same number of evaluations of the integrand.

Table 3.2: The approximation error for Gauss Legendre quadrature applied to the evaluation of (3.13), for $\alpha = 0.10, 0.05$ and 0.02 and $\nu = 1, 2, 3, 4, 5, 10, 100$ and 1000 . The number of nodes m is 65 for $\nu = 1$ and 33 for $\nu = 2, 3, 4, 5, 10, 100$ and 1000 .

α	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$
0.10	-2.84×10^{-13}	-2.02×10^{-10}	-1.26×10^{-13}	-2.47×10^{-13}	-6.75×10^{-14}
0.05	-1.67×10^{-10}	-2.01×10^{-8}	-6.44×10^{-11}	8.88×10^{-13}	1.47×10^{-12}
0.02	2.20×10^{-7}	-5.48×10^{-7}	-3.31×10^{-9}	1.70×10^{-10}	1.15×10^{-12}
α	$\nu = 10$	$\nu = 100$	$\nu = 1000$		
0.10	-3.80×10^{-14}	-9.33×10^{-13}	2.98×10^{-13}		
0.05	3.09×10^{-14}	-1.06×10^{-12}	3.74×10^{-13}		
0.02	6.85×10^{-14}	-1.13×10^{-12}	4.36×10^{-13}		

An additional advantage of the trapezoidal rule is that, unlike Gauss-Legendre quadrature, it leads to a nested sequence of quadrature rules that can be used for the estimation of the approximation error.

3.6.2 Generalized Gauss Laguerre quadrature

To apply Generalized Gauss Laguerre quadrature to the evaluation of (3.1), we first change the variable of integration to $y = \nu x^2/2$, so that

$$\int_0^\infty a(x) f_\nu(x) dx = \frac{1}{\Gamma(\nu/2)} \int_0^\infty d_\nu(y) c(y) dy,$$

where $c(y) = y^{(\nu/2)-1} \exp(-y)$ and $d_\nu(y) = a((2y/\nu)^{1/2})$. We then apply Generalized Gauss Laguerre quadrature, with m nodes (samples), to approximate

$$\int_0^\infty d_\nu(y) c(y) dy$$

by

$$\sum_{j=1}^m w_j d_\nu(y_j) \tag{3.16}$$

for the appropriately chosen w_j 's (which are all positive) and y_j 's ($0 < y_1 < \dots < y_m < \infty$). We define the approximation error to be (3.16) minus $1 - \alpha$.

Graphs of $d_\nu(y)$ as a function of y are shown in Figure 3.4 for $\nu = 1, 2, 3$ and 10 and $\alpha = 0.10, 0.05$ and 0.02. It should be noted that the horizontal scales in each of the four panels of this figure are very different. It is known that Generalized Gauss Laguerre quadrature with m nodes will lead to the exact result if $d_\nu(y)$ is a polynomial in $y \in [0, \infty)$ of degree $2m - 1$ (Chandrasekhar, 1960, p.65).

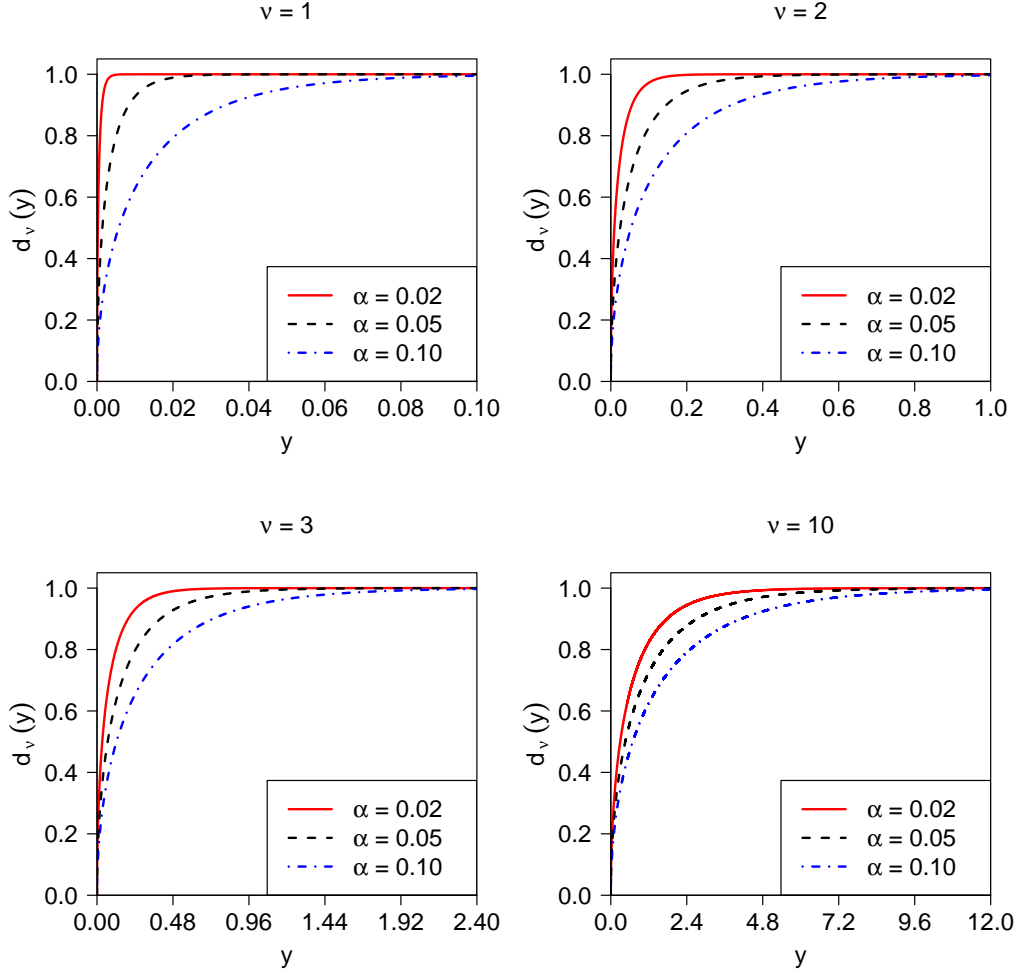


Figure 3.4: Graphs of $d_\nu(y)$ as a function of y for $\nu = 1, 2, 3$ and 10 and $\alpha = 0.10, 0.05$ and 0.02

We assess how well $d_\nu(y)$ can be approximated by a polynomial, over the finite interval of values of y such that $c(y)/\Gamma(\nu/2)$ is substantially greater than 0 , as follows. Any polynomial p of degree u can be written as

$$p(y) = a_0 - \sum_{j=1}^u a_j (1-y)^j$$

as shown in [Appendix B.4](#).

Set $a_0 = 1$ and require that $\sum_{j=1}^u a_j = 1$, so that the functions p and d_ν take the same values at both $y = 0$ and $y = 1$. A first approximation to $d_\nu(y)$ by $p(y)$ over the interval $y \in [0, 1]$ is obtained by minimizing a measure

of distance between $d_\nu(y)$ and $1 - (1 - y)^j$, over $j \in \{1, \dots, u\}$. A better approximation is obtained by minimizing a measure of distance between $d_\nu(y)$ and $1 - \sum_{j=1}^u a_j(1 - y)^j$, over a_1, \dots, a_u , subject to $\sum_{j=1}^u a_j = 1$. It follows from the shapes of the graphs in Figure 3.5 that to approximate $d_\nu(y)$ well by a polynomial, over the finite interval of values of y such that $c(y)/\Gamma(\nu/2)$ is substantially greater than 0, we would require this polynomial to be of very high degree, particularly for small ν . This suggests that Generalized Gauss Laguerre quadrature, with a given number of nodes m , will be most inaccurate for $\nu = 1$ and will have increasing accuracy as ν increases.

This suggested result is borne out by Table 3.3, which lists the approximation error for Generalized Gauss Laguerre quadrature for $\alpha = 0.10, 0.05$ and 0.02 and $\nu = 1, 2, 3, 4, 5, 6, 10, 100$ and 300 . We have chosen the number of nodes m to be the same as the number of integrand evaluations in Table 3.1. In other words, the number of nodes m is 65 for $\nu = 1$ and 33 for $\nu = 2, 3, 4, 5, 6, 10, 100$ and 300 .

Table 3.3: The approximation error for Generalized Gauss Laguerre quadrature for $\alpha = 0.10, 0.05$ and 0.02 and $\nu = 1, 2, 3, 4, 5, 6, 10, 100$ and 300 . The number of nodes m is 65 for $\nu = 1$ and 33 for $\nu = 2, 3, 4, 5, 6, 10, 100$ and 300 .

α	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$
0.10	1.44×10^{-2}	1.32×10^{-3}	1.63×10^{-4}	2.86×10^{-5}	6.08×10^{-6}
0.05	3.25×10^{-2}	2.04×10^{-3}	2.26×10^{-4}	3.77×10^{-5}	7.84×10^{-6}
0.02	2.00×10^{-2}	4.12×10^{-3}	3.39×10^{-4}	5.24×10^{-5}	1.05×10^{-5}
α	$\nu = 6$	$\nu = 10$	$\nu = 100$	$\nu = 300$	
0.10	1.48×10^{-6}	1.23×10^{-8}	-2.00×10^{-14}	-6.22×10^{-15}	
0.05	1.88×10^{-6}	1.52×10^{-8}	-2.11×10^{-14}	-6.21×10^{-15}	
0.02	2.46×10^{-6}	1.91×10^{-8}	-2.18×10^{-14}	-6.43×10^{-15}	

Further confirmation of the unsuitability of Generalized Gauss Laguerre quadrature, in the scenario under consideration, for $\nu = 1$ and $\nu = 2$ is provided by Figure 3.5. The top and bottom panels of this figure are scatterplots of the (y_j, w_j) 's for $(\nu, m) = (1, 65)$ and $(\nu, m) = (2, 33)$, respectively ($y_j \leq 50$). For $(\nu, m) = (1, 65)$ and $(\nu, m) = (2, 33)$ there are 30 values of $y_j > 50$ and 9 values of $y_j > 50$, respectively. When we compare the top panel of Figure 3.5 with the top left panel (the case $\nu = 1$) of Figure 3.4, we observe the following. Generalized Gauss Laguerre quadrature uses very few samples for the values of y where the function $d_\nu(y)$ is changing rapidly with increasing y , while using a large number of samples for values of y at which this function hardly changes with increasing y . Indeed, for $(\nu, m) = (1, 65)$ there are only 2 nodes in the interval $[0, 0.1]$. A similar conclusion results from comparing the bottom panel of Figure 3.5 with the top right panel (the case $\nu = 2$) of Figure 3.4. For $(\nu, m) = (2, 33)$ there are only 3 nodes in the interval $[0, 1]$.

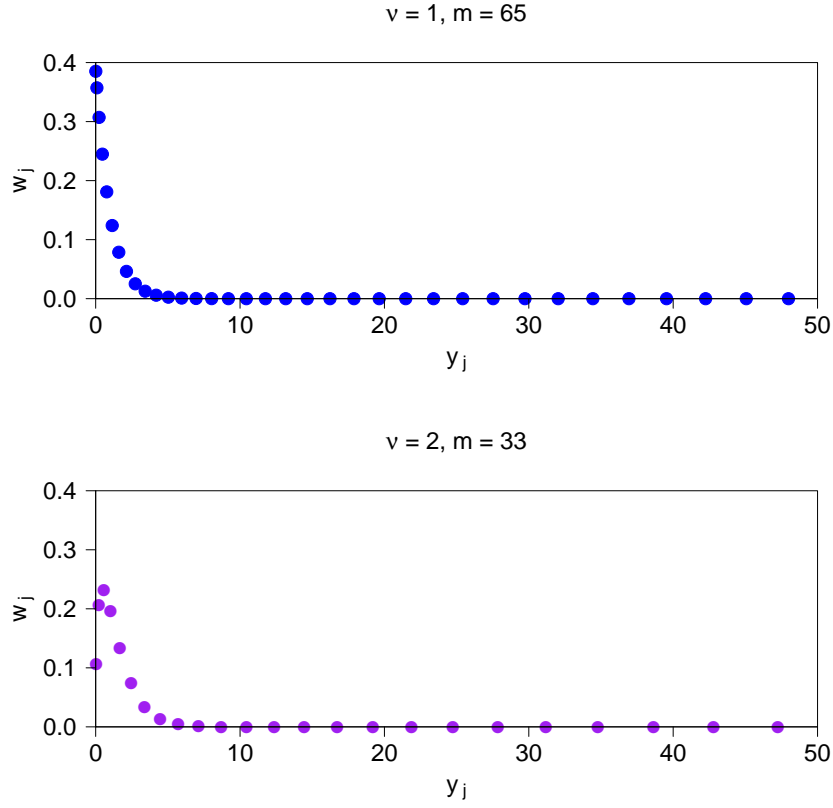


Figure 3.5: The top and bottom panels are scatterplots of the (y_j, w_j) 's for $(\nu, m) = (1, 65)$ and $(\nu, m) = (2, 33)$, respectively

Of course, one could greatly increase the number of nodes m and then approximate (3.16) by $\sum_{j=1}^q w_j d_\nu(y_j)$, where q is much less than m . This is unsatisfactory for the following two reasons. Firstly, the *raison d'être* of Generalized Gauss Laguerre quadrature is that with m nodes it leads to the exact result for polynomials of degree $2m - 1$. This fundamental property is lost when this approximation is carried out. Secondly, this is a rather ad hoc way of forcing more samples of the function $d_\nu(y)$ into the quadrature formula for the values of y for which this function changes rapidly with increasing y .

3.6.3 Inverse cdf method, followed by Gauss Legendre quadrature

Change the variable of integration to $y = F_\nu(x)$, where F_ν denotes the cdf corresponding to the pdf f_ν , so that $x = F_\nu^{-1}(y)$, $dy = f_\nu(x) dx$ and

$$\int_0^\infty a(x) f_\nu(x) dx = \int_0^1 a(F_\nu^{-1}(y)) dy.$$

A similar transformation is used, for example, by [Genz & Bretz \(2009, p.32\)](#). As shown in [Appendix B.5](#), if desired, we can compute $F_\nu^{-1}(y)$ using either $F_\nu^{-1}(y) = (Q_\nu^{-1}(y)/\nu)^{1/2}$ or $F_\nu^{-1}(y) = F_R^{-1}(y)/\nu^{1/2}$, where F_R denotes the χ_ν cdf of R . We then change the variable of integration to $z = 2y - 1$, so that $dy = dz/2$, to obtain

$$\int_0^1 a(F_\nu^{-1}(y)) dy = \int_{-1}^1 b_\nu(z) dz,$$

where $b_\nu(z) = a(F_\nu^{-1}((z+1)/2)) / 2$. We then approximate the right-hand side, using Gauss Legendre quadrature with m nodes, by

$$\sum_{j=1}^m \tilde{w}_j b_\nu(z_j) \tag{3.17}$$

for the appropriately chosen \tilde{w}_j 's (which are all positive) and z_j 's ($-1 < z_1 < \dots < z_m < 1$). We define the approximation error to be (3.17) minus $1 - \alpha$.

Graphs of $b_\nu(z)$ as a function of z are shown in [Figure 3.6](#) for $\nu = 1, 3, 10$ and 100 and $\alpha = 0.10, 0.05$ and 0.02 . It should be noted that the horizontal scale for the $\nu = 1$ panel is different from the horizontal scale for the $\nu = 3$, $\nu = 10$ and $\nu = 100$ panels (which are the same). It is known that Gauss Legendre quadrature with m nodes will lead to the exact result if $b_\nu(z)$ is a polynomial in $z \in [-1, 1]$ of degree $2m - 1$. When interpreting [Figure 3.6](#), it is important to remember that $b_\nu(-1) = 0$ and that $b_\nu(z)$ is an increasing

continuous function of $z \in [-1, 1]$. It is evident, then, from this figure that $b_\nu(z)$ increases very rapidly as z increases from zero for $\nu = 10$ and $\nu = 100$.

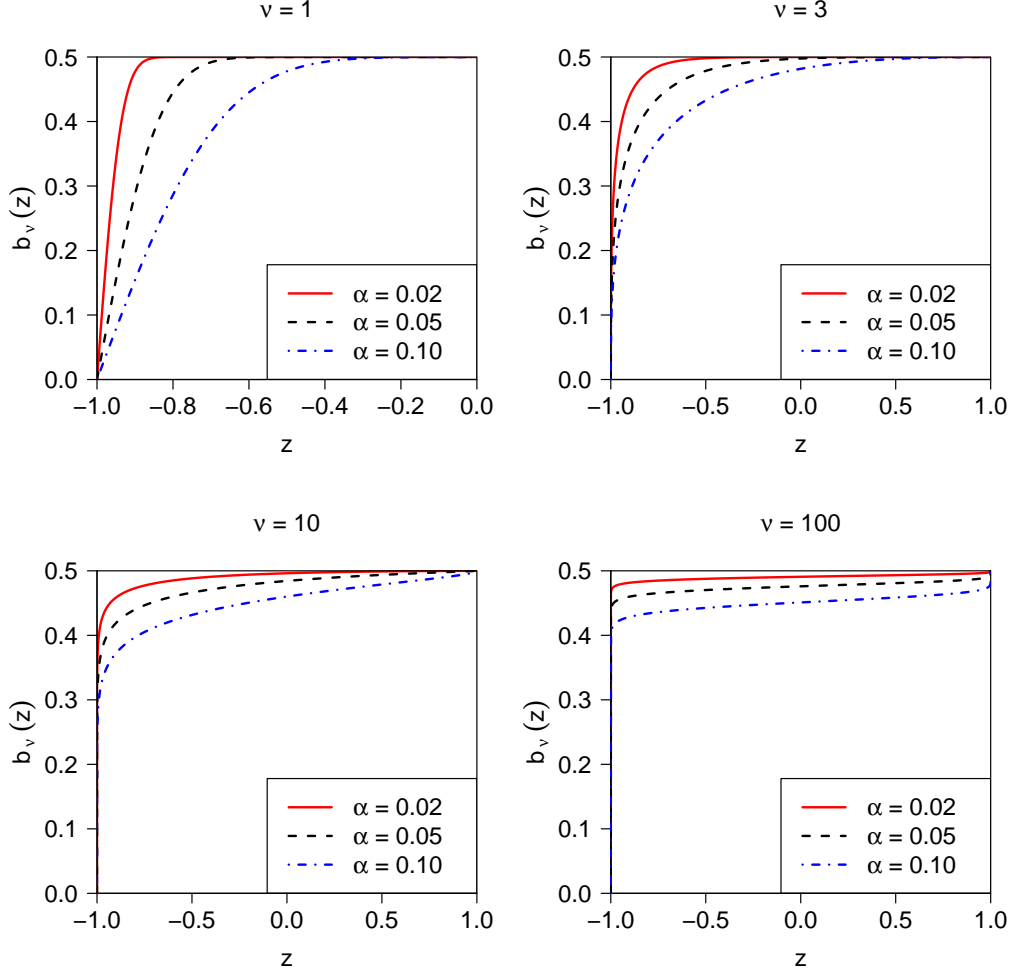


Figure 3.6: Graphs of $b_\nu(z)$ as a function of z for $\nu = 1, 3, 10$ and 100 and $\alpha = 0.10, 0.05$ and 0.02

It follows from Figure 3.6 and the same kinds of considerations as in subsection 3.6.2 that the degree of the polynomial in z needed to approximate $b_\nu(z)$ well in the interval $z \in [-1, 1]$ increases with increasing ν . This suggests that the inverse cdf method, using Gauss Legendre quadrature with a given number of nodes m , will be most accurate for $\nu = 1$ and will have decreasing accuracy as ν increases.

This suggested result is borne out by the first 7 columns (the columns

labelled $\nu = 1$ to $\nu = 10$) of Table 3.4, which lists the approximation error for Gauss Legendre quadrature for $\alpha = 0.10, 0.05$ and 0.02 and $\nu = 1, 2, 3, 4, 5, 6, 10, 100$ and 1000 . We have chosen the number of nodes m to be the same as the number of integrand evaluations in Table 3.1. In other words, the number of nodes m is 65 for $\nu = 1$ and 33 for $\nu = 2, 3, 4, 5, 6, 10, 100$ and 1000 .

Table 3.4: The approximation error for the inverse cdf method, using Gauss Legendre quadrature, for $\alpha = 0.10, 0.05$ and 0.02 and $\nu = 1, 2, 3, 4, 5, 6, 10, 100$ and 1000 . The number of nodes m is 65 for $\nu = 1$ and 33 for $\nu = 2, 3, 4, 5, 6, 10, 100$ and 1000 .

α	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$
0.10	7.77×10^{-16}	6.39×10^{-6}	1.52×10^{-5}	2.07×10^{-5}	2.37×10^{-5}
0.05	8.88×10^{-16}	9.43×10^{-6}	2.06×10^{-5}	2.70×10^{-5}	2.96×10^{-5}
0.02	8.88×10^{-16}	1.53×10^{-5}	2.94×10^{-5}	3.58×10^{-5}	3.75×10^{-5}
α	$\nu = 6$	$\nu = 10$	$\nu = 100$	$\nu = 1000$	
0.10	2.47×10^{-5}	2.30×10^{-5}	4.01×10^{-6}	4.23×10^{-7}	
0.05	3.02×10^{-5}	2.64×10^{-5}	4.06×10^{-6}	4.23×10^{-7}	
0.02	3.68×10^{-5}	2.90×10^{-5}	3.36×10^{-6}	3.33×10^{-7}	

3.7 Application to the computation of the coverage probability and scaled expected volume of non-standard confidence regions

To assess the coverage probability and expected volume properties of the non-standard confidence regions considered in the references co-authored with Kabaila, one needs to evaluate, for given ν , integrals of the form (3.1) for

hundreds, or thousands or even tens of thousands of different functions a . Each of these functions is smooth and bounded and the evaluation of $a(y)$ for any given y is computationally expensive. In this case, the following “set-up costs” are negligible:

1. For the simple procedure described in Section 3.4 (the transformation (2.6) of Mori (1988), followed by application of the trapezoidal rule), the “set-up cost” is computing y_ℓ and d .
2. For Gauss Legendre quadrature, the “set-up cost” is computing y_ℓ , d and the weights, \tilde{w}_j ’s, and nodes, s_j ’s, for this quadrature, followed by the computation of the $((y_u - y_\ell) s_j)/2 + (y_\ell + y_u)/2$ ’s.
3. For Generalized Gauss Laguerre quadrature, the “set-up cost” is computing the weights, w_j ’s, and nodes, y_j ’s, for this quadrature, followed by the computation of the $(2y_j/\nu)^{1/2}$ ’s.
4. For the Inverse cdf method, the “set-up cost” consists of computing the weights, \tilde{w}_j ’s, and nodes, z_j ’s, for Gauss Legendre quadrature, followed by the computation of the $F_\nu^{-1}((z_j + 1)/2)$ ’s.

In other words, the number of evaluations of the function a provides a reasonable guide to the computational effort for each of these methods.

We now consider in detail the evaluations of integrals of the form (3.1) in the references co-authored with Kabaila. Kabaila & Giri (2009a), Kabaila & Giri (2009b), Kabaila & Tissera (2014) and Abeysekera & Kabaila (2017) need to evaluate integrals of the form

$$\int_0^\infty \lambda(x) x^\xi f_\kappa(x) dx \quad (3.18)$$

where ξ and κ are a positive integers and $\lambda : [0, \infty) \rightarrow \mathbb{R}$ is a smooth bounded function. This integral can be converted into the form (3.1), as shown in

Appendix B.6, by changing the variable of integration to $y = c(\kappa, \xi) x$, where $c(\kappa, \xi) = (\kappa/(\kappa + \xi))^{1/2}$, so that

$$\int_0^\infty \lambda(x) x^\xi f_\kappa(x) dx = \left(\frac{2}{\kappa}\right)^{\xi/2} \frac{\Gamma(\nu/2)}{\Gamma(\kappa/2)} \int_0^\infty a(y) f_\nu(y) dy, \quad (3.19)$$

where $\nu = \kappa + \xi$ and $a(y) = \lambda(y/c(\kappa, \xi))$ is a smooth bounded function of $y \geq 0$.

An important measure of the performance of a confidence interval is its coverage probability function. The assessment of the coverage probability functions of (a) the post-model-selection confidence intervals considered by Kabaila & Giri (2009b) and Kabaila & Farchione (2012) and (b) the frequentist model averaged confidence intervals considered by Kabaila *et al.* (2016) is carried out by plotting the graphs of these functions. This requires the evaluation, for some given ν , of an expression of the form (3.1) for, say, 200 different functions a . Each of these functions is smooth and bounded and the evaluation of $a(y)$ for any given y is computationally expensive.

Abeysekera & Kabaila (2017), Kabaila & Giri (2009a), Kabaila & Giri (2013) and Kabaila & Tissera (2014) construct non-standard confidence regions with guaranteed coverage using the following computations. They numerically optimize a criterion related to the expected volume of a parametric family of non-standard confidence regions, subject to a coverage probability constraint. The coverage probability, for a particular true parameter value, of a member of this family is given by an expression of the form (3.1), for some given ν . The function a is smooth and bounded and the evaluation of $a(y)$ for any given y is computationally expensive. As this numerical constrained optimization proceeds, the evaluation of an expression of the form (3.1) needs to be carried out for thousands or even tens of thousands of different functions a , for the same given ν .

3.8 Remarks

For the computation of y_ℓ and d in Step 1 of our procedure, we have evaluated Q_ν using the R function `pchisq`. This evaluation of Q_ν is carried out using well-established methods for the evaluation of the incomplete gamma integral. These methods include the series expansion described by [Shea \(1988\)](#), as well as a continued fraction expansion due to Gauss, which greatly simplifies for ν an even positive integer. As already noted, for the types of problems considered in the references co-authored with Kabaila, the “set-up cost” of Step 1 is negligible. However, if one really needed to reduce the computation time for Step 1 then one could do so by replacing the exact evaluation of the tail probabilities of the χ_ν^2 distribution by upper bounds (such as Chernoff bounds) on these probabilities and by simplifying the minimization and root finding steps needed to evaluate y_ℓ and d .

3.9 Conclusion

In Section [3.6](#), for both Generalized Gauss Laguerre quadrature and the inverse cdf method, we presented graphs whose features accurately predict their performance in terms of accuracy for a given number of evaluations of the function a . As noted in Section [3.7](#), the number of evaluations of the function a is a reasonable measure of computational effort when the “set-up costs” are negligible, as in the situations considered in the references co-authored with Kabaila.

Our findings for the test scenario considered in Section [3.6](#) are as follows. The Gauss Legendre quadrature method has poor performances for all the values of ν than the trapezoidal rule. The Generalized Gauss Laguerre quadrature method performs worst for $\nu = 1$, and has performance that improves with increasing ν . It has the worst performance of the four

methods for $\nu \in \{1, 2, 3, 4\}$. The inverse cdf method, using Gauss Legendre quadrature, has the best performance of the four methods for $\nu = 1$ and $\alpha \in \{0.05, 0.02\}$, and has performance that decreases as ν increases through the values 2, 3, 4, 5, 6 and 10. The method described in Section 3.4 (application of the transformation (2.6) of Mori, 1988) has the best performance for $\nu = 1$ and $\alpha = 0.1$, $\nu \in \{1, 2, 3, 4, 5, 10\}$, has very close to the best performance for $\nu = 100$ and has the best performance for $\nu = 1000$. For many of the situations considered in the references co-authored with Kabaila, the smallest possible value of ν , in the evaluation of integrals of the form (3.1), is 2.

The procedures described in Section 3.4 use a nested sequence of quadrature rules, for the estimation of the approximation error, so that previous evaluations of the integrand are not wasted. This nested sequence can be implemented in a very simple computer program. This is an important advantage of this method over the other three methods.

Taken together, the results presented in this chapter show that the simple procedure described in subsection 3.4.3 is a very suitable candidate for the computation of the coverage and expected volume properties of non-standard confidence regions considered in the references co-authored with Kabaila. We apply this method in the next chapter to compute the Kabaila & Giri (2009a) confidence interval.

Chapter 4

R programs for the computation of the [Kabaila & Giri \(2009a\)](#) confidence interval

4.1 Introduction

[Kabaila & Giri \(2009a\)](#) considered a linear regression model, with unknown error variance. They describe a method for the construction of a confidence interval for a given linear combination of the regression parameters that utilizes the uncertain prior that a distinct linear combination of these parameters takes a specified value. They also describe methods for the efficient computation of the coverage and scaled expected lengths of these confidence intervals. [Kabaila & Giri \(2009a\)](#) computed their confidence interval using **MATLAB** programs. These programs were not efficient and not accessible to users who do not have a **MATLAB** licence. The main reason for this inefficiency is that the **MATLAB** program `dblquad`, used for the numerical evaluation of double integrals, turned out to be very inefficient.

In this chapter, we describe the computational and theoretical innovations that we have implemented in the **R** programs for the computation of

the Kabaila & Giri (2009a) confidence interval. With the application of more efficient methods, we are able to greatly decrease the computation time. By writing these programs in R, the Kabaila & Giri (2009a) confidence interval becomes freely available. These R programs have been placed in the R package `ciuupi2`.

In Section 4.2 of this chapter, we describe the Kabaila & Giri (2009a) confidence interval in detail. In subsection 4.2.4, we describe the nonlinear constrained optimization that needs to be carried out to compute the Kabaila & Giri (2009a) confidence interval for a given choice of a parameter λ , that specifies the weight given to the uncertain prior information. Section 4.3 and subsection 4.4.1 provide a new method for the computation, using the computationally-convenient expressions due to Kabaila & Giri (2009a), of the coverage probability and scaled expected length of this confidence interval. One of the computational innovations we make is to apply Gauss Legendre quadrature, with an appropriate number of nodes, to evaluate the inner integrals of the double integral terms in the formulas for the coverage probability and the scaled expected length. Another important computational innovation is applying the work described in Chapter 3 of this thesis to evaluate the outer integrals of the double integrals that appear in these formulas.

A major theoretical innovation in the present chapter is the analysis of a (new) second definition of the scaled expected length. In subsection 4.4.2, we describe this second definition. We also show how the methods used for the computation of the confidence interval that utilizes the uncertain prior information in the context of the first definition extend to the second definition.

In Section 4.5, we provide a detailed description of the choice of some of the parameters of the new confidence interval and the method used for

its computation. We also compare our results with some of the past results obtained by [Kabaila & Giri \(2009a\)](#), [Kabaila & Giri \(2013\)](#) and [Giri \(2008\)](#) for a given choice of the parameter λ , which specifies the weight given to the uncertain prior information, using MATLAB programs. [Kabaila & Giri \(2009a\)](#) described a choice of the parameter λ that leads to what might be called a “standard” confidence interval that utilizes the uncertain prior information. However, because their MATLAB programs for a particular value of λ took so long to run, they could not compute this confidence interval. In Section 4.6, we describe this choice of the parameter value λ and the computation times using the new R package `ciuupi2`. Finally, in Section 4.7, we numerically compare the two definitions of scaled expected length in terms of the resulting confidence intervals that utilize uncertain prior information and their coverage and scaled expected lengths properties.

4.2 The confidence interval of [Kabaila & Giri \(2009a\)](#)

In this section, we briefly describe the [Kabaila & Giri \(2009a\)](#) confidence interval. In a later subsection, we provide a description of the nonlinear constrained optimization that needs to be carried out to compute this confidence interval.

Consider the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad (4.1)$$

where \mathbf{y} is a random n -vector of responses, \mathbf{X} is a known $n \times p$ matrix with linearly independent columns, $\boldsymbol{\beta}$ is an unknown parameter p -vector and $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$, with σ^2 an unknown positive parameter. Suppose that the parameter of interest is $\theta = \mathbf{a}^\top \boldsymbol{\beta}$, where \mathbf{a} is a specified nonzero p -vector. Let

$\tau = \mathbf{c}^\top \boldsymbol{\beta}$, where \mathbf{c} is a specified nonzero p -vector that is linearly independent of \mathbf{a} . Suppose that we have uncertain prior information that $\tau = t$, where t is a specified number. Our aim is to construct a confidence interval for θ , with minimum coverage probability $1 - \alpha$, that utilizes this uncertain prior information.

Let $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$, the least squares estimator of $\boldsymbol{\beta}$. Then $\hat{\theta} = \mathbf{a}^\top \hat{\boldsymbol{\beta}}$ and $\hat{\tau} = \mathbf{c}^\top \hat{\boldsymbol{\beta}}$ are the least squares estimators of θ and τ , respectively. Now let $v_\theta = \text{var}(\hat{\theta})/\sigma^2 = \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a}$, $v_\tau = \text{var}(\hat{\tau})/\sigma^2 = \mathbf{c}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{c}$ and $\rho = \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{c} / (v_\theta v_\tau)^{1/2}$, which are known quantities. Let $\hat{\sigma}^2 = (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})/m$, where $m = n - p$. Also let $W = \hat{\sigma}/\sigma$, so that it has the same distribution as $\sqrt{Q/m}$, where $Q \sim \chi_m^2$. Now let $\gamma = (\tau - t)/(\sigma v_\tau^{1/2})$ and $\hat{\gamma} = (\hat{\tau} - t)/(\hat{\sigma} v_\tau^{1/2})$.

We consider confidence intervals of the form

$$\text{CI}(b, s) = [\hat{\theta} - v_\theta^{1/2} \hat{\sigma} b(\hat{\gamma}) \pm v_\theta^{1/2} \hat{\sigma} s(\hat{\gamma})],$$

where $b : \mathbb{R} \rightarrow \mathbb{R}$ is an odd continuous function and $s : \mathbb{R} \rightarrow [0, \infty)$ is an even continuous function. In addition, $b(x) = 0$ and $s(x) = t_{m, 1-\alpha/2}$ for all $|x| \geq d$, where d is a sufficiently large positive number.

4.2.1 Choice of d

Giri (2008) found that the value of d that needed to be chosen for the resulting confidence interval to utilize the uncertain prior information effectively, depended on m , with d increasing as m decreases. For $m = 1$ and $m = 2$, Giri (2008) chose $d = 30$ and $d = 10$, respectively. Kabaila & Giri (2013), chose $d = 14$ for $m = 2$. Kabaila & Giri (2013), also chose $d = 12$ for $m = 1, 2, 3$ and 4 for the case that $\rho = 0$. The case that m is large is effectively equivalent to the case that σ^2 is known. In the case that σ^2 is known, Mainzer & Kabaila (2019) have chosen $d = 6$. We have automated the choice of d with

the new formula proposed by Paul Kabaila as described in subsection 4.5.1.

4.2.2 Parametric form for the functions b and s

We specify the functions b and s in much the same way as Kabaila & Giri (2009a). Suppose that x_1, \dots, x_q satisfy $0 = x_1 < x_2 < \dots < x_q = d$. Obviously, $b(x_1) = 0$, $b(x_q) = 0$ and $s(x_q) = t_{m,1-\alpha/2}$. The function b is fully specified by the vector $(b(x_2), \dots, b(x_{q-1}))$ as follows. Because b is an odd function, we know that $b(-x_i) = -b(x_i)$ for $i = 2, \dots, q$. We specify the value of $b(x)$ for any $x \in [-d, d]$ by cubic spline interpolation for these given function values. The function s is fully specified by the vector $(s(x_1), \dots, s(x_{q-1}))$ as follows. Because s is an even function, we know that $s(-x_i) = s(x_i)$ for $i = 1, \dots, q$. We specify the value of $s(x)$ for any $x \in [-d, d]$ by cubic spline interpolation for these given function values.

We consider two types of cubic spline interpolation: natural cubic spline interpolation and clamped cubic spline interpolation where the derivative of the cubic spline is set to 0 at $-d$ and d . Kabaila & Giri (2009a) used a slightly different interpolation procedure to evaluate the function s . We compute the natural cubic spline interpolation using the `splinefun` function in the R package `stats` and the clamped cubic spline interpolation using the `cubicspline` function in the R package `pracma`.

4.2.3 Choice of the knots x_1, \dots, x_q

If the number of knots q is too small, then the functions b and s do not have enough flexibility for the resulting confidence interval to effectively utilize the uncertain prior information. On the other hand, as the number of knots q is increased, the computation time increases and the numerical nonlinear constrained optimization seems to become unstable.

For the numerical example considered by Kabaila & Giri (2009a), the

knots are evenly-spaced and take the values $0, d/6, 2(d/6), \dots, d$, where $d = 6$. For $m = 1$ and $m = 2$, [Giri \(2008\)](#) chose $d = 30$ and $d = 10$, respectively, and the knots at $0, d/6, 2(d/6), \dots, d$. [Kabaila & Giri \(2013\)](#), chose $d = 14$ for $m = 2$ and the knots at $0, 2, 4, \dots, 14$. [Kabaila & Giri \(2013\)](#), also chose $d = 12$ for $m = 1, 2, 3$ and 4 for the case that $\rho = 0$ and the knots at $0, 1.5, 3, 4.5, 6, \dots, 12$. The case that m is large is effectively equivalent to the case that σ^2 is known. In the case that σ^2 is known, [Mainzer & Kabaila \(2019\)](#) have chosen $d = 6$. In all of these examples, the knots are equally-spaced.

4.2.4 Description of the nonlinear constrained optimization that needs to be carried out to compute the [Kabaila & Giri \(2009a\)](#) confidence interval for given λ

Assume that m and ρ are given. As shown by [Kabaila & Giri \(2009a\)](#), the coverage probability of $\text{CI}(b, s)$ is, for given functions b and s , an even function of γ . We denote this function by $\text{CP}(\gamma; b, s)$. A computationally convenient formula for $\text{CP}(\gamma; b, s)$ is provided in part (a) of Theorem 1 of [Kabaila & Giri \(2009a\)](#). [Giri \(2008\)](#) evaluated the double integral in this formula as follows. He first truncated the outer integral and found a convenient formula for an upper bound on the absolute value of the truncation error. He then used the MATLAB function `dblquad` to evaluate the resulting double integral. The function `dblquad` evaluates this double integral by treating it as two nested single integrals and evaluating each single integral using the MATLAB function `quad`. The function `quad` uses recursive adaptive Simpson quadrature.

[Shampine \(2008\)](#) critically reviews the method used by `dblquad` and proposes a two-dimensional adaptive quadrature method based on the product of a particular pair of Gauss-Kronrod formulas. Shampine's method was subsequently implemented in the MATLAB function `integral2`. On 11 November 2020, the MATLAB documentation recommends against the use of `dblquad` and proposes that `integral2` be used instead.

In Section 4.3, we describe a new method for evaluating the double integral in the formula for $CP(\gamma; b, s)$ given in Theorem 1(a) of [Kabaila & Giri \(2009a\)](#). We do not truncate the outer integral and use, instead, the new method described in Chapter 3. The inner integral has some special features that lead us to first divide it into a sum of integrals and then to apply Gauss Legendre quadrature to each integral in this sum.

The usual $1 - \alpha$ confidence interval for θ is $I = [\hat{\theta} - t_{m,1-\alpha/2} v_\theta^{1/2} \hat{\sigma}, \hat{\theta} + t_{m,1-\alpha/2} v_\theta^{1/2} \hat{\sigma}]$. The scaled expected length of $CI(b, s)$ is defined to be

$$\frac{E(\text{length of } CI(b, s))}{E(\text{length of } I)}. \quad (4.2)$$

As shown by [Kabaila & Giri \(2009a\)](#), this scaled expected length is, for given function s , an even function of γ . We denote this scaled expected length by $SEL_1(\gamma; s)$. A computationally convenient formula for $SEL_1(\gamma; s)$ is provided in part (b) of Theorem 1 of [Kabaila & Giri \(2009a\)](#). [Giri \(2008\)](#) evaluated the double integral in this formula in a manner similar to that used to evaluate the double integral in the computationally convenient formula for $CP(\gamma; b, s)$. In Section 4.4, we describe a new method for the evaluation of the computationally convenient formula for $SEL_1(\gamma; s)$ that is similar to the new method for the evaluation of the computationally convenient formula for $CP(\gamma; b, s)$.

Define the objective function

$$\text{OBJ}_1(s; \lambda) = \lambda \int_{-\infty}^{\infty} (\text{SEL}_1(\gamma; s) - 1) d\gamma + (\text{SEL}_1(0; s) - 1), \quad (4.3)$$

where λ is a positive tuning parameter that allows us to adjust the relative weight given to the uncertain prior information that $\gamma = 0$. A computationally convenient formula for this objective function was given by [Kabaila & Giri \(2009a\)](#), which was simplified in Appendix A of [Kabaila & Giri \(2013\)](#).

As in [Kabaila & Giri \(2009a\)](#), our aim is to find the functions b and s as follows. For given $\lambda > 0$, we minimize the objective function $\text{OBJ}_1(s; \lambda)$, with respect to the vector $(b(x_2), \dots, b(x_{q-1}), s(x_1), \dots, s(x_{q-1}))$, where the function s is constrained to take positive values, subject to the coverage constraint $\text{CP}(\gamma; b, s) \geq 1 - \alpha$ for all $\gamma \geq 0$. Denote the resulting confidence interval $\text{CI}(b_{1\lambda}, s_{1\lambda})$. Our computational implementation of this coverage constraint is that $\text{CP}(\gamma; b, s) \geq 1 - \alpha$ for all γ belonging to an appropriately-chosen finite set of nonnegative values, denoted by Γ_{grid} .

4.3 New method for evaluating the computationally convenient formula of [Kabaila & Giri \(2009a\)](#) for the coverage probability of $\text{CI}(b, s)$

Theorem 1(a) of [Kabaila & Giri \(2009a\)](#) provides a computationally convenient formula for $\text{CP}(\gamma; b, s)$, which we now describe. Let $\Psi(x, y; \mu, v) = P(x \leq Z \leq y)$ for $Z \sim N(\mu, v)$. Define the functions

$$k(x, w, \gamma, \rho) = \Psi\left(w(b(x) - s(x)), w(b(x) + s(x)); \rho(wx - \gamma), 1 - \rho^2\right)$$

$$k^\dagger(x, w, \gamma, \rho) = \Psi\left(-t_{m, 1-\alpha/2} w, t_{m, 1-\alpha/2} w; \rho(wx - \gamma), 1 - \rho^2\right).$$

Note that these definitions differ from the definitions of the functions k and k^\dagger used by Kabaila & Giri (2009a). By Theorem 1(a) of Kabaila & Giri (2009a),

$$\text{CP}(\gamma; b, s) = 1 - \alpha + \int_0^\infty \text{ICP}(w, \gamma, \rho) w f_m(w) dw, \quad (4.4)$$

where $f_m(w)$ denotes the pdf of $\sqrt{Q/m}$ where $Q \sim \chi_m^2$, evaluated at w , and

$$\text{ICP}(w, \gamma, \rho) = \int_{-d}^d (k(x, w, \gamma, \rho) - k^\dagger(x, w, \gamma, \rho)) \phi(wx - \gamma) dx.$$

We use this formula for $\text{ICP}(w, \gamma, \rho)$, but with the following modification.

Obviously,

$$\begin{aligned} \text{ICP}(w, \gamma, \rho) &= \int_0^d (k(x, w, \gamma, \rho) - k^\dagger(x, w, \gamma, \rho)) \phi(wx - \gamma) dx \\ &\quad + \int_{-d}^0 (k(x, w, \gamma, \rho) - k^\dagger(x, w, \gamma, \rho)) \phi(wx - \gamma) dx. \end{aligned}$$

Changing the variable of integration to $u = -x$ in the second integral on the right-hand side, we obtain

$$\begin{aligned} \text{ICP}(w, \gamma, \rho) &= \int_0^d \left((k(x, w, \gamma, \rho) - k^\dagger(x, w, \gamma, \rho)) \phi(wx - \gamma) \right. \\ &\quad \left. + (k(-x, w, \gamma, \rho) - k^\dagger(-x, w, \gamma, \rho)) \phi(wx + \gamma) \right) dx. \end{aligned}$$

In other words,

$$\text{ICP}(w, \gamma, \rho) = \int_0^d \text{HICP}(x, w, \gamma, \rho) dx, \quad (4.5)$$

where

$$\begin{aligned} \text{HICP}(x, w, \gamma, \rho) &= (k(x, w, \gamma, \rho) - k^\dagger(x, w, \gamma, \rho)) \phi(wx - \gamma) \\ &\quad + (k(-x, w, \gamma, \rho) - k^\dagger(-x, w, \gamma, \rho)) \phi(wx + \gamma). \end{aligned}$$

While interpolating cubic splines are extremely smooth between successive knots, they are not particularly smooth at the knots (only the second derivative of the interpolating cubic spline is continuous at each knot). We therefore express (4.5) as

$$\int_{x_1}^{x_2} \text{IICP}(x, w, \gamma, \rho) dx + \int_{x_2}^{x_3} \text{IICP}(x, w, \gamma, \rho) dx + \cdots + \int_{x_{q-1}}^{x_q} \text{IICP}(x, w, \gamma, \rho) dx$$

and then compute each of the integrals in this sum using Gauss Legendre quadrature, which is exact for polynomials of degree $2(\text{number of nodes}) - 1$. This computation is implemented using the `gauss.quad` function in the R package `statmod`.

4.3.1 Examination of $\text{IICP}(x, w, \gamma, \rho)$

The examinations of the two terms in the integrand $\text{IICP}(x, w, \gamma, \rho)$ are very similar. For the sake of brevity, we present only the examination of the first term. Observe that

$$\phi(wx - \gamma) = \phi\left(w\left(x - \frac{\gamma}{w}\right)\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2w^{-2}}\left(x - \frac{\gamma}{w}\right)^2\right). \quad (4.6)$$

Considered as a function of x , this function has maximum value $1/\sqrt{2\pi}$ and has peak width w^{-1} . Thus, for small and moderate values of w , $\phi(wx - \gamma)$ is a smooth function of $x \in [0, d]$. Also, for small and moderate values of w , $k(x, w, \gamma, \rho) - k^\dagger(x, w, \gamma, \rho)$ is a smooth function of $x \in [0, d]$.

We now consider the case that w is large and the potential problems that may arise in this case. It follows from (4.6) that $\phi(wx - \gamma) = w^{-1} \phi(x; \gamma/w, w^{-2})$, where $\phi(z; \mu, v)$ denotes the $N(\mu, v)$ pdf, evaluated at z . Thus

$$\begin{aligned} & \int_0^d (k(x, w, \gamma, \rho) - k^\dagger(x, w, \gamma, \rho)) \phi(wx - \gamma) dx \\ &= \frac{1}{w} \int_0^d (k(x, w, \gamma, \rho) - k^\dagger(x, w, \gamma, \rho)) \phi(x; \gamma/w, w^{-2}) dx \end{aligned} \quad (4.7)$$

The problematic term in (4.7) is $\phi(x; \gamma/w, w^{-2})$, whose graph has a peak centred at $x = \gamma/w$, which becomes increasingly narrow as w increases. Integrands with narrow peaks are very difficult for the numerical integration methods to deal with. However, it would seem that (4.7) is close to 0 when w is large. One reason for this is the $1/w$ term in front of the integral in (4.7). In addition, consider the following argument. Consider $\gamma \geq 0$ and $\rho \geq 0$. Suppose that w is so large that both γ/w and w^{-1} are small so that $\phi(x; \gamma/w, w^{-2})$ differs negligibly from 0, unless x is close to 0. Recall that

$$\begin{aligned} & k(x, w, \gamma, \rho) - k^\dagger(x, w, \gamma, \rho) \\ &= \Psi\left(w(b(x) - s(x)), w(b(x) + s(x)); (\rho x)w - \rho\gamma, 1 - \rho^2\right) \\ &\quad - \Psi\left(-t_{m,1-\alpha/2}w, t_{m,1-\alpha/2}w; (\rho x)w - \rho\gamma, 1 - \rho^2\right). \end{aligned}$$

Assume that $b(x) - s(x) < 0$. Obviously, $b(x) + s(x) > 0$. For x close to 0, $(\rho x)w - \rho\gamma$ increases more slowly with increasing w than either $w(b(x) + s(x))$ or $t_{m,1-\alpha/2}w$. Consequently, for x close to 0, $k(x, w, \gamma, \rho) - k^\dagger(x, w, \gamma, \rho)$ approaches 0 as w increases.

4.3.2 Evaluation of the outer integral of the double integral term in the expression of $\mathbf{CP}(\gamma; b, s)$

To evaluate the outer integral of the double integral term in (4.4), we apply the new method described in Chapter 3 as follows.

The integral (3.18) in Section 3.7 with $\kappa = m$ and $\xi = 1$ has a similar form to the integral in (4.4). We change the variable of integration in (4.4) to $y = c(m, 1)w$, where $c(\kappa, \xi) = (\kappa/(\kappa + \xi))^{1/2}$, so that

$$\begin{aligned} & \int_0^\infty \mathbf{ICP}(w, \gamma, \rho) w f_m(w) dw \\ &= \left(\frac{2}{m}\right)^{1/2} \frac{\Gamma((m+1)/2)}{\Gamma(m/2)} \int_0^\infty \mathbf{ICP}(y/c(m, 1), \gamma, \rho) f_{m+1}(y) dy, \end{aligned} \quad (4.8)$$

where $\text{ICP}(y/c(m, 1), \gamma, \rho)$ is a smooth bounded function of $y \geq 0$.

To evaluate the integral in (4.8), we first apply the transformation (2.6) of Mori (1988), namely

$$g(z) = \exp\left(\frac{1}{2}z - e^{-z}\right), \quad (4.9)$$

so that

$$\frac{dg(z)}{dz} = \exp\left(\frac{1}{2}z - e^{-z}\right) \left(\frac{1}{2} + e^{-z}\right)$$

and

$$\begin{aligned} & \int_0^\infty \text{ICP}(y/c(m, 1), \gamma, \rho) f_{m+1}(y) dy \\ &= \int_{-\infty}^\infty \text{ICP}(g(z)/c(m, 1), \gamma, \rho) \psi_{m+1}(z) dz, \end{aligned} \quad (4.10)$$

where

$$\psi_{m+1}(z) = f_{m+1}(g(z)) \frac{dg(z)}{dz}.$$

We approximate (4.10) by

$$h \sum_{j=0}^{N-1} \text{ICP}(g(z_\ell + hj)/c(m, 1), \gamma, \rho) \psi_{m+1}(z_\ell + hj), \quad (4.11)$$

where N denotes the number of evaluations of the integrand $\text{ICP}(g(z)/c(m, 1), \gamma, \rho) \psi_{m+1}(z)$, h denotes the step length and the first evaluation of this integrand is at z_ℓ . We evaluate (4.11) using the procedure described in subsection 3.4.3.

4.4 Computationally convenient formula for the scaled expected length of $\text{CI}(b, s)$

In this section, we consider two definitions of the scaled expected length.

We first describe the first definition of the scaled expected length considered

by [Kabaila & Giri \(2009a\)](#). Then we describe a second (new) definition of the scaled expected length of $CI(b, s)$. We also present new methods for numerically evaluating these two definitions of the scaled expected length efficiently.

4.4.1 First definition of scaled expected length and the objective function based on it

Theorem 1(b) of [Kabaila & Giri \(2009a\)](#) provides a computationally convenient formula for $SEL_1(\gamma; s)$, which is equal to

$$SEL_1(\gamma; s) = 1 + \frac{1}{t_{m,1-\alpha/2} E(W)} \int_0^\infty ISEL(w, \gamma) w^2 f_m(w) dw, \quad (4.12)$$

where

$$ISEL(w, \gamma) = \int_{-d}^d (s(x) - t_{m,1-\alpha/2}) \phi(wx - \gamma) dx. \quad (4.13)$$

The computation of $SEL_1(\gamma; s)$ is given in [Appendix C.1](#).

This definition of the scaled expected length leads to the first objective function [\(4.3\)](#). A computationally convenient formula for this objective function was given by [Kabaila & Giri \(2009a\)](#), which was simplified in [Appendix A](#) of [Kabaila & Giri \(2013\)](#), to yield

$$OBJ_1(s; \lambda) = \frac{2}{t_{m,1-\alpha/2} E(W)} \int_0^d IOBJ_1(x; s, \lambda) dx, \quad (4.14)$$

where

$$IOBJ_1(x; s, \lambda) = (s(x) - t_{m,1-\alpha/2}) \left(\lambda + \frac{1}{\sqrt{2\pi}} \left(\frac{m}{x^2 + m} \right)^{(m/2)+1} \right).$$

Note that the term

$$\frac{1}{\sqrt{2\pi}} \left(\frac{m}{x^2 + m} \right)^{(m/2)+1}$$

in the objective function is a smooth function of x for large m , as shown in

Appendix C.1.4. We compute the integral

$$\int_0^d \text{IOBJ}_1(x; s, \lambda) dx$$

similarly to the computation of $\text{ICP}(w, \gamma, \rho)$ in Section 4.3 using the Gauss Legendre quadrature.

4.4.2 Second definition of scaled expected length and the objective function based on it

The second definition of the scaled expected length of $\text{CI}(b, s)$ is

$$E\left(\frac{\text{length of CI}(b, s)}{\text{length of I, computed from the same data}}\right) = \frac{E(s(\hat{\gamma}))}{t_{m, 1-\alpha/2}}. \quad (4.15)$$

In some ways, this is a more natural definition of scaled expected length than (4.2), as it takes the ratio of the lengths of $\text{CI}(b, s)$ and I for the same data and then averages this ratio. For given function s , (4.15) is an even function of γ that we denote by $\text{SEL}_2(\gamma; s)$.

As shown in Appendix C.2.1, using the methods of Kabaila & Giri (2009a),

$$\text{SEL}_2(\gamma; s) = 1 + \frac{1}{t_{m, 1-\alpha/2}} \int_0^\infty \text{ISEL}(w, \gamma) w f_m(w) dw, \quad (4.16)$$

where $\text{ISEL}(w, \gamma)$ is defined by (4.13). We evaluate (4.16) as shown in Appendix C.2.3.

This second definition of the scaled expected length of $\text{CI}(b, s)$ leads to the second objective function

$$\text{OBJ}_2(s; \lambda) = \lambda \int_{-\infty}^\infty (\text{SEL}_2(\gamma; s) - 1) d\gamma + (\text{SEL}_2(0; s) - 1),$$

where λ is a positive tuning parameter that allows us to adjust the relative weight given to the uncertain prior information. As shown in Appendix

C.2.2, a formula for this objective function is given by

$$\frac{2}{t_{m,1-\alpha/2}} \int_0^\infty \int_0^d (s(x) - t_{m,1-\alpha/2}) (\lambda + \phi(wx)) dx w f_m(w) dw,$$

which can be simplified to yield

$$\text{OBJ}_2(s; \lambda) = \frac{2 E(W)}{t_{m,1-\alpha/2}} \int_0^d \text{IOBJ}_2(x; s, \lambda) dx, \quad (4.17)$$

where

$$\text{IOBJ}_2(x; s, \lambda) = (s(x) - t_{m,1-\alpha/2}) \left(\lambda + \frac{1}{\sqrt{2\pi}} \left(\frac{m}{x^2 + m} \right)^{(m+1)/2} \right).$$

This second objective function is used in the numerical nonlinear constrained optimization to compute the second confidence interval that utilizes the uncertain prior information. For given $\lambda > 0$, we minimize the objective function $\text{OBJ}_2(s; \lambda)$, with respect to the vector $(b(x_2), \dots, b(x_{q-1}), s(x_1), \dots, s(x_{q-1}))$, where the function s is constrained to take positive values, subject to the coverage constraint $\text{CP}(\gamma; b, s) \geq 1 - \alpha$ for all $\gamma \geq 0$. Denote the resulting confidence interval $\text{CI}(b_{2\lambda}, s_{2\lambda})$. Note that the term

$$\frac{1}{\sqrt{2\pi}} \left(\frac{m}{x^2 + m} \right)^{(m+1)/2}$$

in the objective function is a smooth function of x for large m , as shown in Appendix C.2.4. We compute the integral

$$\int_0^d \text{IOBJ}_2(x; s, \lambda) dx$$

similarly to the computation of $\text{ICP}(w, \gamma, \rho)$ in Section 4.3 using the Gauss Legendre quadrature.

4.5 Choice of some of the computational parameters of the new confidence interval and the method used for its computation

In this section, we choose the appropriate values for d and the length of the set Γ_{grid} of γ values, which is used in the implementation of the coverage constraint. Furthermore, we choose the number of intervals in $[0, d]$, where the endpoints of each of these intervals specify the knots of the cubic spline interpolations of $b_{1\lambda}$ and $s_{1\lambda}$, which we denote by `n.ints` in R programs. We also select the number of evaluations requires to compute the inner and outer integrands.

4.5.1 Choosing the value of d and the set Γ_{grid}

Based on the choices of d and Γ_{grid} that were made in the past, Paul Kabaila proposed formulas for the value of d and the set Γ_{grid} .

Let the quantile z_p be defined by $P(Z \leq z_p) = p$ where $Z \sim N(0, 1)$ and the quantile $t_{m,p}$ be defined by $P(T \leq t_{m,p}) = p$ where $T \sim t_m$. We chose p such that $z_p = 1.545$. Then we set $d = 6 t_{m,p}/z_p$. Since $t_{m,p}$ approaches z_p from above as $m \rightarrow \infty$, this formula leads to d approaching 6 from above as $m \rightarrow \infty$. The value of p was chosen so $d = 6 t_{m,p}/z_p$ results in values of d consistent with choices made in the past for $m = 1$ and $m = 2$. Figure 4.1 shows a graph of d as a function of m .

Now consider the choice of the set Γ_{grid} . Let γ_{max} denote the maximum possible value in the set Γ_{grid} . Set $\gamma_{\text{max}} = d + \text{extra}$, where $\text{extra} = t_{m,p'}$, for p' chosen so that $z_{p'} = 2$. Since $t_{m,p'}$ approaches $z_{p'} = 2$ from above, as $m \rightarrow \infty$, this formula leads to extra approaching 2 from above as $m \rightarrow \infty$. Figure 4.1 shows a graph of extra as a function of m .

Now the set Γ_{grid} is determined by the integer $l \in \{60, 120\}$ as follows. Divide the interval $[0, \gamma_{\text{max}}]$ into two intervals $[0, d]$ and $[d, \gamma_{\text{max}}]$. Set the length of the vector of equally-spaced values of Γ_{grid} in the interval $[0, d]$ to l , so that these values are $0, d/(l-1), 2d/(l-1), \dots, d$. Now set the equally-spaced values of Γ_{grid} in $[d, \gamma_{\text{max}}]$ to the sequence $d, d + 2(d/l), d + 4(d/l), \dots$, such that the maximum value of this sequence is less than or equal to γ_{max} .

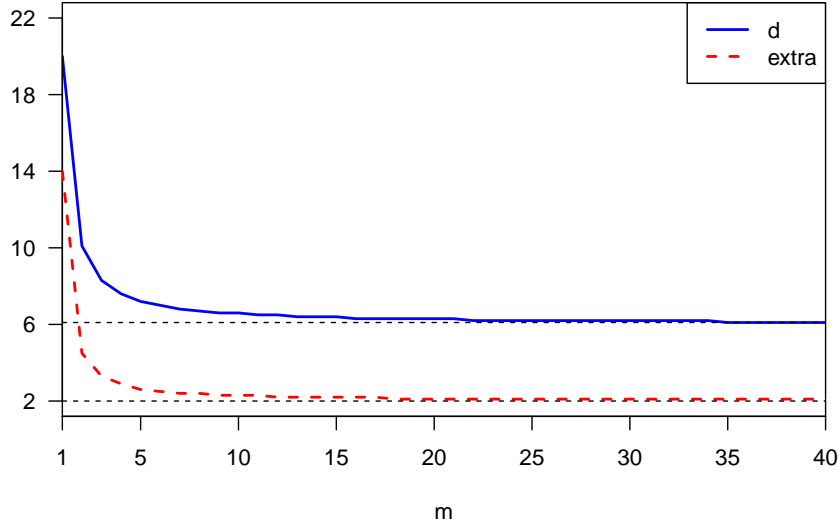


Figure 4.1: Graphs of d and **extra** as a functions of m .

4.5.2 Choosing the values of l and **n.ints**

We choose the values of l and **n.ints** by computing the confidence interval $\text{CI}(b_{1\lambda}, s_{1\lambda})$ for the numerical example given in [Kabaila & Giri \(2013\)](#). Note that, for this example, $m = 2$, $\alpha = 0.05$, $\rho = -0.5$ and $\lambda = 0.15$.

In Table 4.1, we compare the values of $(\text{SEL}_1(0; s_{1\lambda}))^2$, the maximum of $(\text{SEL}_1(\gamma; s_{1\lambda}))^2$ and the corresponding time needed to compute the vector $(b_{1\lambda}(x_2), \dots, b_{1\lambda}(x_{q-1}), s_{1\lambda}(x_1), \dots, s_{1\lambda}(x_{q-1}))$ for $l \in \{60, 120\}$ and **n.ints** $\in \{6, 7\}$. We also consider the two cases, d and $d(1 + (1/6))$ for this comparison.

Recall that N, N_1 and N_2 denote the number of outer integrand evaluations in the expressions for the coverage probability, first definition of the scaled expected length and the second definition of the scaled expected

length, respectively. We have chosen $N = N_1 = N_2$ for simplicity. Recall from Chapter 3 that ϵ is a desired upper bound on the absolute value of the approximation error. We use $N = 17$, Gauss Legendre quadrature with 10 nodes and $\epsilon = 10^{-7}$ for the computations in this subsection. We describe these choices, in detail, in subsection 4.5.3.

Table 4.1: The values of $(\text{SEL}_1(0; s_{1\lambda}))^2$, $(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda}))^2$ and the time needed to compute the vector $(b_{1\lambda}(x_2), \dots, b_{1\lambda}(x_{q-1}), s_{1\lambda}(x_1), \dots, s_{1\lambda}(x_{q-1}))$ for $m = 2$, $\alpha = 0.05$, $\rho = -0.5$ and $\lambda = 0.15$.

	$(\text{SEL}_1(0; s_{1\lambda}))^2$	Maximum of $(\text{SEL}_1(\gamma; s_{1\lambda}))^2$	Time (min)
$l = 120$			
(a) $d = 10.1$, n.ints =6	0.7882841	1.093538	21.40
(b) $d = 10.1$, n.ints =7	0.7885232	1.093049	27.55
(c) $d(1+(1/6)) = 11.8$, n.ints =7	0.7878176	1.093023	33.29
$l = 60$			
(d) $d = 10.1$, n.ints =6	0.7880789	1.093501	12.33
(e) $d = 10.1$, n.ints =7	0.7882304	1.093125	15.37
(f) $d(1+(1/6)) = 11.8$, n.ints =7	0.7875405	1.093098	17.90

Overall, the computation times for (a) - (c) for $l = 120$ are significantly larger than the corresponding computation times for (d) - (f) for $l = 60$. The values of $(\text{SEL}_1(0; s_{1\lambda}))^2$ and the maximum of $(\text{SEL}_1(\gamma; s_{1\lambda}))^2$, for (d), (e) and (f) are very similar to the corresponding values for (a), (b) and (c), respectively. This led us to choose $l = 60$, which gives more efficient results, for the computations carried out in the R package `ciuupi2`. The results in Appendix C.3 led us to choose **n.ints** = 6 for the computations carried out in the R package `ciuupi2`.

To summarize: We choose $l = 60$, `n.ints` = 6 and the value d for the computations carried out in the R package `ciupui2`.

4.5.3 Choosing the number of outer integrand evaluations, the value of ϵ and the number of Gauss Legendre quadrature nodes

We consider the two cases (a) $m > 2$ and (b) $m \in \{1, 2\}$. The results in Appendix C.4 led us to choose the following choices of N , the number of Gauss Legendre quadrature nodes and the value of ϵ .

- Computation of the vector $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$ for case (a) :
 $N = 9$, the Gauss Legendre quadrature with 5 nodes and $\epsilon = 10^{-7}$.
- Computation of the vector $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$ for case (b) :
 $N = 17$, the Gauss Legendre quadrature with 10 nodes and $\epsilon = 10^{-7}$.
- Computation of the graphs of the coverage probability and the squared scaled expected length for the two cases (a) and (b):
 $N = 33$, the Gauss Legendre quadrature with 20 nodes and $\epsilon = 10^{-10}$.

4.5.4 Comparison with some of the past results obtained by Kabaila and Giri

We consider the three examples, (a) [Kabaila & Giri \(2009a\)](#) for $m = 76$ and $\lambda = 0.2$, (b) [Kabaila & Giri \(2013\)](#) for $m = 2$ and $\lambda = 0.15$ and (c) [Giri \(2008\)](#) for $m = 1$ and $\lambda = 0.2$. We compute the plots of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length of

the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for these three examples in Appendix C.6, using the R package `ciuppi2`. We compare these results with some of the past results obtained by Kabaila and Giri as follows.

(a) Figure C.9 for $m = 76$ and $\lambda = 0.2$:

The plots of the functions $b_{1\lambda}$ and $s_{1\lambda}$ are almost identical to Figure 2 of Kabaila & Giri (2009a). The graphs of the coverage probability and the squared scaled expected length are also identical to the corresponding plots in Figure 2 of Kabaila & Giri (2009a). Note that in Figure C.9, $(\text{SEL}_1(0; s_{1\lambda}))^2 = 0.8679$ and $\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda})\right)^2 = 1.1057$. These values are very similar to the corresponding values in Section 4 of Kabaila & Giri (2009a). The time needed to compute the vector $(b_{1\lambda}(1), \dots, b_{1\lambda}(5), s_{1\lambda}(0), \dots, s_{1\lambda}(5))$ is 6.16 min.

(b) Figure C.10 for $m = 2$ and $\lambda = 0.15$:

The plots of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length are very similar to the corresponding plots in Kabaila & Giri (2013). Note that in Figure C.10, $(\text{SEL}_1(0; s_{1\lambda}))^2 = 0.7868$ and $\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda})\right)^2 = 1.0983$. These values are very similar to the corresponding values in Figure 1 of Kabaila & Giri (2013). The time needed to compute the vector $(b_{1\lambda}(2), \dots, b_{1\lambda}(12), s_{1\lambda}(0), \dots, s_{1\lambda}(12))$ is 46.21 min.

(c) Figure C.11 for $m = 1$ and $\lambda = 0.2$:

The plots of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length are approximately similar to the corresponding plots in Figure 3.8 and 3.9 in Giri (2008). Note that in Figure C.11, $(\text{SEL}_1(0; s_{1\lambda}))^2 = 0.7333$ and $\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda})\right)^2 = 1.0698$. These values are approximately similar to the corresponding results in Figure 3.9 of Giri (2008). The time needed to compute the

vector $(b_{1\lambda}(5), \dots, b_{1\lambda}(25), s_{1\lambda}(0), \dots, s_{1\lambda}(25))$ is 1.52 hr.

4.6 Computation of the [Kabaila & Giri \(2009a\)](#) confidence interval for the standard choice of λ .

In this section, we find a standard choice of that leads to what might be called a “standard” confidence interval that utilizes the uncertain prior information. The previous MATLAB programs of [Kabaila & Giri \(2009a\)](#) for a particular value of λ took so long to run and therefore they were not able to compute this confidence interval. However, with the new methods for computing the integrals that we implement in R, we are able to calculate this confidence interval very efficiently.

As proposed by [Kabaila & Giri \(2009a\)](#), we find the standard choice of λ for the first definition of the scaled expected length as follows. We choose λ such that the “gain” when the prior information is correct, as measured by

$$1 - (\text{SEL}_1(0; s_{1\lambda}))^2,$$

is equal to the maximum possible “loss” when the prior information happens to be incorrect, as measured by

$$\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda}) \right)^2 - 1.$$

We denote the resulting confidence interval by $\text{CI}(b_1, s_1)$.

Similarly, we can find the standard choice of λ for the second definition of the scaled expected length by choosing the value of λ such that

$$1 - (\text{SEL}_2(0; s_{2\lambda}))^2 = \left(\max_{\gamma \geq 0} \text{SEL}_2(\gamma; s_{2\lambda}) \right)^2 - 1.$$

We denote this resulting confidence interval by $\text{CI}(b_2, s_2)$.

We consider the confidence interval $\text{CI}(b_1, s_1)$ for the numerical evalua-

tions in this section. We choose the value d and the set Γ_{grid} as described in subsection 4.5.1. According to subsection 4.5.2, $\mathbf{n.ints} = 6$ and $l = 60$. We choose N , the number of Gauss Legendre quadrature nodes and the value of ϵ as described in subsection 4.5.3. We consider the two cases (a) $m > 2$ and (b) $m \in \{1, 2\}$.

4.6.1 Standard choice of λ for $m > 2$

We consider the numerical example in Kabaila & Giri (2009a). Recall that $m = 76, \alpha = 0.05, \rho = -0.707$ and $\lambda = 0.2$. The standard choice of λ is found to be 0.104. The time needed to compute the vector $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$ is 24.13 min. The times needed to compute the graphs of the coverage probability and the squared scaled expected length are 2.42 secs and 0.57 secs, respectively.

Table 4.2 shows that for the standard choice of λ ($\lambda = 0.104$), there is a higher gain than the $\lambda = 0.2$ and the ratio of gain to maximum possible loss approaches 1. The left panels of Figures 4.2 and 4.3 show that the plots of the functions b_1, s_1 , the coverage probability and the squared scaled expected length have the desired properties for this standard choice of λ .

Table 4.2: Performance of the confidence interval $\text{CI}(b_1, s_1)$ for $m = 76, \alpha = 0.05, \rho = -0.707, d = 6.1, \mathbf{n.ints} = 6$ and $l = 60$ for $\lambda \in \{0.2, 0.105\}$.

λ	0.2	0.104
Gain	0.1321	0.1666
Maximum possible loss	0.1056	0.1665
(Gain)/(Maximum possible loss)	1.251	1.0008

4.6.2 Standard choice of λ for $m \in \{1, 2\}$

We consider the numerical example in [Kabaila & Giri \(2013\)](#). Recall that $m = 2, \alpha = 0.05, \rho = -0.5$ and $\lambda = 0.15$. The standard choice of λ is found to be 0.023. The time needed to compute the vector $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$ is 1.97 hr. The times needed to compute the graphs of the coverage probability and the squared scaled expected length are 4.86 secs and 1.16 secs respectively.

Table 4.3 shows that for the standard choice of λ ($\lambda = 0.023$), there is a higher gain than the $\lambda = 0.15$ and the ratio of gain to maximum possible loss approaches 1. The left panels of Figures 4.4 and 4.5 show that the plots of the functions b_1, s_1 , the coverage probability and the squared scaled expected length have the desired properties for this standard choice of λ .

Table 4.3: Performance of the confidence interval $\text{CI}(b_1, s_1)$ for $m = 2, \alpha = 0.05, \rho = -0.5, d = 10.1, \text{n.ints} = 6$ and $l = 60$ for $\lambda \in \{0.15, 0.023\}$.

λ	0.15	0.023
Gain	0.2119	0.3866
Maximum possible loss	0.0935	0.3866
(Gain)/(Maximum possible loss)	2.2669	1.0000

4.7 Comparison of the numerical results for the two definitions of scaled expected length for the standard choice of λ

In this section, we numerically compare the two confidence intervals $\text{CI}(b_1, s_1)$ and $\text{CI}(b_2, s_2)$ for the standard choice of λ in terms of the plots of the functions b_1 and s_1 , the coverage probability and the squared scaled expected length.

Note that when m is getting large, the two objective functions $\text{OBJ}_1(\gamma; s_1)$ and $\text{OBJ}_2(\gamma; s_2)$ become similar.

We choose the value d and the set Γ_{grid} as described in subsection 4.5.1. According to subsection 4.5.2, $\text{n.ints} = 6$ and $l = 60$. We choose N , the number of Gauss Legendre quadrature nodes and the value of ϵ as described in subsection 4.5.3. We consider the two cases (a) $m > 2$ and (b) $m \in \{1, 2\}$.

4.7.1 Comparison for $m > 2$

We consider the numerical example in Kabaila & Giri (2009a) where $m = 76$, $\alpha = 0.05$ and $\rho = -0.707$. The standard choice of λ is equal to 0.104 for the both confidence intervals $\text{CI}(b_1, s_1)$ and $\text{CI}(b_2, s_2)$.

Figure 4.2 shows that the plots of the functions b_1, s_1 (left panel) and b_2, s_2 (right panel) are identical. Figure 4.3 shows that graphs of the coverage probability and the squared scaled expected length of the confidence intervals $\text{CI}(b_1, s_1)$ (left panel) and $\text{CI}(b_2, s_2)$ (right panel) are also identical.

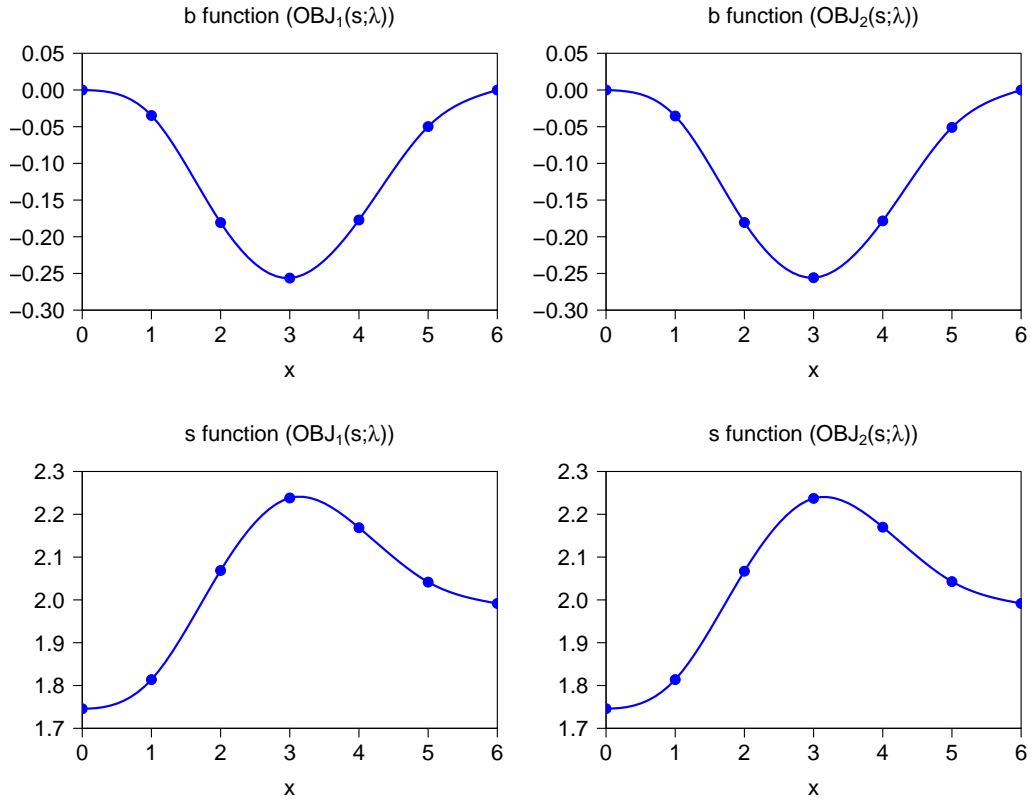


Figure 4.2: Graphs of the functions b_1 and s_1 (left panel) and b_2 and s_2 (right panel) for $m = 76$, $\alpha = 0.05$, $\rho = -0.707$, $d = 6.1$, `n.ints` = 6 and $l = 60$. The standard choice of λ is equal to 0.104 for the both confidence intervals $CI(b_1, s_1)$ and $CI(b_2, s_2)$.

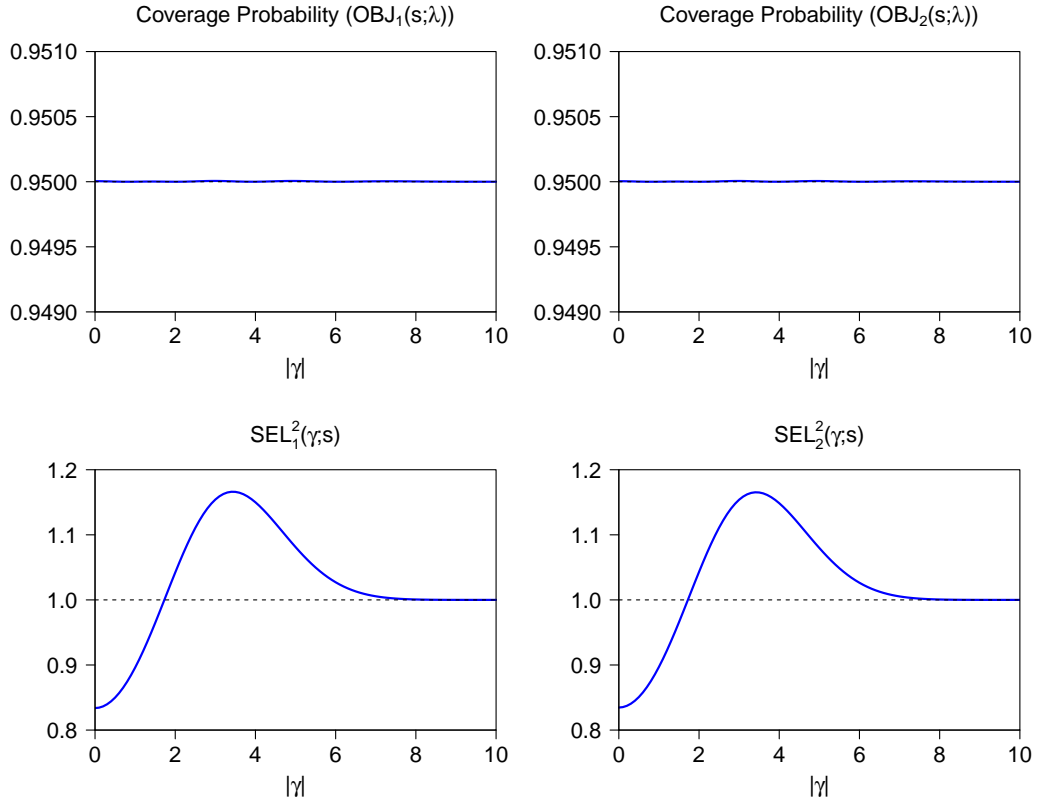


Figure 4.3: Graphs of the coverage probability and the squared scaled expected length of the confidence intervals $CI(b_1, s_1)$ (left panel) and $CI(b_2, s_2)$ (right panel) for $m = 76$, $\alpha = 0.05$, $\rho = -0.707$, $d = 6.1$, $\mathbf{n.ints} = 6$ and $l = 60$. The standard choice of λ is equal to 0.104 for the both confidence intervals $CI(b_1, s_1)$ and $CI(b_2, s_2)$.

4.7.2 Comparison for $m \in \{1, 2\}$

We consider the numerical example in [Kabaila & Giri \(2013\)](#) where $m = 2$, $\alpha = 0.05$, $\rho = -0.5$. The standard choice of λs is equal to 0.023 and 0.020 for the confidence intervals $CI(b_1, s_1)$ and $CI(b_2, s_2)$, respectively.

Figure 4.4 shows that the plots of the functions b_1, s_1 (left panel) and b_2, s_2 (right panel) are not identical but have somewhat similar shapes. Figure 4.5 shows that the graphs of the two coverage probabilities are very similar. There is a moderate difference between the two graphs of the squared scaled

expected length. The values of $(\text{SEL}_1(0; s_1))^2$ and $\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_1)\right)^2$ are comparatively larger than the corresponding values of $(\text{SEL}_2(0; s_2))^2$ and $\left(\max_{\gamma \geq 0} \text{SEL}_2(\gamma; s_2)\right)^2$.

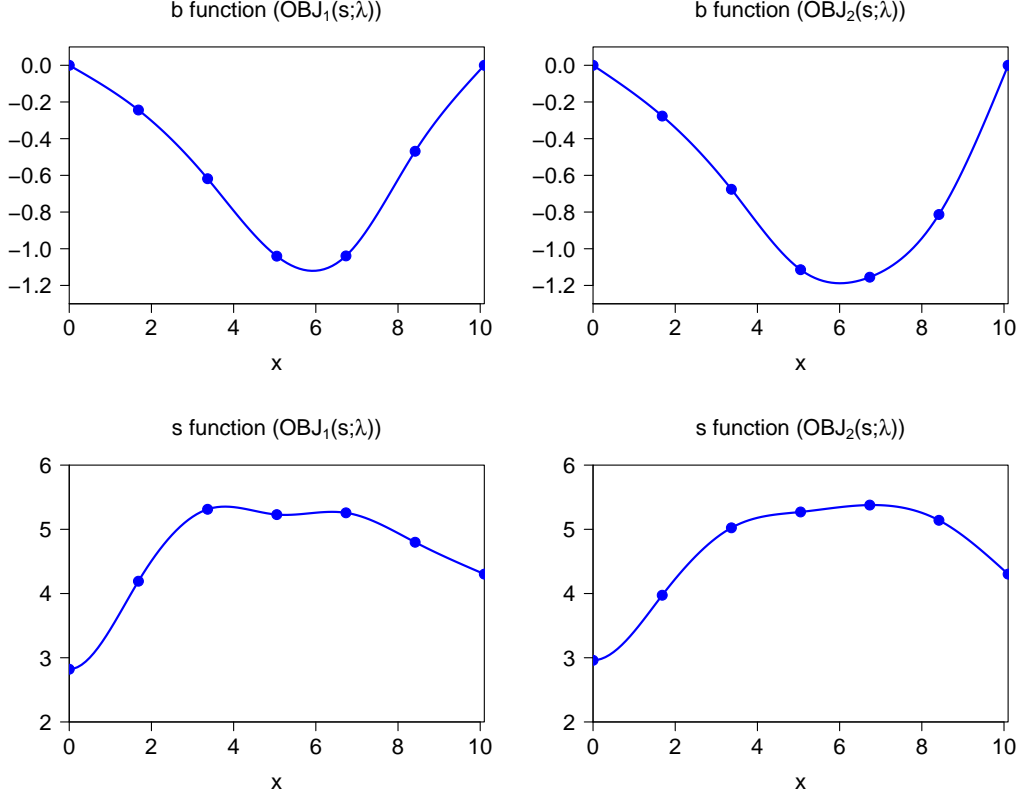


Figure 4.4: Graphs of the functions b_1 and s_1 (left panel) and b_2 and s_2 (right panel) for $m = 2$, $\alpha = 0.05$, $\rho = -0.5$, $d = 10.1$, $\mathbf{n.ints} = 6$ and $l = 60$. The standard choice of λ s is equal to 0.023 and 0.020 for the confidence intervals $\text{CI}(b_1, s_1)$ and $\text{CI}(b_2, s_2)$, respectively.

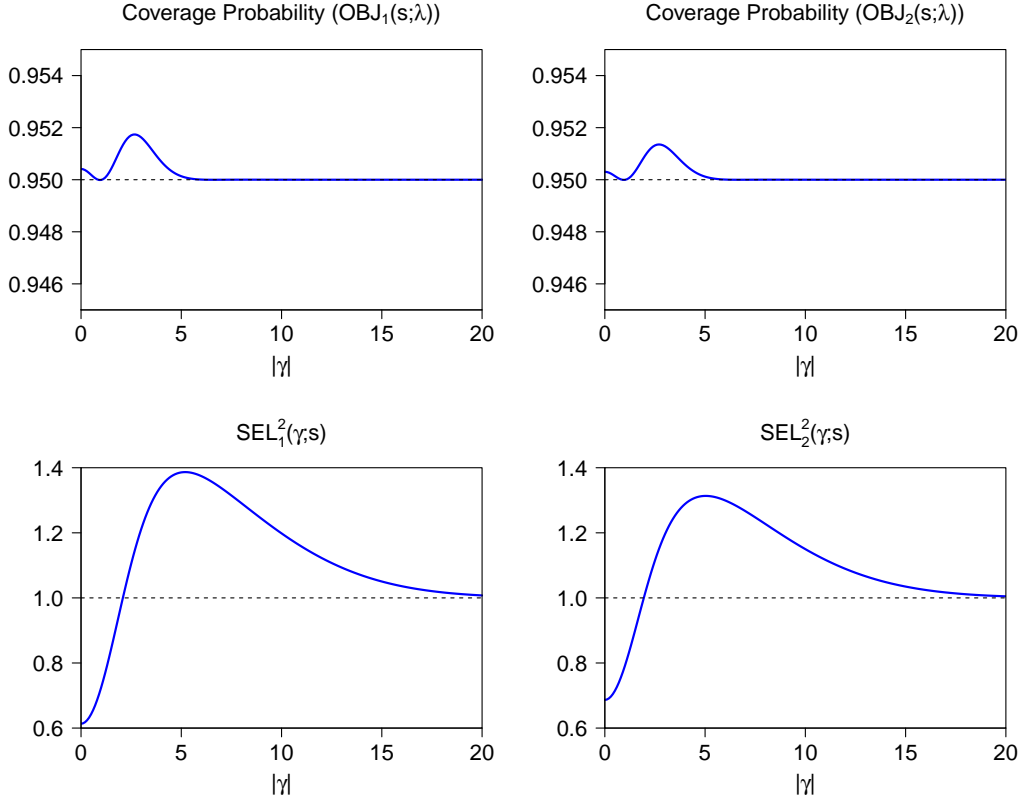


Figure 4.5: Graphs of the coverage probability and the squared scaled expected length of the confidence intervals $CI(b_1, s_1)$ (left panel) and $CI(b_2, s_2)$ (right panel) for $m = 2$, $\alpha = 0.05$, $\rho = -0.5$, $d = 10.1$, `n.ints` = 6 and $l = 60$. The standard choice of λ s is equal to 0.023 and 0.020 for the confidence intervals $CI(b_1, s_1)$ and $CI(b_2, s_2)$, respectively.

4.8 Conclusion

We have created the R package `ciuupi2` to carry out the computations of the [Kabaila & Giri \(2009a\)](#) confidence interval, which utilizes the uncertain prior information in a linear regression model with unknown normal error variance. Apart from the applications to the linear regression, `ciuupi2` can also be used as an approximate approach to the non-linear regression models with unknown random error variance that utilizes uncertain prior information. In

this case, one could use a similar approach, as described in Chapter 2, to find the frequentist confidence intervals.

The development of the `ciupui2` package involved major computational and theoretical innovations that led to computing the [Kabaila & Giri \(2009a\)](#) confidence interval in significantly less time than the past implementations carried out in `MATLAB`. The results obtained from `ciupui2` were compared with some of the past results obtained by Kabaila and Giri and were found to be very similar. Furthermore, the two definitions of the scaled expected length showed identical results for larger values of m and there were some moderate differences for smaller values of m .

Chapter 5

On adaptive Gauss-Hermite quadrature for estimation in generalized linear mixed models

5.1 Introduction

As we have seen from the work in Chapter 3, the efficient numerical evaluation of integrals requires a careful analysis. In the present chapter, we explore the numerical evaluation of integrals in the context of the computation of the log-likelihood function for generalized linear mixed models.

Adaptive Gauss-Hermite quadrature can be used for the computation of the log-likelihood function for generalized linear mixed models. For Gauss-Hermite quadrature, [Liu & Pierce \(1994\)](#) proposed a method to transform the variable of integration in such a way that the integrand is sampled at relatively important values. [Pinheiro & Bates \(1995\)](#) referred to this method put forward by [Liu & Pierce \(1994\)](#) as ‘adaptive’ Gauss-Hermite quadrature. This method has found applications in the computation of the log-likelihood function for generalized linear mixed models ([Lesaffre & Spiessens, 2001](#), [Demidenko, 2004](#), [Hedeker & Gibbons, 2006](#), [Tuerlinckx *et al.*, 2006](#), [Rabe-](#)

Hesketh & Skrondal, 2008, Kim *et al.* , 2013 and Chang & Hoaglin, 2017).

The basic first step in this method is to multiply and divide the integrand of interest by a carefully chosen probability density function. The same first step is used for the computation of this log-likelihood function using simulations that employ importance sampling. As is well-known, importance sampling needs to be applied with extreme care to be successful (Robert & Casella, 2004 and Owen, 2013). We compare these two methods by considering in detail a single cluster from a well-known teratology data set that is modelled using a logistic regression with random intercept.

In Section 5.2, we describe this model and its log-likelihood function. In Section 5.3, we describe the adaptive Gauss-Hermite quadrature approach for the computation of the log-likelihood function for this model. In Section 5.4, we introduce the teratology data set of Weil (1970). We also assess the performance of importance sampling using a simulation method. We show that while importance sampling fails for this computation, adaptive Gauss-Hermite quadrature does not. In Section 5.5, we derive a new upper bound on the error of approximation of adaptive Gauss-Hermite quadrature. Using this new upper bound, we show that the feature of this problem that makes importance sampling fail is useful in disclosing why adaptive Gauss-Hermite quadrature succeeds. The work described in this chapter appears in Kabaila & Ranathunga (2019).

5.2 The logistic regression with random intercept model and its log-likelihood function

We consider a logistic regression model with random intercept. Let y_i and x_i denote the response and covariate, respectively, for the i 'th cluster ($i = 1, \dots, N$). Let $\boldsymbol{\eta} = (\eta_1, \dots, \eta_N)$, where the η_i 's are independent and identically $N(0, \sigma^2)$ distributed. Also let $\mathbf{v} = (v_1, \dots, v_N)$. Suppose that, conditional on $\boldsymbol{\eta} = \mathbf{v}$, the y_i 's are independent and $y_i \sim \text{Binomial}(J_i, \pi_i)$, where J_i denotes the size of the i 'th cluster and

$$\log \left(\frac{\pi_i}{1 - \pi_i} \right) = \beta_1 + \beta_2 x_i + v_i$$

for $i = 1, \dots, N$. Let $\hat{\beta}_1$, $\hat{\beta}_2$ and $\hat{\sigma}$ denote the maximum likelihood estimates obtained from all of the data. Let $\mathbf{y} = (y_1, \dots, y_N)$. The probability mass function (pmf) of \mathbf{y} , conditional on $\boldsymbol{\eta} = \mathbf{v}$, is

$$\prod_{i=1}^N \binom{J_i}{y_i} \pi_i^{y_i} (1 - \pi_i)^{(J_i - y_i)} \quad (5.1)$$

where

$$\pi_i = \frac{\exp(\beta_1 + \beta_2 x_i + v_i)}{1 + \exp(\beta_1 + \beta_2 x_i + v_i)}$$

and so

$$\begin{aligned} (5.1) &= \prod_{i=1}^N \binom{J_i}{y_i} \left(\frac{\exp(\beta_1 + \beta_2 x_i + v_i)}{1 + \exp(\beta_1 + \beta_2 x_i + v_i)} \right)^{y_i} \left(\frac{1}{1 + \exp(\beta_1 + \beta_2 x_i + v_i)} \right)^{(J_i - y_i)} \\ &= \prod_{i=1}^N \binom{J_i}{y_i} \frac{\exp((\beta_1 + \beta_2 x_i + v_i)y_i)}{(1 + \exp(\beta_1 + \beta_2 x_i + v_i))^{J_i}}. \end{aligned}$$

Let $\phi(t; \mu, \sigma^2)$ denote the $N(\mu, \sigma^2)$ pdf, evaluated at t . Let \mathbf{t} denote the

vector of variables (t_1, \dots, t_N) . Now, the pmf of $\mathbf{y} = (y_1, \dots, y_N)$ is the N -fold multiple integral

$$\begin{aligned} & \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \prod_{i=1}^N \frac{\exp((\beta_1 + \beta_2 x_i + t_i)y_i)}{(1 + \exp(\beta_1 + \beta_2 x_i + t_i))^{J_i}} \phi(t_1; 0, \sigma^2) \cdots \phi(t_N; 0, \sigma^2) dt_1 \cdots dt_N \\ &= \prod_{i=1}^N \int_{-\infty}^{\infty} \frac{\exp((\beta_1 + \beta_2 x_i + t_i)y_i)}{(1 + \exp(\beta_1 + \beta_2 x_i + t_i))^{J_i}} \phi(t_i; 0, \sigma^2) dt_i \end{aligned}$$

The additive contribution of a given cluster of size J to the log-likelihood function is, to within an additive constant, the logarithm of

$$\int_{-\infty}^{\infty} \frac{\exp[(\beta_1 + \beta_2 x + t)y]}{[1 + \exp(\beta_1 + \beta_2 x + t)]^J} \phi(t; 0, \sigma^2) dt, \quad (5.2)$$

where y and x denote the observed response and covariate, respectively, for this cluster. Obviously, (5.2) is equal to $c(\theta, \sigma)$ which we define to be

$$\int_{-\infty}^{\infty} g(t; \theta, \sigma) \phi(t; 0, \sigma^2) dt, \quad (5.3)$$

where

$$g(t; \theta, \sigma) = \frac{\exp[(\theta + t)y]}{[1 + \exp(\theta + t)]^J},$$

with $\theta = \beta_1 + \beta_2 x$.

5.3 Adaptive Gauss-Hermite quadrature

We now describe the adaptive Gauss-Hermite quadrature method for the computation of $c(\theta, \sigma)$. Using the Theorem 5.1 on p.142 of [Carlin & Louis \(1996\)](#), we approximate $g(t; \theta, \sigma) \phi(t; 0, \sigma^2)$ by $\phi(t; \mu, \tau^2)$, where

$$\mu = \text{mode of } g(t; \theta, \sigma) \phi(t; 0, \sigma^2), \text{ considered as a function of } t$$

and

$$\tau^2 = \left[-\frac{\partial^2}{\partial t^2} (\log [g(t; \theta, \sigma) \phi(t; 0, \sigma^2)]) \right]^{-1}.$$

These are derived in Appendix [D.1](#).

This method is based on employing $\phi(t; \mu, \tau^2)$ as an approximation to $g(t; \theta, \sigma) \phi(t; 0, \sigma^2)$. The first basic step in the description of adaptive Gauss-Hermite quadrature is to write

$$c(\theta, \sigma) = \int_{-\infty}^{\infty} g(t; \theta, \sigma) \phi(t; 0, \sigma^2) dt = \int_{-\infty}^{\infty} \frac{g(t; \theta, \sigma) \phi(t; 0, \sigma^2)}{\phi(t; \mu, \tau^2)} \phi(t; \mu, \tau^2) dt. \quad (5.4)$$

This step of multiplying and dividing the integrand by a pdf is common to both importance sampling and adaptive Gauss-Hermite quadrature. We re-express [\(5.4\)](#) as

$$c(\theta, \sigma) = \int_{-\infty}^{\infty} h(t; \theta, \sigma) \phi(t; \mu, \tau^2) dt, \quad (5.5)$$

where

$$h(t; \theta, \sigma) = \frac{g(t; \theta, \sigma) \phi(t; 0, \sigma^2)}{\phi(t; \mu, \tau^2)}.$$

We now change the variable of integration in [\(5.5\)](#) to $z = (t - \mu)/(\sqrt{2}\tau)$ so that $t = \mu + \sqrt{2}\tau z$. Thus, as shown in Appendix [D.2](#),

$$c(\theta, \sigma) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} h(\mu + \sqrt{2}\tau z; \theta, \sigma) \exp(-z^2) dz. \quad (5.6)$$

Let

$$\sum_{i=1}^m f(z_i) w_i$$

be the m -node Gauss-Hermite quadrature approximation to

$$\int_{-\infty}^{\infty} f(z) \exp(-z^2) dz.$$

Therefore, the m -node adaptive Gauss-Hermite quadrature approximation to $c(\theta, \sigma)$ is

$$c_m(\theta, \sigma) = \frac{1}{\sqrt{\pi}} \sum_{i=1}^m h(\mu + \sqrt{2}\tau z_i; \theta, \sigma) w_i, \quad (5.7)$$

where z_i and w_i are the Gauss-Hermite quadrature nodes and weights, respectively.

5.4 The teratology data and importance sampling

In this section, we compare the adaptive Gauss-Hermite quadrature approximation (5.7) with importance sampling, using the importance pdf $\phi(t; \mu, \tau^2)$ in (5.5), in the particular context of the teratology data described by [Weil \(1970\)](#).

5.4.1 Description of the dataset and the maximum likelihood estimates

The teratology data set of [Weil \(1970\)](#) lists the number of rat pups in 16 control litters that survived and the number of rat pups in 16 treated litters that survived. Each litter is treated as a cluster, so that the total number of clusters $N = 32$. The covariate x_i takes the value 1 for a litter i that is treated and the value 0 for a litter i that is a control. For this dataset, J_i and y_i denote the number of pups and the number of surviving pups, respectively, in litter i . This data is shown in Table 5.1. The litters are numbered from 1 up to 32 with the litter in row j and column k allocated the number $i = 8(j - 1) + k$.

Table 5.1: Teratology data set of [Weil \(1970\)](#). This data lists the number of rat pups in 16 control litters that survived and the the number of rat pups in 16 treated litters that survived.

	(number survived, number dead)							
Control	(13, 0)	(12, 0)	(9, 0)	(9, 0)	(8, 0)	(8, 0)	(12, 1)	(11, 1)
	(9, 1)	(9, 1)	(8, 1)	(11, 2)	(4, 1)	(5, 2)	(7, 3)	(7, 3)
Treatment	(12, 0)	(11, 0)	(10, 0)	(9, 0)	(10, 1)	(9, 1)	(9, 1)	(8, 1)
	(8, 1)	(4, 1)	(7, 2)	(4, 3)	(5, 5)	(3, 3)	(3, 7)	(0, 7)

5.4.2 The performance of importance sampling

We describe a simulation method that employs importance sampling for the computation of (5.7), using the importance pdf, $\phi(t; \mu, \tau^2)$, in (5.5). We suppose that this simulation consists of M independent simulation runs. Let v_i denote the observation obtained in the i 'th simulation run of a random variable with pdf $\phi(t; \mu, \tau^2)$. The importance sampling estimator of $c(\theta, \sigma)$ is

$$\tilde{c}_M(\theta, \sigma) = \frac{1}{M} \sum_{i=1}^M h(v_i; \theta, \sigma).$$

This is an unbiased estimator of $c(\theta, \sigma)$ and its variance is $\tilde{\sigma}^2/M$, where

$$\tilde{\sigma}^2 = \int_{-\infty}^{\infty} \frac{[g(t; \theta, \sigma) \phi(t; 0, \sigma^2)]^2}{\phi(t; \mu, \tau^2)} dt - c^2(\theta, \sigma),$$

see e.g. [Owen \(2013\)](#). Hence

$$\tilde{\sigma}^2 = \int_{-\infty}^{\infty} r^2(t; \theta, \sigma) dt - c^2(\theta, \sigma),$$

where

$$r(t; \theta, \sigma) = h(t; \theta, \sigma) [\phi(t; \mu, \tau^2)]^{1/2}. \quad (5.8)$$

As shown in Appendix D.3,

$$r(t; \theta, \sigma) = c_0 \frac{\exp(c_1 + c_2 t + c_3 t^2)}{[1 + \exp(\theta + t)]^J},$$

where $c_0 = (\tau/\sqrt{2\pi\sigma^2})^{1/2}$, $c_1 = \theta y + \mu^2/(4\tau^2)$, $c_2 = y - \mu/(2\tau^2)$, $c_3 = 1/(4\tau^2) - 1/(2\sigma^2)$.

It follows from the definition of τ and Appendix D.4 that $c_3 > -1/(4\sigma^2)$. If $c_3 > 0$ then $r(t; \theta, \sigma) \rightarrow \infty$ as $t \rightarrow \infty$ and as $t \rightarrow -\infty$. Thus, if $c_3 > 0$ then $\tilde{\sigma}^2 = \infty$ and importance sampling fails spectacularly. If, however, $c_3 < 0$ then (a) $r(t; \theta, \sigma) \rightarrow 0$ as $t \rightarrow \infty$ and as $t \rightarrow -\infty$ and (b) $\tilde{\sigma}^2$ is finite. Of course, even if $c_3 < 0$, importance sampling may still fail to improve on simple Monte Carlo simulation.

Consider the particular values $\beta_1 = 2.6$, $\beta_2 = -1.1$ and $\sigma = 1.3$. These are examples of values that might be encountered during the computation of the maximum likelihood estimates, which are $\hat{\beta}_1 = 2.6257$, $\hat{\beta}_2 = -1.0824$ and $\hat{\sigma} = 1.3457$. We obtained these values by minimizing (D.6) in Appendix D.5 using the R function `nlminb`. For the particular values of β_1 , β_2 and σ that we have chosen, $\theta = 2.6$ for each litter in the control group and $\theta = 1.5$ for each litter in the treatment group.

For litters 7-12, 14-19 and 21-32, we find that $c_3 > 0$, so that $\tilde{\sigma}^2 = \infty$ and importance sampling fails spectacularly for these litter numbers. Figure 5.1 presents graphs of $\log r(t; \theta, \sigma)$, considered as a function of t , for control litters 12 and 15 and treatment litters 29 and 32. These graphs confirm that $r(t; \theta, \sigma) \rightarrow \infty$ as $t \rightarrow \infty$ and as $t \rightarrow -\infty$. In other words, these graphs confirm that $\tilde{\sigma}^2 = \infty$ for these litters.

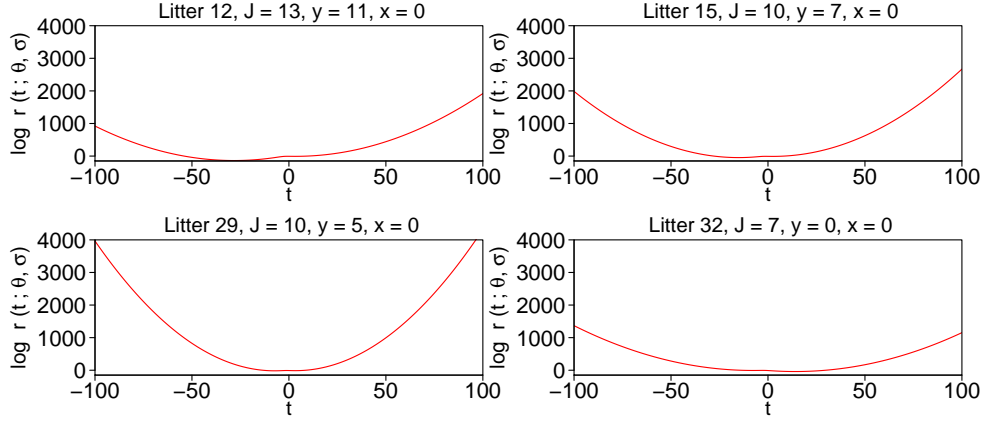


Figure 5.1: Graphs of $\log r(t; \theta, \sigma)$ as a function of t for $\sigma = 1.3$, $\theta = 2.6$ for a control litter and $\theta = 1.5$ for a treatment litter. The top two graphs are for control litters 12 and 15. The bottom two graphs are for treatment litters 29 and 32.

5.5 The performance of adaptive Gauss-Hermite quadrature for cluster 29 of the teratology data

The following theorem, which uses a well-known type of argument for bounding the error of Gauss-Legendre quadrature, see e.g. [Atkinson \(1989\)](#) (pp.277–278), describes a new upper bound on $|c(\theta, \sigma) - c_m(\theta, \sigma)|$. This theorem and its proof, given in Appendix [D.6](#), are due to Paul Kabaila.

Theorem 5.5.1. *Suppose that m , z_ℓ and z_u , where $z_\ell \leq z_1$ and $z_u \geq z_m$, are given. Also suppose that J , y , θ and σ are given. For notational convenience let $k(z) = h(\mu + \sqrt{2}\tau z; \theta, \sigma)$. Define $q_m(z)$ to be the polynomial $p_m(z)$ of degree $2m - 1$ that minimizes (either exactly or numerically)*

$$\max_{z \in [z_\ell, z_u]} |k(z) - p_m(z)|.$$

Let e_m denote this minimized value. Then

$$|c(\theta, \sigma) - c_m(\theta, \sigma)| \leq 2e_m + |a_\ell| + |a_u|, \quad (5.9)$$

where

$$a_\ell = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{z_\ell} (k(z) - q_m(z)) \exp(-z^2) dz$$

and

$$a_u = \frac{1}{\sqrt{\pi}} \int_{z_u}^{\infty} (k(z) - q_m(z)) \exp(-z^2) dz.$$

Furthermore:

If $k(z) \geq q_m(z) \geq 0$ for all $z \leq z_\ell$ then

$$0 \leq a_\ell \leq \int_{-\infty}^{\mu + \sqrt{2}\tau z_\ell} g(t; \theta, \sigma) \phi(t; 0, \sigma^2) dt. \quad (5.10)$$

and

If $k(z) \geq q_m(z) \geq 0$ for all $z \geq z_u$ then

$$0 \leq a_u \leq \int_{\mu + \sqrt{2}\tau z_u}^{\infty} g(t; \theta, \sigma) \phi(t; 0, \sigma^2) dt. \quad (5.11)$$

This theorem shows that if $k(z) \geq q_m(z)$ for all $z \leq z_\ell = z_1$ and for all $z \geq z_u = z_m$ for every m in an increasing sequence of values of m , then $|a_\ell| + |a_u| \rightarrow 0$ as m increases through these values, since $z_m \rightarrow \infty$ as $m \rightarrow \infty$ (see e.g. Szegö (1967)).

Figure 5.2 presents a graph of $k(z) - q_m(z) = h(\mu + \sqrt{2}\tau z; \theta, \sigma) - q_m(z)$, considered as a function of z , for litter 29 for adaptive Gauss-Hermite quadrature with $m = 5$ nodes. The smallest and largest nodes z_1 and z_m , respectively, are shown. This graph shows that, in this case, $k(z) \geq q_m(z)$ for all $z \leq z_\ell = z_1$ and for all $z \geq z_u = z_m$.

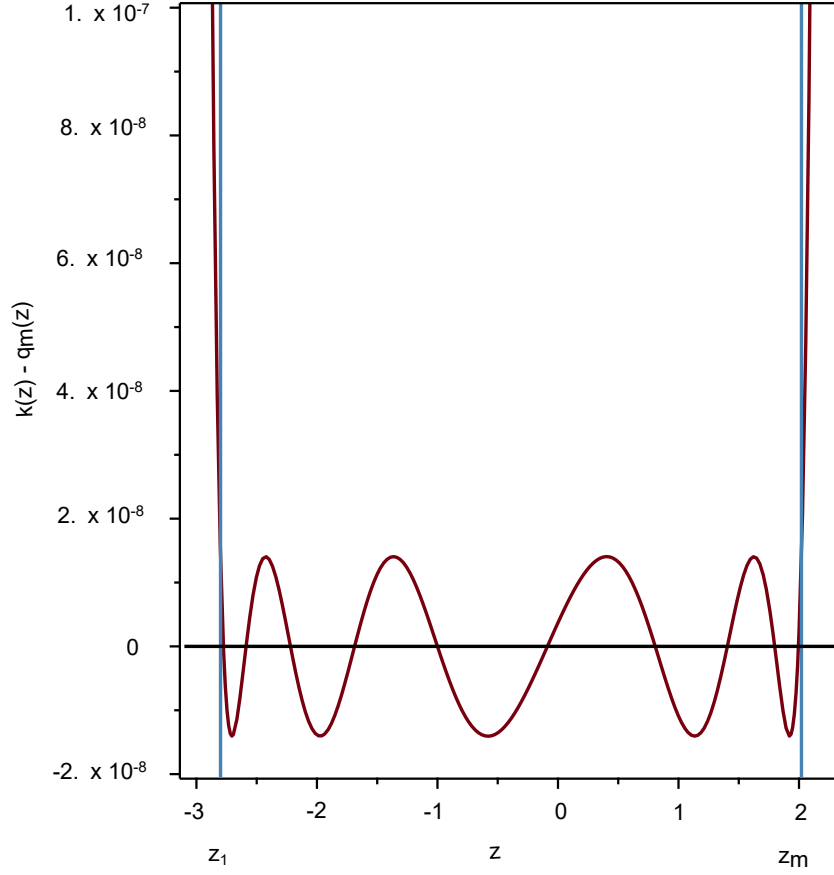


Figure 5.2: Graph of the function $k(z) - q_m(z) = h(\mu + \sqrt{2}\tau z; \theta, \sigma) - q_m(z)$, considered as a function of z , for litter 29 for adaptive Gauss-Hermite quadrature with $m = 5$ nodes. The smallest and largest nodes z_1 and z_m , respectively, are shown.

Table 5.2 presents the values of $10^8 \times e_m$ for cluster 29, where z_1 and z_m are the smallest and largest nodes for Gaussian-Hermite with m nodes, where $m = 3, 5, 7, 9$ and 11 . Here z_ℓ and z_u are the lower and upper limits such that $z_\ell \leq z_1$ and $z_u \geq z_m$. For every value of m in this table, $k(z) \geq q_m(z) \geq 0$ for all $z \leq z_\ell$ and for all $z \geq z_u$. In other words, the upper bound on $|c(\theta, \sigma) - c_m(\theta, \sigma)|$ that results from (5.9), (5.10) and (5.11) applies. Note that the values of e_m in this table decrease as m increases. Also, since z_ℓ decreases and z_u increases as m increases, $|a_\ell|$ and $|a_u|$ decrease as m increases through

the values $m = 3, 5, 7, 9$ and 11 . Our conclusion from this table is that the upper bound on $|c(\theta, \sigma) - c_m(\theta, \sigma)|$ decreases as m increases through these values. Figure 5.2 and Table 5.2 were obtained using the `minimax` command in `Maple`.

Table 5.2: Values of $10^8 \times e_m$ for cluster 29, where z_1 and z_m are the smallest and largest nodes for Gaussian-Hermite with m nodes. Here z_ℓ and z_u are the lower and upper limits such that $z_\ell \leq z_1$ and $z_u \geq z_m$.

m	z_ℓ	z_1	z_m	z_u	$10^8 \times e_m$
3	-2.100000000	-1.224744871	1.224744871	1.280000000	3.9453
5	-2.800000000	-2.020182870	2.020182870	2.020182870	1.4063
7	-3.400000000	-2.651961357	2.651961357	2.651961357	0.79274
9	-3.930000000	-3.190993202	3.190993202	3.190993202	0.51983
11	-4.400000000	-3.668470847	3.668470847	3.668470847	0.39768

5.6 Remark

As noted in section 5.4.2, if $c_3 > 0$ then $r(t; \theta, \sigma) \rightarrow \infty$ as $t \rightarrow \infty$ and as $t \rightarrow -\infty$, so that $\tilde{\sigma}^2 = \infty$ and importance sampling fails spectacularly. However, (5.8) implies that if $c_3 > 0$ then $h(t; \theta, \sigma) \rightarrow \infty$, as $t \rightarrow \infty$ and as $t \rightarrow -\infty$, faster than any polynomial. In other words, if $c_3 > 0$ then $k(z) = h(\mu + \sqrt{2}\tau z; \theta, \sigma) \rightarrow \infty$, as $z \rightarrow \infty$ and as $z \rightarrow -\infty$, faster than any polynomial. This is shown in Appendix D.7. It is this fact that makes it possible for the condition that $k(z) \geq q_m(z)$ for all $z \leq z_\ell$ and for all $z \geq z_u$ to be satisfied. This condition is used in Theorem 5.5.1 to bound $|a_\ell|$ and $|a_u|$ from above. In other words, the property of the function $h(t; \theta, \sigma)$ of t

that leads to importance sampling failing spectacularly is useful in disclosing why adaptive Gauss-Hermite quadrature succeeds.

5.7 Conclusion

Adaptive Gauss-Hermite quadrature and importance sampling share the same basic first step of multiplying and dividing the integrand of interest by a chosen pdf. However, this is where the similarity between these two methods ends. Extreme care is required to apply importance sampling effectively. Fortunately, for users of adaptive Gauss-Hermite quadrature for the computation of the log-likelihood function of generalized linear mixed models, this method can be applied effectively without approaching the same level of concern.

Appendix A

Proofs and R programs for Chapter 2

A.1 Derivation of the expressions for $r_1(\theta')$ and r_2 given in subsection 2.2.1

It follows from the joint distribution of $(\hat{\theta}, \hat{\tau})$, given by (2.6), that the joint pdf of $(\hat{\theta}, \hat{\tau})$ is

$$\frac{1}{2\pi v_\theta^{1/2} v_\tau^{1/2} \sqrt{1-\rho^2}} \times \exp \left(-\frac{1}{2(1-\rho^2)} \left(\left(\frac{\hat{\theta}-\theta}{v_\theta^{1/2}} \right)^2 - 2\rho \left(\frac{\hat{\theta}-\theta}{v_\theta^{1/2}} \right) \left(\frac{\hat{\tau}-\tau}{v_\tau^{1/2}} \right) + \left(\frac{\hat{\tau}-\tau}{v_\tau^{1/2}} \right)^2 \right) \right).$$

Hence the log-likelihood function, denoted by $\ell(\theta, \tau)$, is

$$-\frac{1}{2(1-\rho^2)} \left(\frac{(\hat{\theta}-\theta)^2}{v_\theta} - 2\rho \frac{(\hat{\theta}-\theta)(\hat{\tau}-\tau)}{v_\theta^{1/2} v_\tau^{1/2}} + \frac{(\hat{\tau}-\tau)^2}{v_\tau} \right) + \text{constant}.$$

Let $\hat{\tau}_\theta$ denote the value of τ that maximizes $\ell(\theta, \tau)$ with respect to τ , for given θ . We find $\hat{\tau}_\theta$ by minimizing

$$\frac{(\hat{\theta}-\theta)^2}{v_\theta} - 2\rho \frac{(\hat{\theta}-\theta)(\hat{\tau}-\tau)}{v_\theta^{1/2} v_\tau^{1/2}} + \frac{(\hat{\tau}-\tau)^2}{v_\tau} \quad (\text{A.1})$$

with respect to τ . Thus $\hat{\tau}_\theta$ is the solution for τ of

$$\begin{aligned}\frac{\partial(\text{A.1})}{\partial\tau} &= 2\rho \frac{(\hat{\theta} - \theta)}{v_\theta^{1/2} v_\tau^{1/2}} - 2 \frac{(\hat{\tau} - \tau)}{v_\tau} = 0 \\ \iff \rho \frac{(\hat{\theta} - \theta)}{v_\theta^{1/2}} - \frac{\hat{\tau} - \tau}{v_\tau^{1/2}} &= 0 \\ \iff \frac{\hat{\tau} - \tau}{v_\tau^{1/2}} &= \rho \frac{(\hat{\theta} - \theta)}{v_\theta^{1/2}}.\end{aligned}$$

Therefore

$$\frac{\hat{\tau} - \hat{\tau}_\theta}{v_\tau^{1/2}} = \rho \frac{(\hat{\theta} - \theta)}{v_\theta^{1/2}}.$$

Hence (A.1), with τ replaced by $\hat{\tau}_\theta$, is

$$\begin{aligned}\frac{(\hat{\theta} - \theta)^2}{v_\theta} - 2\rho \frac{(\hat{\theta} - \theta)}{v_\theta^{1/2}} \rho \frac{(\hat{\theta} - \theta)}{v_\theta^{1/2}} + \rho^2 \frac{(\hat{\theta} - \theta)^2}{v_\theta} \\ = \frac{(\hat{\theta} - \theta)^2}{v_\theta} - 2\rho^2 \frac{(\hat{\theta} - \theta)^2}{v_\theta} + \rho^2 \frac{(\hat{\theta} - \theta)^2}{v_\theta} \\ = (1 - \rho^2) \frac{(\hat{\theta} - \theta)^2}{v_\theta}.\end{aligned}$$

Hence, $\ell(\theta, \hat{\tau}_\theta)$ is equal to

$$-\frac{1}{2(1 - \rho^2)} (1 - \rho^2) \frac{(\hat{\theta} - \theta)^2}{v_\theta} + \text{constant} = -\frac{1}{2} \frac{(\hat{\theta} - \theta)^2}{v_\theta} + \text{constant}.$$

Thus

$$2 \left(l(\hat{\theta}, \hat{\tau}) - l(\theta, \hat{\tau}_\theta) \right) = 2 \times \frac{1}{2} \frac{(\hat{\theta} - \theta)^2}{v_\theta}.$$

Therefore, for $\theta = \theta'$, the SRLR statistic is

$$r_1(\theta') = \text{sign}(\hat{\theta} - \theta') \sqrt{2 \left(l(\hat{\theta}, \hat{\tau}) - l(\theta', \hat{\tau}_{\theta'}) \right)} = \frac{\hat{\theta} - \theta'}{v_\theta^{1/2}}. \quad (\text{A.2})$$

Let $\hat{\theta}_\tau$ denote the value of θ that maximizes $l(\theta, \tau)$ with respect to θ , for

given τ . Similarly to (A.2),

$$\text{sign}(\hat{\tau} - \tau) \sqrt{2 \left(l(\hat{\theta}, \hat{\tau}) - l(\hat{\theta}_\tau, \tau) \right)} = \frac{\hat{\tau} - \tau}{v_\tau^{1/2}}.$$

Therefore, for $\tau = t$, the SRLR statistic is

$$r_2 = \text{sign}(\hat{\tau} - t) \sqrt{2 \left(l(\hat{\theta}, \hat{\tau}) - l(\hat{\theta}_t, t) \right)} = \frac{\hat{\tau} - t}{v_\tau^{1/2}}.$$

A.2 Numerical results for the *local coverage probability* and *local scaled expected length* of the confidence interval $\text{ACI}_W(\hat{\beta}^*)$

The Monte Carlo simulation estimation of the local coverage probabilities of the Wald-based confidence intervals $I_W(\mathbf{y}^*; c)$ and $\text{ACI}_W(\hat{\beta}^*)$ are very similar. For the sake of brevity, we describe only the Monte Carlo simulation estimation of the former.

Let $\theta^* = g(\beta^*)$. We estimate the coverage probability $P_{\beta^*}(\theta^* \in I_W(\mathbf{y}^*; c))$ by Monte Carlo simulation as follows. We carry out M independent simulation runs. The k th simulation run we generate an observation of \mathbf{y}^* and record $\mathbf{1}(\theta^* \in I_W(\mathbf{y}^*; c))$. Using the recorded results for the M simulation runs, we estimate the coverage probability and the standard error of this estimate are carried out in the obvious way.

The top panel of Figure A.1 presents approximate 95% confidence intervals for the coverage probability of the confidence interval $I_W(\mathbf{y}^*; 0.05)$ for θ^* , which has nominal coverage 0.95, evaluated at $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$. The bottom panel of this figure presents approximate 95% confidence intervals for the coverage probability of the confidence interval $\text{ACI}_W(\hat{\beta}^*)$ for θ^* , which has nominal coverage 0.95, evaluated on the same grid of values of

γ^* . For each of these panels, the number of simulation runs $M = 40,000$ for each value of γ^* . These panels show that the *local minimum coverage probabilities* of $I_W(\mathbf{y}^*; 0.05)$ and $\text{ACI}_W(\hat{\boldsymbol{\beta}}^*)$ are approximately 0.95 and 0.94, respectively.

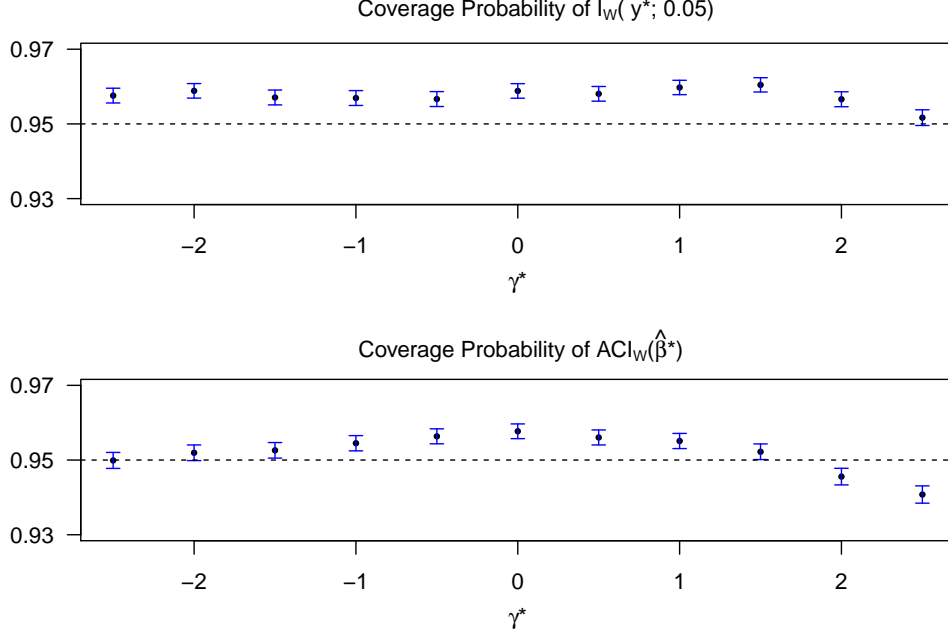


Figure A.1: Approximate 95% confidence intervals for the coverage probabilities of the confidence intervals $I_W(\mathbf{y}^*; 0.05)$ (top panel) and $\text{ACI}_W(\hat{\boldsymbol{\beta}}^*)$ (bottom panel), both with nominal coverage 0.95, for $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$.

We used the method described in subsection A.7.1 to estimate *local minimum coverage probability* of $I_W(\mathbf{y}^*; c)$, for $c = 0.03, 0.05$ and 0.07 . Here, $M' = 10,000$. Figure A.2 plots these estimates. The red straight line is the least squares fit to this data.

We also used the method described in subsection A.7.1 to estimate *local minimum coverage probability* of $\text{ACI}_W(\hat{\boldsymbol{\beta}}^*)$, with nominal coverage 0.95. Here, $M' = 10,000$. The resulting estimate is 0.940866. Using the method described in subsection A.7.2, we estimated $\tilde{c} = 0.06502$.

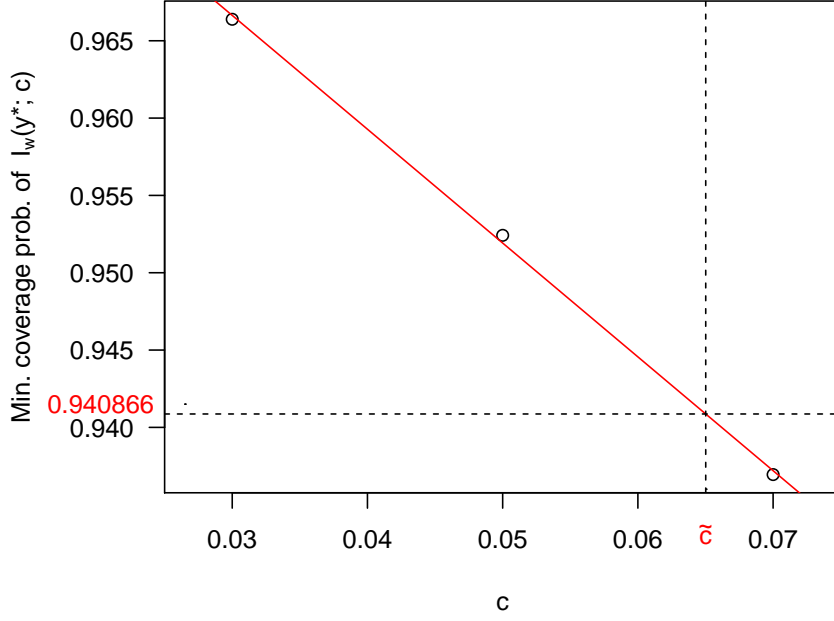


Figure A.2: Plot of the estimated minimum coverage probability of $I_W(\mathbf{y}^*; c)$ for $c = 0.03, 0.05$ and 0.07 . The red straight line is the least squares fit to this data.

The left panel of Figure A.3 presents approximate 95% confidence intervals for the *local scaled expected length* of the confidence interval $ACI_W(\hat{\beta}^*)$, with nominal coverage 0.95, evaluated at each $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$ using $M = 40,000$ simulation runs. These approximate 95% confidence intervals were found using the simplifying approximation that \tilde{c} is computed without error. The right panel of this figure presents the scaled expected length $SEL(\gamma^*; \tilde{\beta})$ for the confidence interval $CI(b_{\rho(\tilde{\beta})}, s_{\rho(\tilde{\beta})})$, found using `ciuupi`, with $\rho(\tilde{\beta}) = -0.399855$.

Note that, for the same nominal coverage 0.95, the *local scaled expected length* of the confidence interval $ACI_W(\hat{\beta}^*)$ substantially exceeds that of $ACI_L(\mathbf{y}^*)$ for every $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$. This shows that, in terms of *local scaled expected length*, $ACI_L(\mathbf{y}^*)$ outperforms $ACI_W(\hat{\beta}^*)$.

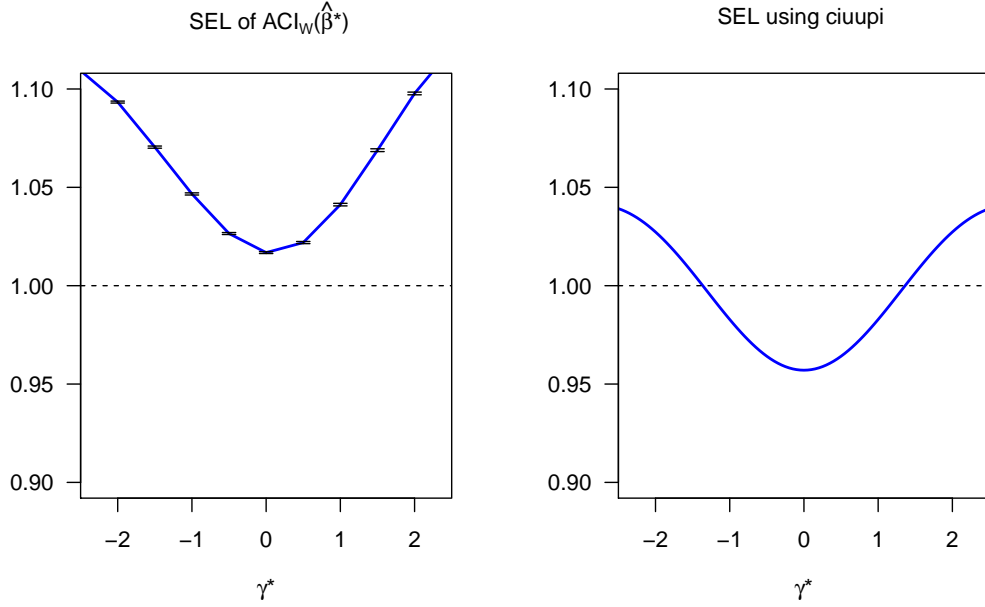


Figure A.3: The left panel presents approximate 95% confidence intervals for the *local scaled expected length* of the confidence interval $\text{ACI}_W(\hat{\beta}^*)$, which has nominal coverage 0.95, for $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$. The right panel is the graph of $SEL(\gamma^*; \tilde{\beta})$ for the confidence interval $\text{CI}(b_{\rho(\tilde{\beta})}, s_{\rho(\tilde{\beta})})$, found using the R package `ciuupi`.

A.3 Expressions for the coverage probabilities of $\text{I}_L(\mathbf{y}; c)$ and $\text{ACI}_L(\mathbf{y})$ that do not require the computation of the endpoints of these confidence intervals

We make the following assumption.

Assumption A: For the chosen true value of β , there is a set \mathcal{Y}_β of values of \mathbf{y} such that (a) the probability that $\mathbf{y} \in \mathcal{Y}_\beta$ is very close to 1 and (b) $r_1(\theta' | \mathbf{y})$ is a decreasing function of θ' for all $\mathbf{y} \in \mathcal{Y}_\beta$.

The coverage probability of the likelihood-based confidence interval $\text{I}_L(\mathbf{y}; c)$

can be computed to a very good approximation, without computing the endpoints of this confidence interval, as follows. Since

$$\left\{\widehat{\theta}_l \leq \theta \leq \widehat{\theta}_u\right\} = \left\{-z_{1-\alpha/2} \leq r_1(\theta | \mathbf{y}) \leq z_{1-\alpha/2}\right\},$$

the coverage probability of $I_L(\mathbf{y}; c)$ is, to a very good approximation, given by $P_{\beta}(-z_{1-\alpha/2} \leq r_1(\theta | \mathbf{y}) \leq z_{1-\alpha/2})$.

Similarly, the coverage probability of $ACI_L(\mathbf{y})$ can be computed to a very good approximation, without computing the endpoints of this confidence interval, as follows. Since

$$\begin{aligned} & \left\{\widetilde{\theta}_l \leq \theta \leq \widetilde{\theta}_u\right\} \\ &= \left\{b_{\rho(\widehat{\beta})}(r_2(\mathbf{y})) - s_{\rho(\widehat{\beta})}(r_2(\mathbf{y})) \leq r_1(\theta | \mathbf{y}) \leq b_{\rho(\widehat{\beta})}(r_2(\mathbf{y})) + s_{\rho(\widehat{\beta})}(r_2(\mathbf{y}))\right\}, \end{aligned}$$

the coverage probability of $ACI_L(\mathbf{y})$ is, to a very good approximation, given by

$$P_{\beta}\left(b_{\rho(\widehat{\beta})}(r_2(\mathbf{y})) - s_{\rho(\widehat{\beta})}(r_2(\mathbf{y})) \leq r_1(\theta | \mathbf{y}) \leq b_{\rho(\widehat{\beta})}(r_2(\mathbf{y})) + s_{\rho(\widehat{\beta})}(r_2(\mathbf{y}))\right).$$

A.4 Evidence in favour of the correctness of Assumption A

In this section we report the results of a Monte Carlo simulation study that provides evidence in favour of the correctness of Assumption A (defined in Section A.3), with β and \mathbf{y} replaced by β^* and \mathbf{y}^* , respectively. For each $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$, we carried out $M = 40,000$ independent simulation runs. The k th simulation run generated an observation of \mathbf{y}^* , followed by a numerical check that $r_1(\theta' | \mathbf{y}^*)$ is a decreasing function of θ' . It was found that, for every $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$ and every simulation run, this numerical check showed that $r_1(\theta' | \mathbf{y}^*)$ is a decreasing function of θ' . This

provides strong evidence in favour of the correctness of Assumption A, with β and \mathbf{y} replaced by β^* and \mathbf{y}^* , respectively.

We now provide a detailed description of this numerical check. In the case of the linear regression model described in Section 2.2, the SRLR statistic

$$r_1(\theta') = \frac{\hat{\theta} - \theta'}{(\text{var}(\hat{\theta}))^{1/2}}.$$

The similarity between the bivariate normal distribution (2.2) and the asymptotic bivariate normal distribution (2.20) suggests that, to a rough approximation,

$$r_1(\theta' | \mathbf{y}^*) \approx \frac{\hat{\theta}^* - \theta'}{(\text{avar}(\hat{\theta}; \tilde{\beta}))^{1/2}}.$$

Based on this approximation, we chose the following equally-spaced grid of values of θ' . Let $\theta'_1 = \hat{\theta}^* - 5(\text{avar}(\hat{\theta}; \tilde{\beta}))^{1/2}$, $\theta'_2 = \hat{\theta}^* - 4(\text{avar}(\hat{\theta}; \tilde{\beta}))^{1/2}, \dots, \theta'_{10} = \hat{\theta}^* + 4(\text{avar}(\hat{\theta}; \tilde{\beta}))^{1/2}$ and $\theta'_{11} = \hat{\theta}^* + 5(\text{avar}(\hat{\theta}; \tilde{\beta}))^{1/2}$. Consequently, to a rough approximation, $r_1(\theta'_1 | \mathbf{y}^*) \approx -5, r_1(\theta'_2 | \mathbf{y}^*) \approx -4, \dots, r_1(\theta'_{10} | \mathbf{y}^*) \approx 4$ and $r_1(\theta'_{11} | \mathbf{y}^*) \approx 5$. For each $\gamma^* \in \{-2.5, -2, \dots, 2, 2.5\}$, in the k th simulation run we recorded whether or not $r_1(\theta'_{i+1} | \mathbf{y}^*) - r_1(\theta'_i | \mathbf{y}^*) < 0$ for all $i \in \{1, 2, \dots, 10\}$.

A.5 Fast evaluation of the functions b_ρ and s_ρ for any given ρ

To compute the Wald-based confidence interval $\text{ACI}_W(\hat{\beta}^*)$, the expression (2.24) and the likelihood-based confidence interval $\text{ACI}_L(\mathbf{y}^*)$, we first compute $\rho(\hat{\beta}^*)$. Then we need to evaluate the functions b_ρ and s_ρ for $\rho = \rho(\hat{\beta}^*)$. For any given ρ , the R package `ciuppi` computes the information needed to evaluate the functions b_ρ and s_ρ . On a typical PC this takes roughly 7 minutes, when we use the natural cubic spline option in `ciuppi`. To greatly speed

up the computation of this information, we use a *look-up table*, followed by the use of *linear interpolation*.

As described by Mainzer & Kabaila (2019), the functions b_ρ and s_ρ have the following properties: $b_\rho : \mathbb{R} \rightarrow \mathbb{R}$ is an odd continuous function and $s_\rho : \mathbb{R} \rightarrow [0, \infty)$ is an even continuous function. In addition, $b_\rho(x) = 0$ and $s_\rho(x) = z_{1-\alpha/2}$ for all $|x| \geq 6$. The functions b_ρ and s_ρ are fully specified by the vector $(b_\rho(1), b_\rho(2), \dots, b_\rho(5), s_\rho(0), s_\rho(1), \dots, s_\rho(5))$. In other words, this is the information needed to evaluate the functions b_ρ and s_ρ . By assumption, $b_\rho(-6) = 0$, $b_\rho(0) = 0$, $b_\rho(6) = 0$, $s_\rho(-6) = z_{1-\alpha/2}$, $s_\rho(6) = z_{1-\alpha/2}$, $(b_\rho(-1), b_\rho(-2), \dots, b_\rho(-5)) = (-b_\rho(1), -b_\rho(2), \dots, -b_\rho(5))$ and $(s_\rho(-1), \dots, s_\rho(-5)) = (s_\rho(1), \dots, s_\rho(5))$. The values of $b_\rho(x)$ and $s_\rho(x)$ for any $x \in [-6, 6]$ are found by natural cubic spline interpolation (using the command `bsspline` from the package `ciuppi`) for the given values of $b_\rho(i)$ and $s_\rho(i)$ for $i = -6, -5, \dots, 0, 1, \dots, 5, 6$. The vector $(b_\rho(1), b_\rho(2), \dots, b_\rho(5), s_\rho(0), s_\rho(1), \dots, s_\rho(5))$ is a smooth function of ρ .

For fast evaluation of the functions b_ρ and s_ρ , for any given ρ , we use the following method. Let $\rho(1) = -0.96, \rho(2) = -0.92, \dots, \rho(25) = 0, \dots, \rho(48) = 0.92, \rho(49) = 0.96$. We carry out the following preliminary computations (using the command `bsciuppi` from the package `ciuppi`) to prepare a *look-up table* of values of the vector

$$(b_{\rho(i)}(1), b_{\rho(i)}(2), \dots, b_{\rho(i)}(5), s_{\rho(i)}(0), s_{\rho(i)}(1), \dots, s_{\rho(i)}(5))$$

for $i = 1, \dots, 49$. This *look-up table* is used as follows. If $\rho \leq \rho(1)$ then we approximate $b_\rho(x)$ and $s_\rho(x)$ by $b_{\rho(1)}(x)$ and $s_{\rho(1)}(x)$, respectively, using the *look-up table*. Similarly, if $\rho \geq \rho(49)$ then we approximate $b_\rho(x)$ and $s_\rho(x)$ by $b_{\rho(49)}(x)$ and $s_{\rho(49)}(x)$, respectively, using the *look-up table*. Otherwise, we compute i such that $\rho(i) \leq \rho \leq \rho(i+1)$. We then approximate the vector

$(b_\rho(1), b_\rho(2), \dots, b_\rho(5), s_\rho(0), s_\rho(1), \dots, s_\rho(5))$ by

$$\begin{aligned} & \left(1 - \frac{\rho - \rho(i)}{\rho(i+1) - \rho(i)}\right) (b_{\rho(i)}(1), b_{\rho(i)}(2), \dots, b_{\rho(i)}(5), s_{\rho(i)}(0), s_{\rho(i)}(1), \dots, s_{\rho(i)}(5)) \\ & + \left(\frac{\rho - \rho(i)}{\rho(i+1) - \rho(i)}\right) (b_{\rho(i+1)}(1), b_{\rho(i+1)}(2), \dots, b_{\rho(i+1)}(5), s_{\rho(i+1)}(0), s_{\rho(i+1)}(1), \dots, \\ & \quad s_{\rho(i+1)}(5)). \end{aligned}$$

In other words, we use *linear interpolation*.

A.6 Computation of the SRLR statistics

We need to compute the signed root likelihood ratio test (SRLR) statistic

$$r_1(\theta^* | \mathbf{y}^*) = \text{sign}(\hat{\theta}^* - \theta^*) \sqrt{2 \left(\ell(\hat{\boldsymbol{\beta}}^* | \mathbf{y}^*) - \ell(\hat{\boldsymbol{\beta}}^*(\theta^*; \theta) | \mathbf{y}^*) \right)},$$

where $\hat{\boldsymbol{\beta}}^*(\theta^*; \theta)$ maximises $\ell(\boldsymbol{\beta} | \mathbf{y}^*)$ with respect to $\boldsymbol{\beta}$, subject to the constraint that $g(\boldsymbol{\beta}) = \theta^*$. For the model used for the Morphine/Amidone data, this is a nonlinear constraint on $\boldsymbol{\beta}$. Consequently, $\hat{\boldsymbol{\beta}}^*(\theta^*; \theta)$ needs to be computed using an optimization method with nonlinear constraints such as the `slsqp` function in the `nloptr` package in R.

We also need to compute the SRLR statistic

$$r_2(\mathbf{y}^*) = \text{sign}(\hat{\tau}^*) \sqrt{2 \left(\ell(\hat{\boldsymbol{\beta}}^* | \mathbf{y}^*) - \ell(\hat{\boldsymbol{\beta}}^*(0; \tau) | \mathbf{y}^*) \right)},$$

where $\hat{\boldsymbol{\beta}}^*(0; \tau)$ maximises $\ell(\boldsymbol{\beta} | \mathbf{y}^*)$ with respect to $\boldsymbol{\beta}$, subject to the constraint that $h(\boldsymbol{\beta}) = 0$. For the model used for the Morphine/Amidone data, this is a linear constraint on $\boldsymbol{\beta}$. Consequently, $\hat{\boldsymbol{\beta}}^*(0; \tau)$ is computed using an optimization method with linear constraints such as the `solnl` function in the ‘`Nlcoptim`’ package in R.

A.7 Computation of \tilde{c}

The method used to compute \tilde{c} such that the *local minimum coverage probabilities* of likelihood-based confidence intervals $\text{ACI}_L(\mathbf{y}^*)$ and $I_L(\mathbf{y}^*; \tilde{c})$ are the same is very similar to the method used to compute \tilde{c} such that the *local minimum coverage probabilities* of Wald-based confidence intervals $\text{ACI}_W(\hat{\beta}^*)$ and $I_W(\mathbf{y}^*; \tilde{c})$ are the same. For the sake of brevity, we describe only the latter method.

A.7.1 Monte Carlo simulation estimation of the *minimum coverage probabilities* of $\text{ACI}_W(\hat{\beta}^*)$ and $I_W(\mathbf{y}^*; c)$

The computation of the *local minimum coverage probabilities* of Wald-based confidence intervals $\text{ACI}_W(\hat{\beta}^*)$ and $I_W(\mathbf{y}^*; c)$, for given c , are very similar. For the sake of brevity, we describe only the latter.

If we estimate the *local minimum coverage probability* of $I_W(\mathbf{y}^*; c)$ by choosing the smallest of the estimated coverage probabilities, for $\gamma^* \in \{-u, -u + \delta, \dots, u - \delta, u\}$, then this estimate will be biased downwards. We therefore use the following three step process.

Step 1 : Estimate the coverage probability for each $\gamma^* \in \{-u, -u + \delta, \dots, u - \delta, u\}$, where $\delta = u/5$. We use M' simulation runs for each value of γ^* . Pick the 3 values of γ^* that have the smallest estimated coverage probability.

Step 2 : For the 3 values of γ^* chosen in Step 1, run new simulations using $10M'$ simulation runs for each value of γ^* . Choose the value of γ^* (out of these 3 values) that minimizes the estimated coverage probability.

Step 3 : For the value of γ^* chosen in Step 2, run a new simulation with $100M'$ simulation runs to obtain the final estimate of the *local minimum coverage probability*.

This technique is a variant of the technique used in Section 3.1 of [Kabaila & Leeb \(2006\)](#).

A.7.2 Use of a fitted straight line in c to compute \tilde{c}

Note that

$$P_{\beta^*}(\theta^* \in I_W(\mathbf{y}^*; c)) = P_{\beta^*} \left(-z_{1-c/2} \leq \frac{\hat{\theta}^* - \theta^*}{(\text{avar}(\hat{\theta}; \hat{\beta}^*))^{1/2}} \leq z_{1-c/2} \right) \approx 1 - c.$$

Consequently, to compute \tilde{c} , we make the very reasonable assumption that the *local minimum coverage probability* of the Wald-based confidence interval $I_W(\mathbf{y}^*; c)$ is approximately a straight line function of c , for c close to α . We choose 3 values of c : $\alpha - \delta_c$, α and $\alpha + \delta_c$, where δ_c is a judiciously-chosen small positive number. We then fit a straight line to the *local minimum coverage probability* of $I_W(\mathbf{y}^*; c)$ evaluated at these 3 values of c to compute \tilde{c} .

A.8 The distribution of q_L^*

We examine the distribution of the random variable q_L^* , which is defined in subsection [2.10.2](#). For each $\gamma^* \in \{-2.5, -1.5, -0.5, 0.5, 1.5, 2.5\}$, we carried out $M = 10,000$ simulation runs to obtain a sample $q_L^*(1), \dots, q_L^*(M)$. Table [A.1](#) presents some descriptive statistics for each sample. These results show that the distribution of q_L^* does not have any long or heavy tails.

Table A.1: Some descriptive statistics for $q_L^*(1), \dots, q_L^*(M)$ for $\gamma^* \in \{-2.5, -1.5, -0.5, 0.5, 1.5, 2.5\}$ where the number of simulation runs, for each value of γ^* , is $M = 10,000$.

γ^*	Min	1st Qu.	Median	Mean	3rd Qu.	Max
-2.5	0.869597	1.002773	1.028431	1.015938	1.041444	1.069838
-1.5	0.862585	0.949561	0.997398	0.991340	1.036458	1.068194
-0.5	0.865069	0.922768	0.947702	0.958856	0.989541	1.077855
0.5	0.834319	0.914651	0.938062	0.950739	0.980454	1.077168
1.5	0.847077	0.924437	0.978760	0.977689	1.033684	1.286713
2.5	0.854067	0.988663	1.027716	1.010364	1.046152	2.631895

A.9 R programs for computing the coverage probability and the scaled expected length of the likelihood-based confidence intervals.

In this appendix, we list the R programs for computing the coverage probability of the profile likelihood confidence interval, $I_L(\mathbf{y}^*; 0.05)$, and the coverage probability and the scaled expected length of the likelihood-based confidence interval, $ACI_L(\mathbf{y}^*)$, that utilizes the uncertain prior information.

A.9.1 Computation of the γ^* values

```
GamVec <- function(Intdata, m, delta) {  
  # Compute the vector of gamma values for a given  
  # vector of tau values.  
  #  
  # Input:  
  # Intdata: Grewal1952.csv dataset  
  # delta: gap between the values of gamma vector  
  # m: choose m and delta such that m*delta = 10  
  #  
  # Output:  
  # A vector of gamma values of the same  
  # length as tau.vec.  
  #  
  # Written by N Ranathunga in May 2020  
  
  avartau <- AvarTau(Intdata)  
  tau.vec <- TauVec(Intdata, m, delta)  
  
  # Set up a vector to store results  
  gamma.vec <- rep(0, length(tau.vec))  
  
  for(i in c(1:length(tau.vec))) {  
    gamma.vec[i] <- tau.vec[i] / sqrt(avartau)  
  }  
  out <- gamma.vec  
}
```

```

AvarTau <- function(Intdata){
  # Compute the asymptotic variance of tau.
  #
  # Input:
  # Intdata: Grewal1952.csv dataset
  # x= the 6 by 4 design matrix
  # nvec= the 6-vector of cluster sizes
  #
  # Output:
  # Asymptotic variance of tau.
  #
  # Written by N Ranathunga in May 2020

  para.vec <- ParaVec(Intdata)
  x <- as.matrix(Intdata[,c(7:10)], nrow=6, ncol=4)
  nvec <- as.vector(Intdata[,6])
  diff.hbeta <- c(0, 1, 0, -1)

  # Set up a vector to store results
  diag.vec <- rep(0, 6)

  for (i in 1 : 6) {
    term1 <- para.vec[1]*x[i, 1] +
      ((para.vec[2] + para.vec[4])/2) * x[i, 2] +
      para.vec[3]*x[i, 3] +
      ((para.vec[2] + para.vec[4])/2) * x[i, 4]

    term2 <- exp(term1) / (1 + exp(term1))^2
    diag.vec[i] <- nvec[i] * term2
  }

  diag.mat <- diag(diag.vec, nrow = 6, ncol = 6)
  info.mat <- t(x) %*% diag.mat %*% x
  inv.info.mat <- solve(info.mat)
  out <- t(diff.hbeta) %*% inv.info.mat %*% diff.hbeta
}

```

```

TauVec <- function(Intdata, m, delta) {
  # Compute the vector of tau values of
  # length 2*m+1
  #
  # Input:
  # Intdata: Grewal1952.csv dataset
  # delta: gap between the values of gamma vector
  # m: choose m and delta such that m*delta = 10
  #
  # Output:
  # A vector of tau values.
  #
  # Written by N Ranathunga in May 2020

  avartau <- AvarTau(Intdata)

  # Set up a vector to store results
  tau.vec <- rep(0, 2*m+1)

  for(i in c(1:(2*m+1))) {
    tau.vec[i] <- (i-m-1)*delta*sqrt(avartau)
  }
  out <- tau.vec
}

```


A.9.2 Coverage probability of the likelihood-based confidence intervals, $I_L(\mathbf{y}^*; 0.05)$ and $ACI_L(\mathbf{y}^*)$

```

CovProblikCI <- function(tau, M, Intdata, z, bsmat, b0.vec,
  alpha){
  # Compute the coverage probabilities of the profile
  # likelihood
  # confidence interval and the confidence interval utilizing
  # uncertain prior information in general regression model
  # using the signed root likelihood ratios for a given tau.
  #
  # Input:
  # tau: scalar paramer with uncertain prior information
  # para.vec: the p-vector of parameters
  # M: number of simulation runs
  # n: number of responses
  # Intdata: Grewal1952.csv dataset
  # z: a specified value where 0 < z < 100
  # bsmat: this matrix has (b(1),b(2),...,b(5),s(0),s(1),...,
  #       s(5))
  #       vectors that specifies the CIUUPI for different
  #       values of rho.
  # b0.vec: a 4-vector of starting values for searching the
  #         optimum
  #
  # Output:
  # List of two values of the coverage probabilities.
  #
  # Written by N Ranathunga in May 2020

  c <- log((z/100) / (1 - (z/100)))
  para.vec <- ParaVec(Intdata)
  theta.obs <- (c - para.vec[1])/((para.vec[2]+para.vec[4])/2
    + tau/2) -
    (c - para.vec[3])/((para.vec[2]+para.vec[4])/2 -
    tau/2)
  x <- as.matrix(Intdata[,c(7:10)], nrow=6, ncol=4)
  nvec <- as.vector(Intdata[,6])
  zquant <- qnorm(1 - alpha/2, 0, 1)

  # Initialize vectors
  record.ACIL <- rep(0, M)
  record.IL <- rep(0, M)

  for (i in 1 : M) {

    # Generating Cases and Controls
    mu.vec <- x %*% para.vec
    term.vec <- exp(mu.vec)
    pvec <- term.vec / (1 + term.vec)
    r2vec <- rbinom(rep(1, 6), nvec, pvec)
    nminr2vec <- nvec - r2vec
    data.mat <- cbind(Intdata, r2vec, nminr2vec)

    # Fitting the binomial logistic regression model
    glm.fit <- glm(cbind(r2vec, nminr2vec) ~ x1 + x2 + x3 +
      x4 - 1,

```

```

data = data.mat, family = binomial(
  logit))

beta1 <- coef(glm.fit)[1]
beta2 <- coef(glm.fit)[2]
beta3 <- coef(glm.fit)[3]
beta4 <- coef(glm.fit)[4]

loglik.thetahat <- logLik(glm.fit)[1]
theta.hat <- (1/beta2)*(c - beta1) - (1/beta4)*(c - beta3
)
tau.hat <- beta2 - beta4

#Finding rho value
inv.info.matrix <- vcov(glm.fit)
diff.gbeta <- c(-1/beta2, (beta1 - c)/(beta2^2),
  1/beta4, (c - beta3)/(beta4^2))
diff.hbeta <- c(0, 1, 0, -1)

avar.theta <- t(diff.gbeta) %*% inv.info.matrix %*% diff.
  gbeta
avar.tau <- t(diff.hbeta) %*% inv.info.matrix %*% diff.
  hbeta
acov <- t(diff.gbeta) %*% inv.info.matrix %*% diff.hbeta
rho.est <- acov / sqrt(avar.theta*avar.tau)

# Find the vector (b(1),b(2),...,b(5),s(0),s(1),...,s(5))
  that specifies the
# CIUUPI:
delta.rho <- 0.04
rho.vec <- seq(-0.96, 0.96, by=delta.rho)
if (rho.est >= -0.96 && rho.est <= 0.96){

  up.i <- 1 + (0.96/delta.rho) + ceiling(rho.est/delta.
    rho)
  low.i <- up.i - 1
  term1 <- as.vector((rho.est - rho.vec[low.i])/(rho.vec
    [up.i] - rho.vec[low.i]))
  bsvec <- (1 - term1)*bsmat[low.i, ] + term1*bsmat[up.i,
    ]

} else if (rho.est < -0.96) {
  bsvec <- bsmat[1, ]
} else {
  bsvec <- bsmat[49, ]
}

# Computing the ritheta function
r1theta <- Fun_r1(theta.prime=theta.obs, theta.hat,
  loglik.thetahat, nvec, rvec=r2vec, x=x,
  b0.vec=b0.vec, z=z)

# Computing the r2 function
r2.stat <- Fun_r2(tau.hat, loglik.thetahat, nvec, rvec=
  r2vec, x,
  b0.vec)

# Record if r1theta is in [br2-sr2, br2+sr2] or not
val <- r2.stat
bs <- ciuupi::bsspline(val, bsvec, alpha, natural = 1)

```

```

br2 <- bs$b
sr2 <- bs$s

if (r1theta >= br2-sr2 && r1theta <= br2+sr2) {
  record.ACIL[i] <- 1
} else {
  record.ACIL[i] <- 0
}

# Record if r1theta is in [-zquant, zquant] or not
if (r1theta >= -zquant && r1theta <= zquant) {
  record.IL[i] <- 1
} else {
  record.IL[i] <- 0
}

}

# Return the coverage probability of I_L(y*) and ACI_L(y*)
cp.IL <- mean(record.IL)
cp.ACIL <- mean(record.ACIL)
out <- list(cpIL=cp.IL, cpACIL=cp.ACIL)
}

```

```

ParaVec <- function(Intdata){
  # Compute the MLE of beta vector.
  #
  # Input:
  # Intdata: Grewal1952.csv dataset
  #
  # Output:
  # A 4-vector of MLEs of beta vector.
  #
  # Written by N Ranathunga in May 2020

  # Fitting a binomial logistic regression
  glm.fit <- glm(cbind(r, n_minus_r) ~ x1 + x2 + x3 + x4 - 1,
                 data = Intdata, family = binomial(logit))

  beta1 <- coef(glm.fit)[1]
  beta2 <- coef(glm.fit)[2]
  beta3 <- coef(glm.fit)[3]
  beta4 <- coef(glm.fit)[4]

  para.vec <- c(beta1, beta2, beta3, beta4)
}

```

```

Fun_r1 <- function(theta.prime, theta.hat, loglik.thetahat,
                  nvec, rvec, x, b0.vec, z){
  # Compute the r1(theta.prime) function.
  #
  # Input:
  # theta.prime: a value for theta
  # theta.hat: MLE of theta
  # loglik.thetahat: log likelihood value
  # nvec: 6-vector of cluster sizes
  # rvec: 6-vector of responses
  # x: the 6 by 4 design matrix
  #
  # Output:
  # Value of the r1(theta.prime) function.
  #
  # Written by N Ranathunga in May 2020

  heqfun <- functional::Curry(Funheq, z=z, theta.prime=theta.
    prime)

  S <- nloptr::slsqp(x0=b0.vec, fn = Loglik, heq = heqfun,
                   nvec=nvec,
                   rvec=rvec, x=x)

  loglik.thetaprime <- -S$value
  if (loglik.thetahat - loglik.thetaprime > 0){
    diff <- loglik.thetahat - loglik.thetaprime
  } else diff <- 0

  r1theta <- sign(theta.hat - theta.prime) * sqrt(2 * diff)
}

```

```

Fun_r2 <- function(tau.hat, loglik.thetahat,
                  nvec, rvec, x, b0.vec){
  # Compute the r2(y) function.
  #
  # Input:
  # tau.hat: MLE of tau
  # loglik.thetahat: log likelihood value
  # nvec: 6-vector of cluster sizes
  # rvec: 6-vector of responses
  # x: the 6 by 4 design matrix
  #
  # Output:
  # Value of the r2(y) function.
  #
  # Written by N Ranathunga in May 2020

  S <- nloptr::slsqp(x0=b0.vec, fn = Loglik, heq = FunheqTau,
                  nvec=nvec,
                  rvec=rvec, x=x)
  loglik.tau <- -S$value

  if (loglik.thetahat - loglik.tau > 0){
    diff0 <- loglik.thetahat - loglik.tau
  } else diff0 <- 0

  r2 <- sign(tau.hat) * sqrt(2 * (diff0))

}

```

```

Funheq <- function(bvec, z, theta.prime) {
  # This function calculates the inequality constraint.
  #
  # Inputs
  # bvec: parameter vector
  # theta.prime: a value for theta
  # z: a specified value 0 < z < 100
  #
  # Output
  # Value of the inequality constraint.
  #
  # Written by N. Ranathunga in March 2020

  c <- log((z/100) / (1 - (z/100)))
  out <- (1/bvec[2])*(c - bvec[1]) - (1/bvec[4])*(c - bvec
    [3]) - theta.prime
}

```

```

FunheqTau <- function(bvec) {
  # This function calculates the equality constraint.
  #
  # Inputs
  # bvec = parameter vector
  #
  # Output
  # Value of the equality constraint
  #
  # Written by N. Ranathunga in March 2020

  bvec[2] - bvec[4]
}

```

```

Loglik <- function(bvec, nvec, rvec, x) {
  # This function calculates the log likelihood function of
  # a logistic regression model.
  #
  # Inputs
  # bvec = 4-vector of parameters
  # nvec = 6-vector of cluster sizes
  # r2vec = 6-vector of responses
  # x: the 6 by 4 design matrix
  #
  # Output
  # Value of the log likelihood function
  #
  # Written by N. Ranathunga in March 2020

  term1.vec <- log(factorial(nvec)) - log(factorial(rvec)) -
    log(factorial(nvec - rvec))
  par.vec <- c(bvec[1], bvec[2], bvec[3], bvec[4])
  mu.vec <- x %*% par.vec
  term2.vec <- mu.vec * rvec
  term3.vec <- nvec * log(1 + exp(mu.vec))

  out <- -sum(term1.vec) - sum(term2.vec) + sum(term3.vec)
}

```

A.9.3 Scaled expected length of the likelihood-based confidence interval $ACI_L(y^*)$

```

SELlikCI <- function(tau, para.vec, M, n, Intdata, z, bsmat,
  b0.vec){
  # Compute the coverage probabilities of the profile
  # likelihood
  # confidence interval and the confidence interval
  # utilizing
  # uncertain prior information using the signed root
  # likelihood
  # ratios for a given tau.
  #
  # Input:
  # tau: scalar parameter with uncertain prior information
  # par.vec: the p-vector of parameters
  # M: number of simulation runs
  # x: the n by p design matrix
  # n: number of responses
  # grp: the n-vector of covariate
  # rvec: the n-vector of response
  # nvec: the n-vector of cluster sizes
  #
  # Output:
  # List of two values of the coverage probabilities.
  #
  # Written by N Ranathunga in May 2020

  c <- log((z/100) / (1 - (z/100)))
  para.vec <- ParaVec(Intdata)
  theta.obs <- (c - para.vec[1])/((para.vec[2]+para.vec[4])/2
    + tau/2) -
    (c - para.vec[3])/((para.vec[2]+para.vec[4])/2 - tau/2)
  x <- as.matrix(Intdata[,c(7:10)], nrow=6, ncol=4)
  nvec <- as.vector(Intdata[,6])
  zquant <- qnorm(1 - alpha/2, 0, 1)

  # Initialize vectors
  length.ACIL <- rep(0, M)
  length.IL <- rep(0, M)

  for (i in 1 : M) {

    # Generating Cases and Controls
    mu.vec <- x %*% para.vec
    term.vec <- exp(mu.vec)
    pvec <- term.vec / (1 + term.vec)
    r2vec <- rbinom(rep(1, 6), nvec, pvec)
    nminr2vec <- nvec - r2vec
    data.mat <- cbind(Intdata, r2vec, nminr2vec)

    # Fitting the binomial logistic regression model
    glm.fit <- glm(cbind(r2vec, nminr2vec) ~ x1 + x2 + x3 +
      x4 - 1,
      data = data.mat, family = binomial(logit))

    beta1 <- coef(glm.fit)[1]
    beta2 <- coef(glm.fit)[2]
  }
}

```

```

beta3 <- coef(glm.fit)[3]
beta4 <- coef(glm.fit)[4]

loglik.thetahat <- logLik(glm.fit)[1]
theta.hat <- (1/beta2)*(c - beta1) - (1/beta4)*(c - beta3
)
tau.hat <- beta2 - beta4

#Finding rho value
inv.info.matrix <- vcov(glm.fit)
diff.gbета <- c(-1/beta2, (beta1 - c)/(beta2^2),
               1/beta4, (c - beta3)/(beta4^2))
diff.hbeta <- c(0, 1, 0, -1)

avar.theta <- t(diff.gbета) %*% inv.info.matrix %*% diff.
gbета
avar.tau <- t(diff.hbeta) %*% inv.info.matrix %*% diff.
hbeta
acov <- t(diff.gbета) %*% inv.info.matrix %*% diff.hbeta
rho.est <- acov / sqrt(avar.theta*avar.tau)

# Find the vector (b(1),b(2),...,b(5),s(0),s(1),...,s(5))
that specifies the
# CIUUPI:
delta.rho <- 0.04
rho.vec <- seq(-0.96, 0.96, by=delta.rho)
if (rho.est >= -0.96 && rho.est <= 0.96){

  up.i <- 1 + (0.96/delta.rho) + ceiling(rho.est/delta.
rho)
  low.i <- up.i - 1
  term1 <- as.vector((rho.est - rho.vec[low.i])/(rho.vec
[up.i] - rho.vec[low.i]))
  bsvec <- (1 - term1)*bsmat[low.i, ] + term1*bsmat[up.i,
]

} else if (rho.est < -0.96) {
  bsvec <- bsmat[1, ]
} else {
  bsvec <- bsmat[49, ]
}

# Computing the r2 function
r2.stat <- Fun_r2(tau.hat, loglik.thetahat, nvec, rvec=
r2vec, x,
                b0.vec)

# Finding the length of ACI_L(y*)
val <- r2.stat
bs <- bsspline(val, bsvec, alpha, natural = 1)
br2 <- bs$b
sr2 <- bs$s

thetatilde.l <- uniroot(rilowACIL, theta.hat=theta.hat,
loglik.thetahat=loglik.thetahat,
                    nvec=nvec, rvec=r2vec, x=x, b0.
vec=b0.vec, z=z, br2=br2,

```



```

                                sr2=sr2, interval = c(-1,
                                1), extendInt="yes")$
                                root

thetatilde.u <- uniroot(r1upACIL, theta.hat=theta.hat,
                        loglik.thetahat=loglik.thetahat,
                                nvec=nvec, rvec=r2vec, x=x, b0.
                                vec=b0.vec, z=z, br2=br2,
                                sr2=sr2, interval = c(-1, 1),
                                extendInt="yes")$root

length.ACIL[i] <- thetatilde.u - thetatilde.l

# Finding the length of I_L(y*)

thetahat.l <- uniroot(r1lowIL, theta.hat=theta.hat,
                      loglik.thetahat=loglik.thetahat,
                                nvec=nvec, rvec=r2vec, x=x, b0.vec=
                                b0.vec, z=z, zquant=zquant,
                                interval = c(-1, 1), extendInt="yes
                                ")$root

thetahat.u <- uniroot(r1upIL, theta.hat=theta.hat, loglik
                      .thetahat=loglik.thetahat,
                                nvec=nvec, rvec=r2vec, x=x, b0.vec=
                                b0.vec, z=z, zquant=zquant,
                                interval = c(-1, 1), extendInt="yes
                                ")$root

length.IL[i] <- thetahat.u - thetahat.l
}

# Return the scaled expected length ACI_L(y*)
SEL.ACIL <- mean(length.ACIL/length.IL)
}

```

```

r1lowACIL <- function(theta.prime, theta.hat, loglik.thetahat
,
                        nvec, rvec, x, b0.vec, z, br2, sr2){
# Compute the function br2 + sr2 - r1(theta.prime)
#
# Input:
# theta.prime: a value for theta
# theta.hat: MLE of theta
# loglik.thetahat: log likelihood value
# nvec: 6-vector of cluster sizes
# rvec: 6-vector of responses
# x: the 6 by 4 design matrix
#
# Output:
# Value of the r1(theta.prime) function.
#
# Written by N Ranathunga in May 2020

heqfun <- functional::Curry(Funheq, z=z, theta.prime=theta.
prime)

S <- nloptr::slsqp(x0=b0.vec, fn = Loglik, heq = heqfun,
nvec=nvec,
rvec=rvec, x=x)

loglik.thetaprime <- -S$value
if (loglik.thetahat - loglik.thetaprime > 0){
diff <- loglik.thetahat - loglik.thetaprime
} else diff <- 0

out <- br2 + sr2 - sign(theta.hat - theta.prime) * sqrt(2 *
diff)
}

```

```

r1upACIL <- function(theta.prime, theta.hat, loglik.thetahat,
                     nvec, rvec, x, b0.vec, z, br2, sr2){
  # Compute the function br2 - sr2 - r1(theta.prime)
  #
  # Input:
  # theta.prime: a value for theta
  # theta.hat: MLE of theta
  # loglik.thetahat: log likelihood value
  # nvec: 6-vector of cluster sizes
  # rvec: 6-vector of responses
  # x: the 6 by 4 design matrix
  #
  # Output:
  # Value of the r1(theta.prime) function.
  #
  # Written by N Ranathunga in May 2020

  heqfun <- functional::Curry(Funheq, z=z, theta.prime=theta.
    prime)

  S <- nloptr::slsqp(x0=b0.vec, fn = Loglik, heq = heqfun,
    nvec=nvec,
    rvec=rvec, x=x)

  loglik.thetaprime <- -S$value
  if (loglik.thetahat - loglik.thetaprime > 0){
    diff <- loglik.thetahat - loglik.thetaprime
  } else diff <- 0

  out <- br2 - sr2 - sign(theta.hat - theta.prime) * sqrt(2 *
    diff)
}

```

```

r1lowIL <- function(theta.prime, theta.hat, loglik.thetahat,
                    nvec, rvec, x, b0.vec, z, zquant){
  # Compute the function zquant - r1(theta.prime)
  #
  # Input:
  # theta.prime: a value for theta
  # theta.hat: MLE of theta
  # loglik.thetahat: log likelihood value
  # nvec: 6-vector of cluster sizes
  # rvec: 6-vector of responses
  # x: the 6 by 4 design matrix
  #
  # Output:
  # Value of the r1(theta.prime) function.
  #
  # Written by N Ranathunga in May 2020

  heqfun <- functional::Curry(Funheq, z=z, theta.prime=theta.
    prime)

  S <- nloptr::slsqp(x0=b0.vec, fn = Loglik, heq = heqfun,
    nvec=nvec,
    rvec=rvec, x=x)

  loglik.thetaprime <- -S$value
  if (loglik.thetahat - loglik.thetaprime > 0){
    diff <- loglik.thetahat - loglik.thetaprime
  } else diff <- 0

  out <- zquant - sign(theta.hat - theta.prime) * sqrt(2 *
    diff)
}

```

```

r1upIL <- function(theta.prime, theta.hat, loglik.thetahat,
                  nvec, rvec, x, b0.vec, z, zquant){
  # Compute the function -zquant - r1(theta.prime)
  #
  # Input:
  # theta.prime: a value for theta
  # theta.hat: MLE of theta
  # loglik.thetahat: log likelihood value
  # nvec: 6-vector of cluster sizes
  # rvec: 6-vector of responses
  # x: the 6 by 4 design matrix
  #
  # Output:
  # Value of the r1(theta.prime) function.
  #
  # Written by N Ranathunga in May 2020

  heqfun <- functional::Curry(Funheq, z=z, theta.prime=theta.
    prime)

  S <- nloptr::slsqp(x0=b0.vec, fn = Loglik, heq = heqfun,
    nvec=nvec,
    rvec=rvec, x=x)

  loglik.thetaprime <- -S$value
  if (loglik.thetahat - loglik.thetaprime > 0){
    diff <- loglik.thetahat - loglik.thetaprime
  } else diff <- 0

  out <- -zquant - sign(theta.hat - theta.prime) * sqrt(2 *
    diff)
}

```

A.9.4 Grewal (1952) data set used in the R programs

Table A.2: Grewal1952.csv data set used in the R programs

ID	Drug	X	r	n_minus_r	n
1	1	0.18	19	84	103
2	1	0.48	53	67	120
3	1	0.78	83	40	123
4	2	0.18	14	46	60
5	2	0.48	54	56	110
6	2	0.78	81	19	100

Appendix B

Proofs and R programs for Chapter 3

B.1 Application of the Poisson summation formula to understanding the accuracy of the trapezoidal rule

In this appendix, we describe a well-known application of the Poisson summation formula to get an understanding of the accuracy of the trapezoidal rule. According to the Poisson summation formula (page 47, [Papoulis \(1962\)](#)),

$$\sum_{j=-\infty}^{\infty} g(y + jh) = \frac{1}{h} \sum_{j=-\infty}^{\infty} e^{ij\omega_0 y} G(j\omega_0), \quad \text{where } \omega_0 = \frac{2\pi}{h}.$$

In other words,

$$\begin{aligned} h \sum_{j=-\infty}^{\infty} g(y + jh) &= \sum_{j=-\infty}^{\infty} e^{ij\omega_0 y} G(j\omega_0) \\ &= G(0) + \sum_{\substack{j=-\infty \\ j \neq 0}}^{\infty} e^{ij\omega_0 y} G(j\omega_0) \\ &= \int_{-\infty}^{\infty} g(y) dy + \sum_{\substack{j=-\infty \\ j \neq 0}}^{\infty} e^{ij\omega_0 y} G(j\omega_0). \end{aligned} \tag{B.1}$$

Thus

$$\left| h \sum_{j=-\infty}^{\infty} g(y + jh) - \int_{-\infty}^{\infty} g(y) dy \right| \leq \sum_{\substack{j=-\infty \\ j \neq 0}}^{\infty} \left| G\left(\frac{2\pi j}{h}\right) \right|.$$

As noted on page 14 of [Papoulis \(1962\)](#), if $f(y) = g(y - y_0)$ then the Fourier transform of f , denoted by $F(\omega)$, is $F(\omega) = G(\omega) e^{-iy_0\omega}$. It follows from (B.1) that

$$\begin{aligned} h \sum_{j=-\infty}^{\infty} f(y + jh) &= \sum_{j=-\infty}^{\infty} e^{ij\omega_0 y} F(j\omega_0) \\ &= \sum_{j=-\infty}^{\infty} e^{ij\omega_0 y - iy_0\omega} G(j\omega_0) \\ &= \sum_{j=-\infty}^{\infty} e^{i(j\omega_0 y - y_0\omega)} G(j\omega_0). \end{aligned}$$

Thus

$$\left| h \sum_{j=-\infty}^{\infty} f(y + jh) - \int_{-\infty}^{\infty} f(y) dy \right| \leq \sum_{\substack{j=-\infty \\ j \neq 0}}^{\infty} \left| G\left(\frac{2\pi j}{h}\right) \right|.$$

In other words,

$$\left| h \sum_{j=-\infty}^{\infty} g(y + jh - y_0) - \int_{-\infty}^{\infty} g(y) dy \right| \leq \sum_{\substack{j=-\infty \\ j \neq 0}}^{\infty} \left| G\left(\frac{2\pi j}{h}\right) \right|.$$

As noted on page 11 of [Papoulis \(1962\)](#), since $g(y)$ is a real-valued function, $G(-\omega) = G(\omega)$ for all ω . Thus

$$\sum_{\substack{j=-\infty \\ j \neq 0}}^{\infty} \left| G\left(\frac{2\pi j}{h}\right) \right| = 2 \sum_{j=1}^{\infty} \left| G\left(\frac{2\pi j}{h}\right) \right|.$$

Suppose $\delta = y - y_0$. Then

$$\left| h \sum_{j=-\infty}^{\infty} g(jh + \delta) - \int_{-\infty}^{\infty} g(y) dy \right| \leq 2 \sum_{j=1}^{\infty} \left| G\left(\frac{2\pi j}{h}\right) \right|,$$

for all $\delta \in [0, h)$.

B.2 Application of the transformation (2.6) of Mori (1988)

Recall that the transformation (2.6) of Mori (1988) is equal to

$$x(y) = \exp\left(\frac{1}{2}y - e^{-y}\right)$$

and

$$\frac{dx(y)}{dy} = \exp\left(\frac{1}{2}y - e^{-y}\right) \times \left(\frac{1}{2} + e^{-y}\right).$$

Also recall that the pdf f_ν is given by

$$f_\nu(x) = \begin{cases} \tau_\nu x^{\nu-1} \exp(-\nu x^2/2) & \text{for } x > 0 \\ 0 & \text{otherwise,} \end{cases}$$

where

$$\tau_\nu = \frac{\nu^{\nu/2}}{\Gamma(\nu/2) 2^{(\nu/2)-1}}.$$

Now consider

$$\begin{aligned} f_\nu(x(y)) &= \tau_\nu (x(y))^{\nu-1} \exp\left(-\frac{\nu}{2}(x(y))^2\right) \\ &= \tau_\nu \left(\exp\left(\frac{1}{2}y - e^{-y}\right)\right)^{\nu-1} \exp\left(-\frac{\nu}{2} \exp(y - 2e^{-y})\right) \\ &= \tau_\nu \exp\left((\nu-1)\left(\frac{1}{2}y - e^{-y}\right) - \frac{\nu}{2} \exp(y - 2e^{-y})\right). \end{aligned}$$

Therefore

$$\begin{aligned} \psi_\nu(y) &= f_\nu(x(y)) \frac{dx(y)}{dy} \\ &= \tau_\nu \exp\left((\nu-1)\left(\frac{1}{2}y - e^{-y}\right) - \frac{\nu}{2} \exp(y - 2e^{-y}) + \frac{1}{2}y - e^{-y}\right) \\ &\quad \times \left(\frac{1}{2} + e^{-y}\right) \\ &= \tau_\nu \exp\left(\frac{\nu}{2}y - \nu e^{-y} - \frac{\nu}{2} \exp(y - 2e^{-y})\right) \left(\frac{1}{2} + e^{-y}\right). \end{aligned}$$

B.3 Proof of Theorem 3.4.1

The proof in this section is due to Paul Kabaila.

Part (a):

Note that, from Section B.2,

$$\psi_\nu(t) = \tau_\nu \exp\left(\frac{\nu}{2}t - \nu e^{-t} - \frac{\nu}{2}\exp(t - 2e^{-t})\right) \left(\frac{1}{2} + e^{-t}\right).$$

We need to bound $\psi_\nu(t)$ from above for large t . Because $e^{-t} \rightarrow 0$ as $t \rightarrow \infty$, we can approximate $\psi_\nu(t)$ by

$$\frac{\tau_\nu}{2} \exp\left(\frac{\nu}{2}t - \frac{\nu}{2}e^t\right), \quad (\text{B.2})$$

for large t . Since $e^t \rightarrow \infty$ faster than $t \rightarrow \infty$, we can approximate (B.2) by

$$\frac{\tau_\nu}{2} \exp\left(-\frac{\nu}{2}e^t\right) \quad (\text{B.3})$$

for large t .

What we need, however, is an upper bound to $\psi_\nu(t)$ for large t , not an approximation. Let

$$\varphi_\nu^1(t) = \frac{\tau_\nu}{2} \exp\left(-\frac{\nu}{2}\exp\left(\frac{9t}{10}\right)\right).$$

We could have used any number belonging to $(0, 1)$ instead of $9/10$. However, we have chosen this specific number for concreteness.

Note that for all $t > 0$,

$$\begin{aligned} \frac{9}{10}t &< t \\ \Rightarrow \exp\left(\frac{9}{10}t\right) &< \exp(t) \\ \Rightarrow -\frac{\nu}{2}\exp\left(\frac{9}{10}t\right) &> -\frac{\nu}{2}\exp(t) \\ \Rightarrow \exp\left(-\frac{\nu}{2}\exp\left(\frac{9t}{10}\right)\right) &> \exp\left(-\frac{\nu}{2}\exp(t)\right). \end{aligned}$$

Therefore, the function $\varphi_\nu^1(t)$ is a good candidate for an upper bound on $\psi_\nu(t)$ since

$$\varphi_\nu^1(t) \geq \text{(B.3)} \quad \text{for all } t > 0.$$

Now, for all $t > 0$,

$$\begin{aligned} & \frac{\psi_\nu(t)}{\varphi_\nu^1(t)} \\ &= \frac{\tau_\nu \exp\left(\frac{\nu}{2}t - \nu e^{-t} - \frac{\nu}{2} \exp(y - 2e^{-t})\right) \times \left(\frac{1}{2} + e^{-t}\right)}{\frac{\tau_\nu}{2} \exp\left(-\frac{\nu}{2} \exp\left(\frac{9t}{10}\right)\right)} \\ &= 2 \exp\left(\frac{\nu}{2}t - \nu e^{-t} - \frac{\nu}{2} \exp(t - 2e^{-t}) + \frac{\nu}{2} \exp\left(\frac{9t}{10}\right)\right) \times \left(\frac{1}{2} + e^{-t}\right) \\ &< 3 \exp\left(\frac{\nu}{2}t - \nu e^{-t} - \frac{\nu}{2} \left(\exp(t - 2e^{-t}) - \exp\left(\frac{9}{10}t\right)\right)\right), \end{aligned} \quad (\text{B.4})$$

since $e^{-t} < 1$ for all $t > 0$.

Consider

$$\begin{aligned} & \exp(t - 2e^{-t}) - \exp\left(\frac{9}{10}t\right) \\ &= \exp\left(\frac{9}{10}t\right) \left(\frac{\exp(t - 2e^{-t})}{\exp\left(\frac{9}{10}t\right)} - 1\right) \\ &= \exp\left(\frac{9}{10}t\right) \left(\exp\left(\frac{1}{10}t - 2e^{-t}\right) - 1\right). \end{aligned} \quad (\text{B.5})$$

Since

$$\exp\left(\frac{1}{10}t - 2e^{-t}\right) \rightarrow \infty \quad \text{as } t \rightarrow \infty,$$

there exists $t' < \infty$ such that

$$\exp\left(\frac{1}{10}t - 2e^{-t}\right) \geq 1 \quad \text{for all } t \geq t'.$$

In other words,

$$\text{(B.5)} \geq \exp\left(\frac{9}{10}t\right) \quad \text{for all } t \geq t'.$$

Hence

$$\begin{aligned}
(\text{B.4}) &\leq 3 \exp \left(\frac{\nu}{2} t - \nu e^{-t} - \frac{\nu}{2} \exp \left(\frac{9}{10} t \right) \right) \\
&= 3 \exp \left(\frac{\nu}{2} \left(t - \exp \left(\frac{9}{10} t \right) \right) - \nu e^{-t} \right) \tag{B.6}
\end{aligned}$$

for all $t \geq t'$. Since $\exp(9t/10) \rightarrow \infty$ faster than $t \rightarrow \infty$ and $\nu e^{-t} \rightarrow 0$ as $t \rightarrow \infty$, (B.6) $\rightarrow 0$, as $t \rightarrow \infty$.

To summarise : $\psi_\nu(t)/\varphi_\nu^1(t) \rightarrow 0$ as $t \rightarrow \infty$. Hence there exists t_1 such that $\psi_\nu(t)/\varphi_\nu^1(t) \leq 1$ for all $t \geq t_1$. Therefore, $\psi_\nu(t) \leq \varphi_\nu^1(t)$ for all $t \geq t_1$. In other words, there exist $t_1 < \infty$ such that

$$\psi_\nu(t) \leq \frac{\tau_\nu}{2} \exp \left(-\frac{\nu}{2} \exp \left(\frac{9}{10} t \right) \right) \quad \text{for all } t \geq t_1.$$

Part (b):

We also need to bound $\psi_\nu(t)$ from above for t negative and $|t|$ large. Now, $e^{-t} \rightarrow \infty$ as $t \rightarrow -\infty$ and $\exp(t - 2e^{-t}) \rightarrow 0$ as $t \rightarrow -\infty$. Hence, for t negative and $|t|$ large, we approximate $\psi_\nu(t)$ by

$$\tau_\nu \exp \left(\frac{\nu}{2} t - \nu e^{-t} \right) \times \exp(-t) = \tau_\nu \exp \left(\left(\frac{\nu}{2} - 1 \right) t - \nu e^{-t} \right). \tag{B.7}$$

Now $-\nu e^{-t} \rightarrow -\infty$ faster than $t \rightarrow -\infty$. Hence we can approximate (B.7) by $\tau_\nu \exp(-\nu e^{-t})$.

What we need, however, is an upper bound to $\psi_\nu(t)$ for t negative and $|t|$ large, not an approximation. Let

$$\varphi_\nu^2(t) = \tau_\nu \exp \left(-\nu \exp \left(-\frac{9}{10} t \right) \right).$$

Now,

$$\frac{\psi_\nu(t)}{\varphi_\nu^2(t)} = \frac{\tau_\nu \exp \left(\frac{\nu}{2} t - \nu e^{-t} - \frac{\nu}{2} \exp \left(t - 2e^{-t} \right) \right) \times \left(\frac{1}{2} + e^{-t} \right)}{\tau_\nu \exp \left(-\nu \exp \left(-\frac{9}{10} t \right) \right)}$$

$$\begin{aligned}
&= \exp \left(\frac{\nu}{2}t - \nu e^{-t} - \frac{\nu}{2} \exp(y - 2e^{-t}) + \nu \exp \left(\frac{-9t}{10} \right) \right) \times \left(\frac{1}{2} + e^{-t} \right) \\
&= \frac{1}{2} \exp \left(\frac{\nu}{2}t - \frac{\nu}{2} \exp(t - 2e^{-t}) + \nu \left(\exp \left(\frac{-9t}{10} \right) - \exp(-t) \right) \right) \\
&\quad + \exp \left(\frac{\nu}{2}t - t - \frac{\nu}{2} \exp(t - 2e^{-t}) + \nu \left(\exp \left(\frac{-9t}{10} \right) - \exp(-t) \right) \right).
\end{aligned} \tag{B.8}$$

Consider

$$\begin{aligned}
&\exp \left(\frac{-9t}{10} \right) - \exp(-t) \\
&= \exp \left(\frac{-9t}{10} \right) \left(1 - \exp \left(-t + \frac{9}{10}t \right) \right) \\
&= \exp \left(\frac{-9t}{10} \right) \left(1 - \exp \left(-\frac{1}{10}t \right) \right).
\end{aligned} \tag{B.9}$$

Now, $1 - \exp(-t/10) \rightarrow -\infty$ as $t \rightarrow -\infty$. Hence, there exists $\tilde{t} < \infty$ such that $1 - \exp(-t/10) \leq -1$ for all $t \geq \tilde{t}$. Thus (B.9) $\leq -\exp(-9t/10)$ for all $t \geq \tilde{t}$. Therefore,

$$\begin{aligned}
(\text{B.8}) &\leq \frac{1}{2} \exp \left(\frac{\nu}{2}t - \frac{\nu}{2} \exp(t - 2e^{-t}) - \nu \exp \left(\frac{-9t}{10} \right) \right) \\
&\quad + \exp \left(\frac{\nu - 2}{2}t - \frac{\nu}{2} \exp(t - 2e^{-t}) - \nu \exp \left(\frac{-9t}{10} \right) \right).
\end{aligned} \tag{B.10}$$

Now, $-t \rightarrow \infty$ more slowly than $\exp(-9t/10)$ as $t \rightarrow -\infty$. Also $(\nu/2) \exp(t - 2e^{-t}) \rightarrow 0$ as $t \rightarrow -\infty$. Thus (B.10) $\rightarrow 0$ as $t \rightarrow -\infty$.

To summarise : $\psi_\nu(t)/\varphi_\nu^2(t) \rightarrow 0$ as $t \rightarrow -\infty$. Hence there exists $t_2 > -\infty$ such that $\psi_\nu(t)/\varphi_\nu^2(t) \leq 1$ for all $t \leq t_2$. Therefore, $\psi_\nu(t) \leq \varphi_\nu^2(t)$ for all $t \leq t_2$. In other words, there exist $t_2 > -\infty$ such that

$$\psi_\nu(t) \leq \tau_\nu \exp \left(-\nu \exp \left(-\frac{9}{10}t \right) \right) \quad \text{for all } t \leq t_2.$$

Part (c):

To bound the trimming error from above, we will need to bound

$$\int_y^\infty \psi_\nu(t) dt \quad (\text{B.11})$$

from above. We know that

$$\psi_\nu(t) \leq \frac{\tau_\nu}{2} \exp\left(-\frac{\nu}{2} \exp\left(\frac{9t}{10}\right)\right) \quad \text{for all } t \geq t_1.$$

Thus

$$\begin{aligned} (\text{B.11}) &\leq \int_y^\infty \frac{\tau_\nu}{2} \exp\left(-\frac{\nu}{2} \exp\left(\frac{9t}{10}\right)\right) dt \quad \text{for all } y \geq t_1 \\ &= \frac{\tau_\nu}{2} \int_y^\infty \exp\left(-\frac{\nu}{2} \exp\left(\frac{9t}{10}\right)\right) dt \quad \text{for all } y \geq t_1. \end{aligned}$$

Change the variable of integration to $z = \exp(9t/10)$, so that

$$\frac{dz}{dt} = \frac{9}{10} \exp\left(\frac{9}{10}t\right)$$

and

$$\begin{aligned} \int_y^\infty \exp\left(-\frac{\nu}{2} \exp\left(\frac{9t}{10}\right)\right) dt &= \frac{10}{9} \int_{\exp(9y/10)}^\infty \exp\left(-\frac{\nu}{2} z\right) \frac{1}{z} dz \\ &\leq \frac{10}{9} \frac{1}{\exp(9y/10)} \int_{\exp(9y/10)}^\infty \exp\left(-\frac{\nu}{2} z\right) dz \\ &= \frac{10}{9} \exp\left(-\frac{9}{10}y\right) \frac{2}{\nu} \exp\left(-\frac{\nu}{2} \exp\left(\frac{9y}{10}\right)\right) \\ &\leq \frac{20}{9\nu} \exp\left(-\frac{\nu}{2} \exp\left(\frac{9y}{10}\right)\right) \quad \text{for all } y \geq t_1. \end{aligned}$$

We have assumed here that $t_1 \geq 0$. Therefore

$$\int_y^\infty \psi_\nu(t) dt \leq \frac{10 \tau_\nu}{9 \nu} \exp\left(-\frac{\nu}{2} \exp\left(\frac{9}{10}y\right)\right) \quad \text{for all } y \geq t_1.$$

We also need to bound

$$\int_{-\infty}^y \psi_\nu(t) dt \quad (\text{B.12})$$

from above. We know that

$$\psi_\nu(t) \leq \tau_\nu \exp\left(-\nu \exp\left(\frac{-9t}{10}\right)\right) \quad \text{for all } t \leq t_2.$$

Thus

$$\begin{aligned} \text{(B.12)} &\leq \int_{-\infty}^y \tau_\nu \exp\left(-\nu \exp\left(\frac{-9t}{10}\right)\right) dt \quad \text{for all } y \leq t_2 \\ &= \tau_\nu \int_{-\infty}^y \exp\left(-\nu \exp\left(\frac{-9t}{10}\right)\right) dt \quad \text{for all } y \leq t_2. \end{aligned} \quad \text{(B.13)}$$

Change the variable of integration to $z = -t$, so that

$$\begin{aligned} \text{(B.13)} &= -\tau_\nu \int_{\infty}^{-y} \exp\left(-\nu \exp\left(\frac{9z}{10}\right)\right) dz \\ &= \tau_\nu \int_{-y}^{\infty} \exp\left(-\nu \exp\left(\frac{9z}{10}\right)\right) dz \quad \text{for all } -y \geq -t_2. \end{aligned}$$

Now change the variable of integration to $w = \exp(9z/10)$, so that

$$\frac{dw}{dz} = \frac{9}{10} \exp\left(\frac{9}{10}z\right)$$

and

$$\begin{aligned} &\int_{-y}^{\infty} \exp\left(-\nu \exp\left(\frac{9z}{10}\right)\right) dz \\ &\leq \frac{10}{9} \frac{1}{\exp(-9y/10)} \int_{\exp(-9y/10)}^{\infty} \exp(-\nu w) dw \\ &= \frac{10}{9} \exp\left(\frac{9}{10}y\right) \frac{1}{\nu} \exp\left(-\nu \exp\left(\frac{-9y}{10}\right)\right) \\ &\leq \frac{10}{9\nu} \exp\left(-\nu \exp\left(\frac{-9y}{10}\right)\right) \quad \text{for all } y \leq t_2. \end{aligned}$$

Therefore

$$\int_{-\infty}^y \psi_\nu(t) dt \leq \frac{10}{9\nu} \tau_\nu \exp\left(-\nu \exp\left(-\frac{9}{10}y\right)\right) \quad \text{for all } y \leq t_2.$$

To summarise :

$$\begin{aligned} \int_y^\infty \psi_\nu(t) dt &\leq \frac{10}{9\nu} \tau_\nu \exp\left(-\frac{\nu}{2} \exp\left(\frac{9y}{10}\right)\right) \quad \text{for all } y \geq t_1 \\ \int_{-\infty}^y \psi_\nu(t) dt &\leq \frac{10}{9\nu} \tau_\nu \exp\left(-\nu \exp\left(\frac{-9y}{10}\right)\right) \quad \text{for all } y \leq t_2. \end{aligned} \quad (\text{B.14})$$

Part (d):

By the proof of Lemma 3.4.1, the *trimming error* for the iteration number k , is bounded above by

$$\int_{y_{u_0} + kh_0}^\infty \psi_\nu(t) dt + \int_{-\infty}^{y_{l_0} - kh_0} \psi_\nu(t) dt. \quad (\text{B.15})$$

It follows from (B.14) that for all sufficiently large iteration numbers k ,

$$\begin{aligned} \int_{y_{u_0} + kh_0}^\infty \psi_\nu(t) dt &\leq \frac{10}{9\nu} \tau_\nu \exp\left(-\frac{\nu}{2} \exp\left(\frac{9}{10}(y_{u_0} + kh_0)\right)\right) \\ &= \frac{10}{9\nu} \tau_\nu \exp\left(-\frac{\nu}{2} \exp\left(\frac{9}{10}y_{u_0}\right) \exp\left(\frac{9h_0}{10}k\right)\right). \end{aligned} \quad (\text{B.16})$$

It follows from (B.14) that for all sufficiently large iteration numbers k ,

$$\begin{aligned} \int_{-\infty}^{y_{l_0} - kh_0} \psi_\nu(t) dt &\leq \frac{10}{9\nu} \tau_\nu \exp\left(-\nu \exp\left(-\frac{9}{10}(y_{l_0} - kh_0)\right)\right) \\ &= \frac{10}{9\nu} \tau_\nu \exp\left(-\nu \exp\left(-\frac{9}{10}y_{l_0}\right) \exp\left(\frac{9h_0}{10}k\right)\right). \end{aligned} \quad (\text{B.17})$$

Now we express the term $\exp(9h_0k/10)$ in (B.16) and (B.17) in the form 2^{ck} . Thus c satisfies

$$\exp\left(\frac{9h_0}{10}k\right) = 2^{ck}$$

is denoted by c_T . Therefore, for all sufficiently large iteration numbers k , (B.15) is bounded above by

$$\frac{10\tau_\nu}{9\nu} \left(\exp\left(-\frac{\nu}{2} \exp\left(\frac{9}{10}y_{u_0}\right) 2^{c_T k}\right) + \exp\left(-\nu \exp\left(\frac{-9}{10}y_{l_0}\right) 2^{c_T k}\right) \right),$$

where $c_T = 9h_0/(10 \log_e(2))$.

Part (e):

According to the Assumption FT,

$$|G(\omega)| \leq c_4 \exp(-c_{FT}|\omega|) \quad \text{for all } \omega \in \mathbb{R}.$$

Note that from (3.3), the *discretization error* is bounded above by

$$\begin{aligned} & 2 \sum_{j=1}^{\infty} \left| G\left(\frac{2\pi j}{h}\right) \right| \\ & \leq 2 c_4 \sum_{j=1}^{\infty} \exp\left(-c_{FT} \frac{2\pi j}{h}\right) \\ & = 2 c_4 \sum_{j=1}^{\infty} \left(\exp\left(-c_{FT} \frac{2\pi}{h}\right) \right)^j \\ & = 2 c_4 \sum_{j=1}^{\infty} a^j, \quad \text{where } a = \exp\left(-c_{FT} \frac{2\pi}{h}\right), \\ & = 2 c_4 \frac{a}{1-a} \quad \text{provided } a < 1 \text{ i.e. } h \text{ is sufficiently small,} \\ & \leq 3 c_4 a \quad \text{provided } h \text{ is sufficiently small,} \\ & = 3 c_4 \exp\left(-\frac{2\pi c_{FT}}{h}\right) \quad \text{for all sufficiently small } h. \end{aligned}$$

Note that, at iteration number k ,

$$y_u - y_l = d_0 + 2h_0 k \quad \text{and} \quad h = \frac{h_0}{2^k},$$

and so

$$n = \frac{d_0 + 2h_0 k}{h_0/2^k} = \left(\frac{d_0}{h_0} + 2k \right) 2^k = (n_0 + 2k) 2^k.$$

Therefore, for all sufficiently large iteration numbers k , the *discretization error* is bounded above by

$$3c_4 \exp\left(-\left(\frac{2\pi c_{FT}}{h_0}\right) 2^k\right).$$

□

B.4 An expression for a polynomial p of degree u using the binomial theorem

Consider a polynomial

$$p(y) = \sum_{j=0}^u b_j y^j$$

in y of degree u . We can express $p(y)$ differently, as follows.

$$\begin{aligned} p(y) &= b_0 + \sum_{j=1}^u b_j y^j \\ &= b_0 + \sum_{j=1}^u b_j (1 - (1 - y))^j. \end{aligned} \tag{B.18}$$

Now, by the binomial theorem,

$$(1 - (1 - y))^j = 1 + \binom{j}{1} (-(1 - y)) + \binom{j}{2} (-(1 - y))^2 + \cdots + (-(1 - y))^j.$$

Therefore (B.18) can be written as

$$\begin{aligned} p(y) &= b_0 + u + \sum_{j=1}^u c_j (1 - y)^j \\ &= a_0 + \sum_{j=1}^u c_j (1 - y)^j, \end{aligned}$$

where $a_0 = b_0 + u$. Now let $a_j = -c_j$ for $j = 1, \dots, u$. Then

$$p(y) = a_0 - \sum_{j=1}^u a_j (1 - y)^j.$$

B.5 Computation of $F_\nu^{-1}(y)$

F_ν denotes the cdf of a random variable with the same distribution as $R/\nu^{1/2}$, where R has a χ_ν distribution. In this section, we show how to compute the inverse of this cdf, using either (a) the inverse cdf of a χ_ν^2 distribution or (b) the inverse cdf of a χ_ν distribution.

Firstly, we can compute $F_\nu^{-1}(y)$ using $F_\nu^{-1}(y) = (Q_\nu^{-1}(y)/\nu)^{1/2}$, where Q_ν denotes the χ_ν^2 cdf. We prove this result as follows. Note that

$$\begin{aligned} F_\nu(x) &= P\left(\frac{R}{\nu^{1/2}} \leq x\right) \\ &= P(R^2 \leq \nu x^2) \\ &= Q_\nu(\nu x^2). \end{aligned}$$

Now $y = F_\nu(x)$ is equivalent to $x = F_\nu^{-1}(y)$. But $y = F_\nu(x)$ is also equivalent to

$$\begin{aligned} y &= Q_\nu(\nu x^2) \\ \iff \nu x^2 &= Q_\nu^{-1}(y) \\ \iff x &= (Q_\nu^{-1}(y)/\nu)^{1/2}, \quad \text{since } x > 0 \\ \iff F_\nu^{-1}(y) &= (Q_\nu^{-1}(y)/\nu)^{1/2}. \end{aligned}$$

Secondly, we can compute $F_\nu^{-1}(y)$ using $F_\nu^{-1}(y) = F_R^{-1}(y)/\nu^{1/2}$, where F_R denotes the χ_ν cdf of R . We prove this result as follows. Note that

$$\begin{aligned} F_\nu(x) &= P\left(\frac{R}{\nu^{1/2}} \leq x\right) \\ &= P(R \leq \nu^{1/2} x) \\ &= F_R(\nu^{1/2} x). \end{aligned}$$

Now $y = F_\nu(x)$ is equivalent to $x = F_\nu^{-1}(y)$. But $y = F_\nu(x)$ is equivalent to

$$\begin{aligned}
y &= F_R(\nu^{1/2} x) \\
\iff F_R^{-1}(y) &= \nu^{1/2} x \\
\iff x &= F_R^{-1}(y)/\nu^{1/2} \\
\iff F_\nu^{-1}(y) &= F_R^{-1}(y)/\nu^{1/2}.
\end{aligned}$$

B.6 Proof of the result (3.19)

In this section, we prove the result (3.19), given in Section 3.7. Note that

$$f_\kappa(x) = \begin{cases} \frac{\kappa^{\kappa/2}}{\Gamma(\kappa/2) 2^{(\kappa/2)-1}} x^{\kappa-1} \exp(-\kappa x^2/2) & \text{for } x > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Thus

$$x^\xi f_\kappa(x) = \frac{\kappa^{\kappa/2}}{\Gamma(\kappa/2) 2^{(\kappa/2)-1}} x^{(\kappa+\xi)-1} \exp(-\kappa x^2/2) \quad \text{for } x > 0.$$

Now, we want to convert $\exp(-\kappa x^2/2)$ in to $\exp(-(\kappa+\xi)y^2/2)$ by changing the variable of integration from x to y . Therefore, we want

$$\begin{aligned}
\kappa x^2 &= (\kappa + \xi) y^2 \\
\iff y^2 &= \left(\frac{\kappa}{\kappa + \xi} \right) x^2 \\
\iff y &= \left(\frac{\kappa}{\kappa + \xi} \right)^{1/2} x = c(\kappa, \xi) x.
\end{aligned}$$

Thus

$$\begin{aligned}
&\int_0^\infty \lambda(x) x^\xi f_\kappa(x) dx \\
&= \int_0^\infty \lambda\left(\frac{y}{c(\kappa, \xi)}\right) \frac{\kappa^{\kappa/2}}{\Gamma(\kappa/2) 2^{(\kappa/2)-1}} \left(\frac{y}{c(\kappa, \xi)}\right)^{(\kappa+\xi)-1} \times \\
&\quad \exp(-(\kappa + \xi) y^2/2) \frac{dy}{c(\kappa, \xi)}
\end{aligned}$$

$$\begin{aligned}
&= \frac{\kappa^{\kappa/2}}{\Gamma(\kappa/2) 2^{(\kappa/2)-1}} \left(\frac{1}{c(\kappa, \xi)} \right)^{\kappa+\xi} \frac{\Gamma((\kappa+\xi)/2) 2^{((\kappa+\xi)/2)-1}}{(\kappa+\xi)^{(\kappa+\xi)/2}} \times \\
&\quad \int_0^\infty \lambda \left(\frac{y}{c(\kappa, \xi)} \right) \frac{(\kappa+\xi)^{(\kappa+\xi)/2}}{\Gamma((\kappa+\xi)/2) 2^{((\kappa+\xi)/2)-1}} \kappa^{(\kappa+\xi)-1} \exp \left(-(\kappa+\xi) y^2/2 \right) dy \\
&= \frac{\kappa^{\kappa/2}}{\Gamma(\kappa/2)} \frac{\Gamma((\kappa+\xi)/2)}{(\kappa+\xi)^{(\kappa+\xi)/2}} (c(\kappa, \xi))^{-(\kappa+\xi)} 2^{\xi/2} \int_0^\infty \lambda \left(\frac{y}{c(\kappa, \xi)} \right) f_{\kappa+\xi}(y) dy.
\end{aligned}$$

Note that

$$\begin{aligned}
\frac{\kappa^{\kappa/2}}{(\kappa+\xi)^{(\kappa+\xi)/2}} (c(\kappa, \xi))^{-(\kappa+\xi)} &= \frac{\kappa^{\kappa/2}}{(\kappa+\xi)^{(\kappa+\xi)/2}} \left(\frac{\kappa}{\kappa+\xi} \right)^{-(\kappa+\xi)/2} \\
&= \frac{\kappa^{\kappa/2}}{(\kappa+\xi)^{(\kappa+\xi)/2}} \frac{(\kappa+\xi)^{(\kappa+\xi)/2}}{\kappa^{(\kappa+\xi)/2}} \\
&= \kappa^{-\xi/2}
\end{aligned}$$

and

$$2^{\xi/2} \kappa^{-\xi/2} = \left(\frac{2}{\kappa} \right)^{\xi/2}.$$

Thus

$$\begin{aligned}
&\int_0^\infty \lambda(x) x^\xi f_\kappa(x) dx \\
&= \left(\frac{2}{\kappa} \right)^{\xi/2} \frac{\Gamma((\kappa+\xi)/2)}{\Gamma(\kappa/2)} \int_0^\infty \lambda \left(\frac{y}{c(\kappa, \xi)} \right) f_{\kappa+\xi}(y) dy \\
&= \left(\frac{2}{\kappa} \right)^{\xi/2} \frac{\Gamma((\kappa+\xi)/2)}{\Gamma(\kappa/2)} \int_0^\infty a(y) f_{\kappa+\xi}(y) dy \tag{B.19}
\end{aligned}$$

where $a(y) = \lambda(y/c(\kappa, \xi))$. □

B.7 R programs for the computation of the integral (3.1)

In this appendix, we list the R programs for computing the integrals of the form

$$\int_0^{\infty} a(x) f_{\nu}(x) dx,$$

defined in Section 3.2, by applying the methods (a) the transformation (2.6) of Mori (1988), followed by the application of the trapezoidal rule, (b) Gauss Legendre quadrature, (c) Generalized Gauss Laguerre quadrature and (e) inverse cdf method followed by Gauss Legendre quadrature.

B.7.1 The transformation (2.6) of Mori (1988), followed by the application of the trapezoidal rule

```
ApproxError <- function(nu.vec, alpha.vec, yl.vec,
                        yu.vec, d.vec, n = 5){
  # Compute the approximation error when approximate
  # the integral,  $\int_0^{\infty} a(x) f_{\nu}(x) dx$ 
  # where  $a(x) = 2 * \Phi(t_{\{\nu, 1-\alpha/2\}} x) - 1$ 
  # and  $f_{\nu}$  is the pdf of a random variable with the
  # same distribution as  $R/\sqrt{R}$  where  $R \sim \chi(\nu)$ 
  # distribution, by the trapezoidal rule.
  #
  # Input:
  # nu.vec: a vector of degrees of freedom
  # alpha.vec: a 3-vector of alpha values where
  #           1-alpha is the nominal coverage
  # n: number of evaluations of the integrand
  # d.vec: a vector of d's with same length as nu.vec
  #        where  $d = n*h$  and  $h$  is the step length
  # yl.vec: a vector of yl's with same length as nu.vec
  #        where the first evaluation of the integrand is at
  #        yl
  # yu.vec: a vector  $yl.vec + d.vec$ 
  #
  # Output:
  # A matrix with 3 rows and number of columns equal to
  # the length of nu.vec.
  #
  # Written by N Ranathunga in November 2019
  #
  # Set up vectors to store the results
  approxerror <- matrix(0, nrow=length(alpha.vec),
                        ncol=length(nu.vec))
```

```

for (i in 1:length(alpha.vec)){
  alpha <- alpha.vec[i]
  for(j in 1:length(nu.vec)){
    nu <- nu.vec[j]
    y1 <- y1.vec[j]
    yu <- yu.vec[j]
    d <- d.vec[j]

    #n = 5
    h <- d/(n-1)
    I1 <- approx_sum(y1, yu, h, nu, alpha)

    #n = 9
    h2 <- h/2
    I2 <- approx_sum(y1, yu, h2, nu, alpha)

    #n = 17
    h3 <- h2/2
    I3 <- approx_sum(y1, yu, h3, nu, alpha)

    #n = 33
    h4 <- h3/2
    I4 <- approx_sum(y1, yu, h4, nu, alpha)

    #n = 65
    h5 <- h4/2
    I5 <- approx_sum(y1, yu, h5, nu, alpha)

    if (nu == 1){
      approxerror[i, j] <- I5 - (1 - alpha)
    } else {
      approxerror[i, j] <- I4 - (1 - alpha)
    }
  }
}

out <- approxerror
}

```

```

ApproxSum <- function(yl, yu, h, nu, alpha){
  # Compute the integral,  $\int_0^{\infty} a(x) f_{\nu}(x) dx$ 
  # where  $a(x) = 2 * \Phi(t_{\nu, 1-\alpha/2} x) - 1$ 
  # and  $f_{\nu}$  is the pdf of a random variable with the
  # same distribution as  $R/\sqrt{R}$  where  $R \sim \chi(\nu)$ 
  # distribution, by applying the trapezoidal rule.
  #
  # Input:
  # nu: degrees of freedom
  # alpha: 1-alpha is the nominal coverage
  # h: step length
  # yl: the first evaluation of the integrand is at yl
  # yu: the last evaluation of the integrand is at yu
  #
  # Output:
  # A value for the integrand  $a(x) f_{\nu}(x)$ .
  #
  # Written by N Ranathunga in November 2019

  yvec <- seq(yl, yu, by = h)
  xvec <- transf(yvec)
  a.xvec <- funct_a(xvec, nu, alpha)
  out <- h * kahanSum(a.xvec * FuncPsine(yvec, nu))
}

```

```

transf <- function(y){
  # This function applies the transformation (2.6)
  # of Mori(1988) to the variable y
  #
  # Inputs:
  # y = a value or vector
  #
  # Written by N. Ranathunga in November 2019

  out <- exp((y/2) - exp(-y))
}

```

```

funct_a <- function(x.vec, nu, alpha){
  # Compute the value of
  #  $a(x(y)) = 2 * \Phi(t_{\nu, 1-\alpha/2} x(y)) - 1$ 
  # where  $x(y) = \exp(y/2 - \exp(-y))$  and  $\Phi$  is
  #  $N(0, 1)$  cdf.
  #
  # Input:
  # x.vec: a vector of  $x(y)$  values
  # nu: degrees of freedom
  # alpha: 1-alpha is the nominal coverage
  #
  # Written by N Ranathunga in November 2019

  tquant <- qt(1 - alpha/2, nu)
  out <- 2 * pnorm(tquant * x.vec) - 1
}

```

```

FuncPsineu <- function(y, nu){
  # Compute the value of
  # f_nu(x(y)) * dx(y)/dy where x(y) = exp(y/2 - exp(-y))
  # and f_nu is the pdf of a random variable with the
  # same distribution as R/sqrt(R) where R ~ chi(nu)
  # distribution.
  #
  # Input:
  # y: a value or a vector where the function
  #     needs to be evaluated
  # nu: degrees of freedom
  #
  # Written by N Ranathunga in November 2019

  const <- exp( (nu/2) * log(nu) - lgamma(nu/2) - ((nu/2) -
    1) * log(2) )
  tmp1 <- exp(-y)
  term1 <- exp(nu*y/2 - nu*tmp1 - (nu/2)*exp(y - 2*tmp1))
  term2 <- 1/2 + tmp1

  out <- const * term1 * term2
}

```

```

Find_yvec <- function(eps.final, nu.vec){
  # Compute the vectors of d, yl and yu for a
  # given epsilon and a vector of degrees of freedom.
  #
  # Input:
  # eps.final: 10e-3 * eps where eps is the upper
  #             bound of the approximation error
  # nu.vec: a vector of degrees of freedom
  #
  # Output:
  # A list containing vectors of d, yl and yu
  # of same length.
  #
  # Written by N Ranathunga in November 2019

  # Set up vectors to store the results
  d.vec <- rep(0, length.out=length(nu.vec))
  yl.vec <- rep(0, length.out=length(nu.vec))
  yu.vec <- rep(0, length.out=length(nu.vec))

  for (i in 1:length(nu.vec)){
    nu <- nu.vec[i]
    temp <- uniroot(MinUpBndTrErrMinEpsFin, nu=nu,
                     eps.final=eps.final,
                     interval = c(0, 10), extendInt="yes")
    d.vec[i] <- temp$root
    yl.vec[i] <- optimize(UpBoundTrimError, d=d.vec[i],
                          nu=nu, interval=c(-5, 1))$minimum
    yu.vec[i] <- yl.vec[i] + d.vec[i]
  }

  out <- list(dvec=d.vec, ylvec=yl.vec, yuvec=yu.vec)
}

```



```

MinUpBndTrErrMinEpsFin <- function(d, nu, eps.final){
  # This function minimize the upper bound of the
  # trimming error and subtract that value from the
  # (eps.final)
  #
  # Input:
  # d: n*h where n denotes the number of integrand
  #     evaluations and h denotes the step length
  # nu: degrees of freedom
  # eps.final: 10e-3 * eps where eps is the upper
  #             bound of the approximation error
  #
  # Written by N Ranathunga in November 2019

  temp <- optimize(UpBoundTrimError, d=d, nu=nu, interval=c
    (-4, 4))
  out <- temp$objective - eps.final
}

```

```

UpBoundTrimError <- function(y, d, nu){
  # This function computes the upper bound of the
  # trimming error when approximate the outer integral by
  # a finite sum.
  #
  # Input:
  # y: location of the function evaluations
  # d: n*h where n denotes the number of evaluations
  #     and h denotes the step length
  # nu: degrees of freedom
  #
  # Written by N Ranathunga in November 2019

  x1 <- transf(y)
  x2 <- transf(y + d)
  term1 <- nu * x1^2
  term2 <- nu * x2^2
  out <- pchisq(term1, df=nu) + 1 - pchisq(term2, df=nu)
}

```

B.7.2 Gauss Legendre quadrature method

```
GaussLegendreError <- function(alpha.vec, nu.vec, yl.vec, yu.
vec){
  # Compute the approximation error when approximate
  # the integral,  $\int_0^{\infty} a(x) f_{\text{nu}}(x) dx$ 
  # where  $a(x) = 2 * \Phi(t_{\text{nu}}, 1-\alpha/2) x - 1$ 
  # and  $f_{\text{nu}}$  is the pdf of a random variable with the
  # same distribution as  $R/\sqrt{R}$  where  $R \sim \chi(\text{nu})$ 
  # distribution, by the Gauss Legendre quadrature rule.
  #
  # Input:
  # nu.vec: a vector of degrees of freedom
  # alpha.vec: a 3-vector of alpha values where
  #           1-alpha is the nominal coverage
  # yl.vec: a vector of yl's with same length as nu.vec
  #           where the first evaluation of the integrand is at
  #           yl
  # yu.vec: a vector yl.vec + d.vec
  #
  # Output:
  # A matrix with 3 rows and number of columns equal to
  # the length of nu.vec.
  #
  # Written by N Ranathunga in November 2019

  # Set up vectors to store the results
  comp.int <- matrix(0, nrow=length(alpha.vec),
                    ncol=length(nu.vec))
  error <- matrix(0, nrow=length(alpha.vec),
                 ncol=length(nu.vec))

  for (i in c(1:length(alpha.vec))) {
    for (j in c(1:length(nu.vec))) {

      alpha <- alpha.vec[i]
      nu <- nu.vec[j]
      yl <- yl.vec[j]
      yu <- yu.vec[j]

      if (nu == 1){
        n <- 65
      } else n <- 33

      tquant <- qt(1 - alpha/2, nu)
      comp.int[i,j] <- LegendreQuadRule(nu, n, alpha, yl, yu)
      error[i, j] <- comp.int[i,j] - (1 - alpha)

    }
  }
  out <- error
}
```

```

LegendreQuadRule <- function(nu, n, alpha, y1, yu){
  # Compute the integral,  $\int_0^{\infty} a(x) f_{\text{nu}}(x) dx$ 
  # where  $a(x) = 2 * \Phi(t_{\{\text{nu}, 1-\alpha/2\}} x) - 1$ 
  # and  $f_{\text{nu}}$  is the pdf of a random variable with the
  # same distribution as  $R/\sqrt{R}$  where  $R \sim \chi(\text{nu})$ 
  # distribution, by applying the Gauss Legendre
  # quadrature rule.
  #
  # Input:
  # nu: degrees of freedom
  # n: number of evaluations of the integrand
  # alpha: 1-alpha is the nominal coverage
  # y1: the first evaluation of the integrand is at y1
  # yu: the last evaluation of the integrand is at yu
  #
  # Output:
  # A value for the integrand  $a(x) f_{\text{nu}}(x)$ .
  #
  # Written by N. Ranathunga in April 2020

  quad.rule <- gauss.quad(n, kind = "legendre")

  z.i <- quad.rule$nodes
  w.i <- quad.rule$weights

  y.i <- ( (yu - y1) * z.i / 2 ) + ( (yu + y1) / 2 )
  func.i <- FuncPsine_w_a(yvec=y.i, nu=nu, alpha=alpha)
  weighted.val <- w.i * func.i

  out <- ((yu - y1)/2) * sum(weighted.val)
}

```

```

FuncPsine_w_a <- function(yvec, nu, alpha){
  # Compute the value of  $a(x(y)) \psi_{\text{nu}}(y)$  where
  #  $a(x(y)) = 2 * \Phi(t_{\{\text{nu}, 1-\alpha/2\}} x(y)) - 1$ 
  # and  $\psi_{\text{nu}}(y) = f_{\text{nu}}(x(y)) * dx(y)/dy$ 
  # where  $x(y) = \exp(y/2 - \exp(-y))$ 
  # and  $f_{\text{nu}}$  is the pdf of a random variable with the
  # same distribution as  $R/\sqrt{R}$  where  $R \sim \chi(\text{nu})$ 
  # distribution.
  #
  # Input:
  # yvec: a vector of y values
  # nu: degrees of freedom
  # alpha: 1-alpha is the nominal coverage
  #
  # Written by N Ranathunga in November 2019

  xvec <- transf(yvec)
  a.xvec <- funct_a(xvec, nu, alpha)

  out <- a.xvec * FuncPsine_w(yvec, nu)
}

```

B.7.3 Generalized Gauss Laguerre quadrature method

```
GaussLaguerreError <- function(alpha.vec, nu.vec){
  # Compute the approximation error when approximate
  # the integral,  $\int_0^{\infty} a(x) f_{\text{nu}}(x) dx$ 
  # where  $a(x) = 2 * \Phi(t_{\text{nu}}, 1-\alpha/2} x) - 1$ 
  # and  $f_{\text{nu}}$  is the pdf of a random variable with the
  # same distribution as  $R/\sqrt{R}$  where  $R \sim \chi(\text{nu})$ 
  # distribution, by the Gauss Laguerre quadrature rule.
  #
  # Input:
  # nu.vec: a vector of degrees of freedom
  # alpha.vec: a 3-vector of alpha values where
  #           1-alpha is the nominal coverage
  #
  # Output:
  # A matrix with 3 rows and number of columns equal to
  # the length of nu.vec.
  #
  # Written by N Ranathunga in November 2019

  # Set up vectors to store the results
  comp.int <- matrix(0, nrow=length(alpha.vec),
                    ncol=length(nu.vec))
  error <- matrix(0, nrow=length(alpha.vec),
                 ncol=length(nu.vec))

  for (i in c(1:length(alpha.vec))) {
    for (j in c(1:length(nu.vec))) {
      alpha <- alpha.vec[i]
      nu <- nu.vec[j]

      if (nu == 1){
        n <- 65
      } else n <- 33

      tquant <- qt(1 - alpha/2, nu)
      comp.int[i,j] <- LaguerreQuadRule(nu, n, tquant)
      error[i, j] <- comp.int[i,j] - (1 - alpha)
    }
  }
  out <- error
}
```

```

LaguerreQuadRule <- function(nu, n, tquant){
  # Compute the integral,  $\int_0^{\infty} a(x) f_{\text{nu}}(x) dx$ 
  # where  $a(x) = 2 * \text{Phi}(t_{\text{nu}}, 1-\alpha/2) x - 1$ 
  # and  $f_{\text{nu}}$  is the pdf of a random variable with the
  # same distribution as  $R/\text{sqrt}(R)$  where  $R \sim \text{chi}(\text{nu})$ 
  # distribution, by applying the Gauss Laguerre
  # quadrature rule.
  #
  # Input:
  # nu: degrees of freedom
  # n: number of evaluations of the integrand
  # talpha: quantile of the t distribution for nu and alpha
  #
  # Output:
  # A value for the integrand  $a(x) f_{\text{nu}}(x)$ .
  #
  # Written by N. Ranathunga in April 2017

  quad.rule <- gauss.quad(n, kind = "laguerre",
                           alpha = (nu / 2) - 1)

  x.i <- quad.rule$nodes
  w.i <- quad.rule$weights
  q.i <- sqrt((2 * x.i)/nu)
  p.i <- w.i / gamma((nu / 2))

  gmw <- (2 * pnorm(tquant * q.i) - 1)
  weighted.val <- p.i * gmw
  out <- sum(weighted.val)
}

```

B.7.4 Inverse cdf method followed by Gauss Legendre quadrature

```
InvCDFLegendreError <- function(alpha.vec, nu.vec){
  # Compute the approximation error when approximate
  # the integral,  $\int_0^{\infty} a(x) f_{\nu}(x) dx$ 
  # where  $a(x) = 2 * \Phi(t_{\nu, 1-\alpha/2} x) - 1$ 
  # and  $f_{\nu}$  is the pdf of a random variable with the
  # same distribution as  $R/\sqrt{R}$  where  $R \sim \chi(\nu)$ 
  # distribution, by the Inverse cdf method followed by the
  # Gauss Legendre quadrature rule.
  #
  # Input:
  # nu.vec: a vector of degrees of freedom
  # alpha.vec: a 3-vector of alpha values where
  #           1-alpha is the nominal coverage
  #
  # Output:
  # A matrix with 3 rows and number of columns equal to
  # the length of nu.vec.
  #
  # Written by N Ranathunga in November 2019

  # Set up vectors to store the results
  comp.int <- matrix(0, nrow=length(alpha.vec),
                    ncol=length(nu.vec))
  error <- matrix(0, nrow=length(alpha.vec),
                 ncol=length(nu.vec))

  for (i in c(1:length(alpha.vec))) {
    for (j in c(1:length(nu.vec))) {
      alpha <- alpha.vec[i]
      nu <- nu.vec[j]

      if (nu == 1){
        n <- 65
      } else n <- 33

      tquant <- qt(1 - alpha/2, nu)
      comp.int[i,j] <- InvCDFLegendreQuadRule(nu, n, tquant)
      error[i, j] <- comp.int[i,j] - (1 - alpha)
    }
  }
  out <- error
}
```

```

InvCDFLegendreQuadRule <- function(nu, n, tquant){
  # Compute the integral,  $\int_0^{\infty} a(x) f_{\nu}(x) dx$ 
  # where  $a(x) = 2 * \Phi(t_{\nu, 1-\alpha/2} x) - 1$ 
  # and  $f_{\nu}$  is the pdf of a random variable with the
  # same distribution as  $R/\sqrt{R}$  where  $R \sim \chi(\nu)$ 
  # distribution, by applying the Inverse cdf method
  # followed by the Gauss Legendre quadrature rule.
  #
  # Input:
  # nu: degrees of freedom
  # n: number of evaluations of the integrand
  # talpha: quantile of the t distribution for nu and alpha
  #
  # Output:
  # A value for the integrand  $a(x) f_{\nu}(x)$ .
  #
  # Written by N Ranathunga in October 2017

  quad.rule <- gauss.quad(n, kind = "legendre")

  y1.i <- quad.rule$nodes
  w.i <- quad.rule$weights

  y1.i <- (y1.i + 1)/2
  y2.i <- sqrt(qchisq(y1.i, nu))/sqrt(nu)
  gmy <- (2 * pnorm(tquant * y2.i) - 1)
  weighted.val <- w.i * gmy

  out <- sum(weighted.val)/2
}

```

Appendix C

Proofs and R programs for Chapter 4

C.1 Computation of the first definition of scaled expected length $\text{SEL}_1(\gamma; s)$

In this section, we provide some details of the computation of the first definition of scaled expected length, $\text{SEL}_1(\gamma; s)$. We also state the formula, taken from [Giri \(2008\)](#), used to compute $E(W)$.

C.1.1 Computation of the inner integral, $\text{ISEL}(w, \gamma)$, of the double integral term in the expression of $\text{SEL}_1(\gamma; s)$

Recall that

$$\text{ISEL}(w, \gamma) = \int_{-d}^d (s(x) - t_{m, 1-\alpha/2}) \phi(wx - \gamma) dx.$$

We use this formula for $\text{ISEL}(w, \gamma)$, but with the following modification. Obviously,

$$\begin{aligned} \text{ISEL}(w, \gamma) &= \int_0^d (s(x) - t_{m,1-\alpha/2}) \phi(wx - \gamma) dx \\ &\quad + \int_{-d}^0 (s(x) - t_{m,1-\alpha/2}) \phi(wx - \gamma) dx. \end{aligned}$$

Changing the variable of integration to $u = -x$ in the second integral on the right-hand side, we obtain

$$\text{ISEL}(w, \gamma) = \int_0^d (s(x) - t_{m,1-\alpha/2}) (\phi(wx - \gamma) + \phi(wx + \gamma)) dx.$$

In other words,

$$\text{ISEL}(w, \gamma) = \int_0^d \text{ISEL}(x, w, \gamma) dx, \quad (\text{C.1})$$

where

$$\text{ISEL}(x, w, \gamma) = (s(x) - t_{m,1-\alpha/2}) (\phi(wx - \gamma) + \phi(wx + \gamma)).$$

While interpolating cubic splines are extremely smooth between successive knots, they are not particularly smooth at the knots (only the second derivative of the interpolating cubic spline is continuous at each knot). We therefore express (C.1) as

$$\int_{x_1}^{x_2} \text{ISEL}(x, w, \gamma) dx + \int_{x_2}^{x_3} \text{ISEL}(x, w, \gamma) dx + \cdots + \int_{x_{q-1}}^{x_q} \text{ISEL}(x, w, \gamma) dx$$

and then compute each of the integrals in this sum using Gauss Legendre quadrature.

C.1.2 Computation of the outer integral of the double integral term in the expression of $\text{SEL}_1(\gamma; s)$

We compute the outer integral of the double integral term in $\text{SEL}_1(\gamma; s)$, similarly to the computation of the outer integral of the double integral term

in $\text{CP}(\gamma; b, s)$ in subsection 4.3.2.

Let $\kappa = m$ and $\xi = 2$. Change the variable of integration in (4.12) to $\tilde{y} = c(m, 2) w$, where $c(m, 2) = (m/(m+2))^{1/2}$, so that

$$\begin{aligned} & \int_0^\infty \text{ISEL}(w, \gamma) w^2 f_m(w) dw \\ &= \left(\frac{2}{m}\right) \frac{\Gamma((m+2)/2)}{\Gamma(m/2)} \int_0^\infty \text{ISEL}(\tilde{y}/c(m, 2), \gamma) f_{m+2}(\tilde{y}) d\tilde{y}, \end{aligned} \quad (\text{C.2})$$

where $\text{ISEL}(\tilde{y}/c(m, 2), \gamma)$ is a smooth bounded function of $\tilde{y} \geq 0$.

After applying the transformation (2.6) of Mori (1988), i.e., $g(z)$ in (4.9), followed by the application of the trapezoidal rule, the integral in (C.2) is approximated by the finite sum

$$h_1 \sum_{j=0}^{N_1-1} \text{ISEL}(g(\tilde{z}_\ell + h_1 j)/c(m, 2), \gamma) \psi_{m+2}(\tilde{z}_\ell + h_1 j), \quad (\text{C.3})$$

where

$$\psi_{m+2}(z) = f_{m+2}(g(z)) \frac{dg(z)}{dz},$$

N_1 denotes the number of evaluations of the integrand $\text{ISEL}(g(z)/c(m, 2), \gamma) \psi_{m+2}(z)$, h_1 denotes the step length and the first evaluation of this integrand is at \tilde{z}_ℓ .

C.1.3 Computation of $E(W)$

As noted by Giri (2008),

$$E(W) = \sqrt{\frac{2}{m}} \frac{\Gamma((m+1)/2)}{\Gamma(m/2)}.$$

When m is even moderately large, $\Gamma(m/2)$ is extremely large. We therefore first compute $\ln(\Gamma((m+1)/2))$ and $\ln(\Gamma(m/2))$ using the R function `lgamma`. We then evaluate $\Gamma((m+1)/2)/\Gamma(m/2)$ by computing

$$\exp \left(\ln \left(\Gamma((m+1)/2) \right) - \ln \left(\Gamma(m/2) \right) \right).$$

C.1.4 Evaluation of the objective function $\text{OBJ}_1(\gamma; s)$

It is important to check that the term

$$\frac{1}{\sqrt{2\pi}} \left(\frac{m}{x^2 + m} \right)^{(m/2)+1} \quad (\text{C.4})$$

in the objective function that results from the first definition of scaled expected length is a smooth function of x for large m . Now

$$\begin{aligned} \left(\frac{m}{x^2 + m} \right)^{(m/2)+1} &= \frac{1}{(1 + (x^2/m))^{(m/2)+1}} \\ &= \frac{1}{(1 + (x^2/m))^{(m/2)}} \frac{1}{1 + (x^2/m)} \\ &= \frac{1}{(1 + (x^2/2)/(m/2))^{(m/2)}} \frac{1}{1 + (x^2/m)} \\ &\approx \frac{1}{\exp(x^2/2)} = \exp(-x^2/2) \end{aligned}$$

for all $x \in [0, d]$ and large m . Thus, (C.4) $\rightarrow \phi(x)$ as $m \rightarrow \infty$, for every $x \in [0, d]$. In other words, (C.4) is a smooth function of x for large m .

C.2 Computation of the second definition of scaled expected length $\text{SEL}_2(\gamma; s)$

In this section, we first derive computationally convenient expressions for the second definition of scaled expected length, $\text{SEL}_2(\gamma; s)$, and the resulting objective function $\text{OBJ}_2(\gamma; s)$. We also provide some details of the computation of this scaled expected length.

C.2.1 Computationally convenient formula for the second definition of scaled expected length $\text{SEL}_2(\gamma; s)$

It follows from the methods of [Kabaila & Giri \(2009a\)](#) that,

$$\text{SEL}_2(\gamma; s) = 1 + \frac{1}{t_{m,1-\alpha/2}} \int_0^\infty \int_{-\infty}^\infty s\left(\frac{|h|}{w}\right) \phi(h - \gamma) dh f_m(w) dw. \quad (\text{C.5})$$

The scaled expected length of the standard $1 - \alpha$ confidence interval is

$$1 = 1 + \frac{1}{t_{m,1-\alpha/2}} \int_0^\infty \int_{-\infty}^\infty t_{m,1-\alpha/2} \phi(h - \gamma) dh f_m(w) dw. \quad (\text{C.6})$$

Note that $s(x) = t_{m,1-\alpha/2}$ for all $|x| \geq d$. By subtracting (C.6) from (C.5), we obtain

$$\text{SEL}_2(\gamma; s) = 1 + \frac{1}{t_{m,1-\alpha/2}} \int_0^\infty \int_{-dw}^{dw} \left(s\left(\frac{|h|}{w}\right) - t_{m,1-\alpha/2} \right) \phi(h - \gamma) dh f_m(w) dw. \quad (\text{C.7})$$

After changing the variable of integration in the inner integral from h to $x = h/w$, we obtain

$$\text{SEL}_2(\gamma; s) = 1 + \frac{1}{t_{m,1-\alpha/2}} \int_0^\infty \int_{-d}^d (s(x) - t_{m,1-\alpha/2}) \phi(wx - \gamma) dx w f_m(w) dw. \quad (\text{C.8})$$

C.2.2 Computationally convenient formula for the objective function $\text{OBJ}_2(\gamma; s)$

We minimize

$$\int_{-\infty}^\infty (\text{SEL}_2(\gamma; s) - 1) d\nu(\gamma) \quad (\text{C.9})$$

where the weight function ν is given by equation (5) of [Kabaila & Giri \(2009a\)](#). By substituting (C.8) into (C.9), we obtain

$$\begin{aligned}
& \frac{1}{t_{m,1-\alpha/2}} \int_{-\infty}^{\infty} \int_0^{\infty} \int_{-d}^d (s(x) - t_{m,1-\alpha/2}) \phi(wx - \gamma) dx w f_m(w) dw d\nu(\gamma) \\
&= \frac{1}{t_{m,1-\alpha/2}} \int_0^{\infty} \int_{-d}^d (s(x) - t_{m,1-\alpha/2}) \int_{-\infty}^{\infty} \phi(wx - \gamma) d\nu(\gamma) dx w f_m(w) dw \\
&= \frac{2}{t_{m,1-\alpha/2}} \int_0^{\infty} \int_0^d (s(x) - t_{m,1-\alpha/2}) (\lambda + \phi(wx)) dx w f_m(w) dw \quad (\text{C.10})
\end{aligned}$$

By interchanging the order of integration in (C.10) we obtain

$$\begin{aligned}
(\text{C.10}) &= \frac{2}{t_{m,1-\alpha/2}} \int_0^d (s(x) - t_{m,1-\alpha/2}) \int_0^{\infty} (\lambda + \phi(wx)) w f_m(w) dw dx \\
&= \frac{2}{t_{m,1-\alpha/2}} \int_0^d (s(x) - t_{m,1-\alpha/2}) \times \\
&\quad \left(\int_0^{\infty} \lambda w f_m(w) dw + \int_0^{\infty} \phi(wx) w f_m(w) dw \right) dx \quad (\text{C.11})
\end{aligned}$$

Note that

$$\begin{aligned}
& \int_0^{\infty} \lambda w f_m(w) dw \\
&= \lambda \int_0^{\infty} w \frac{m^{m/2}}{\Gamma(m/2) 2^{(m/2)-1}} w^{m-1} \exp(-m w^2/2) dw \\
&= \lambda \frac{m^{m/2}}{\Gamma(m/2) 2^{(m/2)-1}} \int_0^{\infty} w^{(m+1)-1} \exp(-m w^2/2) dw. \quad (\text{C.12})
\end{aligned}$$

By the result (A2.1.3) of [Box & Tiao \(1984\)](#), (C.12) is equal to

$$\lambda \sqrt{\frac{2}{m}} \frac{\Gamma(m+1/2)}{\Gamma(m/2)} = \lambda E(W). \quad (\text{C.13})$$

Also note that

$$\int_0^{\infty} \phi(wx) w f_m(w) dw$$

$$\begin{aligned}
&= \int_0^\infty \frac{1}{\sqrt{2\pi}} \exp\left(- (wx)^2/2\right) \frac{m^{m/2}}{\Gamma(m/2) 2^{(m/2)-1}} w^m \exp\left(- m w^2/2\right) dw \\
&= \frac{m^{m/2}}{\sqrt{2\pi} \Gamma(m/2) 2^{(m/2)-1}} \int_0^\infty w^{(m+1)-1} \exp\left(- (x^2 + m) w^2/2\right) dw. \quad (\text{C.14})
\end{aligned}$$

By the result (A2.1.3) of [Box & Tiao \(1984\)](#), (C.14) is equal to

$$\frac{1}{\sqrt{2\pi}} \left(\frac{m}{x^2 + m} \right)^{(m+1)/2} \sqrt{\frac{2}{m}} \frac{\Gamma(m+1/2)}{\Gamma(m/2)} = \frac{1}{\sqrt{2\pi}} \left(\frac{m}{x^2 + m} \right)^{(m+1)/2} E(W). \quad (\text{C.15})$$

Therefore, from the results (C.13) and (C.15), (C.11) is equal to

$$\frac{2 E(W)}{t_{m,1-\alpha/2}} \int_0^d (s(x) - t_{m,1-\alpha/2}) \left(\lambda + \frac{1}{\sqrt{2\pi}} \left(\frac{m}{x^2 + m} \right)^{(m+1)/2} \right) dx.$$

C.2.3 Computation of the outer integral of the double integral term in the expression of $\text{SEL}_2(\gamma; s)$

We compute the inner integral of the double integral term in the expression of $\text{SEL}_2(\gamma; s)$ using the same approach as that described in subsection C.1.1.

We compute the outer integral of this term similarly to the computation of the outer integral of the double integral term in the expression for $\text{CP}(\gamma; b, s)$, described in subsection 4.3.2.

Let $\kappa = m$ and $\xi = 1$. Change the variable of integration to $y = c(m, 1) w$, where $c(m, 1) = (m/(m+1))^{1/2}$, so that

$$\begin{aligned}
&\int_0^\infty \text{ISEL}(w, \gamma) w f_m(w) dw \\
&= \left(\frac{2}{m} \right)^{1/2} \frac{\Gamma((m+1)/2)}{\Gamma(m/2)} \int_0^\infty \text{ISEL}(y/c(m, 1), \gamma) f_{m+1}(y) dy, \quad (\text{C.16})
\end{aligned}$$

where $\text{ISEL}(y/c(m, 1), \gamma)$ is a smooth bounded function of $y \geq 0$.

After applying the transformation (2.6) of [Mori \(1988\)](#), i.e., $g(z)$ in (4.9), followed by the application of the trapezoidal rule, the integral in (C.16) is

approximated by the finite sum

$$h_2 \sum_{j=0}^{N_2-1} \text{ISEL}(g(\tilde{z}_\ell + h_2 j)/c(m, 1), \gamma) \psi_{m+1}(\tilde{z}_\ell + h_2 j), \quad (\text{C.17})$$

where

$$\psi_{m+1}(z) = f_{m+1}(g(z)) \frac{dg(z)}{dz},$$

N_2 denotes the number of evaluations of the integrand $\text{ISEL}(g(z)/c(m, 1), \gamma)$ $\psi_{m+1}(z)$, h_2 denotes the step length and the first evaluation of this integrand is at \tilde{z}_ℓ .

C.2.4 Evaluation of the objective function $\text{OBJ}_2(\gamma; s)$

It is important to check that the term

$$\frac{1}{\sqrt{2\pi}} \left(\frac{m}{x^2 + m} \right)^{(m/2)+(1/2)} \quad (\text{C.18})$$

in the objective function that results from the second definition of scaled expected length is a smooth function of x for large m . Similarly to subsection [C.1.4](#),

$$\begin{aligned} \left(\frac{m}{x^2 + m} \right)^{(m/2)+(1/2)} &= \frac{1}{(1 + (x^2/2)/(m/2))^{(m/2)}} \frac{1}{(1 + (x^2/m))^{1/2}} \\ &\approx \frac{1}{\exp(x^2/2)} = \exp(-x^2/2) \end{aligned}$$

for all $x \in [0, d]$ and large m . Thus, [\(C.18\)](#) $\rightarrow \phi(x)$ as $m \rightarrow \infty$, for every $x \in [0, d]$. In other words, [\(C.18\)](#) is a smooth function of x for large m .

C.3 Exploration of the six cases given in Table 4.1

Figures C.1 - C.4 show the plots of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length of the confidence interval $\text{CI}(b_{1\lambda}, s_{1\lambda})$, respectively, for the six cases (a) - (f) given in Table 4.1.

Recall that $m = 2$, $\lambda = 0.15$ and $\rho = -0.5$. The functions $b_{1\lambda}$ and $s_{1\lambda}$ are obtained for $N = N_1 = 17$, Gauss Legendre quadrature with 10 nodes and $\epsilon = 10^{-7}$. The coverage probability and the squared scaled expected length plots are obtained for $N = N_1 = 33$, $\epsilon = 10^{-10}$ and the Gauss Legendre quadrature with 20 nodes.

For all the cases (a) - (f) in Table 4.1, the plots of the coverage probability and the squared scaled expected length are almost identical. There are some slight improvements in the functions $b_{1\lambda}$ and $s_{1\lambda}$, when `n.ints` is increased to 7 from 6. However, this adds about 3 min to the computation time. In subsection C.3.1, we find that `n.ints` = 6 gives better results for the case $m = 1$ than `n.ints` = 7. These findings led us to choose `n.ints` = 6 for the computations carried out in the R package `ciuppi2`.

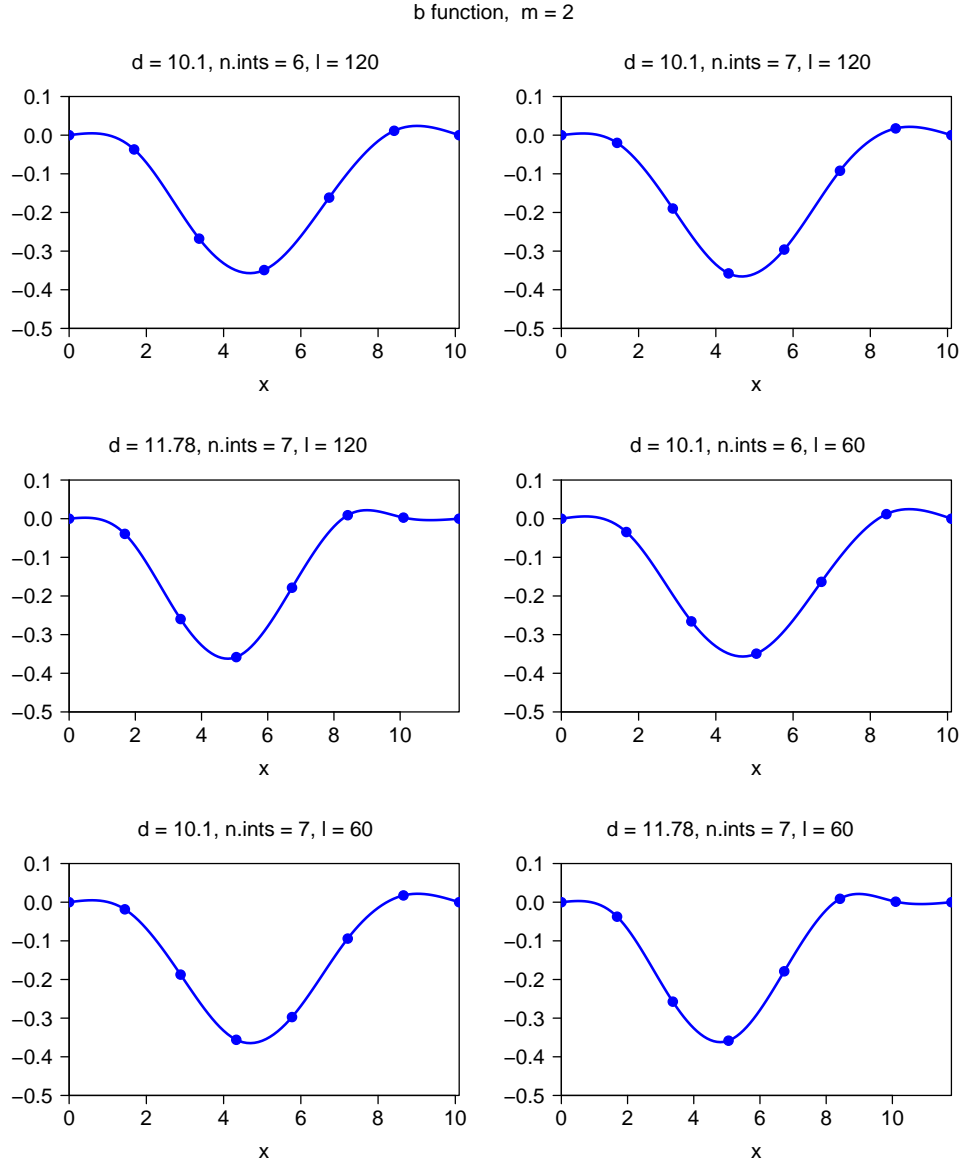


Figure C.1: Graphs of the function $b_{1\lambda}$ for $m = 2$, $\alpha = 0.05$, $\lambda = 0.15$, $\rho = -0.5$, $N = N_1 = 17$, $\epsilon = 10^{-7}$ and Gauss Legendre quadrature with 10 nodes for the cases (a) - (f) in Table 4.1.

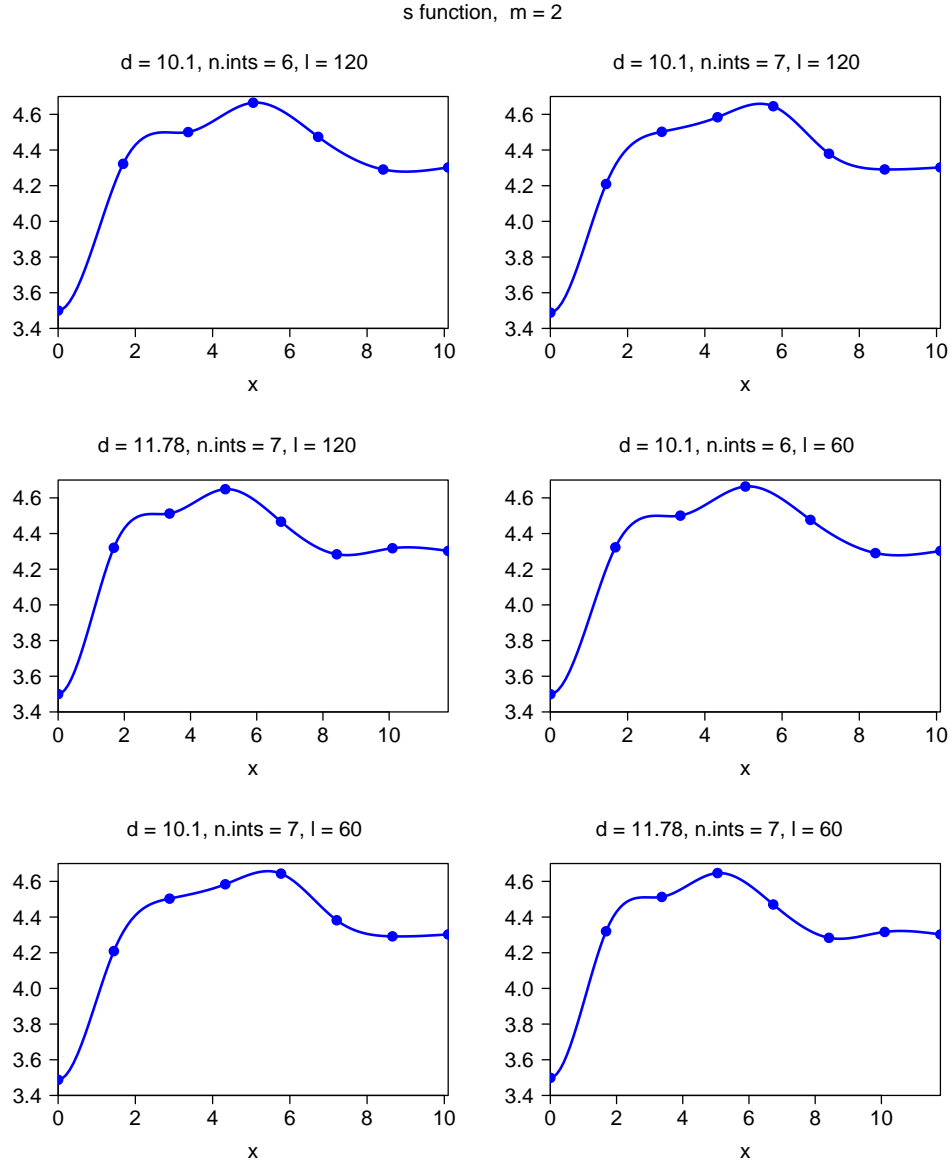


Figure C.2: Graphs of the function $s_{1\lambda}$ for $m = 2$, $\alpha = 0.05$, $\lambda = 0.15$, $\rho = -0.5$, $N = N_1 = 17$, $\epsilon = 10^{-7}$ and Gauss Legendre quadrature with 10 nodes for the cases (a) - (f) in Table 4.1.

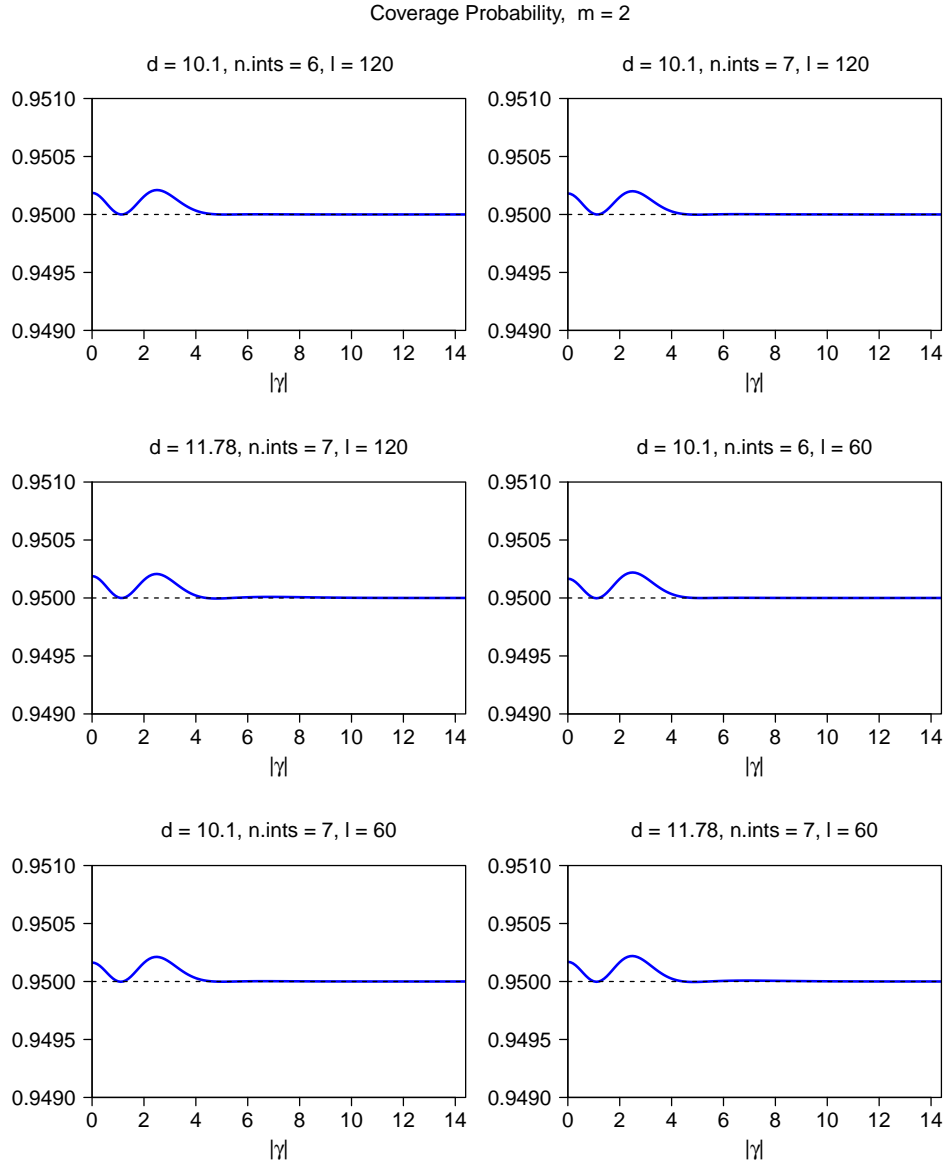


Figure C.3: Graphs of the coverage probability of the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for $m = 2$, $\alpha = 0.05$, $\lambda = 0.15$, $\rho = -0.5$, $N_1 = 33$, $\epsilon = 10^{-10}$ and Gauss Legendre quadrature with 20 nodes for the cases (a) - (f) in Table 4.1.

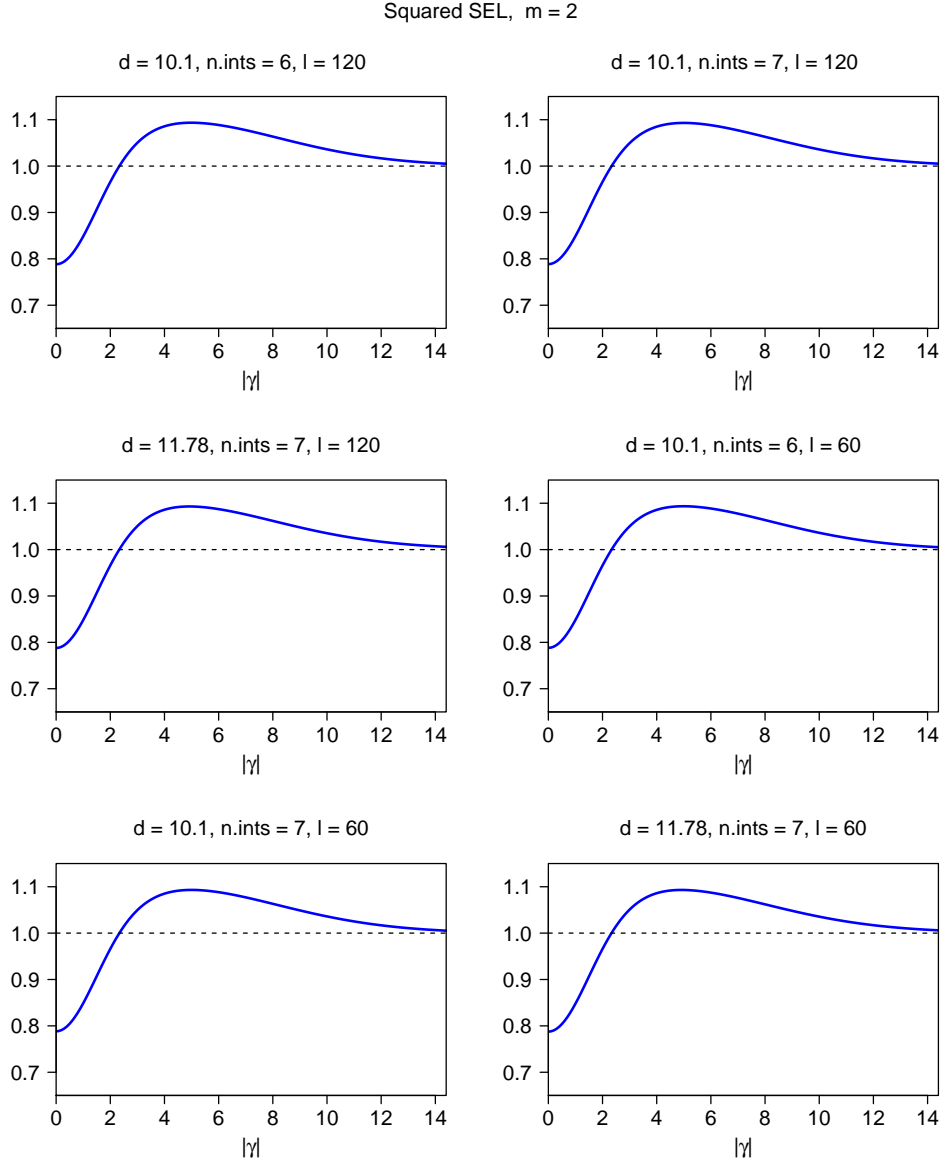


Figure C.4: Graphs of the squared scaled expected length of the confidence interval $\text{CI}(b_{1\lambda}, s_{1\lambda})$ for $m = 2$, $\alpha = 0.05$, $\lambda = 0.15$, $\rho = -0.5$, $N_1 = 33$, $\epsilon = 10^{-10}$ and Gauss Legendre quadrature with 20 nodes for the cases (a) - (f) in Table 4.1.

C.3.1 Numerical results for a given value of λ for $m = 1$

We consider the numerical example given in Giri (2008) where $m = 1$, $\alpha = 0.05$, $\rho = 0.4$ and $\lambda = 0.2$. For $m = 1$, we find $d = 20$ using the method described in subsection 4.5.1. Figures C.5 and C.6 are for the cases (a) $l = 60$,

$d = 20$, $\mathbf{n.ints} = 6$ and (b) $l = 60$, $d = 20$, $\mathbf{n.ints} = 7$, respectively.

We consider the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$. Previous experience of Kabaila & Giri (2009a), Giri (2008) and Kabaila & Giri (2013) with the computation of this confidence interval suggests that the function $b_{1\lambda}$ should only take positive values when $\rho > 0$. The graph of the function $b_{1\lambda}$ in the left panel of Figure C.5, for $\mathbf{n.ints} = 6$, does not dip as far below zero as the corresponding graph in the left panel of Figure C.6, for $\mathbf{n.ints} = 7$. The coverage probability and the squared scaled expected length plots are very similar for both cases of $\mathbf{n.ints}$. The computation time to compute the vector $(b_{1\lambda}(x_2), \dots, b_{1\lambda}(x_{q-1}), s_{1\lambda}(x_1), \dots, s_{1\lambda}(x_{q-1}))$ is 38.89 min when $\mathbf{n.ints} = 7$ and 11.11 min when $\mathbf{n.ints} = 6$. These findings led us to choose $\mathbf{n.ints} = 6$ for the computations carried out in the R package `ciuupi2`.

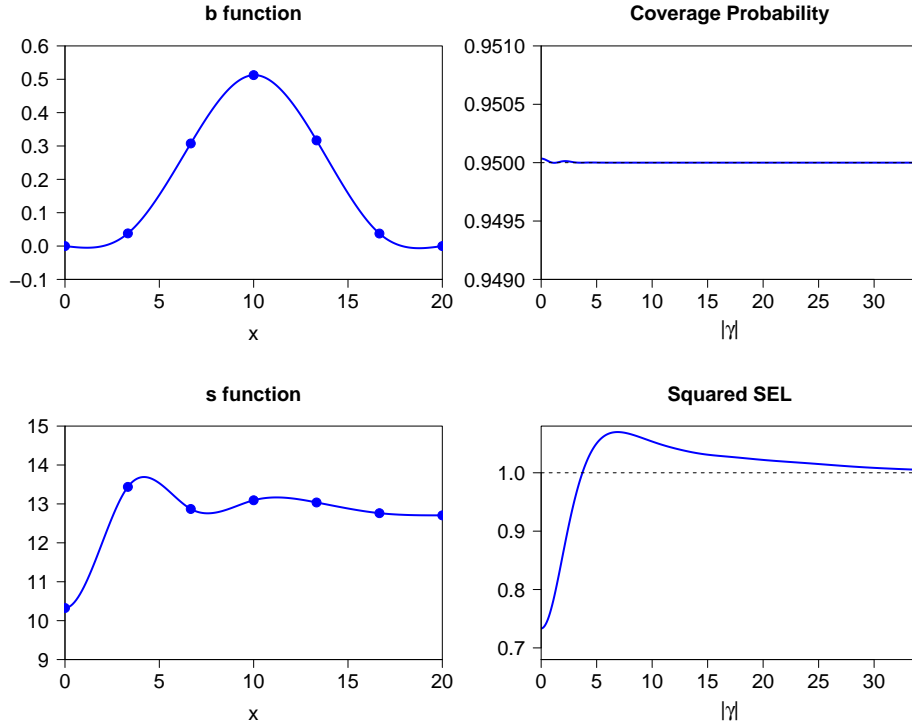


Figure C.5: Graphs of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length of the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for (a) $l = 60$, $d = 20$, $\mathbf{n.ints}=6$. Note that $m = 1$, $\alpha = 0.05$, $\rho = 0.4$ and $\lambda = 0.2$.

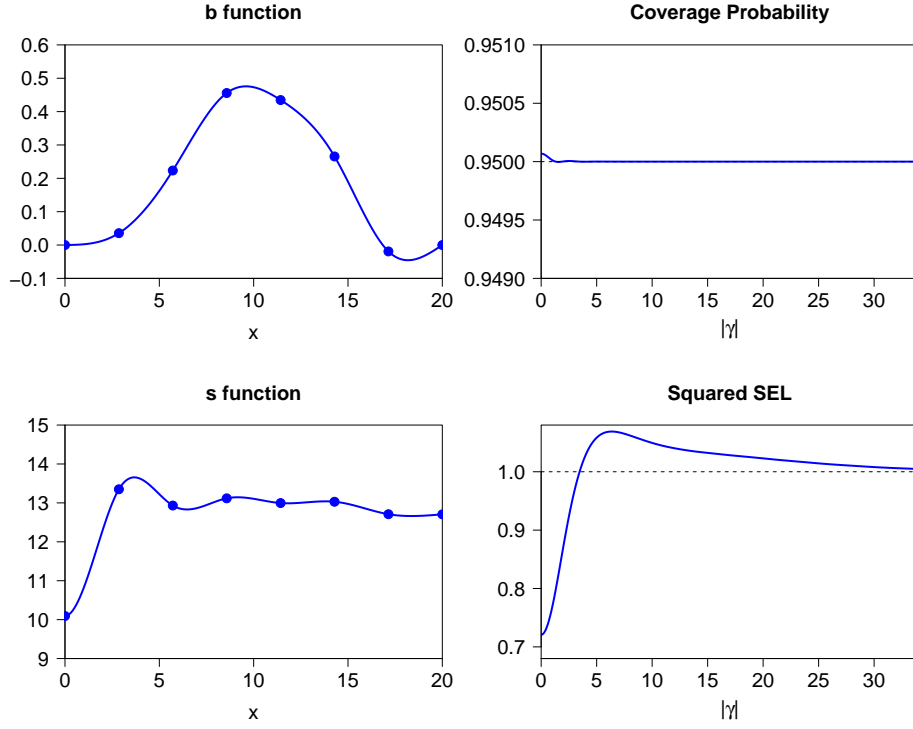


Figure C.6: Graphs of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length of the confidence interval $\text{CI}(b_{1\lambda}, s_{1\lambda})$ for (a) $l = 60$, $d = 20$, $\text{n.ints}=7$. Note that $m = 1$, $\alpha = 0.05$, $\rho = 0.4$ and $\lambda = 0.2$.

C.4 Choosing the number of outer integrand evaluations, the value of ϵ and the number of Gauss Legendre quadrature nodes

The computation times for the graphs of the coverage probability and the squared scaled expected length of $\text{CI}(b_{1\lambda}, s_{1\lambda})$ are negligible compared to the time needed to compute the vector $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$ which specifies the functions $b_{1\lambda}$ and $s_{1\lambda}$. Because of this large difference in computational times, we have chosen the computational parameters differently in these two cases.

Recall that N, N_1 and N_2 denote the number of outer integrand evaluations in the expressions for the coverage probability, first definition of the scaled expected length and the second definition of the scaled expected length, respectively. We have chosen $N = N_1 = N_2$ for simplicity. In subsection C.4.1, we choose N, ϵ and the number of Gauss Legendre quadrature nodes for the computation of $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$. In subsection C.4.2, we choose N, ϵ and the number of Gauss Legendre quadrature nodes for the computation of the graphs of the coverage probability and the squared scaled expected length of $\text{CI}(b_{1\lambda}, s_{1\lambda})$. We consider $N \in \{5, 9, 17, 33\}$.

C.4.1 Choice of N, ϵ and the number of Gauss Legendre quadrature nodes to compute the functions $b_{1\lambda}$ and $s_{1\lambda}$

We chose the common value N, ϵ and the number of Gauss Legendre quadrature nodes by comparing the values of $(\text{SEL}_1(0; s_{1\lambda}))^2$, the maximum of $(\text{SEL}_1(\gamma; s_{1\lambda}))^2$ and the corresponding time needed to compute the vector $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$, for $N \in \{5, 9, 17, 33\}$. We consider the two cases (a) $m > 2$ and (b) $m \in \{1, 2\}$.

Case (a):

Consider the numerical example in Kabaila & Giri (2009a) where $m = 76$, $\alpha = 0.05$, $\rho = -0.707$ and $\lambda = 0.2$. For this example, $d = 6$ and `n.ints` = 6. Note that $\Gamma_{\text{grid}} \in \{0, 0.05, \dots, 8\}$, which is equally spaced grid of values. Initially, we choose $\epsilon = 10^{-7}$ and Gauss Legendre quadrature with 5 nodes.

Table C.1 shows that for $N \in \{17, 33\}$, the values of $(\text{SEL}_1(0; s_{1\lambda}))^2$ and $\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda})\right)^2$ become stable. Also, note that the values of $(\text{SEL}_1(0; s_{1\lambda}))^2$ and $\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda})\right)^2$ are the same to the third decimal points

for $N \in \{9, 17\}$. The computation time for $N = 9$ is 6.16 mins, which is a decrement of roughly 4 mins compared to $N = 17$. This led us to choose $N = 9$, $\epsilon = 10^{-7}$ and Gauss Legendre quadrature with 5 nodes to compute the vector $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$ for $m > 2$.

The functions $b_{1\lambda}$ and $s_{1\lambda}$ for these chosen values are shown in the left panel of Figure C.9. These are almost identical to Figure 2 of Kabaila & Giri (2009a).

Table C.1: The values of $(\text{SEL}_1(0; s_{1\lambda}))^2$, $\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda})\right)^2$ and the times needed to compute the vector $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$ for $N \in \{5, 9, 17, 33\}$ for Gauss Legendre quadrature with 5 nodes and $\epsilon = 10^{-7}$.

N	$(\text{SEL}_1(0; s_{1\lambda}))^2$	$\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda})\right)^2$	Time
5	0.8339138	1.1370920	2.84 min
9	0.8678885	1.1056900	6.16 min
17	0.8679618	1.1056200	10.03 min
33	0.8679618	1.1056200	26.58 min

Case (b):

Consider the numerical example in Kabaila & Giri (2013) where $m = 2$, $\alpha = 0.05$, $\rho = -0.5$ and $\lambda = 0.15$. For this example, $d = 14$ and `n.ints` = 7. Note that $\Gamma_{\text{grid}} \in \{0, 0.05, \dots, 15\}$, which is equally spaced grid of values. Initially, we choose $\epsilon = 10^{-7}$ and Gauss Legendre quadrature with 10 nodes.

Table C.2 shows that the values of $(\text{SEL}_1(0; s_{1\lambda}))^2$ and $\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda})\right)^2$ for $N = 17$ and $N = 33$ are approximately similar. Also, note that for $N = 17$, the computation time is roughly half the computation time for $N = 33$. This led us to choose $N = 17$, $\epsilon = 10^{-7}$ and Gauss Legendre quadrature with 10 nodes to compute the vector $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$

for $m \in \{1, 2\}$.

The functions $b_{1\lambda}$ and $s_{1\lambda}$ for these chosen values are shown in the left panel of Figure C.10. These are very similar to the corresponding results in Kabaila & Giri (2013).

Table C.2: The values of $(\text{SEL}_1(0; s_{1\lambda}))^2$, $\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda})\right)^2$ and the times needed to compute the vector $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$ for $N \in \{5, 9, 17, 33\}$ for Gauss Legendre quadrature with 10 nodes and $\epsilon = 10^{-7}$.

N	$(\text{SEL}_1(0; s_{1\lambda}))^2$	$\left(\max_{\gamma \geq 0} \text{SEL}_1(\gamma; s_{1\lambda})\right)^2$	Time
5	0.8977795	1.0594070	16.72 min
9	0.7454967	1.1173540	31.90 min
17	0.7867843	1.0982670	46.21 min
33	0.7876437	1.0963369	1.56 hr

C.4.2 Choice of N, ϵ and the number of Gauss Legendre quadrature nodes to compute the graphs of the coverage probability and the squared scaled expected length

As noted earlier, the computation times for the graphs of the coverage probability and the squared scaled expected length of $\text{CI}(b_{1\lambda}, s_{1\lambda})$ are negligible compared to the time needed to compute the vector $(b_{1\lambda}(d/6), \dots, b_{1\lambda}(5d/6), s_{1\lambda}(0), \dots, s_{1\lambda}(5d/6))$ which specifies the functions $b_{1\lambda}$ and $s_{1\lambda}$. Therefore, initially, we increase N to 33, the number of Gauss Legendre quadrature nodes to 20 and decrease ϵ to 10^{-10} when computing these graphs.

We consider the two examples used in subsection C.4.1 for the two cases (a) $m > 2$ and (b) $m \in \{1, 2\}$, with $N = 33$, Gauss Legendre quadrature

with 20 nodes and $\epsilon = 10^{-10}$. Figures C.9 and C.10 summarize the following results.

- (a) $m = 76$: The right panel of Figure C.9 shows that the plots of the coverage probability and the squared scaled expected length are identical to the corresponding plots in Kabaila & Giri (2009a).
- (b) $m = 2$: The right panel of Figure C.10 shows that the plots of the coverage probability and the squared scaled expected length are very similar to the corresponding plots in Kabaila & Giri (2013).

This comparison led us to choose $N = 33$, Gauss Legendre quadrature with 20 nodes and $\epsilon = 10^{-10}$ for computing the graphs of the coverage probability and the squared scaled expected length. We verify the appropriateness of these choices in Section C.5.

C.5 Differences between the graphs of the coverage probabilities and the differences between the graphs of the squared scaled expected lengths

We consider the two cases,

(1) $N = 17$, the number of Gauss Legendre nodes = 10, $\epsilon = 10^{-7}$

and

(2) $N = 33$, the number of Gauss Legendre nodes = 20, $\epsilon = 10^{-10}$.

Figures C.7 and C.8 show the plots of the differences, (1) – (2), between the graphs of the coverage probabilities and the differences, (1) – (2), between

the graphs of the squared scaled expected lengths, respectively, for the six cases given in Table 4.1.

The differences, (1) – (2), between the graphs of the coverage probabilities and the differences, (1) – (2), between the graphs of the squared scaled expected lengths are very small for all the six cases. In fact, the absolute differences of the graphs of the coverage probabilities are roughly $< 5 \times 10^{-6}$ and the absolute differences of the graphs of the squared scaled expected lengths are roughly $< 2 \times 10^{-4}$. This suggests that choosing (2) should lead to highly accurate results. Consequently, we have chosen (2).

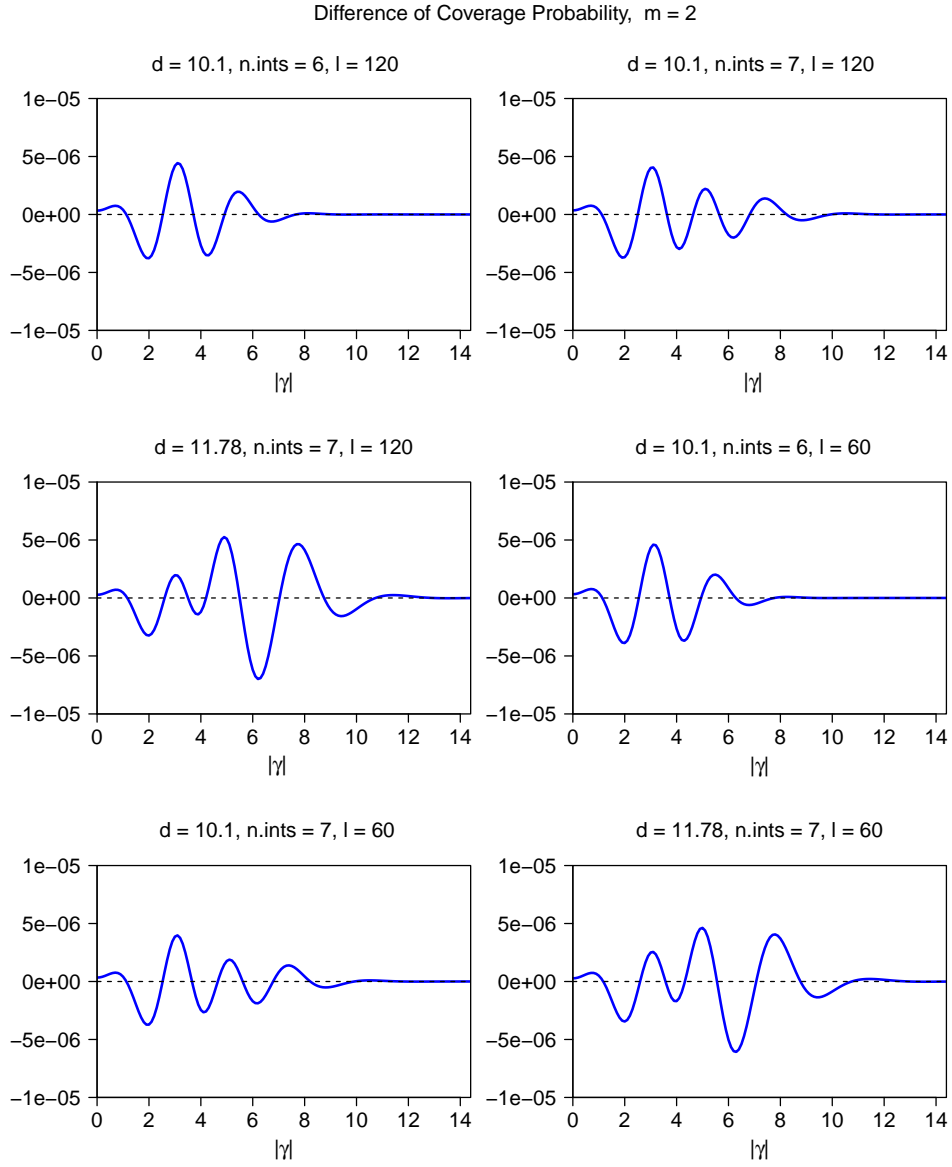


Figure C.7: Plots of the differences, (1) – (2), between the graphs of the coverage probabilities for the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for the six cases (a) – (f) in Table 4.1. Note that $m = 2$, $\alpha = 0.05$, $\rho = -0.5$, $\lambda = 0.15$.

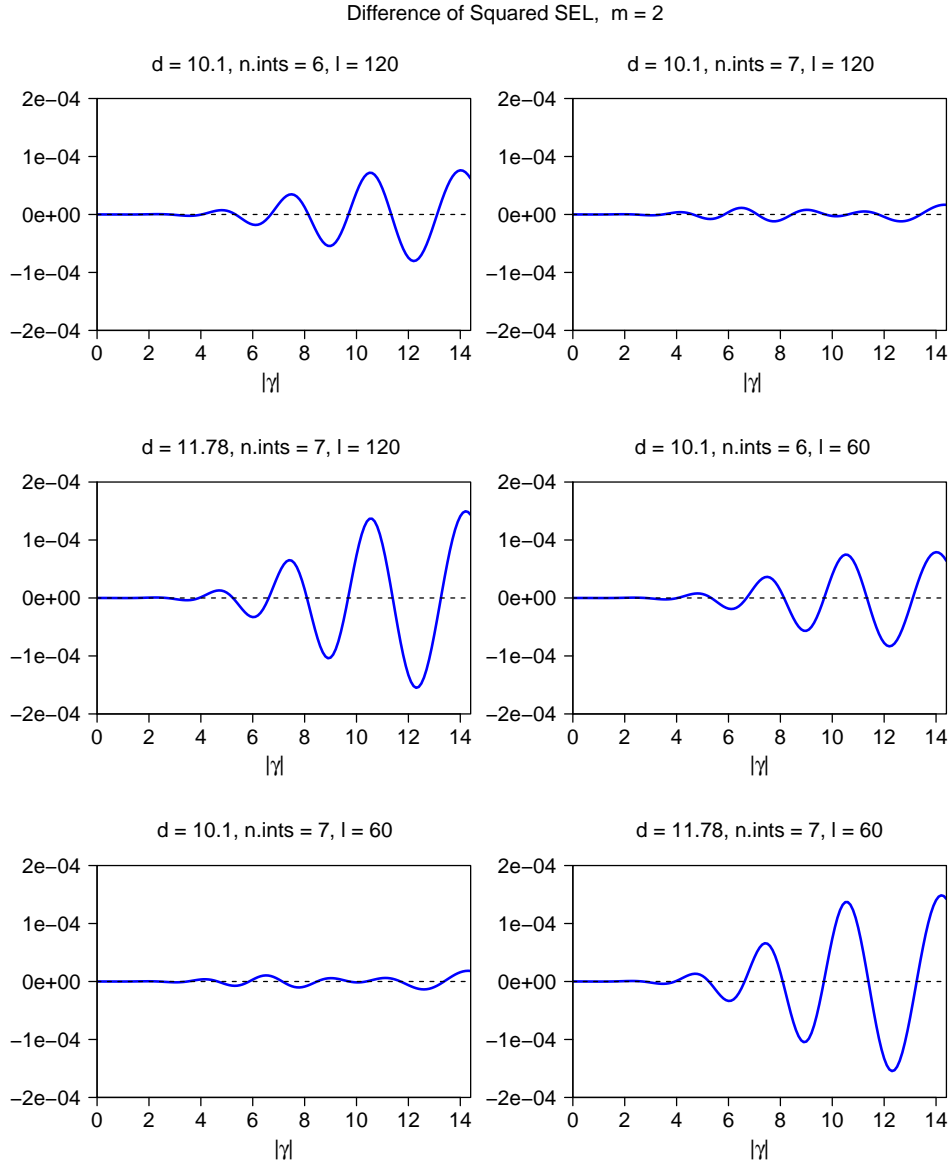


Figure C.8: Plots of the differences, (1) – (2), between the graphs of the squared scaled expected lengths for the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for the six cases (a) – (f) in Table 4.1. Note that $m = 2$, $\alpha = 0.05$, $\rho = -0.5$, $\lambda = 0.15$.

C.6 Comparison with some of the past results obtained by Kabaila and Giri

In this section, we compute the plots of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length of the confidence interval $\text{CI}(b_{1\lambda}, s_{1\lambda})$ for the three examples (a) [Kabaila & Giri \(2009a\)](#) for $m = 76$ and $\lambda = 0.2$, (b) [Kabaila & Giri \(2013\)](#) for $m = 2$ and $\lambda = 0.15$ and (c) [Giri \(2008\)](#) for $m = 1$ and $\lambda = 0.2$. We consider the same values of d and `n.ints` used by these authors. We use $\Gamma_{\text{grid}} = \{0, 0.05, \dots, 8\}$ for $m = 76$, $\Gamma_{\text{grid}} = \{0, 0.05, \dots, 15\}$ for $m = 2$ and $\Gamma_{\text{grid}} = \{0, 0.05, \dots, 30\}$ for $m = 1$, which are all equally spaced grids of values of γ .

Plots of the functions $b_{1\lambda}$ and $s_{1\lambda}$ were obtained using $N = 9$, Gauss Legendre quadrature with 5 nodes and $\epsilon = 10^{-7}$ for the example (a) and $N = 17$, Gauss Legendre quadrature with 10 nodes and $\epsilon = 10^{-7}$ for the examples (b) and (c), respectively. Plots of the coverage probability and the squared scaled expected length were obtained using $N = 33$, Gauss Legendre quadrature with 20 nodes and $\epsilon = 10^{-10}$ for the three examples.

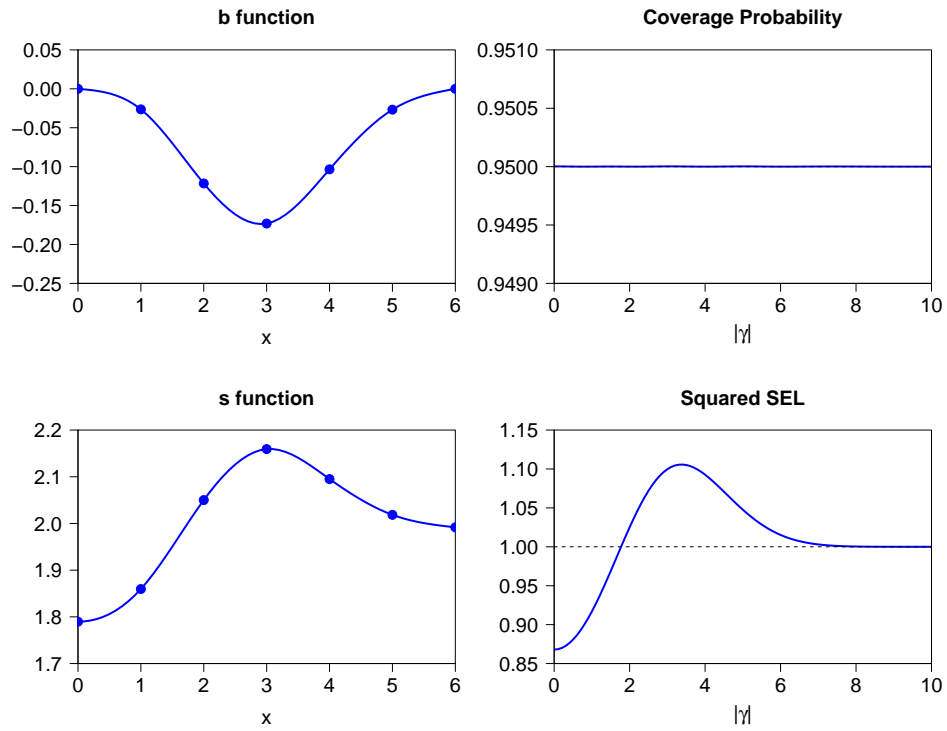


Figure C.9: Graphs of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length of the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for the example (a). In this example, $m = 76$, $\alpha = 0.05$, $\rho = -0.707$, $\lambda = 0.2$, $d = 6$ and `n.ints` = 6.

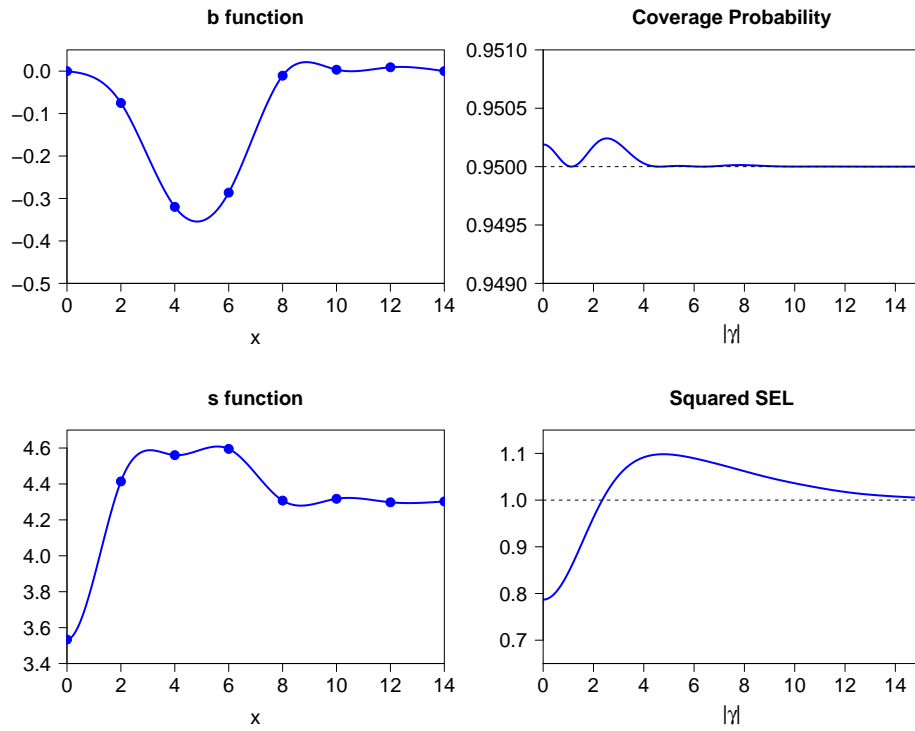


Figure C.10: Graphs of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length of the confidence interval $CI(b_{1\lambda}, s_{1\lambda})$ for the example (b). In this example, $m = 2$, $\alpha = 0.05$, $\rho = -0.5$, $\lambda = 0.15$, $d = 14$ and $\text{n.ints} = 7$.

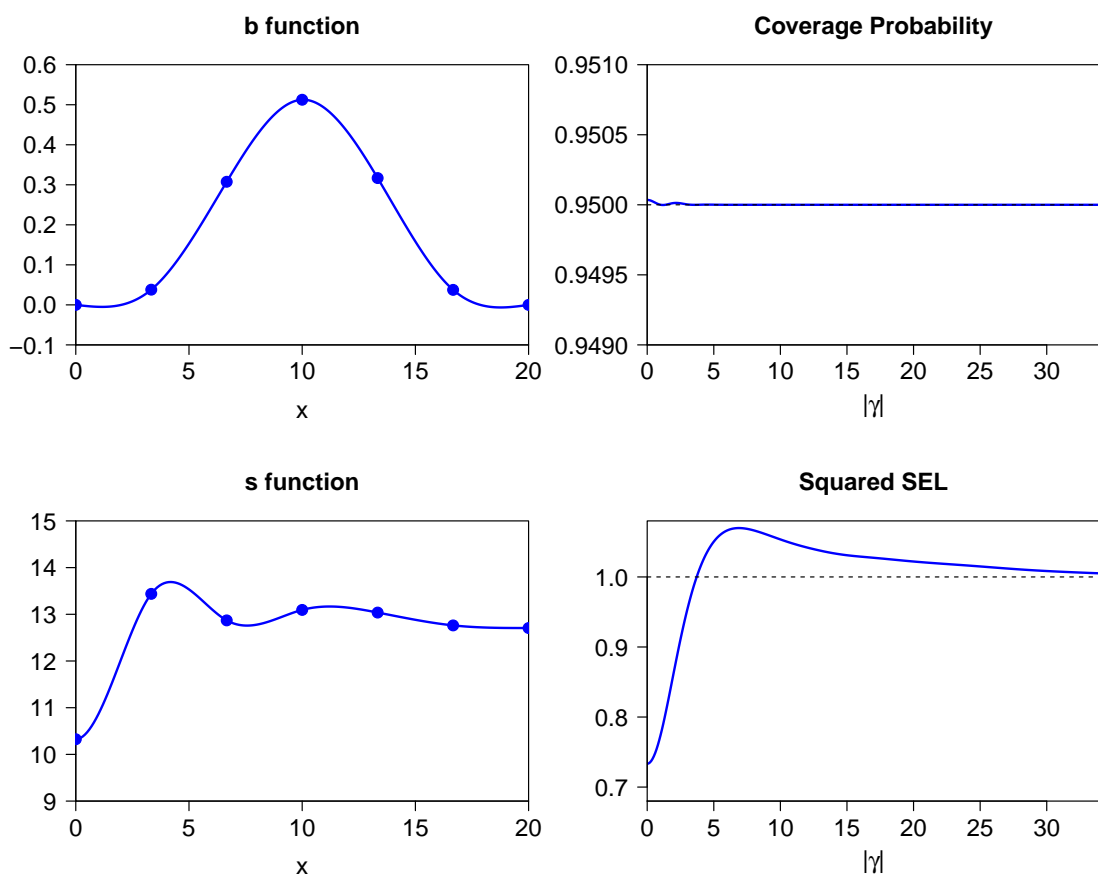


Figure C.11: Graphs of the functions $b_{1\lambda}$ and $s_{1\lambda}$, the coverage probability and the squared scaled expected length of the confidence interval $\text{CI}(b_{1\lambda}, s_{1\lambda})$ for the example (c). In this example, $m = 1$, $\alpha = 0.05$, $\rho = 0.4$, $\lambda = 0.2$, $d = 30$ and `n.ints` = 6.

C.7 R programs for the computation of the [Kabaila & Giri \(2009a\)](#) confidence inter- val for the standard choice of λ

In subsection [C.7.1](#), we list the R programs for the computation of the vector $(b_1(d/6), \dots, b_1(5d/6), s_1(0), \dots, s_1(5d/6))$ which specifies the functions b_1 and s_1 of the [Kabaila & Giri \(2009a\)](#) confidence interval that utilizes uncertain prior information, for the standard choice of λ and for both possible objective functions (corresponding to the two definitions of scaled expected length). In subsection [C.7.2](#), we list the R program for the computation of the graphs of the functions b_1 and s_1 . In subsection [C.7.3](#), we list the R program for the computation of the graph of the coverage probability of this confidence interval. In subsections [C.7.4](#) and [C.7.5](#), we list the R programs for the computation of the graphs of the scaled expected length, for both definitions of scaled expected length, of this confidence interval.

C.7.1 R programs for the computation of the vector

$$(b_1(d/6), \dots, b_1(5d/6), s_1(0), \dots, s_1(5d/6))$$

```
bsciup12 <- function(alpha, m, rho, obj = 1, natural = 1){  
  # Compute the vector (b(d/6),...,b(5d/6),s(0),...,s(5d/6))  
  # that specifies the confidence interval that utilizes  
  # uncertain prior information (CIUUPI) for the case where  
  # the error variance is unknown.  
  #  
  # Inputs:  
  # alpha: minimum coverage probability is 1 - alpha  
  # m: degrees of freedom n - p  
  # rho: a known correlation  
  # obj: equal to 1 (default) for the first definition of the  
  #       scaled expected length or 2 for the second  
  #       definition  
  #       of the scaled expected length  
  # natural: equal to 1 (default) if the functions b and s  
  #           are found by natural cubic spline interpolation  
  #           or 0 if these functions are found by clamped  
  #           cubic  
  #           spline interpolation in the interval [-d, d]  
  #  
  # Output:  
  # The vector (b(d/6),...,b(5d/6),s(0),...,s(5d/6))  
  # that specifies the CIUUPI for unknown variance.  
  #  
  # Written by N Ranathunga in September 2020  
  
  # Specify the values of the inputs to other functions  
  n.iter <- 5  
  n.ints <- 6  
  eps <- 10^{-7}  
  d <- choice_d(m)$d  
  gams <- choice_d(m)$gams  
  
  if(m==1 | m==2) {  
    N <- 17  
    n.nodes <- 10  
  } else {  
    N <- 9  
    n.nodes <- 5  
  }  
  
  # Set t.alpha for m and alpha  
  t.alpha <- stats::qt(1 - alpha/2, m)  
  
  # Find the nodes and weights of the legendre quadrature  
  quad.info <- statmod::gauss.quad(n.nodes, kind="legendre")  
  nodes <- quad.info$nodes  
  weights <- quad.info$weights  
  
  # Specify where the knots for b and s are located  
  # as inputs to other functions  
  knots <- seq(0, d, by = d/n.ints)  
  knots.all <- seq(-d, d, by = d/n.ints)
```

```

# Specify the values of the inputs to compute the
# outer integral in coverage probability
h <- OuterPara(m, nu=m+1, N, eps)$h
wvec <- OuterPara(m, nu=m+1, N, eps)$wvec
psinu.zvec <- OuterPara(m, nu=m+1, N, eps)$psinu.zvec
cons <- OuterPara(m, nu=m+1, N, eps)$cons1

# Find a starting value for the optimization problem
start.vec <- startOptim(n.ints=n.ints, t.alpha=t.alpha)

cat("Computing the bs vector that specifies the Kabaila and
    Giri CI... ")

lambda <- compute_lambda(n.iter, rho, alpha, t.alpha, gams,
                        d, m, n.ints, N, eps, knots, knots
                          .all, nodes,
                          weights, natural, obj, start.vec)

new.par <- optimize_knots(lambda, rho, alpha, t.alpha, gams
                        , d, m,
                          n.ints, knots, knots.all, nodes,
                          weights, wvec,
                          psinu.zvec, h, cons, natural, obj
                          , start.vec)

cat("DONE", "\n")

out <- new.par
}

```

```

choice_d <- function(m, length.out = 60){
  # Compute the value of d and the vector
  # (0:d+constant) of length 60 for a given m.
  #
  # Inputs:
  # m: degrees of freedom n - p
  # length.out: length of gams vector
  #
  # Output:
  # A list which contains the value d and vector gams.
  #
  # Written by P Kabaila in September 2020
  # Modified by N Ranathunga in October 2020

  # Set a cutoff value to the Normal curve
  init.cutoff.d <- 1.545

  # Find a value for d
  init.prob.d <- stats::pnorm(init.cutoff.d, 0, 1)
  multiplier <- 6 / init.cutoff.d
  d <- round(multiplier * stats::qt(init.prob.d, m), 1)

  # Find a maximum possible value for gams
  init.prob.extra <- stats::pnorm(2, 0, 1)
  extra <- round(stats::qt(init.prob.extra, m), 1)
  max.gamma.constr <- d + extra

  # Find the gams vector
  gams <- c(seq(0, d, length.out = length.out),
            seq(d, max.gamma.constr, by = (2 * d / length.out)
              )))

  out <- list(d=d, gams=gams)
}

```

```

compute_lambda <- function(n.iter, rho, alpha, t.alpha, gams,
  d, m,
                                n.ints, N, eps, knots, knots.all,
                                nodes,
                                weights, natural, obj, start.vec){
  # This program will find the optimized value of lambda for
  # given rho and alpha.
  #
  # Details for finding the optimized value of lambda:
  # For 5 iterations, the bisection method is used to find
  # the
  # value of lambda such that the SEL "loss" is equal to the
  # SEL
  # "gain". Once these iterations are performed, a final
  # answer
  # for lambda is found by fitting a straight line to the two
  # last iteration values, and finding the x-axis intercept
  # of
  # this straight line.
  #
  # Inputs:
  # n.iter: number of iterations
  # rho: known correlation
  # alpha: 1 - alpha is the minimum coverage probability of
  # the
  # confidence interval
  # t.alpha: quantile of the t distribution for m and alpha
  # gams: constrain coverage probability at these values
  # d: the b and s functions are optimized in the interval
  # (0, d]
  # m: degrees of freedom n - p
  # n.ints: number of intervals in [0,d]
  # N: number of evaluations in the outer integrand
  # eps: upper bound of the approximation error
  # knots: location of knots in [0, d]
  # knots.all: location of knots in [-d, d]
  # nodes: vector of Gauss Legendre quadrature nodes
  # weights: vector of Gauss Legendre quadrature weights
  # natural: 1 (default) for natural cubic spline
  # interpolation
  # or 0 for clamped cubic spline interpolation
  # obj: 1 for definition 1 of SEL or 2 for definition 2 of
  # SEL
  # start.vec: a starting vector to the optimization problem
  #
  # Output:
  # The optimized value of lambda
  #
  # Written by R Mainzer, August 2017
  # Modified by N Ranathunga in September 2020

  # The lower and upper bounds of the initial search interval
  # for the bisection root finding method
  lower <- 0
  upper <- 0.3

  # Set up vectors to store results
  res.lambda <- rep(0, n.iter)
  res.fun <- rep(0, n.iter)

```

```

# Implement the bisection method
for(i in 1:n.iter){

  tmp.lambda <- (upper + lower)/2
  tmp.fun <- compute_ratio_minus(tmp.lambda, rho, alpha, t.
    alpha,
                                gams, d, m, n.ints, N,
                                eps, knots,
                                knots.all, nodes, weights,
                                , natural,
                                obj, start.vec)

  if(tmp.fun < 0){
    lower <- tmp.lambda
  } else{
    upper <- tmp.lambda
  }

  res.lambda[i] <- tmp.lambda
  res.fun[i] <- tmp.fun

}

# Find lambda by a linear interpolation of the last
# upper and lower values
x1 <- lower
if(tmp.lambda == x1){
  y1 <- tmp.fun
} else {
  y1 <- compute_ratio_minus(x1, rho, alpha, t.alpha,
                                gams, d, m, n.ints, N, eps,
                                knots,
                                knots.all, nodes, weights,
                                natural,
                                obj, start.vec)
}

x2 <- upper
if(tmp.lambda == x2){
  y2 <- tmp.fun
} else {
  y2 <- compute_ratio_minus(x2, rho, alpha, t.alpha,
                                gams, d, m, n.ints, N, eps,
                                knots,
                                knots.all, nodes, weights,
                                natural,
                                obj, start.vec)
}

slope <- (y2 - y1) / (x2 - x1)
x3 <- x1 - (y1 / slope)

# Do the linear interpolation one more time
y3 <- compute_ratio_minus(x3, rho, alpha, t.alpha,
                                gams, d, m, n.ints, N, eps,
                                knots,
                                knots.all, nodes, weights,
                                natural,
                                obj, start.vec)

```

```
    if (sign(y1) != sign(y3)){  
      slope <- (y3 - y1) / (x3 - x1)  
      lambda <- x1 - (y1 / slope)  
    } else {  
      slope <- (y3 - y2) / (x3 - x2)  
      lambda <- x2 - (y2 / slope)  
    }  
  
    out <- lambda  
  }  
}
```



```

compute_ratio_minus <- function(lambda, rho, alpha, t.alpha,
                                gams,
                                d, m, n.ints, N, eps, knots,
                                knots.all,
                                nodes, weights, natural, obj,
                                start.vec){
  # Compute the ratio ( gain / maximum possible loss) - 1.
  # Another program can then use this program to find the
  # value of lambda which makes (this ratio - 1) = 0.
  #
  # Inputs:
  # lambda: used in specifying the objective function
  # rho: known correlation
  # alpha: 1 - alpha is the minimum coverage probability of
  # the
  # confidence interval
  # t.alpha: quantile of the t distribution for m and alpha
  # gams: constrain coverage probability at these values
  # d: the b and s functions are optimized in the interval
  # (0, d]
  # m: degrees of freedom n - p
  # n.ints: number of intervals in [0,d]
  # N: number of evaluations in the outer integrand
  # eps: upper bound of the approximation error
  # knots: location of knots in [0, d]
  # knots.all: location of knots in [-d, d]
  # nodes: vector of Gauss Legendre quadrature nodes
  # weights: vector of Gauss Legendre quadrature weights
  # natural: 1 (default) for natural cubic spline
  # interpolation
  # or 0 for clamped cubic spline interpolation
  # obj: 1 for definition 1 of SEL or 2 for definition 2 of
  # SEL
  # start.vec: a starting vector to the optimization problem
  #
  # Output:
  # The ratio (gain / maximum possible loss) - 1.
  #
  # Written by P.Kabaila in June 2008
  # Rewritten in R by R Mainzer, March 2017
  # Modified by N Ranathunga in September 2020

  # Specify the values of the inputs to compute the
  # outer integral in coverage probability and SEL2
  h <- OuterPara(m, nu=m+1, N, eps)$h
  wvec <- OuterPara(m, nu=m+1, N, eps)$wvec
  psinu.zvec <- OuterPara(m, nu=m+1, N, eps)$psinu.zvec
  cons <- OuterPara(m, nu=m+1, N, eps)$cons1

  new.par <- optimize_knots(lambda, rho, alpha, t.alpha, gams
    , d, m,
    n.ints, knots, knots.all, nodes, weights,
    wvec,
    psinu.zvec, h, cons, natural, obj, start.vec)

  s.spl <- spline_s(new.par, n.ints, knots.all, t.alpha,
    natural)

  # Compute the required ratio

```

```

if (obj == 1){

  h1 <- OuterPara(m, nu=m+2, N, eps)$h
  wvec1 <- OuterPara(m, nu=m+2, N, eps)$wvec
  psinu.zvec1 <- OuterPara(m, nu=m+2, N, eps)$psinu.zvec
  cons1 <- OuterPara(m, nu=m+2, N, eps)$cons2
  exp.w1 <- OuterPara(m, nu=m+2, N, eps)$exp.w

  sel.max <- stats::optimize(compute_sel1_trapez, c(0, d),
    maximum = TRUE,
    knots=knots, t.alpha=t.alpha, nodes=nodes,
    weights=weights,
    s.spl=s.spl, wvec=wvec1, psinu.zvec=psinu.
    zvec1,
    h1=h1, cons=cons1, exp.w=exp.w1)$objective

  sel.min <- compute_sel1_trapez(gam = 0, knots, t.alpha,
    nodes, weights,
    s.spl, wvec=wvec1, psinu.zvec=psinu.zvec1,
    h1=h1, cons=cons1, exp.w=exp.w1)

} else if (obj == 2) {

  sel.max <- stats::optimize(compute_sel2_trapez, c(0, d),
    maximum = TRUE,
    knots=knots, t.alpha=t.alpha, nodes=nodes,
    weights=weights,
    s.spl=s.spl, wvec=wvec, psinu.zvec=psinu.
    zvec,
    h2=h, cons=cons)$objective

  sel.min <- compute_sel2_trapez(gam = 0, knots, t.alpha,
    nodes, weights,
    s.spl=s.spl, wvec, psinu.zvec,
    h2=h, cons)

}

expected.gain <- 1 - sel.min^2
max.potential.loss <- sel.max^2 - 1

# Output the required ratio minus 1
out <- expected.gain / max.potential.loss - 1

}

```

```

optimize_knots <- function(lambda, rho, alpha, t.alpha, gams,
  d, m,
                                n.ints, knots, knots.all, nodes,
                                weights,
                                wvec, psinu.zvec, h, cons, natural
                                , obj,
                                start.vec){
# Find the value of the b s vector
# that specifies the Kabaila and Giri confidence interval,
# for a given value of lambda.
# This vector is found by numerical constrained
# optimization.
#
# Inputs:
# lambda: a positive tuning parameter
# rho: a known correlation
# t.alpha: quantile of the t distribution for m and alpha
# gams: set of gammas at which the coverage is
#       required to be greater than or equal to 1 - alpha
# d: the b and s functions are optimized in the interval
#    (0, d]
# m: degrees of freedom n - p
# n.ints: number of intervals in [0,d]
# knots: location of knots in [0, d]
# knots.all: location of knots in [-d, d]
# nodes: vector of Gauss Legendre quadrature nodes
# weights: vector of Gauss Legendre quadrature weights
# zvec: a vector of length N where outer integrand is
#       evaluated at
# wvec: g(zvec)/sqrt(m / (m + 1)) where g(z)=exp(z/2 - exp
#       (-z))
# psinu.zvec: f_m+1(g(z))*d(g(z))/dz evaluated at z=zvec
# h: step length
# cons: sqrt(2/m) * exp(lgamma((m+1)/2) - lgamma(m/2))
#       where
#       m is the degrees of freedom
# natural: equals to 1 for natural cubic spline
#          interpolation
#          or 0 for clamped cubic spline interpolation
# obj: 1 for definition 1 of SEL or 2 for definition 2 of
#      SEL
# start.vec: a starting vector to the optimization problem
#
# Output:
# The b s vector.
#
# Written by N Ranathunga in September 2020

# Specify lower and upper bounds on the vector of values
# of the b and s functions evaluated at the knots
low <- c(rep(-100, n.ints - 1), rep(0.5, n.ints))
up <- c(rep(100, n.ints - 1), rep(200, n.ints))

# Make the objective function a function of one argument, y
if (obj == 1) {
  obj_fun <- functional::Curry(objective1, lambda = lambda, m
    = m, n.ints = n.ints,
                                knots = knots, knots.all =

```

```

        knots.all, t.alpha = t.
        alpha,
        nodes = nodes, weights =
        weights, natural =
        natural)
} else {
  obj_fun <- functional::Curry(objective2, lambda = lambda, m
    = m, n.ints = n.ints,
        knots = knots, knots.all =
        knots.all, t.alpha = t.
        alpha,
        nodes = nodes, weights =
        weights, natural =
        natural)
}

# Make the constraint function a function of one argument,
# y
cons_fun <- functional::Curry(constraints_slsqp_trapez,
  gams = gams, rho = rho,
        n.ints = n.ints, knots =
        knots, knots.all = knots.
        all,
        alpha = alpha, t.alpha = t.
        alpha, nodes = nodes,
        weights = weights, wvec =
        wvec, psinu.zvec = psinu.
        zvec,
        h = h, cons = cons, natural =
        natural)

# Find the values of the knots using the optimization
# function
res <- nloptr::slsqp(start.vec, obj_fun, hin = cons_fun,
  lower = low,
  upper = up, nl.info = FALSE)
new.par <- res$par

# Output the vector with knot values which specifies the
# new
# confidence interval
out <- new.par
}

```

```

objective1 <- function(y, lambda, m, n.ints, knots, knots.all
,
                        t.alpha, nodes, weights, natural){
# Computes the objective function of the scaled expected
# length 1 of Kabaila and Giri confidence interval.
# The integral from (0, d) is broken down to integrals
# over knots. Each integral is computed using gauss
# legendre quadrature.
#
# Inputs:
# y: contains knots values of the b and s functions
# lambda: a positive tuning parameter
# m: degrees of freedom n - p
# d: the b and s functions are optimized in the interval
# (0, d]
# n.ints: number of intervals in (0, d]
# t.alpha: quantile of the t distribution for m and alpha
# knots: location of knots in [0, d]
# knots.all: location of knots in [-d, d]
# nodes: vector of Gauss Legendre quadrature nodes
# weights: vector of Gauss Legendre quadrature weights
# natural: equals to 1 for natural cubic spline
#          interpolation
#          or 0 for clamped cubic spline interpolation
#
# Output:
# The value of the objective function.
#
# Written by N Ranathunga in September 2020

s.spl <- spline_s(y, n.ints, knots.all, t.alpha, natural)

# Set up a vector to store the results
int <- rep(0, length(knots))

for(i in 1:(length(knots) - 1)){
# Specify bounds of the integral
a <- knots[i]
b <- knots[i+1]

# Find the approximate integral
adj.nodes <- ((b - a) / 2) * nodes + (a + b) / 2
q <- integrand_obj1(adj.nodes, lambda, m, t.alpha, s.spl)
int[i] <- ((b - a) / 2) * sum(weights * q)
}

out <- sum(int)
}

```

```

objective2 <- function(y, lambda, m, n.ints, knots, knots.all
,
                        t.alpha, nodes, weights, natural){
# Computes the objective function of the scaled expected
# length 2 of Kabaila and Giri confidence interval.
# The integral from (0, d) is broken down to integrals
# over knots. Each integral is computed using gauss
# legendre quadrature.
#
# Inputs:
# y: contains knots values of the b and s functions
# lambda: a positive tuning parameter
# m: degrees of freedom n - p
# d: the b and s functions are optimized in the interval
# (0, d]
# n.ints: number of intervals in (0, d]
# t.alpha: quantile of the t distribution for m and alpha
# knots: location of knots in [0, d]
# knots.all: location of knots in [-d, d]
# nodes: vector of Gauss Legendre quadrature nodes
# weights: vector of Gauss Legendre quadrature weights
# natural: equals to 1 for natural cubic spline
#          interpolation
#          or 0 for clamped cubic spline interpolation
#
# Output:
# The value of the objective function.
#
# Written by N Ranathunga in September 2020

s.spl <- spline_s(y, n.ints, knots.all, t.alpha, natural)

# Set up a vector to store the results
int <- rep(0, length(knots))

for(i in 1:(length(knots) - 1)){
# Specify bounds of the integral
a <- knots[i]
b <- knots[i+1]

# Find the approximate integral
adj.nodes <- ((b - a) / 2) * nodes + (a + b) / 2
q <- integrand_obj2(adj.nodes, lambda, m, t.alpha, s.spl)
int[i] <- ((b - a) / 2) * sum(weights * q)
}

out <- sum(int)
}

```

```

integrands_obj1 <- function(x, lambda, m, t.alpha, s.spl){
  # Evaluate the inner integrand of the objective function
  # of the scaled expected length 1 for a vector x.
  # In other words, this function computes
  #
  #  $(s(x) - t\_alpha) (\lambda +$ 
  #  $[(m/(x^2 + m))^{(m/2) + 1} / \sqrt{2 \pi}])$ 
  #
  # Inputs:
  # x: vector at which the integrand is to be evaluated
  # lambda: a positive tuning parameter
  # m: degrees of freedom n - p
  # t.alpha: quantile of the t distribution for m and alpha
  # s.spl: s function
  #
  # Output:
  # A vector of values of the inner integrand with the
  # same dimension as x.
  #
  # Written by N Ranathunga in September 2020

  tmp1 <- s.spl(x) - t.alpha
  term1 <- (1/sqrt(2 * pi)) * (m / (x^2 + m)) ^ (m/2 + 1)
  tmp2 <- lambda + term1

  res <- tmp1 * tmp2
}

```

```

integrands_obj2 <- function(x, lambda, m, t.alpha, s.spl){
  # Evaluate the inner integrand of the objective function
  # of the scaled expected length 2 for a vector x.
  # In other words, this function computes
  #
  #  $(s(x) - t\_alpha) (\lambda +$ 
  #  $[(m/(x^2 + m))^{(m+1)/2} / \sqrt{2 \pi}])$ 
  #
  # Inputs:
  # x: vector at which the integrand is to be evaluated
  # lambda: a positive tuning parameter
  # m: degrees of freedom n - p
  # t.alpha: quantile of the t distribution for m and alpha
  # s.spl: s function
  #
  # Output:
  # A vector of values of the inner integrand with the
  # same dimension as x.
  #
  # Written by N Ranathunga in September 2020

  tmp1 <- s.spl(x) - t.alpha
  term1 <- (1/sqrt(2 * pi)) * (m / (x^2 + m)) ^ ((m + 1)/2)
  tmp2 <- lambda + term1

  res <- tmp1 * tmp2
}

```

```

constraints_slsqp_trapez <- function(gams, rho, y, n.ints,
                                   knots, knots.all,
                                   alpha, t.alpha, nodes,
                                   weights, wvec,
                                   psinu.zvec, h, cons,
                                   natural){
  # This function computes (coverage probability) - (1 -
    alpha)
  # for a vector of gamma values.
  #
  # Inputs:
  # gams: set of gammas at which the coverage is
  #       required to be greater than or equal to 1 - alpha
  # rho: a known correlation
  # y: contains knots values of the b and s functions
  # n.ints: number of intervals in (0, d]
  # knots: location of knots in [0, d]
  # knots.all: location of knots in [-d, d]
  # alpha: nominal coverage is 1 - alpha
  # t.alpha: quantile of the t distribution for m and alpha
  # nodes: vector of Gauss Legendre quadrature nodes
  # weights: vector of Gauss Legendre quadrature weights
  # wvec: g(zvec)/sqrt(m / (m + 1)) where g(z)=exp(z/2 - exp
    (-z))
  # psinu.zvec: f_m+1(g(z))*d(g(z))/dz evaluated at z=zvec
  # h: step length
  # cons: sqrt(2/m) * exp(lgamma((m+1)/2) - lgamma(m/2))
    where
  #       m is the degrees of freedom
  # natural: equals to 1 for natural cubic spline
    interpolation
  #       or 0 for clamped cubic spline interpolation
  #
  # Output:
  # A vector of values of (coverage probability) - (1 - alpha
    )
  #
  # Written by N Ranathunga in September 2020

  len.gams <- length(gams)
  covs <- rep(0, len.gams)

  # Find b and s functions at y
  b.spl <- spline_b(y, n.ints, knots.all, t.alpha, natural)
  s.spl <- spline_s(y, n.ints, knots.all, t.alpha, natural)

  for(i in 1:len.gams){
    covs[i] <- compute_cov_trapez(gams[i], rho, knots, alpha,
      t.alpha,
      nodes, weights, b.spl, s.
        spl, wvec,
        psinu.zvec, h, cons)
  }

  out <- covs - (1 - alpha)
}

```



```

spline_b <- function(y, n.ints, knots.all, t.alpha, natural){
  # Return the value of the b function at a given point x.
  #
  # Inputs:
  # y: contains knot values of the b and s functions
  # n.ints: number of intervals in (0, d]
  # knots.all: location of knots in [-d, d]
  # t.alpha: quantile of the t distribution for m and alpha
  # natural: equals to 1 for natural cubic spline
  #           interpolation
  #           or 0 for clamped cubic spline interpolation
  #
  # Written by R Mainzer, March 2017
  # Modified by N Ranathunga in September 2020

  y.rev <- rev(y[1:(n.ints - 1)])
  b.vals <- c(0, y[1:(n.ints - 1)], 0)
  b.vals.all <- c(0, -y.rev, b.vals)

  # If natural = 1 use natural cubic spline, otherwise use
  #   clamped cubic
  # spline
  if(natural == 1){
    b.spl <- stats::splinefun(knots.all, b.vals.all, method =
      "natural")
  } else {
    b.spl.pp <- pracma::cubicspline(knots.all, b.vals.all,
      endp2nd = TRUE)
    b.spl <- function(x) pracma::ppval(b.spl.pp, x)
  }

  out <- b.spl
}

```

```

spline_s <- function(y, n.ints, knots.all, t.alpha, natural){
  # Return the value of the s function at a given point x.
  #
  # Inputs:
  # y: contains knot values of the b and s functions
  # n.ints: number of intervals in (0, d]
  # knots.all: location of knots in [-d, d]
  # t.alpha: quantile of the t distribution for m and alpha
  # natural: equals to 1 for natural cubic spline
  #           interpolation
  #           or 0 for clamped cubic spline interpolation
  #
  # Written by R Mainzer, March 2017
  # Modified by N Ranathunga in September 2020

  s.vals <- c(y[n.ints:(2 * n.ints - 1)], t.alpha)
  s.vals.all <- c(rev(s.vals), s.vals[2:(n.ints+1)])

  if(natural == 1){
    s.spl <- stats::splinefun(knots.all, s.vals.all, method =
      "natural")
  } else {
    s.spl.pp <- pracma::cubicspline(knots.all, s.vals.all,
      endp2nd = TRUE)
    s.spl <- function(x) pracma::ppval(s.spl.pp, x)
  }

  out <- s.spl
}

```

```

compute_cov_trapez <- function(gam, rho, knots, alpha, t.
  alpha,
                                nodes, weights, b.spl, s.spl,
                                wvec,
                                psinu.zvec, h, cons){
  # Compute the coverage probability of the Kabaila and Giri
  # confidence interval.
  # Apply the transformation (2.6) of Mori (1988), followed
  # by
  # the trapezoidal rule to the outer integral.
  #
  # Inputs:
  # gam: parameter
  # rho: a known correlation
  # knots: location of knots in [0, d]
  # alpha: nominal coverage is 1 - alpha
  # t.alpha: quantile of the t distribution for m and alpha
  # nodes: vector of Gauss Legendre quadrature nodes
  # weights: vector of Gauss Legendre quadrature weights
  # b.spl: b function
  # s.spl: s function
  # wvec: g(zvec)/sqrt(m / (m + 1)) where g(z)=exp(z/2 - exp
  #       (-z))
  # psinu.zvec: f_m+1(g(z))*d(g(z))/dz evaluated at z=zvec
  # h: step length
  # cons: sqrt(2/m) * exp(lgamma((m+1)/2) - lgamma(m/2))
  #       where
  #       m is the degrees of freedom
  #
  # Written by N Ranathunga, September 2020

  # Set up a vector to store the results of ICP
  ICP.zvec <- rep(0, length(wvec))

  for(i in 1:length(wvec)){
    w <- wvec[i]
    ICP.zvec[i] <- ICP_legendre(gam, rho, w, knots, t.alpha,
                                nodes, weights, b.spl, s.spl)
  }

  out.int <- h * PreciseSums::kahanSum(ICP.zvec * psinu.zvec)

  cp <- (1 - alpha) + cons * out.int
}

```

```

ICP_legendre <- function(gam, rho, w, knots, t.alpha,
                        nodes, weights, b.spl, s.spl){
  # Compute the inner integral of the coverage probability
  # of Kabaila and Giri confidence interval.
  # The integral from (0, d) is broken down to integrals
  # over knots. Each integral is computed using gauss
  # legendre quadrature. The number of nodes and weights for
  # the approximation of each integral can be changed.
  #
  # Input:
  # gam: parameter
  # rho: a known correlation
  # w: a value of the variable of integration in the
  #     outer integral
  # knots: location of knots in [0, d]
  # t.alpha: quantile of the t distribution for m and alpha
  # nodes: vector of Gauss Legendre quadrature nodes
  # weights: vector of Gauss Legendre quadrature weights
  # b.spl: b function
  # s.spl: s function
  #
  # Output:
  # A value for the inner integral of Kabaila and Giri
  # confidence interval.
  #
  # Written by N. Ranathunga in September 2020

  # Set up a vector to store the results
  int <- rep(0, length(knots))

  for(i in 1:(length(knots) - 1)){
    # Specify bounds of the integral
    a <- knots[i]
    b <- knots[i+1]

    # Find the approximate integral
    adj.nodes <- ((b - a) / 2) * nodes + (a + b) / 2
    q <- IICP(adj.nodes, gam, rho, w, t.alpha, b.spl, s.spl)
    int[i] <- ((b - a) / 2) * sum(weights * q)
  }

  ICP <- sum(int)
}

```

```

IICP <- function(x, gam, rho, w, t.alpha, b.spl, s.spl){
  # Compute the function
  # (k(x, w, gam, rho) - k_dag(x, w, gam, rho)) *
  # phi(wx - gam) +
  # (k(-x, w, gam, rho) - k_dag(-x, w, gam, rho)) *
  # phi(wx + gam)
  # for a vector x.
  #
  # Inputs:
  # x: vector of nodes of the Gauss Legendre quadrature
  # gam: parameter
  # rho: a known correlation
  # w: a value of the variable of integration in the
  #     outer integral
  # t.alpha: quantile of the t distribution for m and alpha
  # b.spl: b function
  # s.spl: s function
  #
  # Output:
  # A vector with the same dimension as x.
  #
  # Written by N. Ranathunga in September 2020

  # Finding k_dag(x, w, gam, rho))
  mu1 <- rho * (w*x - gam)
  var <- 1 - rho^2
  k.dag1 <- Psi(-t.alpha * w, t.alpha * w, mu1, var)

  # Finding k(x, w, gam, rho)
  term.a1 <- b.spl(x)
  term.b1 <- s.spl(x)
  lh <- w * (term.a1 - term.b1)
  uh <- w * (term.a1 + term.b1)
  k1 <- Psi(lh, uh, mu1, var)

  # Finding phi(wx - gam)
  term1 <- stats::dnorm(w*x - gam, 0, 1)

  # Finding k_dag(-x, w, gam, rho))
  mu2 <- rho * (-w*x - gam)
  k.dag2 <- Psi(-t.alpha * w, t.alpha * w, mu2, var)

  # Finding k(-x, w, gam, rho)
  term.a2 <- b.spl(-x)
  term.b2 <- s.spl(-x)
  lh2 <- w * (term.a2 - term.b2)
  uh2 <- w * (term.a2 + term.b2)
  k2 <- Psi(lh2, uh2, mu2, var)

  # Finding phi(wx + gam)
  term2 <- stats::dnorm(w*x + gam, 0, 1)

  res <- (k1 - k.dag1) * term1 + (k2 - k.dag2) * term2
}

```

```

Psi <- function(x, y, mu, variance){
  # This function calculates
  # Psi(x,y,mu,variance) which equals to
  #  $P(x \leq Z \leq y) = P(Z \leq y) - P(Z \leq x)$ 
  # where  $Z \sim N(\mu, \text{variance})$ .
  #
  # Inputs:
  # x: given value
  # y: given value that is greater than or equal
  #     to x
  # mu: mean of the normal distribution
  # variance: variance of the normal distribution
  #
  # Output:
  # A value for Psi(x,y,mu,variance)
  #
  # Written by N. Ranathunga in September 2020

  sigma <- sqrt(variance)
  term1 <- stats::pnorm(y, mean = mu, sd = sigma)
  term2 <- stats::pnorm(x, mean = mu, sd = sigma)
  out <- term1 - term2
}

```

```

startOptim <- function(n.ints, t.alpha){
  # Calculate a vector which provides
  # starting values to optimize the objective
  # function.
  #
  # Inputs:
  # n.ints: number of intervals in (0, d]
  # t.alpha: quantile of the t distribution for
  #           m and alpha
  #
  # Output:
  # A vector of length 2*n.ints - 1.
  #
  # Written by N Ranathunga in September 2020

  out <- c(rep(0, n.ints - 1), rep(t.alpha, n.ints))
}

```

```

OuterPara <- function(m, nu, N, eps){
  # Compute a list of values and vectors which will
  # be used as inputs to compute the outer integrals
  # of the coverage probability and the two squared scaled
  # expected lengths.
  #
  # Inputs:
  # m: degrees of freedom n - p
  # nu: m + positive integer
  # N: number of evaluations in the outer integrand
  # eps: upper bound of the approximation error
  #
  # Output:
  # A list of values and vectors.
  #
  # Written by N Ranathunga in September 2020

  # Step length to compute the outer integral
  cm <- sqrt(m / nu)
  d2 <- as.numeric(compute_zl(nu, eps)[1])
  z1 <- as.numeric(compute_zl(nu, eps)[2])
  zu <- z1 + d2
  h <- d2/(N - 1)

  # vectors which will be used to compute
  # the outer integral
  zvec <- seq(z1, zu, by = h)
  wvec <- transf(zvec) / cm
  psinu.zvec <- Func_psi_nu(zvec, nu)

  # The constant term in CP for nu=m+1 or
  # the constant term in SEL2 for nu=m+1
  cons1 <- sqrt(2/m) * exp(lgamma(nu/2) - lgamma(m/2))

  # The constant term in SEL1 for nu=m+2
  cons2 <- (2/m) * exp(lgamma(nu/2) - lgamma(m/2))

  # E(W) term in SEL1
  exp.w <- sqrt(2/m) * exp(lgamma((m + 1)/2) - lgamma(m/2))

  out <- list(h=h, cons1=cons1, cons2=cons2, exp.w=exp.w,
             wvec=wvec, psinu.zvec=psinu.zvec)
}

```

```

compute_zl <- function(nu, eps){
  # Compute the length of the interval to apply
  # the trapezoidal rule to the outer integral and
  # compute the location of the first evaluation
  # of the outer integrand.
  #
  # Input:
  # nu: degrees of freedom + positive integer
  # eps: upper bound of the error when approximate the
  #       outer integral by the trapezoidal rule
  #
  # Output:
  # A list of two values
  #
  # Written by N Ranathunga, September 2020

  # Find the length of the interval
  d2 <- stats::uniroot(MinUpBndTrErrMinEpsFin, nu, eps,
                      interval = c(0, 10), extendInt="yes")$root

  # Find the location of the first evaluation
  zl <- stats::optimize(UpBoundTrimError, d2, nu, interval=c
                        (-5, 1))$minimum

  out <- list(d2, zl)
}

```

```

Func_psi_nu <- function(zvec, nu){
  # This function computes a vector of values
  # of  $f_{\text{nu}}(g(z)) * dg(z)/dz$  where  $g(z) = \exp(z/2 - \exp(-z))$ 
  # and  $f_{\text{nu}}$  is the pdf of a random variable with the
  # same distribution as  $\sqrt{Q/\text{nu}}$  where  $Q \sim \text{chisq}(\text{nu})$ 
  # and  $\text{nu} = m + \text{positive integer}$ 
  #
  # Input:
  # zvec: a vector where function evaluations are at
  #       when applying the trapezoidal rule
  # nu: degrees of freedom + positive integer
  #
  # Output:
  # A vector of values of  $f_{\text{nu}}(g(z)) * dg(z)/dz$ 
  # with the same dimension as zvec.
  #
  # Written by N. Ranathunga in September 2020

  const <- exp( (nu/2) * log(nu) - lgamma(nu/2) -
                ((nu/2) - 1) * log(2) )
  tmp1 <- exp(-zvec)
  term1 <- exp(nu*zvec/2 - nu*tmp1 - (nu/2)*exp(zvec - 2*tmp1
  ))
  term2 <- 1/2 + tmp1

  psinu.val <- const * term1 * term2
}

```



```

transf <- function(z){
  # This function applies the
  # transformation (2.6) of Mori(1988)
  # to the value z
  #
  # Inputs:
  # z: a given value
  #
  # Written by N. Ranathunga in September 2020

  out <- exp((z/2) - exp(-z))
}

```

```

MinUpBndTrErrMinEpsFin <- function(d2, nu, eps){
  # This function minimizes
  # (upper bound of the trimming error) - (10(-3) * eps)
  # when approximated the outer integral
  # by the trapezoidal rule.
  #
  # Input:
  # d2: (number of evaluations of the outer
  #      integrand) * (step length)
  # nu: degrees of freedom + positive integer
  # eps: a given value for the upper bound of the
  #      approximation error
  #
  # Written by N Ranathunga, September 2020

  temp <- stats::optimize(UpBoundTrimError, d2, nu,
                          interval=c(-4, 4))

  out <- temp$objective - (10(-3) * eps)
}

```

```

UpBoundTrimError <- function(z, d2, nu){
  # Computes the upper bound of the
  # trimming error when approximate
  # the outer integral by the trapezoidal rule.
  #
  # Input:
  # z: a given value
  # d2: (number of evaluations of the outer
  #      integrand) * (step length)
  # nu: degrees of freedom + positive integer
  #
  # Written by N Ranathunga, September 2020

  x1 <- transf(z)
  x2 <- transf(z + d2)
  term1 <- nu * x12
  term2 <- nu * x22

  out <- stats::pchisq(term1, df=nu) + 1 -
        stats::pchisq(term2, df=nu)
}

```

C.7.2 R program for the computation of the graphs of the functions b_1 and s_1

```

bsspline2 <- function(x, bsvec, alpha, m, natural = 1){
  # Evaluate the functions b and s at x.
  #
  # Inputs:
  # x: a value or vector of values at which the functions
  #     b and s are to be evaluated
  # bsvec: the vector (b(d/6),...,b(5d/6),s(0),...,s(5d/6))
  #         where d is a positive number
  # alpha: minimum coverage probability is 1 - alpha
  # m: degrees of freedom n - p
  # natural: equal to 1 (default) if the functions b and s
  #           are found by natural cubic spline interpolation
  #           or 0 if these functions are found by clamped
  #           cubic
  #           spline interpolation in the interval [-d, d]
  #
  # Output:
  # A data frame containing x and the corresponding values
  # of the functions b and s.
  #
  # Written by N Ranathunga in September 2020

  # Set input
  n.ints <- 6
  d <- choice_d(m)$d

  # Specify where the knots for b and s are located
  knots.all <- seq(-d, d, by = d/n.ints)

  # Set t.alpha for m and alpha
  t.alpha <- stats::qt(1 - alpha/2, m)

  # Find b and s functions
  sspl <- spline_s(bsvec, n.ints, knots.all, t.alpha, natural
  )
  bspl <- spline_b(bsvec, n.ints, knots.all, t.alpha, natural
  )

  x1 <- x[which(x <= -d)]
  x2 <- x[which(x > -d & x < d)]
  x3 <- x[which(x >= d)]

  bspl.res <- c(rep(0, length(x1)), bspl(x2), rep(0, length(
    x3)))
  sspl.res <- c(rep(t.alpha, length(x1)), sspl(x2), rep(t.
    alpha, length(x3)))

  out <- data.frame(x = x, b = bspl.res, s = sspl.res)
}

```

C.7.3 R program for the computation of the graph of the coverage probability $CP(\gamma; b_1, s_1)$

```

cpciuupi2 <- function(gam, bsvec, alpha, m, rho, natural = 1)
{
  # Compute the coverage probability of the CIUUPI for
  # unknown error variance
  #
  # Inputs:
  # gam: a value of gamma or vector of gamma values at which
  #       the coverage probability function is evaluated
  # bsvec: the vector (b(d/6),...,b(5d/6),s(0),...,s(5d/6))
  #         where d is a positive number
  # alpha: minimum coverage probability is 1 - alpha
  # m: degrees of freedom n - p
  # rho: a known correlation
  # natural: equal to 1 (default) if the functions b and s
  #           are found by natural cubic spline interpolation
  #           or 0 if these functions are found by clamped
  #           cubic
  #           spline interpolation in the interval [-d, d]
  #
  # Output:
  # The value(s) of the coverage probability of the
  # CIUUPI for unknown error variance at gam.
  #
  # Written by N Ranathunga in September 2020

  # Specify the values of the inputs to other functions
  n.ints <- 6
  N <- 33
  eps <- 10^{-10}
  n.nodes <- 20
  d <- choice_d(m)$d

  # Set t.alpha for m and alpha
  t.alpha <- stats::qt(1 - alpha/2, m)

  # Find the nodes and weights of the legendre quadrature
  quad.info <- statmod::gauss.quad(n.nodes, kind="legendre")
  nodes <- quad.info$nodes
  weights <- quad.info$weights

  # Specify where the knots for b and s are located
  # as inputs to other functions
  knots <- seq(0, d, by = d/n.ints)
  knots.all <- seq(-d, d, by = d/n.ints)

  # Specify the values of the inputs to compute the
  # outer integral in coverage probability
  h <- OuterPara(m, nu=m+1, N, eps)$h
  wvec <- OuterPara(m, nu=m+1, N, eps)$wvec
  psinu.zvec <- OuterPara(m, nu=m+1, N, eps)$psinu.zvec
  cons <- OuterPara(m, nu=m+1, N, eps)$cons1

  # Find the b and s functions

```

```

b.spl <- spline_b(bsvec, n.ints, knots.all, t.alpha,
  natural)
s.spl <- spline_s(bsvec, n.ints, knots.all, t.alpha,
  natural)

# Compute the coverage probability
res <- rep(0, length(gam))
for(i in 1:length(gam)){
  res[i] <- compute_cov_trapez(gam[i], rho, knots, alpha,
    t.alpha, nodes, weights, b.spl, s.spl
    ,
    wvec, psinu.zvec, h, cons)
}

out <- res
}

```

C.7.4 R programs for the computation of the graph of the first definition of scaled expected length

$$\text{SEL}_1(\gamma; s_1)$$

```
sel1ciuupi2 <- function(gam, bsvec, alpha, m, rho, natural =
1){
# Compute the first definition of the scaled expected
# length of the CIUUPI for unknown error variance.
#
# Inputs:
# gam: a value of gamma or vector of gamma values at which
#       the coverage probability function is evaluated
# bsvec: the vector (b(d/6),...,b(5d/6),s(0),...,s(5d/6))
#       where d is a positive number
# alpha: minimum coverage probability is 1 - alpha
# m: degrees of freedom n - p
# rho: a known correlation
# natural: equal to 1 (default) if the functions b and s
#          are found by natural cubic spline interpolation
#          or 0 if these functions are found by clamped
#          cubic
#          spline interpolation in the interval [-d, d]
#
# Output:
# The value(s) of the first definition of the scaled
# expected
# length of the CIUUPI for unknown error variance at gam.
#
# Written by N Ranathunga in September 2020

# Specify the values of the inputs to other functions
n.ints <- 6
N <- 33
eps <- 10^{-10}
n.nodes <- 20
d <- choice_d(m)$d

# Set t.alpha for m and alpha
t.alpha <- stats::qt(1 - alpha/2, m)

# Find the nodes and weights of the legendre quadrature
quad.info <- statmod::gauss.quad(n.nodes, kind="legendre")
nodes <- quad.info$nodes
weights <- quad.info$weights

# Specify where the knots for b and s are located
# as inputs to other functions
knots <- seq(0, d, by = d/n.ints)
knots.all <- seq(-d, d, by = d/n.ints)

# Specify the values of the inputs to compute the
# outer integral in SEL1
h1 <- OuterPara(m, nu=m+2, N, eps)$h
wvec <- OuterPara(m, nu=m+2, N, eps)$wvec
psinu.zvec <- OuterPara(m, nu=m+2, N, eps)$psinu.zvec
```

```

cons <- OuterPara(m, nu=m+2, N, eps)$cons2
exp.w <- OuterPara(m, nu=m+2, N, eps)$exp.w

# Specify the function s
s.spl <- spline_s(bsvec, n.ints, knots.all, t.alpha,
  natural)

# Compute the scaled expected length
res <- rep(0, length(gam))
for(i in 1:length(gam)){
  res[i] <- compute_sel1_trapez(gam[i], knots, t.alpha,
    nodes, weights,
                                s.spl, wvec, psinu.zvec, h1
                                , cons, exp.w)
}

out <- res
}

```

```

compute_sel1_trapez <- function(gam, knots, t.alpha, nodes,
                               weights, s.spl, wvec, psinu.
                               zvec,
                               h1, cons, exp.w){
  # Compute the value of the scaled expected length 1 for
  # given functions b and s.
  # In other words, this function computes
  #
  #  $1 + (1/(t.alpha E(W))) \int_0^\infty (ISEL1(w, gam)) w^2 f_W(w) dw$ 
  #
  # Inputs:
  # gam: parameter
  # knots: location of knots in [0, d]
  # t.alpha: quantile of the t distribution for m and alpha
  # nodes: vector of Gauss Legendre quadrature nodes
  # weights: vector of Gauss Legendre quadrature weights
  # s.spl: s function
  # wvec:  $g(z)/\sqrt{m / (m + 2)}$  where  $g(z)=\exp(z/2 - \exp(-z))$ 
  #        evaluated at  $z=zvec$ 
  # psinu.zvec:  $f_{m+2}(g(z))*d(g(z))/dz$  evaluated at  $z=zvec$ 
  # h1: step length
  # cons:  $(2/m) * \exp(\lgamma((m+2)/2) - \lgamma(m/2))$  where
  #        m is the degrees of freedom
  # exp.w:  $\sqrt{2/m} * \exp(\lgamma((m + 1)/2) - \lgamma(m/2))$ 
  #
  # Output:
  # The scaled expected length 1 for given functions b and s.
  #
  # Written by N Ranathunga, September 2020

  # Set up a vector to store the results of ISEL1
  ISEL1.zvec <- rep(0, length(wvec))

  for(i in 1:length(wvec)){
    w <- wvec[i]
    ISEL1.zvec[i] <- ISEL_legendre(gam, w, knots, t.alpha,
                                   nodes, weights, s.spl)
  }

  out.int <- h1 * PreciseSums::kahanSum(ISEL1.zvec * psinu.
                                         zvec)

  sel1 <- 1 + (cons * out.int) / (t.alpha * exp.w)
}

```

```

ISEL_legendre <- function(gam, w, knots, t.alpha,
                          nodes, weights, s.spl){
  # Computes the inner integral of the scaled expected
  # length 1 and 2 of Kabaila and Giri confidence interval.
  # The integral from (0, d) is broken down to integrals
  # over knots. Each integral is computed using gauss
  # legendre quadrature. The number of nodes and weights
  # for the approximation of each integral can be changed.
  #
  # Inputs:
  # gam: parameter
  # w: a value of the variable of integration in the
  #     outer integral
  # knots: location of knots in [0, d]
  # t.alpha: quantile of the t distribution for m and alpha
  # nodes: vector of Gauss Legendre quadrature nodes
  # weights: vector of Gauss Legendre quadrature weights
  # s.spl: s function
  #
  # Output:
  # A value for the inner integral of the scaled expected
  # length 1 and 2
  #
  # Written by N.Ranathunga in September 2020.

  # Set up a vector to store the results
  int <- rep(0, length(knots))

  for(i in 1:(length(knots) - 1)){
    # Specify bounds of the integral
    a <- knots[i]
    b <- knots[i+1]

    # Find the approximate integral
    adj.nodes <- ((b - a) / 2) * nodes + (a + b) / 2
    q <- IISEL(adj.nodes, gam, w, t.alpha, s.spl)
    int[i] <- ((b - a) / 2) * sum(weights * q)
  }

  ISEL <- sum(int)
}

```



```

IISEL <- function(x, gam, w, t.alpha, s.spl){
  # Evaluates the function
  # (s(x) - t_alpha) * (phi(wx-gamma) + phi(wx+gamma))
  # for a vector x.
  #
  # Inputs:
  # x: vector of nodes of the Gauss Legendre quadrature
  # gam: parameter
  # m: degrees of freedom n - p
  # w: a value of the variable of integration in the
  #     outer integral
  # t.alpha: quantile of the t distribution for m and alpha
  # s.spl: s function
  #
  # Output:
  # A vector with the same dimension as x.
  #
  # Written by N.Ranathunga in September 2020.

  tmp1 <- s.spl(x) - t.alpha
  tmp2 <- stats::dnorm(w*x - gam, 0, 1) +
    stats::dnorm(w*x + gam, 0, 1)
  res <- tmp1 * tmp2
}

```

C.7.5 R programs for the computation of the graph of the second definition of scaled expected length

$$\text{SEL}_2(\gamma; s_2)$$

```
sel2ciuupi2 <- function(gam, bsvec, alpha, m, rho, natural =
1){
# Compute the second definition of the scaled expected
# length of the CIUUPI for unknown error variance.
#
# Inputs:
# gam: a value of gamma or vector of gamma values at which
#       the coverage probability function is evaluated
# bsvec: the vector (b(d/6),...,b(5d/6),s(0),...,s(5d/6))
#         where d is a positive number
# alpha: minimum coverage probability is 1 - alpha
# m: degrees of freedom n - p
# rho: a known correlation
# natural: equal to 1 (default) if the functions b and s
#           are found by natural cubic spline interpolation
#           or 0 if these functions are found by clamped
#           cubic
#           spline interpolation in the interval [-d, d]
#
# Output:
# The value(s) of the second definition of the scaled
# expected
# length of the CIUUPI for unknown error variance at gam.
#
# Written by N Ranathunga in September 2020

# Specify the values of the inputs to other functions
n.ints <- 6
N <- 33
eps <- 10^{-10}
n.nodes <- 20
d <- choice_d(m)$d

# Set t.alpha for m and alpha
t.alpha <- stats::qt(1 - alpha/2, m)

# Find the nodes and weights of the legendre quadrature
quad.info <- statmod::gauss.quad(n.nodes, kind="legendre")
nodes <- quad.info$nodes
weights <- quad.info$weights

# Specify where the knots for b and s are located
# as inputs to other functions
knots <- seq(0, d, by = d/n.ints)
knots.all <- seq(-d, d, by = d/n.ints)

# Specify the values of the inputs to compute the
# outer integral in coverage probability
h2 <- OuterPara(m, nu=m+1, N, eps)$h
wvec <- OuterPara(m, nu=m+1, N, eps)$wvec
psinu.zvec <- OuterPara(m, nu=m+1, N, eps)$psinu.zvec
```

```

cons <- OuterPara(m, nu=m+1, N, eps)$cons1

# Specify the function s
s.spl <- spline_s(bsvec, n.ints, knots.all, t.alpha,
  natural)

# Compute the scaled expected length
res <- rep(0, length(gam))
for(i in 1:length(gam)){
  res[i] <- compute_sel2_trapez(gam[i], knots, t.alpha,
    nodes, weights,
                                s.spl, wvec, psinu.zvec, h2
                                , cons)
}

out <- res
}

```

```

compute_sel2_trapez <- function(gam, knots, t.alpha, nodes,
                               weights,
                               s.spl, wvec, psinu.zvec, h2,
                               cons){
  # Compute the value of the scaled expected
  # length 2 for given functions b and s.
  # In other words, this function computes
  #
  #  $1 + (1/t\_alpha) \int_0^\infty (ISEL(w, gam)) w f_m(w) dw$ 
  #
  # Inputs:
  # gam: parameter
  # knots: location of knots in [0, d]
  # t.alpha: quantile of the t distribution for m and alpha
  # nodes: vector of Gauss Legendre quadrature nodes
  # weights: vector of Gauss Legendre quadrature weights
  # s.spl: s function
  # wvec:  $g(z)/\sqrt{m / (m + 1)}$  where  $g(z)=\exp(z/2 - \exp(-z))$ 
  #         evaluated at  $z=zvec$ 
  # psinu.zvec:  $f_{m+1}(g(z))*d(g(z))/dz$  evaluated at  $z=zvec$ 
  # h2: step length
  # cons:  $\sqrt{2/m} * \exp(\lgamma((m+1)/2) - \lgamma(m/2))$ 
  #         where
  #         m is the degrees of freedom
  #
  # Output:
  # The scaled expected length 2 for given functions b and s.
  #
  # Written by N Ranathunga, September 2020

  # Set up a vector to store the results of ISEL2
  ISEL2.zvec <- rep(0, length(wvec))

  for(i in 1:length(wvec)){
    w <- wvec[i]
    ISEL2.zvec[i] <- ISEL_legendre(gam, w, knots, t.alpha,
                                   nodes, weights, s.spl)
  }

  out.int <- h2 * PreciseSums::kahanSum(ISEL2.zvec * psinu.
                                         zvec)

  sel2 <- 1 + (cons * out.int) / t.alpha
}

```

Appendix D

Proofs and R programs for Chapter 5

D.1 Derivation of the mode and the variance of the posterior distribution

In Section 5.3, we approximate $g(t; \theta, \sigma) \phi(t; 0, \sigma^2)$ by $\phi(t; \mu, \tau^2)$, where

$\mu = \text{mode of } g(t; \theta, \sigma) \phi(t; 0, \sigma^2)$, considered as a function of t ,

and

$$\tau^2 = \left[-\frac{\partial^2}{\partial t^2} (\log [g(t; \theta, \sigma) \phi(t; 0, \sigma^2)]) \right]^{-1}.$$

In the present section, we derive formulas for μ and τ^2 .

We need to find the value of t maximising $g(t; \theta, \sigma) \phi(t; 0, \sigma^2)$. However, it is algebraically easier to find the value of t maximising $\log (g(t; \theta, \sigma) \phi(t; 0, \sigma^2))$, where

$$\begin{aligned} & \log (g(t; \theta, \sigma) \phi(t; 0, \sigma^2)) \\ &= \log \left(\frac{\exp ((\theta + t)y)}{(1 + \exp(\theta + t))^J} \frac{1}{\sqrt{2\pi} \sigma} \exp \left(\frac{-t^2}{2\sigma^2} \right) \right) \end{aligned}$$

$$= (\theta + t) y - J \log (1 + \exp(\theta + t)) - \frac{t^2}{2\sigma^2} - \log(\sqrt{2\pi} \sigma). \quad (\text{D.1})$$

For given J, y, θ and σ , we maximise (D.1) numerically with respect to t using the R function `optimize`. This value of t maximising (D.1) is the mode.

Note that

$$\frac{\partial}{\partial t} [\log (g(t; \theta, \sigma) \phi(t; 0, \sigma^2))] = y - \frac{J}{1 + \exp(\theta + t)} \exp(\theta + t) - \frac{t}{\sigma^2},$$

so that

$$\frac{\partial^2}{\partial t^2} [\log (g(t; \theta, \sigma) \phi(t; 0, \sigma^2))] = -J \times \frac{\exp(\theta + t)}{(1 + \exp(\theta + t))^2} - \frac{1}{\sigma^2}.$$

Therefore,

$$\tau^2 = \left(- \left[-J \times \frac{\exp(\theta + t)}{(1 + \exp(\theta + t))^2} - \frac{1}{\sigma^2} \right] \right)^{-1},$$

where t is the mode found by maximising (D.1).

D.2 An expression for $c(\theta, \sigma)$ needed for the application of Gauss-Hermite quadrature

We change the variable of integration in (5.5) to $z = (t - \mu)/(\sqrt{2}\tau)$ so that $t = \mu + \sqrt{2}\tau z$ and

$$\begin{aligned} (5.5) &= \int_{-\infty}^{\infty} h(t; \theta, \sigma) \frac{1}{\sqrt{2\pi}\tau} \exp\left(-\frac{(t - \mu)^2}{2\tau^2}\right) dt \\ &= \int_{-\infty}^{\infty} h(\mu + \sqrt{2}\tau z; \theta, \sigma) \frac{1}{\sqrt{2\pi}\tau} \exp\left(-\frac{(\sqrt{2}\tau z)^2}{2\tau^2}\right) \sqrt{2}\tau dz \\ &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} h(\mu + \sqrt{2}\tau z; \theta, \sigma) \exp(-z^2) dz. \end{aligned}$$

D.3 An expression for $r(t; \theta, \sigma)$

The function $r(t; \theta, \sigma)$ is defined in subsection 5.4.2. In the present subsection, we derive the simplified expression for this function stated in subsection 5.4.2.

$$\begin{aligned}
r(t; \theta, \sigma) &= h(t; \theta, \sigma) (\phi(t; \mu, \tau^2))^{1/2} \\
&= \frac{g(t; \theta, \sigma) \phi(t; 0, \sigma^2)}{(\phi(t; \mu, \tau^2))^{1/2}} \\
&= \frac{\left(\exp((\theta + t)y) / (1 + \exp(\theta + t))^J \right) (1/\sqrt{2\pi} \sigma) \exp(-t^2/2\sigma^2)}{(1/(\sqrt{2\pi} \tau)^{1/2}) \exp(-(t - \mu)^2/4\tau^2)} \\
&= \left(\frac{\tau}{\sqrt{2\pi}\sigma^2} \right)^{1/2} \frac{\exp\left((\theta + t)y - (t^2/2\sigma^2) + (t^2 - 2\mu t + \mu^2)/4\tau^2 \right)}{(1 + \exp(\theta + t))^J} \\
&= c_0 \frac{\exp\left(\theta y + (\mu^2/(4\tau^2)) + \left(y - (\mu/(2\tau^2)) \right) t + \left(1/(4\tau^2) - 1/(2\sigma^2) \right) t^2 \right)}{(1 + \exp(\theta + t))^J} \\
&= c_0 \frac{\exp(c_1 + c_2 t + c_3 t^2)}{(1 + \exp(\theta + t))^J} \tag{D.2}
\end{aligned}$$

where

$$c_0 = \left(\frac{\tau}{\sqrt{2\pi}\sigma^2} \right)^{1/2}, \quad c_1 = \theta y + \frac{\mu^2}{4\tau^2}, \quad c_2 = y - \frac{\mu}{2\tau^2} \quad \text{and} \quad c_3 = \frac{1}{4\tau^2} - \frac{1}{2\sigma^2}.$$

D.4 Examination of the possible values for c_3

It follows from

$$\log(g(t; \theta, \sigma) \phi(t; 0, \sigma^2)) = \log(g(t; \theta, \sigma)) - \frac{t^2}{2\sigma^2} - \log \sqrt{2\pi\sigma^2}$$

that

$$-\frac{\partial^2}{\partial t^2} \log(g(t; \theta, \sigma) \phi(t; 0, \sigma^2)) = \frac{1}{\sigma^2} - \frac{\partial^2}{\partial t^2} \log(g(t; \theta, \sigma)).$$

Since $\partial^2 \log(g(t; \theta, \sigma))/\partial t^2$ evaluated at the mode is less than zero,

$$\begin{aligned}
& -\frac{\partial^2}{\partial t^2} \log (g(t; \theta, \sigma) \phi(t; 0, \sigma^2)) > \frac{1}{\sigma^2} \\
\iff & \left(-\frac{\partial^2}{\partial t^2} (\log (g(t; \theta, \sigma) \phi(t; 0, \sigma^2))) \right)^{-1} < \sigma^2 \\
\iff & \tau^2 < \sigma^2.
\end{aligned}$$

Recall that $c_3 = 1/4\tau^2 - 1/2\sigma^2$. Since $\tau^2 < \sigma^2$, $c_3 > -1/(4\sigma^2)$. Now we consider the following three possible cases for the values of c_3 .

Case 1: $c_3 = 0$

When $J \neq 0$,

$$(D.2) = c_0 \frac{\exp(c_1 + c_2 t)}{(1 + \exp(\theta + t))^J}. \quad (D.3)$$

For $c_3 = 0$, (D.3) depends on the value of c_2 . If $c_2 > 0$, (D.3) $\rightarrow 0$ as $|t| \rightarrow \infty$.

If $c_2 < 0$ (D.3) $\rightarrow 0$ as $t \rightarrow \infty$ and (D.3) $\rightarrow \infty$ as $t \rightarrow -\infty$.

Suppose now that $c_2 = 0$. In this case, (D.3) depends on the value of c_1 . If $c_1 > 0$, (D.3) $\rightarrow 0$ as $t \rightarrow \infty$ and (D.3) \rightarrow constant as $t \rightarrow -\infty$ and the value of the constant depends on the values of c_0 and c_1 (note that $c_0 > 0$). If $c_1 \leq 0$, (D.3) $\rightarrow 0$ as $t \rightarrow \infty$ and (D.3) \rightarrow constant as $t \rightarrow -\infty$ and the value of the constant depends on the values of c_0 and c_1 .

Case 2: $c_3 > 0$

When $J \neq 0$,

$$(D.2) = c_0 \frac{\exp(c_1 + c_2 t + c_3 t^2)}{(1 + \exp(\theta + t))^J} \quad (D.4)$$

and (D.4) $\rightarrow \infty$ as $|t| \rightarrow \infty$.

Case 3: $c_3 < 0$

When $J \neq 0$,

$$(D.2) = c_0 \frac{\exp(c_1 + c_2 t + c_3 t^2)}{(1 + \exp(\theta + t))^J} \quad (D.5)$$

and (D.5) $\rightarrow 0$ as $|t| \rightarrow \infty$.

D.5 Computation of the maximum likelihood estimates of the parameters of the logistic regression model with random intercept

Recall that the probability mass function (pmf) of y is the one-dimensional integral

$$\int_{-\infty}^{\infty} g(t; \theta, \sigma) \phi(t; 0, \sigma^2) dt.$$

We change the variable of integration to $u = t/(\sqrt{2}\sigma)$. Then, for a specific cluster i , the pmf of y is the one-dimensional integral

$$\frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} g(u_i; \theta, \sigma) \exp(-u_i^2) du_i.$$

Therefore, the pmf of \mathbf{y} is the following product of N one-dimensional integrals

$$\begin{aligned} & \prod_{i=1}^N \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} g(u_i; \theta, \sigma) \exp(-u_i^2) du_i \\ &= \frac{1}{\pi^{N/2}} \prod_{i=1}^N \int_{-\infty}^{\infty} g(u_i; \theta, \sigma) \exp(-u_i^2) du_i. \end{aligned}$$

Interpreted as a function of the parameters $(\beta_1, \beta_2, \sigma)$, this is the likelihood function. Therefore, the log-likelihood is, to within an additive constant that does not depend on any unknown parameters, equal to

$$\sum_{i=1}^N \log \left(\int_{-\infty}^{\infty} g(u_i; \theta, \sigma) \exp(-u_i^2) du_i \right).$$

We obtain the maximum likelihood estimate by minimizing

$$\begin{aligned} & - \sum_{i=1}^N \log \left(\int_{-\infty}^{\infty} g(u_i; \theta, \sigma) \exp(-u_i^2) du_i \right) \\ & \approx - \sum_{i=1}^N \log \left(\sum_{j=1}^m g(u_{ij}; \theta, \sigma) w_{ij} \right), \end{aligned} \quad (\text{D.6})$$

where u_{ij} and w_{ij} are Gauss-Hermite nodes and weights, respectively. We use the R function `nlminb` to carry out the minimization of (D.6) to find the maximum likelihood estimates $\hat{\beta}_1$, $\hat{\beta}_2$ and $\hat{\sigma}$ from the whole teratology dataset.

D.6 Proof of Theorem 5.5.1

In this section, we prove Theorem 5.5.1, given in Section 5.5. This proof is due to Paul Kabaila. The proof of this theorem consists of Parts (a) - (d) described as follows.

Part (a):

Observe that

$$\begin{aligned} & c(\theta, \sigma) - c_m(\theta, \sigma) \\ &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} k(z) \exp(-z^2) dz - \frac{1}{\sqrt{\pi}} \sum_{i=1}^m k(z_i) w_i \\ &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} k(z) \exp(-z^2) dz - \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} q_m(z) \exp(-z^2) dz \\ &+ \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} q_m(z) \exp(-z^2) dz - \frac{1}{\sqrt{\pi}} \sum_{i=1}^m q_m(z_i) w_i \\ &+ \frac{1}{\sqrt{\pi}} \sum_{i=1}^m q_m(z_i) w_i - \frac{1}{\sqrt{\pi}} \sum_{i=1}^m k(z_i) w_i. \end{aligned} \quad (\text{D.7})$$

Note that

$$\frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} q_m(z) \exp(-z^2) dz - \frac{1}{\sqrt{\pi}} \sum_{i=1}^m q_m(z_i) w_i = 0.$$

Therefore

$$\begin{aligned}
(\text{D.7}) &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} (k(z) - q_m(z)) \exp(-z^2) dz + \frac{1}{\sqrt{\pi}} \sum_{i=1}^m (q_m(z_i) - k(z_i)) w_i \\
&= \frac{1}{\sqrt{\pi}} \int_{z_u}^{\infty} (k(z) - q_m(z)) \exp(-z^2) dz + \\
&\quad \frac{1}{\sqrt{\pi}} \int_{z_l}^{z_u} (k(z) - q_m(z)) \exp(-z^2) dz + \\
&\quad \frac{1}{\sqrt{\pi}} \int_{-\infty}^{z_l} (k(z) - q_m(z)) \exp(-z^2) dz + \frac{1}{\sqrt{\pi}} \sum_{i=1}^m (q_m(z_i) - k(z_i)) w_i \\
&= a_\ell + a_u + \frac{1}{\sqrt{\pi}} \int_{z_\ell}^{z_u} (k(z) - q_m(z)) \exp(-z^2) dz + \\
&\quad \frac{1}{\sqrt{\pi}} \sum_{i=1}^m (q_m(z_i) - k(z_i)) w_i,
\end{aligned}$$

where

$$a_\ell = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{z_\ell} (k(z) - q_m(z)) \exp(-z^2) dz$$

and

$$a_u = \frac{1}{\sqrt{\pi}} \int_{z_u}^{\infty} (k(z) - q_m(z)) \exp(-z^2) dz.$$

Part (b):

Now

$$\begin{aligned}
&\left| \frac{1}{\sqrt{\pi}} \int_{z_\ell}^{z_u} (k(z) - q_m(z)) \exp(-z^2) dz \right| \\
&\leq \frac{1}{\sqrt{\pi}} \int_{z_\ell}^{z_u} |k(z) - q_m(z)| \exp(-z^2) dz \\
&\leq \frac{1}{\sqrt{\pi}} \int_{z_\ell}^{z_u} \max_{z \in [z_\ell, z_u]} |k(z) - q_m(z)| \exp(-z^2) dz \\
&= e_m \frac{1}{\sqrt{\pi}} \int_{z_\ell}^{z_u} \exp(-z^2) dz \\
&\leq e_m \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \exp(-z^2) dz = e_m.
\end{aligned}$$

We evaluate

$$\frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \exp(-z^2) dz \tag{D.8}$$

by changing the variable of integration to $z = l/\sqrt{2}$, so that $l = \sqrt{2}z$ and

$$(D.8) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \exp\left(\frac{-l^2}{2}\right) \frac{dl}{\sqrt{2}} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left(\frac{-l^2}{2}\right) dl = 1.$$

Note that

$$\begin{aligned} & \left| \frac{1}{\sqrt{\pi}} \sum_{i=1}^m (q_m(z_i) - k(z_i)) w_i \right| \\ & \leq \frac{1}{\sqrt{\pi}} \sum_{i=1}^m |q_m(z_i) - k(z_i)| w_i \\ & \leq \frac{1}{\sqrt{\pi}} \sum_{i=1}^m \max_{z \in [z_\ell, z_u]} |k(z) - q_m(z)| w_i \\ & \leq e_m \frac{1}{\sqrt{\pi}} \sum_{i=1}^m w_i = e_m. \end{aligned}$$

Now

$$\int_{-\infty}^{\infty} 1 \exp(-z^2) dz = \sum_{i=1}^m w_i.$$

Hence

$$\frac{1}{\sqrt{\pi}} \sum_{i=1}^m w_i = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \exp(-z^2) dz = 1.$$

Thus

$$|c(\theta, \sigma) - c_m(\theta, \sigma)| \leq 2e_m + |a_\ell| + |a_u|.$$

Part (c):

If $k(z) \geq q_m(z) \geq 0$ for all $z \leq z_\ell$, then

$$\begin{aligned} 0 \leq a_\ell &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{z_\ell} (k(z) - q_m(z)) \exp(-z^2) dz \\ &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{z_\ell} k(z) \exp(-z^2) dz - \frac{1}{\sqrt{\pi}} \int_{-\infty}^{z_\ell} q_m(z) \exp(-z^2) dz \\ &\leq \frac{1}{\sqrt{\pi}} \int_{-\infty}^{z_\ell} k(z) \exp(-z^2) dz. \end{aligned} \tag{D.9}$$

Note that

$$1/\sqrt{\pi} \int_{-\infty}^{z_l} q_m(z) \exp(-z^2) dz \geq 0.$$

Now change the variable of integration to

$$z = \frac{t - \mu}{\sqrt{2\tau}}$$

so that $t = \mu + \sqrt{2\tau}z$ and

$$\begin{aligned} \text{(D.9)} &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\mu + \sqrt{2\tau}z_l} k\left(\frac{t - \mu}{\sqrt{2\tau}}\right) \exp\left(-\frac{(t - \mu)^2}{2\tau^2}\right) \frac{dt}{\sqrt{2\tau}} \\ &= \int_{-\infty}^{\mu + \sqrt{2\tau}z_l} k\left(\frac{t - \mu}{\sqrt{2\tau}}\right) \frac{1}{\sqrt{2\pi\tau}} \exp\left(-\frac{(t - \mu)^2}{2\tau^2}\right) dt \\ &= \int_{-\infty}^{\mu + \sqrt{2\tau}z_l} k\left(\frac{t - \mu}{\sqrt{2\tau}}\right) \phi(t; \mu, \sigma^2) dt. \end{aligned} \quad \text{(D.10)}$$

Now

$$k\left(\frac{t - \mu}{\sqrt{2\tau}}\right) = h\left(\mu + \sqrt{2\tau} \frac{t - \mu}{\sqrt{2\tau}}; \theta, \sigma\right) = h(t; \theta, \sigma).$$

Thus

$$\begin{aligned} \text{(D.10)} &= \int_{-\infty}^{\mu + \sqrt{2\tau}z_l} h(t; \theta, \sigma) \phi(t; \mu, \sigma^2) dt \\ &= \int_{-\infty}^{\mu + \sqrt{2\tau}z_l} g(t; \theta, \sigma) \phi(t; 0, \sigma^2) dt. \end{aligned}$$

Part (d):

If $k(z) \geq q_m(z) \geq 0$ for all $z \geq z_u$, then

$$\begin{aligned} 0 \leq a_u &= \frac{1}{\sqrt{\pi}} \int_{z_u}^{\infty} (k(z) - q_m(z)) \exp(-z^2) dz \\ &= \frac{1}{\sqrt{\pi}} \int_{z_u}^{\infty} k(z) \exp(-z^2) dz - \frac{1}{\sqrt{\pi}} \int_{z_u}^{\infty} q_m(z) \exp(-z^2) dz \\ &\leq \frac{1}{\sqrt{\pi}} \int_{z_u}^{\infty} k(z) \exp(-z^2) dz. \end{aligned} \quad \text{(D.11)}$$

Note that

$$1/\sqrt{\pi} \int_{z_u}^{\infty} q_m(z) \exp(-z^2) dz \geq 0.$$

Change the variable of integration to

$$z = \frac{t - \mu}{\sqrt{2}\tau}$$

so that $t = \mu + \sqrt{2}\tau z$ and

$$(D.11) = \int_{\mu + \sqrt{2}\tau z_u}^{\infty} h(t; \theta, \sigma) \phi(t; \mu, \sigma^2) dt = \int_{\mu + \sqrt{2}\tau z_u}^{\infty} g(t; \theta, \sigma) \phi(t; 0, \sigma^2) dt.$$

D.7 Evaluation of $k(z)$ for $c_3 > 0$

The function $k(z)$ is defined in Section 5.5. Note that

$$\begin{aligned} r(t; \theta, \sigma) &= h(t; \theta, \sigma) (\phi(t; \mu, \tau^2))^{1/2} \\ &= h(t; \theta, \sigma) \left(\frac{1}{\sqrt{2\pi}\tau} \exp\left(-\frac{1}{2}\left(\frac{t - \mu}{\tau}\right)^2\right) \right)^{1/2} \\ &= h(t; \theta, \sigma) \left(\frac{1}{\sqrt{2\pi}\tau} \right)^{1/2} \exp\left(-\frac{1}{4}\left(\frac{t - \mu}{\tau}\right)^2\right). \end{aligned}$$

Hence

$$h(t; \theta, \sigma) = (\sqrt{2\pi}\tau)^{1/2} r(t; \theta, \sigma) \exp\left(\frac{1}{4}\left(\frac{t - \mu}{\tau}\right)^2\right).$$

As noted in subsection 5.4.2, if $c_3 > 0$ then $r(t; \theta, \sigma) \rightarrow \infty$ as $t \rightarrow \infty$ and as $t \rightarrow -\infty$. We know that

$$\exp\left(\frac{1}{4}\left(\frac{t - \mu}{\tau}\right)^2\right)$$

“grows faster than any polynomial” (using the terminology on p.292 of [Spivak \(1967\)](#)). Therefore, if $c_3 > 0$ then $h(t; \theta, \sigma) \rightarrow \infty$ as $t \rightarrow \infty$ and as $t \rightarrow -\infty$, faster than any polynomial. Thus, if $c_3 > 0$ then $k(z) = h(\mu + \sqrt{2}\tau z; \theta, \sigma) \rightarrow \infty$, as $z \rightarrow \infty$ and as $z \rightarrow -\infty$, faster than any polynomial.

D.8 R programs for the evaluation of the performance of importance sampling

In subsection [D.8.1](#), we list the R programs for the computation of the maximum likelihood estimates of the parameters of the logistic regression model with random intercept. In subsection [D.8.2](#), we list the R programs for the computation of the function $\log r(t; \theta, \sigma)$.

D.8.1 R programs for the computation of the maximum likelihood estimates $\hat{\beta}_1$, $\hat{\beta}_2$ and $\hat{\sigma}$

```
CompMLE <- function(m, N, x.vec, J.vec, y.vec){  
  # This function calculates the maximum  
  # likelihood estimates of betal, beta2 and sigma  
  # for the given dataset.  
  #  
  # Input:  
  # m: number of Gauss Hermite quadrature nodes  
  # N: number of clusters  
  # x.vec: vector of covariates  
  # J.vec: vector of cluster sizes  
  # y.vec: response vector  
  #  
  # Output  
  # A vector of three values.  
  #  
  # Written by N. Ranathunga in August 2018  
  
  quad.rule <- gauss.quad(m, kind = "hermite")  
  gh.nodes <- quad.rule$nodes  
  gh.weights <- quad.rule$weights  
  
  #Set a vector to store the results  
  out.int <- seq(0, 0, length.out = N)  
  
  mle.out <- nlminb(start = c(0, 0, 0.2),  
                    objective = MLEGH, hessian = TRUE)$par  
}
```

```

MLEGH <- function(para, m, x.vec, J.vec, y.vec,
                  gh.nodes, gh.weights, N){
  # This function calculates the value of
  # - sum_{i=1}^N log(int_{-infinity}^{infinity}
  #   g(t;theta, sd)*exp(-t^2) dt)
  # by applying Gaussian Hermite quadrature.
  #
  # Input:
  # para: parameters vector
  # m: no.of nodes
  # x.vec: vector of covariates
  # J.vec: vector of cluster sizes
  # y.vec: response vector
  # gh.nodes.vec: GH nodes
  # gh.weights: GH weights
  # N: number of clusters
  #
  # Output
  # A vector of three values.
  #
  # Written by N. Ranathunga in August 2018

  for (i in c(1:N)) {

    J <- J.vec[i]
    v <- y.vec[i]
    x <- x.vec[i]
    out.int[i] <- IntClustGH(m=m, beta1=para[1],
                           beta2=para[2], sd=para[3], x=x, y=y, J=J,
                           nodes.vec=gh.nodes, w.vec=gh.weights)

  }

  out.mle <- -sum(log(out.int))
}

```



```

IntClustGH <- function(m, beta1, beta2, x, sd, y, J, nodes.
  vec, w.vec){
  # This function calculates the integral
  # from -infinity to infinity g(t;theta, sd)*exp(-t^2)
  # by applying Gaussian Hermite quadrature for a single
  # cluster.
  #
  # Inputs
  # m: no.of nodes
  # beta1: intercept
  # beta2: parameter of the covariate x
  # sd: standard deviation
  # x: value the covariate
  # J: total no.of units within the cluster
  # y: no.of survivals within the cluster
  # nodes.vec: GH nodes
  # w.vec: GH weights
  #
  # Written by N. Ranathunga in July 2018

  expresssion0 <- seq(0, 0, length.out = m)
  expresssion1 <- seq(0, 0, length.out = m)
  expresssion2 <- seq(0, 0, length.out = m)

  for (i in c(1:m)) {
    expresssion0[i] <- beta1 + beta2*x + (nodes.vec[i]*sqrt
      (2)*sd)
    expresssion1[i] <- exp(expresssion0[i]*y)/(1 + exp(
      expresssion0[i]))^J
    expresssion2[i] <- expresssion1[i] * w.vec[i]
  }

  out <- sum(expresssion2)
}

```

D.8.2 R programs for the computation of $\log r(t; \theta, \sigma)$

```
Logr <- function(theta, sd, J, y, t, post.mu, post.var){
  # This function calculates the value of
  # the function log r(t; theta, sd) where
  # r(t; theta, sd) = c0 * exp(c1+c2t+c3t^2)/[1+exp(theta+t)
  # ]^J
  #
  # Input:
  # theta: beta1 + beta2*x
  # sd: standard deviation
  # J: total no.of units within the cluster
  # y: no.of survivals within the cluster
  # t: a random value equals to mu + sqrt{2}*tau*z
  # post.mu: posterior mode
  # post.var: posterior variance obtained at posterior mode
  #
  # Output:
  # Value of log r(t; sd, theta)
  #
  # Written by N. Ranathunga in August 2018

  c0 <- sqrt( sqrt(post.var) / (sqrt(2*pi)*sd^2) )
  c1 <- (theta*y) + (post.mu^2)/(4*post.var)
  c2 <- y - post.mu/(2*post.var)
  c3 <- 1/(4*post.var) - 1/(2*sd^2)

  out <- log(c0) + (c1 + c2*t + c3*t^2) -
    J*log(1 + exp(theta + t))
}
```

```

post_mu_var <- function(theta, sd, y.vec, J.vec){
# This function computes the vector of posterior
# means and the vector of posterior variances
# of all the clusters.
#
# Inputs
# theta: beta1 + beta2*x
# sd: standard deviation
# J.vec: vector of cluster sizes
# y.vec: response vector
#
# Output
# A list containg two vectors.
#
# Written by N. Ranathunga in August 2018

post.mean <- seq(0, 0, length.out=length(J.vec))
post.var <- seq(0, 0, length.out=length(J.vec))

for (i in c(1:length(J.vec))) {
  J.cluster <- J.vec[i]
  y.cluster <- y.vec[i]
  theta <- theta.vec[i]
  post.mean[i] <- optimize(LogOfgPhi, c(-10, 10),
                           tol = 0.0001, maximum = TRUE,
                           theta=theta, sd=sd, y=y.cluster,
                           J=J.cluster)$maximum
  post.var[i] <- PostVar(theta=theta, sd=sd,
                        J=J.cluster, t=post.mean[i])
}
out <- list(postmu= post.mean, postvar=post.var)
}

```

```

LogOfgPhi <- function(theta, sd, J, y, t){
# This function calculates the value of
# log( g(t;theta, sd) * phi(t;0, sd^2)) where
# g(t;theta, sd) = exp[(theta+t)y]/[1+exp(theta+t)]^J
# and phi(t;0, sd^2)) denotes the N(0, sd^2) pdf
# evaluated at t.
#
# Input:
# theta: beta1 + beta2*x
# sd: standard deviation
# J: total no.of units within the cluster
# y: no.of survivals within the cluster
# t: a value equals to mu + sqrt{2}*tau*z
#
# Output
# Value of log( g(t;y,sd,theta) * phi(t;0, sd^2))
#
# Written by N. Ranathunga in August 2018

expresssion0 <- theta + t
expresssion1 <- (expresssion0 * y) - ( (t^2) / (2*sd^2) ) -
  (J * log(1 + exp(expresssion0))) -
  log(sqrt(2*pi)*sd)
}

```

```

PostVar <- function(theta, sd, J, t){
  # This function computes the posterior variance,
  # (tau^2), evaluated at posterior mode (mu), where
  # phi(t; mu, tau^2) is the approximation to the
  # normalized/exact posterior density, with
  # mu = posterior mode and tau^2 = posterior variance
  # which is equal to the inverse of the information
  # matrix evaluated at mu.
  #
  # Inputs
  # theta: beta1 + beta2*x
  # sd: standard deviation
  # J: total no.of units within the cluster
  # t: a value equals to mu + sqrt{2}*tau*z
  #
  # Output
  # A value for the posterior variance (tau^2).
  #
  # Written by N. Ranathunga in August 2018

  expresssion0 <- theta + t
  expresssion1 <- (-J) * (exp(expresssion0) /
    (1 + exp(expresssion0))^2) - (1/(sd^2))
  out <- 1 / (-expresssion1)
}

```

Bibliography

- Abeysekera, W. 2014. *New Recentered Confidence Spheres for the Multivariate Normal Mean*. Ph.D. thesis, Department of Mathematics and Statistics, La Trobe University.
- Abeysekera, W., & Kabaila, P. 2017. Optimized recentered confidence spheres for the multivariate normal mean. *Electronic Journal of Statistics*, **11**, 1935–7524.
- Atkinson, K.E. 1989. *An Introduction to Numerical Analysis*. 2nd edn. New York: Wiley.
- Avery, C.W., & Soler, F.P. 1988. Applications of transformations to numerical integration. *The College Mathematics Journal*, **19**, 166–168.
- Bartholomew, D.J. 1971. A comparison of frequentist and Bayesian approaches to inference with prior knowledge (with discussion). *Pages 417–434 of*: Godambe, V.P., & Sprott, D.A. (eds), *Foundations of Statistical Inference*. Toronto: Holt, Rinehart and Winston.
- Box, G.E.P., & Tiao, G.C. 1984. *Bayesian Inference in Statistical Analysis*. New York: Springer.
- Buckland, S.T., Burnham, K.P., & Augustin, N.H. 1997. Model selection: an integral part of inference. *Biometrics*, **53**, 603–618.

- Burnham, K.P., & Anderson, D.R. 2002. *Model Selection and Multimodel Inference*. Second edn. New York: Springer.
- Carlin, B.P., & Louis, T.A. 1996. *Bayes and Empirical Bayes Methods for Data Analysis*. London: Chapman & Hall.
- Casella, G., & Hwang, J.T. 1987. Employing vague prior information in the construction of confidence sets. *Journal of Multivariate Analysis*, **21**, 79–104.
- Casella, G., & Hwang, J.T. 2012. Shrinkage confidence procedures. *Statistical Science*, **27**, 51–60.
- Chandrasekhar, S. 1960. *Radiative Transfer*. New York: Dover.
- Chang, B., & Hoaglin, D.C. 2017. Meta-analysis of odds ratios: current good practices. *Medical Care*, **55**, 328–335.
- Claeskens, G., & Hjort, N.L. 2008. *Model Selection and Model Averaging*. Cambridge University Press, Cambridge, UK.
- Cohen, A. 1974. To pool or not to pool in hypothesis testing. *Journal of the American Statistical Association*, **69**, 721–725.
- Cox, D.R. 2006. *Principles of Statistical Inference*. Cambridge: Cambridge University Press.
- Davis, P.J., & Rabinowitz, P. 1984. *Methods of Numerical Integration*. 2nd edn. San Diego, CA: Academic Press.
- Demidenko, E.Z. 2004. *Mixed Models: Theory and Applications*. Hoboken, NJ: Wiley.
- Dunnett, C.W. 1989. Algorithm AS 251: Multivariate normal probability integrals with product correlation structure. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, **38**, 564–579.

- Dunnett, C.W., & Sobel, M. 1955. Approximations to the probability integral and certain percentage points of a multivariate analogue of Student's t-distribution. *Biometrika*, **42**, 258–260.
- Efron, B. 2014. Estimation and accuracy after model selection. *Journal of the American Statistical Association*, **109**, 991–1007.
- Farchione, D., & Kabaila, P. 2008. Confidence intervals for the normal mean utilizing prior information. *Statistics and Probability Letters*, **78**, 1094–1100.
- Finney, D.J. 1955. *Experimental Design and Its Statistical Basis*. Chicago: University of Chicago Press.
- Finney, D.J. 1964. *Statistical Method in Biological Assay*. 2nd edn. New York: Hafner.
- Fletcher, D. 2018. *Model Averaging*. Berlin: Springer.
- Fletcher, D., & Turek, D. 2011. Model-averaged profile likelihood intervals. *Journal of Agricultural, Biological, and Environmental Statistics*, **17**, 38–51.
- Genz, A., & Bretz, F. 2009. *Computation of Multivariate Normal and t Probabilities*. London: Springer.
- Giri, K. 2008. *Confidence intervals in regression utilizing prior information*. PhD thesis, Department of Mathematics and Statistics, La Trobe University.
- Grewal, R.S. 1952. A method for testing analgesics in mice. *British Journal of Pharmacology and Chemotherapy*, **7**, 433.

- Gupta, S.S., & Panchapakesan, S. 2002. *Multiple Decision Procedures: Theory and Methodology of Selecting and Ranking Populations*. Philadelphia: SIAM.
- Hedeker, D., & Gibbons, R.D. 2006. *Longitudinal Data Analysis*. Hoboken, NJ: Wiley.
- Hjort, N.L., & Claeskens, G. 2003. Frequentist model average estimators. *Journal of the American Statistical Association*, **98**, 879–899.
- Hochberg, Y., & Tamhane, A.C. 1987. *Multiple Comparison Procedures*. New York: Wiley.
- Hodges, J.L., & Lehmann, E.L. 1952. The use of previous experience in reaching statistical decisions. *Annals of Mathematical Statistics*, **23**, 396–407.
- Hoff, P., & Yu, C. 2019. Exact adaptive confidence intervals for linear regression coefficients. *Electronic Journal of Statistics*, **13**, 94–119.
- Imhof, J.P. 1963. On the method for numerical integration of Clenshaw and Curtis. *Numerische Mathematik*, **5**, 138–141.
- Johnson, S.G. 2014. *The nlopt nonlinear-optimization package*. URL <http://ab-initio.mit.edu/nlopt>.
- Kabaila, P. 2009. The coverage properties of confidence regions after model selection. *International Statistical Review*, **77**, 405–414.
- Kabaila, P. 2018. On the minimum coverage probability of model averaged tail area confidence intervals. *Canadian Journal of Statistics*, **46**, 279–297.
- Kabaila, P., & Farchione, D. 2012. The minimum coverage of confidence intervals in regression after a preliminary F test. *Journal of Statistical Planning and Inference*, **142**, 956–964.

- Kabaila, P., & Giri, K. 2009a. Confidence intervals in regression utilizing prior information. *Journal of Statistical Planning and Inference*, **139**, 3419–3429.
- Kabaila, P., & Giri, K. 2009b. Upper bounds on the minimum coverage probability of confidence intervals in regression after model selection. *Australian & New Zealand Journal of Statistics*, **51**, 271–287.
- Kabaila, P., & Giri, K. 2013. Further properties of frequentist confidence intervals in regression that utilize uncertain prior information. *Australian & New Zealand Journal of Statistics*, **55**, 259–270.
- Kabaila, P., & Giri, K. 2014. Simultaneous confidence intervals for the population cell means, for two-by-two factorial data, that utilize uncertain prior information. *Communications in Statistics – Theory and Methods*, **43**, 4074–4087.
- Kabaila, P., & Kuveke, R.E.H. 2019. The large sample coverage probability of confidence intervals in general regression models after a preliminary hypothesis test. *Scandinavian Journal of Statistics*, **46**, 432–445.
- Kabaila, P., & Leeb, H. 2006. On the large-sample minimal coverage probability of confidence intervals after model selection. *Journal of the American Statistical Association*, **101**, 619–629.
- Kabaila, P., & Ranathunga, N. 2019. On Adaptive Gauss-Hermite Quadrature for Estimation in GLMM's. *Pages 130–139 of: Nguyen, H. (ed), Statistics and Data Science. RSSDS 2019. Communications in Computer and Information Science*, vol. 1150. Singapore: Springer.
- Kabaila, P., & Ranathunga, N. 2020. Confidence intervals in general regression models that utilize uncertain prior information. arXiv:2009.07464.

- Kabaila, P., & Ranathunga, N. 2021. Computation of the expected value of a chi-distributed random variable. *Computational Statistics*, **36**, 313–332.
- Kabaila, P., & Tissera, D. 2014. Confidence intervals in regression that utilize uncertain prior information about a vector parameter. *Australian & New Zealand Journal of Statistics*, **56**, 371–383.
- Kabaila, P., & Wijethunga, C. 2019a. Confidence intervals centred on bootstrap smoothed estimators. *Australian & New Zealand Journal of Statistics*, **61**, 19–38.
- Kabaila, P., & Wijethunga, C. 2019b. On confidence intervals centred on bootstrap smoothed estimators. *Stat*, **8**, e233.
- Kabaila, P., Welsh, A.H., & Abeysekera, W. 2016. Model-averaged confidence intervals. *Scandinavian Journal of Statistics*, **43**, 35–48.
- Kabaila, P., Welsh, A.H., & Mainzer, R. 2017. The performance of model averaged tail area confidence intervals. *Communications in Statistics - Theory and Methods*, **46**, 10718–10732.
- Kabaila, P., Welsh, A.H., & Wijethunga, C. 2020. Finite sample properties of confidence intervals centered on a model averaged estimator. *Journal of Statistical Planning and Inference*, **207**, 10–26.
- Kempthorne, P.J. 1983. Minimax-Bayes compromise estimators. *Proc. Bus. and Econ. Statist. Sec. Amer. Statist. Assoc.*, 563–573.
- Kempthorne, P.J. 1987. Numerical specification of discrete least favorable prior distributions. *SIAM Journal on Scientific and Statistical Computing*, **8**, 171–184.
- Kempthorne, P.J. 1988. Controlling risks under different loss functions: the compromise decision problem. *Annals of Statistics*, **16**, 1594–1608.

- Khan, S. 2003. Estimation of the parameters of two parallel regression lines under uncertain prior information. *Biometrical Journal*, **45**, 73–90.
- Khan, S. 2008. Shrinkage estimators of intercept parameters of two simple regression models with suspected equal slopes. *Communications in Statistics - Theory and Methods*, **37**, 247–260.
- Kim, Y., Choi, Y., & Emery, S. 2013. Logistic regression with multiple random effects: a simulation study of estimation methods and statistical packages. *American Statistician*, **67**, 171–182.
- Leamer, E.E. 1978. *Specification Searches: Ad Hoc Inference with Nonexperimental Data*. New York: Wiley.
- Lesaffre, E., & Spiessens, B. 2001. On the effect of the number of quadrature points in a logistic random-effects model: an example. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, **50**, 325–335.
- Liu, Q., & Pierce, D.A. 1994. A note on Gauss—Hermite quadrature. *Biometrika*, **81**, 624–629.
- Maata, J.M., & Casella, G. 1990. Developments in decision-theoretic variance estimation. *Statistical Science*, **28**, 151–156.
- Mainzer, R., & Kabaila, P. 2019. ciuupi: An R package for computing confidence intervals that utilize uncertain prior information. *R Journal*, **11**, 323–336.
- McCullagh, P., & Nelder, J.A. 1989. *Generalized Linear Models*. 2nd edn. London: Chapman & Hall.
- Mead, R. 1988. *The Design of Experiments*. Cambridge: Cambridge University Press.

- Meeker, W.Q., & Escobar, L.A. 1995. Teaching about approximate confidence regions based on maximum likelihood estimation. *American Statistician*, **49**, 48–53.
- Miller, R.G. 1981. *Simultaneous Statistical Inference*. 2nd edn. New York: Springer.
- Morgan, B.J.T. 1992. *Analysis of Quantal Response Data*. Boston, MA: Springer.
- Mori, M. 1985. Quadrature formulas obtained by variable transformation and the DE-rule. *Journal of Computational and Applied Mathematics*, **12** & **13**, 119–130.
- Mori, M. 1988. The double exponential formula for numerical integration over the half infinite interval. *Pages 367–379 of: Agarwal, R.P., Chow, Y.M., & Wilson, S.J. (eds), Numerical Mathematics (Singapore 1988)*. Basel: Birkhauser.
- Mori, M., & Sugihara, M. 2001. The double-exponential transformation in numerical analysis. *Journal of Computational and Applied Mathematics*, **127**, 287–296.
- Owen, A.B. 2013. *Monte Carlo Theory, Methods and Examples*. URL <http://statweb.stanford.edu/~owen/mc/>.
- Papoulis, A. 1962. *The Fourier Integral and Its Applications*. New York: McGraw-Hill.
- Pawitan, Y. 2000. A reminder of the fallibility of the Wald statistic: likelihood explanation. *American Statistician*, **54**, 54–56.
- Pawitan, Y. 2013. *In All Likelihood : Statistical Modelling and Inference Using Likelihood*. Oxford: Clarendon Press.

- Pinheiro, J.C., & Bates, D.M. 1995. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of Computational and Graphical Statistics*, **4**, 12–35.
- Pratt, J.W. 1961. Length of confidence intervals. *Journal of the American Statistical Association*, **56**, 549–567.
- Pratt, J.W. 1963. Shorter confidence intervals for the mean of a normal distribution with known variance. *Annals of Mathematical Statistics*, **34**, 574–586.
- Pratt, J.W. 1965. Bayesian interpretation of standard inference statements. *Journal of the Royal Statistical Society, Series B*, **27**, 169–203.
- Puza, B., & O'Neill, T. 2006a. Generalised Clopper-Pearson confidence intervals for the binomial proportion. *Journal of Statistical Computation and Simulation*, **76**, 498–508.
- Puza, B., & O'Neill, T. 2006b. Interval estimation via tail functions. *The Canadian Journal of Statistics*, **34**, 299–310.
- Puza, B., & Yang, M. 2016. Improved confidence intervals for the exponential mean via tail functions. *Communications in Statistics - Theory and Methods*, **45**, 529–539.
- Rabe-Hesketh, S., & Skrondal, A. 2008. *Multilevel and Longitudinal Modeling Using Stata*. 2nd edn. College Station, Texas: Stata Press Publication.
- Robert, C.P., & Casella, G. 2004. *Monte Carlo Statistical Methods*. 2nd edn. New Jersey: Springer.
- Robertson, J.L., Jones, M.M., Olguin, E., & Alberts, B. 2017. *Bioassays with Arthropods*. 3rd edn. Boca Raton, FL: CRC Press.

- Sag, T.W., & Szekeres, G. 1964. Numerical evaluation of high-dimensional integrals. *Mathematics of Computation*, **18**, 245–253.
- Schwartz, C. 1969. Numerical integration of analytic functions. *Journal of Computational Physics*, **4**, 19–29.
- Shampine, L.F. 2008. MATLAB program for quadrature in 2D. *Applied Mathematics and Computation*, **202**, 266–274.
- Shea, B.L. 1988. Algorithm AS 239: Chi-squared and incomplete gamma integral. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, **37**, 466–473.
- Spivak, M. 1967. *Calculus*. New York: Benjamin.
- Stein, C. 1962. Confidence sets for the mean of a multivariate normal distribution. *Journal of the Royal Statistical Society, Series B*, **24**, 265–296.
- Szegő, G. 1967. *Orthogonal Polynomials*. 3rd edn. Providence, Rhode Island: American Mathematical Society.
- Takahasi, H., & Mori, M. 1973. Quadrature formulas obtained by variable transformation. *Numerische Mathematik*, **21**, 206–219.
- Trefethen, L.N., & Weideman, J.A.C. 2014. The exponentially convergent trapezoidal rule. *SIAM Review*, **56**, 385–458.
- Tuerlinckx, F., Rijmen, F., Verbeke, G., & Boeck, P. 2006. Statistical inference in generalized linear mixed models: a review. *British Journal of Mathematical and Statistical Psychology*, **59**, 225–255.
- Turek, D., & Fletcher, D. 2012. Model-averaged Wald confidence intervals. *Computational Statistics & Data Analysis*, **56**, 2809–2815.

- Weil, C.S. 1970. Selection of the valid number of sampling units and a consideration of their combination in toxicological studies involving reproduction, teratogenesis or carcinogenesis. *Food and Cosmetics Toxicology*, **8**, 177 – 182.
- Wu, C.F.J., & Hamada, M. 2000. *Experiments: Planning, Analysis, and Parameter Optimization*. New York: Wiley.
- Yang, M., & Puza, B. 2020. Optimal confidence intervals for the geometric parameter. *Communications in Statistics - Theory and Methods*, **49**, 590–606.
- Young, G.A., & Smith, R.L. 2005. *Essentials of Statistical Inference*. Cambridge: Cambridge University Press.
- Yu, C., & Hoff, P. 2018. Adaptive multigroup confidence intervals with constant coverage. *Biometrika*, **105**, 319–335.