Journal of Statistical Models and Methods (2025)
1(1), 45–73, https://doi.org/10.65905/hbw5d814

Article Type: Research Article

Assessing Method-Strategy Pairs in the R nleqslv Package: A Comparative Study

Farouq Mohammad A. Alam[1],

[1]Department of Statistics, Faculty of Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia

Farouq Mohammad A. Alam
fmalam@kau.edu.sa

Article History

Received: 10 November 2025
Revised: 09 December 2025
Accepted: 10 December 2025

Keywords
Point Estimation;
Data Analysis;
Computational Statistics;
Reliability;
Weibull Distribution

Abstract

This study presents a comparative analysis of 14 method-strategy pairs in the **R** package **nleqslv**, focusing on solving complex nonlinear systems. The primary objective is to evaluate the performance of these pairs across two models using maximum likelihood estimation. Both models introduce challenges in convergence and stability, providing a rigorous test for the optimization methods. The study assesses the methods based on key metrics such as convergence speed, estimation efficiency, and robustness (i.e., sensitivity analysis) across varying simulation setups. Findings offer insights into the strengths and limitations of different optimization strategies, highlighting optimal combinations for specific scenarios involving censored data and nonlinear estimation. The study aims to guide practitioners and researchers in selecting optimal method-strategy pairs for similar statistical problems.

1 Introduction

Reliability analysis has a pivotal role in engineering and system design, aiming to assess and ensure the operational effectiveness of complex systems under uncertainty. A key challenge lies in modeling and estimating the reliability of systems composed of multiple interconnected components, mainly when these systems exhibit serial or parallel configurations. In a serial system, the failure of any of its components leads to system failure, making early failures a critical concern; an example is the conventional remote control of a device, where the failure of a single battery renders it inoperative. Conversely, parallel systems provide redundancy, allowing the system to function as long as at least one component remains operational. For instance, a dual-battery backup system, such as those found in uninterruptible power supplies, continues functioning as long as at least one battery remains operational. Accurately estimating system reliability under diverse operational conditions, while addressing the computational challenges posed by solving nonlinear systems, necessitates efficient optimization approaches.

Nonlinear systems of equations arise naturally in numerous fields, including physics, engineering, economics, and life sciences, where they model complex phenomena such as statistical modeling, fluid dynamics, structural mechanics, financial systems, and population dynamics. In modern computational sciences, solving nonlinear systems is fundamental to model fitting, optimization, and simulation modeling, where accuracy and efficiency are critical. In statistics, nonlinear systems are often encountered in parameter estimation problems, such as fitting nonlinear regression models, optimizing likelihood functions, and solving moment equations in generalized estimating equations. For reliability analysis, solving nonlinear systems provides insights into the behavior and performance of engineered systems, enabling predictions of failure rates and improving risk assessment methodologies. Despite their ubiquity,

Copyright © 2025 by the author. Published under Creative Commons Attribution (CC BY) license.

solving nonlinear systems is inherently challenging due to their sensitivity to initial conditions, potential non-uniqueness of solutions, and computational complexity in high-dimensional settings. Developing and evaluating robust optimization methods for solving nonlinear systems is crucial, as these underpin applications in both theoretical and applied domains. A well-known optimization method is Newton's (Newton-Raphson) method, introduced by Isaac Newton in the late 17th century to find the roots of real-valued functions numerically under some regular conditions, Deuflhard [1]. Despite its original use in polynomial equations, the method evolved into a core tool in numerical analysis for solving nonlinear equations. Over time, it has become known for its rapid quadratic convergence under favorable conditions but is sensitive to the choice of initial values and flat derivative regions. Suppose that $\mathbf{x}$ is a vector of observed data, while $\boldsymbol{\theta}$ is a vector of model parameters, and the aim is to solve $g(\boldsymbol{\theta}|\mathbf{x}) = 0$. Newton's method finds a solution by iteratively updating the parameter vector $\boldsymbol{\theta}$ as:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \left[\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}^{(k)} \mid \mathbf{x})\right]^{-1} g(\boldsymbol{\theta}^{(k)} \mid \mathbf{x}),$$

where $\boldsymbol{\theta}^{(k)}$ is the parameter estimate at iteration k, and $\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}^{(k)} \mid \boldsymbol{x})$ is the Jacobian matrix of partial derivatives evaluated at $\boldsymbol{\theta}^{(k)}$. The method continues until a convergence criterion is achieved. For example, when $|g(\boldsymbol{\theta}^{(k+1)} \mid \mathbf{x})| < \epsilon$, where ϵ is a small positive tolerance. It is important to mention that no general rule exists regarding when optimization algorithms should be stopped and convergence should be declared. Suggested convergence criteria can be found in Searle et al. [2]. In contrast to Newton's method, Broyden's method builds upon the quasi-Newton approach concept, i.e., approximating the Jacobian matrix, Broyden [3]. It avoids calculating the full Jacobian matrix, making it more efficient for large-scale problems. This derivative-free method approximates updates iteratively, balancing convergence speed and computational complexity. Broyden's approach is beneficial when the Jacobian is either unavailable or expensive to compute, offering robust solutions in optimization settings. Instead of using the Jacobian matrix $\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}^{(k)} \mid \boldsymbol{x})$, Broyden's method approximate its inverse by using a rank-1 formula $\mathbf{J}^{(k+1)}$, such that:

$$\mathbf{J}^{(k+1)} = \mathbf{J}^{(k)} + \frac{\left(\Delta\boldsymbol{\theta}^{(k)} - \mathbf{J}^{(k)}\Delta\boldsymbol{g}^{(k)}\right)\left(\Delta\boldsymbol{\theta}^{(k)}\right)^{\top}}{\left(\Delta\boldsymbol{\theta}^{(k)}\right)^{\top}\Delta\boldsymbol{g}^{(k)}},$$

with $\Delta\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}$ and $\Delta\boldsymbol{g}^{(k)} = g(\boldsymbol{\theta}^{(k+1)} \mid \boldsymbol{x}) - g(\boldsymbol{\theta}^{(k)} \mid \boldsymbol{x})$. See Nocedal and Wright [4] for additional details about Newton and quasi-Newton methods.

Newton's methods and their variants are increasingly utilized in modern statistics, machine learning, and optimization. Sub-sampled Newton methods are particularly effective in addressing the computational challenges of large datasets by approximating the Hessian matrix with a subset of data points. This approach retains fast convergence properties while reducing computational costs, making it well-suited for logistic regression and empirical risk minimization, see Ye et al. [5], and Erdogdu and Montanari [6]. Projected Newton-type methods, frequently applied in machine learning for constrained optimization, ensure each iterative step improves the objective function. These methods rely on techniques like line search and trust regions to maintain convergence and are effective even when dealing with complex parameter spaces, Nocedal and Wright [4]. Hybrid models, which combine Newton's methods with stochastic optimization techniques, further enhance performance by balancing computational feasibility with precision, proving valuable in high-dimensional parameter estimation problems, Khorasani and Mahoney [7].

Newton's and Broyden's methods are implemented directly in **R**, a powerful and flexible environment for statistical computing, see R Core Team [8], and Nash [9], through the **nleqslv** package, which offers multiple approaches and global strategies for solving dense nonlinear systems of equations, Hasselman [10]. The **nleqslv** package in **R** is a versatile tool designed to solve nonlinear systems of equations. It implements various strategies, including Newton-based and quasi-Newton methods enhanced using trust region and line search algorithms. These features make it a practical choice for solving complex optimization problems in reliability modeling. Despite its utility, the package's performance across different method-strategy combinations has not been comprehensively evaluated, particularly in the context of parameter estimation for reliability models. This package provides complete Newton iterations,

where the Jacobian matrix is recalculated at each step to ensure precision. Broyden's, on the other hand, uses rank-1 updates to approximate the Jacobian, balancing computational efficiency and accuracy. It builds on a simplified and modernized version of the algorithm introduced by Reichel and Gragg [11]. To ensure robust convergence, even with ill-conditioned or singular Jacobians, **nleqslv** employs global search strategies. These include cubic, quadratic, and geometric line search methods alongside three trust-region algorithms: (1) the double dogleg method, Dennis and Schnabel [12], (2) Powell's dogleg method, Powell [13], and (3) a Levenberg-Marquardt-like algorithm, Moré [14], and Nocedal [4]. The double dogleg is a trust region method that combines two-step directions (a steepest descent direction and a quasi-Newton direction) to balance convergence speed and stability. In contrast, Powell's dogleg trust region method approximates the optimal step within a constrained region, aiming for efficient and robust convergence. These features make the package highly effective for optimization problems in diverse fields such as machine learning, econometrics, and nonlinear regression, where balancing computational efficiency and reliable convergence is critical. In their discussion of optimization tools, Varadhan and Gilbert [15] highlights a limitation of the **nleqslv** package, noting it may struggle with large-scale problems. On the other hand, Reynaerts et al. [16] applies **nleqslv** as part of their research, demonstrating its usefulness in practical applications. Table 1 summarizes the global strategies alongside their abbreviations.

Table 1: **nleqslv** package global strategies (N: Newton, B: Broyden).

Abbreviation	Strategy
`N-cline, B-cline`	A cubic line search.
`N-qline, B-qline`	A quadratic line search.
`N-gline, B-gline`	A geometric line search.
`N-dbldog, B-dbldog`	A double dogleg trust region method.
`N-pwldog, B-pwldog`	A Powell dogleg trust region method.
`N-hook, B-hook`	A Levenberg-Marquardt-like trust region method.
`N-none, B-none`	Pure local Newton or Broyden methods.

The motivation for this study stems from the complexity of selecting the appropriate global strategy when solving nonlinear systems using **nleqslv**. As the package's author indicates, choosing between trust-region methods and line search strategies often requires trial and error, as no systematic performance evaluation exists. Each approach has strengths and weaknesses, with the double dogleg method showing favorable results in many cases but not consistently outperforming others. This lack of clarity can leave practitioners without guidelines for selecting the most suitable strategy for specific problems. Nonlinear systems of equations are ubiquitous in real-world applications, from reliability analysis and econometrics to machine learning. Yet, their solutions often require robust optimization methods to ensure accuracy and efficiency. This is particularly critical in reliability modeling, where parameter estimation challenges are compounded by the need to enforce positivity constraints on model parameters. While tools like **nleqslv** offer a range of methods, identifying the most effective combination of approaches for specific problem types remains an open challenge. This research systematically evaluates the performance of **nleqslv** method-strategy pairs by addressing the problem of assessing two bipartite system with independent nonidentical Weibull lifetimes, requiring parameter estimation. By exploring convergence rates, robustness, and estimation efficiency, the study provides practical insights into the strengths and limitations of various strategies. This guidance will assist researchers in selecting optimal methods tailored to their problem's characteristics. The importance of comparative studies in optimization is well-documented; for example, Abdulkadirov et al. [17] analyzed various optimization techniques to guide the selection of algorithms for neural network training. Similarly, this work effectively bridges the gap between theory and practice, offering actionable recommendations for solving nonlinear systems.

Although prior studies have discussed Newton and quasi-Newton algorithms and analyzed reliability models such as the additive Weibull distribution, no work has systematically evaluated the full set of method–strategy pairs implemented in the **nleqslv** package. In particular, the literature lacks a

comparative assessment of how these strategies perform when solving the nonlinear likelihood equations arising from serial and parallel system models. This study addresses this gap by providing the first comprehensive, simulation-based comparison of 14 method–strategy combinations, thereby offering practical guidance for selecting effective optimization strategies in reliability estimation.

Clearly, the primary objective of this study is to conduct a systematic and comprehensive evaluation of 14 method–strategy pairs implemented in the R package nleqslv for solving nonlinear systems arising in maximum likelihood estimation under serial and parallel reliability models. By examining convergence behavior, robustness to initial values, computational accuracy, and estimation efficiency through extensive Monte Carlo simulations. Hence, this work aims to provide practitioners with general practical guidance on selecting the most effective optimization strategy for complex nonlinear estimation problems. The remaining parts of this article are organized as follows. Section 2 reviews the study's theoretical framework, Section 3 details the estimation methodology, Section 4 presents the results of numerical experiments, and Section 5 concludes with a discussion of the findings and potential future research directions.

2 The Underlying Problem

This section describes two models used to evaluate the performance of **nleqslv**. These models represent key reliability analysis configurations: serial and parallel systems. Serial systems fail due to early failures, while parallel systems ensure higher reliability by allowing individual component failures. In reliability analysis, computer scientists and systems engineers aim to evaluate the likelihood that a system will function correctly beyond a specified time, denoted by τ. Statistically, this reliability is expressed as $R(\tau) = \Pr(T > \tau)$, where T represents the time-to-failure or lifetime of the system. Here, the underlying problem is estimating the reliability of a specific type of system, which depends on estimating the underlying model parameters. In practical applications, a system often comprises multiple independent components arranged in series, parallel, or hybrid configurations. To model the lifetime of each component, researchers commonly utilize lifetime distributions—probability distributions with non-negative support. Among these, the Weibull distribution is widely applied due to its flexibility, as it generalizes both the exponential and Rayleigh distributions. Under the assumption of non-identical independent Weibull lifetimes, the additive Weibull distribution captures the behavior of a serial system where the failure time is governed by the minimum lifetime of two components. The second model considers a parallel system where the failure time corresponds to the maximum lifetime of its components. These models are ideal for evaluating the robustness and efficiency of the **nleqslv** package in estimating reliability, as detailed in Section 4.

Without the loss of generality, consider a bipartite system comprising two independent components, each sourced from a different manufacturer. Suppose that the lifetime of component i is X_i which follows a Weibull distribution with scale parameter $\beta_i > 0$ and shape parameter $\alpha_i > 0$ with a cumulative distribution function (CDF) and probability density function (PDF) of the following forms:

$$F_i(x; \alpha_i, \beta_i) = 1 - e^{-\left(\frac{x}{\beta_i}\right)^{\alpha_i}}, \quad x > 0,\ \alpha_i > 0,\ \beta_i > 0,$$

and

$$f_i(x; \alpha_i, \beta_i) = \frac{\alpha_i}{\beta_i}\left(\frac{x}{\beta_i}\right)^{\alpha_i - 1} e^{-\left(\frac{x}{\beta_i}\right)^{\alpha_i}}, \quad x > 0,\ \alpha_i > 0,\ \beta_i > 0,$$

Throughout this study, the following assumptions are imposed:

1. The component lifetimes X_1 and X_2 are statistically independent.
2. Each component lifetime follows a Weibull distribution with strictly positive shape and scale parameters, as specified.
3. The observed system lifetimes used in the simulation experiments are uncensored.
4. Standard regularity conditions for maximum likelihood estimation are assumed to hold, ensuring the existence of well-behaved score functions and hessian matrices, as well as the validity of the optimization procedures used.

2.1 Serial System Model

If the components are connected **serially**, then $T = \min(X_1, X_2)$, and the reliability function (RF) of the system is defined as:

$$\Pr(T > t) = R_s(t;\boldsymbol{\theta}) = e^{-\sum_{j=1}^{2}\left(\frac{t}{\beta_j}\right)^{\alpha_j}}, \quad t > 0. \tag{1}$$

such that $\boldsymbol{\theta} = (\alpha_1, \alpha_2, \beta_1, \beta_2) \in \mathbb{R}_+^4$ is the parameter vector. Accordingly, the corresponding CDF and PDF of the system are given by:

$$\Pr(T \leq t) = F_s(t;\boldsymbol{\theta}) = 1 - e^{-\sum_{j=1}^{2}\left(\frac{t}{\beta_j}\right)^{\alpha_j}}, \tag{2}$$

and

$$f_s(t;\boldsymbol{\theta}) = \left[\sum_{j=1}^{2}\frac{\alpha_j}{\beta_j}\left(\frac{t}{\beta_j}\right)^{\alpha_j-1}\right] e^{-\sum_{j=1}^{2}\left(\frac{t}{\beta_j}\right)^{\alpha_j}}, \tag{3}$$

respectively. This lifetime model, known as the additive Weibull (AW) distribution, was proposed by Xie et al. [18]. The AW distribution belongs to a family of Gurvich's generalized family of Weibull distributions, Gurvich et al. [19] (see also Nadarajah, and Kotz [20], and Lai et al. [21] in this connection). Lemonte et al. [22] derived explicit expressions for the moments and other fundamental distribution properties, discussed model parameters estimation by the maximum likelihood methodology, and obtained the corresponding observed information matrix. Recently, Abd EL-Baset and Ghazal [23] generalized the AW distribution by exponentiating its CDF. They discussed some statistical properties of the new generalization and maximum likelihood estimation for parameters.

2.2 Parallel System Model

On the contrary, if the connected components are **parallel**, then $T = \max(X_1, X_2)$, and the RF of the system becomes:

$$\Pr(T > t) = R_p(t;\boldsymbol{\theta}) = 1 - \prod_{i=1}^{2}\left[1 - e^{-\left(\frac{t}{\beta_i}\right)^{\alpha_i}}\right], \quad t > 0. \tag{4}$$

Accordingly, the corresponding CDF and PDF are given by:

$$\Pr(T \leq t) = F_p(t;\boldsymbol{\theta}) = \prod_{i=1}^{2}\left[1 - e^{-\left(\frac{t}{\beta_i}\right)^{\alpha_i}}\right], \tag{5}$$

and

$$f_p(t;\boldsymbol{\theta}) = \frac{\alpha_1}{\beta_1}\left(\frac{t}{\beta_1}\right)^{\alpha_1-1} e^{-\left(\frac{t}{\beta_1}\right)^{\alpha_1}}\left[1 - e^{-\left(\frac{t}{\beta_2}\right)^{\alpha_2}}\right] + \frac{\alpha_2}{\beta_2}\left(\frac{t}{\beta_2}\right)^{\alpha_2-1} e^{-\left(\frac{t}{\beta_2}\right)^{\alpha_2}}\left[1 - e^{-\left(\frac{t}{\beta_1}\right)^{\alpha_1}}\right], \tag{6}$$

respectively.

3 Maximum Likelihood Estimation

In statistical inference, the maximum likelihood estimation methodology is the state-of-the-art method many researchers utilize due to the attractive properties of the resultant estimators. Let $\mathbf{t} = (t_1, \ldots, t_n)$ be an observed random sample of size n two-component systems. This section outlines the mathematical framework for the maximum likelihood theory applied to the serial and parallel system models. These derivations form the foundation for solving the optimization problems using the **nleqslv** package. In its documentation, the main function `nleqslv()` requires the argument `fn`, which represents $g(\boldsymbol{\theta} \mid \mathbf{t}) = 0$. In the context of maximum likelihood theory, $g(\boldsymbol{\theta} \mid \mathbf{t})$ is the gradient of the log-likelihood function, which must be solved to obtain the parameter estimates assuming their existence and uniqueness. The mathematical formulations presented here as well as those in the appendix are implemented in the **nleqslv** package and serve as the foundation for the numerical experiments described in Section 4.

3.1 Estimation assuming a Serial System

To estimate the model parameters assuming Eq. (3), the corresponding maximum likelihood estimators (MLEs) for the model parameters are obtained by solving the following maximization problem:

$$\begin{aligned} &\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\text{maximize}}\ \ell_s(\boldsymbol{\theta}|\mathbf{t}) = \sum_{i=1}^{n} \log\left\{f_1(t_i;\alpha_1,\beta_1)\left[1-F_2(t_i;\alpha_2,\beta_2)\right] + f_2(t_i;\alpha_2,\beta_2)\left[1-F_1(t_i;\alpha_1,\beta_1)\right]\right\} \\ &\text{s.t. } \boldsymbol{\theta}\in\mathbb{R}_+^4, \end{aligned} \tag{7}$$

such that $\ell_s(\boldsymbol{\theta}|\mathbf{t})$ is the log-likelihood function of the AW distribution. By solving the above problem, the MLEs of $\boldsymbol{\theta}$; say, $\hat{\boldsymbol{\theta}}_s = (\hat{\alpha}_1^s, \hat{\alpha}_2^s, \hat{\beta}_1^s, \hat{\beta}_2^s)$, are defined as:

$$\hat{\boldsymbol{\theta}}_s = \underset{\boldsymbol{\theta}\in\mathbb{R}_+^4}{\arg\max}\, \ell_s(\boldsymbol{\theta}|\mathbf{t}).$$

Appendix A provides the mathematical expressions for the gradient and Hessian matrix of the objective function required by Newton-based methods, ensuring clarity and focus in the main text while maintaining conciseness.

3.2 Estimation assuming a Parallel System

To estimate the model parameters assuming Eq. (6), the corresponding MLEs for the model parameters are acquired by solving the following maximization problem:

$$\begin{aligned} &\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\text{maximize}}\ \ell_p(\boldsymbol{\theta}|\mathbf{t}) = \sum_{i=1}^{n} \log\left\{f_1(t_i;\alpha_1,\beta_1)F_2(t_i;\alpha_2,\beta_2) + f_2(t_i;\alpha_2,\beta_2)F_1(t_i;\alpha_1,\beta_1)\right\} \\ &\text{s.t. } \boldsymbol{\theta}\in\mathbb{R}_+^4, \end{aligned} \tag{8}$$

such that $\ell_p(\boldsymbol{\theta}|\mathbf{t})$ is the log-likelihood function based on Eq. (6). The MLEs of $\boldsymbol{\theta}$; say, $\hat{\boldsymbol{\theta}}_p = (\hat{\alpha}_1^p, \hat{\alpha}_2^p, \hat{\beta}_1^p, \hat{\beta}_2^p)$, are the solution of the above optimization problem, i.e.,

$$\hat{\boldsymbol{\theta}}_p = \underset{\boldsymbol{\theta}\in\mathbb{R}_+^4}{\arg\max}\, \ell_p(\boldsymbol{\theta}|\mathbf{t}).$$

Once again, the mathematical expressions for the gradient and Hessian matrix of the log-likelihood function, essential for the optimization process, are provided in Appendix A for brevity and clarity.

4 Numerical Experimentation and Outcomes

This section reports the results of numerical experimentation and the outcomes of Monte Carlo simulations.

4.1 Computational Considerations

When estimating model parameters that are constrained to be positive, it is common to impose explicit constraints during optimization or use log-transformed parameters. The latter approach, often called a log-scaled transformation and sometimes parameter exponential transformation, has a distinct advantage: it eliminates the need for explicit constraints by transforming the problem into an unconstrained optimization framework. This simplification simplifies the optimization process and improves numerical stability and convergence, see Delignette-Muller and Dutang [24, 25]. For example, applying a log transformation on the parameters in fitting distributions like the Weibull allows the optimization process to focus on accuracy without handling boundary constraints. As demonstrated by the **fitdistrplus package** in **R**, this approach ensures that parameter positivity is maintained naturally while avoiding potential

computational issues. Operating in the log scale simplifies the optimization landscape, making parameter estimation efficient and robust, Delignette-Muller and Dutang [25].

This subsection outlines the computational framework and key considerations in assessing the performance of **nleqslv** method-strategy pairs for solving nonlinear systems in serial and parallel reliability models. The analysis begins by acknowledging that the existence and uniqueness of MLEs for the models remain an open problem in statistical theory, motivating the reliance on numerical methods to approximate solutions. The **nleqslv** package is employed using its default settings, which include various globally convergent strategies tailored to unconstrained optimization problems. To ensure compatibility with the positivity constraints of model parameters, the optimization problems are redefined using a log-scaled transformation, as mentioned in Section 3. This transformation simplifies the computational landscape, enhancing numerical stability and convergence. For instance, the functional invariance property of MLEs ensures that optimizing on the log scale naturally satisfies positivity constraints without additional restrictions. If the model parameters are $(\alpha_1, \alpha_2, \beta_1, \beta_2) \in \mathbb{R}^4_+$, then the log-transformed parameters are $(\alpha_1^*, \alpha_2^*, \beta_1^*, \beta_2^*) \in \mathbb{R}^4$, such that $\alpha_j^* = \log(\alpha_j)$ and $\beta_j^* = \log(\beta_j)$. This one-to-one transformation ensures that the MLEs of $\alpha_j > 0$ and $\beta_j > 0$ are given by $\exp(\hat{\alpha}_j^*)$ and $\exp(\hat{\beta}_j^*)$ for $j = 1, 2$, leveraging the functional invariance property, see Casella and Berger [26].

The following propositions illustrate the effects of the log-transformation on the gradient and hessian matrix, which are critical for numerical optimization in the **nleqslv** framework:

Proposition 4.1. If $\ell(\theta)$ is an objective function, $\theta > 0$ is a model parameter, and $\theta^* = \log(\theta) \in \mathbb{R}$, then:

$$\frac{\partial \ell(\theta)}{\partial \theta^*} = \theta \frac{\partial \ell(\theta)}{\partial \theta}.$$

Proof. Since $\theta^* = \log(\theta)$ and $\partial \ell(\theta)/\partial \theta = g(\theta) \Rightarrow \partial \ell(\theta) = g(\theta)\partial\theta$, then:

$$\frac{\partial \ell(\theta)}{\partial \theta^*} = g(\theta)\frac{\partial \theta}{\partial \theta^*} = e^{\theta^*} g(\theta) = \theta \frac{\partial \ell(\theta)}{\partial \theta}.$$

□

Proposition 4.2. If $\ell(\theta_1, \theta_2)$ is an objective function, $\theta_1, \theta_2 > 0$ are model parameters, and $\theta_j^* = \log(\theta_j) \in \mathbb{R}$, $\forall j = 1, 2$, then:

$$\frac{\partial^2 \ell(\theta_1, \theta_2)}{\partial (\theta_j^*)^2} = \theta_j \frac{\partial \ell(\theta_1, \theta_2)}{\partial \theta_j} + \theta_j^2 \frac{\partial^2 \ell(\theta_1, \theta_2)}{\partial \theta_j^2},$$

and

$$\frac{\partial^2 \ell(\theta_1, \theta_2)}{\partial \theta_1^* \partial \theta_2^*} = \theta_1 \theta_2 \frac{\partial^2 \ell(\theta_1, \theta_2)}{\partial \theta_1 \partial \theta_2}.$$

4.2 Illustrative Examples

The examples presented in this subsection serve multiple purposes. First, they provide reproducible scenarios for researchers intending to replicate or extend the analysis using the **nleqslv** package. By specifying the parameter values and optimization outcomes, these examples ensure transparency and practical accessibility. Second, they demonstrate the correctness of the mathematical expressions derived in Section 3, highlighting their practical implementation in the optimization process. Finally, the examples empirically validate the existence and uniqueness of the MLEs under the assumed models, showcasing the numerical feasibility of solving these nonlinear systems using the proposed framework. The existence and uniqueness of MLEs are validated by profiling the log-likelihood function using 4D data visualization.

Example 4.1. Consider the following practical scenario for a serial system. Suppose a particular remote control requires two AAA alkaline batteries, one manufactured by Company A and the other by an independent Company B. Both companies claim that their batteries have a lifespan of up to one year under typical usage conditions. The lifetime of the battery from Company A is assumed to follow a $Weibull(\alpha_1 = 3, \beta_1 = 2)$ distribution, while the lifetime of the battery from Company B follows a

$Weibull(\alpha_2 = 0.75, \beta_2 = 2)$ distribution. In this setup, the remote control can be viewed as a two-component serial system that fails when the first of its components (i.e., batteries) is depleted. Under these assumptions, the actual reliability of the remote control operating for more than one year is calculated to be approximately 0.4869453. The following data represent the simulated lifetimes (in years) for a random sample of 10 remote controls: 1.81331943, 0.03451881, 0.59331475, 0.76090354, 0.49119571, 1.19811332, 2.38026149, 1.18567619, 0.76650414, 2.22910072.

First, it is critical to scientifically verify the existence and uniqueness of the MLEs. This verification can be examined visually using a 4D profile chart of the log-likelihood function, as shown in Figure 1. The chart is generated assuming that the scale parameters (β_1 and β_2) are equal and highlight a region where a global maximum is likely to exist (dark red spot). Next, starting values for the model parameters are required for the optimization process. These starting values can be obtained by profiling the log-likelihood through Monte Carlo simulations. Using the R package nleqslv, all considered method-strategy pairs successfully calculated the MLEs for the model parameters as $\hat{\boldsymbol{\theta}}_s = (3.2746207, 0.9473735, 2.0456769, 1.9703744)$. The estimated reliability of the remote control exceeding one year, based on these MLEs, is approximately 0.5369085. Notably, this reliability estimate can be improved by increasing the sample size.

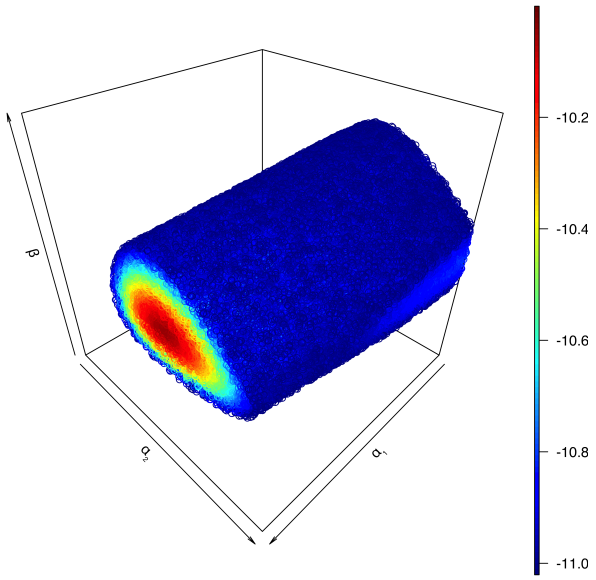

Figure 1: 4D profiling of the log-likelihood assuming a serial system model.

Example 4.2. Consider the following practical scenario for a parallel system. Suppose a portable power bank requires two rechargeable lithium-ion batteries connected in parallel. In this configuration, the system continues functioning as long as at least one battery is operational, but it stops working once both batteries are out of charge. Two independent companies manufacture the batteries, and both claim that their batteries have a lifespan of up to one year under typical usage conditions. Let the lifetime of batter from company A follow the $Weibull(\alpha_1 = 3, \beta_1 = 2)$, while the lifetime of batter from company B follows the same distribution but with $\alpha_2 = 0.75$. Thus, the actual reliability of the portable power bank exceeding one year is equal to 0.9473329, and the following data simulate the lifetimes of a random sample of 10 portable power banks (in years): 2.102093 1.165120 1.076597 1.016960 2.085863 3.195779 8.004690 1.665308 2.298227, 3.163854.

Once again, it is essential to scientifically confirm the existence and uniqueness of the MLEs. This confirmation can be visually assessed using a 4D profile chart of the log-likelihood function, as depicted in Figure 2. The chart assumes that the scale parameters (β_1 and β_2) are equal, highlighting a region where a global maximum might be locatable (a dark red spot). Next, initial values for the model parameters are necessary to facilitate the optimization process. These values can be derived by profiling the log-likelihood through Monte Carlo simulations. Employing the R package nleqslv,

all tested method-strategy pairs successfully computed the MLEs for the model parameters as $\hat{\boldsymbol{\theta}}_p = (2.5462896, 0.6292557, 2.0193981, 1.1215490)$. Based on these MLEs, the estimated reliability of the remote control operating for more than one year is approximately 0.9068405. This reliability estimate can be further improved by increasing the sample size.

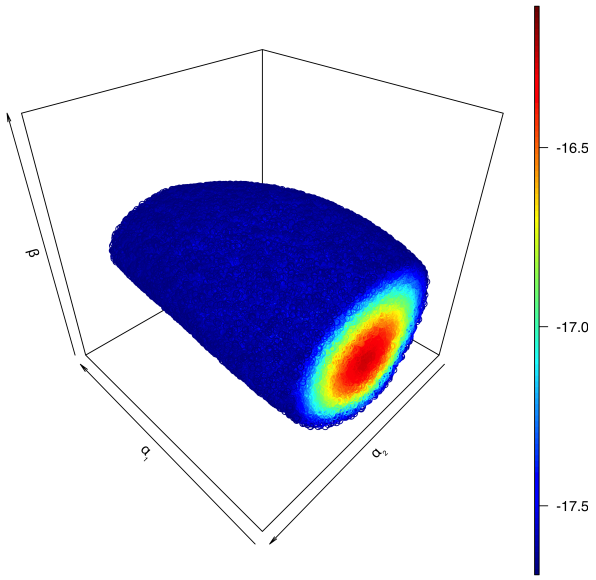

Figure 2: 4D profiling of the log-likelihood assuming a parallel system model.

4.3 Assessing Method-Strategy Pairs

To numerically assess and compare the performances of the 14 method-strategy pairs in the **R** package **nleqslv**, 1,000 Monte Carlo simulation runs are carried out assuming various values for the model parameters (Table 2) to calculate the following statistical/computational metrics:

- Convergence speed is measured by calculating the average of the required iterations to reach convergence and the elapsed time (in seconds) required to declare convergence. In this study, a restrictive approach is considered; that is, it is assumed that the convergence is declared (i.e., solutions are obtained) only when **nleqslv** returns a termination code (**termcd**) equal to 1. According to its documentation, the latter code indicates that "Function criterion is near zero. Convergence of function values has been achieved." Nevertheless, researchers may accept results when **termcd** is 2 or 3. If the termination code is 2, then "x-values within tolerance. This means that the relative distance between two consecutive x-values is smaller than **xtol** but that the function value criterion is still larger than **ftol**. Function values may not be near zero; therefore, the user must check if function values are acceptably small." In contrast, when the termination code is 3, then "No better point found. This means that the algorithm has stalled and cannot find an acceptable new point. This may or may not indicate acceptably small function values."
- Estimation efficiency is evaluated using two goodness-of-fit criteria, namely, the average absolute difference between the actual and estimated RFs (D_{abs}) and the maximum absolute difference between the actual and estimated RFs (D_{max}). These criteria are computed as follows:

$$D_{\text{abs}} = \frac{1}{n \times M} \sum_{j=1}^{M} \sum_{i=1}^{n} \left| R(x_i; \alpha_1, \alpha_2, \beta_1, \beta_2) - R(x_i; \hat{\alpha}_{1j}, \hat{\alpha}_{2j}, \hat{\beta}_{1j}, \hat{\beta}_{2j}) \right|,$$

and

$$D_{\text{max}} = \frac{1}{M} \sum_{j=1}^{M} \max_{i=1,\ldots,n} \left| R(x_i; \alpha_1, \alpha_2, \beta_1, \beta_2) - R(x_i; \hat{\alpha}_{1j}, \hat{\alpha}_{2j}, \hat{\beta}_{1j}, \hat{\beta}_{2j}) \right|.$$

Examples of recent papers that incorporated these two measurements based on CDFs instead of RFs (1) and (4) include, but are not limited to, Ali et al. [27], Alam and Nassar [28], Alam [29], and Caeiro and Norouzirad [30] and some of their references cited therein. These two measurements take values between 0 and 1 because the RF values lie within this range. A smaller value for these metrics indicates a better goodness-of-fit.

- The estimated model parameters depart from the true parameters due to sampling error. A potential factor that might increase sampling error is the choice of initial values for the model parameters. Therefore, it is important to investigate the effects of the choice of initial values on convergence. This is achieved by calculating the convergence percentage given a certain departure percentage from the actual model parameters (e.g., $\boldsymbol{\theta} \pm 10\%$ and $\boldsymbol{\theta} \pm 5\%$). This is sometimes called robustness evaluation or sensitivity analysis. The initial values for the parameters are obtained as follows. Suppose that the aim is to assess convergence assuming $\theta \pm 10\%$ with $\alpha_1 = 2, \alpha_2 = 4, \beta_1 = 2, \beta_2 = 4$, then the convergence percentage for each simulation setting is obtained by assuming the initial values to be 1.8, 3.6, 1.8, 3.6, respectively, then the simulations are conducted again with initial values to be 2.2, 4.4, 2.2, 4.4, respectively. The averages of the resulting percentages from both cases are computed to represent the convergence rate assuming $\theta \pm 10\%$.

Table 2: Values considered for the model parameters, assuming sample sizes $n = 10(10)100, 250, 500, 1000$.

$\boldsymbol{\alpha}_1$	$\boldsymbol{\alpha}_2$	$\boldsymbol{\beta}_1$	$\boldsymbol{\beta}_2$
0.5	2	0.5	2
0.5	4	0.5	2
2	4	0.5	2
0.5	2	0.5	4
0.5	4	0.5	4
2	4	0.5	4
0.5	2	2	0.5
0.5	4	2	0.5
2	4	2	0.5
0.5	2	2	4
0.5	4	2	4
2	4	2	4
0.5	2	4	0.5
0.5	4	4	0.5
2	4	4	0.5
0.5	2	4	2
0.5	4	4	2
2	4	4	2

To incorporate the power of data visualization instead of using traditional means such as tabulating data in a table, heat maps are considered to represent the assessment metrics. Odd-numbered Figures 3 to 13 show the head maps that illustrate the outcomes of Monte Carlo simulations for the serial system model, while even-numbered Figures 4 to 14 are the counterparts of the previous ones, assuming the parallel system model. Based on the provided figures, one can make the following observations:

- Overall, as the sample size n increases, all metrics improve for all pairs, except for the elapsed time, which logically increases as the sample size increases.
- As expected, the convergence percentages slightly improve as the initial values become closer to the actual values. However, the convergence percentages of Newton's and Broyden's methods without any strategy are incredibly unsatisfactory in all settings. Furthermore, under the parallel system model,

the convergence percentages are also unsatisfactory for all method-strategy pairs except for Newton's method with the hook strategy when the values of the scale parameters are relatively small and the values of the shape parameters are relatively large. It is noted that **nleqslv** documentation mentioned that choosing a global strategy for a specific scenario often involves trial and error. If a trust region method fails, it is advisable to try a line search strategy. Both approaches have their strengths, but neither consistently outperforms the other.

- Assuming convergence is declared (i.e., **termcd = 1**), all 14 method-strategy pairs performed similarly when estimating the reliability in terms of D_{abs}. Slight differences are noticeable for relatively small, and the values of the shape parameters are relatively large, assuming both models. In contrast, $D_{\max}$ provided clearer differences in estimation efficiency. Nevertheless, as the sample sizes increase, all differences become trivial.
- With regards to the number of iterations, Newton's method required fewer iterations compared to Broyden's method, which aligns with the well-known fact that Newton's method exhibits quadratic convergence. In contrast, Broyden's method has a superlinear convergence rate. Across all settings, Newton's method, regardless of the chosen strategy, consistently achieved the smallest number of iterations. Increasing the sample size did not significantly reduce the number of iterations.
- Regarding elapsed time, although Broyden's method has a slower convergence rate than Newton's method, it requires less time to reach convergence. This result is expected because Broyden's method approximates the inverse of the Jacobian matrix (or inverse of the Hessian in optimization). In contrast, Newton's method computes it explicitly at each iteration, increasing computational overhead.

4.4 Overall Performance Analysis

This subsection summarizes the findings by determining the best and worst method-strategy pairs across all cases considered. Using the sample size and the values of the model parameters as grouping factors, the metrical results are normalized using the Min-Max normalization approach to make all metrics scaled in the range (0, 1). Afterwards, the normalized values are ranked using minimum ranking approach. Note that this normalization is applied to convergence percentages, while its reverse version is used for the remaining metrics since their small values, especially those close to 0, indicate good performance. For more details about this normalization, see Han et al. [31]. Tables 3 and 4 (Tables 5 and 6) summarize the best-performed and worst-performed method-strategy pairs, assuming the serial (parallel) model. A best-performing method-strategy pair has the highest percentage of achieving a ranking of 1. In contrast, the worst-performing method-strategy pair has the highest percentage of attaining a ranking of 14.

From Table 3, it is clear that `N-hook` consistently outperformed all other method-strategy pairs in terms of convergence rate. The goodness-of-fit metric D_{abs} showed that N-none provided the best performance, while for $D_{\max}$, `N-none` achieved the top performance. Regarding the number of iterations, `N-gline` demonstrated the best performance. On the other hand, when considering elapsed time, `B-none` emerged as the fastest method, achieving the best performance, as expected, since this method does not evaluate the whole Hessian matrix at each iteration or incorporate additional strategies. Conversely, Table 4 reveals that `B-none` performed the worst in terms of convergence rates, while `B-hook` underperformed for the goodness-of-fit metrics, achieving the worst performance for both D_{abs} and $D_{\max}$. Additionally, `B-hook` was the worst-performing method regarding the number of iterations. In comparison, `N-none` performed the worst in elapsed time due to its nature of evaluating the whole Hessian matrix at each iteration. Similar trends can be observed in Tables 5 and 6 for the parallel methods. Specifically, `N-hook` outperformed other methods in convergence, while `N-none` demonstrated the best D_{abs}. However, `B-hook` showed the worst performance in multiple metrics, including D_{abs} and $D_{\max}$, confirming its underperformance relative to other methods. Overall, `N-hook` stands out as the best-performing method-strategy pair in terms of convergence rates. `B-none` and `B-hook` often exhibit poor performance across most metrics.

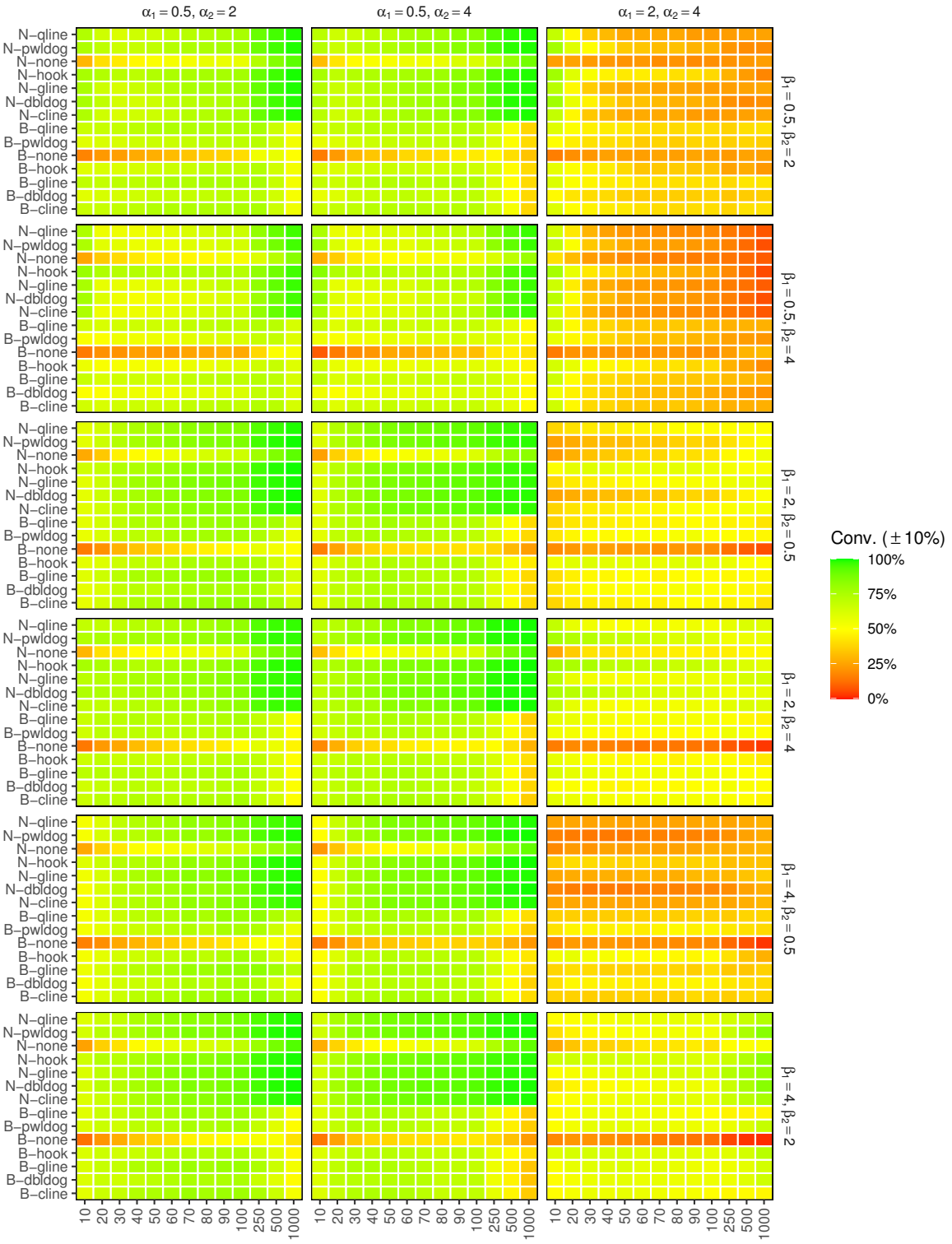

Figure 3: Convergence percentages (%) assuming a serial system model and $\boldsymbol{\theta} \pm 10\%$.

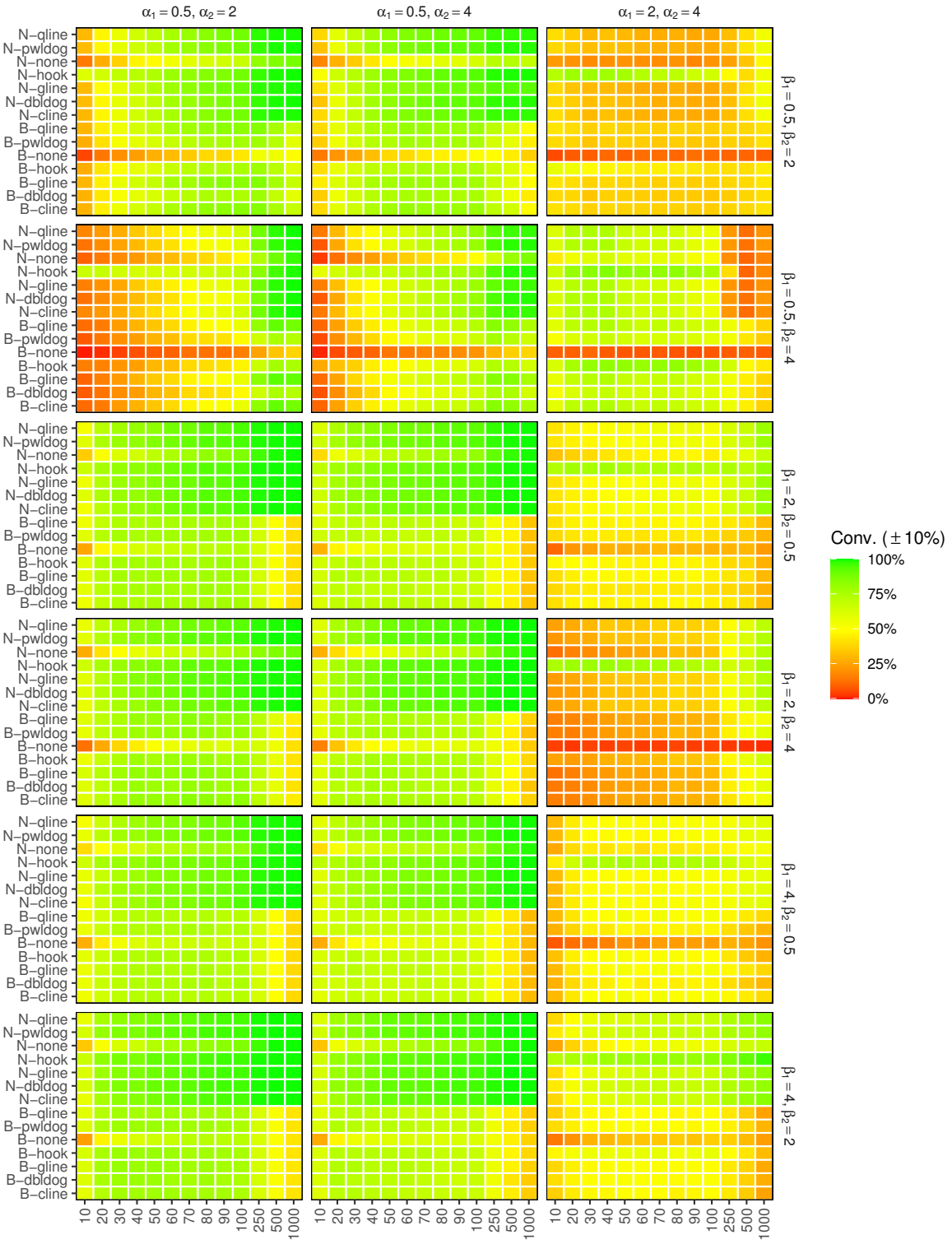

Figure 4: Convergence percentages (%) assuming a parallel system model and $\boldsymbol{\theta} \pm 10\%$.

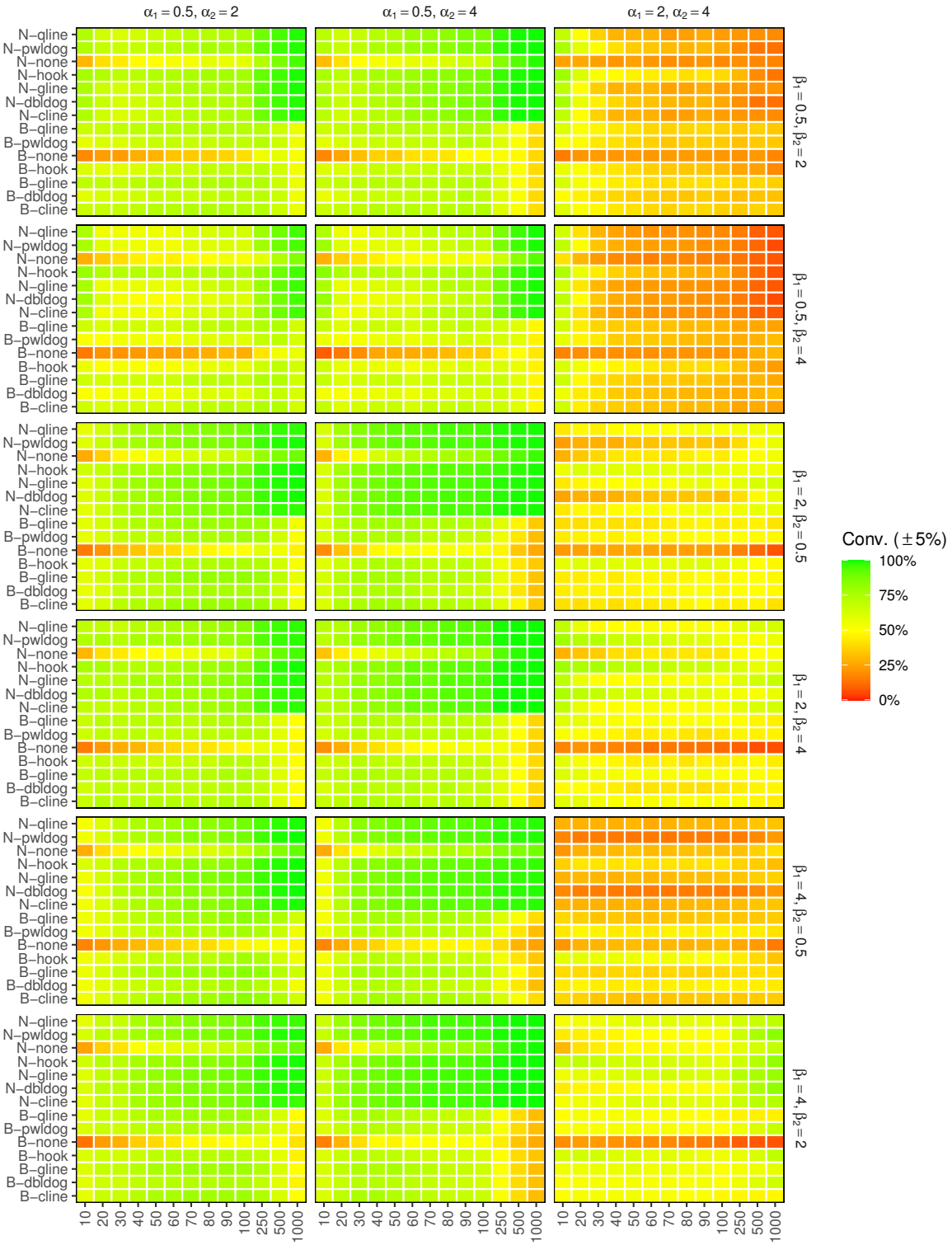

Figure 5: Convergence percentages (%) assuming a serial system model and $\boldsymbol{\theta} \pm 5\%$.

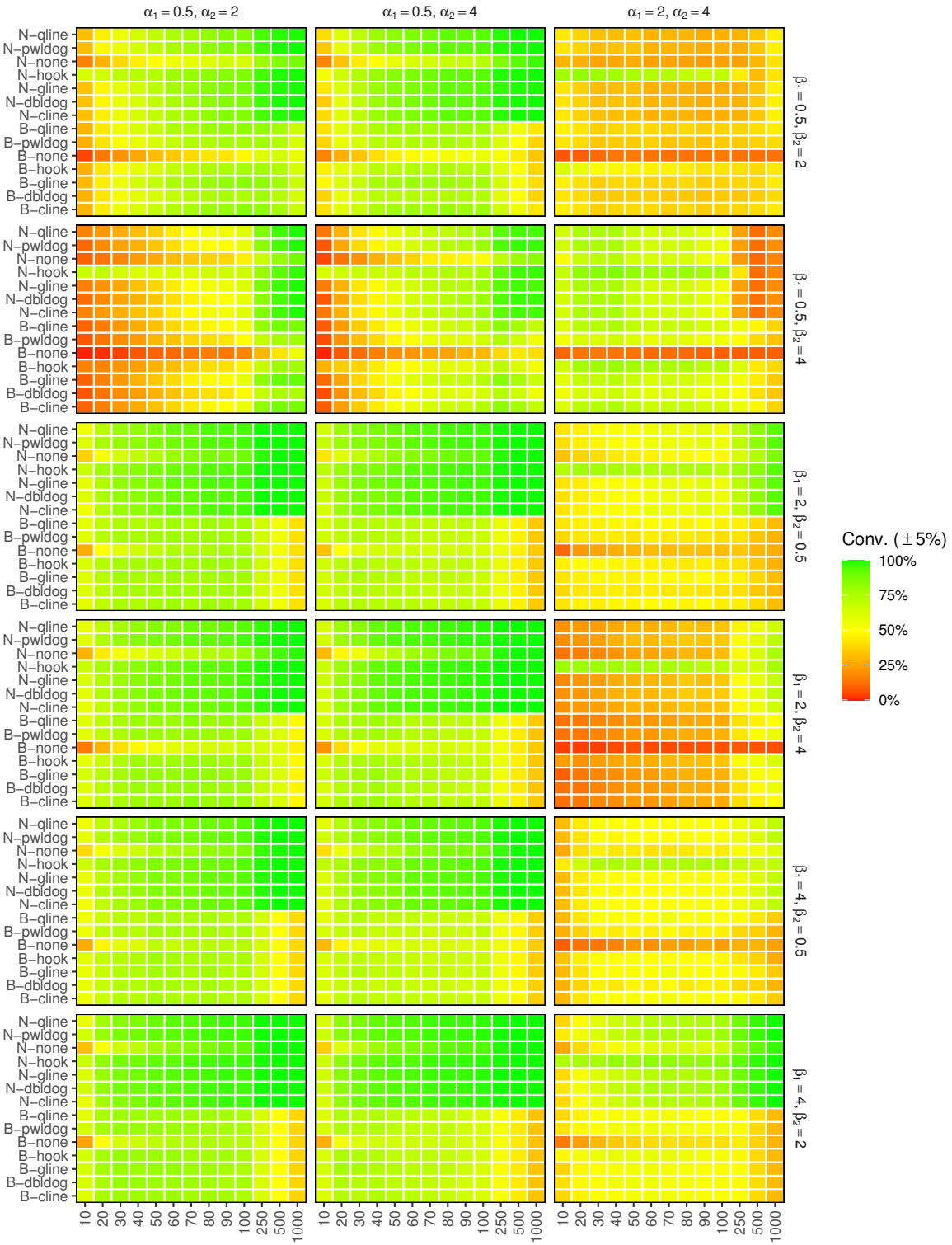

Figure 6: Convergence percentages (%) assuming a parallel system model and $\boldsymbol{\theta} \pm 5\%$.

Figure 7: Simulated D_{abs} assuming a serial system model.

AdvantaCore Scientific Publishing

Figure 8: Simulated D_{abs} assuming a parallel system model.

AdvantaCore Scientific Publishing

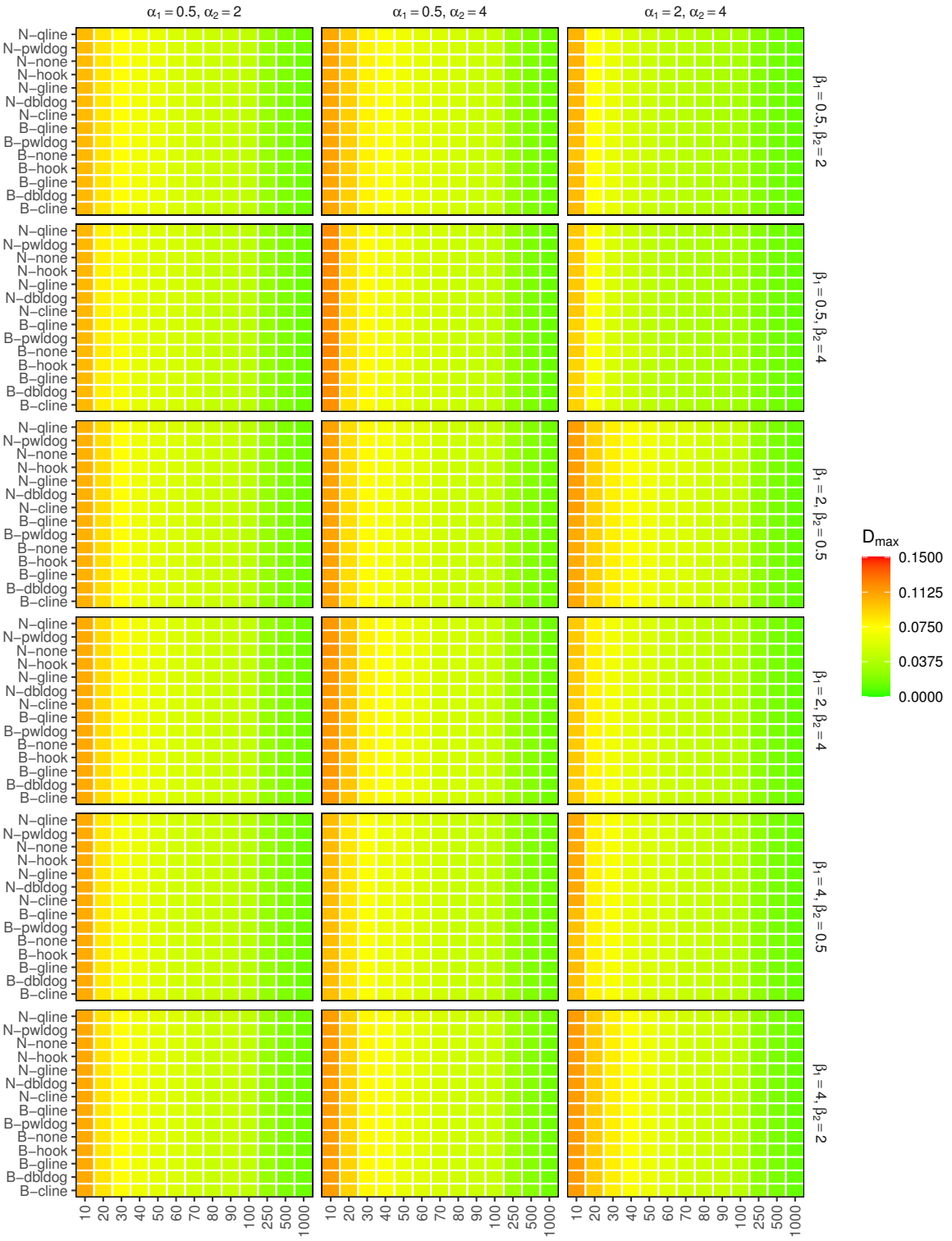

Figure 9: Simulated $D_{\max}$ assuming a serial system model.

AdvantaCore Scientific Publishing

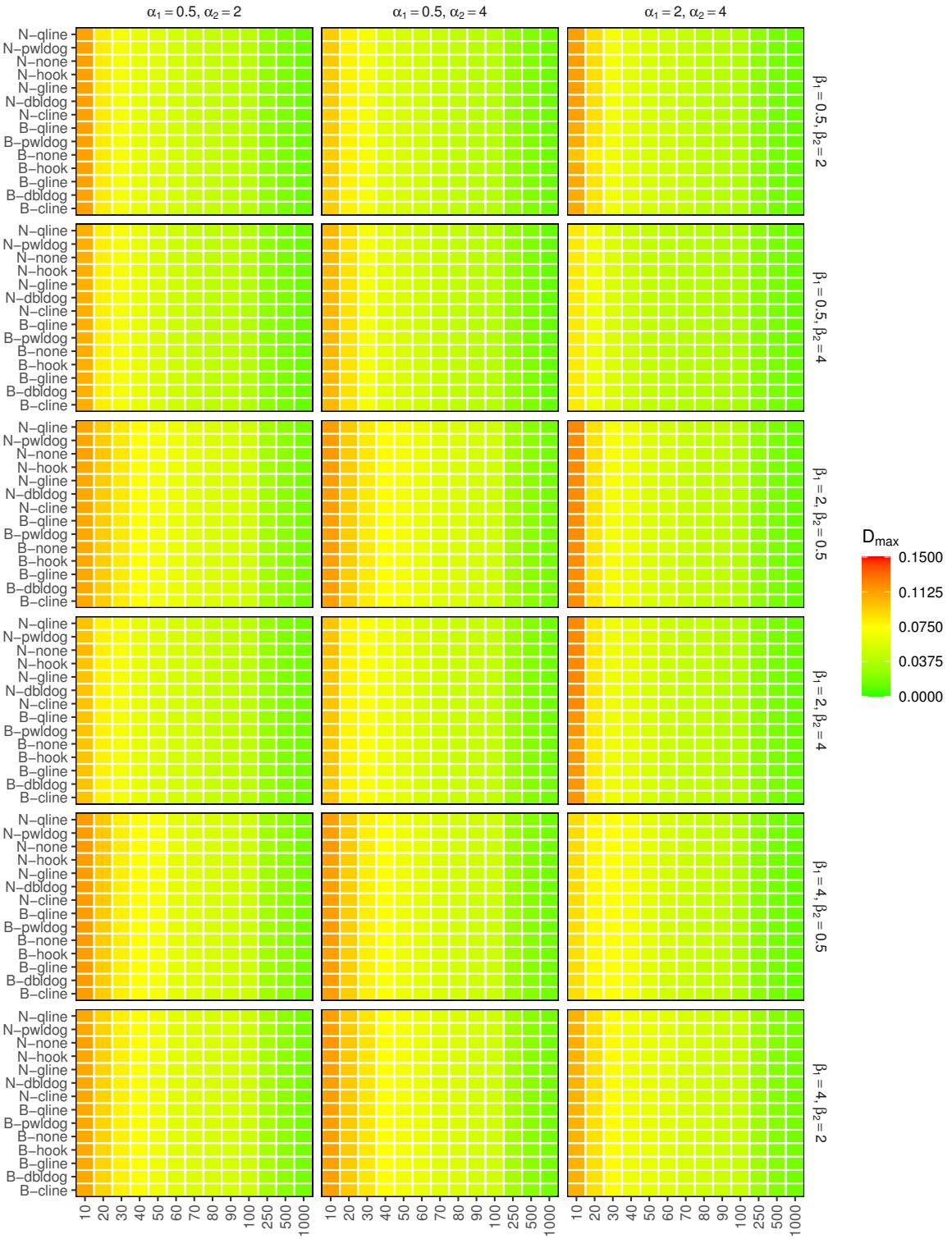

Figure 10: Simulated $D_{\max}$ assuming a parallel system model.

AdvantaCore Scientific Publishing

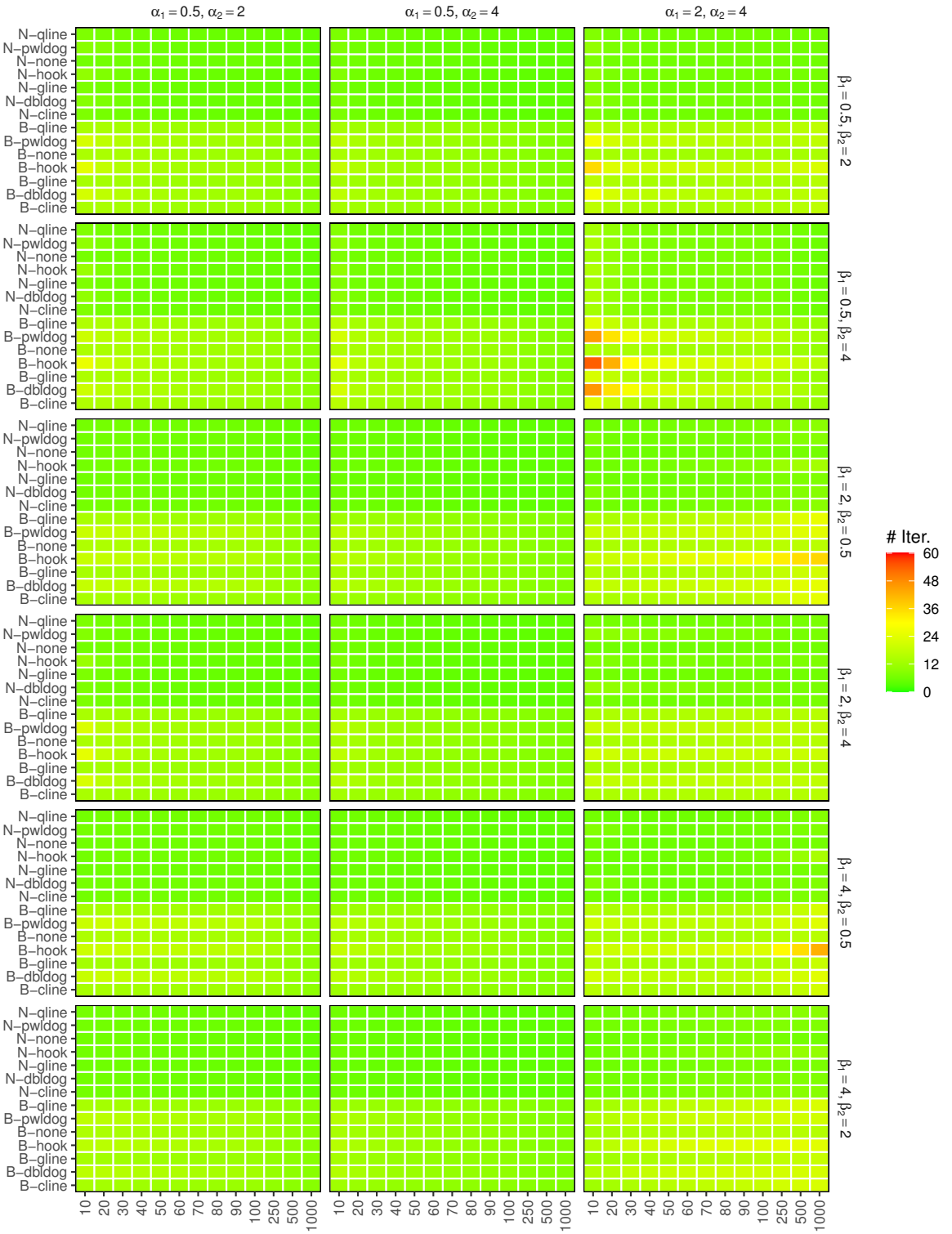

Figure 11: Simulated averages of the number of iterations to convergence assuming a serial system model.

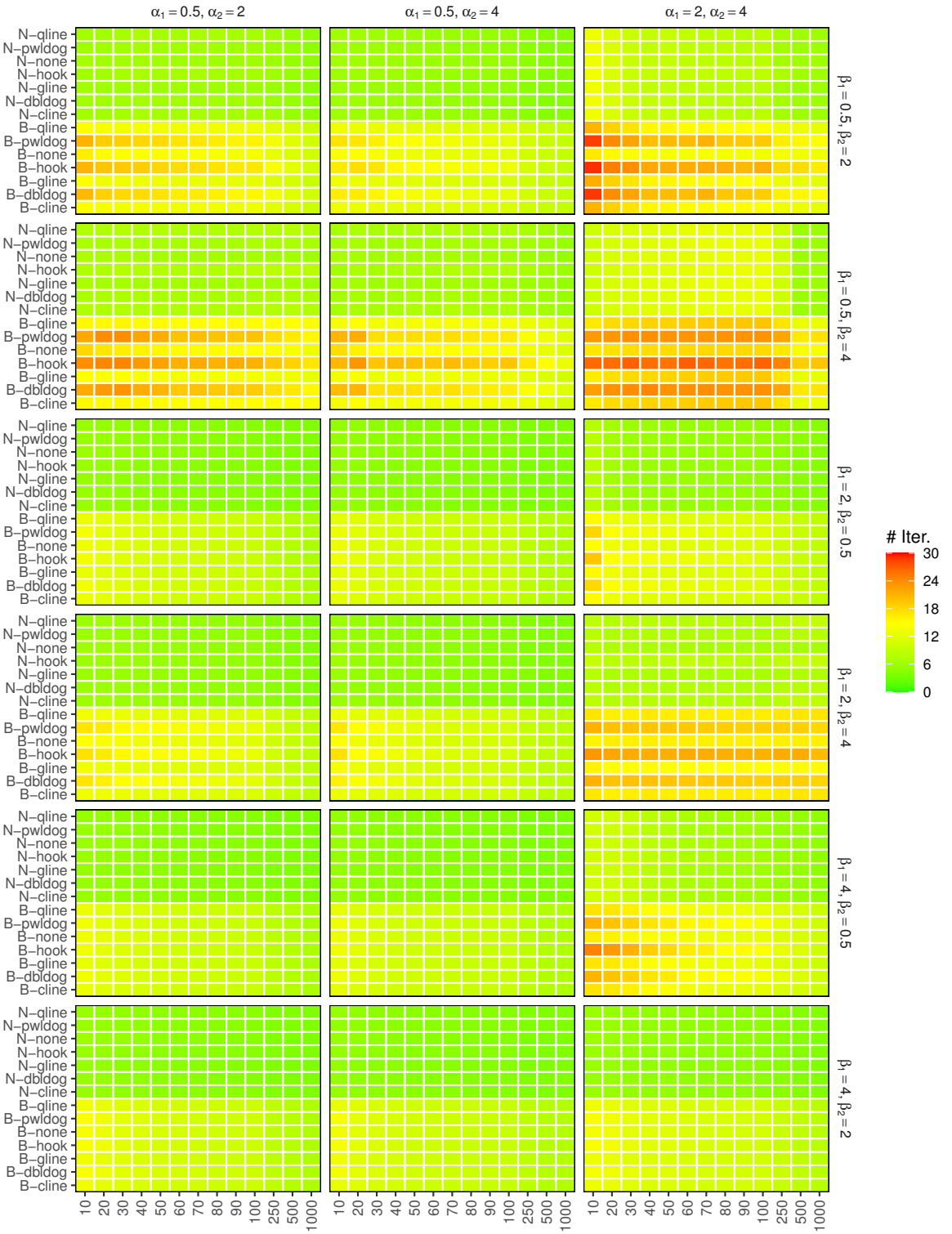

Figure 12: Simulated averages of the number of iterations to convergence assuming a parallel system model.

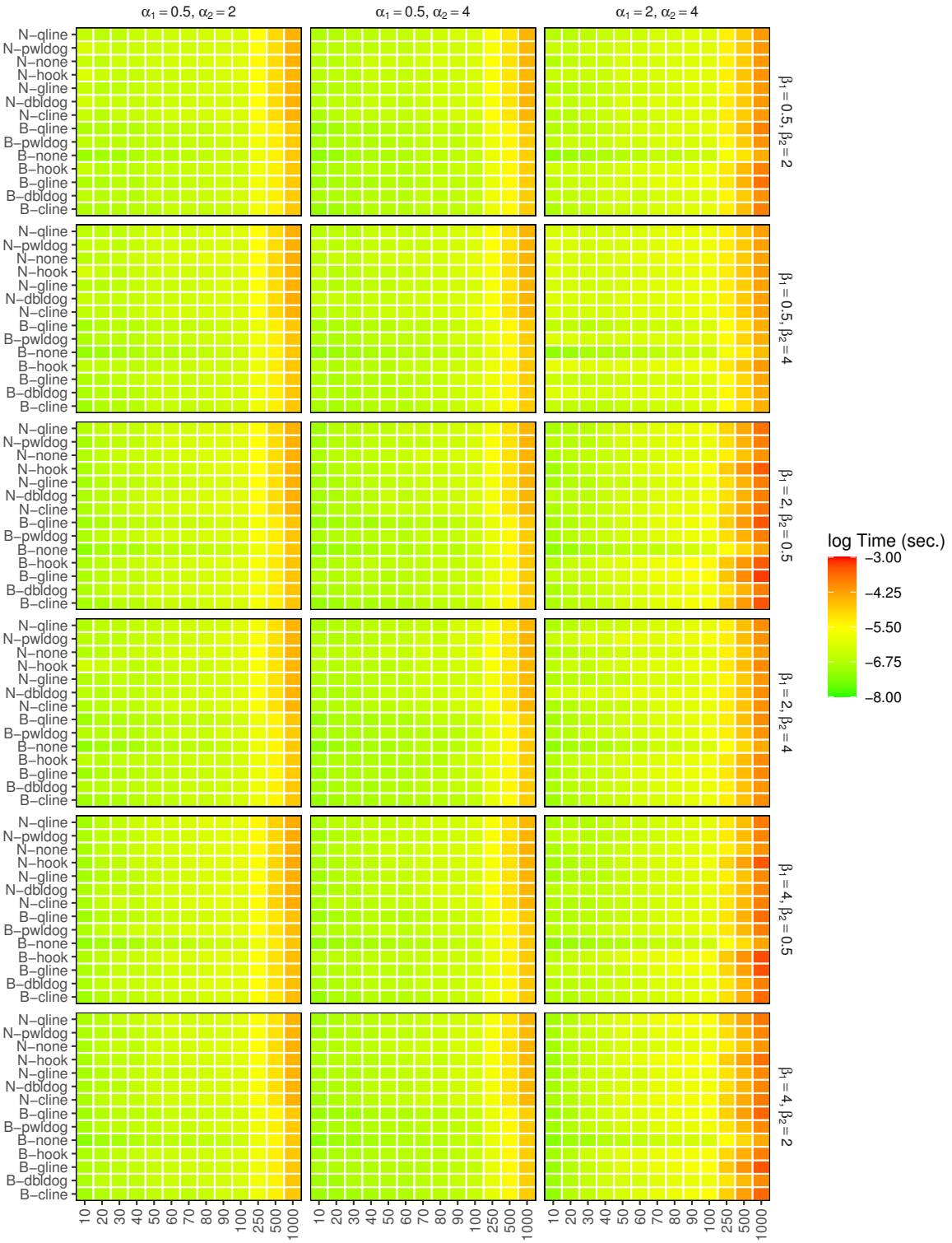

Figure 13: Simulated elapsed times to convergence (in seconds) assuming a serial system model.

AdvantaCore Scientific Publishing

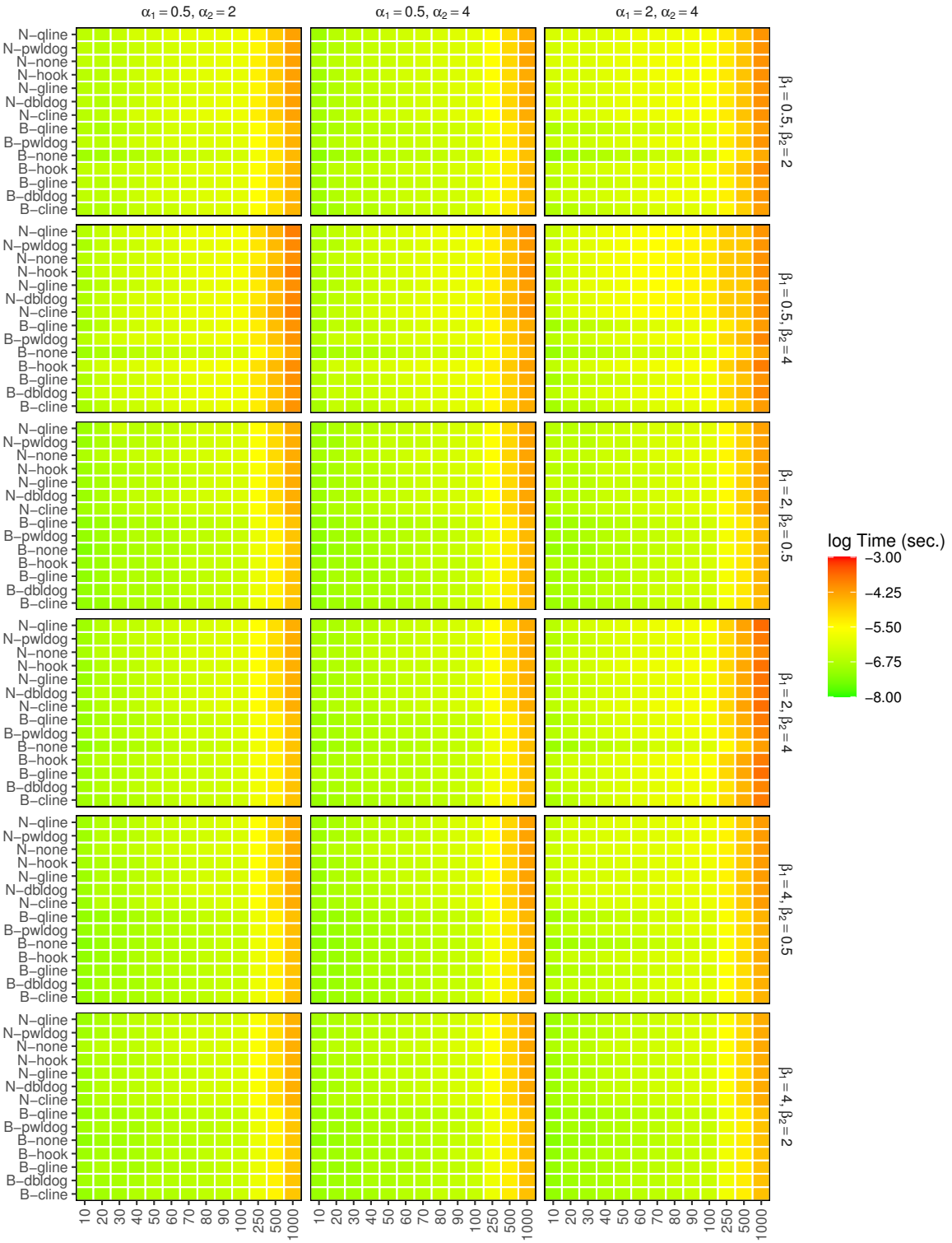

Figure 14: Simulated elapsed times to convergence (in seconds) assuming a parallel system model.

AdvantaCore Scientific Publishing

Table 3: Overall best-performing pairs assuming the serial system model.

Metric	Method	Best Performance (%)
Convergence ($\theta \pm 5\%$)	`N-hook`	37.18
Convergence ($\theta \pm 10\%$)	`N-hook`	36.75
D_{abs}	`N-none`	81.20
D_{max}	`N-none`	69.66
Number of Iterations	`N-gline`	91.45
Elapsed Time	`B-none`	84.19

Table 4: Overall worst-performing pairs assuming the serial system model.

Metric	Method	Best Performance (%)
Convergence ($\theta \pm 5\%$)	`B-none`	87.18
Convergence ($\theta \pm 10\%$)	`B-none`	87.61
D_{abs}	`B-hook`	69.23
D_{max}	`B-hook`	52.14
Number of Iterations	`B-hook`	42.74
Elapsed Time	`N-none`	36.75

Table 5: Overall best-performing pairs assuming the parallel system model.

Metric	Method	Best Performance (%)
Convergence ($\theta \pm 5\%$)	`N-hook`	64.53
Convergence ($\theta \pm 10\%$)	`N-hook`	71.79
D_{abs}	`N-none`	73.50
D_{max}	`N-none`	59.83
Number of Iterations	`N-gline`	98.72
Elapsed Time	`B-none`	66.67

Table 6: Overall worst-performing pairs assuming the parallel system model.

Metric	Method	Best Performance (%)
Convergence ($\theta \pm 5\%$)	`B-none`	93.59
Convergence ($\theta \pm 10\%$)	`B-none`	95.73
D_{abs}	`B-hook`	71.37
D_{max}	`B-hook`	53.85
Number of Iterations	`B-hook`	35.90
Elapsed Time	`N-none`	51.71

5 Concluding Discussion

This study has systematically examined the performance of 14 method-strategy pairs implemented in the **nleqslv** package for solving nonlinear systems of equations under challenging settings. By employing two distinct models based on reliability analysis (i.e., serial and parallel system configurations), the study

evaluated vital performance metrics, including convergence speed, estimation efficiency, robustness, and computational overhead.

Monte Carlo simulations confirmed that Newton's method required fewer iterations due to its quadratic convergence rate. In contrast, Broyden's method achieved faster runtimes by approximating the inverse of the Jacobian, thereby reducing computational cost. The simulations also demonstrated a slight impact of initial value selection and parameter variations on convergence behavior, underscoring the importance of robustness evaluation. Notably, all 14 method-strategy pairs exhibited similar performance in estimation efficiency. These findings emphasize that the choice of global strategies (e.g., trust region versus line search) heavily depends on the problem's structure, with the double dogleg method emerging as a robust performer across multiple scenarios, consistent with prior recommendations in the **nleqslv** documentation.

It is worth noting that the findings of this study align closely with the guidance provided by the author of the **nleqslv** package. The simulation results demonstrate that no single global strategy is universally optimal across all model structures, parameter settings, and performance metrics. Instead, the choice of method–strategy pair is inherently problem-dependent, with trust-region and line-search strategies each exhibiting advantages under different conditions. Although the double dogleg method performs reliably in many scenarios, it does not consistently dominate. These empirical results reinforce the practical recommendation that selecting an appropriate global strategy in **nleqslv** package often requires trial and error, guided by the specific nonlinear system's characteristics and numerical behavior.

This study opens several promising avenues for future exploration. One exciting direction is to extend the applicability of the **nleqslv** package to Bayesian optimization problems, such as posterior estimation or Bayesian hierarchical models, offering valuable insights into its utility in a probabilistic framework. Another potential research direction is to conduct a detailed comparative study of the **nleqslv** package and the **R** built-in `nlm` function, highlighting their relative strengths and weaknesses, particularly for unconstrained problems and motivating further methodological advancements.

Funding Statement

The author declared that no specific funding was received for this work.

Data Availability

Data sharing is not applicable to this article.

Conflicts of Interest

The author declares that he has no conflicts of interest regarding this study.

Author Contributions

The author is solely responsible for all components of this work.

AI Usage Statement

No AI-based tools were used in the generation, writing, revising, analysis, or interpretation of the research data and findings presented in this article.

Appendix

This appendix provides detailed derivations of the gradient and Hessian matrices used in the optimization process for the serial and parallel systems described in Section 3. Under the serial system setting, the elements of the objective function gradient and its hessian required for implementing Newton's

method are defined as follows:

$$\frac{\partial \ell_s}{\partial \alpha_j} = \sum_{i=1}^{n} \frac{A_j^s(t_i,\boldsymbol{\theta})}{f_s(t_i;\boldsymbol{\theta})}, \quad \frac{\partial \ell_s}{\partial \beta_j} = \sum_{i=1}^{n} \frac{B_j^s(t_i,\boldsymbol{\theta})}{f_s(t_i;\boldsymbol{\theta})}$$

$$\frac{\partial^2 \ell_s}{\partial \alpha_j^2} = \sum_{i=1}^{n} \frac{A_{jj}^s(t_i;\boldsymbol{\theta}) f_s(t_i;\boldsymbol{\theta}) - \left[A_j^s(t_i,\boldsymbol{\theta})\right]^2}{f_s^2(t_i;\boldsymbol{\theta})}, \quad \frac{\partial^2 \ell_s}{\partial \beta_j^2} = \sum_{i=1}^{n} \frac{B_{jj}^s(t_i;\boldsymbol{\theta}) f_s(t_i;\boldsymbol{\theta}) - \left[B_j^s(t_i,\boldsymbol{\theta})\right]^2}{f_s^2(t_i;\boldsymbol{\theta})}$$

$$\frac{\partial^2 \ell_s}{\partial \alpha_j \partial \beta_k} = \frac{\partial^2 \ell_s}{\partial \alpha_k \partial \beta_j} = \sum_{i=1}^{n} \frac{C_{jk}^s(t_i;\boldsymbol{\theta}) f_s(t_i;\boldsymbol{\theta}) - A_j^s(t_i,\boldsymbol{\theta}) B_k^s(t_i,\boldsymbol{\theta})}{f_s^2(t_i;\boldsymbol{\theta})}, \quad j,k = 1,2,$$

$$\frac{\partial^2 \ell_s}{\partial \alpha_1 \partial \alpha_2} = \frac{\partial^2 \ell_s}{\partial \alpha_2 \partial \alpha_1} = \sum_{i=1}^{n} \frac{D_1^s(t_i;\boldsymbol{\theta}) f_s(t_i;\boldsymbol{\theta}) - A_1^s(t_i,\boldsymbol{\theta}) A_2^s(t_i,\boldsymbol{\theta})}{f_s^2(t_i;\boldsymbol{\theta})},$$

$$\frac{\partial^2 \ell_s}{\partial \beta_1 \partial \beta_2} = \frac{\partial^2 \ell_s}{\partial \beta_2 \partial \beta_1} = \sum_{i=1}^{n} \frac{D_2^s(t_i;\boldsymbol{\theta}) f_s(t_i;\boldsymbol{\theta}) - B_1^s(t_i,\boldsymbol{\theta}) B_2^s(t_i,\boldsymbol{\theta})}{f_s^2(t_i;\boldsymbol{\theta})},$$

where

$$A_1^s(t_i;\boldsymbol{\theta}) = \frac{\partial f_1(t_i;\alpha_1,\beta_1)}{\partial \alpha_1}\left[1 - F_2(t_i;\alpha_2,\beta_2)\right] - f_2(t_i;\alpha_2,\beta_2)\frac{\partial F_1(t_i;\alpha_1,\beta_1)}{\partial \alpha_1},$$

$$A_2^s(t_i;\boldsymbol{\theta}) = \frac{\partial f_2(t_i;\alpha_2,\beta_2)}{\partial \alpha_2}\left[1 - F_1(t_i;\alpha_1,\beta_1)\right] - f_1(t_i;\alpha_1,\beta_1)\frac{\partial F_2(t_i;\alpha_2,\beta_2)}{\partial \alpha_2},$$

$$B_1^s(t_i;\boldsymbol{\theta}) = \frac{\partial f_1(t_i;\alpha_1,\beta_1)}{\partial \beta_1}\left[1 - F_2(t_i;\alpha_2,\beta_2)\right] - f_2(t_i;\alpha_2,\beta_2)\frac{\partial F_1(t_i;\alpha_1,\beta_1)}{\partial \beta_1},$$

$$B_2^s(t_i;\boldsymbol{\theta}) = \frac{\partial f_2(t_i;\alpha_2,\beta_2)}{\partial \beta_2}\left[1 - F_1(t_i;\alpha_1,\beta_1)\right] - f_1(t_i;\alpha_1,\beta_1)\frac{\partial F_2(t_i;\alpha_2,\beta_2)}{\partial \beta_2},$$

$$A_{11}^s(t_i;\boldsymbol{\theta}) = \frac{\partial^2 f_1(t_i;\alpha_1,\beta_1)}{\partial \alpha_1^2}\left[1 - F_2(t_i;\alpha_2,\beta_2)\right] - f_2(t_i;\alpha_2,\beta_2)\frac{\partial^2 F_1(t_i;\alpha_1,\beta_1)}{\partial \alpha_1^2},$$

$$A_{22}^s(t_i;\boldsymbol{\theta}) = \frac{\partial^2 f_2(t_i;\alpha_2,\beta_2)}{\partial \alpha_2^2}\left[1 - F_1(t_i;\alpha_1,\beta_1)\right] - f_1(t_i;\alpha_1,\beta_1)\frac{\partial^2 F_2(t_i;\alpha_2,\beta_2)}{\partial \alpha_2^2},$$

$$B_{11}^s(t_i;\boldsymbol{\theta}) = \frac{\partial^2 f_1(t_i;\alpha_1,\beta_1)}{\partial \beta_1^2}\left[1 - F_2(t_i;\alpha_2,\beta_2)\right] - f_2(t_i;\alpha_2,\beta_2)\frac{\partial^2 F_1(t_i;\alpha_1,\beta_1)}{\partial \beta_1^2},$$

$$B_{22}^s(t_i;\boldsymbol{\theta}) = \frac{\partial^2 f_2(t_i;\alpha_2,\beta_2)}{\partial \beta_2^2}\left[1 - F_1(t_i;\alpha_1,\beta_1)\right] - f_1(t_i;\alpha_1,\beta_1)\frac{\partial^2 F_2(t_i;\alpha_2,\beta_2)}{\partial \beta_2^2},$$

$$C_{11}^s(t_i;\boldsymbol{\theta}) = \frac{\partial^2 f_1(t_i;\alpha_1,\beta_1)}{\partial \alpha_1 \partial \beta_1}\left[1 - F_2(t_i;\alpha_2,\beta_2)\right] - f_2(t_i;\alpha_2,\beta_2)\frac{\partial^2 F_1(t_i;\alpha_1,\beta_1)}{\partial \alpha_1 \partial \beta_1},$$

$$C_{22}^s(t_i;\boldsymbol{\theta}) = \frac{\partial^2 f_2(t_i;\alpha_2,\beta_2)}{\partial \alpha_2 \partial \beta_2}\left[1 - F_1(t_i;\alpha_1,\beta_1)\right] - f_1(t_i;\alpha_1,\beta_1)\frac{\partial^2 F_2(t_i;\alpha_2,\beta_2)}{\partial \alpha_2 \partial \beta_2},$$

$$C_{12}^s(t_i;\boldsymbol{\theta}) = -\frac{\partial f_1(t_i;\alpha_1,\beta_1)}{\partial \alpha_1}\frac{\partial F_2(t_i;\alpha_2,\beta_2)}{\partial \beta_2} - \frac{\partial f_2(t_i;\alpha_2,\beta_2)}{\partial \beta_2}\frac{\partial F_1(t_i;\alpha_1,\beta_1)}{\partial \alpha_1},$$

$$C_{21}^s(t_i;\boldsymbol{\theta}) = -\frac{\partial f_1(t_i;\alpha_1,\beta_1)}{\partial \beta_1}\frac{\partial F_2(t_i;\alpha_2,\beta_2)}{\partial \alpha_2} - \frac{\partial f_2(t_i;\alpha_2,\beta_2)}{\partial \alpha_2}\frac{\partial F_1(t_i;\alpha_1,\beta_1)}{\partial \beta_1},$$

$$D_1^s(t_i;\boldsymbol{\theta}) = -\frac{\partial f_1(t_i;\alpha_1,\beta_1)}{\partial \alpha_1}\frac{\partial F_2(t_i;\alpha_2,\alpha_2)}{\partial \beta_2} - \frac{\partial f_2(t_i;\alpha_2,\beta_2)}{\partial \alpha_2}\frac{\partial F_1(t_i;\alpha_1,\beta_1)}{\partial \alpha_1},$$

$$D_2^s(t_i;\boldsymbol{\theta}) = -\frac{\partial f_1(t_i;\alpha_1,\beta_1)}{\partial \beta_1}\frac{\partial F_2(t_i;\alpha_2,\alpha_2)}{\partial \beta_2} - \frac{\partial f_2(t_i;\alpha_2,\beta_2)}{\partial \alpha_2}\frac{\partial F_1(t_i;\alpha_1,\beta_1)}{\partial \beta_1},$$

AdvantaCore Scientific Publishing

such that

$$\frac{\partial f_j(t_i;\alpha_j,\beta_j)}{\partial \alpha_j} = f_j(t_i;\alpha_j,\beta_j)\left\{\alpha^{-1} + \log\left(\frac{t_i}{\beta_j}\right)\left[1-\left(\frac{t_i}{\beta_j}\right)^{\alpha_j}\right]\right\}, \tag{A.1}$$

$$\frac{\partial f_j(t_i;\alpha_j,\beta_j)}{\partial \beta_j} = \alpha_j\beta_j^{-1} f_j(t_i;\alpha_j,\beta_j)\left[\left(\frac{t_i}{\beta_j}\right)^{\alpha_j} - 1\right], \tag{A.2}$$

$$\begin{aligned}\frac{\partial^2 f_j(t_i;\alpha_j,\beta_j)}{\partial \alpha_j^2} = \frac{\partial f_j(t_i;\alpha_j,\beta_j)}{\partial \alpha_j}&\left\{\alpha^{-1} + \log\left(\frac{t_i}{\beta_j}\right)\left[1-\left(\frac{t_i}{\beta_j}\right)^{\alpha_j}\right]\right\}\\ &-f_j(t_i;\alpha_j,\beta_j)\left\{\alpha^{-2} + \left[\log\left(\frac{t_i}{\beta_j}\right)\right]^2\left(\frac{t_i}{\beta_j}\right)^{\alpha_j}\right\},\end{aligned} \tag{A.3}$$

$$\begin{aligned}\frac{\partial^2 f_j(t_i;\alpha_j,\beta_j)}{\partial \beta_j^2} =&\frac{\partial f_j(t_i;\alpha_j,\beta_j)}{\partial \beta_j}\left\{\alpha_j\beta_j^{-1}\left[\left(\frac{t_i}{\beta_j}\right)^{\alpha_j} - 1\right]\right\}\\ &- f_j(t_i;\alpha_j,\beta_j)\alpha_j\beta_j^{-2}\left[\left(\frac{t_i}{\beta_j}\right)^{\alpha_j} - 1\right]\\ &- f_j(t_i;\alpha_j,\beta_j)\alpha_j^2\beta_j^{-2}\left(\frac{t_i}{\beta_j}\right)^{\alpha_j},\end{aligned} \tag{A.4}$$

$$\begin{aligned}\frac{\partial^2 f_j(t_i;\alpha_j,\beta_j)}{\partial \alpha_j\partial \beta_j} =&\frac{\partial f_j(t_i;\alpha_j,\beta_j)}{\partial \alpha_j}\left\{\alpha_j\beta_j^{-1}\left[\left(\frac{t_i}{\beta_j}\right)^{\alpha_j} - 1\right]\right\}\\ &+ f_j(t_i;\alpha_j,\beta_j)\beta_j^{-1}\left[\left(\frac{t_i}{\beta_j}\right)^{\alpha_j} - 1\right]\\ &+ f_j(t_i;\alpha_j,\beta_j)\alpha_j\beta_j^{-1}\left(\frac{t_i}{\beta_j}\right)^{\alpha_j}\log\left(\frac{t_i}{\beta_j}\right),\end{aligned} \tag{A.5}$$

$$\frac{\partial F_j(t_i;\alpha_j,\beta_j)}{\partial \alpha_j} = \left[1-F_j(t_i;\alpha_j,\beta_j)\right]\left(\frac{t_i}{\beta_j}\right)^{\alpha_j}\log\left(\frac{t_i}{\beta_j}\right), \tag{A.6}$$

$$\frac{\partial F_j(t_i;\alpha_j,\beta_j)}{\partial \beta_j} = -\left[1-F_j(t_i;\alpha_j,\beta_j)\right]\alpha_j\beta_j^{-1}\left(\frac{t_i}{\beta_j}\right)^{\alpha_j}, \tag{A.7}$$

$$\frac{\partial^2 F_j(t_i;\alpha_j,\beta_j)}{\partial \alpha_j^2} = \left[1-F_j(t_i;\alpha_j,\beta_j)\right]\left[1-\left(\frac{t_i}{\beta_j}\right)^{\alpha_j}\right]\left[\log\left(\frac{t_i}{\beta_j}\right)\right]^2, \tag{A.8}$$

$$\frac{\partial^2 F_j(t_i;\alpha_j,\beta_j)}{\partial \beta_j^2} = \left[1-F_j(t_i;\alpha_j,\beta_j)\right]\alpha_j\beta_j^{-2}\left(\frac{t_i}{\beta_j}\right)^{\alpha_j}\left\{1+\alpha_j\left[1-\left(\frac{t_i}{\beta_j}\right)^{\alpha_j}\right]\right\}, \tag{A.9}$$

and

$$\frac{\partial^2 F_j(t_i;\alpha_j,\beta_j)}{\partial \alpha_j\partial \beta_j} = -\left[1-F_j(t_i;\alpha_j,\beta_j)\right]\beta^{-1}\left(\frac{t_i}{\beta_j}\right)^{\alpha_j}\left\{1+\alpha_j\log\left(\frac{t_i}{\beta_j}\right)\left[1-\left(\frac{t_i}{\beta_j}\right)^{\alpha_j}\right]\right\}, \tag{A.10}$$

for $j,k = 1,2$. To implement Newton's method under the parallel system scenario, the elements of the objective function's gradient and hessian required for optimization are the same as mentioned before, but they differ slightly in the internal terms $A_{11}^s(t_i;\boldsymbol{\theta})$ to $D_2^s(t_i;\boldsymbol{\theta})$ although these terms still incorporated Eq. (A.1) to Eq. (A.10). To obtain $A_{11}^p(t_i;\boldsymbol{\theta})$ to $C_{22}^p(t_i;\boldsymbol{\theta})$, replace the terms $\left[1-F_j(t_i;\alpha_j,\beta_j)\right]$ by $F_j(t_i;\alpha_j,\beta_j)$ in $A_{11}^s(t_i;\boldsymbol{\theta})$ to $C_{22}^s(t_i;\boldsymbol{\theta})$, and change any negative sign to a positive sign afterward. With regards to terms $C_{12}^p(t_i;\boldsymbol{\theta})$ to $D_2^p(t_i;\boldsymbol{\theta})$, they are obtained by converting the negative signs to positive signs in $C_{12}^s(t_i;\boldsymbol{\theta})$ to $D_2^s(t_i;\boldsymbol{\theta})$.

References

[1] Deuflhard, P. (2012). A short history of Newton's method. Documenta Mathematica, Optimization stories. 25-30.

[2] Searle, S. R., Casella, G., and McCulloch, C. E. (2009). *Variance components*. John Wiley & Sons, New York.

[3] Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations. Mathematics of computation. 19(92), 577-593.

[4] Nocedal, J., and Wright, S. J. (2006). *Numerical optimization*. New York, NY: Springer New York.

[5] Ye, H., Luo, L., and Zhang, Z. (2021). Approximate newton methods. Journal of Machine Learning Research. 22(66), 1-41.

[6] Erdogdu, M. A., and Montanari, A. (2015). Convergence rates of sub-sampled newton methods. Advances in Neural Information Processing Systems, vol. 28. Red Hook, New York: Curran Associates, Inc. pp. 3034–3042.

[7] Khorasani, S., and Mahoney, M W. (2016). Hybrid optimization methods for large-scale estimation problems. MIT conference proceedings. Available at https://optml.mit.edu.

[8] R Core Team. (2021). R: A language and environment for statistical computing. R foundation for statistical computing, Vienna, Austria.

[9] Nash, J. C. (2014). On best practice optimization methods in R. Journal of Statistical Software. 60, 1-14.

[10] Hasselman, B. (2023). nleqslv: Solve Systems of Nonlinear Equations. Version 3.3.5.

[11] Reichel, L., and Gragg, W. B. (1990). Algorithm 686: A variant of broyden's method for solving nonlinear systems of equations. ACM Transactions on Mathematical Software (TOMS). 16(1):93–97.

[12] Dennis Jr, J. E., and Schnabel, R. B. (1996). Numerical methods for unconstrained optimization and nonlinear equations. Society for Industrial and Applied Mathematics. Philadelphia, PA.

[13] Powell, M. J. D. (1970). A hybrid method for nonlinear equations. In Numerical Methods for Nonlinear Algebraic Equations, pages 87–114. Gordon and Breach.

[14] Moré, J. J. (1978). The Levenberg-Marquardt algorithm: implementation and theory. Numerical Analysis. 630:105–116.

[15] Varadhan, R., and Gilbert, P. (2010). BB: An R package for solving a large system of nonlinear equations and for optimizing a high-dimensional nonlinear objective function. Journal of statistical software, 32, 1-26.

[16] Reynaerts, J., Varadhan, R., and Nash, J. C. (2012). The convergence properties of the blp contraction mapping and alternative algorithms in r. In Proceedings of the use R 2010 Conference.

[17] Abdulkadirov, R., Lyakhov, P., and Nagornov, N. (2023). Survey of optimization algorithms in modern neural networks. Mathematics. 11(11), 2466.

[18] Xie, M., and Lai, C. D. (1996). Reliability analysis using an additive Weibull model with bathtub-shaped failure rate function. Reliability Engineering & System Safety. 52(1), 87-93.

[19] Gurvich, M. R., Dibenedetto, A. T., and Ranade, S. V. (1997). A new statistical distribution for characterizing the random strength of brittle materials. Journal of materials science. 32(10), 2559-2564.

[20] Nadarajah, S., and Kotz, S. (2005). On some recent modifications of Weibull distribution. IEEE Transactions on Reliability. 54(4), 561-562.

[21] Lai, C. D., Xie, M., and Murthy, D. N. P. (2005). Reply to" On some recent modifications of Weibull distribution". IEEE Transactions on Reliability. 54(4), 563-563.

[22] Lemonte, A. J., Cordeiro, G. M., and Ortega, E. M. (2014). On the additive Weibull distribution. Communications in Statistics-Theory and Methods. 43(10-12), 2066-2080.

[23] Abd EL-Baset, A. A., and Ghazal, M. G. (2020). Exponentiated additive Weibull distribution. Reliability Engineering & System Safety. 193, 106663.

[24] Delignette-Muller, M. L., and Dutang, C. (2015). fitdistrplus: An R package for fitting distributions. Journal of Statistical Software. 64(4):1–34.

[25] Delignette-Muller, M. L., and Dutang, C. (2023). Optimal starting points for fitting distributions in R: A guide to the ‘fitdistrplus‘ package. Accessed: 2024-11-17.

[26] Casella, G., and Berger, R. (2002). *Statistical inference*. Duxbury, Pacific Grove, CA, 2nd edition.

[27] Ali, S., Dey, S., Tahir, M. H., and Mansoor, M. (2021). The Poisson Nadarajah-Haghighi distribution: different methods of estimation. Journal of Reliability and Statistical Studies. 415-450.

[28] Alam, F. M. A., and Nassar, M. (2021). On Modeling Concrete Compressive Strength Data Using Laplace Birnbaum-Saunders Distribution Assuming Contaminated Information. Crystals, 11(7). 830.

[29] Alam, F. M. A. (2022). On Comparing Different Methods of Estimation for the Parameters of a Pathological Distribution with Application to Climate Data. Austrian Journal of Statistics. 51(4), 120-147.

[30] Caeiro, F., and Norouzirad, M. (2024). Comparing Estimation Methods for the Power–Pareto Distribution. Econometrics. 12(3), 20.

[31] Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3rd edition.

Disclaimer / Publisher’s Note:

The statements and data in all publications are solely those of the author(s) and contributor(s) and do not necessarily reflect the views of the publisher. The publisher accepts no responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.