Journal of Computational Physics 321 (2016) 369–413

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Scheduled Relaxation Jacobi method: Improvements and applications

J.E. Adsuara [a], I. Cordero-Carrión [b], P. Cerdá-Durán [a], M.A. Aloy [a,*]

[a] Departamento de Astronomía y Astrofísica, Universidad de Valencia, E-46100, Burjassot, Spain
[b] Departamento de Matemática Aplicada, Universidad de Valencia, E-46100, Burjassot, Spain

ARTICLE INFO

Article history:
Received 6 November 2015
Received in revised form 6 April 2016
Accepted 24 May 2016
Available online 27 May 2016

Keywords:
Iterative method
Jacobi method
Finite differences method
Elliptic equations

ABSTRACT

Elliptic partial differential equations (ePDEs) appear in a wide variety of areas of mathematics, physics and engineering. Typically, ePDEs must be solved numerically, which sets an ever growing demand for efficient and highly parallel algorithms to tackle their computational solution. The Scheduled Relaxation Jacobi (SRJ) is a promising class of methods, atypical for combining simplicity and efficiency, that has been recently introduced for solving linear Poisson-like ePDEs. The SRJ methodology relies on computing the appropriate parameters of a multilevel approach with the goal of minimizing the number of iterations needed to cut down the residuals below specified tolerances. The efficiency in the reduction of the residual increases with the number of levels employed in the algorithm. Applying the original methodology to compute the algorithm parameters with more than 5 levels notably hinders obtaining optimal SRJ schemes, as the mixed (non-linear) algebraic-differential system of equations from which they result becomes notably stiff. Here we present a new methodology for obtaining the parameters of SRJ schemes that overcomes the limitations of the original algorithm and provide parameters for SRJ schemes with up to 15 levels and resolutions of up to 2^{15} points per dimension, allowing for acceleration factors larger than several hundreds with respect to the Jacobi method for typical resolutions and, in some high resolution cases, close to 1000. Most of the success in finding SRJ optimal schemes with more than 10 levels is based on an analytic reduction of the complexity of the previously mentioned system of equations. Furthermore, we extend the original algorithm to apply it to certain systems of non-linear ePDEs.

1. Introduction

Partial differential equations (PDEs) are the appropriate mathematical language for modeling many phenomena [1]. In particular, we are interested in elliptic PDEs (ePDEs), that arise when we face the solution of equilibrium problems, in which the time evolution of the system is either neglected or irrelevant. Poisson and Laplace equations are prototype second order ePDEs, with and without source terms respectively.

Though the aforementioned Poisson and Laplace equations possess analytic solutions in a limited number of simple cases, we usually need a numerical solution when more general problems are considered. One of the standard approaches

* Corresponding author.
E-mail addresses: jose.adsuara@uv.es (J.E. Adsuara), isabel.cordero@uv.es (I. Cordero-Carrión), pablo.cerda@uv.es (P. Cerdá-Durán), miguel.a.aloy@uv.es (M.A. Aloy).

http://dx.doi.org/10.1016/j.jcp.2016.05.053

for solving these equations numerically is using finite differences methods. In this approach, both functions and operators are discretized on a numerical mesh, leading to a system of linear algebraic equations, which can be solved with direct or iterative methods. One of the simplest and most studied iterative schemes is the so called Jacobi method [2,3], whose main drawback is its poor convergence rate.

In oder to improve the efficiency of the Jacobi method, many alternatives have been considered. A popular possibility is the use of preconditioners [4–6] applied to linear systems, that make the associated Jacobi and Gauss–Seidel methods converge asymptotically faster than the unpreconditioned ones. Indeed, the method we improve on here, can be adapted as a preconditioner for other methods (e.g., the conjugate gradient method). Very widespread is the use of multigrid methods [e.g., 7] that, in many cases, provide the solution with $\mathcal{O}(N)$ operations, or that can even be employed as preconditioners. Relaxation algorithms [originally introduced in 3], improve the performance of the Jacobi method by considering modifications of the Gauss–Seidel algorithm that include a weight, for instance, successive overrelaxation (SOR) methods [8].

Along this line, [9, YM14 henceforth] has recently presented a significant acceleration (of the order of 100) over the Jacobi algorithm, employing the Scheduled Relaxation Jacobi (SRJ) method. The SRJ method is a generalization of the weighted Jacobi method which adds an overrelaxation factor to the classical Jacobi in a similar fashion to the SOR. This generalization includes a number P of different levels, in each of which, the overrelaxation (or underrelaxation) parameter or weight is tuned to achieve a significant reduction of the number of iterations, thus leading to a faster convergence rate. The optimal set of weights depends on the actual discretization of the problem at hand. Although the method greatly improves the convergence rate with respect to the original Jacobi, the schemes presented by YM14, optimal up to $P = 5$ and resolutions of up to 512 points per spatial dimension, are still not competitive with other methods used currently in the field (e.g., spectral methods [10], or multigrid methods as commented above). The main advantage of the SRJ method over other alternatives to solve numerically ePDEs is its simplicity and the straightforward parallelization, since SRJ methods preserve the insensitivity of the Jacobi method to domain decomposition (in contrast, e.g., to multigrid methods).

Following basically the same procedure as in YM14, [11, ACCA15 henceforth] has obtained optimal SRJ algorithms with up to $P = 10$ levels and multiple numerical resolutions. However, the limitations of the methodology of YM14 to compute optimal parameters for multilevel SRJ schemes prevent to develop algorithms with more than 10 levels. In this paper, we will show a new methodology to evaluate the parameters of optimal SRJ schemes with up to $P = 15$ levels and resolutions of up to 2^{15} points per spatial dimension of the problem, which in some cases may yield accelerations of order 10^3 with respect to the Jacobi method. For an straightforward use of the newly developed SRJ schemes, we provide the readers with a comprehensive set of tables for different SRJ schemes and different resolutions.

We begin the paper giving an overview of the SRJ method (Sect. 2) and describing the original methodology for obtaining optimal schemes together with the improvements on them already made in ACCA15 (Sect. 3). Then, we will present in Sect. 3.4, some analytical work which reduces the number of unknowns to solve for to $\mathcal{O}(P)$ (instead of $\mathcal{O}(P^2)$ as in YM14 and ACCA15). In Sect. 4 we show a comparison of the new method to compute optimal parameters for SRJ schemes with that of YM14.[1] Furthermore, we test the SRJ methods in a case study, namely a Poisson equation with Dirichlet boundary conditions (Sect. 5.1) that has analytic solution, and show that optimal SRJ parameters computed for resolutions close to that of the problem at hand can bring two orders of magnitude smaller number of iterations than the Jacobi method to solve such the problem. We have also assessed the performance of the new SRJ schemes with a large number of sublevels with respect to other standard methods to solve ePDEs (Sect. 5.2). In particular, we compare SRJ schemes with $P = 6$ and $P = 15$ to direct inversion methods and to spectral methods implemented in the `LAPACK` and `LORENE` packages, respectively. We outline the most prominent conclusions of our study and discuss the limitations of the current methodology in Sect. 6.

2. SRJ schemes

In this section we recap the most salient results obtained by YM14 and set the notation for the rest of the paper.

First of all, if we define $\omega_i J$ as a single step in a weighted Jacobi iteration using the weight ω_i $(i = 1, \ldots, P)$, then the SRJ method can be cast as a successive application of elementary relaxation steps of the form

$$\overbrace{\overbrace{\omega_1 J \ldots \omega_1 J}^{q_1}\underbrace{\omega_2 J \ldots \omega_2 J}_{q_2} \ldots \overbrace{\omega_P J \ldots \omega_P J}^{q_P}}^{M}\overbrace{\omega_1 J \ldots \omega_1 J}^{q_1} \ldots ,$$

where the largest weight, ω_1 is applied q_1 times, and each of the remaining and progressively smaller weights ω_i $(i = 1, \ldots, P)$ is applied q_i times, respectively. A single cycle of the scheme ends after M elementary steps, where $M := \sum_{i=1}^{P} q_i$. In order to reach a prescribed tolerance goal, we need to repeat a number times the basic M-cycle of the SRJ method. Both, a vector of weights and a vector with the number of times we use each weight, define each optimal scheme. We emphasize that, from the point of view of the implementation, the only difference with the traditional weighted Jacobi is that, instead of having a fixed weight, SRJ schemes of P-levels require the computation of P weights.

[1] ASCII files containing the optimal parameters for the SRJ algorithms shown in this paper with P values between 6 and 15 can be found at `http://www.uv.es/camap/SRJ.html`.

In order to simplify the notation, we define $\boldsymbol{\omega} := (\omega_1, \ldots, \omega_P)$, with $\omega_i > \omega_{i+1}$ and $\boldsymbol{q} := (q_1, \ldots, q_P)$ which is in one-to-one correspondence with the previous $\boldsymbol{\omega}$. Also, we define $\boldsymbol{\beta} := (\beta_1, \ldots, \beta_P)$, where $\beta_i = q_i/M$ is the fraction of the iteration counts that a given weight ω_i is repeated in an M-cycle, with $\beta_P := 1 - \sum_{i=1}^{P-1} \beta_i$.

The basic idea of the SRJ schemes is finding the optimal values for $\boldsymbol{\omega}$ and $\boldsymbol{\beta}$ that minimize the total number of iterations to reach a prescribed tolerance for a given number of points (i.e., numerical resolution) N.

3. Finding the optimal parameters

Below we explain how to compute the optimal values of $\boldsymbol{\omega}$ and $\boldsymbol{\beta}$, following the prescription of YM14 and rewrite some parts of the YM14 algorithm to make them amenable for extension to a larger number of levels and resolutions.

3.1. Convergence analysis and optimization problem

We perform a convergence analysis of the method in order to obtain a number of restrictions that the parameters of the SRJ scheme must fulfill. As a model problem, we use the Laplace equation with homogeneous Neumann boundary conditions in two spatial dimensions, in Cartesian coordinates and over a domain with unitary size:

$$\begin{cases} \dfrac{\partial^2}{\partial x^2} u(x,y) + \dfrac{\partial^2}{\partial y^2} u(x,y) = 0, & (x,y) \in (0,1) \times (0,1) \\ \left.\dfrac{\partial}{\partial x}\right|_{x=0} u(x,y) = 0, \ \left.\dfrac{\partial}{\partial x}\right|_{x=1} u(x,y) = 0, & y \in (0,1) \\ \left.\dfrac{\partial}{\partial y}\right|_{y=0} u(x,y) = 0, \ \left.\dfrac{\partial}{\partial y}\right|_{y=1} u(x,y) = 0, & x \in (0,1). \end{cases} \tag{1}$$

Let us consider a 2nd-order central-difference discretization of Eq. (1) on a uniform grid consisting of $N_x \times N_y$ zones, and define $N = \max(N_x, N_y)$. Then, we apply the Jacobi method with a relaxation parameter ω, so that the following iterative scheme results:

$$u_{i,j}^{n+1} = (1-\omega) u_{i,j}^n + \frac{\omega}{4}(u_{i,j-1}^n + u_{i,j+1}^n + u_{i-1,j}^n + u_{i+1,j}^n) \tag{2}$$

$$= u_{i,j}^n + \frac{\omega}{4}(u_{i,j-1}^n + u_{i,j+1}^n + u_{i-1,j}^n + u_{i+1,j}^n - 4u_{ij}^n), \tag{3}$$

where n is the index of iteration.

At this point, we perform a von Neumann stability analysis for obtaining the amplification factor,

$$G_\omega(\kappa) = (1 - \omega\kappa), \tag{4}$$

where κ is a function of the wave-numbers in each dimension. For the problem at hand,

$$\kappa(k_x, k_y) = \sin^2\left(\frac{k_x}{2N_x}\right) + \sin^2\left(\frac{k_y}{2N_y}\right). \tag{5}$$

G_ω expresses how much the error can grow up from one iteration to the next one using the relaxation Jacobi method. Thus, if a single relaxation step is performed, we require $|G_\omega| < 1$ to ensure convergence. However, in an SRJ scheme, we perform a series of M-cycles (Sect. 2). Hence, even if on an elementary step of the algorithm one may violate the condition $|G_\omega| < 1$ (which may happen, e.g., if such step is an overrelaxation of the Jacobi method), the condition for convergence shall be obtained for the composition of M elementary amplification factors. As we apply Eq. (3) M-times but with P different weights ω_i, the following composition of amplifications factors is obtained:

$$\overbrace{\overbrace{G_{\omega_1} \ldots G_{\omega_1}}^{q_1} \underbrace{G_{\omega_2} \ldots G_{\omega_2}}_{q_2} \ldots \overbrace{G_{\omega_P} \ldots G_{\omega_P}}^{q_P}}^{M} = \prod_{i=1}^{P} {G_{\omega_i}}^{q_i}. \tag{6}$$

Finally, it is not important how many times we use each of the weights q_i but their relative frequency of use during an M-cycle, which is defined by β_i. Therefore, following YM14, one can define the per-iteration amplification factor function as a geometric mean of the modulus of the cycle amplification factor (Eq. (6)):

$$\Gamma(\kappa) = \prod_{i=1}^{P} |1 - \omega_i \kappa|^{\beta_i}. \tag{7}$$

The previous transformation is very convenient to find deterministic optimal parameters for the SRJ schemes, since it avoids working with a Diophantine equation (Eq. (6)), because $q_i \in \mathbb{N}$, while $\beta_i \in \mathbb{R}$.

From the definition of $\Gamma(\kappa)$, it is evident that larger values of the per-iteration amplification factor yield a larger number of iterations for the algorithm to converge. Thus, the optimal values for the SRJ parameters are obtained by looking for the extrema of $\Gamma(\kappa)$ in $[\kappa_m, \kappa_M]$, which is the interval bounding the allowed values of κ, namely

$$\kappa_m = \sin^2\left(\frac{\pi}{2N}\right), \kappa_M = 2\,, \tag{8}$$

and then, to minimize *globally* these extrema, so that the error per iteration decreases as much as possible. This sets our optimization problem.

We explicitly point out that the value of κ_m depends on the type of boundary conditions of the problem, on the discretization of the elliptic operator and on the dimensionality of the problem. This is not the case for κ_M, which equals 2 independent on the boundary conditions and dimensionality of the problem, though it depends on the discretization of the elliptic operator. For later reference, we write the explicit form of κ_m as a function of the number of dimensions, d, for a Cartesian discretization of the elliptic operator an Neumann boundary conditions:

$$\kappa_m^{(\mathrm{d})} = \frac{2}{d}\sin^2\left(\frac{\pi}{2N^{(\mathrm{d})}}\right). \tag{9}$$

Obviously, we recover Eq. (8) setting $d = 2$. For practical purposes, it is possible to obtain the optimal SRJ parameters for any value of d once we know the optimal parameters in 2D. This is done by computing the effective number of points in 2D, $N_{\mathrm{eff}}^{(2)}$, corresponding to a given problem size $N^{(\mathrm{d})}$ in d-dimensions through the relation:

$$N_{\mathrm{eff}}^{(2)} = \frac{\pi}{2\arcsin\left(\sqrt{\frac{2}{d}}\sin\left(\frac{\pi}{2N^{(\mathrm{d})}}\right)\right)} \simeq N^{(\mathrm{d})}\sqrt{\frac{d}{2}}, \tag{10}$$

where the approximated result holds for large values of $N^{(\mathrm{d})}$.

Finally, as stated above, the values of κ_m change depending on whether Neumann or Dirichlet boundary conditions are considered, and so the optimal parameters change. Fortunately, there is a simple way to obtain the optimal parameters in case of Dirichlet boundary conditions from the 2D optimal parameters computed for Neumann boundary conditions, namely

$$N_{\mathrm{eff}}^{(2)} = \frac{\pi}{2\arcsin\left(\sqrt{\frac{2}{d}\sum_{i=1}^{d}\sin^2\left(\frac{\pi}{2N_{i,\mathrm{Dirichlet}}^{(\mathrm{d})}}\right)}\right)} \simeq \frac{\sqrt{\frac{d}{2}}}{\sqrt{\sum_{i=1}^{d}\frac{1}{(N_{i,\mathrm{Dirichlet}}^{(\mathrm{d})})^2}}}, \tag{11}$$

where the approximated result holds when the number of points in each dimension is sufficiently large.

3.2. The non-linear system

In the optimization problem stated in the previous section we need to compute the location of the extrema $\Gamma(\kappa)$, $\boldsymbol{\kappa}$ hereafter. From the optimization process one must also obtain the rest of the parameters of the SRJ scheme, namely, $\boldsymbol{\omega}$ and $\boldsymbol{\beta}$. Thus, we need to solve a system $\mathcal{S}(\boldsymbol{\omega}, \boldsymbol{\beta}, \boldsymbol{\kappa})$ to determine all these unknowns.

For the evaluation of $\boldsymbol{\kappa}$, we must take into account that the location of the maxima can be either at the edges of the domain, $\kappa_0 := \kappa_m$ and $\kappa_P := \kappa_M$ (set by Eq. (8)), or in other $P-1$ internal values κ_i $(i = 1, \ldots, P-1)$ determined (each of them) by the following condition:

$$\frac{\partial}{\partial\kappa}\log\Gamma(\kappa) = \sum_{i=1}^{P}\frac{\beta_i\omega_i}{1-\kappa\omega_i} = 0. \tag{12}$$

From the solutions of Eq. (12), we obtain the $P-1$ different $\kappa_i = \kappa_i(\boldsymbol{\omega}, \boldsymbol{\beta})$, which allows us to reduce the number of unknowns of the system $\mathcal{S}(\boldsymbol{\omega}, \boldsymbol{\beta}, \boldsymbol{\kappa})$ from $3P-2$ $(\omega_1, \ldots, \omega_P, \beta_1, \ldots, \beta_{P-1}, \kappa_1, \ldots, \kappa_{P-1})$ to $2P-1$. Hence, we need to also obtain $2P-1$ equations to find a unique solution of the system.

For obtaining the set of the first P equations for the system, and following YM14, we equalize all the maxima:

$$\Gamma(\kappa_0) = \Gamma(\kappa_i),\ i = 1, \ldots, P. \tag{13}$$

Furthermore, if we assume that $\boldsymbol{\omega} = \boldsymbol{\omega}(\boldsymbol{\beta})$, and therefore $\boldsymbol{\kappa} = \boldsymbol{\kappa}(\boldsymbol{\beta})$, a second set of $P-1$ equations can be obtained from the minimization of $\Gamma(\kappa_0)$:

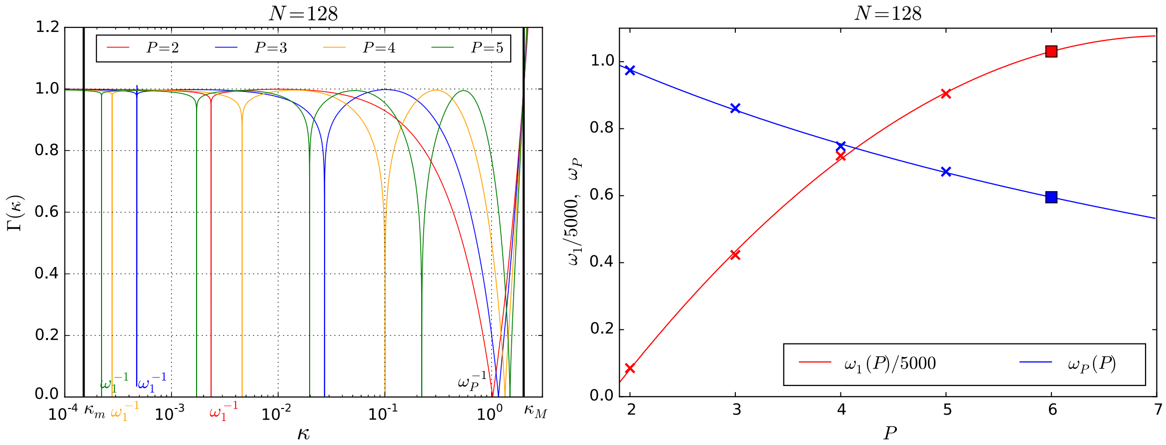

Fig. 1. *Left*: $\Gamma(\kappa)$ functions corresponding to different values of the numbers of levels P for $N=128$. The inverse of the minimal weight, ω_1^{-1}, and the inverse of the maximal one, ω_P^{-1} moves towards κ_m and κ_M, respectively, when P increases. The rest of the weights are distributed roughly logarithmically equally spaced inside (ω_1, ω_P). In the right panel, we show the discrete values of ω_1 and of ω_P for the $P=6$, $N=128$ scheme, together with both fits of their respective values as a function of P to conics as described in Sect. 3.3.

$$\frac{\partial}{\partial \beta_j}\Gamma(\kappa_0)=0,\ j=1,\dots P-1. \tag{14}$$

Thereby, our system is now $S(\boldsymbol{\omega},\boldsymbol{\beta},\frac{\partial\boldsymbol{\omega}}{\partial\boldsymbol{\beta}})$, since the differentiation in Eq. (14) introduces $P(P-1)$ new ancillary variables, namely $\frac{\partial\omega_i}{\partial\beta_j}$; $i=1,\dots,P$, $j=1,\dots,P-1$. The final set of $P(P-1)$ additional equations to account for this extra ancillary variables results from applying the same condition as in Eq. (14) to the remaining values of κ_i, deduced from Eqs. (13) and (14):

$$0=\frac{\partial}{\partial \beta_j}\Gamma(\kappa_i),\ i=1,\dots P,\ j=1\dots P-1. \tag{15}$$

We note that ACCA15 have shown that setting the optimization problem in terms of $\boldsymbol{\omega}$ as reference variables and considering $\boldsymbol{\beta}$ as function of the later ($\boldsymbol{\beta}(\boldsymbol{\omega})$) brings the same numerical solution for the system $\mathcal{S}(\boldsymbol{\omega},\boldsymbol{\beta},\boldsymbol{\kappa})$.

3.3. Basic improvements on the original SRJ algorithm

In order to make the SRJ method competitive with other existing algorithms to solve ePDEs, we must find the optimal parameters of SRJ schemes with a *sufficiently* large number of levels. Furthermore, since the optimal parameters depend on the resolution of the discretization used to solve a given problem, we also need to compute optimal parameters for a range of numerical resolutions larger than in YM14. In this section we summarize some improvements done in ACCA15, which allow us to solve the system from $P=6$ up to $P=10$ and for resolutions up to 2^{15}. Additional improvements based on analytical results will be commented in the following subsection.

Firstly, the stiffness of $S(\boldsymbol{\omega},\boldsymbol{\beta},\frac{\partial\boldsymbol{\omega}}{\partial\boldsymbol{\beta}})$ increases with the number of levels P, which prevented YM14 to compute optimal SRJ schemes for $P>5$. We have been able to reduce the complexity of the numerical solution by manipulating parts of them algebraically. On the one hand, we have hidden the part of the non-linear system involving the $\boldsymbol{\kappa}$ unknowns by solving for them symbolically and using those symbolic placeholders later when solving numerically for $\boldsymbol{\omega}$ and $\boldsymbol{\beta}$. On the other hand, and after the previous manipulation, we have seen in Sect. 3.2 that we need to solve numerically a non-linear system of P^2+P-1 equations with the same number of unknowns. We aim to rewrite this system $S(\boldsymbol{\omega},\boldsymbol{\beta},\frac{\partial\boldsymbol{\omega}}{\partial\boldsymbol{\beta}})$ as $S(\boldsymbol{\omega},\boldsymbol{\beta})$, which requires obtaining $\frac{\partial\boldsymbol{\omega}}{\partial\boldsymbol{\beta}}$ as a function of $\boldsymbol{\omega}$ and $\boldsymbol{\beta}$. We also compute the solutions of this linear subsystem symbolically. These manipulations of the original system of equations permitted ACCA15 to compute the optimal SRJ parameters for a larger number of levels (up to $P=10$) than in YM14.

Secondly, in order to increase the number of points, for the numerical solution of the system, since the accuracy of the results for large values of P critically depends on having sufficiently large precision, we needed Mathematica's ability to perform arbitrary precision arithmetic. We ended up using up to twenty four digits in the representation of the numbers in some cases.

Finally, the non-linear system we have to solve is very sensitive to the initial values that we guess for the unknowns. In ACCA15, we developed a systematic way of setting the initial guesses from the values obtained from lower levels and now we have improved it. As we can see in Fig. 1, when we increase the number of levels P, the values of ω_1^{-1} and ω_P^{-1} move towards κ_m and κ_M, respectively. We can also observe that the inverse of the rest of the weights of a scheme are roughly logarithmically equally spaced between the values ω_1^{-1} and ω_P^{-1}. Hence, we use as approximate location of the initial guesses for the inverse of any weight for an SRJ scheme of P levels the following expressions:

$$\{\omega_1^{-1}, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{1}{2(P-1)}}, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{3}{2(P-1)}}, \ldots, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{P-4}{2(P-1)}}, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{P-1}{2(P-1)}},$$
$$\omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{P+2}{2(P-1)}}, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{P+4}{2(P-1)}}, \ldots, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{2P-3}{2(P-1)}}, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right) = \omega_P^{-1}\} \tag{16}$$

when P is odd, and

$$\{\omega_1^{-1}, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{1}{2(P-1)}}, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{3}{2(P-1)}}, \ldots, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{P-5}{2(P-1)}}, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{P-2}{2(P-1)}},$$
$$\omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{P}{2(P-1)}}, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{P+3}{2(P-1)}}, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{P+5}{2(P-1)}}, \ldots, \omega_1^{-1}\left(\frac{\omega_1}{\omega_P}\right)^{\frac{2P-3}{2(P-1)}}, \omega_P^{-1}\} \tag{17}$$

when P is even.

Looking at Eqs. (16) and (17), the initial (guess) values of the weights for a new SRJ scheme with an additional level ($P' = P + 1$) can be built providing suitable estimates of the smallest and largest weights, which are obtained by fitting to two conics the values of the smallest and largest weights computed for SRJ schemes with $P - 3$ to P levels, and then extrapolating the result. For instance, in Fig. 1 we show on the right panel the values of ω_1 and ω_P as a function of P with red and blue symbols, respectively. If we want to obtain the initial values of ω_1 and ω_P for $P = 6$, we fit the values of ω_1 and, separately, those of ω_P for $P = 2, \ldots, 5$ to a hyperbola $\omega = \frac{a}{bP-c} + d$ or a parabola $\omega = aP^2 + bP + c$ depending on the flatness of the points, an infer the value for our P (in Fig. 1, the fit functions are plot with continuous lines, and the extrapolated values of ω_1 and ω_P for $P = 6$ with squares).

As we shall see in the next section, and improving on the procedure outlined in ACCA15, we do not need to provide initial values of $\boldsymbol{\beta}$, since they can be obtained analytically from the values of $\boldsymbol{\omega}$.

3.4. Advanced analytical improvements

In this section, we prove two important theorems, which tell us how to calculate analytically the ancillary variables $\frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{\beta}}$ and the parameters $\boldsymbol{\beta}$ of the SRJ schemes in terms of the $\boldsymbol{\omega}$ and $\boldsymbol{\kappa}$ variables. Let us start with some technical results we need for the proof of these theorems. Notice that in all products appearing from now on, each index of the product refers only to expressions containing that particular index.

Lemma 1. *Let A and B be two matrices defined as $A := (a_{ij}) = \left(\frac{\kappa_i \beta_j}{1-\kappa_i \omega_j}\right)$ and $B := (b_{ij}) = \left(\frac{1-\omega_j/\omega_P}{1-\kappa_i \omega_j}\right)$, $i, j = 1, \ldots, P-1$, respectively. The inverse matrices, $A^{-1} = \tilde{A} = (\tilde{a}_{ij})$ and $B^{-1} = \tilde{B} = (\tilde{b}_{ij})$, are given by:*

$$\tilde{a}_{ij} = \prod_{\substack{k=1\\k\neq j}}^{P}\prod_{\substack{l=1\\l\neq i}}^{P} \frac{(1-\kappa_j \omega_i)(1-\kappa_k \omega_i)(1-\kappa_j \omega_l)}{\beta_i \kappa_j (\kappa_k - \kappa_j)(\omega_l - \omega_i)}, \tag{18}$$

$$\tilde{b}_{ij} = \frac{\omega_P (1-\kappa_j \omega_i)}{(1-\kappa_j \omega_P)} \prod_{\substack{k=1\\k\neq j}}^{P-1}\prod_{\substack{l=1\\l\neq i}}^{P} \frac{(1-\kappa_k \omega_i)(1-\kappa_j \omega_l)}{(\kappa_k - \kappa_j)(\omega_l - \omega_i)}. \tag{19}$$

Proof. We just need to check that $\sum_{j=1}^{P} \tilde{a}_{ij} a_{jm} = \sum_{j=1}^{P} \tilde{b}_{ij} b_{jm} = \delta_{im}$. For convenience, we define $\kappa_P := \kappa_M$.

We will start checking that

$$\sum_{j=1}^{P} \frac{1}{(-1+\kappa_j \omega_m)} \left(\prod_{\substack{k=1\\k\neq j}}^{P}\prod_{\substack{l=1\\l\neq i}}^{P} \frac{(-1+\kappa_j \omega_l)}{(\kappa_j - \kappa_k)}\right) = \delta_{im} \left(\prod_{k=1}^{P}\prod_{\substack{l=1\\l\neq i}}^{P} \frac{(\omega_i - \omega_l)}{(-1+\kappa_k \omega_i)}\right). \tag{20}$$

We consider first the case where $i \neq m$. In general, taking into account that all the κ_i are strictly different, for a polynomial $F(x)$, with $\deg F(x) < P - 1$, we can do a partial fraction decomposition of the following form:

$$\frac{F(x)}{\prod_{j=1}^{P-1}(x-\kappa_j)} = \sum_{j=1}^{P-1} \frac{F(\kappa_j)}{(x-\kappa_j)} \left(\prod_{\substack{k=1\\k\neq j}}^{P-1} \frac{1}{(\kappa_j-\kappa_k)} \right). \tag{21}$$

Considering $F(x) = \prod_{\substack{l=1\\l\neq i,m}}^{P}(-1+x\omega_l)$ in the above expression, and evaluating at $x=\kappa_P$, we get the desired expression.

We consider now the remaining case $i=m$. For convenience, we define $\kappa_0 = 1/\omega_i$, that satisfies $\kappa_0 \neq \kappa_j$, $j=1,\ldots,P-1$. For a polynomial $F(x)$, with $\deg F(x) < P$, we can do a partial fraction decomposition of the following form:

$$\frac{F(x)}{\prod_{j=0}^{P-1}(x-\kappa_j)} = \sum_{j=0}^{P-1} \frac{F(\kappa_j)}{(x-\kappa_j)} \left(\prod_{\substack{k=0\\k\neq j}}^{P-1} \frac{1}{(\kappa_j-\kappa_k)} \right). \tag{22}$$

Considering $F(x) = \prod_{\substack{l=1\\l\neq i}}^{P}(-1+x\omega_l)$ in the above expression, and evaluating at $x=\kappa_P$, we get the desired expression.

We use this equality to check the expression for the inverse matrix A^{-1} (as well as B^{-1} below):

$$\begin{aligned}
\sum_{j=1}^{P} \tilde{a}_{ij} a_{jm} &= \frac{\beta_m}{\beta_i} \sum_{j=1}^{P} \frac{(1-\kappa_j\omega_i)}{(1-\kappa_j\omega_m)} \prod_{\substack{k=1\\k\neq j}}^{P} \prod_{\substack{l=1\\l\neq i}}^{P} \frac{(-1+\kappa_k\omega_i)(-1+\kappa_j\omega_l)}{(\kappa_j-\kappa_k)(\omega_i-\omega_l)} \\
&= \frac{\beta_m}{\beta_i} \left(\prod_{k=1}^{P} \prod_{\substack{l=1\\l\neq i}}^{P} \frac{(-1+\kappa_k\omega_i)}{(\omega_i-\omega_l)} \right) \sum_{j=1}^{P} \frac{1}{(-1+\kappa_j\omega_m)} \prod_{\substack{k=1\\k\neq j}}^{P} \prod_{\substack{l=1\\l\neq i}}^{P} \frac{(-1+\kappa_j\omega_l)}{(\kappa_j-\kappa_k)} \\
&= \frac{\beta_m}{\beta_i} \left(\prod_{k=1}^{P} \prod_{\substack{l=1\\l\neq i}}^{P} \frac{(-1+\kappa_k\omega_i)}{(\omega_i-\omega_l)} \right) \left(\prod_{k=1}^{P} \prod_{\substack{l=1\\l\neq i}}^{P} \frac{(\omega_i-\omega_l)}{(-1+\kappa_k\omega_i)} \right) \delta_{im} = \delta_{im}.
\end{aligned} \tag{23}$$

Finally, we check the expression for the inverse matrix B^{-1}:

$$\begin{aligned}
&\sum_{j=1}^{P} \tilde{b}_{ij} b_{jm} \\
&= \frac{(\omega_m-\omega_P)}{(\omega_i-\omega_P)} \left(\prod_{k=1}^{P-1} \prod_{\substack{l=1\\l\neq i}}^{P-1} \frac{(-1+\kappa_k\omega_i)}{(\omega_i-\omega_l)} \right) \sum_{j=1}^{P} \frac{-1}{(-1+\kappa_j\omega_m)} \prod_{\substack{k=1\\k\neq j}}^{P-1} \prod_{\substack{l=1\\l\neq i}}^{P-1} \frac{(-1+\kappa_j\omega_l)}{(\kappa_j-\kappa_k)} \\
&= \frac{\omega_m-\omega_P}{\omega_i-\omega_P} \delta_{im} = \delta_{im}. \qquad \square
\end{aligned} \tag{24}$$

Theorem 1.

$$\frac{\partial}{\partial \beta_q}\omega_i = \sum_{j=1}^{P} \prod_{\substack{k=1\\k\neq j}}^{P} \prod_{\substack{l=1\\l\neq i}}^{P} \frac{(1-\kappa_j\omega_i)}{\beta_i\kappa_j} \frac{(1-\kappa_k\omega_i)}{(\kappa_k-\kappa_j)} \frac{(1-\kappa_j\omega_l)}{(\omega_l-\omega_i)} \log\left| \frac{1-\kappa_j\omega_q}{1-\kappa_j\omega_P} \right|. \tag{25}$$

Proof. We have that κ_j with $j=1,\ldots,P-1$ are the roots where the local extrema are located. We will also use the already defined κ_P. Taking into account the Eqs. (14) and (15), for a fixed value of l, $1 \le l \le P-1$, we construct the linear system $A\,\frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{\beta}} = \boldsymbol{f}$, which in components reads:

$$[a_{ij}] \cdot \begin{bmatrix} \frac{\partial}{\partial\beta_l}\omega_1 \\ \frac{\partial}{\partial\beta_l}\omega_2 \\ \vdots \\ \frac{\partial}{\partial\beta_l}\omega_P \end{bmatrix} = \begin{bmatrix} \log|1-\kappa_1\omega_l/1-\kappa_1\omega_P| \\ \log|1-\kappa_2\omega_l/1-\kappa_2\omega_P| \\ \vdots \\ \log|1-\kappa_P\omega_l/1-\kappa_P\omega_P| \end{bmatrix}, \tag{26}$$

where a_{ij} is defined in Lemma 1. We can solve for the ancillary variables $\frac{\partial \omega_i}{\partial \beta_j}$, $i = 1, \ldots, P$, $j = 1, \ldots, P-1$, analytically, just inverting the matrix A:

$$\begin{bmatrix} \frac{\partial}{\partial \beta_l}\omega_1 \\ \frac{\partial}{\partial \beta_l}\omega_2 \\ \vdots \\ \frac{\partial}{\partial \beta_l}\omega_P \end{bmatrix} = [a_{ij}]^{-1} \cdot \begin{bmatrix} \log|1-\kappa_1\omega_l/1-\kappa_1\omega_P| \\ \log|1-\kappa_2\omega_l/1-\kappa_2\omega_P| \\ \vdots \\ \log|1-\kappa_P\omega_l/1-\kappa_P\omega_P| \end{bmatrix} \tag{27}$$

Using Lemma 1 to get the expression of the inverse matrix and doing the corresponding matrix product, we obtain Eq. (25). Notice that we obtain the same results when we consider κ_m instead of κ_P. $\square$

Theorem 2.

$$\beta_i = \prod_{k=1}^{P-1} \prod_{\substack{l=1 \\ l\neq i}}^{P} (1-\kappa_k\omega_i)\frac{\omega_l}{(\omega_l-\omega_i)}. \tag{28}$$

Proof. We consider now the $P-1$ equations resulting from Eq. (12) when κ is replaced by $\kappa_i, i = 1, \ldots, P-1$. We write $\beta_P = \beta_P(\beta_j)$, and obtain:

$$\sum_{j=1}^{P-1} \frac{(1-\omega_j/\omega_P)}{(1-\kappa_i\,\omega_j)}\beta_j = 1,\; i = 1, \ldots, P-1, \tag{29}$$

These equations can be rewritten in matrix form, $B\boldsymbol{\beta} = \mathbf{g}$, which in components reads:

$$[b_{ij}] \cdot \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{P-1} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{30}$$

where b_{ij} is defined in Lemma 1. Using Lemma 1 to get the expression of the inverse matrix and doing the corresponding matrix product, we obtain:

$$\begin{aligned} \beta_i &= \omega_P \sum_{j=1}^{P-1} \frac{(1-\kappa_j\,\omega_i)}{(1-\kappa_j\,\omega_P)} \prod_{\substack{k=1 \\ k\neq j}}^{P-1} \prod_{\substack{l=1 \\ l\neq i}}^{P} \frac{(-1+\kappa_k\,\omega_i)(-1+\kappa_j\,\omega_l)}{(\kappa_j-\kappa_k)(\omega_i-\omega_l)} \\ &= \omega_P \left(\prod_{k=1}^{P-1} \prod_{\substack{l=1 \\ l\neq i}}^{P} \frac{(-1+\kappa_k\,\omega_i)}{(\omega_i-\omega_l)} \right) \sum_{j=1}^{P-1} \prod_{\substack{k=1 \\ k\neq j}}^{P-1} \prod_{\substack{l=1 \\ l\neq i}}^{P-1} \frac{(-1+\kappa_j\,\omega_l)}{(\kappa_j-\kappa_k)}. \end{aligned} \tag{31}$$

In particular, using the simplification proven at the beginning of Lemma 1 for $i = m$, changing P by $P-1$, and setting $\omega_i = 0$ just formally, we also get that:

$$\sum_{j=1}^{P-1} \left(\prod_{\substack{k=1 \\ k\neq j}}^{P-1} \prod_{\substack{l=1 \\ l\neq i}}^{P-1} \frac{(-1+\kappa_j\,\omega_l)}{(\kappa_j-\kappa_k)} \right) = \prod_{\substack{l=1 \\ l\neq i}}^{P-1} \omega_l. \tag{32}$$

Using this expression,

$$\beta_i = \prod_{k=1}^{P-1} \prod_{\substack{l=1 \\ l\neq i}}^{P} \frac{(-1+\kappa_k\,\omega_i)\,\omega_l}{(\omega_i-\omega_l)}. \qquad \square \tag{33}$$

3.5. Advantages of the rewritten system

In the most general case, as we have commented above, we need to solve a system $S(\boldsymbol{\omega}, \boldsymbol{\beta}, \boldsymbol{\kappa}, \frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{\beta}})$. With the theorems recently introduced in Sect. 3.4, which basically express $\frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{\beta}}$ as $\frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{\beta}}(\boldsymbol{\omega}, \boldsymbol{\kappa})$ and $\boldsymbol{\beta}$ also as $\boldsymbol{\beta}(\boldsymbol{\omega}, \boldsymbol{\kappa})$, and substituting them into Eq. (13) and Eq. (14), the non-linear, algebraic-differential system reduces to a purely algebraic system of the form $S(\boldsymbol{\omega}, \boldsymbol{\kappa})$.

We have developed a code that implements everything commented in Sect. 3.3 and that automatically constructs the system and solves it for any value of P. This program is written in Mathematica (a pseudocode of which can be found in Algorithm 1) and combines both symbolic with numerical calculations.

```
Data: Γ, N
Result: ω, β
(κm, κM) ← computeKappas(N)                                   // Eqs. (8), (9)
for j = 1, ..., P − 1 do
    dΓβ[j] ← ∂/∂β_j Γ
end
for i = 1, ..., P do
    equMax[i] ← Γ(κ0) == Γ(κi)                                 // Eq. (13)
end
for i = 0, ..., P do
    for j = 1, ..., P − 1 do
        equDer[i] ← dΓβ[j](κi) == 0                           // Eqs. (14), (15)
    end
end
for i = 1, ..., P do
    for q = 1, ..., P − 1 do
        dωdβ[i][q] ← symbolicAncillaryVariables()             // Eq. (25)
    end
end
for i = 1, ..., P − 1 do
    β[i] ← symbolicBetas()                                     // Eq. (28)
end
(ωg, κg) ← computeInitialGuesses()                            // Eqs. (16), (17), (35)
(ω, κ) ← findScheme(equMax[], equDer[], dωdβ[], β[], ωg, κg)
β ← computeBetas(ω, κ, β[])                                   // Eq. (28)
return ω, β
```

Algorithm 1. Pseudocode for computing optimal schemes.

We obtain two major benefits from the new system of equations to be solved:

1. For any given number of levels P, we reduce by orders of magnitude the computing time in the generation of the symbolic system that we need to solve. Furthermore, the analytic reduction of the system presented in the previous section allows us to reduce its dimensionality, by reducing the number of equations and unknowns from $P^2 + P - 1$ to P. The consequence of such reduction is that the number of operations needed to solve the system as well as the number of initial guesses we must provide to begin its iterative solution decreases drastically. Thereby, the numerical stiffness of the original system decreases, since it is fundamentally brought by the difficulty in finding good initial guesses for all the variables involved in its solution. In ACCA15, we built $S(\boldsymbol{\omega}, \boldsymbol{\beta})$ and needed a lot of time for the symbolic calculations involved in obtaining, on the one hand, $\frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{\beta}}(\boldsymbol{\omega}, \boldsymbol{\beta})$, solving the corresponding linear subsystem (Eq. (30)) and, on the other hand, $\boldsymbol{\kappa}(\boldsymbol{\omega}, \boldsymbol{\beta})$. With the new methodology, employing Eqs. (25) and (28), we exchange the role of $\boldsymbol{\beta}$ and $\boldsymbol{\kappa}$, since we have analytic formulae to express the ancillary variables and $\boldsymbol{\beta}$ as functions of $\boldsymbol{\omega}$ and $\boldsymbol{\kappa}$, respectively. With the consequent reduction in the calculation time, the computational time to solve the remaining equations becomes negligible.
2. The solution of the non-linear system to obtain the optimal SRJ parameters needs a suitable methodology to compute their initial values (Sect. 3.3). In ACCA15, we had to provide initial values for $\boldsymbol{\omega}$ and $\boldsymbol{\beta}$. With the newly derived theorems of the previous section, we only need to provide initial guesses for $\boldsymbol{\omega}$, since employing Eq. (28) $\boldsymbol{\beta} = \boldsymbol{\beta}(\boldsymbol{\omega}, \boldsymbol{\kappa})$, and $\boldsymbol{\kappa}$ satisfies

$$\kappa_i \in \left(\frac{1}{\omega_i}, \frac{1}{\omega_{i+1}}\right). \tag{34}$$

From the plots of Γ (see, e.g., Fig. 1), we can see that each maximum κ_i is roughly placed at:

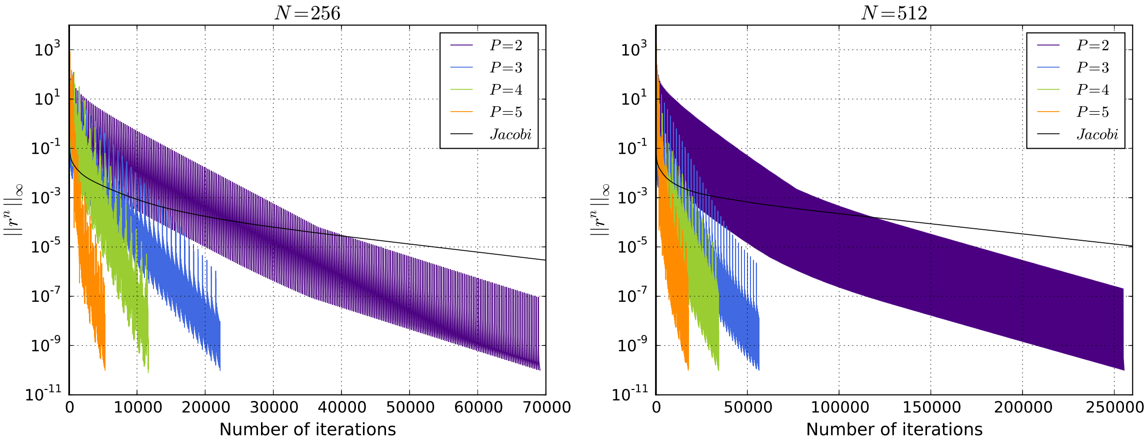

Fig. 2. Comparison of the evolution of the difference between consecutive approximate solutions ($||r^n||_\infty$) of Eq. (3) of SRJ schemes from $P = 2$ to $P = 5$ for a grid with $N = 256$ and with $N = 512$ zones per dimension. We also include the evolution of the residual for the Jacobi method as a reference.

$$\kappa_i \approx \frac{1}{\omega_i} + \frac{\frac{1}{\omega_{i+1}} - \frac{1}{\omega_i}}{3}, \tag{35}$$

which are the values that we will use as initial guesses.

With these two improvements, we have reduced by four orders of magnitude the total time for finding the parameters of an optimal scheme. For example, for $P = 10$, it was necessary to spend a calculation time in Mathematica of the order of one week with the methodology employed by ACCA15. In contrast, with the improvements reported here, we can accomplish the same task in tens of seconds. While, in practice, in ACCA15 we were limited (due to the computing time) to SRJ schemes with $P \leq 10$, now we can tackle larger number of levels.

4. Results

In this section we first calibrate our new method comparing the parameters of our optimal SRJ schemes with those of YM14 for $P \leq 5$ (Sect. 4.1). Later (Sect. 4.2), we present new optimal schemes computed employing the new methodology sketched in the previous section, up to $P = 15$.

4.1. Calibration of the method

To calibrate the new methodology, we have recomputed the optimal parameters for SRJ schemes with $P \leq 5$ and found that our results are the same as those obtained by YM14, when we use the same number of points per dimension N on the same model problem (Eq. (1)).

Following the ideas of YM14, the performance of any SRJ scheme with respect to the Jacobi method can be quantified estimating the *convergence performance index*, ρ,

$$\rho := \sum_{i=1}^{P} \omega_i \beta_i \tag{36}$$

which we have calculated for each SRJ method we have computed, and checked that it approaches its theoretical value when we solve numerically (Eq. (1)). We point out that the value of ρ depends on the dimensionality of the problem since the value of κ_m does (see, Eq. (9)).

YM14 showed that the optimal parameters computed for coarser grids can be used for finer ones. Nevertheless, minimizing the gaps between different values of N is important because the acceleration of the convergence with respect to the Jacobi method may not be the largest possible unless we compute the optimal SRJ parameters corresponding to a given problem size. Thus, we have completed the tables presented by YM14 minimizing the possible gaps between resolutions. Furthermore, we have computed the optimal SRJ parameters for a number of intermediate values of N in Table A.3, where we also show the value of ρ. In order to verify the correct behavior of the schemes computed, we monitor in Fig. 2 the evolution of the difference between two consecutive approximations of the solution for the model problem specified in Eq. (1),

$$r_{ij}^n = u_{ij}^n - u_{ij}^{n-1}, \tag{37}$$

using element-wise norms and operations, that in the bidimensional case would be, for example,

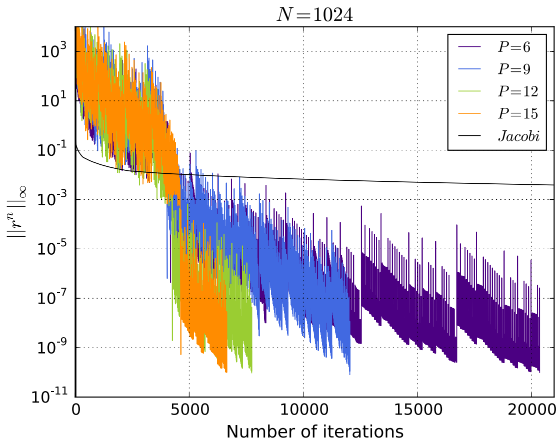

Fig. 3. Comparison of the residual evolution for the optimal SRJ schemes with $P = 6, 9, 12$ and 15 and $N = 1024$ points per dimension. For reference, we also include the evolution of the residual for the Jacobi method (black line).

$$||r^n||_\infty = \max_{ij} r^n_{ij}, \tag{38}$$

as a function of the number of iterations n for SRJ schemes having all the values of P given by YM14. In the same figure, we also include the residual evolution for the Jacobi method (black line). As expected, the number of iterations to reach the prescribed tolerance decreases[2] as P increases. For all the schemes shown in Fig. 2, where the number of points is set to $N = 256$, we have obtained the expected theoretical value of ρ.

4.2. New SRJ optimal schemes

After verifying that we recover the optimal parameters computed in YM14, we have improved on their results computing the optimal values of SRJ schemes with $P > 5$. In Appendix A we provide the Tables A.4 to A.19, corresponding to the optimal parameters for SRJ schemes with $P = 6, \dots, 13$ and various resolutions for the Laplace problem (Eq. (1)). In Tables A.21 to A.23, we show the optimal solution parameters for $P = 14$ and $P = 15$. We encountered that finding optimal parameters at low resolution is increasingly more difficult as the number of levels increases. Indeed, as we can see, for $P = 15$ the minimum value of N we have been able to compute is 64. The reason for the inability of the proposed method to find optimal parameters for low N and large P is that larger values of P imply that the results are extremely sensitive to tiny changes in the smaller wave numbers (i.e., to the values of κ_i close to κ_m), and small numerical errors prevent a full evaluation of the solution of the non-linear system $\mathcal{S}$, unless the (guessed) initial values are very close to the optimal ones.

We remark that thanks to the improvements done (Sect. 3.3) and specially with the analytic solution of a part of the unknowns of the system (Sect. 3.4), not only the optimal solution is achievable, but also it is reachable with a moderate computational cost: employing Mathematica on a standard workstation, the computational time of the optimal parameters ranges from tenths of a second for the $P = 6$ scheme to tens of seconds for $P = 15$.

In Fig. 3 we show the evolution of the residual for some of the new optimal SRJ schemes solving the model problem used throughout the paper. These new schemes show a progressively larger efficiency as P grows. A good proxy for the performance of the method is the convergence performance index, which grows with the number of levels. We achieve a reduction in the time of computation to solve the problem because of the reduction in the number of iterations to reach convergence. This reduction is roughly proportional to $P \log_{10}(P + 1)$. However, the rate of reduction of the error is non-monotonic. For large values of P (namely, $P \geq 12$), a direct inspection of Fig. 3 shows a faster decline of the residual once any given SRJ method reduces its residual below the one corresponding to the Jacobi method (in this case, this happens after about 4.500 iterations).

4.3. Obtention of the integer parameters $\boldsymbol{q}$

There is a step in the practical implementation of SRJ methods that may impact on the performance of the resulting algorithm, measured by the number of iterations needed to reduce the residual below a prescribed tolerance. Once the solution has been found and we know the *real* values of $\boldsymbol{\omega}$ and $\boldsymbol{\beta}$, one must obtain the *integer* values of $\boldsymbol{q}$. The conversion to integer begins by defining $\bar{\boldsymbol{\beta}} := \frac{\boldsymbol{\beta}}{\beta_1}$, so that $\beta_1 = q_1 = 1$. For the conversion to integer of the rest of the $\bar{\beta}_i$ ($i = 2, \dots, P$), we have tested several possibilities, including the floor $\boldsymbol{q} = \lfloor \bar{\boldsymbol{\beta}} \rfloor$, rounding $\bar{\boldsymbol{\beta}}$ to the nearest integer, taking the ceiling function

[2] If not explicitly mentioned, in all the cases considered in this paper the absolute tolerance is fixed so that $||r^n||_\infty < 10^{-10}$.

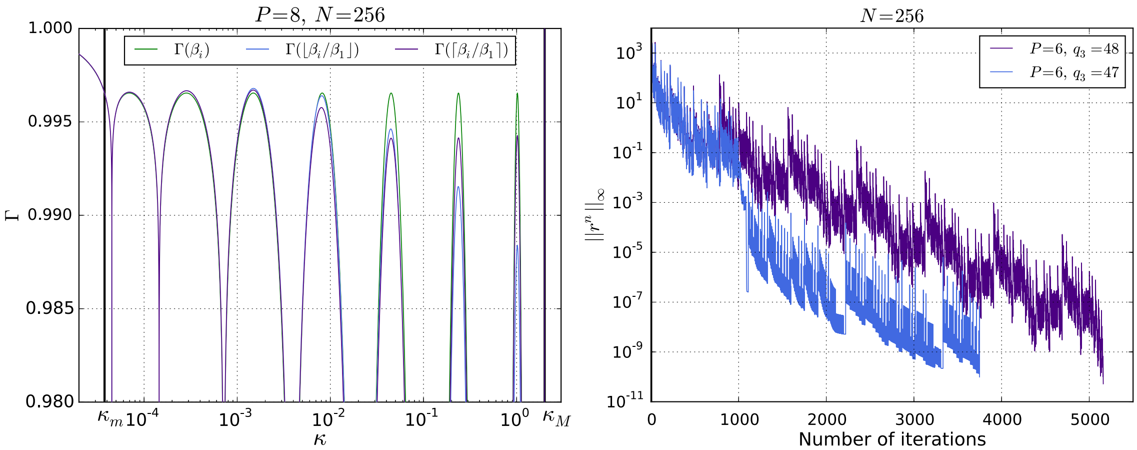

Fig. 4. Comparison of the evolution of the residual of Eq. (3) of two SRJ schemes having $P = 6$ and $N = 256$ zones per dimension. Top: mean amplification factor per M-cycle. Bottom: evolution of the residual as a function of the number of iterations. The different variants of the $P = 6$ SRJ method are displayed with different color lines showing the dependence of the performance of the method on the conversion of the real values of the optimal solution for β_i to the integer values $q_i = \lfloor \beta_i/\beta_1 \rfloor$ (blue) and $q_i = \lceil \beta_i/\beta_1 \rceil$ (purple). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$\boldsymbol{q} = \lceil \bar{\boldsymbol{\beta}} \rceil$, or combinations of the former alternatives, since it is possible to apply different recipes for every β_i/β_1. Each of these alternatives may yield a different number of iterations to reach convergence (see below). After computing the integer values of $\boldsymbol{q}$, a key point to account for is that the $\Gamma(\kappa)$ function must remain below 1, since otherwise our method diverges. In Fig. 4 (upper panel) we observe that the amplification factor per M-cycle may change by more than 10%, for values of κ close to κ_M, depending on the method adopted to convert $\boldsymbol{\beta}$ to integer.

While the number of levels is small, the differences among the distinct conversions from real to integer do not change much either the number of iterations or the convergence rate of the resulting scheme. However, when P increases, there can be non-negligible changes in the total number of iterations to reduce the residual of our model equation below a prescribed tolerance. In the lower panel of the Fig. 4 we show the evolution of the residual as a function of the iteration number for two different choices of the integer conversion of $\boldsymbol{\beta}$ into $\boldsymbol{q}$ in the case $P = 6$ (the optimal parameters of which can be found on Table A.5). We note that there is a difference of more than 1200 iterations ($\sim 25\%$) between the distinct integer conversions. Unfortunately, changing the number of levels, the same recipes for converting reals to integers yield efficiencies of the methods that do not display a clear trend. Fortunately, increasing the number of levels by one unit results in a reduction of the number of iterations to reach convergence which is larger than that resulting from any manipulation of the integer values of q_i in an SRJ scheme with a given P. Hence, in the following, the results we will provide are obtained by taking simply $q_i = \lfloor \beta_i/\beta_1 \rfloor$.

5. Numerical examples

5.1. Poisson equation with Dirichlet boundary conditions

So far we have considered only the application of SRJ schemes to the solution of the Laplace equation with homogeneous Neumann boundary conditions (Eq. (1)). This was also the case in YM14. In this section we consider a case study consisting on solving a Poisson equation in two dimensions endowed with Dirichlet boundary conditions. The exact problem setting reads

$$\begin{aligned} \frac{\partial^2}{\partial x^2}u(x,y) + \frac{\partial^2}{\partial y^2}u(x,y) &= -e^{xy}(x^2+y^2), \ \ (x,y) \in (0,1)\times(0,1), \\ u(0,y) = -1, \ \ u(1,y) &= -e^{y}, \ \ y \in [0,1], \\ u(x,0) = -1, \ \ u(x,1) &= -e^{x}, \ \ x \in [0,1], \end{aligned} \tag{39}$$

which has analytic solution:

$$u(x,y) = -e^{xy}. \tag{40}$$

This kind of problem will help us to assess whether the change in the boundary conditions affects the efficiency of an SRJ scheme. Imposing Dirichlet boundary conditions is typically less challenging than dealing with Neumann ones, since Dirichlet boundary conditions change the value of κ_m so that (YM14)

$$\kappa_{m,\text{Dirichlet}} = \sin^2\left(\frac{\pi}{2N_x}\right) + \sin^2\left(\frac{\pi}{2N_y}\right), \tag{41}$$

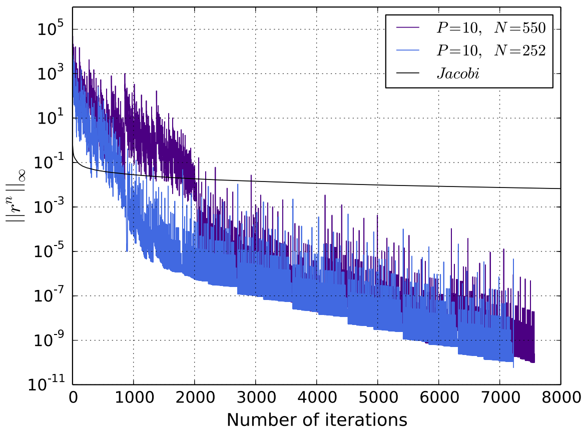

Fig. 5. Evolution of the residual as a function of the iteration number for different SRJ schemes. With violet line we show the case in which the parameters of an SRJ scheme with $P = 10$, $N = 550$ and Neumann boundary conditions are chosen to compute the solution of the problem stated in Eq. (40), on a grid of $N_x \times N_y = 585 \times 280$ and *Dirichlet* boundary conditions. With blue line we show the case in which the parameters of an SRJ scheme with $P = 10$, $N = 252$ and Neumann boundary conditions are used. The latter case corresponds to the closest value of N to the effective resolution $N^{(2)}_{\text{eff}} = 252.56$ that shall be used when Dirichlet boundary conditions (instead of Neumann ones) are used. For comparison, we also display the evolution of the residual for the Jacobi method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to be compared with Eq. (8). Hence, the optimal SRJ values obtained for a given N and Neumann boundary conditions do not exactly coincide with those optimal in problems involving Dirichlet boundary conditions, hence we must follow the recipe provided in Eq. (11). Furthermore, since in Eq. (40) we are considering a Poisson equation, we can test whether the presence of source terms modifies the performance of SRJ methods.

For the case studied we choose a discretization consisting on $N_x \times N_y = 585 \times 280$ uniform numerical zones. Although here we do not have Neumann boundary conditions, we will use the optimal values of the SRJ scheme for this case in order to show that even though this choice is not optimal, still it substantially speeds up the solution of the problem with respect to Jacobi. In this case, we have that $N = 585$. If we apply the SRJ scheme with $P = 10$, we must look for the $\boldsymbol{\omega}$ and the $\boldsymbol{\beta}$ parameters in Table A.12. In this case, the table does not provide an entry for $N = 585$, but as YM14 point out, we can chose as (non-optimal) parameters for the SRJ scheme those corresponding to a smaller resolution.[3] In our case, the closest resolution that matches this criterion in Table A.12 is that corresponding to the row with $N = 550$.

We use the simplest way to obtain the q_i from the β_i ensuring convergence, namely $q_i = \left\lfloor \frac{\beta_i}{\beta_1} \right\rfloor$ with $1 \le i \le P$, resulting in

$$\boldsymbol{q} = \{1, 1, 3, 9, 21, 49, 116, 268, 587, 1014\}, \; M = 2069. \tag{42}$$

This means that we will use $\omega_1 = 106105$ and $\omega_2 = 40577.2$ once per M-cycle, $\omega_3 = 10230.6$ three times per M-cycle, etc. In practice, it is necessary to distribute the largest over-relaxation steps over the whole M-cycle to prevent the occurrence of overflows. YM14 provide a Matlab script that generates a schedule for the distribution of ω_i on the M-cycle that guarantees the absence of overflows. We find that an even distribution of the over-relaxations over the entire M-cycle is sufficient in order to avoid overflows.

In Fig. 5 we plot the evolution of the residual as a function of the number of iterations for a SRJ scheme with $P = 10$, $N = 550$ (instead of $N = 585$), as well as the residual evolution employing the Jacobi method for the solution of Eq. (40). This example shows that even picking an SRJ scheme whose parameters are non-optimal for the problem size at hand ($N = 585$ in this case), for the presence of source terms and for the kind of boundary conditions specified (Dirichlet for the problem at hand), we can largely speed up the convergence with respect to the Jacobi method. Theoretically, for the optimal $P = 10$, $N = 550$ SRJ method, an acceleration of the order of $\rho = 125.85$ with respect to the Jacobi method is expected, something confirmed with our numerical results (Fig. 5).

Finally, we briefly illustrate how to find the optimal scheme for Dirichlet boundary conditions (which are, indeed, the ones with which the problem at hand is set). According to Eq. (11), $N^{(2)}_{\text{eff}} = 252.56$. Therefore, we compute the optimal parameters for the SRJ scheme with $N = 252$, obtaining:

$$\boldsymbol{\omega} = \{22912.5, 10310.7, 3128.79, 836.313, 216.061, 55.4614,$$
$$14.3482, 3.85729, 1.19337, 0.556912\} \tag{43}$$
$$\mathbf{q} = \{1, 1, 3, 6, 13, 29, 62, 131, 256, 401\}. \tag{44}$$

[3] It is, however, possible to compute the optimal values for $N = 585$ employing our algorithm.

In Fig. 5 we can see that the results are slightly better than for the (non-optimal) set of parameters with $P = 10$, $N = 550$ and Neumann boundary conditions, since in the case with $N = 252$, the accuracy goal is reached with $\sim 5\%$ less iterations. However, this small deviation (motivated by the choice of non-optimal parameters) can be overcome in, at least two ways when selecting a non-optimal scheme for the problem at hand. First, by noting that the difference in the number of iterations can be reduced by employing different methods for the conversion from real to integer of the values of $\mathbf{q}$ (see Sect. 4.3). Second, by increasing the number of levels of the SRJ scheme. This is another reason to add to the relevance of the work presented here, since increasing P tends to yield schemes that converge in less iterations (basically at no extra computational cost).

5.2. Poisson equation in spherical coordinates

The Poisson equation appears, among others, in problems involving gravity, either Newtonian or some approximations to General Relativity, and electrostatics. In numerical simulations, e.g., in Astrophysics and Cosmology, the computation of the gravitational potential is usually coupled to a hyperbolic set of equations describing the dynamics of the fluid, e.g., the Euler equations. In those cases the Poisson equation is solved on each time step (or every several time steps) of the evolution of the hyperbolic part. It is thus crucial to test the efficiency of the SRJ compared with other methods currently used by the scientific community. In simulations of stellar interiors, spherical coordinates is a popular choice of coordinates, so we adopt it for our test. To mimic typical astrophysical scenarios we have chosen a test in which the source has compact support and boundary conditions are applied at radial infinity.

The Poisson equation in spherical coordinates (r, θ, φ) reads

$$\frac{\partial^2 u}{\partial r^2} + \frac{2}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} + \frac{\cot\theta}{r^2}\frac{\partial u}{\partial \theta} + \frac{1}{r^2 \sin\theta}\frac{\partial^2 u}{\partial \varphi^2} = s, \tag{45}$$

being u and s functions of (r, θ, φ). For our test we choose the source term to be the series

$$s(r,\theta,\varphi) = \begin{cases} -\displaystyle\sum_{n=0}^{n_{\max}} \sum_{m=-m_{\max}}^{m_{\max}} a_{2n}\, k_{2n}^2\, j_{2n}(k_{2n} r)\, Y_{2n}^{m,c}(\theta,\varphi), & \text{for } r \le 1 \\ 0, & \text{for } r > 1, \end{cases} \tag{46}$$

being j_l the spherical Bessel functions of the first kind and $Y_l^{m,c}$ the real part of the spherical harmonics. Only even parity terms, $l = 2n$ are considered. k_l is the first root of the spherical Bessel function of order l, such that $s(1, \theta, \varphi) = 0$. We chose $a_l = 1/2^l$, such that the series is convergent. We impose homogeneous Neumann boundary conditions at $r = 0$, $\theta = 0$ and $\theta = \pi$, and periodic boundary conditions in the φ direction. If we impose homogeneous Dirichlet condition at radial infinity ($r \to \infty$), the solution of this elliptic problem is

$$u(r,\theta,\varphi) = \begin{cases} \displaystyle\sum_{n=0}^{n_{\max}} \sum_{m=-m_{\max}}^{m_{\max}} \left(a_{2n}\, j_{2n}(k_{2n} r) + b_{2n} r^{2n} \right) Y_{2n}^{m,c}(\theta,\varphi), & \text{for } r \le 1 \\ \displaystyle\sum_{n=0}^{n_{\max}} \sum_{m=-m_{\max}}^{m_{\max}} \frac{c_{2n}}{r^{2n+1}}\, Y_{2n}^{m,c}(\theta,\varphi), & \text{for } r > 1, \end{cases} \tag{47}$$

where the coefficients c_l and b_l can be computed imposing continuity of u and its first derivatives at $r = 1$, resulting in

$$b_l = c_l = -\frac{a_l}{2l+1} \partial_r j_l(k_l r)|_{r=1} = -\frac{a_l}{2l+1} \left[l\, j_l(k_l) - k_l\, j_{l+1}(k_l) \right] \tag{48}$$

Since our interest is to assess the performance of the SRJ method under the conditions which are found on real applications, we solve this equation numerically in the domain $r \in [0, 1]$, $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi]$, i.e. only in the region where the sources are non-zero, and apply Dirichlet boundary conditions at $r = 1$, using the analytical solution given by Eq. (47).

We emphasize that this problem set up includes boundary conditions of mixed type (Neumann and Dirichlet) and, hence, none of the schemes whose optimal parameters have been tabulated in this paper is strictly optimal. However, as we shall see, even in such conditions, the new schemes presented in this paper with $P = 15$ will be competitive with other alternatives in the literature.

We set up three versions of the test with different dimensionality. In the 3D test, we choose $n_{\max} = \infty$ and $m_{\max} = 2n$ and solve the equation in the domain $r \in [0, 1]$, $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi]$, discretized in an equidistant grid of size $N \times N \times N$ points. In the 2D case we consider axisymmetry, i.e., no φ dependence in u and s. We choose $n_{\max} = \infty$ and $m_{\max} = 0$ and solve in the domain $r \in [0, 1]$ and $\theta \in [0, \pi]$, discretized in a grid with $N \times N$ points. In the 1D case we consider spherical symmetry, i.e. no θ or φ dependence in u and s. We choose $n_{\max} = 0$, $m_{\max} = 0$ and solve in the domain $r \in [0, 1]$ with N points. Since the series in Eqs. (46) and (47) are convergent, we compute them numerically by adding terms until the last significant digit does not change. We use a second order finite difference discretization for Eq. (45)

and one ghost cell in each direction to impose boundary conditions. For convenience, we multiply Eq. (45) by r^2 in the discretized version.

As an example, we present explicitly the discretization of the 1D problem. The 2D and 3D discretizations are analogous to what is described here. We use a staggered grid with ghost cells, $r_i = (i - 1/2)\Delta r$ with $i = 0, ..., N + 1$, where $\Delta r = 1/N$. Points $i = 0$ and $i = N + 1$ are ghost cells used only for the purpose of imposing boundary conditions. Using second order centered derivatives and imposing spherical symmetry ($\partial_\theta = \partial_\varphi = 0$) Eq. (45), multiplied by r^2, can be discretized as

$$r_i^2 \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta r^2} + 2r_i \frac{u_{i+1} - u_{i-1}}{2\Delta r} = r_i^2 s_i, \quad (i = 1, ..., N) \tag{49}$$

where sub-index i indicates a function evaluated at r_i. By imposing boundary conditions it is possible to set the values of u_0 and u_{N+1}. The resulting linear system of N equations with N unknowns, u_i, $i = 1, ..., N$, can be written in matrix form

$$\sum_{j=1}^{N} \mathcal{A}_{ij} u_j = r_i^2 s_i, \quad (i = 1, ..., N), \tag{50}$$

being $\mathcal{A}_{ij}$ the elements of the coefficient matrix, which in the 1D case is a $N \times N$ tridiagonal matrix

$$\begin{aligned} \mathcal{A}_{ii-1} &= (r_i - 2\Delta r) \frac{r_i}{\Delta r^2} \\ \mathcal{A}_{ii} &= \frac{-2r_i^2}{\Delta r^2} \\ \mathcal{A}_{ii+1} &= (r_i + 2\Delta r) \frac{r_i}{\Delta r^2} \\ \mathcal{A}_{ij} &= 0, \quad \text{otherwise.} \end{aligned} \tag{51}$$

Note that this matrix is diagonally dominant by rows and columns except for the first two rows and the first column. If the r^2 factor were not present, the matrix would not be diagonally dominant by columns and the convergence of the iterative methods could not be guaranteed. Once the boundary conditions are applied, the coefficient matrix is effectively modified. Whether the resulting effective matrix is diagonally dominant or not depends crucially on how the boundary conditions are applied.

We impose Dirichlet boundary conditions at the outer boundary by setting $u_{N+1} = u_{\text{analytic}}(r_{N+1})$, being the analytic solution that given by Eq. (47). In this case the equation at $i = N$ results

$$\mathcal{A}_{NN-1} u_{N-1} + \mathcal{A}_{NN} u_N = r_N^2 s_N - \mathcal{A}_{NN+1} u_{\text{analytic}}(r_{N+1}), \tag{52}$$

being $\mathcal{A}_{NN+1}$ an extension of the coefficient matrix used for practical purposes. At the inner boundary we impose homogeneous Neumann conditions, i.e. $\partial_r u|_{r_0} = 0$. Standard numerical techniques to deal with this kind of boundary conditions (see e.g. [12], pp. 76–77), expand the Laplacian operator around $r = 0$, resulting in

$$\partial_{rr} u + 2/r\, \partial_r u = 3\partial_{rr} u + \mathcal{O}(\Delta r^2). \tag{53}$$

The second order discretization of this equation at the first radial cell yields

$$\Delta u|_{r_1} \approx 3 \frac{u_2 - 2u_1 + u_0}{\Delta r^2}, \tag{54}$$

and a second order discretization of the boundary condition results in

$$\frac{u_1 - u_0}{\Delta r} = 0, \tag{55}$$

i.e. $u_1 = u_0$, which results in

$$\Delta u|_{r_1} \approx 3 \frac{u_2 - u_1}{\Delta r^2}. \tag{56}$$

Using this prescription for the discretization of the Laplacian operator at the first radial cell, ensures that the matrix is diagonally dominant by rows and columns.

An additional simplification can be made by noticing that, as long as the source s is regular at $r = 0$, Eq. (56) implies that $u_1 = u_2 + \mathcal{O}(\Delta r^2)$. Therefore, for a second order method, one can assume $u_0 = u_1 = u_2$, as an alternative prescription to solve the problem of the diagonal dominance. Since this prescription not only fixes the value of the ghost cell, u_0, but also the value u_1, this condition reduces the dimensionality of the linear system by 1. The resulting effective coefficient matrix $\hat{\mathcal{A}}_{ij}$ is a tridiagonal matrix of size $(N - 1) \times (N - 1)$, with indices $i, j = 2, ...N$. The elements read

$$\begin{aligned}
\hat{\mathcal{A}}_{22} &= -(r_2 + 2\Delta r)\,\frac{r_2}{\Delta r^2} = -\frac{21}{4} \\
\hat{\mathcal{A}}_{23} &= (r_2 + 2\Delta r)\,\frac{r_2}{\Delta r^2} = +\frac{21}{4} \\
\hat{\mathcal{A}}_{ij} &= \mathcal{A}_{ij}, \quad \text{otherwise.}
\end{aligned} \tag{57}$$

The new matrix is diagonally dominant by rows and columns, which guarantees the convergence of Jacobi-based iterative methods. In practice, the effective coefficient matrix, $\hat{\mathcal{A}}_{ij}$, is not used in the iterative methods directly. Instead we use a coefficient matrix $\mathcal{A}_{ij}$ extended to the ghost cells and we set at each iteration the values at the ghost cells (u_0 and u_{N+1}) and at u_1 according to prescription given above. This procedure is equivalent to using the effective coefficient matrix, but it eases the implementation of the algorithm.

We have tested both prescriptions to make the matrix diagonally dominant, namely that of Eq. (56) and the additional simplification in Eq. (57). We have found that, although both prescriptions result in convergent methods, the number of iterations needed to converge is systematically $\sim 30\%$ smaller with the second prescription, for all the iterative methods tested. Therefore, we provide here only results for the second (faster) prescription.

We perform series of calculations for different values of the number of points N. For each calculation we use the SRJ coefficients computed in the previous sections matching the corresponding value of N. For each series we perform calculations using coefficients computed with different values of P. The convergence criterion is that the L_∞-norm of the residual, defined as

$$||r||_\infty = \max_{i=1,\dots,N} \left| \sum_{j=1}^{N} \mathcal{A}_{ij} u_j - r_i^2 s_i \right|, \tag{58}$$

is smaller than the tolerance. Note that this criterion differs from previous sections, Eqs. (37) and (38), in a factor ω_i. The tolerance goal is set to $10^{-5}/N^2$, which depends on the number of points. Since we use a second order discretization, this scaling in the tolerance ensures that the difference between the numerical and the analytical solution is dominated by finite differencing errors and at the same time avoids unnecessary iterations in the low resolution calculations. This prescription for the tolerance mimics the tolerance choice that is used under realistic conditions and renders a fairer comparison in the computational cost between different resolutions.

For comparison we also perform calculations using other iterative methods: Jacobi, Gauss–Seidel and SOR (weight equal to 1.9). For each case involving iterative methods we perform two calculations: the *ab initio calculation* in which the solution is initialized to zero, and the *realistic calculation* in which the solution is initialized to $u_{\text{analytic}}(1 + ran(-0.5, 0.5)/N)$, being $ran(-0.5, 0.5)$ a random number in the interval $[-0.5, 0.5]$. The *realistic calculation* tries to mimic the conditions encountered in many numerical simulations in which an elliptic equation (or a system of PDEs) is solved coupled with evolutionary (hyperbolic) PDEs,[4] which are typically solved using explicit methods, whose time step is limited by the Courant–Friedrichs–Lewy (CFL) condition. This means that the change in the source of the Poisson equation between subsequent calculations is $\mathcal{O}(\Delta x)$; therefore, if the solution of the previous time step is used for the iteration in the elliptic solver, this should differ only $\mathcal{O}(\Delta x) \sim 1/N$, from the solution.

In addition to iterative methods, we perform the calculations using a direct inversion method and spectral methods. In the *direct inversion method*, we compute the LU factorization of the matrix associated with the coefficients of the discretized version of the equation by performing Gaussian elimination using partial pivoting with row interchanges. We use the implementation in the `dgbtrf` routine of the `LAPACK` library [13], which allows for efficient storage of the matrix coefficient in bands. Once the LU decomposition is known, we solve the system of linear equations using the `dgbtrs` routine. Since this method is non-iterative, its computational cost does not depend on the initial value. However, this approach has advantages when used repeatedly, coupled to evolution equations (e.g., for a fluid). Most of the computational cost of this method is due to the LU decomposition, but once it has been performed, solving the linear system for different values of the sources is computationally less intensive. Therefore, we consider the computational cost of the whole process, LU decomposition and solution of the system, in the *ab initio calculations*, while only the solution of the linear system, assuming the LU decomposition is given, in the *realistic calculations*.

For the spectral solver we use the `LORENE` library [14]. To provide results that are comparable to all other numerical methods used in the present work we use the following procedure: first we evaluate the source, s, at the *finite differences grid* used in all other numerical methods; then, the source is interpolated to the collocation points in the *spectral grid*, which do not coincide with the *finite differences grid*; next the solution is computed by means of the `LORENE` library; finally the function is evaluated at the cells of the *finite differences grid*. The details of the procedure are described in [15]. The accuracy of the numerical method is dominated by the second order finite differences discretization error associated with the *finite differences grid*, provided sufficient number of collocation points are used in the *spectral grid*. We have tested that it is sufficient to use $N/2$ collocation points per dimension to fulfill this requirement. When using the spectral solver, there is no difference in the computational cost in *ab initio* or *realistic calculations*.

[4] For instance, this is the case of the fluid equations.

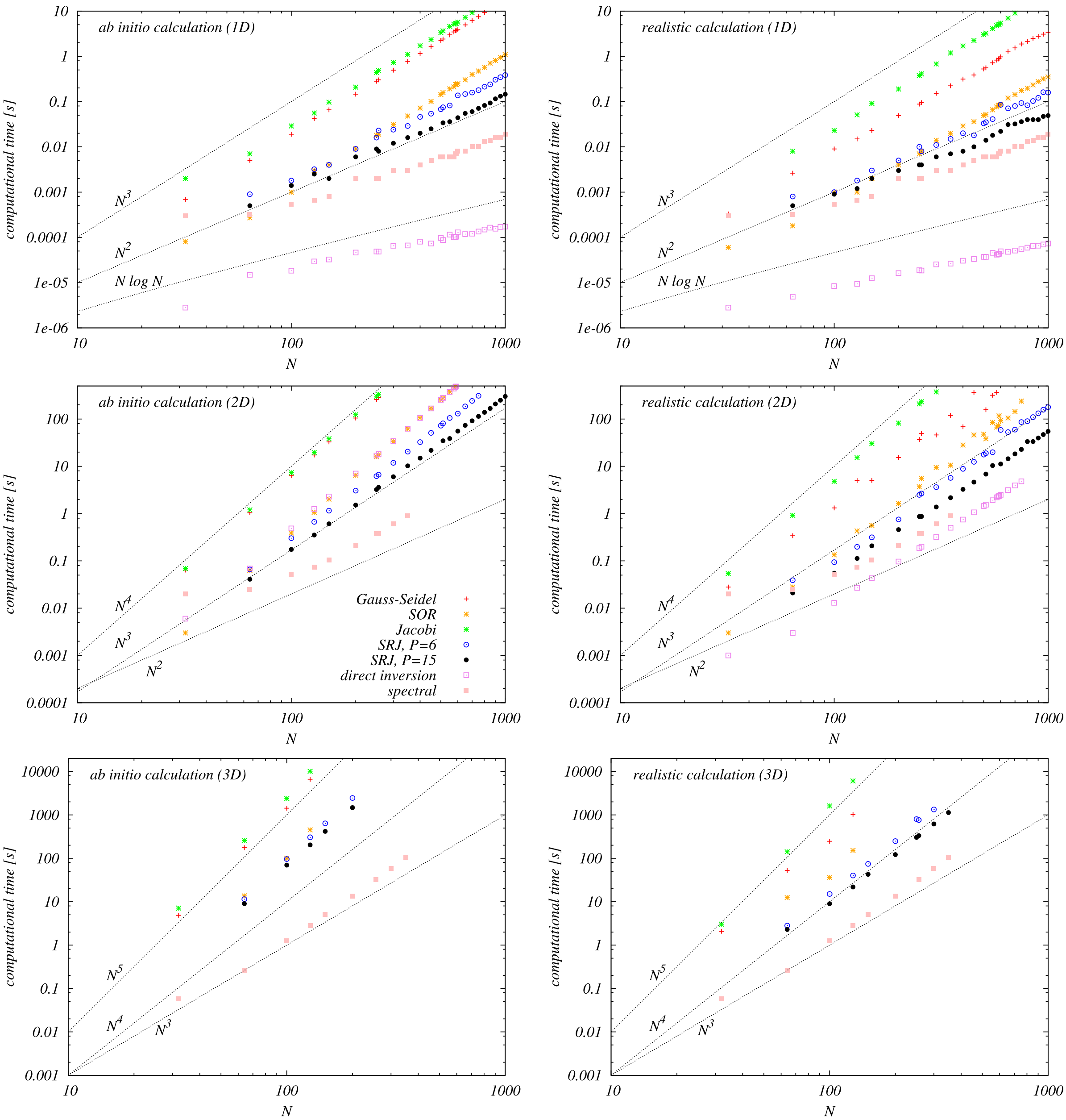

Fig. 6. Computational cost of the solution of the Poisson equation in spherical coordinates, depending on the size of the problem N, using different numerical methods, including SRJ for the minimum ($P = 6$) and maximum ($P = 15$) set of coefficients computed in this work. Upper, middle and lower panels show the 1D, 2D and 3D test respectively. Right and left column panels show *ab initio* and *realistic calculations* respectively. (For a color version of this figure, please check the web version of this article.)

We have performed the calculations using a 3.4 GHz Intel core i7 and 16 GB of memory. We have measured the computational time for each method timing exclusively the part of the code involved in the computation and not the allocation and initialization of variables. Fig. 6 shows the dependence of the computational time for 1D, 2D and 3D tests in the *ab initio calculations* and the *realistic calculations* setups. As expected, for any dimensionality, the SRJ methods render a significant speed up with respect to other iterative methods, due to the smaller number of iterations needed. Only SOR method has comparable computational time for low resolutions ($N < 100$). The computational time for SRJ methods scales approximately as N^{d+1}, being d the dimensionality of the test, i.e. the number of iterations is proportional to N. In comparison, the computational cost of other iterative methods (Jacobi, Gauss–Seidel, SOR), scale approximately as N^{d+2}, i.e. the number of iterations needed scales as N^2. This factor N improvement of SRJ with respect other iterative methods ensures that the method will always be less costly for sufficient high resolution. Compared to non-iterative methods the results depend on the dimensionality of the test.

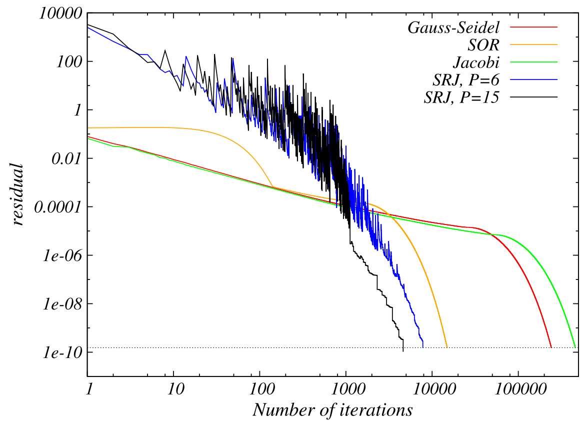

Fig. 7. Example of the evolution of the residual during the iterative procedure, for several of the iterative methods used in this work. Dashed-black horizontal line corresponds to the tolerance goal. (For a color version of this figure, please check the web version of this article.)

For the 1D test, both spectral and the direct inversion method are significantly faster than SRJ. The computational cost of both methods are close to $N \log N$. Therefore, we conclude that SRJ methods are not competitive for 1D problems, even when *realistic* conditions are considered.

In 2D, the computational cost of the direct inversion method for *ab initio calculations* increases significantly, scaling as N^4, because the associated matrix is not tridiagonal anymore, as in the 1D case, but is a banded matrix of band size $2N+1$. Hence, the direct inversion method is more costly than SRJ for resolutions $N > 100$. However, in the *realistic calculation*, in which the LU decomposition is not performed, the direct inversion method is still the fastest, with a computational time scaling as N^3 (the same as SRJ) but with lower computational cost. Due to limitations of the LORENE library, we were not able to perform multidimensional computations using spectral methods for $N > 350$. Compared to $P = 15$, spectral methods are about a factor 2 faster than SRJ in *realistic calculations* and become comparable for $N < 100$. It seems fair to say that spectral methods perform better for *ab initio calculations*, since in this case, SRJ methods (see $P = 6$ and $P = 15$ in Fig. 6) scale as N^3. Therefore, we conclude that for 2D calculations SRJ is a competitive method, when compared with spectral methods. Although the direct inversion method is the fastest in the range of values of N selected for our tests, we expect that this advantage will disappear when going to larger number of points; the memory needed for the direct inversion method scales as N^3 (due to the explicit use of the banded structure of the matrix) in comparison with N^2 of all other methods (iterative and spectral). This strongly limits the size of the problem to be solved without using parallelization.

In 3D all computations are significantly more costly, so we limit our tests to what is achievable within ~ 1 hour of computation time. For the SRJ methods tested this is $N \leq 200$. For $N > 100$ the computational cost of spectral methods is a factor ≈ 10 lower than a SRJ method with $P = 15$, in the *realistic calculation*. Using the SRJ parameters for $P = 15$ and an effective number of points per dimension as given by Eq. (10), the SRJ method becomes $\sim 20\%$ faster, so that it is "only" ≈ 8 slower than the spectral method. The conclusion is that spectral methods still seem to have advantages over SRJ methods, for the 3D test presented. However, both spectral and SRJ methods scale approximately as N^4 in 3D. Due to the large amount of memory needed for the direct inversion method, which scales as N^5, we did not present any such calculation for the 3D case. In practice, this limitation makes the direct inversion method unfeasible for computations in a single CPU. The performance of all these methods and a comparison between them in a parallel architecture is beyond the scope of this work.

Fig. 7 shows the evolution of the tolerance, computed using Eq. (58), for a 2D simulation with $N = 256$, and the initial conditions of the *realistic calculation*. Note, that the residual at the first iteration is significantly larger than in Jacobi, due to the use of a weight with large value in this iteration ($\omega_1 = 19127$ and 25234 for $P = 6$ and 15, respectively). This is compensated by a faster average convergence in cycles of M iterations as expected from SRJ methods, being $M = 781$ and 1154 for $P = 6$ and 15, respectively.

Finally we have estimated numerically the value of ρ for different SRJ weights, to be compared with the theoretical predictions. For this purpose we compute the ratio of number of iterations needed with Jacobi and a given SRJ method, using the same tolerance and resolution, N. Left panels of Fig. 8 show the dependence of ρ on P for several values of N, computed using the set of *ab initio calculations*. Regardless of the dimensionality, in all calculations the numerical values of ρ are close to the theoretical predictions (solid lines). In the 1D test problems there is a tendency of the theoretical values to overestimate the numerically computed value. This trend is exacerbated for large values of N (namely, $N > 512$). In 3D the situation is reversed, and the theoretical value of ρ falls below the numerical one. To explain these differences, we shall consider that the optimal weights depend on the dimensionality of the problem, since κ_m does also depend on dimensionality (see Eq. (9)). As the optimization of the weights has been performed for the 2D case, it is not surprising to find such discrepancies when using the same weights and the same value of ρ in a problem with different dimensionality. Indeed, we have repeated some of the 3D and 1D test problems employing the optimal SRJ parameters corresponding to

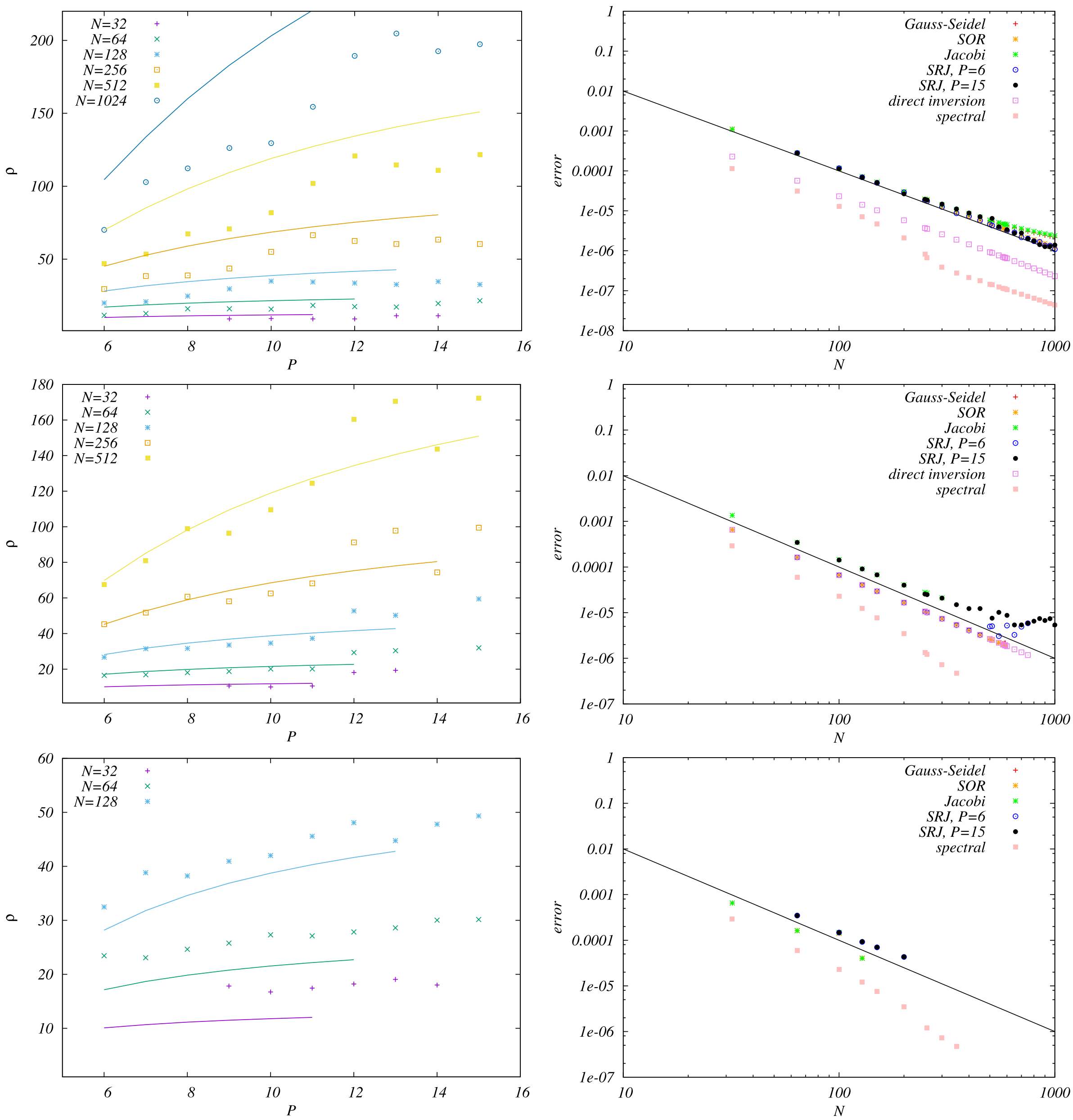

Fig. 8. Detailed analysis of the solution of the Poisson equation for 1D, 2D and 3D calculations (upper, middle and lower panels respectively). Left panels show the dependence of the numerically estimated value of ρ on P, for several values of N ranging from 32 to 1024. Solid lines of the same color represent the theoretical estimate of ρ, for each case. Right panels show the error in the solution as a function of N, computed as the L_∞-norm of the difference between the numerical and the analytical solution. The solid black line represents $1/N^2$, which is an estimation of the expected finite difference error. (For a color version of this figure, please check the web version of this article.)

the effective number of points set according to Eq. (10), and found that (i) the SRJ scheme runs is $\sim 20\%$ less iterations and, (ii) for this effective number of points, the theoretical convergence performance index, computed with the dimensionality corrections mentioned below Eq. (36), becomes an upper bound for the numerical values of ρ. Adding to this arguments, we also note that the discretization of the Laplacian operator in spherical coordinates may also change slightly the optimal weights. Finally, another factor that explains the discrepancies is that the boundary conditions of this problem are mixed (as commented above), and the optimal weights are computed for purely Neumann boundary conditions.

We find that for 2D applications increasing P from 6 to 15 yields an increase in ρ of $\sim 2-3$ for the largest resolutions considered here (Fig. 8; left panels). In 3D, the increment of ρ is expected to be similar (we do not show examples with larger values of N due to the long duration of the tests and the fact that such numerical grids are typically computed with parallel algorithms, which we do not discuss here). Hence, it is worthwhile employing SRJ schemes with a larger number of levels than those originally proposed in YM14, specially considering that there is no extra complexity in the algorithm implementation for any $P \geq 2$, once the weights for large values of P are known.

The right panels of Fig. 8 show the error in the solution as a function of N, computed as the L_∞-norm of the difference between the numerical and the analytical solution. In all cases the error is dominated by the finite difference error associated to the discretization of the elliptic operator, which, for a second order method, is expected to be $\sim \mathcal{O}((\Delta x)^2) \sim \mathcal{O}(1/N^2)$. This is a symptom that our prescription for the tolerance is yielding converged numerical solutions, in iterative methods. It also shows that the number of spectral grid points used is sufficient for such calculations.

We have also tried different discretizations of the equation and the boundary conditions, although not as systematically as the presented case. In general, using discretizations which lead to non-diagonally dominant coefficient matrices, increases the number of iterations to converge or, in some cases, they do not converge at all. The Jacobi method is the most sensitive to this, while all other iterative methods (Gauss–Seidel, SOR, SRJ) seem less affected by this issue. As an example, if just $u_0 = u_1$ is used for the inner boundary condition (consistent with Eq. (55)), the Jacobi method needs about 5 times more iterations in 1D, while all other iterative methods remain almost unaltered (only SOR shows differences for $N \leq 64$). This is an indication that the new method is not only faster than well-known iterative methods but can also be more robust than some of them.

5.3. Grad–Shafranov equation in spherical coordinates

The Grad–Shafranov (GS) equation [16,17] describes equilibrium solutions in ideal magnetohydrodynamics for a two dimensional plasma. It is of interest in studying the plasma in magnetic confinement fusion (e.g. Tokamaks), the solar corona and neutron star magnetospheres, among others.

In spherical coordinates (r, θ, φ) the magnetic field of an axisymmetric ($\partial_\varphi = 0$) plasma configuration can be expressed as

$$\boldsymbol{B}(r,\theta) = \boldsymbol{\nabla} \times \boldsymbol{A} = \frac{1}{r\sin\theta}\boldsymbol{\nabla}\Psi(r,\theta) \times \hat{\boldsymbol{e}}_{\boldsymbol{\varphi}} + \frac{F(r,\theta)}{r\sin\theta}\hat{\boldsymbol{e}}_{\boldsymbol{\varphi}}, \tag{59}$$

where $\boldsymbol{A}$ is the vector potential and $\hat{\boldsymbol{e}}_{\boldsymbol{\varphi}}$ is the unit vector in the φ direction. The flux function, $\Psi \equiv r\sin\theta A_\varphi$, is constant along magnetic field lines and is a measure of the poloidal magnetic field strength. The toroidal function, $F \equiv B_\varphi r \sin\theta$, is a measure of the toroidal field strength. Using Ampere's law, $\boldsymbol{J} = \nabla \times \boldsymbol{B}$, being $\boldsymbol{J}$ the electric current, the flux function can be linked to the toroidal current as

$$\Delta^*\Psi \equiv \partial_{rr}\Psi + \frac{1}{r^2}\partial_{\theta\theta}\Psi - \frac{\cot\theta}{r^2}\partial_\theta\Psi = -J_\varphi r\sin\theta, \tag{60}$$

where Δ^* is the GS elliptic operator. For simplicity we consider here the case in which the inertia of the fluid can be neglected (magnetically dominated). In this case, if we impose force balance, $\boldsymbol{J} \times \boldsymbol{B} = 0$, the toroidal function depends on the flux function, $F(\Psi)$. As a result Eq. (60) leads to the GS equation

$$\Delta^*\Psi = -F(\Psi)F'(\Psi). \tag{61}$$

Not neglecting the inertia of the fluid leads to additional pressure terms, which are not considered here. A popular choice for the toroidal function is $F(\Psi) = C\Psi$, being C a constant. In this case the GS equation results in

$$\Delta^*\Psi + C^2\Psi = 0, \tag{62}$$

which is a suitable elliptic problem to be solved with SRJ methods. Equation (62) resembles the Helmholtz differential equation in that it contains a Laplacian-like operator and a linear term in Ψ. Therefore, this test will show the ability of SRJ methods to handle more complicated elliptic operators. In addition we use this test to demonstrate the ability of iterative methods to handle boundary conditions imposed at arbitrarily shaped boundaries.

We compute the solution of Eq. (62) for two sets of boundary conditions, in the numerical domain $r \in [1, 10]$ and $\theta \in [0, \pi]$. In all cases we impose homogeneous Dirichlet conditions at $\theta = 0$ and $\theta = \pi$. In *test A* we impose Dirichlet boundary conditions at $r = 1$ and $r = 10$ with $\Psi = \sin^2\theta / r$. In the case $C = 0$, the solution for this test is a dipolar field. As the value of C is increased the solution results in a twisted dipole.

In *test B* we solve the GS equation in part of the domain, the region defined by

$$\begin{gathered} r < \left(4.5\sin^2\theta + 2.5\sin^2(2\theta)\right)(1 - 0.4\cos(3\theta) + 0.3\cos(5\theta) + 0.05\sin(25\theta)) \\ \& \\ (r\sin\theta - 4)^2 + (r\cos\theta - 1.6)^2 < 1, \end{gathered} \tag{63}$$

inside the aforementioned numerical domain. The boundary of this region intersects the sphere $r = 1$ at $\theta_1 = 0.3037$ and $\theta_2 = 2.8903$. At $r = 1$ we impose Dirichlet boundary conditions with $\Psi = \sin((\theta - \theta_1)/(\theta_2 - \theta_1)\pi)^2$, and homogeneous

Table 1
Number of iterations and computational time used by the SRJ method with $N = 300$ and $P = 14$ to solve the GS equation, depending on the value of C.

test A			*test B*		
C	iterations	computational time [s]	C	iterations	computational time [s]
0	5350	8.55	0	3450	1.78
0.01	5350	8.55	0.01	3450	1.78
0.1	5350	8.58	0.1	3450	1.78
0.2	4660	7.48	0.5	3600	1.86
0.3	7750	12.43	1.0	3550	1.84
0.31	9830	15.73	1.3	3410	1.78
0.32	13540	21.80	1.4	3660	2.41
0.33	21390	34.38	1.45	4980	2.57
0.34	49080	79.22	1.47	11620	6.03

Table 2
Number of iterations used by different methods to solve the GS equation (*test A*), for several values of C. In parenthesis the ratio of the number of iterations using Jacobi to the number of iterations, i.e. an estimation of the value of ρ.

C	0	0.2	0.32
SRJ, $P = 14$	5350 (46)	4660 (67)	13540 (85)
SRJ, $P = 6$	9820 (25)	9180 (34)	22380 (52)
SOR	41050 (6.0)	57650 (5.4)	228100 (5.1)
Gauss–Seidel	132940 (1.9)	187560 (1.7)	788290 (1.5)
Jacobi	246260 (1)	310490 (1)	1153610 (1)

Dirichlet conditions at the remaining boundaries. Imposing boundary conditions in arbitrarily shaped boundaries is straightforward when using iterative methods such as SRJ; we set $\Psi = 0$ everywhere outside the region (63) and apply the SRJ iteration only inside this region using a mask.

In both tests we use a second order discretization of the GS equation and a numerical resolution of 300×300 equispaced grid points covering the numerical domain. We solve the equations using the SRJ method with weights corresponding to $N = 300$ and $P = 14$. In both tests we initialize Ψ to zero in the whole domain. Table 1 shows the number of iterations and computational time to obtain a numerical solution with residual below 10^{-12}, depending on the value of C used. We have used the same convergence criterion as in Section 5.2. Table 2 compares SRJ method with $P = 14$ with the other iterative methods presented in the previous subsection, for the *test A*. In all cases the $P = 14$ is the fastest method, by a factor comparable to those obtained in Sect. 5.2. Again, the fact that using $P = 6$ results in about twice as many iterations for solving the problem as when employing $P = 14$ shows the advantage of employing SRJ schemes with the largest available number of levels.

The upper panels of Fig. 9 show the results for *test A*, for three different values of C. For the case $C = 0$, the analytical solution is $\Psi = \sin^2\theta/r$. In this case the maximum difference between the analytical and the numerical result, in absolute value, is 8.5×10^{-5}, which is consistent with the second order discretization ($9/N^2 = 10^{-4}$). For $C = 0.1$ a toroidal component appears, but the flux function, Ψ, remains essentially the same. For higher values of C there is a tendency of the magnetic field lines to become more inflated, to support the increased magnetic tension due to the high magnetic field. In this regime the number of iterations needed in the SRJ method increases. We were not able to obtain solutions for values larger than $C = 0.34$. This is not a problem of the numerical method itself, since other methods (Jacobi, Gauss–Seidel, SOR) show similar behavior. The value $C \approx 0.35$ corresponds to an eigenvalue of the GS operator. For this case the matrix associated to the discretization of the GS equation is singular and hence it cannot be inverted. This is causing the convergence problems near this point.

Lower panels of Fig. 9 show the results for *test B*, for three different values of C. This case behaves qualitatively similar to *test A* but with more complicated geometry. The case $C = 0$ shows no toroidal field, which appears as C is increased. For $C = 1.0$ the flux function is still similar to that of the untwisted case, albeit slightly deformed. For $C = 1.47$, the maximum value that we were able to achieve, the topology of the field has changed, showing a region of close magnetic field lines in the upper right part of the domain. As in *test A*, the difficulty to achieve convergence for larger values of C is related to the presence of an eigenvalue of the GS operator. Note that the solution is everywhere smooth, and magnetic field lines (black lines) are tangent to the domain boundary (blue curve) as expected (except for $r = 1$ where non-zero Dirichlet boundary conditions are applied).

In general the SRJ method shows reasonable rates of convergence and computational time to solve the problem with high accuracy, despite of the complicated boundary conditions. This renders a method which can be used in real applications of the GS equation with a good trade of excellent performance and ease of implementation.

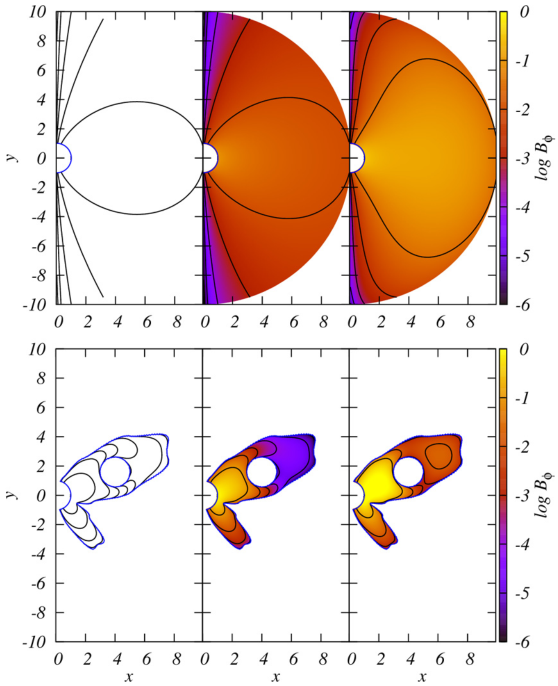

Fig. 9. Numerical solution of the GS equation for *test A* (upper panels) and *test B* (lower panels) for different values of the constant C. From left to right $C = 0$, 0.1 and 0.34 (upper panels) and $C = 0$, 1.0 and 1.47 (lower panels). Isocontours of $\log\Psi$ (solid black lines), which coincide with magnetic field lines, are plotted in increments of 1. Colors show $\log B_\varphi$. For convenience we plot the (x, y) plane, being $x \equiv r\sin\theta$ and $y \equiv r\cos\theta$. Blue line in lower panels show the boundary of the region in which the GS equation is solved. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6. Conclusions and future work

Building upon the results of YM14, we have devised a new method for obtaining the optimal parameters for SRJ schemes applied to the numerical solution of ePDEs.

We have shown that the new multilevel SRJ schemes keep improving the convergence performance index of the scheme, which means that increasing the value of P we obtain ever larger acceleration factors with respect to the Jacobi method. In the present paper we report acceleration factors of a few hundreds and, in some cases, more than 1000 with respect to the Jacobi method if a sufficiently large number of points per dimension (namely, $N > 16000$) and number of levels are considered. For multidimensional applications increasing P from 5 (original maximum number of levels in YM14) to 15 yields a decrease in the computational cost by factors $\sim 2-3$ for the largest resolutions considered here. Since even larger resolutions result in correspondingly larger gains, we note that the benefit of employing SRJ algorithms with $P = 15$ will be really advantageous in three-dimensional supercomputing applications. In such cases it is worthwhile employing SRJ schemes with a larger number of levels than those originally proposed in YM14, specially considering that there is no extra complexity in the algorithm implementation for any $P \geq 2$ once the weights for large values of P are known. Thus, in this paper we have provided a comprehensive set of tables with all the necessary optimal coefficients for a large number of different numbers of points per dimension.

Mainly due to the fact that we have derived analytic solutions for part of the unknowns, our new method reduces the stiffness of the non-linear system of equations from which optimal parameters are computed, allowing us to obtain new SRJ methods for up to $P = 15$ and arbitrarily large number of points per dimension N.

From this number of levels, new problems arise, which hinder the computation of optimal coefficients at relatively low number of discretization points. These problems are related to the fact that for large values of P the solution to the problem are very sensitive to tiny changes in the smaller wave numbers, and small numerical errors prevent the successful evaluation of the solution of even the simplified system of non-linear equations resulting from the algebraic simplifications we have shown here. In order to tackle this problem, we are working in two new improvements: an alternative equivalent new system, and alternative methods for the solution of the optimization problem (including genetic algorithms).

Currently, we have reached acceleration factors that have made that the SRJ methods become competitive (depending on the dimensionality of the problem and its size) with, e.g., spectral methods for the solution of some ePDEs. In particular, we have made the comparison in the case of an astrophysical problem that we are interested in for whose solution we were using spectral methods. We find that for 1D Poisson-like problems, the fastest method of solution is the direct inversion method implemented in `LAPACK`. This happens because the LU decomposition of the matrix solver, where most of the computational work is done, needs to be performed once, and the it can be stored for the rest of the evolution. In 2D, the best performing method depends on whether our initial guess is close to the actual solution or far off. In realistic applications, where ePDEs are coupled to systems of hyperbolic PDEs, the solution from a previous time iteration does not change significantly over the course of a single timestep. In such conditions, the LAPACK libraries are the best performing. However, spectral methods are advantageous if, in 2D, the initial values are far from the actual solution of the problem. We further note that in realistic coupled systems, and for a relatively large number of points per dimension ($N > 500$), the SRJ methods are competitive with spectral ones. In 3D applications, we find that the total computational cost of SRJ methods scales in 3D as N^4, i.e., as in the case of spectral methods. Considering that (i) applying direct inversion methods to 3D problems is unfeasible because of memory restrictions, and that (ii) SRJ methods can be parallelized straightforwardly (much more easily than, e.g., spectral or multigrid methods), we foresee that they are a competitive alternative for the solution of elliptic problems in supercomputing applications and in 3D. We are studying improvements in the method from this point of view. Finally, we outline that the easy implementation of complex boundary conditions in SRJ methods is also an advantage with respect to other existing alternatives.

Acknowledgements

We thank P. Mimica for bringing the paper YM14 to our attention. We acknowledge the support from the European Research Council (Starting Independent Researcher Grant CAMAP-259276), and the partial support of grants AYA2013-40979-P, AYA2015-66899-C2-1-P, PROMETEO-II-2014-069 and SAF2013-49284-EXP. We thankfully acknowledge the computer resources, technical expertise and assistance provided by the Servei de Informàtica of the University of Valencia.

Appendix A. Compendium of parameters of optimal SRJ schemes (Tables A.3–A.23)

Table A.3
Parameters $\boldsymbol{w}$, $\boldsymbol{\beta}$ and the estimation of the convergence performance index $\rho = \sum_{i=1}^{P} \omega_i \beta_i$ of the $P = 2$, $P = 3$, $P = 4$ and $P = 5$ schemes for a number of values of N and the model problem specified in Eq. (1).

N		$P = 2$	$P = 3$	$P = 4$
	$\boldsymbol{\omega}$	{321.074, 0.968096}	{1420.73, 30.0648, 0.845599}	{2308.12, 162.259, 8.50839, 0.732499}
100	$\boldsymbol{\beta}$	{0.00993673, 0.990063}	{0.00502828, 0.0729552, 0.922017}	{0.00430412, 0.0245487, 0.158309, 0.812838}
	ρ	4.15	10.12	15.86
	$\boldsymbol{\omega}$	{509.976, 0.977667}	{2724.66, 41.8246, 0.870558}	{4707.62, 259.325, 10.9382, 0.756243}
150	$\boldsymbol{\beta}$	{0.0064850, 0.99352}	{0.00304593, 0.0574955, 0.939459}	{0.00268244, 0.0182254, 0.138417, 0.840676}
	ρ	4.28	11.52	19.50
	$\boldsymbol{\omega}$	{704.099, 0.982735}	{4295., 52.6521, 0.886485}	{7968.04, 368.694, 13.2257, 0.774085}
200	$\boldsymbol{\beta}$	{0.00478657, 0.995213}	{0.00211898, 0.048302, 0.949579}	{0.00185082, 0.0144313, 0.124148, 0.85957}
	ρ	4.35	12.49	22.38
	$\boldsymbol{\omega}$	{901.84, 0.985888}	{6090.23, 62.8089, 0.897814}	{11871.8, 481.378, 15.1867, 0.786453}
250	$\boldsymbol{\beta}$	{0.00378101, 0.996219}	{0.00159331, 0.0420853, 0.956321}	{0.00139269, 0.0120297, 0.114584, 0.871993}
	ρ	4.39	13.21	24.75
	$\boldsymbol{\omega}$	{1102.34, 0.988045}	{8082.34, 72.4478, 0.906414}	{16301., 591.753, 17.0536, 0.797245}
300	$\boldsymbol{\beta}$	{0.00311809, 0.996882}	{0.00125948, 0.0375472, 0.961193}	{0.00110797, 0.0104108, 0.106471, 0.88201}
	ρ	4.42	13.77	26.74
	$\boldsymbol{\omega}$	{1305.06, 0.989617}	{10250.9, 81.6684, 0.913233}	{21362.1, 707.502, 18.7691, 0.805757}
350	$\boldsymbol{\beta}$	{0.00264906, 0.997351}	{0.001031, 0.0340609, 0.964908}	{0.000908394, 0.0091685, 0.100232, 0.889691}
	ρ	4.44	14.23	28.49
	$\boldsymbol{\omega}$	{1509.63, 0.990814}	{12580.2, 90.5404, 0.918815}	{27421.7, 850.177, 20.3972, 0.812165}
400	$\boldsymbol{\beta}$	{0.00230021, 0.9977}	{0.000866032, 0.0312827, 0.967851}	{0.000751992, 0.00802258, 0.0955775, 0.895648}
	ρ	4.46	14.62	30.12
	$\boldsymbol{\omega}$	{1715.8, 0.991758}	{15057.7, 99.1151, 0.923493}	{35453.1, 1161.24, 24.1452, 0.825463}
450	$\boldsymbol{\beta}$	{0.00203086, 0.997969}	{0.000742062, 0.0290068, 0.970251}	{0.000612978, 0.00627071, 0.0853787, 0.907738}
	ρ	4.47	14.94	31.82
	$\boldsymbol{\omega}$	{1923.36, 0.992522}	{17673.1, 107.432, 0.927487}	{41329., 1177.71, 25.2645, 0.831242}
500	$\boldsymbol{\beta}$	{0.00181679, 0.998183}	{0.000645947, 0.0271018, 0.972252}	{0.000546643, 0.00623392, 0.0815398, 0.91168}
	ρ	4.49	15.23	32.75

(continued on next page)

Table A.3 (*continued*)

N		$P = 5$
	$\boldsymbol{\omega}$	{2846.74, 411.781, 40.0941, 3.97003, 0.659793}
100	$\boldsymbol{\beta}$	{0.00395334, 0.0134445, 0.0549429, 0.22302, 0.70464}
	ρ	20.34
	$\boldsymbol{\omega}$	{6083.43, 723.916, 58.9841, 4.88096, 0.679269}
150	$\boldsymbol{\beta}$	{0.00248163, 0.00967631, 0.04472, 0.205462, 0.73766}
	ρ	26.24
	$\boldsymbol{\omega}$	{10402.8, 1077.5, 77.4789, 5.6526, 0.693256}
200	$\boldsymbol{\beta}$	{0.00176797, 0.00760734, 0.038404, 0.192846, 0.759375}
	ρ	31.18
	$\boldsymbol{\omega}$	{15750.6, 1464.91, 95.6673, 6.33405, 0.704153}
250	$\boldsymbol{\beta}$	{0.00135255, 0.00628676, 0.0340103, 0.183094, 0.775256}
	ρ	35.47
	$\boldsymbol{\omega}$	{22085.5, 1881.21, 113.603, 6.95081, 0.713063}
300	$\boldsymbol{\beta}$	{0.00108339, 0.00536588, 0.0307308, 0.175201, 0.787619}
	ρ	39.29
	$\boldsymbol{\omega}$	{29373.9, 2322.91, 131.323, 7.51817, 0.720588}
350	$\boldsymbol{\beta}$	{0.000896175, 0.00468493, 0.028165, 0.168605, 0.797649}
	ρ	42.75
	$\boldsymbol{\omega}$	{37587.8, 2787.39, 148.854, 8.04621, 0.727091}
400	$\boldsymbol{\beta}$	{0.000759202, 0.00415986, 0.0260888, 0.162962, 0.806030}
	ρ	45.91
	$\boldsymbol{\omega}$	{46703.1, 3272.60, 166.218, 8.54196, 0.732811}
450	$\boldsymbol{\beta}$	{0.000655107, 0.00374204, 0.0243656, 0.158048, 0.813189}
	ρ	48.84
	$\boldsymbol{\omega}$	{56698.8, 3776.87, 183.430, 9.01057, 0.737910}
500	$\boldsymbol{\beta}$	{0.000573622, 0.00340130, 0.0229066, 0.153708, 0.819411}
	ρ	51.56

Table A.4
Parameters for optimized $P = 6$ SRJ schemes for various values of N ($N = 100 + 50k$, $k = 0, \ldots, 18$).

N	Optimal Scheme Parameters	ρ
100	$\boldsymbol{\omega} = \{3177.91, 734.924, 110.636, 15.7187, 2.41695, 0.614642\}$ $\boldsymbol{\beta} = \{0.00367219, 0.00906128, 0.0276578, 0.0863039, 0.261008, 0.612297\}$	23.75
150	$\boldsymbol{\omega} = \{6903.64, 1362.54, 176.155, 21.6587, 2.85787, 0.629548\}$ $\boldsymbol{\beta} = \{0.0023493, 0.00645108, 0.0217415, 0.0745986, 0.249258, 0.645601\}$	31.57
200	$\boldsymbol{\omega} = \{11953.5, 2106.17, 244.75, 27.1876, 3.22218, 0.640444\}$ $\boldsymbol{\beta} = \{0.00169767, 0.00503682, 0.018218, 0.0669113, 0.240064, 0.668072\}$	38.38
250	$\boldsymbol{\omega} = \{18283.1, 2948.76, 315.677, 32.4263, 3.53814, 0.649052\}$ $\boldsymbol{\beta} = \{0.00131373, 0.00414205, 0.01583, 0.0613214, 0.232565, 0.684828\}$	44.49
300	$\boldsymbol{\omega} = \{25858.1, 3878.89, 388.49, 37.4429, 3.82008, 0.656173\}$ $\boldsymbol{\beta} = \{0.00106245, 0.00352207, 0.0140831, 0.0570002, 0.226254, 0.698078\}$	50.06
350	$\boldsymbol{\omega} = \{34650.4, 4888.21, 462.888, 42.2805, 4.07647, 0.662248\}$ $\boldsymbol{\beta} = \{0.000886166, 0.00306586, 0.0127387, 0.0535196, 0.220818, 0.708972\}$	55.22
400	$\boldsymbol{\omega} = \{44636.2, 5970.3, 538.656, 46.9691, 4.31277, 0.667545\}$ $\boldsymbol{\beta} = \{0.000756216, 0.00271546, 0.0116658, 0.0506321, 0.216052, 0.718178\}$	60.04
450	$\boldsymbol{\omega} = \{55794.8, 7119.99, 615.629, 51.5305, 4.53277, 0.67224\}$ $\boldsymbol{\beta} = \{0.000656794, 0.00243753, 0.0107858, 0.0481827, 0.211815, 0.726123\}$	64.57
500	$\boldsymbol{\omega} = \{68108.3, 8333.03, 693.68, 55.9812, 4.7392, 0.676457\}$ $\boldsymbol{\beta} = \{0.000578495, 0.0022115, 0.0100486, 0.0460682, 0.208004, 0.733089\}$	68.86
550	$\boldsymbol{\omega} = \{81560.3, 9605.84, 772.706, 60.3341, 4.93412, 0.680283\}$ $\boldsymbol{\beta} = \{0.000515383, 0.00202394, 0.0094204, 0.044217, 0.204545, 0.739278\}$	72.94
600	$\boldsymbol{\omega} = \{96136.1, 10935.4, 852.621, 64.5997, 5.1191, 0.683785\}$ $\boldsymbol{\beta} = \{0.000463536, 0.00186573, 0.00887743, 0.0425775, 0.201382, 0.744834\}$	76.82
650	$\boldsymbol{\omega} = \{111822., 12318.9, 933.355, 68.7865, 5.29542, 0.687012\}$ $\boldsymbol{\beta} = \{0.000420261, 0.00173043, 0.00840264, 0.0411115, 0.198469, 0.749866\}$	80.55
700	$\boldsymbol{\omega} = \{128607., 13754.1, 1014.85, 72.9015, 5.46409, 0.690005\}$ $\boldsymbol{\beta} = \{0.000383651, 0.00161335, 0.00798331, 0.0397896, 0.195772, 0.754458\}$	84.12
750	$\boldsymbol{\omega} = \{146478., 15239., 1097.05, 76.9509, 5.62594, 0.692794\}$ $\boldsymbol{\beta} = \{0.000352319, 0.00151104, 0.00760979, 0.0385893, 0.193262, 0.758676\}$	87.56
800	$\boldsymbol{\omega} = \{165426., 16771.6, 1179.9, 80.9397, 5.78167, 0.695405\}$ $\boldsymbol{\beta} = \{0.000325232, 0.00142084, 0.00727458, 0.0374926, 0.190916, 0.76257\}$	90.88
850	$\boldsymbol{\omega} = \{185440., 18350.2, 1263.38, 84.8724, 5.93187, 0.69786\}$ $\boldsymbol{\beta} = \{0.000301608, 0.00134071, 0.00697179, 0.0364852, 0.188715, 0.766185\}$	94.09
900	$\boldsymbol{\omega} = \{206511., 19973.4, 1347.45, 88.7529, 6.07703, 0.700175\}$ $\boldsymbol{\beta} = \{0.000280844, 0.00126905, 0.00669671, 0.0355554, 0.186644, 0.769554\}$	97.20
950	$\boldsymbol{\omega} = \{228631., 21639.7, 1432.07, 92.5845, 6.21758, 0.702366\}$ $\boldsymbol{\beta} = \{0.000262466, 0.00120458, 0.0064455, 0.0346935, 0.184687, 0.772707\}$	100.21
1000	$\boldsymbol{\omega} = \{251790., 23347.8, 1517.22, 96.3704, 6.3539, 0.704445\}$ $\boldsymbol{\beta} = \{0.000246098, 0.00114626, 0.00621504, 0.0338915, 0.182835, 0.775666\}$	103.13

Table A.5
Parameters for optimized $P = 6$ SRJ schemes for various values of N ($N = 2^k\, k = 5, \ldots, 15$).

N	Optimal Scheme Parameters	ρ
32	$\boldsymbol{\omega} = \{354.762, 126.248, 29.5834, 6.40284, 1.53177, 0.576323\}$ $\boldsymbol{\beta} = \{0.0119532, 0.0222174, 0.051387, 0.123168, 0.282127, 0.509147\}$	10.08
64	$\boldsymbol{\omega} = \{1349.25, 370.523, 66.1485, 11.0483, 2.01538, 0.59895\}$ $\boldsymbol{\beta} = \{0.0059103, 0.0130086, 0.0356183, 0.100213, 0.271745, 0.573505\}$	17.15
128	$\boldsymbol{\omega} = \{5098.06, 1070.77, 146.886, 19.1063, 2.67591, 0.62365\}$ $\boldsymbol{\beta} = \{0.00280225, 0.00737736, 0.0239167, 0.0790606, 0.254045, 0.632799\}$	28.28
256	$\boldsymbol{\omega} = \{19127., 3055.94, 324.322, 33.039, 3.57356, 0.649974\}$ $\boldsymbol{\beta} = \{0.00127813, 0.00405608, 0.0155927, 0.0607468, 0.231752, 0.686574\}$	45.18
512	$\boldsymbol{\omega} = \{71233.5, 8633.18, 712.561, 57.0344, 4.78697, 0.677408\}$ $\boldsymbol{\beta} = \{0.000562137, 0.00216338, 0.00988896, 0.0456021, 0.207144, 0.734639\}$	69.86
1024	$\boldsymbol{\omega} = \{263274.200, 24182.2023, 1558.26459, 98.1721442, 6.41792734, 0.70540635\}$ $\boldsymbol{\beta} = \{0.000238864, 0.00112020, 0.00611101, 0.0335258, 0.181980, 0.777025\}$	104.5
2048	$\boldsymbol{\omega} = \{965411.762, 67235.3401, 3391.89306, 168.347866, 8.59713962, 0.73342290\}$ $\boldsymbol{\beta} = \{0.0000982338, 0.000563635, 0.00368554, 0.0241846, 0.157526, 0.813942\}$	151.3
4096	$\boldsymbol{\omega} = \{3511588.84, 185687.226, 7348.16859, 287.430514, 11.4912775, 0.76094204\}$ $\boldsymbol{\beta} = \{0.0000391657, 0.000275898, 0.00217265, 0.0171466, 0.134626, 0.845740\}$	211.8
8192	$\boldsymbol{\omega} = \{12668072.6, 509620.165, 15841.4365, 488.419035, 15.3117934, 0.787502901\}$ $\boldsymbol{\beta} = \{0.0000151650, 0.000131569, 0.00125389, 0.0119658, 0.113783, 0.872850\}$	287.3
16384	$\boldsymbol{\omega} = \{45320546.3, 1390363.37, 33981.7837, 825.831823, 20.3259724, 0.81271835\}$ $\boldsymbol{\beta} = \{0.00000571292, 0.0000612205, 0.000709569, 0.00823073, 0.0952379, 0.895755\}$	377.6
32768	$\boldsymbol{\omega} = \{160790516, 3771703.63, 72530.6624, 1389.31261, 26.8709119, 0.83628657\}$ $\boldsymbol{\beta} = \{0.00000209794, 0.0000278446, 0.000394356, 0.00558784, 0.0790411, 0.914947\}$	481.6

Table A.6
Parameters for optimized $P = 7$ SRJ schemes for various values of N ($N = 100 + 50k$, $k = 0, \ldots, 18$).

N	Optimal Scheme Parameters	ρ
100	$\boldsymbol{\omega} = \{3392.32, 1082.34, 224.848, 42.594, 8.09726, 1.71942, 0.585658\}$ $\boldsymbol{\beta} = \{0.00342091, 0.0068221, 0.0169684, 0.0438593, 0.113205, 0.27876, 0.536965\}$	26.38
150	$\boldsymbol{\omega} = \{7440.09, 2083.58, 378.528, 63.3295, 10.6285, 1.97391, 0.59717\}$ $\boldsymbol{\beta} = \{0.00222099, 0.00482464, 0.0130515, 0.0364807, 0.101845, 0.272802, 0.568775\}$	35.79
200	$\boldsymbol{\omega} = \{12975.6, 3307.79, 547.016, 83.8798, 12.8988, 2.18024, 0.605662\}$ $\boldsymbol{\beta} = \{0.00162297, 0.00375204, 0.0107739, 0.0318451, 0.0940257, 0.267403, 0.590577\}$	44.19
250	$\boldsymbol{\omega} = \{19962.8, 4727.89, 727.329, 104.286, 14.9921, 2.35673, 0.612421\}$ $\boldsymbol{\beta} = \{0.00126735, 0.00307737, 0.00925585, 0.0285774, 0.0881481, 0.262644, 0.60703\}$	51.87
300	$\boldsymbol{\omega} = \{28373.8, 6325.18, 917.591, 124.573, 16.9537, 2.51252, 0.618048\}$ $\boldsymbol{\beta} = \{0.00103282, 0.00261183, 0.00815942, 0.0261112, 0.0834859, 0.258429, 0.62017\}$	59.01
350	$\boldsymbol{\omega} = \{38185.5, 8085.74, 1116.5, 144.759, 18.8121, 2.65293, 0.622874\}$ $\boldsymbol{\beta} = \{0.000867201, 0.00227035, 0.00732423, 0.0241631, 0.0796508, 0.254661, 0.631063\}$	65.71
400	$\boldsymbol{\omega} = \{49378.2, 9998.61, 1323.09, 164.856, 20.586, 2.78137, 0.627102\}$ $\boldsymbol{\beta} = \{0.000744401, 0.00200871, 0.00666341, 0.0225732, 0.0764118, 0.251261, 0.640337\}$	72.05
450	$\boldsymbol{\omega} = \{61935., 12054.9, 1536.61, 184.873, 22.2891, 2.90016, 0.630867\}$ $\boldsymbol{\beta} = \{0.000649961, 0.0018016, 0.00612545, 0.0212434, 0.0736213, 0.248167, 0.648391\}$	78.08
500	$\boldsymbol{\omega} = \{75840.7, 14247.2, 1756.46, 204.819, 23.9316, 3.01099, 0.634261\}$ $\boldsymbol{\beta} = \{0.000575235, 0.00163343, 0.00567765, 0.0201097, 0.0711791, 0.24533, 0.655495\}$	83.85
550	$\boldsymbol{\omega} = \{91082., 16569.2, 1982.16, 224.699, 25.5213, 3.11509, 0.637353\}$ $\boldsymbol{\beta} = \{0.000514743, 0.00149409, 0.00529822, 0.0191281, 0.0690147, 0.242712, 0.661838\}$	89.38
600	$\boldsymbol{\omega} = \{107646., 19015.4, 2213.29, 244.52, 27.0643, 3.21343, 0.640191\}$ $\boldsymbol{\beta} = \{0.000464851, 0.00137668, 0.00497197, 0.0182675, 0.0670763, 0.240282, 0.667561\}$	94.70
650	$\boldsymbol{\omega} = \{125523., 21581., 2449.49, 264.284, 28.5658, 3.30677, 0.642816\}$ $\boldsymbol{\beta} = \{0.000423052, 0.00127638, 0.00468803, 0.0175049, 0.0653251, 0.238015, 0.672767\}$	99.84
700	$\boldsymbol{\omega} = \{144700., 24261.7, 2690.48, 283.996, 30.0298, 3.39572, 0.645256\}$ $\boldsymbol{\beta} = \{0.000387569, 0.00118966, 0.00443832, 0.0168231, 0.0637313, 0.235892, 0.677538\}$	104.8
750	$\boldsymbol{\omega} = \{165170., 27053.7, 2935.97, 303.66, 31.4598, 3.48076, 0.647538\}$ $\boldsymbol{\beta} = \{0.000357102, 0.00111394, 0.00421676, 0.0162087, 0.0622713, 0.233896, 0.681936\}$	109.6
800	$\boldsymbol{\omega} = \{186923., 29953.6, 3185.75, 323.277, 32.8588, 3.56231, 0.649679\}$ $\boldsymbol{\beta} = \{0.000330683, 0.00104722, 0.00401865, 0.0156514, 0.0609266, 0.232012, 0.686013\}$	114.3
850	$\boldsymbol{\omega} = \{209951., 32958.3, 3439.6, 342.851, 34.2293, 3.64072, 0.651697\}$ $\boldsymbol{\beta} = \{0.000307575, 0.000987994, 0.00384032, 0.015143, 0.0596819, 0.23023, 0.689809\}$	118.9
900	$\boldsymbol{\omega} = \{234245., 36064.8, 3697.33, 362.383, 35.5733, 3.71629, 0.653606\}$ $\boldsymbol{\beta} = \{0.000287208, 0.000935052, 0.00367881, 0.0146766, 0.0585247, 0.228538, 0.693359\}$	123.3
950	$\boldsymbol{\omega} = \{259798., 39270.7, 3958.79, 381.876, 36.893, 3.78925, 0.655415\}$ $\boldsymbol{\beta} = \{0.000269134, 0.000887441, 0.00353176, 0.0142469, 0.0574448, 0.226929, 0.696691\}$	127.6
1000	$\boldsymbol{\omega} = \{286604., 42573.3, 4223.81, 401.331, 38.1899, 3.85983, 0.657136\}$ $\boldsymbol{\beta} = \{0.000252996, 0.000844392, 0.00339723, 0.0138493, 0.0564334, 0.225395, 0.699828\}$	131.9

Table A.7
Parameters for optimized $P = 7$ SRJ schemes for various values of N ($N = 2^k$ $k = 5, \ldots, 15$).

N	Optimal Scheme Parameters	ρ
32	$\boldsymbol{\omega} = \{370.035, 167.331, 51.1952, 13.9321, 3.80777, 1.18727, 0.556551\}$ $\boldsymbol{\beta} = \{0.0107542, 0.0171537, 0.0336988, 0.0699421, 0.144888, 0.282064, 0.441499\}$	10.68
64	$\boldsymbol{\omega} = \{1426.38, 523.554, 126.345, 27.5077, 6.01245, 1.48211, 0.57366\}$ $\boldsymbol{\beta} = \{0.005425, 0.00988035, 0.0224061, 0.0531545, 0.125891, 0.28267, 0.500573\}$	18.67
128	$\boldsymbol{\omega} = \{5473.26, 1613.5, 308.842, 54.2321, 9.55437, 1.86957, 0.592601\}$ $\boldsymbol{\beta} = \{0.00263361, 0.00553096, 0.0144785, 0.0392473, 0.106251, 0.275379, 0.55648\}$	31.80
256	$\boldsymbol{\omega} = \{20897.4, 4910.51, 749.664, 106.726, 15.2337, 2.37639, 0.613148\}$ $\boldsymbol{\beta} = \{0.00123422, 0.00301272, 0.0091062, 0.0282465, 0.0875342, 0.262111, 0.608755\}$	52.76
512	$\boldsymbol{\omega} = \{79377.3, 14792.9, 1810.11, 209.596, 24.3177, 3.03654, 0.635029\}$ $\boldsymbol{\beta} = \{0.00055958, 0.00159767, 0.00558101, 0.0198616, 0.0706365, 0.244683, 0.65708\}$	85.20
1024	$\boldsymbol{\omega} = \{299914.156, 44192.3571, 4352.25238, 410.656961, 38.8047154, 3.89292506, 0.65793356\}$ $\boldsymbol{\beta} = \{0.000245852, 0.000825159, 0.00333663, 0.0136687, 0.0559701, 0.224683, 0.701271\}$	133.8
2048	$\boldsymbol{\omega} = \{1127025.97, 131100.588, 10426.4645, 802.498148, 61.8325621, 4.99924687, 0.68152912\}$ $\boldsymbol{\beta} = \{0.000104810, 0.000415215, 0.00194787, 0.00921977, 0.0436271, 0.203420, 0.741265\}$	204.5
4096	$\boldsymbol{\omega} = \{4211571.43, 386588.015, 24893.3205, 1563.73805, 98.3060734, 6.42233602, 0.70547174\}$ $\boldsymbol{\beta} = \{0.0000434063, 0.000203656, 0.00111148, 0.00610263, 0.0335008, 0.181922, 0.777116\}$	303.8
8192	$\boldsymbol{\omega} = \{15648209.9, 1133886.05, 59234.7606, 3037.61946, 155.857928, 8.24483034, 0.72942127\}$ $\boldsymbol{\beta} = \{0.0000174827, 0.0000974264, 0.000620536, 0.00396819, 0.0253729, 0.160955, 0.808968\}$	438.7
16384	$\boldsymbol{\omega} = \{57801287.0, 3309510.67, 140475.554, 5881.12587, 246.314088, 10.5684196, 0.75305435\}$ $\boldsymbol{\beta} = \{0.00000685565, 0.0000454934, 0.000339312, 0.00253741, 0.0189738, 0.141061, 0.837036\}$	616.2
32768	$\boldsymbol{\omega} = \{212234180, 9615316.86, 331986.888, 11346.7836, 387.921369, 13.5177566, 0.77607577\}$ $\boldsymbol{\beta} = \{0.00000262045, 0.0000207546, 0.000181910, 0.00159715, 0.0140222, 0.122590, 0.861585\}$	842.0

Table A.8
Parameters for optimized $P = 8$ SRJ schemes for various values of N ($N = 100 + 50k$, $k = 0, \ldots, 18$).

N	Optimal Scheme Parameters	ρ
100	$\boldsymbol{\omega} = \{3537.92, 1419.79, 377.85, 89.614, 20.816, 4.96187, 1.3481, 0.566216\}$ $\boldsymbol{\beta} = \{0.00319391, 0.00549334, 0.0117303, 0.0263687, 0.0597575, 0.134042, 0.283454, 0.47596\}$	28.46
150	$\boldsymbol{\omega} = \{7806.72, 2810.62, 663.524, 140.897, 29.3973, 6.27138, 1.51276, 0.575281\}$ $\boldsymbol{\beta} = \{0.00209747, 0.00386747, 0.00888556, 0.0213663, 0.0517106, 0.124107, 0.282297, 0.505669\}$	39.17
200	$\boldsymbol{\omega} = \{13677.4, 4551.55, 987.73, 194.11, 37.56, 7.41325, 1.64438, 0.582004\}$ $\boldsymbol{\beta} = \{0.00154611, 0.00299977, 0.00725956, 0.018313, 0.0464484, 0.116968, 0.280231, 0.526234\}$	48.90
250	$\boldsymbol{\omega} = \{21120.6, 6606.74, 1343.7, 248.789, 45.4233, 8.44441, 1.75577, 0.587377\}$ $\boldsymbol{\beta} = \{0.00121592, 0.00245618, 0.00618861, 0.0162041, 0.0426301, 0.111443, 0.277985, 0.541877\}$	57.94
300	$\boldsymbol{\omega} = \{30113.4, 8950.92, 1727.05, 304.653, 53.0547, 9.39499, 1.85328, 0.591865\}$ $\boldsymbol{\beta} = \{0.000996887, 0.00208219, 0.00542207, 0.0146373, 0.0396817, 0.106964, 0.275766, 0.554449\}$	66.43
350	$\boldsymbol{\omega} = \{40636.8, 11564.9, 2134.67, 361.517, 60.4978, 10.2833, 1.94055, 0.595727\}$ $\boldsymbol{\beta} = \{0.000841422, 0.00180845, 0.00484238, 0.0134154, 0.0373083, 0.103214, 0.273641, 0.564929\}$	74.48
400	$\boldsymbol{\omega} = \{52674.7, 14433.5, 2564.23, 419.246, 67.7827, 11.1214, 2.01992, 0.599121\}$ $\boldsymbol{\beta} = \{0.000725636, 0.00159908, 0.00438648, 0.0124289, 0.0353398, 0.1, 0.271626, 0.573894\}$	82.16
450	$\boldsymbol{\omega} = \{66212.9, 17544.1, 3013.86, 477.737, 74.9318, 11.918, 2.09294, 0.60215\}$ $\boldsymbol{\beta} = \{0.000636233, 0.00143358, 0.00401723, 0.0116114, 0.03367, 0.0971967, 0.269722, 0.581713\}$	89.53
500	$\boldsymbol{\omega} = \{81238.9, 20886.1, 3482.07, 536.911, 81.9618, 12.6794, 2.16076, 0.60489\}$ $\boldsymbol{\beta} = \{0.000565234, 0.00129935, 0.00371125, 0.0109201, 0.03223, 0.0947158, 0.267924, 0.588635\}$	96.62
550	$\boldsymbol{\omega} = \{97741.4, 24450.5, 3967.62, 596.704, 88.8863, 13.4104, 2.22421, 0.607385\}$ $\boldsymbol{\beta} = \{0.000507567, 0.00118823, 0.003453, 0.010326, 0.0309663, 0.0924954, 0.266225, 0.594839\}$	103.5
600	$\boldsymbol{\omega} = \{115710., 28229.4, 4469.46, 657.063, 95.716, 14.1148, 2.28393, 0.609684\}$ $\boldsymbol{\beta} = \{0.000459858, 0.00109468, 0.00323173, 0.00980844, 0.0298481, 0.0904893, 0.264614, 0.600453\}$	110.1
650	$\boldsymbol{\omega} = \{135136., 32215.7, 4986.68, 717.944, 102.46, 14.7956, 2.34042, 0.611813\}$ $\boldsymbol{\beta} = \{0.000419774, 0.00101481, 0.00303977, 0.0093526, 0.0288479, 0.0886624, 0.263086, 0.605577\}$	116.5
700	$\boldsymbol{\omega} = \{156011., 36403.2, 5518.5, 779.308, 109.125, 15.4554, 2.39408, 0.613796\}$ $\boldsymbol{\beta} = \{0.000385654, 0.000945799, 0.00287144, 0.00894728, 0.0279458, 0.0869875, 0.261633, 0.610284\}$	122.8
750	$\boldsymbol{\omega} = \{178325., 40786.3, 6064.21, 841.124, 115.718, 16.0962, 2.44524, 0.615653\}$ $\boldsymbol{\beta} = \{0.000356284, 0.000885566, 0.00272248, 0.00858393, 0.0271264, 0.0854428, 0.260247, 0.614635\}$	128.9
800	$\boldsymbol{\omega} = \{202073., 45359.8, 6623.2, 903.364, 122.244, 16.7198, 2.49417, 0.617399\}$ $\boldsymbol{\beta} = \{0.000330755, 0.000832525, 0.00258962, 0.00825589, 0.0263774, 0.0840111, 0.258925, 0.618678\}$	134.9
850	$\boldsymbol{\omega} = \{227246., 50119.2, 7194.93, 966.001, 128.708, 17.3276, 2.5411, 0.619047\}$ $\boldsymbol{\beta} = \{0.000308374, 0.000785453, 0.00247029, 0.00795791, 0.0256892, 0.082678, 0.257659, 0.622452\}$	140.7
900	$\boldsymbol{\omega} = \{253839., 55060., 7778.89, 1029.01, 135.114, 17.921, 2.58622, 0.620607\}$ $\boldsymbol{\beta} = \{0.000288606, 0.00074339, 0.00236245, 0.00768573, 0.0250537, 0.0814319, 0.256446, 0.625988\}$	146.4
950	$\boldsymbol{\omega} = \{281844., 60178.5, 8374.63, 1092.39, 141.465, 18.501, 2.6297, 0.622089\}$ $\boldsymbol{\beta} = \{0.000271028, 0.000705575, 0.00226446, 0.00743591, 0.0244645, 0.0802629, 0.255282, 0.629313\}$	151.9
1000	$\boldsymbol{\omega} = \{311256., 65470.9, 8981.74, 1156.09, 147.766, 19.0687, 2.67166, 0.6235\}$ $\boldsymbol{\beta} = \{0.000255303, 0.000671391, 0.00217498, 0.0072056, 0.0239159, 0.0791628, 0.254163, 0.632451\}$	157.4

Table A.9
Parameters for optimized $P = 8$ SRJ schemes for various values of N ($N = 2^k\, k = 5, \dots, 15$).

N	Optimal Scheme Parameters	ρ
32	$\boldsymbol{\omega} = \{380.243, 203.177, 76.1463, 24.9346, 7.91926, 2.60779, 0.993193, 0.543507\}$ $\boldsymbol{\beta} = \{0.00977378, 0.0140492, 0.0244443, 0.0454149, 0.0855457, 0.15853, 0.273683, 0.388559\}$	11.14
64	$\boldsymbol{\omega} = \{1478.41, 666.114, 202.545, 54.3776, 14.2439, 3.8448, 1.19182, 0.55682\}$ $\boldsymbol{\beta} = \{0.0050073, 0.00800348, 0.0157692, 0.0329114, 0.0693924, 0.144485, 0.282072, 0.44236\}$	19.84
128	$\boldsymbol{\omega} = \{5729.04, 2152.82, 532.541, 118.056, 25.6833, 5.72098, 1.44559, 0.571677\}$ $\boldsymbol{\beta} = \{0.00247575, 0.004441, 0.00991522, 0.0232218, 0.0547725, 0.128023, 0.283006, 0.494145\}$	34.60
256	$\boldsymbol{\omega} = \{22118.3, 6873.21, 1388.33, 255.434, 46.3503, 8.5623, 1.76811, 0.587956\}$ $\boldsymbol{\beta} = \{0.00118505, 0.00240419, 0.00608363, 0.0159926, 0.0422379, 0.110858, 0.277715, 0.543523\}$	58.98
512	$\boldsymbol{\omega} = \{83648.1, 21109.1, 3504.48, 539.515, 82.2596, 12.7104, 2.1634, 0.604989\}$ $\boldsymbol{\beta} = \{0.000565345, 0.00129282, 0.00369896, 0.0108933, 0.0321732, 0.094622, 0.267859, 0.588895\}$	98.21
1024	$\boldsymbol{\omega} = \{325872.429, 68072.0251, 9277.08426, 1186.79068, 150.772525, 19.3370248, 2.69130897, 0.62415405\}$ $\boldsymbol{\beta} = \{0.000248331, 0.000656122, 0.00213472, 0.00710128, 0.0236657, 0.0786571, 0.253641, 0.633896\}$	160.0
2048	$\boldsymbol{\omega} = \{1243390.54, 211839.541, 23840.6659, 2549.96741, 271.441418, 29.0896995, 3.33872799, 0.64369876\}$ $\boldsymbol{\beta} = \{0.000109012, 0.000330956, 0.00122248, 0.00459014, 0.0172536, 0.0647443, 0.237249, 0.674501\}$	254.3
4096	$\boldsymbol{\omega} = \{4724941.11, 655366.979, 61084.1980, 5467.76799, 487.846845, 43.7324841, 4.15010426, 0.66392578\}$ $\boldsymbol{\beta} = \{0.0000465981, 0.000163126, 0.000685047, 0.00291125, 0.0123798, 0.0525884, 0.219311, 0.711915\}$	394.6
8192	$\boldsymbol{\omega} = \{17880305.2, 2017253.16, 156092.848, 11699.7068, 875.030230, 65.6587728, 5.16366699, 0.68460941\}$ $\boldsymbol{\beta} = \{0.0000194120, 0.0000785878, 0.000375889, 0.00181326, 0.00875016, 0.0421983, 0.200638, 0.746127\}$	438.7
16384	$\boldsymbol{\omega} = \{67375460.0, 6181549.54, 397871.724, 24979.4457, 1566.01561, 98.3988686, 6.42559672, 0.70552038\}$ $\boldsymbol{\beta} = \{0.00000788702, 0.0000370175, 0.000202091, 0.00110994, 0.00609737, 0.0334824, 0.181878, 0.777185\}$	882.9
32768	$\boldsymbol{\omega} = \{252775864, 18866153.6, 1011634.78, 53208.1901, 2795.89696, 147.142217, 7.99143284, 0.72643283\}$ $\boldsymbol{\beta} = \{.000000312768, 0.0000170557, 0.000106532, 0.000668220, 0.00419188, 0.0262904, 0.163531, 0.805192\}$	1273

Table A.10
Parameters for optimized $P = 9$ SRJ schemes for various values of N ($N = 100 + 50k$, $k = 0, \ldots, 18$).

N	Optimal Scheme Parameters	ρ
100	$\boldsymbol{\omega} = \{3640.86, 1729.24, 559.604, 158.917, 43.5786, 11.9645, 3.41756, 1.12661, 0.552629\}$ $\boldsymbol{\beta} = \{0.00298948, 0.00462251, 0.0087721, 0.0176949, 0.0362139, 0.0740779, 0.149046, 0.280346, 0.426238\}$	30.12
150	$\boldsymbol{\omega} = \{8067.04, 3497.09, 1016.04, 260.97, 65.0344, 16.2237, 4.19023, 1.24183, 0.559916\}$ $\boldsymbol{\beta} = \{0.00198105, 0.00324426, 0.00656828, 0.0140648, 0.0304787, 0.0660202, 0.141034, 0.282856, 0.453753\}$	41.92
200	$\boldsymbol{\omega} = \{14177.3, 5750.32, 1548.34, 370.696, 86.3811, 20.1478, 4.84929, 1.33292, 0.565337\}$ $\boldsymbol{\beta} = \{0.0014704, 0.00251182, 0.00532492, 0.0118951, 0.0268474, 0.0605723, 0.135008, 0.283441, 0.472929\}$	52.78
250	$\boldsymbol{\omega} = \{21947.5, 8446.56, 2144.68, 486.462, 107.643, 23.8396, 5.43484, 1.4094, 0.56968\}$ $\boldsymbol{\beta} = \{0.0011629, 0.00205427, 0.00451336, 0.0104186, 0.0242715, 0.0565249, 0.130206, 0.283258, 0.48759\}$	62.97
300	$\boldsymbol{\omega} = \{31358.5, 11555.4, 2797.26, 607.253, 128.836, 27.3558, 5.9677, 1.4759, 0.573317\}$ $\boldsymbol{\beta} = \{0.000957974, 0.0017401, 0.00393644, 0.009334, 0.0223169, 0.0533419, 0.126229, 0.282718, 0.499425\}$	72.62
350	$\boldsymbol{\omega} = \{42394.6, 15053.5, 3500.43, 732.375, 149.969, 30.7322, 6.46034, 1.53511, 0.576452\}$ $\boldsymbol{\beta} = \{0.000811947, 0.00151052, 0.00350254, 0.00849594, 0.0207656, 0.050741, 0.122843, 0.282002, 0.509327\}$	81.84
400	$\boldsymbol{\omega} = \{55042.3, 18922., 4249.87, 861.316, 171.05, 33.993, 6.92094, 1.5887, 0.579211\}$ $\boldsymbol{\beta} = \{0.000702809, 0.00133513, 0.00316284, 0.00782458, 0.0194944, 0.0485563, 0.119902, 0.281197, 0.517825\}$	90.69
450	$\boldsymbol{\omega} = \{69289.9, 23145.3, 5042.15, 993.684, 192.085, 37.1561, 7.35522, 1.63782, 0.581677\}$ $\boldsymbol{\beta} = \{0.000618273, 0.00119662, 0.00288877, 0.00727198, 0.0184271, 0.0466826, 0.117306, 0.28035, 0.525258\}$	99.23
500	$\boldsymbol{\omega} = \{85126.9, 27710.2, 5874.43, 1129.16, 213.078, 40.2346, 7.76736, 1.68326, 0.58391\}$ $\boldsymbol{\beta} = \{0.000550948, 0.00108438, 0.00266241, 0.00680743, 0.0175139, 0.0450492, 0.114986, 0.279488, 0.531857\}$	107.49
550	$\boldsymbol{\omega} = \{102544., 32605.2, 6744.37, 1267.5, 234.034, 43.2388, 8.16055, 1.72565, 0.58595\}$ $\boldsymbol{\beta} = \{0.000496122, 0.000991529, 0.00247193, 0.00641024, 0.0167209, 0.0436064, 0.112892, 0.278627, 0.537784\}$	115.52
600	$\boldsymbol{\omega} = \{121532., 37820.1, 7649.95, 1408.49, 254.954, 46.1771, 8.53726, 1.76542, 0.587829\}$ $\boldsymbol{\beta} = \{0.000450652, 0.000913401, 0.00230916, 0.0060659, 0.0160234, 0.0423181, 0.110985, 0.277774, 0.54316\}$	123.32
650	$\boldsymbol{\omega} = \{142084., 43346.1, 8589.42, 1551.94, 275.842, 49.0561, 8.89946, 1.80294, 0.589571\}$ $\boldsymbol{\beta} = \{0.000412363, 0.000846729, 0.00216828, 0.0057639, 0.0154037, 0.0411574, 0.109236, 0.276936, 0.548076\}$	130.93
700	$\boldsymbol{\omega} = \{164193., 49175.1, 9561.25, 1697.7, 296.7, 51.8815, 9.24875, 1.8385, 0.591197\}$ $\boldsymbol{\beta} = \{0.000379703, 0.00078915, 0.002045, 0.00549643, 0.0148481, 0.0401035, 0.107622, 0.276115, 0.552601\}$	138.35
750	$\boldsymbol{\omega} = \{187851., 55299.9, 10564.1, 1845.64, 317.53, 54.6579, 9.58648, 1.87232, 0.59272\}$ $\boldsymbol{\beta} = \{0.000351533, 0.00073891, 0.00193613, 0.00525752, 0.0143463, 0.0391402, 0.106125, 0.275312, 0.556792\}$	145.61
800	$\boldsymbol{\omega} = \{213052., 61713.8, 11596.7, 1995.64, 338.333, 57.3892, 9.91376, 1.9046, 0.594153\}$ $\boldsymbol{\beta} = \{0.000327001, 0.000694683, 0.0018392, 0.00504256, 0.0138901, 0.0382548, 0.10473, 0.27453, 0.560692\}$	152.72
850	$\boldsymbol{\omega} = \{239791., 68410.7, 12658.1, 2147.59, 359.111, 60.0791, 10.2315, 1.9355, 0.595507\}$ $\boldsymbol{\beta} = \{0.000305456, 0.000655443, 0.00175229, 0.0048479, 0.0134729, 0.0374366, 0.103425, 0.273767, 0.564338\}$	159.69
900	$\boldsymbol{\omega} = \{268061., 75385.1, 13747.1, 2301.4, 379.865, 62.7304, 10.5406, 1.96515, 0.59679\}$ $\boldsymbol{\beta} = \{0.000286394, 0.000620388, 0.00167386, 0.00467063, 0.0130894, 0.0366774, 0.102199, 0.273024, 0.56776\}$	166.53
950	$\boldsymbol{\omega} = \{297859., 82631.8, 14863., 2456.98, 400.597, 65.3458, 10.8416, 1.99367, 0.59801\}$ $\boldsymbol{\beta} = \{0.000269416, 0.000588879, 0.0016027, 0.00450836, 0.0127354, 0.0359699, 0.101043, 0.272299, 0.570983\}$	173.24
1000	$\boldsymbol{\omega} = \{329177., 90146., 16004.9, 2614.27, 421.306, 67.9276, 11.1353, 2.02115, 0.599172\}$ $\boldsymbol{\beta} = \{0.000254204, 0.000560401, 0.00153782, 0.00435916, 0.0124072, 0.0353083, 0.0999517, 0.271594, 0.574027\}$	179.84

Table A.11
Parameters for optimized $P = 9$ SRJ schemes for various values of N ($N = 2^k \, k = 5, \dots, 13$).

N	Optimal Scheme Parameters	ρ
32	$\boldsymbol{\omega} = \{387.382, 233.468, 102.424, 38.9705, 14.1267, 5.1286, 1.96283, 0.872543, 0.534479\}$ $\boldsymbol{\beta} = \{0.00895557, 0.0119599, 0.0189643, 0.0322828, 0.0562016, 0.0977771, 0.165994, 0.261519, 0.346345\}$	11.50
64	$\boldsymbol{\omega} = \{1515.04, 792.702, 289.003, 91.8543, 28.038, 8.56994, 2.7402, 1.0158, 0.545103\}$ $\boldsymbol{\beta} = \{0.00464454, 0.00676243, 0.0119546, 0.0225735, 0.0433789, 0.0833122, 0.156892, 0.275129, 0.395353\}$	20.78
128	$\boldsymbol{\omega} = \{5910.34, 2656.28, 804.934, 214.987, 55.6093, 14.4, 3.8679, 1.195, 0.557016\}$ $\boldsymbol{\beta} = \{0.00232987, 0.00372963, 0.007362, 0.0154007, 0.0326349, 0.0691242, 0.144247, 0.282116, 0.443056\}$	36.87
256	$\boldsymbol{\omega} = \{22990.5, 8798.35, 2220.15, 500.707, 110.19, 24.27, 5.50127, 1.41784, 0.570149\}$ $\boldsymbol{\beta} = \{0.00113407, 0.00201056, 0.00443415, 0.0102715, 0.0240097, 0.0561044, 0.129691, 0.283207, 0.489137\}$	64.15
512	$\boldsymbol{\omega} = \{89163.2, 28855.3, 6079.87, 1162.11, 218.111, 40.962, 7.86335, 1.6937, 0.584415\}$ $\boldsymbol{\beta} = \{0.000536789, 0.00106054, 0.00261378, 0.00670659, 0.0173137, 0.044687, 0.114465, 0.279281, 0.533335\}$	109.44
1024	$\boldsymbol{\omega} = \{344749., 93846.5, 16562.1, 2690.35, 431.24, 69.1555, 11.2738, 2.034, 0.599711\}$ $\boldsymbol{\beta} = \{0.000247452, 0.000547682, 0.00150865, 0.00429171, 0.0122579, 0.0350055, 0.0994484, 0.271262, 0.575431\}$	182.96
2048	$\boldsymbol{\omega} = \{1328880, 303078., 44937.7, 6215.79, 851.582, 116.707, 16.1894, 2.45258, 0.615917\}$ $\boldsymbol{\beta} = \{0.000111223, 0.000276977, 0.000853022, 0.00269416, 0.00853108, 0.0270113, 0.0852253, 0.260049, 0.615248\}$	299.64
4096	$\boldsymbol{\omega} = \{5106340, 972991., 121553., 14336.3, 1679.42, 196.78, 23.2619, 2.96607, 0.632898\}$ $\boldsymbol{\beta} = \{0.0000487855, 0.000137195, 0.000472679, 0.00166043, 0.00584223, 0.0205547, 0.0721511, 0.246475, 0.652658\}$	480.54
8192	$\boldsymbol{\omega} = \{19559500, 3107886, 327956., 33012.6, 3307.31, 331.387, 33.4177, 3.59437, 0.650509\}$ $\boldsymbol{\beta} = \{0.0000208973, 0.0000665666, 0.000256798, 0.00100536, 0.00393987, 0.0154394, 0.0604121, 0.231281, 0.687578\}$	754.47

Table A.12
Parameters for optimized $P = 10$ SRJ schemes for various values of N ($N = 100 + 50k$, $k = 0, \ldots, 18$).

N	Optimal Scheme Parameters	ρ
100	$\boldsymbol{\omega} = \{3716.11, 2003.92, 758.907, 249.869, 78.5973, 24.4799, 7.71415, 2.55553, 0.983449, 0.542795\}$ $\boldsymbol{\beta} = \{0.0028058, 0.00400978, 0.00692995, 0.0128148, 0.0241718, 0.0457763, 0.0862794, 0.159105, 0.272849, 0.385258\}$	31.49
150	$\boldsymbol{\omega} = \{8257.94, 4120.6, 1415.79, 425.028, 122.54, 35.0395, 10.1187, 3.05606, 1.06871, 0.548763\}$ $\boldsymbol{\beta} = \{0.00187294, 0.00280821, 0.00514163, 0.0100348, 0.019919, 0.0396644, 0.0786806, 0.153149, 0.278034, 0.410696\}$	44.20
200	$\boldsymbol{\omega} = \{14544.8, 6857.12, 2199.03, 618.829, 167.848, 45.1996, 12.278, 3.47555, 1.13556, 0.553212\}$ $\boldsymbol{\beta} = \{0.00139791, 0.00217152, 0.00414295, 0.00839998, 0.0172885, 0.0356771, 0.0733849, 0.148408, 0.280624, 0.428506\}$	56.02
250	$\boldsymbol{\omega} = \{22556.4, 10166.9, 3091.02, 827.681, 214.189, 55.0715, 14.2712, 3.84341, 1.19131, 0.556783\}$ $\boldsymbol{\beta} = \{0.00111059, 0.00177456, 0.00349571, 0.00730005, 0.0154531, 0.0327874, 0.0693676, 0.1445, 0.282038, 0.442173\}$	67.18
300	$\boldsymbol{\omega} = \{32276.8, 14016.1, 4079.92, 1049.31, 261.365, 64.719, 16.1413, 4.17472, 1.23956, 0.559776\}$ $\boldsymbol{\beta} = \{0.000918413, 0.00150238, 0.00303809, 0.00649915, 0.014078, 0.0305574, 0.0661576, 0.141186, 0.282825, 0.453239\}$	77.83
350	$\boldsymbol{\omega} = \{43692.7, 18378.4, 5157.02, 1282.12, 309.244, 74.1832, 17.9147, 4.47839, 1.28233, 0.56236\}$ $\boldsymbol{\beta} = \{0.000781034, 0.0013037, 0.00269539, 0.00588469, 0.012998, 0.0287629, 0.0635007, 0.138314, 0.283239, 0.462521\}$	88.06
400	$\boldsymbol{\omega} = \{56792.9, 23232.2, 6315.6, 1524.93, 357.729, 83.4929, 19.6092, 4.76023, 1.3209, 0.564637\}$ $\boldsymbol{\beta} = \{0.000678071, 0.00115204, 0.00242805, 0.00539537, 0.0121206, 0.0272748, 0.0612448, 0.135783, 0.283418, 0.470505\}$	97.94
450	$\boldsymbol{\omega} = \{71567.4, 28559.9, 7550.26, 1776.81, 406.749, 92.6693, 21.2376, 5.02426, 1.35614, 0.566674\}$ $\boldsymbol{\beta} = \{0.000598118, 0.00103237, 0.00221302, 0.00499467, 0.0113895, 0.0260125, 0.0592921, 0.133524, 0.283441, 0.477503\}$	107.50
500	$\boldsymbol{\omega} = \{88007.4, 34346., 8856.52, 2037.02, 456.248, 101.729, 22.8094, 5.27342, 1.38866, 0.568519\}$ $\boldsymbol{\beta} = \{0.000534297, 0.000935438, 0.00203589, 0.00465934, 0.0107682, 0.0249227, 0.0575758, 0.131484, 0.283358, 0.483727\}$	116.80
550	$\boldsymbol{\omega} = \{106105., 40577.2, 10230.6, 2304.96, 506.181, 110.684, 24.3319, 5.5099, 1.4189, 0.570207\}$ $\boldsymbol{\beta} = \{0.000482215, 0.000855288, 0.00188718, 0.00437377, 0.0102318, 0.0239683, 0.0560489, 0.129626, 0.2832, 0.489327\}$	125.85
600	$\boldsymbol{\omega} = \{125853., 47241.7, 11669.3, 2580.09, 556.511, 119.545, 25.8109, 5.73541, 1.44721, 0.571763\}$ $\boldsymbol{\beta} = \{0.000438937, 0.000787878, 0.00176037, 0.00412708, 0.00976257, 0.0231228, 0.0546765, 0.127921, 0.282988, 0.494414\}$	134.69
650	$\boldsymbol{\omega} = \{147244., 54329.2, 13169.7, 2861.99, 607.205, 128.321, 27.2511, 5.95132, 1.47387, 0.573207\}$ $\boldsymbol{\beta} = \{0.000402428, 0.000730372, 0.00165081, 0.00391144, 0.00934766, 0.0223663, 0.0534326, 0.126347, 0.282739, 0.499072\}$	143.33
700	$\boldsymbol{\omega} = \{170274., 61830., 14729.5, 3150.28, 658.239, 137.019, 28.6564, 6.15872, 1.49908, 0.574555\}$ $\boldsymbol{\beta} = \{0.000371232, 0.000680723, 0.00155511, 0.00372101, 0.00897738, 0.0216838, 0.052297, 0.124886, 0.282461, 0.503366\}$	151.80
750	$\boldsymbol{\omega} = \{194935., 69735.6, 16346.3, 3444.62, 709.588, 145.645, 30.0301, 6.35853, 1.52302, 0.575818\}$ $\boldsymbol{\beta} = \{0.000344283, 0.000637414, 0.00147072, 0.00355139, 0.00864431, 0.0210638, 0.0512538, 0.123523, 0.282164, 0.507348\}$	160.09
800	$\boldsymbol{\omega} = \{221222., 78038.2, 18018.1, 3744.71, 761.234, 154.205, 31.3748, 6.55149, 1.54582, 0.577009\}$ $\boldsymbol{\beta} = \{0.000320779, 0.000599295, 0.00139569, 0.00339916, 0.00834266, 0.0204969, 0.0502903, 0.122246, 0.281852, 0.511058\}$	168.23
850	$\boldsymbol{\omega} = \{249131., 86730.6, 19743.3, 4050.3, 813.158, 162.702, 32.693, 6.73825, 1.56762, 0.578133\}$ $\boldsymbol{\beta} = \{0.000300108, 0.000565482, 0.0013285, 0.00326163, 0.0080678, 0.0199759, 0.0493961, 0.121045, 0.281529, 0.51453\}$	176.23
900	$\boldsymbol{\omega} = \{278657., 95806., 21520.1, 4361.15, 865.346, 171.141, 33.9867, 6.91935, 1.5885, 0.5792\}$ $\boldsymbol{\beta} = \{0.000281793, 0.00053528, 0.00126795, 0.00313666, 0.00781602, 0.0194947, 0.0485629, 0.119913, 0.2812, 0.517792\}$	184.10
950	$\boldsymbol{\omega} = \{309795., 105258., 23347., 4677.05, 917.783, 179.526, 35.2576, 7.09526, 1.60855, 0.580214\}$ $\boldsymbol{\beta} = \{0.000265459, 0.000508136, 0.00121307, 0.00302252, 0.00758428, 0.0190484, 0.0477835, 0.118841, 0.280866, 0.520867\}$	191.84
1000	$\boldsymbol{\omega} = \{342541., 115082., 25222.8, 4997.8, 970.458, 187.859, 36.5072, 7.2664, 1.62786, 0.581182\}$ $\boldsymbol{\beta} = \{0.000250807, 0.000483607, 0.00116308, 0.00291777, 0.00737009, 0.0186328, 0.0470519, 0.117825, 0.280529, 0.523776\}$	199.47

Table A.13
Parameters for optimized $P = 10$ SRJ schemes for various values of N ($N = 2^k\ k = 5, \dots, 13$).

N	Optimal Scheme Parameters	ρ
32	$\boldsymbol{\omega} = \{392.56, 258.705, 128.513, 55.3119, 22.4376, 8.96645, 3.64502, 1.57769, 0.792114, 0.527982\}$ $\boldsymbol{\beta} = \{0.0082622, 0.0104568, 0.0154228, 0.0244627, 0.0399088, 0.0655551, 0.106845, 0.169025, 0.248, 0.312061\}$	11.78
64	$\boldsymbol{\omega} = \{1541.73, 902.275, 380.269, 138.845, 48.1557, 16.5052, 5.73583, 2.10711, 0.90061, 0.536644\}$ $\boldsymbol{\beta} = \{0.00432745, 0.00588253, 0.00954679, 0.0166301, 0.0296487, 0.0531258, 0.094633, 0.164386, 0.264966, 0.356854\}$	21.54
128	$\boldsymbol{\omega} = \{6043.13, 3109.46, 1109.83, 345.372, 103.01, 30.452, 9.09794, 2.84852, 1.03416, 0.546386\}$ $\boldsymbol{\beta} = \{0.00219628, 0.00323088, 0.00578319, 0.0110516, 0.021504, 0.0419877, 0.081641, 0.155591, 0.276224, 0.40079\}$	38.74
256	$\boldsymbol{\omega} = \{23632.9, 10600.9, 3204.77, 853.638, 219.81, 56.2401, 14.5015, 3.88488, 1.19745, 0.557168\}$ $\boldsymbol{\beta} = \{0.00108359, 0.00173667, 0.00343275, 0.00719109, 0.015268, 0.0324906, 0.0689461, 0.144074, 0.282159, 0.443618\}$	68.49
512	$\boldsymbol{\omega} = \{92199.9, 35801.3, 9180.24, 2100.65, 468.194, 103.887, 23.179, 5.33126, 1.39611, 0.568938\}$ $\boldsymbol{\beta} = \{0.000520856, 0.000914849, 0.0019979, 0.00458673, 0.0106325, 0.0246824, 0.0571936, 0.131022, 0.283326, 0.485123\}$	118.99
1024	$\boldsymbol{\omega} = \{358829., 119927., 26140., 5153.43, 995.822, 191.842, 37.1, 7.34695, 1.63687, 0.58163\}$ $\boldsymbol{\beta} = \{0.000244298, 0.000472653, 0.00114063, 0.00287047, 0.00727288, 0.0184432, 0.0467163, 0.117355, 0.280367, 0.525117\}$	203.09
2048	$\boldsymbol{\omega} = \{1393090, 398936., 74096.7, 12614.5, 2115.46, 354.041, 59.4078, 10.1522, 1.92781, 0.595172\}$ $\boldsymbol{\beta} = \{0.000111934, 0.000239591, 0.000638994, 0.00176393, 0.00489192, 0.0135738, 0.0376372, 0.103747, 0.273958, 0.563438\}$	340.42
4096	$\boldsymbol{\omega} = \{5395040, 1319170, 209311., 30823.9, 4488.88, 652.834, 95.1119, 14.0506, 2.27851, 0.609477\}$ $\boldsymbol{\beta} = \{0.0000501434, 0.000119186, 0.000351366, 0.00106505, 0.00323843, 0.00984977, 0.0299458, 0.0906682, 0.264761, 0.599951\}$	560.18
8192	$\boldsymbol{\omega} = \{20841177, 4339863, 589668., 75210.5, 9514.64, 1202.61, 152.183, 19.4605, 2.70028, 0.624451\}$ $\boldsymbol{\beta} = \{0.0000219770, 0.0000581897, 0.000189695, 0.000632223, 0.00211144, 0.00705278, 0.0235524, 0.0784280, 0.253403, 0.634551\}$	904.73

Table A.14
Parameters for optimized $P = 11$ SRJ schemes for various values of N ($N = 100 + 50k$, $k = 0, \ldots, 18$).

N	Optimal Scheme Parameters	ρ
100	$\boldsymbol{\omega} = \{3772.69, 2243.38, 965.764, 359.834, 127.002, 44.0668, 15.2981, 5.41117, 2.02855, 0.885269, 0.535461\}$ $\boldsymbol{\beta} = \{0.00264078, 0.0035551, 0.00569791, 0.00980872, 0.0173037, 0.0307448, 0.054615, 0.0962556, 0.165235, 0.263076, 0.351068\}$	32.64
150	$\boldsymbol{\omega} = \{8401.79, 4674.24, 1842.62, 630.247, 205.246, 65.8794, 21.1553, 6.90331, 2.37588, 0.951043, 0.540431\}$ $\boldsymbol{\beta} = \{0.00177338, 0.00248623, 0.0041962, 0.00758878, 0.0140228, 0.0260629, 0.048432, 0.0894173, 0.161257, 0.27015, 0.374614\}$	46.12
200	$\boldsymbol{\omega} = \{14822.2, 7852.9, 2907.6, 936.628, 288.313, 87.62, 26.639, 8.21547, 2.66282, 1.00226, 0.544141\}$ $\boldsymbol{\beta} = \{0.00132965, 0.00192098, 0.00336441, 0.0063002, 0.0120299, 0.023085, 0.0442925, 0.0845072, 0.157812, 0.274205, 0.391154\}$	58.75
250	$\boldsymbol{\omega} = \{23016.7, 11730.6, 4137.15, 1272.65, 375.138, 109.305, 31.8602, 9.40806, 2.9118, 1.04478, 0.547122\}$ $\boldsymbol{\beta} = \{0.00106031, 0.00156902, 0.0028284, 0.00544114, 0.0106571, 0.0209649, 0.0412369, 0.0807077, 0.15484, 0.276815, 0.403879\}$	70.76
300	$\boldsymbol{\omega} = \{32971.7, 16271.6, 5515.11, 1634.24, 465.066, 130.944, 36.8811, 10.5132, 3.13414, 1.08142, 0.549623\}$ $\boldsymbol{\beta} = \{0.000879604, 0.00132793, 0.00245107, 0.00482002, 0.00963873, 0.0193512, 0.0388452, 0.0776262, 0.152245, 0.27861, 0.414205\}$	82.28
350	$\boldsymbol{\omega} = \{44676.1, 21447.6, 7029.52, 2018.49, 557.653, 152.544, 41.7411, 11.5505, 3.3365, 1.11379, 0.551784\}$ $\boldsymbol{\beta} = \{0.000750095, 0.00115207, 0.00216949, 0.00434619, 0.00884531, 0.018067, 0.0368981, 0.0750455, 0.149947, 0.279896, 0.422883\}$	93.38
400	$\boldsymbol{\omega} = \{58120.2, 27235.7, 8671., 2423.21, 652.571, 174.109, 46.4672, 12.533, 3.52319, 1.1429, 0.553689\}$ $\boldsymbol{\beta} = \{0.000652808, 0.00101792, 0.00195046, 0.00397066, 0.00820512, 0.017012, 0.0352674, 0.0728327, 0.147888, 0.280843, 0.430359\}$	104.15
450	$\boldsymbol{\omega} = \{73295.5, 33616.3, 10432., 2846.69, 749.573, 195.645, 51.0794, 13.47, 3.69717, 1.16942, 0.555395\}$ $\boldsymbol{\beta} = \{0.000577107, 0.000912097, 0.00177473, 0.00366439, 0.00767482, 0.0161243, 0.0338724, 0.0709012, 0.146025, 0.281552, 0.436922\}$	114.61
500	$\boldsymbol{\omega} = \{90194.6, 40572.9, 12306.1, 3287.54, 848.462, 217.153, 55.5924, 14.3682, 3.86057, 1.19384, 0.556941\}$ $\boldsymbol{\beta} = \{0.000516568, 0.000826427, 0.00163028, 0.003409, 0.00722646, 0.0153633, 0.0326589, 0.0691913, 0.144324, 0.282088, 0.442766\}$	124.81
550	$\boldsymbol{\omega} = \{108811., 48090.9, 14288., 3744.58, 949.075, 238.635, 60.018, 15.233, 4.01502, 1.21651, 0.558356\}$ $\boldsymbol{\beta} = \{0.000467079, 0.000755611, 0.00150923, 0.0031922, 0.00684112, 0.0147012, 0.031589, 0.0676602, 0.142759, 0.282494, 0.448031\}$	134.77
600	$\boldsymbol{\omega} = \{129138., 56157.2, 16373.2, 4216.85, 1051.28, 260.095, 64.3657, 16.0685, 4.16174, 1.23769, 0.559662\}$ $\boldsymbol{\beta} = \{0.000425891, 0.000696065, 0.00140618, 0.00300546, 0.00650546, 0.0141179, 0.0306352, 0.0662761, 0.141312, 0.2828, 0.452819\}$	144.52
650	$\boldsymbol{\omega} = \{151170., 64760.4, 18557.4, 4703.5, 1154.97, 281.534, 68.6429, 16.8779, 4.30172, 1.2576, 0.560873\}$ $\boldsymbol{\beta} = \{0.000391093, 0.000645281, 0.00131728, 0.00284264, 0.00620977, 0.0135987, 0.0297771, 0.065015, 0.139966, 0.283028, 0.457208\}$	154.08
700	$\boldsymbol{\omega} = \{174902., 73889.9, 20837.1, 5203.8, 1260.04, 302.953, 72.8562, 17.664, 4.43575, 1.27639, 0.562005\}$ $\boldsymbol{\beta} = \{0.000361319, 0.000601444, 0.00123973, 0.00269919, 0.0059468, 0.0131326, 0.0289991, 0.0638582, 0.138708, 0.283195, 0.461258\}$	163.46
750	$\boldsymbol{\omega} = \{200329., 83536.2, 23209.1, 5717.11, 1366.41, 324.353, 77.011, 18.4291, 4.56449, 1.29422, 0.563067\}$ $\boldsymbol{\beta} = \{0.000335564, 0.00056321, 0.00117144, 0.0025717, 0.00571101, 0.012711, 0.0282888, 0.0627909, 0.137528, 0.283312, 0.465017\}$	172.68
800	$\boldsymbol{\omega} = \{227446., 93690.5, 25670.5, 6242.84, 1474., 345.736, 81.1118, 19.175, 4.68847, 1.31117, 0.564067\}$ $\boldsymbol{\beta} = \{0.000313074, 0.000529564, 0.00111079, 0.00245751, 0.0054981, 0.0123272, 0.0276365, 0.0618011, 0.136416, 0.283388, 0.468522\}$	181.75
850	$\boldsymbol{\omega} = \{256250., 104345., 28218.7, 6780.48, 1582.76, 367.102, 85.1626, 19.9034, 4.80816, 1.32735, 0.565012\}$ $\boldsymbol{\beta} = \{0.000293271, 0.000499722, 0.00105653, 0.00235455, 0.00530463, 0.0119758, 0.0270345, 0.060879, 0.135366, 0.283431, 0.471805\}$	190.67
900	$\boldsymbol{\omega} = \{286735., 115491., 30851.3, 7329.59, 1692.62, 388.453, 89.167, 20.6157, 4.92393, 1.34284, 0.565909\}$ $\boldsymbol{\beta} = \{0.000275706, 0.000473069, 0.00100769, 0.00226114, 0.00512788, 0.0116523, 0.0264763, 0.0600167, 0.13437, 0.283447, 0.474892\}$	199.47
950	$\boldsymbol{\omega} = \{318899., 127124., 33566., 7889.72, 1803.54, 409.788, 93.128, 21.3131, 5.03613, 1.3577, 0.566763\}$ $\boldsymbol{\beta} = \{0.000260025, 0.000449118, 0.000963456, 0.00217597, 0.00496559, 0.0113534, 0.0259566, 0.0592074, 0.133425, 0.283439, 0.477805\}$	208.14
1000	$\boldsymbol{\omega} = \{352738., 139235., 36360.9, 8460.51, 1915.46, 431.11, 97.0482, 21.9966, 5.14503, 1.37198, 0.567577\}$ $\boldsymbol{\beta} = \{0.000245944, 0.000427475, 0.000923201, 0.00209793, 0.00481594, 0.0110759, 0.025471, 0.0584454, 0.132524, 0.283412, 0.480561\}$	216.69

Table A.15
Parameters for optimized $P = 11$ SRJ schemes for various values of N ($N = 2^k\, k = 5, \ldots, 12$).

N	Optimal Scheme Parameters	ρ
32	$\boldsymbol{\omega} = \{396.431, 279.642, 153.434, 73.1608, 32.6594, 14.2231, 6.19822, 2.7735, 1.32935, 0.735639, 0.523157\}$ $\boldsymbol{\beta} = \{0.00766724, 0.00932051, 0.0129802, 0.0194268, 0.0300202, 0.0469263, 0.0733318, 0.113197, 0.169003, 0.234379, 0.283748\}$	12.02
64	$\boldsymbol{\omega} = \{1561.74, 995.92, 472.015, 193.505, 74.7492, 28.3009, 10.7213, 4.15116, 1.71187, 0.820855, 0.530347\}$ $\boldsymbol{\beta} = \{0.00404857, 0.00522535, 0.00791843, 0.012907, 0.0216291, 0.0365655, 0.0618008, 0.103457, 0.168173, 0.2534, 0.324875\}$	22.17
128	$\boldsymbol{\omega} = \{6143.1, 3508.98, 1431.82, 506.33, 170.137, 56.292, 18.6338, 6.27437, 2.23256, 0.92445, 0.538451\}$ $\boldsymbol{\beta} = \{0.00207453, 0.00286193, 0.00473326, 0.00839676, 0.0152373, 0.0278259, 0.0508049, 0.0921114, 0.162945, 0.267575, 0.365434\}$	40.31
256	$\boldsymbol{\omega} = \{24118.5, 12241.2, 4294.93, 1314.76, 385.775, 111.904, 32.4724, 9.54483, 2.93972, 1.04945, 0.547444\}$ $\boldsymbol{\beta} = \{0.00103495, 0.00153545, 0.00277641, 0.00535641, 0.0105195, 0.0207491, 0.0409204, 0.0803054, 0.154511, 0.277065, 0.405226\}$	72.17
512	$\boldsymbol{\omega} = \{94506.1, 42326.5, 12772.1, 3395.79, 872.457, 222.311, 56.6621, 14.5787, 3.89841, 1.19943, 0.557292\}$ $\boldsymbol{\beta} = \{0.000503804, 0.000808234, 0.00159933, 0.00335382, 0.00712881, 0.0151963, 0.0323902, 0.0688089, 0.143937, 0.282196, 0.444078\}$	127.22
1024	$\boldsymbol{\omega} = \{369575., 145216., 37730.4, 8738.16, 1969.52, 441.339, 98.9161, 22.3201, 5.19622, 1.37865, 0.567955\}$ $\boldsymbol{\beta} = \{0.000239683, 0.00041781, 0.000905131, 0.00206274, 0.00474813, 0.0109496, 0.0252489, 0.058095, 0.132107, 0.283393, 0.481833\}$	220.76
2048	$\boldsymbol{\omega} = \{1442350, 494839., 110897., 22426.1, 4440.04, 875.522, 172.665, 34.2146, 6.95091, 1.59211, 0.579383\}$ $\boldsymbol{\beta} = \{0.00011157, 0.000212226, 0.000503422, 0.001247, 0.00311134, 0.00777184, 0.0194126, 0.0484209, 0.119719, 0.281141, 0.518349\}$	376.94
4096	$\boldsymbol{\omega} = \{5617760, 1676280, 324683., 57443.8, 9998.5, 1735.51, 301.272, 52.4724, 9.32049, 1.84572, 0.591523\}$ $\boldsymbol{\beta} = \{0.0000508577, 0.000105956, 0.00027523, 0.000741393, 0.00200742, 0.00543916, 0.0147371, 0.0398943, 0.1073, 0.275945, 0.553504\}$	633.15

Table A.16
Parameters for optimized $P = 12$ SRJ schemes for various values of N ($N = 100 + 50k$, $k = 0, \ldots, 18$).

N	Optimal Scheme Parameters	ρ
100	$\boldsymbol{\omega} = \{3816.09, 2449.48, 1171.54, 484.552, 188.591, 71.7552, 27.1524, 10.3398, 4.03462, 1.67921, 0.814946, 0.530037\}$ $\boldsymbol{\beta} = \{0.00249324, 0.0032053, 0.00483143, 0.0078347, 0.0130695, 0.0220179, 0.0371777, 0.0626105, 0.104333, 0.168776, 0.25022, 0.32343\}$	33.60
150	$\boldsymbol{\omega} = \{8512.57, 5158.9, 2278.59, 870.502, 314.372, 111.324, 39.2426, 13.9067, 5.03399, 1.93522, 0.867388, 0.534161\}$ $\boldsymbol{\beta} = \{0.00168233, 0.00223889, 0.00353484, 0.00599785, 0.0104405, 0.0183241, 0.0322191, 0.0565309, 0.0983065, 0.166362, 0.259643, 0.344719\}$	47.75
200	$\boldsymbol{\omega} = \{15036.2, 8734.56, 3644.21, 1316.62, 451.186, 151.917, 50.9495, 17.1663, 5.89563, 2.1434, 0.907955, 0.537264\}$ $\boldsymbol{\beta} = \{0.0012661, 0.00172882, 0.0028221, 0.00494455, 0.00886996, 0.0160265, 0.0290014, 0.0523866, 0.0938883, 0.164052, 0.265144, 0.35987\}$	61.09
250	$\boldsymbol{\omega} = \{23372.3, 13127.2, 5239.13, 1813.2, 596.796, 193.289, 62.3814, 20.2163, 6.66795, 2.32215, 0.941482, 0.539767\}$ $\boldsymbol{\beta} = \{0.00101274, 0.00141158, 0.00236512, 0.00424802, 0.0078001, 0.0144149, 0.0266745, 0.0492834, 0.0904131, 0.161946, 0.268825, 0.371606\}$	73.83
300	$\boldsymbol{\omega} = \{33509.2, 18300.1, 7043.04, 2353.87, 749.797, 235.292, 73.5996, 23.1092, 7.37598, 2.48054, 0.97028, 0.541873\}$ $\boldsymbol{\beta} = \{0.000842345, 0.00119444, 0.00204463, 0.00374746, 0.00701309, 0.0132018, 0.0248808, 0.0468262, 0.0875581, 0.160039, 0.271482, 0.381171\}$	86.10
350	$\boldsymbol{\omega} = \{45437.3, 24224.4, 9040.67, 2934.05, 909.215, 277.826, 84.6431, 25.8779, 8.03469, 2.62379, 0.995654, 0.543695\}$ $\boldsymbol{\beta} = \{0.000719967, 0.00103614, 0.00180618, 0.00336745, 0.00640404, 0.012245, 0.0234383, 0.0448065, 0.0851418, 0.158305, 0.273495, 0.389234\}$	97.97
400	$\boldsymbol{\omega} = \{59148.4, 30876.1, 11220., 3550.26, 1074.33, 320.818, 95.5393, 28.5444, 8.65395, 2.75524, 1.01842, 0.545305\}$ $\boldsymbol{\beta} = \{0.000627865, 0.000915427, 0.00162115, 0.00306749, 0.00591535, 0.0114649, 0.0222425, 0.0431011, 0.0830517, 0.156721, 0.275074, 0.396198\}$	109.52
450	$\boldsymbol{\omega} = \{74635.1, 38235.1, 13571.2, 4199.71, 1244.59, 364.215, 106.308, 31.1249, 9.24068, 2.87718, 1.03912, 0.546747\}$ $\boldsymbol{\beta} = \{0.000556078, 0.000820244, 0.001473, 0.00282369, 0.00551249, 0.0108128, 0.0212283, 0.0416316, 0.0812134, 0.155264, 0.276343, 0.402322\}$	120.77
500	$\boldsymbol{\omega} = \{91891., 46283.8, 16086.1, 4880.15, 1419.54, 407.973, 116.965, 33.6311, 9.79994, 2.99126, 1.05814, 0.548056\}$ $\boldsymbol{\beta} = \{0.00049858, 0.000743205, 0.00135144, 0.002621, 0.00517332, 0.0102569, 0.0203527, 0.0403451, 0.0795753, 0.153916, 0.277382, 0.407785\}$	131.77
550	$\boldsymbol{\omega} = \{110910., 55006.9, 18757.6, 5589.69, 1598.81, 452.059, 127.522, 36.0724, 10.3356, 3.09868, 1.07577, 0.549254\}$ $\boldsymbol{\beta} = \{0.000451511, 0.000679537, 0.00124974, 0.00244939, 0.0048829, 0.00977557, 0.0195859, 0.0392045, 0.0780998, 0.152663, 0.278245, 0.412713\}$	142.53
600	$\boldsymbol{\omega} = \{131688., 64390.8, 21579.6, 6326.71, 1782.12, 496.444, 137.99, 38.4561, 10.8506, 3.20041, 1.09222, 0.55036\}$ $\boldsymbol{\beta} = \{0.000412285, 0.000626012, 0.00116328, 0.00230192, 0.00463078, 0.0093535, 0.0189066, 0.0381824, 0.076759, 0.151492, 0.278972, 0.4172\}$	153.09
650	$\boldsymbol{\omega} = \{154218., 74423.2, 24546.8, 7089.83, 1969.18, 541.105, 148.375, 40.7882, 11.3475, 3.29717, 1.10766, 0.551388\}$ $\boldsymbol{\beta} = \{0.000379105, 0.000580369, 0.00108879, 0.00217362, 0.00440935, 0.0089794, 0.0182987, 0.0372586, 0.0755316, 0.150394, 0.279588, 0.421318\}$	163.47
700	$\boldsymbol{\omega} = \{178498., 85093., 27654.3, 7877.82, 2159.79, 586.021, 158.686, 43.0737, 11.8282, 3.38957, 1.12222, 0.552348\}$ $\boldsymbol{\beta} = \{0.000350682, 0.000540974, 0.00102389, 0.0020608, 0.00421299, 0.00864481, 0.0177503, 0.0364172, 0.0744008, 0.149361, 0.280116, 0.425121\}$	173.67
750	$\boldsymbol{\omega} = \{204521., 96390., 30898., 8689.62, 2353.73, 631.175, 168.927, 45.3167, 12.2943, 3.4781, 1.13601, 0.553249\}$ $\boldsymbol{\beta} = \{0.00032607, 0.000506619, 0.000966783, 0.00196071, 0.00403737, 0.00834321, 0.017252, 0.035646, 0.0733534, 0.148384, 0.28057, 0.428654\}$	183.70
800	$\boldsymbol{\omega} = \{232286., 108305., 34273.9, 9524.26, 2550.85, 676.553, 179.104, 47.5208, 12.7472, 3.56315, 1.14911, 0.554098\}$ $\boldsymbol{\beta} = \{0.000304555, 0.000476389, 0.00091612, 0.00187122, 0.00387917, 0.00806952, 0.0167964, 0.0349352, 0.0723784, 0.147459, 0.280963, 0.431951\}$	193.59
850	$\boldsymbol{\omega} = \{261787., 120829., 37778.5, 10380.9, 2750.99, 722.141, 189.221, 49.689, 13.188, 3.64508, 1.16161, 0.554902\}$ $\boldsymbol{\beta} = \{0.000285593, 0.000449578, 0.00087084, 0.00179064, 0.00373575, 0.00781967, 0.0163775, 0.0342768, 0.0714671, 0.14658, 0.281305, 0.435042\}$	203.34
900	$\boldsymbol{\omega} = \{293021., 133954., 41408.7, 11258.7, 2954.01, 767.927, 199.282, 51.824, 13.6178, 3.72416, 1.17355, 0.555664\}$ $\boldsymbol{\beta} = \{0.000268759, 0.000425635, 0.000830107, 0.00171765, 0.00360498, 0.0075904, 0.0159906, 0.0336644, 0.0706121, 0.145743, 0.281603, 0.43795\}$	212.97
950	$\boldsymbol{\omega} = \{325986., 147673., 45161.3, 12157., 3159.78, 813.901, 209.291, 53.928, 14.0374, 3.80065, 1.185, 0.556389\}$ $\boldsymbol{\beta} = \{0.000253717, 0.000404119, 0.000793252, 0.00165119, 0.00348515, 0.00737903, 0.0156317, 0.0330924, 0.0698072, 0.144944, 0.281863, 0.440695\}$	222.47
1000	$\boldsymbol{\omega} = \{360677., 161978., 49033.7, 13075.1, 3368.2, 860.054, 219.25, 56.0032, 14.4476, 3.87476, 1.19599, 0.557081\}$ $\boldsymbol{\beta} = \{0.000240198, 0.000384678, 0.000759733, 0.00159036, 0.00337486, 0.00718335, 0.0152974, 0.0325564, 0.0690473, 0.14418, 0.28209, 0.443295\}$	231.85

Table A.17
Parameters for optimized $P = 12$ SRJ schemes for various values of N ($N = 2^k\, k = 5, \ldots, 12$).

N	Optimal Scheme Parameters	ρ
32	$\boldsymbol{\omega} = \{399.282, 296.323, 175.654, 90.9514, 43.9229, 20.5641, 9.54942, 4.47911, 2.17305, 1.13535, 0.692165, 0.520193\}$ $\boldsymbol{\beta} = \{0.00717978, 0.00847231, 0.0112881, 0.0161292, 0.0238518, 0.0357931, 0.0539181, 0.0808575, 0.119387, 0.170519, 0.20623, 0.266374\}$	12.22
64	$\boldsymbol{\omega} = \{1576.99, 1074.79, 560.296, 253.075, 107.095, 44.1519, 18.083, 7.46292, 3.16734, 1.43996, 0.763014, 0.525858\}$ $\boldsymbol{\beta} = \{0.00380547, 0.0047207, 0.00676987, 0.0104438, 0.0166204, 0.0267627, 0.0432181, 0.0695623, 0.110575, 0.170246, 0.236935, 0.300341\}$	22.71
128	$\boldsymbol{\omega} = \{6219.99, 3856.45, 1757.41, 692.507, 257.487, 93.7674, 33.9796, 12.3838, 4.61567, 1.8302, 0.846238, 0.532512\}$ $\boldsymbol{\beta} = \{0.0019642, 0.00257829, 0.00399715, 0.00666322, 0.0114074, 0.0197032, 0.0340994, 0.0588784, 0.100697, 0.167428, 0.256209, 0.336374\}$	41.64
256	$\boldsymbol{\omega} = \{24493.8, 13707.4, 5444.91, 1875.85, 614.788, 198.299, 63.738, 20.5709, 6.75595, 2.34211, 0.945155, 0.540038\}$ $\boldsymbol{\beta} = \{0.000988845, 0.00138134, 0.0023209, 0.00417958, 0.00769344, 0.0142519, 0.0264357, 0.0489597, 0.0900422, 0.161707, 0.269188, 0.372852\}$	75.32
512	$\boldsymbol{\omega} = \{96295., 48316.3, 16713.2, 5047.85, 1462.18, 418.525, 119.508, 34.2226, 9.93052, 3.0176, 1.06249, 0.548353\}$ $\boldsymbol{\beta} = \{0.000486446, 0.000726847, 0.00132543, 0.00257728, 0.00509962, 0.0101352, 0.0201597, 0.0400592, 0.0792075, 0.153607, 0.277603, 0.409012\}$	134.37
1024	$\boldsymbol{\omega} = \{377942., 169051., 50934.3, 13522.7, 3469.15, 882.268, 224.013, 56.9895, 14.6414, 3.90954, 1.20112, 0.557403\}$ $\boldsymbol{\beta} = \{0.000234184, 0.000375997, 0.000744695, 0.00156296, 0.00332495, 0.00709444, 0.0151449, 0.0323108, 0.0686972, 0.143825, 0.282189, 0.444496\}$	236.32
2048	$\boldsymbol{\omega} = \{1480870, 587626., 154364., 36112.1, 8217.37, 1858.2, 419.746, 94.9399, 21.6287, 5.08649, 1.36434, 0.567144\}$ $\boldsymbol{\beta} = \{0.000110457, 0.00019134, 0.000411749, 0.000932607, 0.00213412, 0.00489319, 0.0112226, 0.0257294, 0.0588522, 0.133007, 0.283417, 0.479098\}$	409.64
4096	$\boldsymbol{\omega} = \{5792620, 2030850, 465804., 96218.1, 19441.1, 3910.7, 786.112, 158.157, 31.985, 6.63805, 1.55596, 0.577534\}$ $\boldsymbol{\beta} = \{0.0000510879, 0.0000958256, 0.000224079, 0.000547792, 0.00134943, 0.0033284, 0.00821094, 0.0202514, 0.0498712, 0.121686, 0.281701, 0.512683\}$	699.67

Table A.18
Parameters for optimized $P = 13$ SRJ schemes for various values of N ($N = 100 + 50k$, $k = 0, \dots, 18$).

N	Optimal Scheme Parameters	ρ
100	$\boldsymbol{\omega} = \{3850.42, 2628.04, 1372.85, 621.295, 263.269, 108.54, 44.3191, 18.1012, 7.47022, 3.1732, 1.46178, 0.770791, 0.52618\}$ $\boldsymbol{\beta} = \{0.00235853, 0.00292322, 0.00418701, 0.00645168, 0.0102578, 0.0165109, 0.0266858, 0.0431288, 0.0694093, 0.110458, 0.161845, 0.243877, 0.301907\}$	34.43
150	$\boldsymbol{\omega} = \{8600.11, 5583., 2713.17, 1140.26, 450.396, 173.646, 66.4069, 25.4013, 9.80185, 3.88069, 1.65106, 0.810599, 0.529505\}$ $\boldsymbol{\beta} = \{0.0015982, 0.00204078, 0.00304774, 0.00489728, 0.00810045, 0.0135406, 0.0227101, 0.0380865, 0.06365, 0.105277, 0.164788, 0.25184, 0.320423\}$	49.15
200	$\boldsymbol{\omega} = \{15205.3, 9511.83, 4388.27, 1750.36, 658.133, 242.067, 88.403, 32.2912, 11.8866, 4.4795, 1.80488, 0.84212, 0.532062\}$ $\boldsymbol{\beta} = \{0.00120674, 0.00157554, 0.00242509, 0.00401446, 0.00682843, 0.0117242, 0.0201874, 0.0347575, 0.059662, 0.101442, 0.165082, 0.257141, 0.333953\}$	63.10
250	$\boldsymbol{\omega} = \{23653.2, 14365.9, 6363.85, 2438.1, 882.631, 313.079, 110.345, 38.8978, 13.808, 5.00982, 1.93667, 0.868438, 0.534148\}$ $\boldsymbol{\beta} = \{0.000967836, 0.00128633, 0.00202738, 0.00343434, 0.00596933, 0.0104642, 0.0183895, 0.0323153, 0.0566335, 0.0983815, 0.164575, 0.260976, 0.34458\}$	76.48
300	$\boldsymbol{\omega} = \{33933.9, 20108.6, 8615.78, 3194.46, 1121.42, 386.232, 132.247, 45.289, 15.6094, 5.4916, 2.05313, 0.891162, 0.535915\}$ $\boldsymbol{\beta} = \{0.000806822, 0.00108844, 0.00174925, 0.00301942, 0.00534147, 0.00952376, 0.0170188, 0.0304106, 0.0542077, 0.0958339, 0.163779, 0.263906, 0.353314\}$	89.41
350	$\boldsymbol{\omega} = \{46039., 26711., 11125.8, 4012.89, 1372.77, 461.214, 154.117, 51.5071, 17.3169, 5.93653, 2.15818, 0.911239, 0.537452\}$ $\boldsymbol{\beta} = \{0.000690964, 0.000944197, 0.00154278, 0.00270562, 0.00485811, 0.00878712, 0.0159262, 0.0288638, 0.0521946, 0.0936541, 0.162881, 0.266231, 0.36072\}$	101.97
400	$\boldsymbol{\omega} = \{59961.3, 34148.9, 13879.2, 4888.34, 1635.37, 537.796, 175.961, 57.581, 18.9481, 6.35222, 2.25433, 0.929279, 0.538812\}$ $\boldsymbol{\beta} = \{0.000603625, 0.000834233, 0.00138288, 0.00245871, 0.00447198, 0.00818994, 0.0150272, 0.0275709, 0.050481, 0.091751, 0.161958, 0.268127, 0.367144\}$	114.20
450	$\boldsymbol{\omega} = \{75694.4, 42401.8, 16864.2, 5816.77, 1908.21, 615.806, 197.782, 63.5318, 20.5156, 6.74396, 2.34329, 0.945698, 0.540035\}$ $\boldsymbol{\beta} = \{0.000535451, 0.000747535, 0.00125506, 0.00225859, 0.00415487, 0.00769319, 0.0142696, 0.0264665, 0.0489941, 0.0900639, 0.161042, 0.269708, 0.372811\}$	126.15
500	$\boldsymbol{\omega} = \{93232.9, 51452., 20070.7, 6794.87, 2190.47, 695.105, 219.583, 69.3753, 22.0284, 7.11557, 2.42632, 0.960792, 0.541147\}$ $\boldsymbol{\beta} = \{0.000480773, 0.000677371, 0.00115032, 0.0020926, 0.00388878, 0.0072716, 0.0136193, 0.025507, 0.0476845, 0.0885501, 0.16015, 0.271048, 0.377879\}$	137.85
550	$\boldsymbol{\omega} = \{112572., 61283.9, 23490., 7819.84, 2481.47, 775.581, 241.365, 75.124, 23.4937, 7.46998, 2.50434, 0.974781, 0.542167\}$ $\boldsymbol{\beta} = \{0.000435959, 0.000619389, 0.00106281, 0.00195237, 0.00366162, 0.00690802, 0.0130527, 0.024662, 0.0465171, 0.0871783, 0.159289, 0.2722, 0.382461\}$	149.33
600	$\boldsymbol{\omega} = \{133706., 71883.4, 27114.6, 8889.31, 2780.65, 857.141, 263.131, 80.7878, 24.9172, 7.80944, 2.57806, 0.987834, 0.543109\}$ $\boldsymbol{\beta} = \{0.00039857, 0.000570647, 0.000988491, 0.00183209, 0.00346494, 0.00659031, 0.012553, 0.0239094, 0.0454659, 0.085925, 0.158461, 0.273201, 0.38664\}$	160.61
650	$\boldsymbol{\omega} = \{156632., 83238., 30937.8, 10001.2, 3087.51, 939.707, 284.882, 86.3749, 26.3034, 8.13575, 2.64805, 1.00008, 0.543985\}$ $\boldsymbol{\beta} = \{0.000366911, 0.000529085, 0.000924528, 0.00172762, 0.00329263, 0.00630962, 0.0121077, 0.0232328, 0.0445115, 0.0847721, 0.157665, 0.274079, 0.390481\}$	171.71
700	$\boldsymbol{\omega} = \{181345., 95336.2, 34953.7, 11153.7, 3401.64, 1023.21, 306.618, 91.8922, 27.6561, 8.45039, 2.71475, 1.01163, 0.544804\}$ $\boldsymbol{\beta} = \{0.000339764, 0.000493215, 0.000868846, 0.00163592, 0.00314016, 0.00605932, 0.0117075, 0.0226198, 0.0436389, 0.0837054, 0.156902, 0.274856, 0.394033\}$	182.64
750	$\boldsymbol{\omega} = \{207842., 108167., 39157., 12345.1, 3722.65, 1107.6, 328.341, 97.3453, 28.9783, 8.75456, 2.77854, 1.02256, 0.545574\}$ $\boldsymbol{\beta} = \{0.000316234, 0.000461934, 0.000819899, 0.00155467, 0.00300409, 0.00583432, 0.0113452, 0.0220606, 0.042836, 0.0827132, 0.15617, 0.275548, 0.397336\}$	193.41
800	$\boldsymbol{\omega} = \{236120., 121722., 43542.7, 13574.1, 4050.24, 1192.81, 350.051, 102.739, 30.2727, 9.04927, 2.83973, 1.03294, 0.5463\}$ $\boldsymbol{\beta} = \{0.000295649, 0.00043441, 0.000776506, 0.00148213, 0.00288174, 0.00563066, 0.011015, 0.0215473, 0.0420935, 0.0817863, 0.155467, 0.276167, 0.400423\}$	204.03
850	$\boldsymbol{\omega} = \{266174., 135991., 48106.5, 14839.3, 4384.1, 1278.81, 371.75, 108.078, 31.5415, 9.33538, 2.89858, 1.04284, 0.546987\}$ $\boldsymbol{\beta} = \{0.000277491, 0.000409999, 0.000737751, 0.00141689, 0.00277102, 0.00544519, 0.0107124, 0.0210739, 0.0414035, 0.0809169, 0.154791, 0.276725, 0.403319\}$	214.53
900	$\boldsymbol{\omega} = \{298003., 150967., 52844.3, 16139.5, 4723.96, 1365.56, 393.437, 113.366, 32.7868, 9.61362, 2.95529, 1.0523, 0.54764\}$ $\boldsymbol{\beta} = \{0.000261358, 0.000388199, 0.000702912, 0.00135787, 0.00267024, 0.00527538, 0.0104337, 0.0206352, 0.0407597, 0.0800987, 0.154141, 0.27723, 0.406046\}$	224.89
950	$\boldsymbol{\omega} = \{331603., 166640., 57752.4, 17473.7, 5069.61, 1453.01, 415.114, 118.606, 34.0103, 9.88464, 3.01007, 1.06136, 0.548261\}$ $\boldsymbol{\beta} = \{0.000246931, 0.00036861, 0.000671408, 0.00130418, 0.00257804, 0.00511916, 0.0101759, 0.0202269, 0.0401568, 0.0793262, 0.153514, 0.277689, 0.408623\}$	235.13
1000	$\boldsymbol{\omega} = \{366972., 183005., 62827.2, 18840.9, 5420.8, 1541.14, 436.781, 123.801, 35.2134, 10.149, 3.06307, 1.07006, 0.548854\}$ $\boldsymbol{\beta} = \{0.000233955, 0.000350909, 0.000642773, 0.0012551, 0.0024933, 0.00497482, 0.00993636, 0.0198456, 0.0395904, 0.0785949, 0.152911, 0.278107, 0.411065\}$	245.27

Table A.19
Parameters for optimized $P = 13$ SRJ schemes for various values of N ($N = 2^k\, k = 5, \ldots, 12$).

N	Optimal Scheme Parameters	ρ
32	$\boldsymbol{\omega} = \{401.471, 309.92, 195.459, 108.339, 55.9022, 27.8438, 13.6785, 6.72758, 3.36452, 1.75033, 1.05207, 0.686717, 0.518762\}$ $\boldsymbol{\beta} = \{0.00675791, 0.0077945, 0.0100255, 0.0137902, 0.0196578, 0.0285024, 0.0415987, 0.0607023, 0.088016, 0.126799, 0.117918, 0.222136, 0.256303\}$	12.38
64	$\boldsymbol{\omega} = \{1589.04, 1142.32, 644.729, 316.512, 145.007, 64.4121, 28.2867, 12.4267, 5.52727, 2.53974, 1.28491, 0.733286, 0.522941\}$ $\boldsymbol{\beta} = \{0.00358591, 0.00431074, 0.00590608, 0.00869324, 0.0132327, 0.0204319, 0.0317095, 0.0492101, 0.0759855, 0.11602, 0.153206, 0.23494, 0.282768\}$	23.16
128	$\boldsymbol{\omega} = \{6280.77, 4159.32, 2079.6, 899.633, 365.213, 144.533, 56.7035, 22.2513, 8.81404, 3.58658, 1.57341, 0.794381, 0.528164\}$ $\boldsymbol{\beta} = \{0.00186278, 0.00235054, 0.00345302, 0.00545819, 0.00888982, 0.0146421, 0.0242045, 0.0400093, 0.0658841, 0.107333, 0.16405, 0.248777, 0.313085\}$	42.78
256	$\boldsymbol{\omega} = \{24790.3, 15008.7, 6619.9, 2525.38, 910.571, 321.752, 112.975, 39.6752, 14.0299, 5.06992, 1.95136, 0.871332, 0.534375\}$ $\boldsymbol{\beta} = \{0.000945278, 0.00125876, 0.00198896, 0.00337751, 0.00588404, 0.0103375, 0.0182063, 0.0320629, 0.0563154, 0.0980522, 0.164489, 0.261368, 0.345714\}$	78.06
512	$\boldsymbol{\omega} = \{97710.2, 53741., 20872.2, 7036.69, 2259.54, 714.315, 224.812, 70.7632, 22.3842, 7.2021, 2.44547, 0.964243, 0.5414\}$ $\boldsymbol{\beta} = \{0.000469226, 0.000662473, 0.00112793, 0.00205685, 0.00383108, 0.00717958, 0.0134764, 0.0252947, 0.0473925, 0.0882089, 0.159941, 0.27134, 0.379019\}$	140.62
1024	$\boldsymbol{\omega} = \{384576., 191103., 65321.5, 19508.6, 5591.29, 1583.67, 447.177, 126.28, 35.7841, 10.2736, 3.08792, 1.07412, 0.549129\}$ $\boldsymbol{\beta} = \{0.000228181, 0.000343005, 0.000629932, 0.001233, 0.002455, 0.00490933, 0.00982729, 0.0196712, 0.0393302, 0.0782572, 0.152629, 0.278294, 0.412193\}$	250.09
2048	$\boldsymbol{\omega} = \{1511490, 675363., 203229., 53892.9, 13810.2, 3507.87, 889.156, 225.393, 57.2711, 14.697, 3.92023, 1.20273, 0.557498\}$ $\boldsymbol{\beta} = \{0.000108834, 0.000174858, 0.000346582, 0.000727906, 0.00154954, 0.00330852, 0.00706863, 0.0151018, 0.0322407, 0.0685947, 0.143668, 0.282261, 0.444849\}$	438.96
4096	$\boldsymbol{\omega} = \{5932130, 2373560, 629341., 148493., 34065.2, 7764.3, 1767.21, 402.245, 91.7049, 21.0624, 4.99621, 1.35246, 0.566463\}$ $\boldsymbol{\beta} = \{0.0000509606, 0.0000878108, 0.000187895, 0.000423376, 0.000964004, 0.00219946, 0.00502014, 0.011458, 0.0261403, 0.0594939, 0.133745, 0.283445, 0.476784\}$	760.22

Table A.20
Parameters for optimized $P = 14$ SRJ schemes for various values of N ($N = 100 + 50k$, $k = 0, \ldots, 18$).

N	Optimal Scheme Parameters	ρ
100	$\boldsymbol{\omega}$ = {3878.55, 2785.67, 1570.02, 769.498, 351.868, 155.898, 68.1753, 29.7098, 12.9941, 5.75755, 2.75287, 1.36341, 0.743467, 0.523835} $\boldsymbol{\beta}$ = {0.00223066, 0.00268294, 0.00367875, 0.00541949, 0.00825729, 0.012765, 0.019848, 0.0309124, 0.0481036, 0.0748715, 0.0959847, 0.170428, 0.2368, 0.288017}	35.12
150	$\boldsymbol{\omega}$ = {8671.64, 5959.56, 3144.72, 1438.03, 615.285, 255.886, 105.238, 43.1569, 17.7519, 7.38476, 3.24126, 1.48992, 0.772445, 0.526323} $\boldsymbol{\beta}$ = {0.00151803, 0.00187383, 0.00266836, 0.00408745, 0.00646311, 0.0103505, 0.0166552, 0.0268341, 0.0431958, 0.0694114, 0.100573, 0.169923, 0.243817, 0.302629}	50.34
200	$\boldsymbol{\omega}$ = {15342.9, 10203.7, 5132.76, 2234.12, 912.112, 362.768, 142.862, 56.1217, 22.1054, 8.79573, 3.65467, 1.59758, 0.796262, 0.528322} $\boldsymbol{\beta}$ = {0.00115001, 0.00144748, 0.00211891, 0.00333763, 0.00541838, 0.00889808, 0.0146726, 0.0242198, 0.0399456, 0.0657302, 0.101006, 0.169306, 0.248859, 0.313891}	64.84
250	$\boldsymbol{\omega}$ = {23881.3, 15470.9, 7496.09, 3140.75, 1236.83, 475.305, 181.025, 68.7946, 26.2065, 10.0762, 4.02155, 1.69204, 0.816583, 0.529997} $\boldsymbol{\beta}$ = {0.000924757, 0.00118231, 0.00176872, 0.00284695, 0.00471686, 0.00789814, 0.0132733, 0.0223266, 0.0375256, 0.0629164, 0.100286, 0.168614, 0.252701, 0.323018}	78.79
300	$\boldsymbol{\omega}$ = {34278.1, 21725.7, 10207.4, 4146.18, 1585.72, 592.623, 219.656, 81.2544, 30.1232, 11.2643, 4.35503, 1.7766, 0.834356, 0.531439} $\boldsymbol{\beta}$ = {0.000772581, 0.00100079, 0.00152417, 0.002497, 0.00420622, 0.00715567, 0.0122136, 0.020863, 0.0356119, 0.0606376, 0.0992048, 0.167894, 0.25575, 0.330669}	92.3
350	$\boldsymbol{\omega}$ = {46526., 28939.3, 13245.9, 5241.76, 1956.08, 714.096, 258.693, 93.545, 33.8951, 12.3814, 4.66283, 1.85345, 0.850189, 0.532707} $\boldsymbol{\beta}$ = {0.000662864, 0.000868418, 0.00134284, 0.00223296, 0.00381437, 0.00657652, 0.0113733, 0.0196825, 0.0340395, 0.0587264, 0.0980248, 0.16717, 0.258242, 0.337243}	105.45
400	$\boldsymbol{\omega}$ = {60618.7, 37088., 16594.7, 6420.75, 2345.86, 839.25, 298.085, 105.695, 37.5482, 13.4414, 4.95006, 1.92411, 0.864496, 0.533839} $\boldsymbol{\beta}$ = {0.000580015, 0.000767473, 0.00120256, 0.00202561, 0.00350221, 0.00610869, 0.010685, 0.0187016, 0.0327121, 0.0570842, 0.0968455, 0.166458, 0.260328, 0.343}	118.29
450	$\boldsymbol{\omega}$ = {76550.6, 46151.6, 20240.1, 7677.7, 2753.45, 967.72, 337.793, 117.724, 41.1009, 14.4541, 5.22033, 1.98968, 0.877567, 0.534862} $\boldsymbol{\beta}$ = {0.000515249, 0.000687865, 0.00109052, 0.00185784, 0.00324648, 0.00572076, 0.0101073, 0.017868, 0.0315686, 0.0556475, 0.0957059, 0.165764, 0.262105, 0.348115}	130.87
500	$\boldsymbol{\omega}$ = {94317., 56112.2, 24170.1, 9008.13, 3177.53, 1099.21, 377.785, 129.647, 44.5668, 15.4266, 5.47629, 2.05098, 0.889618, 0.535797} $\boldsymbol{\beta}$ = {0.000463236, 0.000623426, 0.000998804, 0.00171891, 0.00303236, 0.00539248, 0.00961327, 0.0171471, 0.0305679, 0.054373, 0.0946203, 0.165092, 0.263642, 0.352715}	143.20
550	$\boldsymbol{\omega}$ = {113913., 66954.2, 28374.8, 10408.2, 3617.01, 1233.48, 418.035, 141.476, 47.9566, 16.3643, 5.71997, 2.10865, 0.900811, 0.536657} $\boldsymbol{\beta}$ = {0.000420555, 0.000570165, 0.00092222, 0.0016017, 0.00284993, 0.00511011, 0.00918421, 0.0165148, 0.0296809, 0.0532295, 0.0935924, 0.164444, 0.264987, 0.356893}	155.31
600	$\boldsymbol{\omega}$ = {135336., 78663.8, 32845.3, 11874.7, 4070.95, 1370.32, 458.522, 153.221, 51.2786, 17.2714, 5.95295, 2.16317, 0.91127, 0.537454} $\boldsymbol{\beta}$ = {0.000384908, 0.000525385, 0.000857231, 0.0015013, 0.00269227, 0.00486396, 0.00880696, 0.015954, 0.0288864, 0.0521941, 0.0926211, 0.163818, 0.266176, 0.360718}	167.24
650	$\boldsymbol{\omega}$ = {158580., 91228.1, 37573.6, 13404.7, 4538.56, 1509.57, 499.227, 164.888, 54.5397, 18.1514, 6.17652, 2.21494, 0.921097, 0.538198} $\boldsymbol{\beta}$ = {0.000354693, 0.000487197, 0.000801333, 0.0014142, 0.00255438, 0.00464697, 0.00847177, 0.0154516, 0.0281686, 0.0512493, 0.0917036, 0.163215, 0.267237, 0.364244}	178.98
700	$\boldsymbol{\omega}$ = {183644., 104636., 42552.7, 14995.7, 5019.15, 1651.07, 540.135, 176.484, 57.7456, 19.0071, 6.39171, 2.26428, 0.930369, 0.538894} $\boldsymbol{\beta}$ = {0.000328761, 0.000454234, 0.000752702, 0.00133783, 0.00243256, 0.00445388, 0.00817133, 0.0149979, 0.0275151, 0.0503814, 0.090836, 0.162634, 0.268191, 0.367514}	190.56
750	$\boldsymbol{\omega}$ = {210523., 118876., 47776.4, 16645.5, 5512.09, 1794.7, 581.231, 188.016, 60.9012, 19.8408, 6.59939, 2.31147, 0.939153, 0.53955} $\boldsymbol{\beta}$ = {0.000306266, 0.000425485, 0.000709977, 0.00127025, 0.00232401, 0.00428065, 0.00789997, 0.0145854, 0.0269165, 0.0495796, 0.0900146, 0.162073, 0.269053, 0.370562}	201.99
800	$\boldsymbol{\omega}$ = {239214., 133939., 53238.8, 18352.1, 6016.86, 1940.35, 622.504, 199.487, 64.0103, 20.6544, 6.80028, 2.35671, 0.947503, 0.540169} $\boldsymbol{\beta}$ = {0.000286569, 0.000400187, 0.00067212, 0.00120995, 0.00222654, 0.00412413, 0.00765326, 0.0142079, 0.026365, 0.0488352, 0.0892357, 0.161532, 0.269837, 0.373415}	213.28
850	$\boldsymbol{\omega}$ = {269715., 149816., 58934.6, 20113.6, 6532.95, 2087.9, 663.944, 210.901, 67.0766, 21.4498, 6.99499, 2.40021, 0.955464, 0.540756} $\boldsymbol{\beta}$ = {0.000269181, 0.000377748, 0.000638326, 0.00115578, 0.00213845, 0.00398185, 0.00742767, 0.0138606, 0.0258544, 0.048141, 0.0884956, 0.161009, 0.270553, 0.376097}	224.44
900	$\boldsymbol{\omega}$ = {302024., 166499., 64858.9, 21928.4, 7059.93, 2237.28, 705.541, 222.262, 70.1032, 22.2283, 7.18405, 2.44212, 0.963074, 0.541314} $\boldsymbol{\beta}$ = {0.000253722, 0.000357707, 0.00060796, 0.00110682, 0.00205836, 0.00385178, 0.00722033, 0.0135396, 0.0253796, 0.0474912, 0.0877914, 0.160504, 0.271211, 0.378626}	235.47
950	$\boldsymbol{\omega}$ = {336137., 183978., 71007.3, 23794.9, 7597.41, 2388.4, 747.287, 233.574, 73.0926, 22.9911, 7.36793, 2.48257, 0.970367, 0.541846} $\boldsymbol{\beta}$ = {0.000239888, 0.000339698, 0.000580514, 0.00106231, 0.00198518, 0.00373232, 0.00702889, 0.0132417, 0.0249364, 0.0468808, 0.0871199, 0.160016, 0.271817, 0.38102}	246.38
1000	$\boldsymbol{\omega}$ = {372052., 202248., 77375.6, 25711.8, 8145.01, 2541.19, 789.175, 244.838, 76.0472, 23.7394, 7.54701, 2.52169, 0.97737, 0.542354} $\boldsymbol{\beta}$ = {0.000227438, 0.000323423, 0.000555578, 0.00102166, 0.001918, 0.0036221, 0.0068514, 0.0129641, 0.0245213, 0.0463056, 0.0864787, 0.159543, 0.272378, 0.38329}	257.19

Table A.21
Parameters for optimized $P = 14$ SRJ schemes for various values of N ($N = 2^k\, k = 5, \dots, 12$).

N	Optimal Scheme Parameters	ρ
32	$\boldsymbol{\omega} = \{402.8, 318.533, 208.846, 120.929, 65.1531, 33.8031, 17.2446, 8.7701, 4.50164, 2.36779, 1.58821, 1.06413, 0.669773, 0.517215\}$ $\boldsymbol{\beta} = \{0.00647086, 0.0073607, 0.00926188, 0.0124338, 0.0173072, 0.0245363, 0.035068, 0.0502174, 0.0717549, 0.104058, 0.0350767, 0.168979, 0.212666, 0.24481\}$	12.47
64	$\boldsymbol{\omega} = \{1598.72, 1200.1, 724.027, 381.957, 187.598, 88.992, 41.5807, 19.3493, 9.04533, 4.29382, 2.29786, 1.25078, 0.716642, 0.521476\}$ $\boldsymbol{\beta} = \{0.00338329, 0.0039658, 0.00523063, 0.00739484, 0.0108296, 0.0161218, 0.0241681, 0.0363079, 0.0545086, 0.0822675, 0.082198, 0.170812, 0.229322, 0.27349\}$	23.51
128	$\boldsymbol{\omega} = \{6330.52, 4427.82, 2398.11, 1126.86, 494.844, 210.953, 88.8667, 37.3202, 15.7238, 6.70433, 3.03854, 1.43707, 0.760478, 0.525303\}$ $\boldsymbol{\beta} = \{0.00176623, 0.00215768, 0.00302688, 0.00456605, 0.00711566, 0.0112388, 0.0178429, 0.0283674, 0.0450598, 0.0714887, 0.0994468, 0.170158, 0.241043, 0.296722\}$	43.75
256	$\boldsymbol{\omega} = \{25031., 16170., 7803.54, 3256.38, 1277.48, 489.145, 185.638, 70.2999, 26.6852, 10.2232, 4.06315, 1.70266, 0.818836, 0.530181\}$ $\boldsymbol{\beta} = \{0.000903457, 0.00115703, 0.00173491, 0.00279897, 0.00464739, 0.00779789, 0.0131313, 0.022132, 0.0372734, 0.0626189, 0.100168, 0.168529, 0.253103, 0.324005\}$	80.43
512	$\boldsymbol{\omega} = \{98853.4, 58634.4, 25154.6, 9337.93, 3281.64, 1131.19, 387.422, 132.494, 45.3869, 15.6547, 5.53583, 2.06513, 0.892377, 0.536009\}$ $\boldsymbol{\beta} = \{0.000452243, 0.000609742, 0.000979198, 0.00168901, 0.00298598, 0.00532095, 0.00950493, 0.016988, 0.0303456, 0.0540876, 0.0943683, 0.164935, 0.26398, 0.353754\}$	146.12
1024	$\boldsymbol{\omega} = \{389930., 211296., 80509.3, 26649.4, 8411.36, 2615.1, 809.329, 250.229, 77.4537, 24.0938, 7.63137, 2.54003, 0.980636, 0.542591\}$ $\boldsymbol{\beta} = \{0.000221894, 0.000316155, 0.000544398, 0.00100336, 0.00188766, 0.00357215, 0.00677068, 0.0128374, 0.0243311, 0.0460409, 0.086181, 0.159321, 0.272632, 0.38434\}$	262.34
2048	$\boldsymbol{\omega} = \{1536240, 757099., 256177., 75783.7, 21526.3, 6043.96, 1691.56, 473.123, 132.43, 37.1993, 10.5906, 3.1509, 1.08412, 0.549805\}$ $\boldsymbol{\beta} = \{0.000106867, 0.000161473, 0.000298341, 0.000587241, 0.00117546, 0.00236283, 0.00475453, 0.00956894, 0.0192541, 0.038699, 0.077294, 0.152063, 0.278729, 0.414945\}$	465.33
4096	$\boldsymbol{\omega} = \{6045072, 2698661, 811234., 214918., 55022.7, 13963.2, 3535.64, 894.898, 226.617, 57.5285, 14.7527, 3.93026, 1.20411, 0.557585\}$ $\boldsymbol{\beta} = \{0.0000505714, 0.0000812954, 0.000161234, 0.000338826, 0.000721677, 0.00154174, 0.00329580, 0.00704624, 0.0150625, 0.0321750, 0.0684563, 0.143616, 0.282283, 0.445170\}$	815.34

Table A.22
Parameters for optimized $P = 15$ SRJ schemes for various values of N ($N = 100 + 50k$, $k = 0, \ldots, 18$).

N	Optimal Scheme Parameters	ρ
100	ω = {3900.7, 2917.56, 1750.24, 917.535, 447.788, 211.033, 97.8831, 45.1243, 20.8015, 9.63811, 5.19063, 2.69523, 1.30463, 0.729676, 0.52263} β = {0.00211354, 0.00248261, 0.00328491, 0.00466027, 0.00684846, 0.0102307, 0.015395, 0.0232377, 0.0351381, 0.0539546, 0.0440768, 0.115928, 0.169266, 0.232834, 0.28055}	35.62
150	ω = {8731.9, 6299.82, 3574.97, 1764.89, 812.568, 362.251, 159.248, 69.6404, 30.449, 13.37, 6.43047, 3.027, 1.40229, 0.752546, 0.524622} β = {0.00143811, 0.00172513, 0.00235617, 0.00345693, 0.00524685, 0.00808252, 0.0125283, 0.0194675, 0.0302848, 0.0474906, 0.0542321, 0.11255, 0.169529, 0.238971, 0.29264}	51.31
200	ω = {15459.6, 10836.2, 5887.55, 2775.66, 1222.6, 522.574, 220.57, 92.6754, 38.9339, 16.42, 7.3844, 3.28605, 1.47625, 0.769405, 0.526065} β = {0.00109279, 0.0013333, 0.00186699, 0.0028111, 0.00437317, 0.00689647, 0.010935, 0.017374, 0.0276256, 0.0441322, 0.0577854, 0.110203, 0.169452, 0.243027, 0.301093}	66.29
250	ω = {24074.2, 16482.5, 8649.12, 3932.14, 1672.98, 691.895, 282.873, 115.18, 46.8951, 19.1619, 8.22785, 3.52027, 1.54157, 0.783992, 0.527297} β = {0.00088128, 0.00109027, 0.00155754, 0.00239368, 0.00379676, 0.006099, 0.00984518, 0.0159203, 0.0257581, 0.0418003, 0.0589403, 0.108288, 0.169245, 0.246258, 0.308126}	80.75
300	ω = {34568., 23204.9, 11831.2, 5221.34, 2159.63, 869.493, 346.353, 137.457, 54.5501, 21.7213, 9.00657, 3.73816, 1.60113, 0.797059, 0.528388} β = {0.000738052, 0.000923817, 0.00134184, 0.00209712, 0.00337956, 0.00551145, 0.00902828, 0.0148121, 0.0243112, 0.0399833, 0.0591017, 0.106635, 0.168961, 0.248953, 0.314222}	94.79
350	ω = {46934.7, 30975.9, 15411.5, 6633.06, 2679.17, 1054.58, 410.977, 159.617, 61.9913, 24.1524, 9.7385, 3.94279, 1.65611, 0.808933, 0.52937} β = {0.00063454, 0.000802316, 0.00118197, 0.00187368, 0.00306013, 0.00505457, 0.00838329, 0.0139235, 0.0231325, 0.0384872, 0.058814, 0.105172, 0.168631, 0.251248, 0.319602}	108.48
400	ω = {61168.7, 39772.5, 19371.5, 8159.09, 3228.87, 1246.47, 476.666, 181.708, 69.2673, 26.4843, 10.4333, 4.13611, 1.70724, 0.819823, 0.530261} β = {0.000556209, 0.000709559, 0.00105828, 0.00169833, 0.00280596, 0.0046861, 0.00785612, 0.0131873, 0.0221419, 0.0372151, 0.0583153, 0.103854, 0.168272, 0.253232, 0.324411}	121.87
450	ω = {77265.3, 49575., 23696., 9792.69, 3806.5, 1444.59, 543.333, 203.752, 76.408, 28.7354, 11.0971, 4.3196, 1.7551, 0.829887, 0.531078} β = {0.000494861, 0.000636339, 0.000959489, 0.00155652, 0.00259789, 0.0043809, 0.00741432, 0.0125628, 0.0212909, 0.0361099, 0.0577191, 0.102656, 0.167899, 0.25497, 0.328752}	135.00
500	ω = {95220., 60366.1, 28371.8, 11528.2, 4410.2, 1648.44, 610.9, 225.763, 83.4336, 30.9182, 11.7345, 4.49446, 1.80013, 0.839249, 0.531833} β = {0.000445513, 0.00057702, 0.000878606, 0.00143911, 0.00242379, 0.00412284, 0.00703687, 0.0120236, 0.0205478, 0.0351345, 0.0570832, 0.101557, 0.167517, 0.256507, 0.332706}	147.89
550	ω = {115029., 72130.4, 33387.7, 13360.8, 5038.38, 1857.63, 679.299, 247.749, 90.3589, 33.0419, 12.3487, 4.66169, 1.8427, 0.848005, 0.532533} β = {0.00040496, 0.000527957, 0.000811066, 0.0013401, 0.00227553, 0.00390105, 0.00670944, 0.0115513, 0.0198904, 0.034263, 0.0564379, 0.100541, 0.167134, 0.25788, 0.336333}	160.58
600	ω = {136689., 84854.1, 38733.6, 15286.4, 5689.71, 2071.79, 748.469, 269.715, 97.1953, 35.1137, 12.9426, 4.82212, 1.88311, 0.856234, 0.533187} β = {0.000371046, 0.000486682, 0.000753751, 0.00125532, 0.00214748, 0.00370786, 0.00642182, 0.0111329, 0.0193026, 0.0334767, 0.0557995, 0.099597, 0.166751, 0.259115, 0.339681}	173.08
650	ω = {160196., 98524.8, 44400.5, 17301.1, 6363., 2290.64, 818.358, 291.664, 103.952, 37.1391, 13.5182, 4.97644, 1.92159, 0.863999, 0.5338} β = {0.000342267, 0.000451462, 0.000704454, 0.00118179, 0.00203555, 0.00353769, 0.00616654, 0.0107586, 0.0187724, 0.0327614, 0.0551766, 0.0987156, 0.166373, 0.260234, 0.342789}	185.41
700	ω = {185548., 113131., 50380.6, 19401.8, 7057.24, 2513.9, 888.92, 313.598, 110.635, 39.1228, 14.0774, 5.12526, 1.95835, 0.871356, 0.534377} β = {0.000317542, 0.000421048, 0.000661567, 0.00111734, 0.00193672, 0.00338637, 0.00593793, 0.010421, 0.0182905, 0.0321064, 0.0545736, 0.0978891, 0.165999, 0.261253, 0.345689}	197.58
750	ω = {212741., 128662., 56666.5, 21585.6, 7771.5, 2741.35, 960.116, 335.519, 117.253, 41.0685, 14.6216, 5.26907, 1.99358, 0.878346, 0.534923} β = {0.000296072, 0.000394511, 0.000623891, 0.00106032, 0.0018487, 0.00325072, 0.00573167, 0.0101143, 0.0178497, 0.0315029, 0.0539927, 0.0971112, 0.165631, 0.262187, 0.348405}	209.60
800	ω = {241773., 145109., 63251.7, 23849.8, 8504.98, 2972.78, 1031.91, 357.429, 123.809, 42.9795, 15.1522, 5.40832, 2.0274, 0.88501, 0.53544} β = {0.000277257, 0.00037115, 0.000590511, 0.00100947, 0.00176971, 0.00312826, 0.00554433, 0.00983409, 0.0174443, 0.030944, 0.0534343, 0.0963765, 0.16527, 0.263047, 0.350959}	221.48
850	ω = {272642., 162462., 70130., 26192., 9256.94, 3208.01, 1104.27, 379.329, 130.308, 44.8586, 15.6702, 5.54338, 2.05996, 0.891377, 0.535932} β = {0.000260633, 0.000350423, 0.000560715, 0.000963807, 0.00169836, 0.00301701, 0.00537318, 0.0095766, 0.0170695, 0.0304242, 0.0528984, 0.0956807, 0.164917, 0.263841, 0.353369}	233.24
900	ω = {305344., 180712., 77296., 28610.2, 10026.7, 3446.87, 1177.17, 401.219, 136.753, 46.708, 16.1767, 5.67458, 2.09136, 0.897476, 0.536401} β = {0.000245842, 0.000331905, 0.000533945, 0.000922543, 0.00163353, 0.00291539, 0.00521603, 0.00933889, 0.0167216, 0.0299387, 0.0523842, 0.0950198, 0.16457, 0.264579, 0.355649}	244.87
950	ω = {339879., 199853., 84744.3, 31102.1, 10813.7, 3689.21, 1250.58, 423.1, 143.15, 48.5298, 16.6725, 5.80222, 2.12169, 0.903331, 0.53685} β = {0.000232596, 0.00031526, 0.000509752, 0.000885049, 0.00157432, 0.00282213, 0.00507108, 0.0091185, 0.0163973, 0.0294837, 0.0518908, 0.0943906, 0.16423, 0.265265, 0.357814}	256.39
1000	ω = {376243., 219876., 92470.3, 33666.1, 11617.4, 3934.9, 1324.47, 444.973, 149.499, 50.326, 17.1582, 5.92656, 2.15105, 0.908961, 0.537279} β = {0.000220668, 0.000300214, 0.000487773, 0.000850813, 0.00152, 0.00273615, 0.00493681, 0.0089134, 0.016094, 0.0290559, 0.0514173, 0.0937903, 0.163898, 0.265906, 0.359873}	267.81

Table A.23
Parameters for optimized $P = 15$ SRJ schemes for various values of N ($N = 2^k\, k = 6, \ldots, 11$).

N	Optimal Scheme Parameters	ρ
64	$\boldsymbol{\omega}$ = {1604.55, 1236.6, 777.72, 429.57, 220.699, 109.268, 53.1653, 25.7023, 12.4395, 6.06839, 3.77684, 2.26342, 1.17188, 0.697364, 0.519746} $\boldsymbol{\beta}$ = {0.00324844, 0.00375019, 0.00483085, 0.00665688, 0.00950942, 0.0138266, 0.0202681, 0.0298105, 0.0439172, 0.0661899, 0.0257826, 0.120006, 0.167699, 0.222552, 0.261952}	23.72
128	$\boldsymbol{\omega}$ = {6371.83, 4666.19, 2709.05, 1370.56, 646.134, 294.622, 132.367, 59.1371, 26.4159, 11.8529, 5.94244, 2.89717, 1.36449, 0.743778, 0.523863} $\boldsymbol{\beta}$ = {0.00167178, 0.00198802, 0.00268004, 0.00387851, 0.00580983, 0.00883865, 0.0135361, 0.0207864, 0.031965, 0.0496738, 0.0510383, 0.113836, 0.169489, 0.236713, 0.288096}	44.51
256	$\boldsymbol{\omega}$ = {25234.4, 17233., 9009.26, 4080.06, 1729.54, 712.786, 290.428, 117.862, 47.8274, 19.4772, 8.32423, 3.5472, 1.54899, 0.785632, 0.527435} $\boldsymbol{\beta}$ = {0.000861253, 0.0010671, 0.00152771, 0.00235298, 0.00373991, 0.00601951, 0.00973541, 0.0157724, 0.0255664, 0.0415604, 0.0589968, 0.108078, 0.169214, 0.246607, 0.308902}	82.46
512	$\boldsymbol{\omega}$ = {99805.2, 63101.3, 29545., 11959.4, 4558.78, 1698.18, 627.242, 231.042, 85.1043, 31.433, 11.8839, 4.53525, 1.81056, 0.841402, 0.532005} $\boldsymbol{\beta}$ = {0.000435073, 0.000564418, 0.000861316, 0.00141386, 0.00238609, 0.00406664, 0.00695417, 0.0119047, 0.0203829, 0.0349166, 0.0569284, 0.101306, 0.167425, 0.256851, 0.333604}	150.96
1024	$\boldsymbol{\omega}$ = {394347., 229799., 96276., 34921.9, 12008.9, 4053.99, 1360.11, 455.47, 152.531, 51.1795, 17.388, 5.98513, 2.16481, 0.91159, 0.537479} $\boldsymbol{\beta}$ = {0.000215354, 0.000293494, 0.00047792, 0.00083541, 0.00149547, 0.0026972, 0.00487579, 0.00881986, 0.0159551, 0.0288594, 0.0511968, 0.0935117, 0.163741, 0.266199, 0.360827}	273.25
2048	$\boldsymbol{\omega}$ = {1556575., 832736., 312142., 101721., 31639.4, 9698.49, 2959.69, 902.095, 274.961, 83.9203, 25.799, 8.03399, 2.62374, 0.99542, 0.543653} $\boldsymbol{\beta}$ = {0.000104643, 0.000150277, 0.000261282, 0.000485959, 0.000922098, 0.0017595, 0.00336258, 0.00642874, 0.0122909, 0.0234935, 0.0445452, 0.0851626, 0.158288, 0.273725, 0.38902}	489.09

References

[1] L. Evans, Differential Equations (Graduate Studies in Mathematics), 2nd edition, American Mathematical Society, 2010, ebenda, URL http://books.google.es/books?id=TwFInwEACAAJ.
[2] C. Jacobi, Ueber eine neue auflösungsart der bei der methode der kleinsten quadrate vorkommenden linären gleichungen, Astron. Nachr. 22 (20) (1845) 297–306, http://dx.doi.org/10.1002/asna.18450222002.
[3] L.F. Richardson, The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam, R. Soc. Lond. Philos. Trans. Ser. A 210 (1911) 307–357, http://dx.doi.org/10.1098/rsta.1911.0009.
[4] M.L. Juncosa, T.W. Mulliken, On the increase of convergence rates of relaxation procedures for elliptic partial differential equations, J. Assoc. Comput. Math. 7 (1960) 29–36.
[5] A.D. Gunawardena, S.K. Jain, L. Snyder, Modified iterative methods for consistent linear systems, Linear Algebra Appl. 154 (1991) 123–143, http://dx.doi.org/10.1016/0024-3795(91)90376-8.
[6] D. Noutsos, M. Tzoumas, On optimal improvements of classical iterative schemes for Z-matrices, J. Comput. Appl. Math. 188 (2006) 89–106.
[7] U. Trottenberg, C.W. Oosterlee, A. Schüller, Multigrid, Academic Press, 2001.
[8] D. Young, Iterative methods for solving partial difference equations of elliptic type, Trans. Am. Math. Soc. 76 (1954) 92–111.
[9] X.I. Yang, R. Mittal, Acceleration of the Jacobi iterative method by factors exceeding 100 using scheduled relaxation, J. Comput. Phys. 274 (2014) 695–708, http://dx.doi.org/10.1016/j.jcp.2014.06.010, URL http://www.sciencedirect.com/science/article/pii/S0021999114004173.
[10] P.E. Bjorstad, O.B. Widlund, Iterative methods for the solution of elliptic problems on regions partitioned into substructures, SIAM J. Numer. Anal. 23 (1986) 1097–1120, http://dx.doi.org/10.1137/0723075.
[11] J.E. Adsuara, I. Cordero-Carrión, P. Cerdá-Durán, M.A. Aloy, Improvements in the Scheduled Relaxation Jacobi method, in: J.M.C. Díaz Moreno, C. García Vázquez, J. Medina Moreno, F. Ortegón Gallego, F. Pérez Martínez, M.V. Redondo Neble, J.R. Rodríguez Galván (Eds.), Proceedings of the XXIV Congress on Differential Equations and Applications/XIV Congress on Applied Mathematics, Cádiz, June 8–12, 2015, Servicio de Publicaciones de la Universidad de Cádiz, 2015.
[12] G.D. Smith, Numerical Solution of Partial Differential Equations, third edition, Oxford University Press, 1985.
[13] Lapack – linear algebra package, URL http://www.netlib.org/lapack/.
[14] Lorene, URL http://www.lorene.obspm.fr/.
[15] H. Dimmelmeier, J. Novak, J.A. Font, J.M. Ibáñez, E. Müller, Combining spectral and shock-capturing methods: a new numerical approach for 3D relativistic core collapse simulations, Phys. Rev. D 71 (6) (2005) 064023, http://dx.doi.org/10.1103/PhysRevD.71.064023, arXiv:astro-ph/0407174.
[16] H. Grad, J. Hogan, Classical diffusion in a Tokomak, Phys. Rev. Lett. 24 (1970) 1337–1340, http://dx.doi.org/10.1103/PhysRevLett.24.1337.
[17] V.D. Shafranov, On magnetohydrodynamical equilibrium configurations, Sov. J. Exp. Theor. Phys. 6 (1958) 545.