Journal of Computational Physics 332 (2017) 446–460

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

On the equivalence between the Scheduled Relaxation Jacobi method and Richardson's non-stationary method

J.E. Adsuara [a], I. Cordero-Carrión [b], P. Cerdá-Durán [a], V. Mewes [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 12 July 2016
Received in revised form 20 October 2016
Accepted 13 December 2016
Available online 15 December 2016

Keywords:
Iterative methods for linear systems
Jacobi method
Richardson method
Scheduled relaxation Jacobi method
Finite difference methods
Elliptic equations

ABSTRACT

The Scheduled Relaxation Jacobi (SRJ) method is an extension of the classical Jacobi iterative method to solve linear systems of equations ($Au = b$) associated with elliptic problems. It inherits its robustness and accelerates its convergence rate computing a set of P relaxation factors that result from a minimization problem. In a typical SRJ scheme, the former set of factors is employed in cycles of M consecutive iterations until a prescribed tolerance is reached. We present the analytic form for the optimal set of relaxation factors for the case in which all of them are strictly different, and find that the resulting algorithm is equivalent to a non-stationary generalized Richardson's method where the matrix of the system of equations is preconditioned multiplying it by $D = \mathrm{diag}(A)$. Our method to estimate the weights has the advantage that the explicit computation of the maximum and minimum eigenvalues of the matrix A (or the corresponding iteration matrix of the underlying weighted Jacobi scheme) is replaced by the (much easier) calculation of the maximum and minimum frequencies derived from a von Neumann analysis of the continuous elliptic operator. This set of weights is also the optimal one for the general problem, resulting in the fastest convergence of all possible SRJ schemes for a given grid structure. The amplification factor of the method can be found analytically and allows for the exact estimation of the number of iterations needed to achieve a desired tolerance. We also show that with the set of weights computed for the optimal SRJ scheme for a fixed cycle size it is possible to estimate numerically the optimal value of the parameter ω in the Successive Overrelaxation (SOR) method in some cases. Finally, we demonstrate with practical examples that our method also works very well for Poisson-like problems in which a high-order discretization of the Laplacian operator is employed (e.g., a 9- or 17-points discretization). This is of interest since the former discretizations do not yield consistently ordered A matrices and, hence, the theory of Young cannot be used to predict the optimal value of the SOR parameter. Furthermore, the optimal SRJ schemes deduced here are advantageous over existing SOR implementations for high-order discretizations of the Laplacian operator in as much as they do not need to resort to multi-coloring schemes for their parallel implementation.

* 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), vassilios.mewes@uv.es (V. Mewes), miguel.a.aloy@uv.es (M.A. Aloy).

http://dx.doi.org/10.1016/j.jcp.2016.12.020
0021-9991/

1. Introduction

The Jacobi method [1] is an iterative algorithm to solve systems of linear equations. Due to its simplicity and its convergence properties it is a popular choice as preconditioner, in particular when solving elliptic partial differential equations. However, its slow rate of convergence, compared to other iterative methods (e.g. Gauss–Seidel, SOR, Conjugate gradient, GMRES), makes it a poor choice to solve linear systems. The scheduled relaxation Jacobi method [2], SRJ hereafter, is an extension of the classical Jacobi method, which increases the rate of convergence in the case of linear problems that arise in the finite difference discretization of elliptic equations. It consists of executing a series of weighted Jacobi steps with carefully chosen values for the weights in the sequence. The SRJ method can be expressed for a linear system $Au = b$ as

$$u^{n+1} = u^n + \omega_n D^{-1}(b - Au^n), \tag{1}$$

where D is the diagonal of the matrix A. If we consider a set of P different relaxation factors, ω_n, $n = 1, \ldots, P$, such that $\omega_n > \omega_{n+1}$ and we apply each relaxation factor q_n times, the *total amplification factor* after $M := \sum_{n=1}^{P} q_n$ iterations is

$$G_M(\kappa) = \prod_{n=1}^{P} (1 - \omega_n \kappa)^{q_n}, \tag{2}$$

which is an estimation of the reduction of the residual during one cycle (M iterations). In the former expression κ is a function of the wave-numbers obtained from a von Neumann analysis of the system of linear equations resulting from the discretization of the original elliptical problem by finite differences (for more details see [2,3]). Yang & Mittal [2] argued that, for a fixed number P of different weights, there is an optimal choice of the weights ω_n and repetition numbers q_n that minimizes the maximum *per-iteration amplification factor*, $\Gamma_M(\kappa) = |G_M(\kappa)|^{1/M}$, in the interval $\kappa \in [\kappa_{\min}, \kappa_{\max}]$ and therefore also the number of iterations needed for convergence. The boundaries of the interval in κ correspond to the minimum and the maximum weight numbers allowed by the discretization mesh and boundary conditions used to solve the elliptic problem under consideration.

In the aforementioned paper, [2] computed numerically the optimal weights for $P \leq 5$ and Adsuara et al. [3] extended the calculations up to $P = 15$. The main properties of the SRJ, obtained by [2] and confirmed by [3], are the following:

1. Within the range of P studied, increasing the number of weights P improves the rate of convergence.
2. The resulting SRJ schemes converge significantly faster than the classical Jacobi method by factors exceeding 100 in the methods presented by [2] and ~ 1000 in those presented by [3]. Increasing grid sizes, i.e. decreasing $\kappa_{\min}$, results in larger acceleration factors.
3. The optimal schemes found use each of the weights multiple times, resulting in a total number of iterations M per cycle significantly larger than P, e.g. for $P = 2$, [2] found an optimal scheme with $M = 16$ for the smallest grid size they considered ($N = 16$), while for larger grids M notably increases (e.g., $M = 1173$ for $N = 1024$).

The optimization procedure outlined by [2] has a caveat though. Even if the amplification factor were to reduce monotonically by increasing P, for sufficiently high values of P, the number of iterations per cycle M may be comparable to the total number of iterations needed to solve a particular problem for a prescribed tolerance. At this point, using a method with higher P, and thus higher M, would increase the number of iterations to converge, even if the $\Gamma(\kappa)$ is nominally smaller. With this limitation in mind we outline a procedure to obtain optimal SRJ schemes, minimizing the total number of iterations needed to reduce the residual by an amount sufficient to reach convergence or, equivalently, to minimize $|G_M(\kappa)|$. Note that the total number of iterations can be chosen to be equal to M without loss of generality, i.e. one cycle of M iterations is needed to reach convergence. To follow this procedure one should find the optimal scheme for fixed values of M, and then choose M such that the maximum value of $|G_M(\kappa)|$ is similar to the residual reduction needed to solve a particular problem. The first step, the minimization problem, is in general difficult to solve, since fixing M gives an enormous freedom in the choice of the number of weights P, which can range from 1 to M. However, the numerical results of [2] and [3], seem to suggest that in general increasing the number of weights P will always lead to better convergence rates. This leads us to conjecture that the optimal SRJ scheme, for fixed M, is the one with $P = M$, i.e. all weights are different and each weight is used once per cycle, $q_i = 1$, $(i = 1, \ldots, M)$. In terms of the total amplification factor $G_M(\kappa)$, it is quite reasonable to think that if one maximizes the number of different roots by choosing $P = M$, the resulting function is, on average, closer to zero than in methods with smaller number of roots, $P < M$, and one might therefore expect smaller maxima for the optimal set of coefficients. One of the aims of this work is to compute the optimal coefficients for this particular case and demonstrate that $P = M$ is indeed the optimal case.

Another goal of this paper is to show the performance of optimal SRJ methods compared with optimal SOR algorithms applied to a number of different discretizations of the Laplacian operator in two-dimensional (2D) and three-dimensional (3D) applications (Sect. 3). We will show that optimal SRJ methods applied to high-order discretizations of the Laplacian, which yield iteration matrices that cannot be consistently ordered, perform very similarly to optimal SOR schemes (when an optimal SOR weight can be computed). We will further discuss that the trivial parallelization of the SRJ methods outbalances the slightly better scalar performance of SOR in some cases (Sect. 3.3). Also, we will show that the optimal weight of the

SOR method can be suitably approximated by functions related to the geometric mean of the set of weights obtained for optimal SRJ schemes. This is of particular relevance when the iteration matrix is non-consistently ordered and hence, the analytic calculation of the optimal SOR weight is extremely intricate.

2. Optimal $P = M$ SRJ scheme

Let us consider a SRJ method with $P = M$ and hence $q_n = 1$, $(n = 1, \dots, M)$. For this particular choice, the amplification factor $G_M(\kappa)$ is a polynomial of degree M in κ with M different roots. In this case, the set of weights ω_n that minimizes the value of the maximum of $|G_M(\kappa)|$, given by Eq. (2), in the interval $\kappa \in [\kappa_{\min}, \kappa_{\max}]$, $0 < \kappa_{\min} \leq \kappa_{\max}$,[1] can be determined by the definition of the amplification factor

$$G_M(0) = 1, \tag{3}$$

and by the following M conditions[2]:

$$G_M(\kappa_n) = -G_M(\kappa_{n+1}), \quad n = 0, \dots, M-1, \tag{4}$$

where $\kappa_0 = \kappa_{\min}$, $\kappa_M = \kappa_{\max}$, and κ_n, $n = 1, \dots, M-1$ are the relative extrema of the function $G_M(\kappa)$. To simplify further we rescale κ as follows:

$$\tilde{\kappa} = 2\frac{\kappa - \kappa_{\min}}{\kappa_{\max} - \kappa_{\min}} - 1. \tag{5}$$

As a function of $\tilde{\kappa}$ the amplification factor is $\tilde{G}_M(\tilde{\kappa}) = G_M(\kappa(\tilde{\kappa}))$. In the resulting interval, $\tilde{\kappa} \in [-1, 1]$, there is a unique polynomial of degree M such that the absolute value of $\tilde{G}_M(\tilde{\kappa})$ at the extrema $\tilde{\kappa}_i$ is the same (fulfilling Eqs. (4)) and such that $\tilde{G}_M(\tilde{\kappa}(0)) = 1$. This polynomial is proportional to the Chebyshev polynomial of first kind of degree M, $T_M(\kappa)$, which can be defined through the identity $T_M(\cos\theta) = \cos(M\theta)$. This polynomial satisfies that

$$|T_M(-1)| = |T_M(\tilde{\kappa}_n)| = |T_M(+1)| = 1, \quad n = 1, \dots, M-1, \tag{6}$$

with $\tilde{\kappa}_i$ being the local extrema of $T_M(\tilde{\kappa})$ in $[-1, 1]$. The constant of proportionality can be determined from Eq. (3), and the amplification factor reads in this case

$$\tilde{G}_M(\tilde{\kappa}) = \frac{T_M(\tilde{\kappa})}{T_M(\tilde{\kappa}(0))} \quad ; \quad \tilde{\kappa}(0) = -\frac{(1 + \kappa_{\min}/\kappa_{\max})}{(1 - \kappa_{\min}/\kappa_{\max})} < -1. \tag{7}$$

This result is equivalent to Markoff's theorem.[3] Note that the value of $\tilde{\kappa}(0)$ does not depend on the actual values of $\kappa_{\min}$ and $\kappa_{\max}$, but only on the ratio $\kappa_{\min}/\kappa_{\max}$. The roots and local extrema of the polynomial $T_M(\tilde{\kappa})$ are located, respectively, at

$$\tilde{\omega}_n^{-1} = -\cos\left(\pi\frac{2n-1}{2M}\right), \quad n = 1, \dots, M, \tag{8}$$

$$\tilde{\kappa}_n = \cos\left(\pi\frac{n}{M}\right), \quad n = 1, \dots, M-1, \tag{9}$$

which coincide with those of $\tilde{G}_M(\tilde{\kappa})$. Therefore, the set of weights

$$\omega_n = 2\left[\kappa_{\max} + \kappa_{\min} - (\kappa_{\max} - \kappa_{\min})\cos\left(\pi\frac{2n-1}{2M}\right)\right]^{-1}, \quad n = 1, \dots, M, \tag{10}$$

corresponds to the optimal SRJ method for $P = M$.

We have found with the simple analysis of this section that the optimal SRJ scheme when $P = M$ is fixed turns out to be closely related to a Chebyshev iteration or Chebyshev semi-iteration for the solution of systems of linear equations (see, for instance, [6] for a review). This is especially easy to realize if we consider the original formulation of this kind of methods, which appeared in the literature as special implementations of the non-stationary or semi-iterative Richardson's method (RM, hereafter; see, e.g., [7,8] for generic systems of linear equations, or [9] for the application to boundary-value problems). Yang & Mittal [2] argued that, for a uniform grid, Eq. (1) is identical to that of the RM [10]. There is, nevertheless, a minor difference between Eq. (1) of the SRJ method and the RM as it has been traditionally written [11], that using our notation would be $u^{n+1} = u^n + \hat{\omega}_n(b - Au^n)$, which gives the obvious relation $\hat{\omega}_n = \omega_n d^{-1}$, in the case in which all elements in D are the same and equal to d. We note that this difference disappears in more modern formulations of the RM (e.g.,

[1] In this work, $\kappa_{\min}$ and $\kappa_{\max}$ are assumed to be strictly positive as the discretization of an elliptic problem leads to a matrix A that is positive definite. In problems where it is not, a simple option is to work with the matrix $A^T A$ and the equivalent system $A^T Au = A^T b$.

[2] These conditions result from the solution of a global min–max optimization problem over $G_M(\kappa)$ or, equivalently, over $\Gamma_M(\kappa)$ (see Appendix B of [2]).

[3] For an accessible proof of the original theorem [4], see Young's textbook [5], Theorem 9-3.1.

[12]), in which the RM is also written as a fix point iteration of the form $u^{n+1} = Tu^n + c$, with $T = I - B^{-1}A$, $c = B^{-1}b$ and B any non-singular matrix. Differently from the RM in its definition by Young [11], our method in the case $M = 1$ would fall in the category of stationary Generalized Richardson's (GRF) methods according to the textbook of Young [5, Chap. 3]. GRF methods are defined by the updating formula

$$u^{n+1} = u^n + \mathcal{P}(Au^n - b) \tag{11}$$

where $\mathcal{P}$ is any non-singular matrix (in our case, $\mathcal{P} = -\omega_n D^{-1}$). In the original work of Richardson [10], all the values of $\hat{\omega}_n^{-1}$ where set either equal or evenly distributed in $[a, b]$, where a and b are, respectively, lower and upper bounds to the minimum and maximum eigenvalues, λ_i of the matrix A (optimally, $a = \min(\lambda_i)$, $b = \max(\lambda_i)$). If a single weight is used throughout the iteration procedure, a convenient choice is $\hat{\omega} = 2/(b + a)$.[4]

Yang & Mittal [2] state that the SRJ approach to maximizing convergence is fundamentally different from that of the stationary RM. They argue that the RM aims to reduce $\Gamma(\kappa)$ uniformly over the range $[\kappa_{\min}, \kappa_{\max}]$ by generating equally spaced nodes of Γ in this interval, while SRJ methods set a min–max problem whose goal is to minimize $|\Gamma|_{\max}$.[5] As a result, SRJ methods require computing a set of weights yielding two differences with respect to the non-stationary RM in its original formulation [2]:

1. The nodes in the SRJ method are not evenly distributed in the range $[\kappa_{\min}, \kappa_{\max}]$;
2. Optimal SRJ schemes naturally have many repetitions of the same relaxation factor whereas RM generated distinct values of $\hat{\omega}_n$ in each iteration of a cycle.

From these two main differences, Yang & Mittal [2] conclude that while optimal SRJ schemes actually gain in convergence rate over Jacobi method as grids get larger, the convergence rate gain for Richardson's procedure (in its original formulation) never produces acceleration factors larger than 5 with respect to the Jacobi method. This result was supported by Young in his Ph.D. thesis [13, p. 4], but on the basis of employing orderings of the weights which did pile-up roundoff errors, preventing a faster method convergence (see point 2 below).

The difference outlined in point 1 above is non existent for GRF methods, where the eigenvalues of A are not necessarily evenly distributed in the spectral range of matrix A (i.e., in the interval $[a, b]$). We note that Young [7] attempted to chose the $\hat{\omega}_n$ parameters of the RM to be the reciprocals of the roots of the corresponding Chebyshev polynomials in $[a, b]$, which resulted in a method that is *almost the same* as ours, but with two differences:

First, we do not need to compute the maximum and minimum eigenvalues of the matrix A; instead, we compute $\kappa_{\max}$ and $\kappa_{\min}$, which are related to the maximum and minimum frequencies that can be developed on the grid of choice employing a straightforward von Neumann analysis. Indeed, this procedure to estimate the maximum and minimum frequencies for the elliptic operators (e.g., the Laplacian) in the continuum limit allows applying it to matrices that are not necessarily consistently ordered, like, e.g., the ones resulting from the 9-point discretization of the Laplacian [14]. In Sect. 3.3 we show how our method can be straightforwardly prescribed in this case and other more involved (high-order) discretizations of the Laplacian.

Second, in Young's method [7] the two-term recurrence relation given by Eq. (1) turned out to be unstable. Young found that the reason for the instability was the build up of roundoff errors in the evaluation of the amplification factor (Eq. (2)), which resulted as a consequence of the fact that many of the values of ω_n can be much larger than one. Somewhat unsuccessfully, Young [7] tried different orderings of the sequence of weights ω_n, and concluded that, though they ameliorated the problem for small values of M, did not cure it when M was sufficiently large. Later, Young [11,15] examines a number of orderings and concluded that some gave better results than others. However, the key problem of existence of orderings for which RM defines a stable numerical algorithm amenable to a practical implementation was not shown until the work of Anderssen & Golub [16]. These authors showed that employing the ordering developed by Lebedev & Finogenov [17] for the iteration parameters in the Chebyshev cyclic iteration method, the RM devised by Young [7] was stable against the pile-up of round-off errors. However, Anderssen & Golub [16] left open the question of whether other orderings are possible. In our case, numerical stability is brought about by the ordering of the weights in the iteration procedure. This ordering is directly inherited from the SRJ schemes of [2], and notably differs from the prescriptions given for two- or three-term iteration relations in Chebyshev semi-iterations [6] and from those suggested by [7]. Indeed, the ordering we use differs from that of [17–19] (see Appendix A). Thus, though we do not have a theoretical proof for it, we empirically confirm that other alternative orderings work.

Taking advantage of the analysis made by [7], we point out that the average rate of convergence of the method in a cycle of M iterations is

$$R_M = \frac{1}{M} \log |T_M(\tilde{\kappa}(0))|, \tag{12}$$

[4] In the case of SRJ schemes with $P = M$, it is easy to demonstrate (see Appendix B) that the harmonic mean of the weights ω_n very approximately equals the value of the inverse weight of the stationary RM ($2d^{-1}/(\kappa_{\max} + \kappa_{\min}) \simeq 2/(b + a)$).

[5] We note that this argument does not hold in the implementation of the non-stationary RM method made by Young [7], since in this case one also attempts to minimize $|\Gamma|_{\max}$.

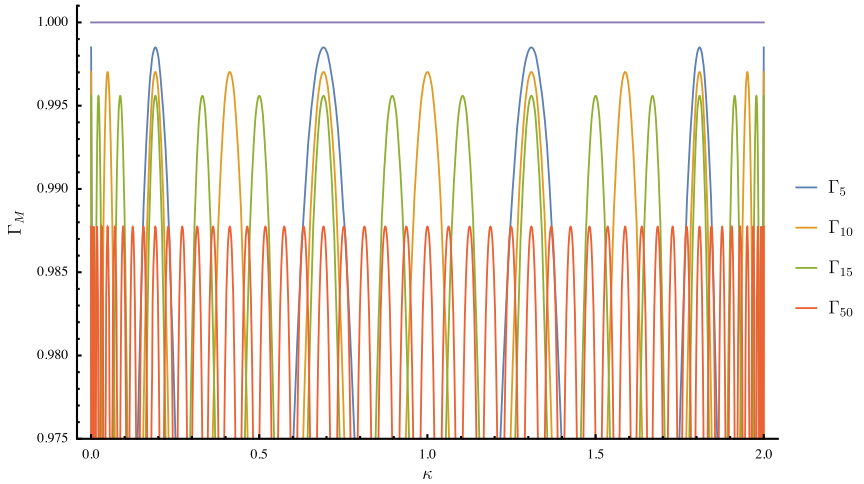

Fig. 1. Plot $\Gamma_M(\kappa)$ for the following different values of M (5, 10, 15, and 50) for a bidimensional mesh of 128×128 points. One can see that all the extrema are equal. The plot also shows that the higher the value of M, the lower the local maxima of Γ_M. A color version of the figure can be found in the online version.

and it is trivial to prove that for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$

$$G_M(\kappa) \leq \left| \frac{1}{T_M(\tilde{\kappa}(0))} \right| < 1, \tag{13}$$

providing a simple way to compute an upper bound for the amplification factor for the optimal scheme. This condition also guarantees the convergence of the optimal SRJ method. Therefore, if we aim to reduce the initial residual of the method by a factor σ, we have to select a sufficiently large M such that

$$\sigma \geq |T_M(\tilde{\kappa}(0))|^{-1}. \tag{14}$$

It only remains to demonstrate that the optimal SRJ scheme with $P = M$ is also the optimal SRJ scheme for any $P \leq M$. Markoff's theorem states that for any polynomial $Q(x)$ of degree smaller or equal to M, such that $\exists x_0 \in \mathbb{R}, x_0 < -1$, with $Q(x_0) = 1$, and $Q(x) \neq T_M(x)/T_M(x_0)$, then

$$\max |Q(x)| > \max \left| \frac{T_M(x)}{T_M(x_0)} \right| \quad \forall x \in [-1, 1]. \tag{15}$$

This theorem implies that any other polynomial of order $P \leq M$, different from Eq. (7), is a poorer choice as amplification factor. The first implication is that $G_M(\tilde{\kappa}(0)) < G_{M-1}(\tilde{\kappa}(0))$, i.e., increasing M decreases monotonically the amplification factor $G_M(\kappa)$. As a consequence, the per iteration amplification factor $\Gamma_M(\kappa)$ also decreases by increasing M. The second consequence is that the case $P < M$ results in an amplification factor with larger extrema than the optimal $P = M$ case, and hence proves that our numerical scheme leads to the optimal set of weights for any SRJ method with M steps. This confirms our intuition that adding additional roots to the polynomial would decrease the value of its maxima, resulting in faster numerical methods. Though the SRJ algorithm with $P = M$ we have presented here turns out to be nearly equivalent to the non-stationary RM of Young [7], in order to single it out as the optimum among the SRJ schemes, we will refer to it as the *Chebyshev–Jacobi* method (CJM) henceforth.

Finally, we plot in Fig. 1 the per-iteration amplification factor, $\Gamma_M(\kappa)$, for different values of M. It is evident from the plot that all the maxima are of equal height, and that the maxima decrease as M increases.

3. Numerical examples

3.1. Laplace equation

In order to assess the performance of the new optimal set of schemes devised, we resort to the same prototype numerical example considered in [2], namely, the solution of the Laplace equation with homogeneous Neumann boundary conditions in two spatial dimensions, in Cartesian coordinates and over a domain with unitary size:

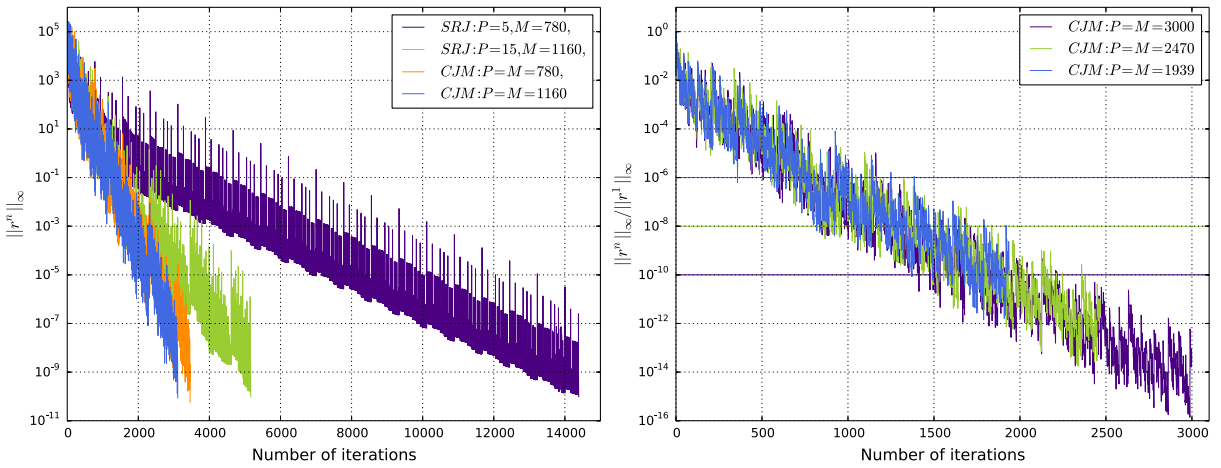

Fig. 2. Left: Evolution of the residual $||r^n||_\infty$, defined in Eq. (18), as a function of the number of iterations for the problem set in Eq. (16) and a Cartesian grid of 256×256 uniform zones. The different color lines correspond to different schemes (see legends). We can observe that the reduction of the residual is faster in the new Chebyshev–Jacobi schemes than in the corresponding SRJ schemes with the same value of M. Right: We show three examples where we computed the optimal value of the M for reaching the desired residual in one cycle. The cases $P = 1939$, 2470 and 3000 correspond to schemes that (theoretically) should reduce the initial residual by factors $\simeq 10^6$, 10^8 and 10^{10}. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

$$\begin{cases} \dfrac{\partial^2}{\partial x^2}u(x,y) + \dfrac{\partial^2}{\partial y^2}u(x,y) = 0, & (x,y) \in (0,1)\times(0,1) \\ \dfrac{\partial}{\partial x}u(x,y)\Big|_{x=0} = \dfrac{\partial}{\partial x}u(x,y)\Big|_{x=1} = 0, & y \in (0,1) \\ \dfrac{\partial}{\partial y}u(x,y)\Big|_{y=0} = \dfrac{\partial}{\partial y}u(x,y)\Big|_{y=1} = 0, & x \in (0,1). \end{cases} \tag{16}$$

We consider a spatial discretization of the Laplacian operator employing a second-order, 5-point formula

$$\Delta u_{ij} = \frac{1}{h^2}\Big[u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{ij}\Big], \tag{17}$$

where we are assuming that the grid spacing, h, is the same along the x and y directions. In all examples presented in this work, we will use initial random data to initialize our computations. To compare the performance of different numerical schemes we monitor the evolution of the difference between two consecutive approximations of the solution for the model problem specified in Eq. (16),

$$||r^n||_\infty = \max_{ij} |u_{ij}^n - u_{ij}^{n-1}|, \tag{18}$$

where u_{ij}^n is the numerical approximation computed after n iterations at the grid point (x_i, y_j).

In Fig. 2 (left), we compare the evolution of the residual as a function of the number of iterations for several SRJ schemes, as well as for the new schemes developed here. The violet line corresponds to the best SRJ scheme presented in [2] for the solution of the problem set above and a spatial grid of $N_x \times N_y = 256 \times 256$ uniform zones, i.e. the SRJ scheme with $P = 5$ and $M = 780$. Comparing with the new CJM for $P = M = 780$ (orange line in Fig. 2 left), it is evident that the new scheme reduces the number of iterations to reach the prescribed tolerance ($||r^n||_\infty \leq 10^{-10}$ in this example) by about a factor of 5. We also include in Fig. 2 (left; green line) the residual evolution corresponding to the best SRJ optimal algorithm developed by [3] for the proposed resolution, namely, the scheme with $P = 15$ levels and $M = 1160$. It is obvious that even the CJM with $P = M = 780$ reduces the residual faster than the $P = 15$ SRJ scheme. However, since the $P = 15$ SRJ scheme requires a larger value of M than in the case of $P = 5$, for a fair comparison, we also include in Fig. 2 (left; blue line) the CJM with $P = M = 1160$. The latter is the best performing scheme, thought the difference between the two new CJM with different values of P is very small (in Fig. 2 the blue and orange lines practically overlap).

A positive property of the new algorithm presented in Sect. 2 is its predictability, i.e., the easiness to estimate the size of the M-cycle in order to reduce the tolerance by a prescribed amount (Eq. (14)). Indeed, it is not necessary to monitor the evolution of the residual in every iteration (as in many other non-stationary methods akin to the Richardson's method – e.g., in the gradient method), with the obvious reduction in computational load per iteration that this implies. In Fig. 2 (right) we show that our algorithm performs as expected, reducing the initial residual by factors of larger than 10^6, 10^8 and 10^{10} in a single cycle consisting of $P = 1939$, 2470 and 3000 iterations, respectively, since for the problem at hand we have $\kappa_{\min} = \sin(\frac{\pi}{2\times 256})^2 = 3.76491 \times 10^{-5}$, $\kappa_{\max} = 2$, and thus, $\tilde{\kappa}(0) = -1.00004$.

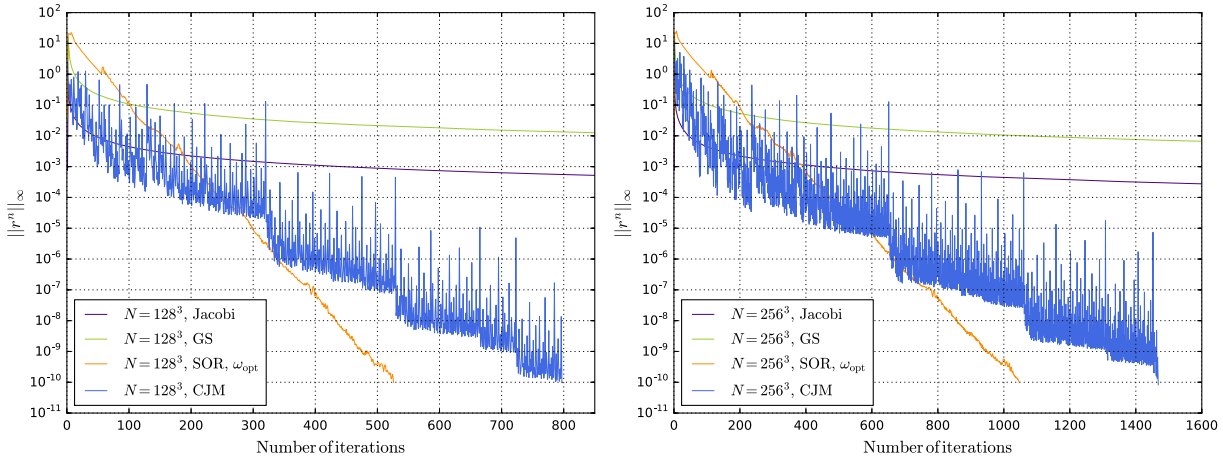

Fig. 3. The evolution of the residual for the solution of the Poisson equation (19) in 3D, with $N = 128$ (left panel) and $N = 256$ (right panel) for different iterative methods. The apparent faster convergence of Jacobi relative to Gauss–Seidel is due to the shown small interval of iterations which was chosen to highlight the convergence behavior of CJM relative to SOR. Gauss–Seidel actually reaches the desired tolerance faster than Jacobi in all examples, as expected. A color version of the figure can be found in the online version.

In this simple example the upper bound for the residual obtained from Eq. (14) is very rough and clearly overestimates the number of iterations to reduce the residual below the prescribed values. In more complex problems this will not necessarily be the case as we will show in the following, more demanding example.

3.2. Poisson equation in 3D

Here we test the CJM and the predictability of the residual evolution in a three-dimensional elliptic equation with a source term. For this test, we use infrastructure provided by the Einstein Toolkit [20,21]. The actual calculation is finding the static field of a uniformly charged sphere of radius R in 3D Cartesian coordinates subject to Dirichlet boundary conditions, solving the Poisson equation:

$$\Delta\phi(x, y, z) = -4\pi\rho, \tag{19}$$

where $\rho = \frac{3Q}{4\pi R^3}$ and Q is the charge of the sphere. We solve the elliptic equation (19) with a standard second-order accurate 7-point stencil with $h_x = h_y = h_z = h$

$$\Delta u_{ijk} = \frac{1}{h^2}\Big[u_{i-1,jk} + u_{i+1,jk} + u_{i,j-1,k} + u_{i,j+1,k} + u_{i,j,k-1} + u_{ij,k+1} - 6u_{ijk}\Big]. \tag{20}$$

We consider two different grid sizes with $N_x = N_y = N_z = N = 128$ and $N_x = N_y = N_z = N = 256$ points and the following iterative methods: Jacobi, Gauss–Seidel (SOR with $\omega = 1$), SOR with the optimal relaxation factor $\omega_{opt} = 2/(1 + \sin(\pi/N))$, and CJM with the optimal sequence of weights for a given resolution. The results for the two grid resolutions are shown in Fig. 3. Both SOR and CJM (slightly less than twice the number of iterations of SOR) are more than an order of magnitude faster than the Jacobi and Gauss–Seidel methods. While the CJM method is not as fast as SOR when using the optimal relaxation factor ω_{opt}, we note here two arguments that should favor the use of the CJM over SOR: Firstly, Young's theory of relating ω_{opt} to the spectral radius of the Jacobi iteration matrix $\rho(J)$ via $\omega_{opt} = 2/(1 + \sqrt{1 - \rho(J)^2})$ only applies when the original matrix of the linear system $Au = b$ is consistently ordered. Secondly, the CJM method is trivially parallelized, while SOR requires multicolor schemes for a successful parallelization, as we will discuss below presenting results for 9-point and 17-point Laplacians in 2D.

Next, we solve equation (19) subject to reflection symmetry (homogeneous Neumann boundary conditions) at the $x = 0$, $y = 0$, $z = 0$ planes (so-called octant symmetry) with $N_x = N_y = N_z = N = 64$ points, using the same iterative methods as before. For the CJM, we choose the same sequence of weights as those we used for the full 3D domain using $N = 128$ points. Because of the boundary conditions used to impose octant symmetry, the resulting matrix A is non-consistently ordered and hence there is no analytic expression to calculate ω_{opt} for SOR; in this case we test a sequence of values of ω to empirically estimate the optimal value for the given problem. The residuals of the different iterative methods are shown in Fig. 4. The CJM now performs better than SOR for any ω we have tested. Furthermore, as seen in the plot, SOR is very sensitive to the exact value of ω that is chosen, as is well known. The CJM method is free of this need to estimate and choose a sensitive parameter.

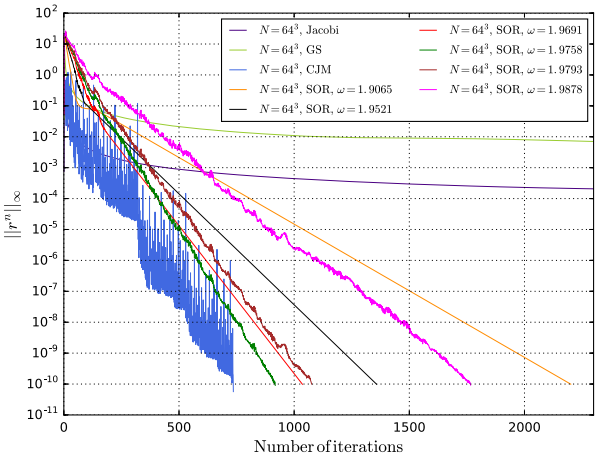

Fig. 4. The evolution of the residual for the solution of the Poisson equation (19) in 3D using octant symmetry, with $N = 64$ for different iterative methods and different relaxation factors ω in SOR. The apparent faster convergence of Jacobi relative to Gauss–Seidel is due to the shown small interval of iterations which was chosen to highlight the convergence behavior of CJM relative to SOR. Gauss–Seidel actually reaches the desired tolerance faster than Jacobi in all examples, as expected. A color version of the figure can be found in the online version.

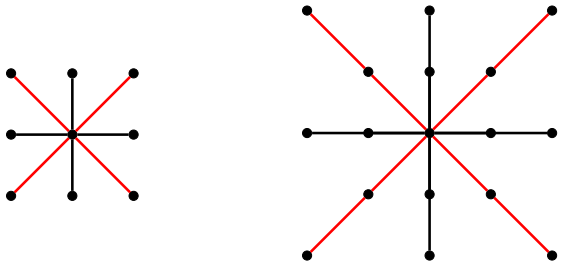

Fig. 5. Schematic representation of the 9- and 17-point stencils. The black and red lines correspond to the standard stencil S_+ and rotated stencil $S_\times$, respectively. See main text for details. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.3. CJM for non-consistently ordered matrices: high-order discretization of the Laplacian operator in 2D with 9 and 17 points

As we have already mentioned, Young's theory of relating the optimal SOR parameter to the spectral radius of the Jacobi iteration matrix does not apply in the case of non-consistently ordered (NCO) matrices. In this section, we will investigate two of these cases, namely a 9-point and 17-point discretization of the Laplacian in 2D.

One way of obtaining this type of discretizations is doing a convex combination between the discretization of the Laplacian operator using the standard stencil, S_+, with its discretization in a rotated stencil, $S_\times$ (see Fig. 5):

$$\alpha S_+ + (1-\alpha) S_\times \,. \tag{21}$$

Writing α as a rational number a/b, the resulting 9-points discretized Laplacian is

$$\Delta u_{ij} = \frac{1}{2bh^2}\Big[2au_{i-1,j} + 2au_{i+1,j} + 2au_{i,j-1} + 2au_{i,j+1} + (b-a)u_{i-1,j-1} + (b-a)u_{i+1,j+1} + (b-a)u_{i-1,j+1} + (b-a)u_{i+1,j-1} - 4(a+b)u_{i,j}\Big], \tag{22}$$

where, for simplicity, we assume that the grid spacing, h, is the same in the x- and y-directions. From this general form, we can recover the standard 5-points discretization simply taking $a = b = 1$. In the same way, we can recover the 9-points discretization of the Laplacian studied in [14] by imposing $a = 2$ and $b = 3$:

$$\Delta u_{ij} = \frac{1}{6h^2}\Big[4u_{i-1,j} + 4u_{i+1,j} + 4u_{i,j-1} + 4u_{i,j+1} + u_{i-1,j-1} + u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} - 20u_{i,j}\Big]. \tag{23}$$

From the von Neumann stability analysis of Eq. (16), we obtain the following expression of the amplification factor for the Laplacian discretization of Eq. (22)

$$G_M = 1 - \omega\left[\frac{2a}{a+b}\sin^2\frac{k_x\Delta x}{2} + \frac{2a}{a+b}\sin^2\frac{k_y\Delta y}{2} + \frac{b-a}{a+b}\big[1-\cos(k_x\Delta x)\cos(k_y\Delta y)\big]\right]. \tag{24}$$

For $\alpha = a = b = 1$, we recover the expression of the amplification factor shown in [2,3]. It is easy to check that when $a = 2$ and $b = 3$, Eq. (24) reduces to

$$G_M = 1 - \frac{\omega}{5}\left[4\sin^2\frac{k_x\Delta x}{2} + 4\sin^2\frac{k_y\Delta y}{2} + 1 - \cos(k_x\Delta x)\cos(k_y\Delta y)\right]. \tag{25}$$

The factor multiplying ω in the previous expression is related to the weights of any SRJ scheme and singularly with the CJM. As a function of the wave number κ, the minimum amplification factor results for $k_x = k_y = \pi/L$, while the maximum amplification factor is attained for $k_x = \pi/\Delta x$ and $k_y = \pi/\Delta y$, with respective wave numbers $\kappa_{\min}$ and $\kappa_{\max}$, whose expressions are

$$\kappa_{\min} = \frac{4}{5}\sin^2\frac{\pi}{2N_x} + \frac{4}{5}\sin^2\frac{\pi}{2N_y} + \frac{1}{5}\left[1 - \cos\frac{\pi}{Nx}\cos\frac{\pi}{N_y}\right], \tag{26}$$

$$\kappa_{\max} = \frac{8}{5}. \tag{27}$$

It can be shown that the 9-point discretization of the Laplacian provides a fourth-order accurate method for the Poisson equation when the source term is smooth [22].

Next, we consider the case of a 17-point discretization of the Laplacian. From the general form of Eq. (21), again writing $\alpha = a/b$ one obtains

$$\begin{aligned}\Delta u_{ij} = \frac{1}{24bh^2}\Big[&-2au_{i-2,j} + 32au_{i-1,j} + 32au_{i+1,j} - 2au_{i+2,j} - 2au_{i,j-2} + 32au_{i,j-1} + 32au_{i,j+1} - 2au_{i,j+2} \\ &- (b-a)u_{i-2,j-2} + 16(b-a)u_{i-1,j-1} + 16(b-a)u_{i+1,j+1} - (b-a)u_{i+2,j+2} - (b-a)u_{i-2,j+2} \\ &+ 16(b-a)u_{i-1,j+1} + 16(b-a)u_{i+1,j-1} - (b-a)u_{i+2,j-2} - 60(a+b)u_{i,j}\Big].\end{aligned} \tag{28}$$

The standard 9-point discretization of the Laplacian is recovered for $a = b = 1$ in Eq. (28). Performing the von Neumann stability analysis for Eq. (16), we obtain the following expression of the amplification factor for the Laplacian discretization of Eq. (28)

$$\begin{aligned}G_M = 1 - \omega\frac{1}{15(a+b)}\Big[&-2a\big(\sin^2(k_x\Delta x) + \sin^2(k_y\Delta y)\big) + 32a\left(\sin^2\left(\frac{k_x\Delta x}{2}\right) + \sin^2\left(\frac{k_y\Delta y}{2}\right)\right) \\ &- (b-a)\big([1 - \cos(2k_x\Delta x)\cos(2k_y\Delta y)] - 16[1 - \cos(k_x\Delta x)\cos(k_y\Delta y)]\big)\Big],\end{aligned} \tag{29}$$

and, therefore, taking into account the minimum and maximum wave numbers as in the previous case, the extremal values of κ are:

$$\begin{aligned}\kappa_{\min} = \frac{1}{15(a+b)}\Big[&-2a\left(\sin^2\frac{\pi}{N_x} + \sin^2\frac{\pi}{N_y}\right) + 32a\left(\sin^2\frac{\pi}{2N_x} + \sin^2\frac{\pi}{2N_y}\right) \\ &- (b-a)\left([1 - \cos\frac{2\pi}{N_x}\cos\frac{2\pi}{N_y}] - 16[1 - \cos\frac{\pi}{N_y}\cos\frac{\pi}{N_y}]\right)\Big],\end{aligned} \tag{30}$$

$$\kappa_{\max} = \frac{64a}{15(a+b)}. \tag{31}$$

Let us consider the particular case $a = 1$ and $b = 2$. For the Laplacian discretization (28), we have

$$\begin{aligned}\Delta u_{ij} = \frac{1}{48h^2}\Big[&-2u_{i-2,j} + 32u_{i-1,j} + 32u_{i+1,j} - 2u_{i+2,j} - 2u_{i,j-2} + 32u_{i,j-1} + 32u_{i,j+1} - 2u_{i,j+2} - u_{i-2,j-2} \\ &+ 16u_{i-1,j-1} + 16u_{i+1,j+1} - u_{i+2,j+2} - u_{i-2,j+2} + 16u_{i-1,j+1} + 16u_{i+1,j-1} - u_{i+2,j-2} - 180u_{i,j}\Big]\end{aligned} \tag{32}$$

and the expressions for $\kappa_{\min}$ and $\kappa_{\max}$ of Eqs. (30) and (31) reduce to

$$\begin{aligned}\kappa_{\min} = \frac{1}{45}\Big[&-2\left(\sin^2\frac{\pi}{N_x} + \sin^2\frac{\pi}{N_y}\right) + 32\left(\sin^2\frac{\pi}{2N_x} + \sin^2\frac{\pi}{2N_y}\right) \\ &- [1 - \cos\frac{2\pi}{N_x}\cos\frac{2\pi}{N_y}] + 16[1 - \cos\frac{\pi}{N_y}\cos\frac{\pi}{N_y}]\Big]\end{aligned} \tag{33}$$

$$\kappa_{\max} = \frac{64}{45} \tag{34}$$

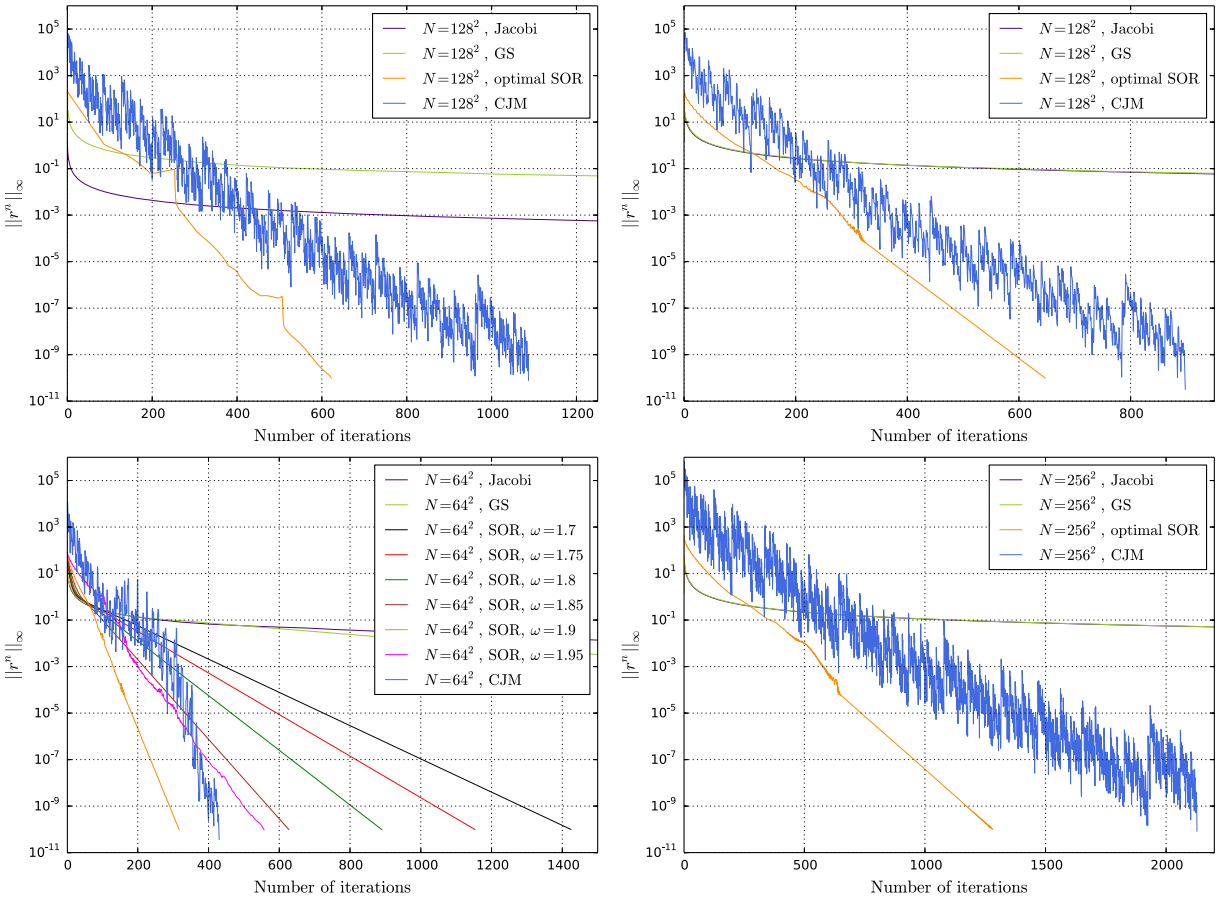

Fig. 6. Evolution of the residual for the solution of the Poisson equation (35) in 2D, with 5-points discrete Laplacian (Eq. (17); top left panel), 9-points (Eq. (23); right panels) and 17-points Laplacian (Eq. (32); bottom left panel) for different iterative methods and for the resolutions indicated in the legends. Note that the top and bottom right panels correspond to a problem set up with $N_x = N_y = 128$ and $N_x = N_y = 256$ points, respectively. A color version of the figure can be found in the online version.

Next, we numerically test the performance of the CJM for the two high-order discretizations of the Laplacian operator we have discussed above. To do so, we numerically solve the following problem:

$$\Delta u = -(x^2 + y^2)e^{xy}, \tag{35}$$

in the unit square with appropriate Dirichlet boundary conditions. The boundaries are specified easily in this case, since there exists an analytic solution for the problem at hand that we can compute at the edges of the computational domain. The analytic solution reads

$$u(x, y) = -e^{xy}. \tag{36}$$

In Fig. 6 we show the residual evolution obtained when solving problem (35) with different high-order discretizations of the Laplacian. In the top left panel we use the classical 5-points discrete approximation for the Laplacian (Eq. (17)). It is evident that our method almost reaches the performance of the optimal SOR [23]. In fact, as we prove in Appendix B this optimal weight for the SOR method coincides, up to first order with the geometrical mean of the weights obtained with our optimal scheme. In the right panels we display the evolution of the residual when solving the same problem but using the 9-point discretization of the Laplacian proposed by [14] (Eq. (23)). In the top right panel of Fig. 6, we use a mesh with 128 points in each dimension, while in the bottom right panel we use 256 points per dimension. In both cases, the performance is comparable with the optimal SOR whose weight is calculated in [14]. Finally, the left-bottom panel of Fig. 6 shows the number of iterations when solving the same problem, but using a 64^2 grid and our 17-points Laplacian (Eq. (32)), with the optimal CJM obtained with the $\kappa_{\min}$ and $\kappa_{\max}$ of Eqs. (30) and (31) (i.e., in the case $a = 1$, $b = 2$, which gives equal weight to all points in the neighborhood). In this case, the optimal weight of the SOR is unknown, so we compute the numerical solution for several values of the SOR weight. Remarkably, the CJM scheme compares fairly well with SOR.

Last but not least, we are interested in the parallel implementation of these schemes. It is known that in the case of the standard 5-points discretization of the Laplacian, one needs to implement a red-black coloring strategy for the efficient

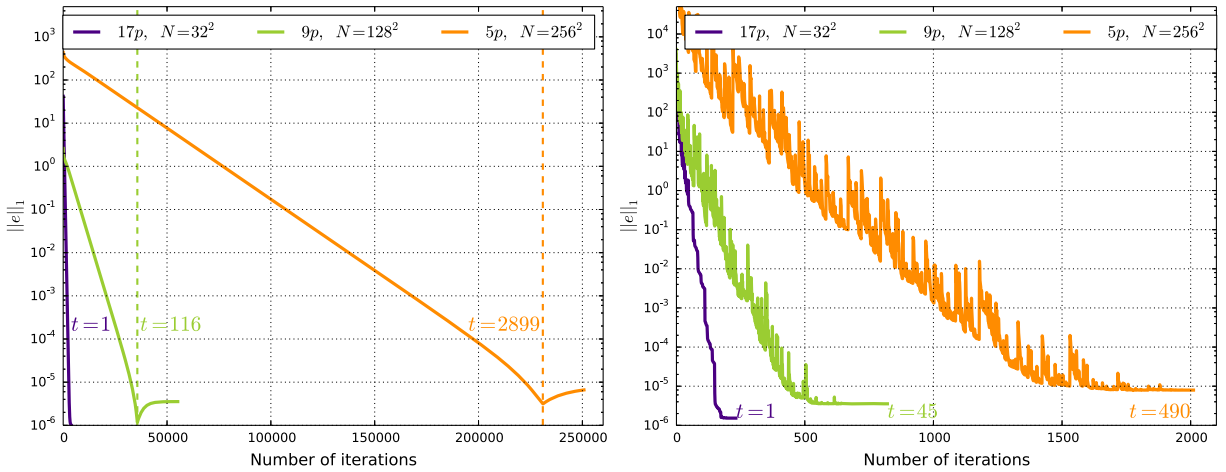

Fig. 7. Norm-1 error of the numerical solution with respect to the analytic solution (Eq. (36)) for the Jacobi method (left panel) and CJM (right panel) with different resolutions and orders of discretization of the Laplacian in Eq. (35). The numerical solution is evolved until the norm-1 error reaches, approximately, 10^{-6}. We also annotate besides each of the lines the time needed to run the model at hand normalized to the run time of the fastest case. A color version of the figure can be found in the online version.

parallel implementation of SOR. In the case of the 9-points discretization of the Laplacian, [14] points out that one needs four colors for a parallel implementation. Furthermore, the ordering strategy with more than two colors is not unique. Adams [14] find 72 different four-color orderings, which lead to different convergence rates. In contrast, our CJM scheme (as any SRJ scheme) is trivially parallelizable since there is no need for a coloring strategy and, consequently, it possesses a unique convergence rate. We find that the tiny performance difference between the SOR method, when applied to problems where the optimal weight is unknown, and the CJM is outbalanced by the simplicity in the parallelization of the latter.

The numerical solution of physical problems in computational physics often involves solving elliptic–hyperbolic PDEs when (physical) constraints need to be enforced during the evolution of the system. A popular example is the so-called *projection scheme* used to enforce the zero divergence of the magnetic field constraint in magnetohydrodynamics [24], which involves the solution of a Poisson equation. Similarly, projection schemes can be used in incompressible hydrodynamics to ensure that the zero divergence of the velocity field constraint is fulfilled (see, e.g. [25–27]). A final example is the numerical integration of the Einstein equations, where the construction of initial data involves solving the so-called constraint equations, a set of elliptic PDEs (see e.g. [28] for a detailed review). A popular way of evolving the resulting initial data is via the hyperbolic BSSN [29,30] scheme. Projection schemes obtain the same order of accuracy as the underlying base schemes [31], which means that high-order finite differences are desirable when solving elliptic PDEs associated with projection schemes in combination with high-order methods in resolving the hyperbolic evolution equations. An example for the use of a 13-point stencil for the Laplacian when using a projection scheme in incompressible fluid flows can be found in [32]. Similarly, constraint fulfilling initial data for the numerical integration in time needs to be constructed with the same spatial accuracy as the one employed in the finite difference scheme used to solve the hyperbolic evolution equations. In numerical relativity simulations, it is customary to use a fourth-order Runge–Kutta time integration, which requires at least fourth order finite differencing in spatial derivatives (see [33] for a review).

Furthermore, to discuss advantages arising from higher order discretizations, let us consider Eq. (35) once more. As we know the analytic solution to our problem (Eq. (36)), we can monitor the real error at each iteration in the computation of our numerical solution.[6] In Fig. 7 we show that a significantly higher number of grid points is needed when employing lower order discretization stencils in order to achieve approximately the same error (i.e. a solution of the same quality). With a mesh of only 32×32 points we reach the sought accuracy goal employing a discretization of the Lagrangian with a 17-points stencil. To achieve the same accuracy with our 9-point stencil discretization, about 128×128 grid zones are needed, resulting in approximately 4 to 10 times more iterations than with the 17-points stencil when using the CJM or the Jacobi schemes, respectively. In the case of the standard second-order 5-point stencil, the grid should contain more than 256×256 points and the number of iterations increases by about 60 times when applying the Jacobi method, and 10 times in CJM with respect to the number required when using the maximum order stencil. Although each step of the iterative algorithm performs more operations for higher order discretizations, this penalty is negligible compared to the considerable reduction in the number of iterations, which in turn translates into a huge decrease in the actual calculation time (see the labels of Fig. 7). Therefore, we have shown that not only the number of iterations increases when employing low order discretizations of the Laplacian, but also that the computational time needed to arrive to a prescribed norm-1 error goal is also substantially larger.

[6] In actual computations, where we do not know the real error, we monitor the residual that shall be proportional to the real error.

As a final point we note that these NCO matrices lead to more compact stencils which effectively reduce the communications in parallelizations with distributed memory (message passing paradigm).

4. Conclusions

In this work we have obtained the optimal coefficients for the SRJ method to solve linear systems arising in the finite difference discretization of elliptic problems in the case $P = M$, i.e., using each weight only once per cycle. We have proven that these are the optimal coefficients for the general case, where we fix P but allow for repetitions of the coefficients ($P \leq M$). Furthermore, we have provided a simple estimate to compute the optimal value of M to reduce the initial residual by a prescribed factor.

We have tested the performance of the method with two simple examples (in 2 and 3 dimensions), showing that the analytically derived amplification factors can be obtained in practice. When comparing the optimal $P = M$ set of coefficients with those in the literature [2,3], our method always gives better results, i.e., it achieves a larger reduction of the residual for the same number of iterations M. Additionally, the new coefficients can be computed analytically, as a function of M, $\kappa_{\max}$, and $\kappa_{\min}$, which avoids the numerical resolution of the minimization problem involved in previous works on the SRJ. The result is a numerical method that is easy to implement, and where all necessary coefficients can easily be calculated given the grid size, boundary conditions and tolerance of the elliptic problem at hand *before* the actual iteration procedure is even started.

We have found that following the same philosophy that inspired the development of SRJ methods, the case $P = M$ results in an iterative method nearly equivalent to the non-stationary Richardson method as implemented by Young [7]; namely, where the coefficients ω_n are taken to be the reciprocals of the roots of the corresponding Chebyshev polynomials in the interval bounding the spectrum of eigenvalues of the matrix (A) of the linear system. Furthermore, inspired by the same ideas as in the original SRJ methods, the actual minimum and maximum eigenvalues of A do not need to be explicitly computed. Instead, we resort to a (much simpler) von Neumann analysis of the linear system which yields the values of the $\kappa_{\min}$ and $\kappa_{\max}$ that *replace* the (larger) values of the minimum and maximum eigenvalues of A.[7] The key to our success in the practical implementation of the Chebyshev–Jacobi methods stems from a suitable ordering (or scheduling) of the weights ω_n in the algorithm. Though other orderings have also been shown to work, our choice clearly limits the growth of round-off errors when the number of iterations is large. This ordering is inherited from the SRJ schemes.

We have also tested the performance of the CJM for more than second order discretizations of the elliptic Laplacian operator. These cases are especially involved since the matrix of iteration cannot be consistently ordered. Thus, Young's theory cannot be employed to find the value of the optimal weight of a SOR scheme applied to the resulting problems. For the particular case of the 9-points discretization of the Laplacian, even though the iteration matrix cannot be consistently ordered, Adams [14] found the optimal weight for the corresponding SOR scheme in a rather involved derivation. Comparing the results for the numerical solution of a simple Poisson-like problem of the SOR method derived by Adams and the CJM we obtain here for the same 9-points discretization of the Laplacian, it is evident that both methods perform quite similarly (though the optimal SOR scheme is still slightly better). However, the SOR method requires a multi-coloring parallelization strategy with up to 72 four-color orderings (each with different performance), when applied to the 9-points discretizations of the Laplacian operator. The parallelization strategy is even more intricate when a 17-points discretization of the Laplacian is used. In contrast, CJM methods are trivially parallelizable and do not require any multi-coloring strategy. Thus, we conclude that the slightly smaller performance difference between the CJM and the SOR method in sequential applications is easily outbalanced in parallel implementations of the former method. Furthermore, we have shown that employing higher order discretizations of the Lagrangian operator is very advantageous to reduce both the number of iterations and the computational time needed to reach a preestablished real error goal (i.e., the true error one makes comparing the exact solution of a problem with the numerical approximation of it). Given the stencil increase needed to implement a 17-points discretization of the Lagrangian, we infer that a parallel implementation of this method may require a very modest increment in the number of zones transferred as internal boundaries among different computational subdomains. Hence, applying high-order discretizations of the Laplacian is ideally suited for problems that combine the solution of elliptic and hyperbolic systems of coupled equations (e.g., as in the case of Euler–Poisson systems dealing with self-gravitating fluids).

Acknowledgements

We thank Prof. P. Mulet for comments on Theorem 2 of Appendix B. We acknowledge the support from the European Research Council (Starting Independent Researcher Grant CAMAP-259276) and from the Spanish Ministerio de Economía y Competitividad through the grants SAF2013-49284-EXP, AYA2015-66899-C2-1-P, AYA2013-40979-P, as well as the partial support of the Valencian Government through the grant PROMETEO-II-2014-069. We finally acknowledge the computational time obtained from the Spanish Supercomputing Network in the local Valencian node *Tirant*.

[7] We note that many other iterative schemes rely on a dynamic choice of the relaxation parameter or on dynamic preconditioning techniques and may be applied to matrices that come from discretizations of physical problems over generic grids (see e.g. [34–38]).

Appendix A. Ordering of the weights

As we point out in Sect. 2, the ordering of the weights ω_n is the key to avoid the pile up of roundoff errors. In this appendix, we show that the ordering provided by [2] for SRJ schemes, and that we also use for the optimal $P = M$ schemes, differs from the one suggested by other authors.

Lebedev & Finogenov [17] provided orderings for the cases in which the number of weights is a power of 2. Translated to our notation, we shall have $M = 2^r$, $r = 0, 1, \ldots$. In such a case, let the ordering of the set $(\omega_1, \omega_2, \ldots, \omega_M)$ as obtained from Eq. (10), be mapped with the vector of indices $(1, 2, \ldots, M)$. Let us consider an integer permutation of the vector of indices of order M, $\Xi_M := (j_1, j_2, \ldots, j_M)$, where $(1 \leq j_k \leq M, j_i \neq j_k)$, which are constructed according to the following recurrence relation:

$$\Xi_{2^0} = \Xi_1 := (1) \quad \text{and} \quad \Xi_{2^{r-1}} := (j_1, j_2, \ldots, j_{2^{r-1}}) \tag{A.1}$$

$$\Xi_{2^r} = \Xi_M := (j_1, 2^r + 1 - j_1, j_2, 2^r + 1 - j_2, \ldots, j_{2^{r-1}}, 2^r + 1 - j_{2^{r-1}}) \tag{A.2}$$

In particular, we have,

$$\Xi_2 = (1, 2),$$

$$\Xi_4 = (1, 4, 2, 3),$$

$$\Xi_8 = (1, 8, 4, 5, 2, 7, 3, 6),$$

$$\Xi_{16} = (1, 16, 8, 9, 4, 13, 5, 12, 2, 15, 7, 10, 3, 14, 6, 11).$$

In contrast, we can obtain different SRJ schemes, and correspondingly, different orderings, for the same number of weights, because of the later depend on the number of points employed in the discretization (see Eq. (10)). Furthermore, the ordering also depends on the tolerance goal, σ (which sets the value of M; Eq. (14)). Next we list some of the orderings we can obtain for different discretizations (annotated in parenthesis in the form $N_x \times N_y$) and values of σ:

$$\Xi_2^{\text{SRJ}} = (1, 2),$$

$$\Xi_4^{\text{SRJ}} = (1, 4, 3, 2),$$

$$\Xi_8^{\text{SRJ}} = (1, 8, 5, 2, 3, 7, 4, 6) \text{ for } (4 \times 4, \sigma = 0.01),$$

$$(1, 8, 5, 3, 6, 2, 7, 4) \text{ for } (8 \times 8, \sigma = 0.15),$$

$$\Xi_{16}^{\text{SRJ}} = (1, 15, 9, 2, 12, 3, 4, 13, 5, 6, 7, 8, 10, 11, 14, 16) \text{ for } (4 \times 4, \sigma = 2 \times 10^{-5}),$$

$$(1, 16, 9, 6, 12, 3, 14, 7, 10, 4, 13, 5, 15, 8, 2, 11) \text{ for } (8 \times 8, \sigma = 6 \times 10^{-3}),$$

which obviously differ from the orderings Ξ_j for $j \geq 4$.

We note that [18] provided also orderings for arbitrary values of M, which coincide with those of Lebedev & Finogenov [17] when M is a power of 2 (i.e., $M = 2^r$). Finally, more recently, Lebedev & Finogenov [19] have extended their previous work to a larger number of cases (e.g., $M = 2^r 3^s$) and applied also to Chebyshev iterative methods. We remark that the SRJ ordering of the weights can be applied to arbitrary values of M.

Appendix B. Properties of the weights

In this appendix we show some algebraic properties of the weights of the CJM. The first one is that the harmonic mean of the weights equals the average of the maximum and minimum weight numbers:

Theorem 1. *Let ω_i be the weights given by Eq. (10). Then it holds that*

$$\frac{1}{M} \sum_{i=1}^{M} \omega_i^{-1} = \frac{\kappa_{\max} + \kappa_{\min}}{2}. \tag{B.1}$$

Proof.

$$\frac{1}{M} \sum_{i=1}^{M} \omega_i^{-1} = \frac{(\kappa_{\max} + \kappa_{\min})}{2} - \frac{(\kappa_{\max} - \kappa_{\min})}{2M} \sum_{i=1}^{M} \cos\left(\frac{\pi (i - 1/2)}{M}\right). \tag{B.2}$$

Let $j \in [1, M/2]$. Since

$$\cos\left(\frac{\pi (j - 1/2)}{M}\right) = -\cos\left(\frac{\pi ((M - j + 1) - 1/2)}{M}\right), \tag{B.3}$$

all the terms in the summation cancel out, except the central one in case M is odd. In this last case, $M = 2n+1$, and the only remaining term is $\cos\left(\frac{\pi(n+1/2)}{2n+1}\right) = \cos\left(\frac{\pi}{2}\right) = 0$. In general, the summation reads

$$\frac{1}{M}\sum_{i=1}^{M}\omega_i^{-1} = \frac{\kappa_{\max}+\kappa_{\min}}{2}. \qquad \square \tag{B.4}$$

Corollary. *Since the relation between the weights of the stationary RM and the CJM is $\hat{\omega} = \omega d^{-1}$, where $D = \mathrm{diag}(A)$, having all its elements equal to d, and since $\hat{\omega} = 2/(a+b)$, where $a = \min(\lambda_i)$ and $b = \max(\lambda_i)$, being λ_i the eigenvalues of matrix A, it turns out that*

$$\frac{2d^{-1}}{\kappa_{\max}+\kappa_{\min}} = \frac{2}{a+b} = \hat{\omega}. \tag{B.5}$$

Theorem 2. *Let ω_i be the weights given by Eq. (10). Then it holds that*

$$\lim_{n\to+\infty}\left[\prod_{i=1}^{n}\omega_i^{-1}\right]^{1/n} = \left(\frac{\sqrt{\kappa_{\max}}+\sqrt{\kappa_{\min}}}{2}\right)^2. \tag{B.6}$$

Proof. Let us define $A = (\kappa_{\max}+\kappa_{\min})/2$ and $B = (\kappa_{\max}-\kappa_{\min})/2$. It is well known that the Chebyshev polynomials of first kind of degree n, $T_n(x)$, satisfy the following recurrence relation:

$$T_0(x) = 1; \quad T_1(x) = x; \quad T_n(x) = 2xT_n(x) - T_{n-1}(x), n > 2. \tag{B.7}$$

From this property, it is easy to check that the leading coefficient of $T_n(x)$ is 2^{n-1}. Taking into account the leading coefficient and the roots of $T_n(x)$ from Eq. (8), we get that

$$T_n(x) = 2^{n-1}\prod_{k=1}^{n}\left\{x - \cos\left[\frac{(2k-1)\pi}{2n}\right]\right\}. \tag{B.8}$$

Therefore,

$$\begin{aligned}\prod_{i=1}^{n}\omega_i^{-1} &= \left[A - B\cos\left(\frac{\pi}{2n}\right)\right]\left[A - B\cos\left(\frac{3\pi}{2n}\right)\right]\ldots\left[A - B\cos\left(\frac{(2n-1)\pi}{2n}\right)\right] \\ &= \frac{B^n}{2^{n-1}}T_n\left(\frac{A}{B}\right) = \frac{B^n}{2^n}\left[\left(\frac{A}{B} - \sqrt{\frac{A^2}{B^2}-1}\right)^n + \left(\frac{A}{B} + \sqrt{\frac{A^2}{B^2}-1}\right)^n\right],\end{aligned} \tag{B.9}$$

where last equality uses the explicit expression of $T_n(x)$ for $x = A/B > 1$. From this equality, we can bound the geometrical mean of the inverse of the weights ω_i:

$$\frac{B}{2}\left(\frac{A}{B} + \sqrt{\frac{A^2}{B^2}-1}\right) \le \left(\prod_{i=1}^{n}\omega_i^{-1}\right)^{1/n} \le \frac{B}{2^{(n-1)/n}}\left(\frac{A}{B} + \sqrt{\frac{A^2}{B^2}-1}\right). \tag{B.10}$$

Taking limits for $n \to \infty$, we obtain that

$$\lim_{n\to+\infty}\left[\prod_{i=1}^{n}\omega_i^{-1}\right]^{1/n} = \frac{1}{2}\left(A + \sqrt{A^2-B^2}\right) = \left(\frac{\sqrt{\kappa_{\max}}+\sqrt{\kappa_{\min}}}{2}\right)^2. \qquad \square \tag{B.11}$$

References

[1] C. Jacobi, Über 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.

[2] 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, http://www.sciencedirect.com/science/article/pii/S0021999114004173.

[3] J.E. Adsuara, I. Cordero-Carrión, P. Cerdá-Durán, M.A. Aloy, Scheduled Relaxation Jacobi method: Improvements and applications, J. Comput. Phys. 321 (2016) 369–413, http://dx.doi.org/10.1016/j.jcp.2016.05.053.

[4] W. Markoff, Über Poynome, die in einem gegeben Intervalle möglichst wenig von Null abweichen, Math. Ann. 77 (1916) 213–258.

[5] D.M. Young, Iterative Solution of Large Linear Systems, dover ed edition, Dover Books on Mathematics, Dover Publications, 1971, http://gen.lib.rus.ec/book/index.php?md5=1b9a661e5563daba113b7ffc30d26c18.

[6] M.H. Gutknecht, S. Röllin, The Chebyshev iteration revisited, Parallel Comput. 28 (2) (2002) 263–283, http://dx.doi.org/10.1016/S0167-8191(01)00139-9, http://www.sciencedirect.com/science/article/pii/S0167819101001399.
[7] D. Young, On Richardson's method for solving linear systems with positive definite matrices, J. Math. Phys. 32 (1) (1953) 243–255, http://dx.doi.org/10.1002/sapm1953321243.
[8] W.L. Frank, Solution of linear systems by Richardson's method, J. ACM 7 (3) (1960) 274–286, http://dx.doi.org/10.1145/321033.321041, http://doi.acm.org/10.1145/321033.321041.
[9] G. Shortley, Use of Tschebyscheff-polynomial operators in the numerical solution of boundary-value problems, J. Appl. Phys. 24 (4) (1953) 392–396, http://dx.doi.org/10.1063/1.1721292, http://scitation.aip.org/content/aip/journal/jap/24/4/10.1063/1.1721292.
[10] 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, Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 210 (1911) 307–357, http://dx.doi.org/10.1098/rsta.1911.0009.
[11] D. Young, On Richardson's method for solving linear systems with positive definite matrices, J. Math. Phys. 3 (1954) 243–255.
[12] G. Opfer, G. Schober, Richardson's iteration for nonsymmetric matrices, Linear Algebra Appl. 58 (1984) 343–361, http://dx.doi.org/10.1016/0024-3795(84)90219-2, http://www.sciencedirect.com/science/article/pii/0024379584902192.
[13] D. Young, Iterative Methods for Solving Partial Difference Equations of Elliptic Type, Ph.D. thesis, Harvard University, Mathematics Department, Cambridge MA, USA, 1950.
[14] L.M. Adams, R.-J. LeVeque, D. Young, Analysis of the SOR iteration for the 9-point Laplacian, SIAM J. Numer. Anal. 25 (5) (1988) 1156–1180, http://dx.doi.org/10.1137/0725066.
[15] D. Young, On the solution of linear systems by iteration, in: Proc. Sixth Symp. in Appl. Math., vol. 6, Amer. Math. Soc., 1956, pp. 283–298.
[16] R.S. Anderssen, G.H. Golub, Richardson's Non-stationary Matrix Iterative Procedure, Tech. Rep. STAN-CS-72-304, Computer Science Dept., Stanford Univ., 1972, http://i.stanford.edu/pub/cstr/reports/cs/tr/72/304/CS-TR-72-304.pdf.
[17] V.I. Lebedev, S.A. Finogenov, On the order of choice of the iteration parameters in the Chebyshev cyclic iteration method, Zh. Vychisl. Mat. Mat. Fiz. 11 (1) (1971) 425–438;
English translation in R.S. Anderssen, G.H. Golub, Richardson's Non-stationary Matrix Iterative Procedure, Rep. STAN-CS-72-304, Computer Science Dept., Stanford Univ., 1972.
[18] E.S. Nikolaev, A.A. Samarskii, Selection of the iterative parameters in Richardson's method, Zh. Vychisl. Mat. Mat. Fiz. 12 (4) (1972) 960–973, English translation by J. Berry.
[19] V.I. Lebedev, S.A. Finogenov, On construction of the stable permutations of parameters for the Chebyshev iterative methods. Part I, Russ. J. Numer. Anal. Math. Model. 17 (5) (2002) 437–456.
[20] Einstein Toolkit, http://einsteintoolkit.org/.
[21] F. Löffler, J. Faber, E. Bentivegna, T. Bode, P. Diener, R. Haas, I. Hinder, B.C. Mundim, C.D. Ott, E. Schnetter, G. Allen, M. Campanelli, P. Laguna, The Einstein Toolkit: a community computational infrastructure for relativistic astrophysics, Class. Quantum Gravity 29 (11) (2012) 115001, http://dx.doi.org/10.1088/0264-9381/29/11/115001, arXiv:1111.3344.
[22] R.J. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-state and Time-dependent Problems, vol. 98, SIAM, 2007.
[23] R.-J. LeVeque, L. Trefethen, Fourier analysis of the SOR iteration, IMA J. Numer. Anal. 8 (3) (1988) 273–279, http://dx.doi.org/10.1093/imanum/8.3.273.
[24] J.U. Brackbill, D.C. Barnes, The effect of nonzero product of magnetic gradient and B on the numerical solution of the magnetohydrodynamic equations, J. Comput. Phys. 35 (1980) 426–430, http://dx.doi.org/10.1016/0021-9991(80)90079-0.
[25] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (104) (1968) 745–762.
[26] J.B. Bell, P. Colella, H.M. Glaz, A second order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (1989) 257–283, http://dx.doi.org/10.1016/0021-9991(89)90151-4.
[27] A.S. Almgren, J.B. Bell, W.G. Szymczak, A numerical method for the incompressible Navier–Stokes equations based on an approximate projection, SIAM J. Sci. Comput. 17 (2) (1996) 358–369.
[28] G. Cook, Initial data for numerical relativity, Living Rev. Relativ. 3 (2000), http://dx.doi.org/10.12942/lrr-2000-5, arXiv:gr-qc/0007085.
[29] T.W. Baumgarte, S.L. Shapiro, Numerical integration of Einstein's field equations, Phys. Rev. D 59 (2) (1999) 024007, http://dx.doi.org/10.1103/PhysRevD.59.024007, arXiv:gr-qc/9810065.
[30] M. Shibata, T. Nakamura, Evolution of three-dimensional gravitational waves: harmonic slicing case, Phys. Rev. D 52 (1995) 5428–5444, http://dx.doi.org/10.1103/PhysRevD.52.5428.
[31] G. Tóth, The $\nabla \cdot B = 0$ constraint in shock-capturing magnetohydrodynamics codes, J. Comput. Phys. 161 (2000) 605–652, http://dx.doi.org/10.1006/jcph.2000.6519.
[32] A.S. Almgren, A. Aspden, J.B. Bell, M.L. Minion, On the use of higher-order projection methods for incompressible turbulent flow, SIAM J. Sci. Comput. 35 (1) (2013) B25–B42.
[33] J. Centrella, J.G. Baker, B.J. Kelly, J.R. van Meter, Black-hole binaries, gravitational waves, and numerical relativity, Rev. Mod. Phys. 82 (2010) 3069–3119, http://dx.doi.org/10.1103/RevModPhys.82.3069, arXiv:1010.5260.
[34] Y. Saad, M. Schultz, Conjugate gradient-like algorithms for solving nonsymmetric linear systems, Math. Comput. 44 (1985) 417–424.
[35] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd edition, Society for Industrial and Applied Mathematics, 2003, http://epubs.siam.org/doi/abs/10.1137/1.9780898718003.
[36] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. 14 (2) (1985) 461–469.
[37] M. Dehghan, M. Hajarian, Improving preconditioned SOR-type iterative methods for L-matrices, Int. J. Numer. Methods Biomed. Eng. 27 (5) (2011) 774–784.
[38] M. Antuono, G. Colicchio, Delayed over-relaxation for iterative methods, J. Comput. Phys. 321 (2016) 892–907.