Computers and Mathematics with Applications 67 (2014) 2088–2098

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Iterative across-time solution of linear differential equations: Krylov subspace versus waveform relaxation☆

M.A. Botchev [a,*], I.V. Oseledets [b,c], E.E. Tyrtyshnikov [c,d,e,f]

[a] Department of Applied Mathematics and MESA+ Institute for Nanotechnology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
[b] Skolkovo Institute of Science and Technology, Novaya St. 100, Skolkovo, Odintsovsky District, 143025 Moscow Region, Russia
[c] Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkin St. 8, Moscow 119333, Russia
[d] Lomonosov Moscow State University, Moscow 119991, Russia
[e] Moscow Institute of Science and Technology, Russia
[f] University of Siedlce, Poland

ARTICLE INFO

Article history:
Available online 6 April 2014

Keywords:
Krylov subspace methods
Waveform relaxation
Matrix exponential
Low rank approximation
Residual
Anderson acceleration

ABSTRACT

The aim of this paper is two-fold. First, we propose an efficient implementation of the continuous time waveform relaxation (WR) method based on block Krylov subspaces. Second, we compare this new WR–Krylov implementation against Krylov subspace methods combined with the shift and invert (SAI) technique. Some analysis and numerical experiments are presented. Since the WR–Krylov and SAI–Krylov methods build up the solution simultaneously for the whole time interval and there is no time stepping involved, both methods can be seen as iterative across-time methods. The key difference between these methods and standard time integration methods is that their accuracy is not directly related to the time step size.

1. Introduction

An important sub-task frequently arising in the numerical solution of partial differential equations (PDEs) is the solution of the following initial value problem (IVP):

$$y'(t) = -Ay(t), \qquad y(0) = v, \quad t \in [0, T], \ A \in \mathbb{R}^{n\times n}. \tag{1}$$

Here A is typically very large and sparse. Note that (1) is equivalent to the problem of computing the action of the matrix exponential [1,2]:

$$y(t) = \exp(-tA)v, \quad t \in [0, T]. \tag{2}$$

Krylov subspace methods have been successfully used for the evaluation of the matrix exponential and for the numerical solution of various time dependent problems since the late 80s. We mention in chronological order some early pa-

☆ Supported by the Russian Federal Program "Scientific and scientific-pedagogical personnel of innovative Russia", grant 8500, by the RFBR grants 12-01-00565-a, 12-01-91333, 13-01-12061, 14-01-00804 and is based on work funded by Skolkovo Institute of Science and Technology (SkolTech) within the framework of the SkolTech/MIT Initiative.

* Corresponding author. Tel.: +31 534893396.
E-mail addresses: m.a.botchev@utwente.nl (M.A. Botchev), ivan.oseledets@gmail.com (I.V. Oseledets), eugene.tyrtyshnikov@gmail.com (E.E. Tyrtyshnikov).

http://dx.doi.org/10.1016/j.camwa.2014.03.002
0898-1221/

pers [3–10]. For a more recent work see the surveys in [11] and in corresponding chapters of [2,12]. These methods are based on a projection of the original IVP (1) onto the Krylov subspace

$$\mathcal{K}_m(A, w) = \mathrm{span}(w, Aw, A^2w, \ldots, A^{m-1}w),$$

where usually $w = v$ or $w = Av$ holds. A significant part of the computational work in Krylov subspace methods is spent for building up a basis of $\mathcal{K}_m(A, w)$, which is usually done by the Arnoldi or Lanczos process [13,14]. The process computes the columns $v_1, v_2, \ldots, v_m$ of $V_m \in \mathbb{R}^{n\times m}$ which form an orthonormal basis of $\mathcal{K}_m(A, w)$ and $v_1 = w/\|w\|$. The matrix V_m satisfies the so-called Arnoldi decomposition [13,14], namely,

$$AV_m = V_{m+1}H_{m+1,m} = V_mH_{m,m} + v_{m+1}h_{m+1,m}e_m^T, \quad \mathbb{R}^m \ni e_m = (0, \ldots, 0, 1)^T,$$

where $H_{m+1,m} = V_{m+1}^TAV_m \in \mathbb{R}^{(m+1)\times m}$ and $H_{m,m} = V_m^TAV_m \in \mathbb{R}^{m\times m}$ are upper Hessenberg and $h_{m+1,m}$ is the only nonzero entry in the last row of $H_{m+1,m}$. Furthermore, if the Krylov subspace method converges successfully then for some $m \ll n$ it holds

$$AV_m \approx V_mH_{m,m},$$

i.e., the colspan of V_m is an approximate invariant subspace of A.

An attractive feature of the method is that in some situations it suffices to build up just a single Krylov basis for the whole time interval of interest $t \in [0, T]$. Indeed, with $w = v$ an approximate solution $y_m(t)$ to problem (1) can be computed as

$$y(t) = \exp(-tA)v = \exp(-tA)V_m\beta e_1 \approx \underbrace{V_m \exp(-tH_{m,m})\beta e_1}_{y_m(t)}, \quad t \in [0, T], \tag{3}$$

where $\beta = \|w\|$ and $\mathbb{R}^m \ni e_1 = (1, 0, \ldots, 0)^T$. The approximation y_m in (3) should satisfy

$$\|y(t) - y_m(t)\| \leqslant \text{tolerance}, \quad t \in [0, T],$$

which can be checked in practice by some error estimates, for example, with the help of the exponential residual defined as [15–17]

$$r_m(t) \equiv -Ay_m(t) - y_m'(t). \tag{4}$$

The property of having a single Krylov basis for the whole time interval makes the methods computationally efficient. In some cases this property can be extended [18] to a more general IVP

$$y'(t) = -Ay(t) + g(t), \qquad y(0) = v, \quad t \in [0, T], \ A \in \mathbb{R}^{n\times n} \tag{5}$$

where $g : [0, T] \to \mathbb{R}^n$ is a given function. Furthermore, this property allows one to regard the Krylov subspace methods applied in this setting as, to some extent, *time stepping free* methods.

Remark 1. Here we use the term "time stepping free" to indicate that the accuracy does not depend on a time step Δt, as is the case for the standard time integration solvers such as Runge–Kutta or multistep methods. One should emphasize that this independence on the time step is partial, i.e., the efficiency does depend on the length of the time interval (typically, the smaller T, the smaller Krylov dimension m suffices [19,9]).

If A is close to a symmetric positive definite matrix with a stiff spectrum,[1] convergence of Krylov subspace methods can be slow. In this case the performance can often be dramatically improved with rational Krylov subspace methods [22,12], in particular by switching to the shift-and-invert (SAI) Krylov subspace $\mathcal{K}((I+\gamma A)^{-1}, w)$ [23,24], where $\gamma > 0$ is a parameter related to T. The price for the faster convergence is that in these methods a system with the matrix $I + \gamma A$ has to be solved at each Krylov step.

Another useful property of the Krylov subspace methods is that they can be applied to solve (1) iteratively. More specifically, assume we have an approximation $y_k(t) \approx y(t)$ for which the residual $r_k(t)$, defined by (4), is known (here we intentionally changed the subindex from m to k). Then a better approximation can be obtained by the following iteration:

(a) find an approximate solution $\xi_k(t)$ of $\begin{cases} \xi'(t) = -A\xi(t) + r_k(t), \\ \xi(0) = 0, \end{cases}$ (6)

(b) update $y_{k+1}(t) = y_k(t) + \xi_k(t)$. (7)

Clearly, if the correction problem (6) is solved exactly then the iteration converges to the exact solution $y(t)$ after one step. One possible option is to solve (6) by a Krylov subspace method: at each iteration k, a number m of Krylov iterations are applied to solve (6) approximately. In fact this can be seen as an efficient restarting procedure for the Krylov subspace methods [15,17]. See also related work on Krylov subspace methods and restarting [25–28,12].

[1] Following [20, p. 36], by a matrix with a stiff spectrum we understand a matrix A such that implicit time integrators perform much better for $y' = -Ay$ than explicit ones. For a more formal definition of stiffness see [21].

Another possibility is to choose a matrix $M \in \mathbb{R}^{n\times n}$, $M \approx A$ and take ξ_k to be the solution of the approximate correction problem

$$\begin{cases} \xi_k'(t) = -M\xi_k(t) + r_k(t), \\ \xi_k(0) = 0. \end{cases} \tag{8}$$

In this paper we follow an approach where this approximate correction problem is solved "exactly", i.e., to a high accuracy rather than by a time stepping method with some non-negligible time error. The iteration (7), (8) can then be seen as the continuous time waveform relaxation (WR) method [29–31].

The aim of the paper is two-fold. First, for A with a sectorial spectrum, we show how iteration (7), (8) can be efficiently implemented with the help of block SAI–Krylov subspace and low rank approximation techniques, while allowing a broad choice of M. Second, we compare the performance of this WR–Krylov method with that of the SAI–Krylov method (i.e.,the iteration (6), (7) where the correction problem is solved by the SAI–Krylov projection). The two approaches are related as they both essentially involve two components: SAI–Krylov subspace projection and an approximation $M \approx A$. In the SAI–Krylov method the approximation M can be used as a preconditioner for solving the SAI-systems with the matrix $I+\gamma A$ (more specifically, the preconditioner is then $I + \gamma M$).

It is instructive to compare these two approaches as the comparison contributes to answering the old important question [9,32,24]: how to use some knowledge on A, available as a "simpler" matrix M, for solving (1) or (5)? The first approach, the WR–Krylov method, does so directly on the differential equation level, solving the differential equation with M. On the other hand, the second approach, SAI–Krylov, employs M on the linear algebra level.

Furthermore, both methods are attractive because they both are iterative across-time methods and, as such, are often efficient. Moreover, they both are good candidates for the construction of efficient time parallel methods (a subject of further research).

The structure of this paper is as follows. In Section 2 we present a block Krylov subspace implementation of WR methods. Its essential component is a low rank representation of the residual $r_k(t)$, which are briefly discussed in Section 3. The convergence of WR schemes is analyzed in Section 4. In Section 5 numerical tests that compare the WR methods with the SAI–Krylov subspace methods are presented, and the results of the tests are discussed. Finally, we draw some conclusions in Section 6.

2. WR–Krylov: a low rank block Krylov subspace implementation of WR methods

In this section we describe how the continuous time WR methods [29–31] can be implemented by block SAI–Krylov subspace and low rank approximation techniques.

We consider the IVP (1) and assume throughout the paper that the matrix A has a dominant positive definite symmetric part ($\|A+A^T\| > \|A-A^T\|$), so that the field of values of A lies in a sector in a complex plane around the positive real semiaxis. In this case the SAI–Krylov subspace method [23,24] appears to be a fast solver for (1), provided that linear systems with the matrix $I + \gamma A$ can be solved efficiently. Here the parameter $\gamma > 0$ is chosen depending on the final time T [24].

In many real life situations, e.g., when A stems from a spatial discretization of a three-dimensional partial differential operator, A is very large and sparse. However, in this case sparse direct solvers tend to be inefficient for solving systems with $I+\gamma A$. Iterative solution of the systems with $I+\gamma A$ is then an attractive option, especially if a preconditioner $I+\gamma M \approx I+\gamma A$ is available. As outlined in the introduction, one of the issues we address in this paper is whether this knowledge of $M \approx A$ can be used for solving (1) in another way, namely with the WR iteration (7), (8).

This WR method can be implemented as follows. First, the parameter $\gamma > 0$ is set (normally, $\gamma = T/10$, see [24]) and a suitable preconditioner $I + \gamma M$ for the matrix $I + \gamma A$ is chosen.

Remark 2. We emphasize that the preconditioner M can be chosen in any of many possible ways, as done for linear system solution [13,14,33,34]. The freedom in the choice of M is thus not restricted by the requirement that the IVP (8) should be easily solvable, as normally the case in conventional WR methods [29,30]. In the numerical experiments section we explore, in the context of three different test problems, a lower order finite volume approximation (Section 5.1), incomplete LU (ILU) factorization (Section 5.2) and a saddle point Schur complement preconditioner (Section 5.3).

Assume for instance that an ILU preconditioner is chosen which has the form

$$I + \gamma M = LU \approx I + \gamma A,$$

with L and U being lower and upper triangular matrices, respectively. The matrix M is then set to

$$M = \frac{1}{\gamma}(LU - I).$$

We now take the initial guess $y_0(t) = v$ and notice that the initial residual is $r_0(t) = -Av$ (cf. (4)). It will be clear from the further discussion that other choices for $y_0(t)$ are possible, as soon as

$$r_0(t) = U_0 p_0(t), \quad U_0 \in \mathbb{R}^{n\times q},\ q \ll n, \tag{9}$$

with $p_0(t) : \mathbb{R} \to \mathbb{R}^q$ being some simple function such as polynomial. Obviously, $r_0(t) = -Av$ fits into the form (9) with $q = 1$. Furthermore, without loss of generality, we assume that U_0 has orthonormal columns; this can always be achieved by carrying out the thin QR factorization of U_0.

We now solve the correction IVP (8) to a high accuracy by a SAI–Krylov method for the matrix M. This can be done cheaply: the main costs are the solution of the SAI systems with the matrix $I + \gamma M$ for which the LU factorization $I + \gamma M = LU$ has already been computed. We carry out the projection of (8) onto the Krylov subspace in a block way, namely, the subspace is

$$\mathcal{K}_m((I + \gamma M)^{-1}, U_0) = \mathrm{span}(U_0, (I + \gamma M)^{-1}U_0, \ldots, (I + \gamma M)^{-(m-1)}U_0)$$

and is spanned by the columns of the matrix

$$V_{[m]} = \begin{bmatrix} V_1 & \cdots & V_m \end{bmatrix} \in \mathbb{R}^{n\times mq}, \quad V_i \in \mathbb{R}^{n\times q},\ i = 1, \ldots, m, \tag{10}$$

where $V_1 = U_0$. The matrix $V_{[m]}$ satisfies the block Arnoldi (Lanczos) decomposition [13,14]

$$AV_{[m]} = V_{[m+1]}H_{[m+1,m]} = V_{[m]}H_{[m,m]} + V_{m+1}H_{m+1,m}E_m^T, \tag{11}$$

where $H_{[m+1,m]} \in \mathbb{R}^{(m+1)q\times mq}$ is a block upper Hessenberg matrix, with blocks $H_{ij} \in \mathbb{R}^{q\times q},\ i = 1, \ldots, m+1,\ j = 1, \ldots, m$. The matrix $H_{[m,m]}$ is formed by the first mq rows of $H_{[m+1,m]}$, $H_{m+1,m} \in \mathbb{R}^{q\times q}$ is the only nonzero block in the last $(m+1)$st block row of $H_{[m+1,m]}$ and $E_m \in \mathbb{R}^{n\times q}$ consists of the last q columns of the $mq \times mq$ identity matrix.

Let $\xi_{k,m}(t)$ be the mth iterand of this inner block iterative process to obtain solution $\approx \xi_k(t)$ to problem (8) at outer iteration k. Consider the residual of $\xi_{k,m}(t)$,

$$\tilde{r}_m(t) \equiv -M\xi_{k,m}(t) + r_k(t) - \xi'_{k,m}(t). \tag{12}$$

Then it can be shown [18] that

$$\tilde{r}_m(t) = \frac{1}{\gamma}(I + \gamma M)V_{m+1}\tilde{H}_{m+1,m}E_m^T\tilde{H}_{[m,m]}^{-1}u_{[m]}(t), \tag{13}$$

where $u_{[m]}(t) : \mathbb{R} \to \mathbb{R}^{mq}$ is the solution of IVP (8) projected onto the block SAI–Krylov subspace. This small projected IVP is usually solved by a standard ODE (ordinary differential equation) solver. For more details on this block method and its implementation see [18].

If m steps SAI–Krylov steps are done to compute the correction $\xi_k(t)$, then $\xi_k(t)$ is available as $V_{[m]}u_{[m]}(t)$. Next, the s time samples $\xi_k(t_1), \ldots, \xi_k(t_s)$ of the correction $\xi_0(t)$ are stored.

The next iterand $y_1(t)$ can now be obtained as prescribed by (7). It is not difficult to see that for the residual of the updated solution $y_{k+1}(t) = y_k(t) + \xi_k(t)$ holds (see e.g. [35,17])

$$r_{k+1}(t) = (M - A)\xi_k(t), \quad k \geqslant 0.$$

This relation is used to check the accuracy of the computed $y_{k+1}(t)$ as a stopping criterion. The iteration (7), (8) can be continued provided that the low rank form (9) still holds, possibly with a different q, for the new residual $r_{k+1}(t)$, namely

$$r_{k+1}(t) = U_{k+1}p_{k+1}(t), \quad U_{k+1} \in \mathbb{R}^{n\times q},\ q \ll n, \tag{14}$$

where the notation is as in (9). In general, the low rank representation (14) cannot be guaranteed exactly and we settle for an approximation

$$\begin{aligned} &r_{k+1}(t) \approx U_{k+1}p_{k+1}(t), \quad U_{k+1} \in \mathbb{R}^{n\times q},\ q \ll n, \\ &\|r_{k+1}(t) - U_{k+1}p_{k+1}(t)\| \leqslant \epsilon, \quad t = t_1, \ldots, t_s, \end{aligned} \tag{15}$$

with ϵ being a certain tolerance. Different low rank approximation techniques can be used to obtain (15) and we briefly discuss them in Section 3.

The WR–Krylov iteration presented in this section is outlined in Fig. 1. We note that this iteration can be seen as a preconditioned Richardson (or Picard) iteration, as it essentially uses the residual notion (4). As such, it can be accelerated using the Anderson mixing approach [36]. We have implemented this approach for our WR iteration. In the numerical experiments we have carried out so far, we see little to none improvement as compared to a non-accelerated iteration, both in terms of CPU time and a number of iterations.

3. Low rank approximation

In this section a low rank approximation technique is briefly described, which is used in this paper to build up the approximation (15).

It is based on the widely known truncated singular value decomposition (SVD), see e.g. [37]. Assume that the time samples $r_{k+1}(t_i) = (M - A)\xi_k(t_i),\ i = 1, \ldots, s$, are arranged columnwise in a matrix $R \in \mathbb{R}^{n\times s}$. We compute a thin SVD of R by first carrying out a thin QR factorization on R, followed by the SVD of the QR triangular factor of size $s \times s$. Each of the s samples $r_{k+1}(t_i)$ is now represented as a linear combination of the columns of the first SVD factor, which we denote by U_{k+1}. The

For given $A \in \mathbb{R}^{n\times n}$, $v \in \mathbb{R}^n$, $T > 0$ this method solves IVP $y' = -Ay$, $y(0) = v$

choose number of samples s (usually 20 to 40)
choose points t_i, $i = 1, \ldots, s$, such that $t_1 = 0 < t_1 < \cdots < t_s = T$
set γ (usually $\gamma = T/10$, see [54])
construct a preconditioner $LU :\approx I + \gamma A$
set $M := \frac{1}{\gamma}(LU - I)$
set $y_0(t) := v$, $r_0 := -Av$, $U_0 := r_0/\|r_0\|$, $p_0(t) := \|r_0\|$
for $k = 0, 1, 2, \ldots$
    solve IVP $\begin{cases} \xi_k'(t) = -M\xi_k + U_k p_k(t), \\ \xi_k(0) = 0 \end{cases}$ by block SAI-Krylov method,
        using $LU = I + \gamma M$ as a direct SAI solver
    compute s residual time samples $r_{k+1}(t_i) = (M - A)\xi_k(t_i)$, $i = 1, \ldots, s$
    stop if $\|r_{k+1}(t_i)\|$, $i = 1, \ldots, s$, small enough
    based on residual samples, set up a low rank approximation $U_{k+1}p_{k+1}(t) :\approx r_{k+1}(t)$
endfor

Fig. 1. The low rank block SAI–Krylov subspace implementation of the WR iteration.

coefficients of these s linear combinations are determined by the other two SVD factors, which are both $s \times s$ matrices. These coefficients can be seen as values of an unknown vector function whose values are known for discrete times $t_1, \ldots, t_s$. Interpolation of these values yields then a vector polynomial function $p_{k+1}(t)$. For more details on this procedure we refer to [18]. We emphasize that other methods than the SVD can be used to obtain (15), for example, the cross approximation algorithm and related techniques [38,39].

4. Convergence of the WR–Krylov iteration

Convergence of the WR iteration is studied e.g. in [30,31,17]. In this section we first formulate a result from [17] on convergence of the *exact* WR iteration (7), (8), i.e., the iteration where the correction IVP (8) is solved exactly. Then we extend the convergence result to the case when the correction IVP is solved approximately.

We now set up some notation and assumptions. Following [40,17], we assume that there exist constants $C_A > 0$ and $\omega \geqslant 0$ such that

$$\|\exp(-tA)\| \leqslant C_A e^{-t\omega}, \quad t \geqslant 0. \tag{16}$$

This bound is satisfied in the spectral norm for $C_A = 1$ provided that A has its field of values contained in the complex half plane $\mathbb{C}_\omega := \{z \in \mathbb{C}: \operatorname{Re} z \geqslant \omega\}$. We then have $\omega = -\mu(-A)$, where $\mu(\cdot)$ is the logarithmic matrix norm [41,20,2]. The inequality (16) holds if A is diagonalizable. Indeed, if $X^{-1}AX = \Lambda$, then $C_A = \kappa(X) = \|X\|\,\|X^{-1}\|$ for any operator norm $\|\cdot\|$ as soon as the spectrum of A is contained in $\mathbb{C}_\omega$. For further discussion on how C_A and ω can be obtained in practice we refer to [20, Chapter I.2.3].

Since $M \approx A$, we also assume the existence of constants $C_M > 0$ and $\tilde{\omega} \geqslant 0$ such that

$$\|\exp(-tM)\| \leqslant C_M e^{-t\tilde{\omega}}, \quad t \geqslant 0. \tag{17}$$

In practice, estimates for the constants C_M and $\tilde{\omega}$ can be obtained based on the knowledge of the preconditioner[2] and its logarithmic norm [20, Chapter I.2.3]. For instance, if $-A$ is an M-matrix and M is based on a regular splitting of $I + \gamma A$ then logarithmic norms of A and M can be easily computed in the infinity- or 1-norms. If A is close to a symmetric matrix then either the "energy equivalence" approach [43] or the P-regular splitting technique [44] can be used, which link the spectral boundaries of M to those of A. Examples on how the spectral boundaries of M can be estimated for a nonsymmetric A can be found e.g. in [45–47].

Next, we introduce the functions φ_k (see e.g. [40]):

$$\varphi_k(z) = \int_0^1 e^{(1-\theta)z} \frac{\theta^{k-1}}{(k-1)!} d\theta, \quad k \geqslant 1. \tag{18}$$

It is not difficult to see that $\varphi_k(0) = 1/k!$ and

$$\varphi_{k+1}(z) = \frac{\varphi_k(z) - \varphi_k(0)}{z}, \qquad \varphi_0(z) = e^z. \tag{19}$$

[2] Indeed, the analysis of many preconditioners [13,42,14,43,33,34] often provides an information on the preconditioner spectrum bounds in terms of the spectrum of the original matrix: a small spectral radius of an iteration matrix implies that the preconditioned matrix is close, in a certain norm, to the identity matrix.

An important implication of (16) is that

$$\|\varphi_k(-tA)\| \leqslant \int_0^1 \|\exp(-t(1-\theta)A)\| \frac{\theta^{k-1}}{(k-1)!} \mathrm{d}\theta \leqslant C_A \varphi_k(-t\omega) \leqslant C_A \frac{1}{k!}, \quad t \geqslant 0.$$

The following result is given in [17].

Theorem 1. *Assume that (17) holds. Then for the residual $r_k(t)$ of the iteration (7), (8) it holds*

$$\begin{aligned}\|r_k(t)\| &\leqslant \left(C_M \|M-A\|\, t\right)^k \mathrm{e}^{-t\tilde{\omega}} \varphi_k(t\tilde{\omega})\mu_0 \\ &\leqslant \left(C_M \|M-A\|\, t\right)^k \frac{1}{k!}\,\mu_0, \quad k = 1, 2, \ldots,\ t \geqslant 0,\end{aligned}$$

where $\mu_0 = \max_{0\leqslant s\leqslant t} \|r_0(s)\|$.

In practice, the correction problem (8) is solved approximately with a certain number m of block SAI–Krylov iterations. Hence, the correction $\xi_k(t)$ satisfies a perturbed IVP

$$\xi_k'(t) \equiv -M\xi_k(t) + r_k(t) - \tilde{r}_m(t), \qquad \xi_k(0) = 0, \tag{20}$$

with inner iteration residual $\tilde{r}_m(t)$ defined in (12). Of course, the number of inner iterations m may vary with k but, to keep the notation simple, we prefer to write m rather than m_k. For this inexact WR iteration (7), (20) the following result holds.

Theorem 2. *Assume that (17) holds and the inner residual $\tilde{r}_m(t)$ satisfies, for some small $\delta > 0$,*

$$\|\tilde{r}_m(t)\| \leqslant \delta \|r_k(t)\|, \quad k = 0, 1, \ldots,\ t \geqslant 0. \tag{21}$$

Then for the residual $r_k(t)$ of the iteration (7), (20) it holds

$$\begin{aligned}\|r_k(t)\| &\leqslant \left(C_M \|M-A\|\, t(1+\delta)\right)^k \mathrm{e}^{-t\tilde{\omega}} \varphi_k(t\tilde{\omega})\mu_0 \\ &\leqslant \left(C_M \|M-A\|\, t(1+\delta)\right)^k \frac{1}{k!}\,\mu_0, \quad k = 1, 2, \ldots,\ t \geqslant 0,\end{aligned} \tag{22}$$

where $\mu_0 = \max_{0\leqslant s\leqslant t} \|r_0(s)\|$.

The proof of the theorem closely follows the lines of the proof of Theorem 1 (see [17]) and is given in the Appendix.

As Theorem 1 shows, the convergence of the WR iteration is superlinear. Theorem 2 assures that the superlinear convergence is not lost in the inexact WR iteration (7), (20). Finally, we note that it is easy to check in practice whether (21) holds because $\tilde{r}_m(t)$ is readily computable by (13).

5. Numerical experiments

In this section results of several numerical tests are discussed. We compare the new implementation of the WR iteration (7), (8) with the SAI–Krylov method (6), (7). In the SAI–Krylov method, the correction problem (6) is solved approximately by projecting it onto the SAI–Krylov subspace $\mathcal{K}_m((I+\gamma A)^{-1}, r_k(t))$. If we start with the initial guess $y_0(t) = v$ then $r_0(t) = -Av$ and it can be shown (see Lemma 2.2 in [17]) that $r_k(t)$ is parallel, for all t, to the vector $(I+\gamma A)v_{m+1}$, where v_{m+1} is the last $(m+1)$st Krylov basis vector. For more details on this method and its implementation we refer to [17] and to a related earlier work [15,16,48] where conventional (non-SAI) and extended Krylov subspace methods are discussed.

We implement the WR–Krylov as described in Section 2, with a small modification to incorporate the Anderson mixing [36]. In all tests we use 5 mixing terms and the results appear not to be sensitive to this value. All numerical experiments are carried out on a Linux cluster with ten Intel Xeon 2.40 GHz CPUs and 64 Gb memory.

5.1. A fourth order discretization of 3D Poisson problem

In this test the matrix A in (1) is a fourth order accurate finite volume discretization [49] of the 3D Laplacian operator with periodic boundary conditions, defined in the spatial domain $[0, 1]^3$. The initial vector v contains the mesh values of the function

$$\sin(2\pi x)\sin(2\pi y)\sin(2\pi z) + x(1-x)y(1-y)z(1-z).$$

The final time is chosen as $T = 1000$.

Since the discretization is fourth order accurate, the matrix A in this test is not an M-matrix, the SAI-systems with A are harder to solve than for the standard second order discretization. However, the second order discretization can be used to form the matrix M because the systems with $I+\gamma M$ are then easy to solve. This is the choice we make in this test. In the WR method we set the number of samples to 40 and we use the CG iterative method with the preconditioner $I+\gamma M$ to solve the SAI-systems. Both the SAI–Krylov and WR iterations are stopped with the help of the residual stopping criterion (cf. (4)).

Table 1
Results for the 3D Poisson test problem.

Method	Toler.	Error	CPU time (s)	Iterations (matvecs)
Mesh $20 \times 20 \times 20$				
WR–Krylov	1e−08	1.9e−06	4.8	5
WR–Krylov	1e−10	2.0e−06	7.5	7
SAI–Krylov	1e−08	1.1e−04	3.8	7 (45 mvecs)
SAI–Krylov	1e−10	2.2e−07	5.2	7 (64 mvecs)
ITR	$\tau = 20$	9.8e−06	12.2	–
ITR	$\tau = 50$	6.7e−03	8.0	–
Mesh $40 \times 40 \times 40$				
WR–Krylov	1e−05	7.6e−04	26	3
WR–Krylov	1e−08	4.4e−06	41	5
WR–Krylov	1e−10	3.6e−06	60	7
SAI–Krylov	1e−05	2.2e−03	26	4 (6 mvecs)
SAI–Krylov	1e−08	3.9e−06	30	9 (23 mvecs)
SAI–Krylov	1e−10	2.6e−07	35	13 (39 mvecs)
ITR	$\tau = 20$	8.2e−05	1088	–
Mesh $40 \times 40 \times 40$, anisotropic				
WR–Krylov	1e−05	1.0e−04	48	4
WR–Krylov	1e−08	9.8e−05	83	7
WR–Krylov	1e−10	9.9e−05	114	9
SAI–Krylov	1e−05	1.7e−05	37	10 (28 mvecs)
SAI–Krylov	1e−08	2.3e−08	45	15 (59 mvecs)
SAI–Krylov	1e−10	4.3e−09	52	15 (85 mvecs)
ITR	$\tau = 5$	1.2e−02	1187	–
ITR	$\tau = 2.5$	2.6e−03	1588	–

Table 2
Results for the 3D Poisson test problem, Gaussian initial data.

Method	Toler.	Error	CPU time (s)	Iterations (matvecs)
Mesh $40 \times 40 \times 40$				
WR–Krylov	1e−05	4.8e−04	49	4
WR–Krylov	1e−08	1.6e−05	88	6
WR–Krylov	1e−10	1.6e−05	129	8
SAI–Krylov	1e−05	4.7e−03	39	7 (16 mvecs)
SAI–Krylov	1e−08	6.9e−06	47	14 (44 mvecs)
SAI–Krylov	1e−10	2.4e−08	54	19 (72 mvecs)

To compare both the SAI–Krylov and WR methods to a more conventional time stepping techniques, we include the implicit trapezoidal rule (ITR, also known as the Crank–Nicolson scheme) in the comparisons. The linear systems in the ITR scheme are solved by the UMFPACK sparse direct solver [50,51], provided in MATLAB. Furthermore, we also include results for a modified anisotropic problem, where A corresponds to the differential operator $10^4\partial_{xx} + 10^2\partial_{yy} + \partial_{zz}$. The results of the test runs are presented in Table 1. We see that the SAI–Krylov method clearly outperforms the WR method and both SAI–Krylov and WR are much faster than the conventional ITR scheme. We also note the accuracy loss observed in the WR method for higher tolerances. This is due to the residual low rank representation procedure required by the method.

Finally, to explore the dependence on the smoothness of the initial data, we include in Table 2 the runs where the initial vector v corresponds to a Gaussian peak

$$\mathrm{e}^{-500((x-0.5)^2+(y-0.5)^2+(z-0.5)^2)}.$$

The performance of both methods deteriorates for this choice of v, and the effect seems to be more pronounced for the WR method.

5.2. Convection–diffusion problem

We solve the initial-value problem (1) where A is the standard five point finite-difference approximation of the convection–diffusion operator

$$L[u] = -(D_1 u_x)_x - (D_2 u_y)_y + Pe(v_1 u_x + v_2 u_y),$$

$$D_1(x,y) = \begin{cases} 10^3 & (x,y) \in [0.25, 0.75]^2, \\ 1 & \text{otherwise}, \end{cases} \qquad D_2(x,y) = \frac{1}{2} D_1(x,y),$$

$$v_1(x,y) = x + y, \qquad v_2(x,y) = x - y.$$

Table 3
Numerical results for the convection–diffusion test. The inner iteration count for the WR method is shown as $(m_1 \times r_1 + m_2 \times r_2 + \cdots)$, where m_k is the number of inner block Krylov iterations at iteration k and r_k is the block size.

	CPU time (s)	Error	Iterations (matvecs)
Choice 1 of v (equal components)			
SAI–Krylov	35	2.76e−08	100 (101)
WR–Krylov	15	2.39e−09	2 ($12 \times 1 + 3 \times 5$)
Choice 2 of v ("smooth" function)			
SAI–Krylov	35	3.91e−10	100 (114)
WR–Krylov	24	2.94e−09	2 ($29 \times 1 + 12 \times 8$)
Choice 3 of v (Gaussian peak)			
SAI–Krylov	37	5.95e−09	100 (113)
WR–Krylov	27	1.10e−07	3 ($21 \times 1 + 12 \times 7 + 5 \times 7$)

Here L acts on functions defined on the unit square $(x, y) \in [0, 1]^2$ which satisfy homogeneous Dirichlet boundary conditions. Before discretization the convection terms are rewritten as

$$v_1 u_x + v_2 u_y = \frac{1}{2}(v_1 u_x + v_2 u_y) + \frac{1}{2}((v_1 u)_x + (v_2 u)_y).$$

This guarantees that discretizations of the diffusion and convection terms yield symmetric and skew-symmetric matrices [45], respectively. The discretization is scaled with h_x^2, where h_x is the mesh size in the x direction.

We take M such that $I + \gamma M$ is the incomplete LU factorization of $I + \gamma A$, performed with pivoting and a threshold value 10^{-7} [14]. To show the dependence of the low rank approximation on the initial data, the tests are carried out for three different choices of the initial value vector v:

(1) v has all its components equal;
(2) v corresponds to the function $\cos(\pi x)\cos(\pi y)$;
(3) v corresponds to the function $\exp(-500(x - 0.5)^2 - 500(y - 0.5)^2)$.

In each case v is normalized to have $\|v\|_2 = 1$.

The problem is discretized by standard second order finite differences on a 402×402 mesh, so that the problem size is $n = 400^2$. In the SAI–Krylov method the SAI systems are solved by GMRES(100) with the ILU preconditioner $I + \gamma M$. The solvers are run with the residual tolerance 10^{-8} and $s = 33$ time samples are used in the WR method. The results are presented in Table 3. As we see, the WR method is faster but the provided gain deteriorates as the initial vector v becomes less "smooth" (and, hence, the low rank representation becomes less efficient).

5.3. 3D Maxwell electromagnetic imaging model

This test problem is taken from the field electromagnetic imaging in gas-and-oil industry [52,53]. The IVP (1) is a spatial discretization of Maxwell's equations posed in a three-dimensional (3D) spatial domain:

$$\begin{aligned} \mu \partial_t \boldsymbol{H} &= -\nabla \times \boldsymbol{E}, \\ \varepsilon \partial_t \boldsymbol{E} &= \nabla \times \boldsymbol{H} - \sigma \boldsymbol{E} + \boldsymbol{J}, \end{aligned} \tag{23}$$

where $\boldsymbol{H} = \boldsymbol{H}(x, y, z, t)$ and $\boldsymbol{E} = \boldsymbol{E}(x, y, z, t)$ are unknown vector functions of the magnetic and electric fields, respectively, $\mu = \mu_0$ is the magnetic permeability, $\varepsilon = \varepsilon_0$ is the electric permittivity (μ_0 and ε_0 are the magnetic permeability and electric permittivity of vacuum, respectively), $\sigma = \sigma(x, y, z)$ is electric conductivity and known $\boldsymbol{J} = \boldsymbol{J}(x, y, z, t)$ is the electric current. The equations are posed in a cubical physical domain $\Omega = [-20, 20]^3$ (the size is given in meters), with far field boundary conditions (homogeneous Dirichlet). The initial conditions are provided as a result of previous simulations and/or measurements.

In this test the conductivity σ is piecewise constant and defined as

$$\sigma = \begin{cases} 0.1 & \text{S/m}, \quad x \leqslant 10, \\ 0.001 & \text{S/m}, \quad x > 10. \end{cases} \tag{24}$$

The current source $\boldsymbol{J}$ (A) is zero everywhere in the domain except on a coil of a square shape, with four vertices whose coordinates (x, y, z) are $(-2, -2, 0)$, $(-2, 2, 0)$, $(2, 2, 0)$ and $(2, -2, 0)$. The current source can be switched on and off, and simulation in this test starts at the time moment when the source current is just switched off and reached its zero value. The time interval is $[0, \frac{4}{3} \cdot 10^{-5}]$ in seconds or $[0, 100]$ in dimensionless units (we use the usual dimensionless scaling of the Maxwell equations).

The standard spatial discretization by the Yee finite differences leads to an IVP of the form (1), with $y(t)$ containing the components of both fields. We use a mesh of size either $20 \times 20 \times 20$ or $40 \times 40 \times 40$, meaning that the problem size is

Table 4
Results of the 3D Maxwell test problem. The inner iteration count for the WR method is shown as $(m_1 \times r_1 + m_2 \times r_2 + \cdots)$, where m_k is the number of inner block Krylov iterations at iteration k and r_k is the block size.

	Toler	CPU time (s)	Error	Iterations (matvecs)
$n = 55\,566$				
WR–Krylov	1e−6	82	3.9e−06	$3\,(21 \times 1 + 12 \times 8 + 7 \times 9)$
SAI–Krylov	1e−6	38	9.6e−06	15 (34)
$n = 413\,526$				
WR–Krylov	1e−6	3614	5.1e−06	$3\,(21 \times 1 + 14 \times 8 + 9 \times 9)$
SAI–Krylov	1e−6	1056	3.7e−07	18 (46)

either $n = 55\,566$ or $n = 413\,526$. For more details on this problem we refer to [18] as well as to [52,53]. For other work where Krylov subspace and related techniques are successfully used to solve Maxwell's equations in different settings see e.g. (in chronological order) [54–59].

The matrix A in this test has a two-by-two block structure corresponding to the two fields, namely,

$$A = \begin{bmatrix} 0 & D_\mu^{-1}K \\ -D_\varepsilon^{-1}K^T & D_\varepsilon^{-1}D_\sigma \end{bmatrix},$$

where K is the discretized curl operator and D_* are diagonal matrices containing the mesh values of μ, ε and σ (the matrices D_* would be the mass matrices in case a finite element discretization is employed [60–62]). Hence, the matrix $I + \gamma A$ can be block factorized as

$$I + \gamma A \equiv \begin{bmatrix} I & \gamma D_\mu^{-1}K \\ -\gamma D_\varepsilon^{-1}K^T & I + \gamma D_\varepsilon^{-1}D_\sigma \end{bmatrix} = \begin{bmatrix} I & 0 \\ -\gamma D_\varepsilon^{-1}K^T & L \end{bmatrix} \begin{bmatrix} I & \gamma D_\mu^{-1}K \\ 0 & L^T D_\varepsilon \end{bmatrix},$$

where L is the triangular Cholesky factor of the Schur complement

$$LL^T = D_\varepsilon^{-1} + \gamma D_\varepsilon^{-1}D_\sigma D_\varepsilon^{-1} + \gamma^2 D_\varepsilon^{-1}K^T D_\mu^{-1}K D_\varepsilon^{-1}.$$

The matrix $I + \gamma M \approx I + \gamma A$ is then obtained as an incomplete Cholesky factorization $\hat{L}\hat{L}^T$ of the Schur complement, i.e.,

$$I + \gamma M := \begin{bmatrix} I & 0 \\ -\gamma D_\varepsilon^{-1}K^T & \hat{L} \end{bmatrix} \begin{bmatrix} I & \gamma D_\mu^{-1}K \\ 0 & \hat{L}^T D_\varepsilon \end{bmatrix}$$

where $\hat{L}$ is obtained from L by setting its entries below a threshold value (in this test 10^{-3}) to zero.

The SAI systems in the SAI–Krylov method are solved by GMRES(100) with the ILU preconditioner $I + \gamma M$ and the sample number s in the WR–Krylov method is set to 20 in this test. The results of the test runs are presented in Table 4. As can be seen from the results, the SAI–Krylov method appears to be faster for this problem. In this test, a significant CPU time is required to form the matrix M which is at disadvantage of the WR method.

6. Conclusions

The both compared methods employ the knowledge of an approximation $M \approx A$ for solving the system $y'(t) = -Ay(t)$, each in its own way. The SAI–Krylov method does so on the linear algebra level, when solving the SAI systems with the matrix $I + \gamma A$ iteratively. The WR–Krylov employs the knowledge of M on the differential equation level (cf. (7), (8)). Several conclusions can be drawn from the presented results. The SAI–Krylov method appears to be faster in two out of the three presented tests. Moreover, the method seems to be slightly simpler conceptually as well as in implementation. Finally, for higher accuracy requirements the proposed implementation of the WR–Krylov method seems to suffer from an accuracy loss due to the low rank representation of the residual samples. Our general conclusion therefore is that a preference should be given to the SAI–Krylov method. However, for some problems, e.g. where the solution is expected to be smooth and of a low rank, the presented WR–Krylov method can be promising.

Appendix. Proof of Theorem 2

Here we give the proof of Theorem 2. Using the variation of constants formula and the fact that $\xi_k(t)$ solves the perturbed IVP (20), we have

$$r_{k+1}(t) = (M - A)\xi_k(t) = (M - A)\int_0^t \exp\big(-(t-s)M\big)(r_k(s) - \tilde{r}_m(s))\mathrm{d}s. \tag{25}$$

It follows from (21) that for $k = 0, 1, \ldots$

$$\|r_k(s) - \tilde{r}_m(s)\| \leqslant \|r_k(s)\|\left(1 + \frac{\|\tilde{r}_m(s)\|}{\|r_k(s)\|}\right) \leqslant \|r_k(s)\|(1 + \delta), \quad s \geqslant 0.$$

We will prove the first inequality in (22) by induction on k. Setting $k = 1$ and employing (17) and the last inequality, we obtain

$$\begin{aligned}
\|r_1(t)\| &\leqslant \|M - A\| \int_0^t \left\|\exp\bigl(-(t-s)M\bigr)\right\| \|r_0(s) - \tilde{r}_m(s)\| \mathrm{d}s \\
&\leqslant \|M - A\| \int_0^t C_M \mathrm{e}^{-(t-s)\tilde{\omega}} \|r_0(s) - \tilde{r}_m(s)\| \mathrm{d}s \\
&\leqslant C_M \|M - A\| \int_0^t \mathrm{e}^{-(t-s)\tilde{\omega}} (1+\delta) \|r_0(s)\| \mathrm{d}s \\
&\leqslant C_M \|M - A\| (1+\delta) \mu_0 \mathrm{e}^{-t\tilde{\omega}} \int_0^t \mathrm{e}^{s\tilde{\omega}} \mathrm{d}s = C_M \|M - A\| (1+\delta) \mu_0 \mathrm{e}^{-t\tilde{\omega}} t \varphi_1(t\tilde{\omega}).
\end{aligned}$$

Thus, the bound (17) is true for $k = 1$. Let us now assume that it is true for a certain k and carry out the induction step. Using (25) we have

$$\begin{aligned}
\|r_{k+1}(t)\| &\leqslant \|M - A\| \int_0^t \left\|\exp\bigl(-(t-s)M\bigr)\right\| \|r_k(s) - \tilde{r}_m(t)\| \mathrm{d}s \\
&\leqslant \|M - A\| \int_0^t C_M \mathrm{e}^{-(t-s)\tilde{\omega}} \|r_k(s) - \tilde{r}_m(s)\| \mathrm{d}s \\
&\leqslant C_M \|M - A\| (1+\delta) \int_0^t \mathrm{e}^{-(t-s)\tilde{\omega}} \|r_k(s)\| \mathrm{d}s.
\end{aligned}$$

The induction assumption on $\|r_k(s)\|$ leads to

$$\begin{aligned}
\|r_{k+1}(t)\| &\leqslant C_M \|M - A\| (1+\delta) \int_0^t \mathrm{e}^{-(t-s)\tilde{\omega}} \Bigl(C_M \|M - A\| \, s(1+\delta)\Bigr)^k \mathrm{e}^{-s\tilde{\omega}} \varphi_k(s\tilde{\omega}) \mu_0 \mathrm{d}s \\
&= \Bigl(C_M \|M - A\| \, (1+\delta)\Bigr)^{k+1} \mu_0 \int_0^t \mathrm{e}^{-(t-s)\tilde{\omega}} \mathrm{e}^{-s\tilde{\omega}} s^k \varphi_k(s\tilde{\omega}) \mathrm{d}s \\
&= \Bigl(C_M \|M - A\| \, (1+\delta)\Bigr)^{k+1} \mu_0 \mathrm{e}^{-t\tilde{\omega}} \int_0^t s^k \varphi_k(s\tilde{\omega}) \mathrm{d}s \\
&= \Bigl(C_M \|M - A\| \, (1+\delta)\Bigr)^{k+1} \mu_0 \mathrm{e}^{-t\tilde{\omega}} t^{k+1} \varphi_{k+1}(t\tilde{\omega}) \mathrm{d}s,
\end{aligned}$$

which proves the bound. Here the relation

$$\int_0^t s^k \varphi_k(s\tilde{\omega}) \mathrm{d}s = t^{k+1} \varphi_{k+1}(t\tilde{\omega}) \mathrm{d}s, \quad k = 0, 1, \ldots$$

is used. □

References

[1] C.B. Moler, C.F. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, SIAM Rev. 45 (1) (2003) 3–49.
[2] N.J. Higham, Functions of Matrices: Theory and Computation, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.
[3] T.J. Park, J.C. Light, Unitary quantum time evolution by iterative Lanczos reduction, J. Chem. Phys. 85 (1986) 5870–5876.
[4] V.L. Druskin, L.A. Knizhnerman, Two polynomial methods of calculating functions of symmetric matrices, USSR Comput. Math. Math. Phys. 29 (6) (1989) 112–121.
[5] L.A. Knizhnerman, Calculation of functions of unsymmetric matrices using Arnoldi's method, USSR Comput. Math. Math. Phys. 31 (1) (1991) 1–9.
[6] Y. Saad, Analysis of some Krylov subspace approximations to the matrix exponential operator, SIAM J. Numer. Anal. 29 (1) (1992) 209–228.
[7] E. Gallopoulos, Y. Saad, Efficient solution of parabolic equations by Krylov approximation methods, SIAM J. Sci. Stat. Comput. 13 (5) (1992) 1236–1264.
[8] V.L. Druskin, L.A. Knizhnerman, Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic, Numer. Linear Algebra Appl. 2 (1995) 205–217.
[9] M. Hochbruck, C. Lubich, On Krylov subspace approximations to the matrix exponential operator, SIAM J. Numer. Anal. 34 (5) (1997) 1911–1925.
[10] M. Hochbruck, C. Lubich, H. Selhofer, Exponential integrators for large systems of differential equations, SIAM J. Sci. Comput. 19 (5) (1998) 1552–1574.
[11] A. Frommer, V. Simoncini, Matrix functions, in: W.H.A. Schilders, H.A. van der Vorst, J. Rommes (Eds.), Model Order Reduction: Theory, Research Aspects and Applications, Springer, 2008, pp. 275–304.
[12] S. Güttel, Rational Krylov methods for operator functions, Ph.D. Thesis, Technischen Universität Bergakademie Freiberg, March 2010. www.guettel.com.
[13] H.A. van der Vorst, Iterative Krylov Methods for Large Linear Systems, Cambridge University Press, 2003.
[14] Y. Saad, Iterative methods for sparse linear systems. Book out of print, 2000. www-users.cs.umn.edu/~saad/books.html.
[15] E. Celledoni, I. Moret, A Krylov projection method for systems of ODEs, Appl. Numer. Math. 24 (2–3) (1997) 365–378.
[16] V.L. Druskin, A. Greenbaum, L.A. Knizhnerman, Using nonorthogonal Lanczos vectors in the computation of matrix functions, SIAM J. Sci. Comput. 19 (1) (1998) 38–54.
[17] M.A. Botchev, V. Grimm, M. Hochbruck, Residual, restarting, and richardson iteration for the matrix exponential, SIAM J. Sci. Comput. 35 (3) (2013) A1376–A1397.

[18] M.A. Botchev, A block Krylov subspace time-exact solution method for linear ordinary differential equation systems, Numer. Linear Algebra Appl. 20 (4) (2013) 557–574.
[19] R.B. Sidje, Expokit. A software package for computing matrix exponentials, ACM Trans. Math. Software 24 (1) (1998) 130–156. www.maths.uq.edu.au/expokit/.
[20] W. Hundsdorfer, J.G. Verwer, Numerical Solution of Time-Dependent Advection–Diffusion–Reaction Equations, Springer Verlag, 2003.
[21] R.P. Fedorenko, Stiff systems of ordinary differential equations, in: Numerical Methods and Applications, CRC, Boca Raton, FL, 1994, pp. 117–154.
[22] V.L. Druskin, L.A. Knizhnerman, Extended Krylov subspaces: approximation of the matrix square root and related functions, SIAM J. Matrix Anal. Appl. 19 (3) (1998) 755–771.
[23] I. Moret, P. Novati, RD rational approximations of the matrix exponential, BIT 44 (2004) 595–615.
[24] J. van den Eshof, M. Hochbruck, Preconditioning Lanczos approximations to the matrix exponential, SIAM J. Sci. Comput. 27 (4) (2006) 1438–1457.
[25] H. Tal-Ezer, On restart and error estimation for Krylov approximation of $w = f(A)v$, SIAM J. Sci. Comput. 29 (6) (2007) 2426–2441.
[26] M. Afanasjew, M. Eiermann, O.G. Ernst, S. Güttel, Implementation of a restarted Krylov subspace method for the evaluation of matrix functions, Linear Algebra Appl. 429 (2008) 2293–2314.
[27] M. Eiermann, O.G. Ernst, S. Güttel, Deflated restarting for matrix functions, SIAM J. Matrix Anal. Appl. 32 (2) (2011) 621–641.
[28] J. Niehoff, Projektionsverfahren zur approximation von matrixfunktionen mit anwendungen auf die implementierung exponentieller integratoren, Ph.D. Thesis, Mathematisch-Naturwissenschaftlichen Fakultät der Heinrich-Heine-Universität Düsseldorf, December 2006.
[29] A.R. Newton, A.L. Sangiovanni-Vincentelli, Relaxation-based electrical simulation, IEEE Trans. Electron Devices 30 (9) (1983) 1184–1207.
[30] J. White, F. Odeh, A.L. Sangiovanni-Vincentelli, A. Ruehli, Waveform relaxation: theory and practice, Technical Report UCB/ERL M85/65, EECS Department, University of California, Berkeley, 1985.
[31] U. Miekkala, O. Nevanlinna, Convergence of dynamic iteration methods for initial value problems, SIAM J. Sci. Stat. Comput. 8 (4) (1987) 459–482.
[32] P. Castillo, Y. Saad, Preconditioning the matrix exponential operator with applications, J. Sci. Comput. 13 (3) (1998) 275–302.
[33] W. Hackbusch, Iterative Solution of Large Sparse Systems of Equations, Springer-Verlag, 1994.
[34] O. Axelsson, Iterative Solution Methods, Cambridge University Press, Cambridge, 1994.
[35] C. Lubich, A. Ostermann, Multi-grid dynamic iteration for parabolic equations, BIT 27 (1987) 216–234. http://dx.doi.org/10.1007/BF01934186.
[36] H.F. Walker, P. Ni, Anderson acceleration for fixed-point iterations, SIAM J. Numer. Anal. 49 (4) (2011) 1715–1735.
[37] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., The Johns Hopkins University Press, Baltimore, London, 1996.
[38] S.A. Goreinov, Cross approximation of a multi-index array, Dokl. Math. 77 (3) (2008) 404–406.
[39] S.A. Goreinov, I.V. Oseledets, D.V. Savostyanov, E.E. Tyrtyshnikov, N.L. Zamarashkin, How to find a good submatrix, in: Matrix Methods: Theory, Algorithms and Applications, World Sci. Publ., Hackensack, NJ, 2010, pp. 247–256.
[40] M. Hochbruck, A. Ostermann, Exponential integrators, Acta Numer. 19 (2010) 209–286.
[41] K. Dekker, J.G. Verwer, Stability of Runge–Kutta Methods for Stiff Non-Linear Differential Equations, North-Holland Elsevier Science Publishers, 1984.
[42] R.S. Varga, Matrix Iterative Analysis, Prentice-Hall, 1962.
[43] A.A. Samarskii, E.S. Nikolaev, Numerical Methods for Grid Equations. Vol. I&II, Birkhäuser Verlag, Basel, 1989.
[44] J.M. Ortega, Introduction to Parallel and Vector Solution of Linear Systems, Plenum Press, 1988.
[45] L.A. Krukier, Implicit difference schemes and an iterative method for solving them for a certain class of systems of quasi-linear equations, Sov. Math. 23 (7) (1979) 43–55. Translation from Izv. Vyssh. Uchebn. Zaved., Mat. 1979, No. 7 (206), 41–52 (1979).
[46] M.A. Bochev, L.A. Krukier, Iterative solution of strongly nonsymmetric systems of linear algebraic equations, Russian Comput. Math. Math. Phys. 37 (11) (1997) 1241–1251.
[47] C.-Y. Zhang, M. Benzi, p-regular splitting iterative methods for non-Hermitian positive definite linear systems, ETNA: Electron. Trans. Numer. Anal. 36 (2009–2010) 39–53.
[48] L. Knizhnerman, V. Simoncini, A new investigation of the extended Krylov subspace method for matrix function evaluations, Numer. Linear Algebra Appl. 17 (4) (2010) 615–638.
[49] R.W.C.P. Verstappen, A.E.P. Veldman, Symmetry-preserving discretization of turbulent flow, J. Comput. Phys. 187 (1) (2003) 343–368. http://dx.doi.org/10.1016/S0021-9991(03)00126-8.
[50] T.A. Davis, A column pre-ordering strategy for the unsymmetric-pattern multifrontal method, ACM Trans. Math. Software 30 (2) (2004) 167–195.
[51] T.A. Davis, Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method, ACM Trans. Math. Software 30 (2) (2004) 196–199.
[52] A.V. Gelber, E.P. Shurina, M.I. Epov, An application of finite element method for solving the problem of modeling non-stationary electromagnetic fields of defectoscope, in: International Conference on Computational Mathematics. Part I, II, ICM&MG Pub., Novosibirsk, 2002, pp. 427–431.
[53] E.P. Shurina, A.V. Gelber, M.A. Gelber, M.I. Epov, Mathematical modelling of non-stationary electromagnetic fields of defectoscope of casings, Comput. Technol. 7 (6) (2002) 114–129 (in Russian). www.ict.nsc.ru/jct/annotation/346?l=eng.
[54] R.F. Remis, P.M. van den Berg, A modified Lanczos algorithm for the computation of transient electromagnetic wavefields, IEEE Trans. Microw. Theory Tech. 45 (12) (1997) 2139–2149.
[55] H. De Raedt, K. Michielsen, J.S. Kole, M.T. Figge, One-step finite-difference time-domain algorithm to solve the Maxwell equations, Phys. Rev. E 67 (2003) 056706.
[56] X. Ma, X. Zhao, Y. Zhao, A 3-d precise integration time-domain method without the restraints of the Courant–Friedrich–Levy stability condition for the numerical solution of Maxwell's equations, IEEE Trans. Microw. Theory Tech. 54 (7) (2006) 3026–3037.
[57] K. Busch, J. Niegemann, M. Pototschnig, L. Tkeshelashvili, A Krylov-subspace based solver for the linear and nonlinear Maxwell equations, Phys. Status Solidi b 244 (10) (2007) 3479–3496.
[58] M. Zaslavsky, V.L. Druskin, Solution of time-convolutionary Maxwell's equations using parameter-dependent Krylov subspace reduction, J. Comput. Phys. 229 (12) (2010) 4831–4839.
[59] V. Druskin, R. Remis, A Krylov stability-corrected coordinate-stretching method to simulate wave propagation in unbounded domains, SIAM J. Sci. Comput. 35 (2) (2013) B376–B400.
[60] G. Rodrigue, D. White, A vector finite element time-domain method for solving Maxwell's equations on unstructured hexahedral grids, SIAM J. Sci. Comput. 23 (3) (2001) 683–706.
[61] M.A. Botchev, J.G. Verwer, Numerical integration of damped Maxwell equations, SIAM J. Sci. Comput. 31 (2) (2009) 1322–1346. http://dx.doi.org/10.1137/08072108X.
[62] J.G. Verwer, M.A. Botchev, Unconditionally stable integration of Maxwell's equations, Linear Algebra Appl. 431 (3–4) (2009) 300–317.