Time stepping free numerical solution of linear differential equations: Krylov subspace versus waveform relaxation[*]

M. A. Botchev[†] I. V. Oseledets[‡] E. E. Tyrtyshnikov[‡]

October 10, 2013

Abstract

The aim of this paper is two-fold. First, we propose an efficient implementation of the continuous time waveform relaxation method based on block Krylov subspaces. Second, we compare this new implementation against Krylov subspace methods combined with the shift and invert technique.

Keywords: Krylov subspace methods, waveform relaxation, matrix exponential, low rank approximation, residual, Richardson iteration, Anderson acceleration

AMS subject classifications: 65F60, 65L05, 65F10, 65F30, 65N22

1 Introduction

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

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

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

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

Krylov subspace methods have been successfully used for the evaluation of matrix exponential and for numerical solution of various time dependent problems since the late 80s. We mention in chronological order some early papers [34, 11, 27, 36, 16, 12, 23, 24]. For a more recent work see a survey [15] and corresponding chapters in [22, 21]. These methods are based on a projection of the original IVP (1) onto the Krylov subspace

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

[*] Supported by the Russian Federal Program "Scientific and scientific-pedagogical personnel of innovative Russia", grant 8500.

[†] Department of Applied Mathematics and MESA+ Institute for Nanotechnology, University of Twente, P.O. Box 217, 7500 AE Enschede, the Netherlands, mbotchev@na-net.ornl.gov.

[‡] Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkin St. 8, Moscow 119333, Russia, [ivan.oseledets,eugene.tyrtyshnikov]@gmail.com.

Here 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 [42, 37]. The process computes the columns v_1, v_2, ..., 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 [42, 37], namely,

$$AV_m = V_{m+1}H_{m+1,m} = V_m H_{m,m} + v_{m+1}h_{m+1,m}e_m^T, \qquad \mathbb{R}^m \ni e_m = (0, \dots, 0, 1)^T,$$

where $H_{m+1,m} = V_{m+1}^* AV_m \in \mathbb{R}^{(m+1)\times m}$ and $H_{m,m} = V_m^* AV_m \in \mathbb{R}^{m\times m}$ 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

$$AV_m \approx V_m H_{m,m},$$

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

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

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

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

which can be checked in practice by some error estimates, for example, with the help of the exponential residual defined as [6, 10, 3]

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

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 [2] to a more general IVP

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

where $g : [0, T] \to \mathbb{R}^n$ is a given function. Furthermore, this property allows to regard the Krylov subspace methods applied in this setting as *time stepping free* methods. 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, for example, the efficiency does depend on the length of the time interval (typically, the smaller T, the smaller Krylov dimension m suffices [39]).

If A is close to a symmetric positive definite matrix with a stiff spectrum, the convergence of Krylov subspace methods can be slow. In this case the convergence and can often be dramatically improved by switching to the shift-and-invert (SAI) Krylov subspace $\mathcal{K}((I + \gamma A)^{-1}, w)$ [31, 41], where $\gamma > 0$ is a parameter related to T. These methods, where at each Krylov step a system with the matrix $I + \gamma A$ has to be solved, belong to the class of rational Krylov subspace methods [21], see also related work [13].

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 $y_k(t) = y(t)$ for which the residual $r_k(t)$, defined by (4), is known. Then a better approximation can be obtained by the following iteration:

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

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

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 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 [6, 3]. See also related work on Krylov subspace methods and restarting [40, 1, 14, 33, 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

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

If 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 (and this is the approach we follow in this paper), then iteration (7),(8) can be seen as the continuous time waveform relaxation (WR) method [32, 46, 30].

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 SAI-Krylov subspace and low rank approximation techniques, while allowing a broad choice of M. Second, we compare performance of this WR-Krylov method with that of the SAI-Krylov method (i.e., of 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 $I + \gamma M$).

Thus, it is instructive to compare these two approaches as this contributes to answering the old important question [23, 5, 41]: how to use a 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 system 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 time stepping free methods and, as such, are often efficient. Moreover, they both are good candidates for 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 to compare the WR methods versus the SAI-Krylov subspace methods are presented and results of the tests are discussed. We draw some conclusions in Section 6 and describe a possible way to accelerate the WR iterations by Anderson mixing in Appendix A.

2 A low rank block Krylov subspace implementation of WR methods

In this section we describe how the continuous time WR methods [32, 46, 30] can be implemented by block SAI-Krylov subspace and low rank approximation techniques.

We consider the IVP (1) and throughout the paper assume 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 [31, 41] 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 [41].

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, whereas 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, we choose s sample time points $t_1 = 0 < t_1 < t_2 < \cdots < t_s = T$ covering the time interval $[0,T]$. Second, the parameter $\gamma > 0$ is set (see [41]) and a suitable preconditioner $I+\gamma M$ for the matrix $I+\gamma A$ is chosen.

Remark 1 *We emphasize that the preconditioner can be chosen in any of many possible ways, as done for the linear system solution [42, 37]. 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 [32, 46]. 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 respectively lower and upper triangular matrices. 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 $r_0(t) = -Av$. It will be clear from the further discussion that other choices for $y_0(t)$ are possible, all of them allowing a representation

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

with $p_0(t) : \mathbb{R} \to \mathbb{R}^r$ being some simple function such as polynomial. As we will see later, for presentation purposes it is convenient to assume that relation (9) holds (obviously, $r_0(t) = -Av$ fits into the form (9) with $r = 1$). 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 onto the Krylov subspace in a block way, namely, the subspace is

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

and is spanned by the columns of the matrix

$$V_{[m]} = \begin{bmatrix} V_1 & \dots & V_m \end{bmatrix} \in \mathbb{R}^{n\times mr}, \qquad V_i \in \mathbb{R}^{n\times r}, \quad i=1,\dots,m, \tag{10}$$

where $V_1 = U_0$. The matrix $H_{[k+1,k]}$ is block upper Hessenberg, with $m\times m$ blocks H_{ij}, $i=1,\dots,k+1$, $j=1,\dots,k$. The matrices $V_{[m]}$ satisfies the block Arnoldi (Lanczos) decomposition [42, 37]

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

where $H_{[m+1,m]} \in \mathbb{R}^{(m+1)r\times mr}$ is a block upper Hessenberg matrix, $H_{m+1,m}\in\mathbb{R}^{r\times r}$ is the only nonzero block in the last $m+1$ block row of $H_{[m+1,m]}$ and $E_m\in\mathbb{R}^{n\times r}$ consists of the last r columns of the $mr\times mr$ identity matrix.

Let $\xi_{k,m}(t) \approx \xi_k(t)$ be the mth iterand of this inner block iterative process, with the residual

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

Then it can be shown [2] that

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

where $u_{[m]}(t):\mathbb{R}\to\mathbb{R}^{mr}$ is the solution of the IVP (8) projected onto the block SAI-Krylov subspace. This small projected IVP is usually solved by a standard ODE solver. For more details on this block method and its implementation see [2].

If m steps SAI-Krylov steps are done to compute the correction $\xi_0(t)$, then $\xi_0(t)$ is available as $V_{[m]}u_{[m]}(t)$, Next, the s time samples $\xi_0(t_1)$, …, $\xi_0(t_s)$ 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. [29, 3])

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

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 r, for the new residual $r_{k+1}(t)$, namely

$$r_{k+1}(t) = U_{k+1}p_{k+1}(t), \qquad U_{k+1}\in\mathbb{R}^{n\times r}, \quad r\ll n, \tag{13}$$

where the notation is as in (9). In general, the low rank representation (13) can not be guaranteed and we settle for an approximation

$$\begin{aligned} &r_{k+1}(t) \approx U_{k+1}p_{k+1}(t), \qquad U_{k+1}\in\mathbb{R}^{n\times r}, \quad r\ll n, \\ &\|r_{k+1}(t) - U_{k+1}p_{k+1}(t)\| \leqslant \epsilon, \qquad t = t_1,\dots,t_s, \end{aligned} \tag{14}$$

```
For given $A \in \mathbb{R}^{n\times n}$, $v \in \mathbb{R}^n$, $T > 0$ solves IVP $y' = -Ay$, $y(0) = v$
choose number of samples $s$ (usually 20 to 40)
choose points $t_i$, $i = 1, \dots, s$, such that $t_1 = 0 < t_1 < \cdots < t_s = T$
set $\gamma$ (usually $\gamma = T/10$, see [41])
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 iteration $k = 0, 1, 2, \dots$
        solve IVP $\xi_k'(t) = -M\xi_k + U_k p_k(t)$, $\xi_k(0) = 0$ by block SAI-Krylov method,
            using $LU = I + \gamma M$ as a direct SAI solver
        compute $s$ residual time samples $r_{k+1}(t_i) = (M - A)\xi_k(t_i)$, $i = 1, \dots, s$
        stop if $\|r_{k+1}(t_i)\|$, $i = 1, \dots, s$, small enough
        based on residual samples, set up a low rank approximation $r_{k+1}(t) :\approx U_{k+1} p_{k+1}(t)$
endfor
```

Figure 1: The low rank block SAI-Krylov subspace implementation of the WR iteration

with ϵ being a certain tolerance. Different low rank approximation techniques can be used to obtain (14) and we briefly discuss two of them in Section 3.

The WR iteration presented in this section is outlined in Figure 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 []. 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. However, for completeness purposes, we describe how Anderson mixing can be carried out in Appendix A.

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 (14).

It is based on the widely known truncated singular value decomposition (SVD), see e.g. [18]. Assume the time samples $r_{k+1}(t_i) = (M - A)\xi_k(t_i)$ 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 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 time moments $t_1, \dots, t_s$. Interpolation of these values yields then a vector polynomial function $p_{k+1}(t)$. For more details on this procedure we refer to [2]. We emphasize that other methods than the SVD can be used to obtain (14), for example, the cross approximation algorithm and related techniques [19, 20].

4 Convergence of the block SAI-Krylov WR iteration

Convergence of the WR iteration is studied e.g. in [46, 30, 3]. In this section we first formulate a result from [3] on convergence of *exact* WR iteration (7),(8), i.e. 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 [25, 3], we assume that there exist constants $C_A > 0$ and $\omega \geqslant 0$ such that

$$\|\exp(-tA)\| \leqslant C_A \mathrm{e}^{-t\omega}, \qquad t \geqslant 0. \tag{15}$$

This bound is satisfied in the spectral norm for $C_A = 1$ provided that A has its field of values contained in the complex halfplane $\mathbb{C}_\omega := \{z \in \mathbb{C} \colon \operatorname{Re} z \geqslant \omega\}$. We then have $\omega = -\mu(-A)$, where $\mu(\cdot)$ is the logarithmic matrix norm [9, 22]. The inequality (15) 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 $\mathbb{C}_\omega$.

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

$$\|\exp(-tM)\| \leqslant C_M \mathrm{e}^{-t\tilde{\omega}}, \qquad t \geqslant 0. \tag{16}$$

Next, we introduce the functions φ_k (see e.g. [25]):

$$\varphi_k(z) = \int_0^1 \mathrm{e}^{(1-\theta)z} \frac{\theta^{k-1}}{(k-1)!} \mathrm{d}\theta, \quad k \geqslant 1. \tag{17}$$

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

$$\varphi_{k+1}(z) = \frac{\varphi_k(z) - \varphi_k(0)}{z}, \quad \varphi_0(z) = \mathrm{e}^z. \tag{18}$$

An important implication of (15) is that

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

The following result is given in [3].

Theorem 1 *Assume that* (16) *holds. Then for the residual* $r_k(t)$ *of the iteration* (7),(8) *holds*

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

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 the block SAI-Krylov iterations. Hence, the correction $\xi_k(t)$ satisfies a perturbed IVP

$$\xi_k'(t) \equiv -M\xi_k(t) + r_k(t) - \tilde{r}_m(t), \qquad \xi_k(0) = 0, \tag{19}$$

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),(19) the following result holds.

Theorem 2 *Assume that* (16) *holds and the inner residual* $\tilde{r}_m(t)$ *satisfies, for some small* $\delta > 0$,

$$\|\tilde{r}_m(t)\| \leqslant \delta \|r_k(t)\|, \qquad k = 0, 1, \dots, \qquad t \geqslant 0. \tag{20}$$

Then for the residual $r_k(t)$ *of the iteration* (7),(19) *holds*

$$\begin{aligned} \|r_k(t)\| &\leqslant \Big(C_M \|M - A\|\, t(1+\delta)\Big)^k \mathrm{e}^{-t\tilde{\omega}} \varphi_k(t\tilde{\omega}) \mu_0 \\ &\leqslant \Big(C_M \|M - A\|\, t(1+\delta)\Big)^k \frac{1}{k!}\, \mu_0, \qquad k = 1, 2, \dots, \qquad t \geqslant 0, \end{aligned} \tag{21}$$

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

Proof Using a variation of constants formula and the fact that $\xi_k(t)$ solves the perturbed IVP (19), we have

$$r_{k+1}(t) = (M - A)\xi_k(t) = (M - A) \int_0^t \exp\big(-(t-s)M\big)(r_k(s) - \tilde{r}_m(s)) \mathrm{d}s. \tag{22}$$

It follows from (20) that for $k = 0, 1, \dots$

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

We will prove the first inequality in (21) by induction on k. Setting $k = 1$ and employing (16) and the last inequality, we obtain

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

Thus, the bound (16) is true for $k = 1$. Let us now assume that it is true for certain k and carry out the induction step. Using (22) we have

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

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

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

which proves the bound. Here the relation

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

is used. □

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),(19).

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). The correction problem (6) in the SAI-Krylov method 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 [3]) that $r_k(t)$ is a time dependent vector function with values in a one-dimensional subspace spanned by $I+\gamma A$ times the last $(m+1)$ Krylov basis vector. For more details on this method and its implementation we refer to [3] and to a related earlier work [6, 10, 26] where conventional (non-SAI) and extended Krylov subspace methods are discussed.

We implement the WR method as described in Section 2, with a small modification to incorporate the Anderson mixing [45] as described in Appendix A. 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 2.40GHz CPUs and 64Gb 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 [43] 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

Table 1: Results for the 3D Poisson test problem

method	toler.	error	CPU time, s	iterations (matvecs)
		mesh $20 \times 20 \times 20$		
WR method	1e-08	1.9e-06	4.8	5
WR method	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 method	1e-05	7.6e-04	26	3
WR method	1e-08	4.4e-06	41	5
WR method	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 method	1e-05	1.0e-04	48	4
WR method	1e-08	9.8e-05	83	7
WR method	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 method	1e-05	4.8e-04	49	4
WR method	1e-08	1.6e-05	88	6
WR method	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)

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)).

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 [8, 7], 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 dependence on the smoothness of the initial data, in Table 2 we include 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 initial-value problem (1) where A is the standard five point finite-difference approximation of the convection–diffusion operator

$$L[u] = -(D_1 u_x)_x - (D_2 u_y)_y + Pe(v_1 u_x + v_2 u_y),$$
$$D_1(x,y) = \begin{cases} 10^3 & (x,y) \in [0.25, 0.75]^2, \\ 1 & \text{otherwise}, \end{cases} \qquad D_2(x,y) = \frac{1}{2} D_1(x,y),$$
$$v_1(x,y) = x + y, \qquad v_2(x,y) = x - y.$$

Here L acts on functions defined on the unite 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 discretization of the diffusion and convection terms yields respectively symmetric and skew-symmetric matrices [28]. 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} [37]. 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 $\|v\|_2 = 1$.

Table 3: Numerical results for the convection–diffusion test. The inner iteration count for the WR method is shown as $(m_1 \times b_1 + m_2 \times b_2 + \dots)$, where m_k is the number of inner block Krylov iterations at iteration k and b_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 method	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 method	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 method	27	1.10e-07	3 $(21 \times 1 + 12 \times 7 + 5 \times 7)$

The problem is discretized by the 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 [17, 38]. 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, respectively, the magnetic and electric fields, $\mu = \mu_0$ is the magnetic permeability, $\varepsilon = \varepsilon_0$ is the electric permittivity (μ_0 and ε_0 are respectively the magnetic permeability and electric permittivity of vacuum), $\sigma = \sigma(x, y, z)$ 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 the 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

Table 4: Results of the 3D Maxwell test problem. The inner iteration count for the WR method is shown as $(m_1 \times b_1 + m_2 \times b_2 + \dots)$, where m_k is the number of inner block Krylov iterations at iteration k and b_k is the block size.

	toler	CPU time, s	error	iterations (matvecs)
			$n = 55\,566$	
WR method	`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 method	`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)

at the time moment wheh 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 either $n = 55\,566$ or $n = 413\,526$. For more detail on this problem we refer to [2] as well as to [17, 38].

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 (D_* would be mass matrices for finite element discretization [35, 4, 44]). 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 number of the samples s the WR 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 method employs the knowledge of M on the differential equation level (cf. (7),(8)). Several conclusions can be drawn from the presented results. SAI-Krylov method appeares to be faster in two out of the three presented tests. Moreover, the method seems to slightly simpler conceptually as well as in implementation. Finally, for higher accuracy requirements the proposed implementation of the WR 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.

References

[1] M. Afanasjew, M. Eiermann, O. G. Ernst, and S. Güttel. Implementation of a restarted Krylov subspace method for the evaluation of matrix functions. *Linear Algebra Appl.*, 429:2293–2314, 2008.

[2] M. A. Botchev. A block Krylov subspace time-exact solution method for linear ordinary differential equation systems. *Numer. Linear Algebra Appl.*, 20(4):557–574, 2013.

[3] M. A. Botchev, V. Grimm, and M. Hochbruck. Residual, restarting, and richardson iteration for the matrix exponential. *SIAM Journal on Scientific Computing*, 35(3):A1376–A1397, 2013.

[4] M. A. Botchev and J. G. Verwer. Numerical integration of damped Maxwell equations. *SIAM J. Sci. Comput.*, 31(2):1322–1346, 2009. http://dx.doi.org/10.1137/08072108X.

[5] P. Castillo and Y. Saad. Preconditioning the matrix exponential operator with applications. *J. Sci. Comput.*, 13(3):275–302, 1998.

[6] E. Celledoni and I. Moret. A Krylov projection method for systems of ODEs. *Appl. Numer. Math.*, 24(2-3):365–378, 1997.

[7] T. A. Davis. Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Software*, 30(2):196–199, 2004.

[8] T. A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Trans. Math. Software*, 30(2):167–195, 2004.

[9] K. Dekker and J. G. Verwer. *Stability of Runge–Kutta methods for stiff non-linear differential equations.* North-Holland Elsevier Science Publishers, 1984.

[10] V. L. Druskin, A. Greenbaum, and L. A. Knizhnerman. Using nonorthogonal Lanczos vectors in the computation of matrix functions. *SIAM J. Sci. Comput.*, 19(1):38–54, 1998.

[11] V. L. Druskin and L. A. Knizhnerman. Two polynomial methods of calculating functions of symmetric matrices. *U.S.S.R. Comput. Maths. Math. Phys.*, 29(6):112–121, 1989.

[12] V. L. Druskin and L. A. Knizhnerman. Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic. *Numer. Lin. Alg. Appl.*, 2:205–217, 1995.

[13] V. L. Druskin and L. A. Knizhnerman. Extended Krylov subspaces: approximation of the matrix square root and related functions. *SIAM J. Matrix Anal. Appl.*, 19(3):755–771 (electronic), 1998.

[14] M. Eiermann, O. G. Ernst, and S. Güttel. Deflated restarting for matrix functions. *SIAM J. Matrix Anal. Appl.*, 32(2):621–641, 2011. `http://dx.doi.org/10.1137/090774665`.

[15] A. Frommer and V. Simoncini. Matrix functions. In W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, editors, *Model Order Reduction: Theory, Research Aspects and Applications*, pages 275–304. Springer, 2008.

[16] E. Gallopoulos and Y. Saad. Efficient solution of parabolic equations by Krylov approximation methods. *SIAM J. Sci. Statist. Comput.*, 13(5):1236–1264, 1992.

[17] A. V. Gelber, E. P. Shurina, and 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*, pages 427–431. ICM&MG Pub., Novosibirsk, 2002.

[18] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore and London, third edition, 1996.

[19] S. A. Goreinov. Cross approximation of a multi-index array. *Dokl. Math.*, 77(3):404–406, 2008.

[20] S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. How to find a good submatrix. In *Matrix methods: theory, algorithms and applications*, pages 247–256. World Sci. Publ., Hackensack, NJ, 2010.

[21] S. Güttel. *Rational Krylov Methods for Operator Functions*. PhD thesis, Technischen Universität Bergakademie Freiberg, March 2010. `www.guettel.com`.

[22] N. J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[23] M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 34(5):1911–1925, Oct. 1997.

[24] M. Hochbruck, C. Lubich, and H. Selhofer. Exponential integrators for large systems of differential equations. *SIAM J. Sci. Comput.*, 19(5):1552–1574, 1998.

[25] M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numer.*, 19:209–286, 2010.

[26] L. Knizhnerman and V. Simoncini. A new investigation of the extended Krylov subspace method for matrix function evaluations. *Numer. Linear Algebra Appl.*, 2009. To appear.

[27] L. A. Knizhnerman. Calculation of functions of unsymmetric matrices using Arnoldi's method. *U.S.S.R. Comput. Maths. Math. Phys.*, 31(1):1–9, 1991.

[28] 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):43–55, 1979. Translation from Izv. Vyssh. Uchebn. Zaved., Mat. 1979, No. 7(206), 41–52 (1979).

[29] C. Lubich and A. Ostermann. Multi-grid dynamic iteration for parabolic equations. *BIT Numerical Mathematics*, 27:216–234, 1987. 10.1007/BF01934186.

[30] U. Miekkala and O. Nevanlinna. Convergence of dynamic iteration methods for initial value problems. *SIAM Journal on Scientific and Statistical Computing*, 8(4):459–482, 1987.

[31] I. Moret and P. Novati. RD rational approximations of the matrix exponential. *BIT*, 44:595–615, 2004.

[32] A. Newton, A.R. ; Sangiovanni-Vincentelli. Relaxation-based electrical simulation. *IEEE Transactions on Electron Devices*, 30(9):1184–1207, 1983.

[33] J. Niehoff. *Projektionsverfahren zur Approximation von Matrixfunktionen mit Anwendungen auf die Implementierung exponentieller Integratoren.* PhD thesis, Mathematisch--Naturwissenschaftlichen Fakultät der Heinrich-Heine-Universität Düsseldorf, December 2006.

[34] T. J. Park and J. C. Light. Unitary quantum time evolution by iterative Lanczos reduction. *J. Chem. Phys.*, 85:5870–5876, 1986.

[35] G. Rodrigue and D. White. A vector finite element time-domain method for solving Maxwell's equations on unstructured hexahedral grids. *SIAM J. Sci. Comput.*, 23(3):683–706, 2001.

[36] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 29(1):209–228, 1992.

[37] Y. Saad. *Iterative Methods for Sparse Linear Systems.* Book out of print, 2000. `www-users.cs.umn.edu/~saad/books.html`.

[38] E. P. Shurina, A. V. Gelber, M. A. Gelber, and M. I. Epov. Mathematical modelling of non-stationary electromagnetic fields of defectoscope of casings. *Computational technologies*, 7(6):114–129, 2002. In Russian. `www.ict.nsc.ru/jct/annotation/346?l=eng`.

[39] R. B. Sidje. Expokit. A software package for computing matrix exponentials. *ACM Trans. Math. Softw.*, 24(1):130–156, 1998. `www.maths.uq.edu.au/expokit/`.

[40] H. Tal-Ezer. On restart and error estimation for Krylov approximation of $w = f(A)v$. *SIAM J. Sci. Comput.*, 29(6):2426–2441 (electronic), 2007.

[41] J. van den Eshof and M. Hochbruck. Preconditioning Lanczos approximations to the matrix exponential. *SIAM J. Sci. Comput.*, 27(4):1438–1457, 2006.

[42] H. A. van der Vorst. *Iterative Krylov methods for large linear systems.* Cambridge University Press, 2003.

[43] R. W. C. P. Verstappen and A. E. P. Veldman. Symmetry-preserving discretization of turbulent flow. *J. Comput. Phys.*, 187(1):343–368, 2003. `http://dx.doi.org/10.1016/S0021-9991(03)00126-8`.

[44] J. G. Verwer and M. A. Botchev. Unconditionally stable integration of Maxwell's equations. *Linear Algebra and its Applications*, 431(3–4):300–317, 2009.

[45] H. F. Walker and P. Ni. Anderson acceleration for fixed-point iterations. *SIAM J. Numer. Anal.*, 49(4):1715–1735, 2011.

[46] J. White, F. Odeh, A. L. Sangiovanni-Vincentelli, and A. Ruehli. Waveform relaxation: Theory and practice. Technical Report UCB/ERL M85/65, EECS Department, University of California, Berkeley, 1985.

A Richardson WR iteration and Anderson acceleration

Assume we solve initial-value problem (1) by the WR (Richardson) iteration (7),(8). An Anderson acceleration for this iteration can be formulated as follows.

To facilitate presentation, we first rewrite the WR iteration (7),(8) in a different form. Let $y_0(t)$ be an initial guess function for which the residual (4) is known (see Section 2 for more detail). For instance, we can choose $y_0(t) = v$ (y_0 is a constant function), so that $r_0(t) = -Av$. Since the error function $\varepsilon_0(t) \equiv y(t) - y_0(t)$ satisfies the IVP

$$\varepsilon_0'(t) = -A\varepsilon_0(t) + r_0(t), \quad \varepsilon_0(t) = 0,$$

a better solution

$$y_1(t) = y_0(t) + \xi_0(t) \tag{25}$$

can be obtained by solving the correction problem (8) for $k = 0$. The correction function $\xi_0(t)$ can be written as

$$\xi_0(t) = \int_0^t \exp((s-t)M) r_0(s) \mathrm{d}s \tag{26}$$

We now introduce, for a given matrix $B \in \mathbb{R}^{n\times n}$, a mapping

$$\mathcal{R}_B[u(t)] = Bu(t) + u'(t),$$

defined for all sufficiently smooth functions $u(t)$. The inverse of $\mathcal{R}_B$ can be defined, for all $u(t)$ satisfying the initial condition $u(0) = 0$, as

$$\mathcal{R}_B^{-1}[u(t)] = \int_0^t \exp((s-t)B) u(s) \mathrm{d}s.$$

Note that

$$\mathcal{R}_A[\varepsilon_0(t)] = r_0(t) \quad \text{and} \quad -\mathcal{R}_A[y_0(t)] = r_0(t).$$

Then, the Richardson update step (7) can be written as a fixed point iteration

$$\begin{aligned} y_1(t) &= \mathcal{G}[y_0(t)], \\ \mathcal{G}[y_0(t)] &= y_0(t) + \mathcal{R}_M^{-1}[r_0(t)] = y_0(t) + \mathcal{R}_M^{-1}[-\mathcal{R}_A[y_0(t)]] \\ &= y_0(t) + \int_0^t \exp((s-t)M)(-y_0'(s) - Ay_0(s)) \mathrm{d}s. \end{aligned} \tag{27}$$

```
    Choose $\hat{m} \approx 5$ and $y_0(t)$ such that $y_0(0) = v$.
1.  Carry out Richardson iteration $y_1(t) = \mathcal{G}[y_0(t)]$
    for $k = 1, 2, \dots$
2.      set $\hat{m}_k = \min\{\hat{m}, k\}$
3.      set matrix $R_k = \left[\ \bar{r}_{k-m_k} \quad \dots \quad \bar{r}_k\ \right]$
        with columns $\bar{r}_j = \int_0^t \mathcal{R}_M^{-1}[r_j(s)]\mathrm{d}s$, $j = k - \hat{m}_k, \dots, k$
4.      determine $\vec{\alpha}_k = (\alpha_0, \dots, \alpha_{\hat{m}_k})$ that solves
            $\min_{\vec{\alpha}} \|R_k\vec{\alpha}\|_2$ such that $\sum_{i=0}^{\hat{m}_k} \alpha_i = 1$
5.      $y_{k+1}(t) = \sum_{i=0}^{\hat{m}_k} \alpha_i \mathcal{G}[y_{k-\hat{m}_k+i}]$
    endfor
```

Figure 2: An algorithm of the Anderson acceleration for the WR iteration. Since $\mathcal{G}$ is linear, a single $\mathcal{G}$ evaluation at step 5 suffices.

Note that the mapping $\mathcal{G}$ is linear. On the other hand, we have

$$\mathcal{G}[y_0(t)] = y_0(t) + \mathcal{R}_M^{-1}[r_0(t)] = y_0(t) + \mathcal{R}_M^{-1}[\mathcal{R}_A[\varepsilon_0(t)]].$$

From this, it is immediate to see that $y_1(t)$ is the exact solution $y(t)$ of the original problem as soon as $M = A$.

In Figure 2 we outline an Anderson acceleration [45] scheme for the WR iteration. To evaluate the integrals at step 3, a simple numerical quadrature is employed. The linearity of $\mathcal{G}$ allows to have just a single evaluation of $\mathcal{G}$ at step 5, this is done by solving the correction IVP on the mixed residuals. The number of mixing terms $\hat{m}$ is usually taken around 5.