

Computational Science Laboratory  
Technical Report CSL-TR-1-2013  
January 30, 2015

Paul Tranquilli and Adrian Sandu

*“Rosenbrock-Krylov Methods for Large  
Systems of Differential Equations”*

**Cite as:** Paul Tranquilli and Adrian Sandu. Rosenbrock-Krylov methods for large systems of differential equations. SIAM Journal of Scientific Computing. Volume 36, Issue 3, Pages A1313 – A1338, 2014.

Computational Science Laboratory  
Computer Science Department  
Virginia Polytechnic Institute and State University  
Blacksburg, VA 24060  
Phone: (540)-231-2193  
Fax: (540)-231-6075  
Email: sandu@cs.vt.edu  
Web: <http://cs1.cs.vt.edu>



# ROSENBRACK-KRYLOV METHODS FOR LARGE SYSTEMS OF DIFFERENTIAL EQUATIONS

PAUL TRANQUILLI <sup>†§</sup> AND ADRIAN SANDU <sup>‡§</sup>

**Abstract.** This paper develops a new class of Rosenbrock-type integrators based on a Krylov space solution of the linear systems. The new family, called Rosenbrock-Krylov (Rosenbrock-K), is well suited for solving large-scale systems of ODEs or semi-discrete PDEs. The time discretization and the Krylov space approximation are treated as a single computational process, and the Krylov space properties are an integral part of the new Rosenbrock-K order condition theory developed herein. Consequently, Rosenbrock-K methods require a small number of basis vectors determined solely by the temporal order of accuracy. The subspace size is independent of the ODE under consideration, and there is no need to monitor the errors in linear system solutions at each stage. Numerical results show favorable properties of Rosenbrock-K methods when compared to current Rosenbrock and Rosenbrock-W schemes.

**1. Introduction.** This paper is concerned with the numerical solution of large initial value problems of the form

$$(1.1) \quad \frac{dy}{dt} = f(t, y), \quad t_0 \leq t \leq t_F, \quad y(t_0) = y_0; \quad y(t), f(t, y) \in \mathbb{R}^N.$$

Ordinary differential equations (ODEs) model the evolution of chemical kinetic systems, electronic circuits, or the motion of planets. Other physical systems, such as dynamics of the atmosphere and oceans, or the behavior of materials, require the solution of a system of partial differential equations (PDEs). A standard approach for solving evolutionary PDEs is to discretize in space, reducing the problem to a large system of ODEs which are then integrated forward in time.

The equations of interest in many simulations are driven by multiple physical processes, e.g., associated with the simulation of fluid-structure interaction, sub-grid-scale physics, or chemical reaction terms. These processes have different dynamical characteristics, with some being slow (e.g., transport) and some fast (e.g., stiff chemistry). In addition, multiple spatial and temporal scales are associated with non-uniform meshes, with boundary layers, with fast waves, and with the structure of the system (e.g., the existence of jet fans). The existence of fast and slow dynamics poses considerable challenges to the solution of the semi-discrete PDEs (1.1) by explicit time stepping methods. Specifically, due to the Courant-Friedrichs-Lewy (CFL) stability condition, the largest allowable step sizes are bounded above by the shortest (fastest) time scale in the system.

To avoid stability restrictions on the step size, implicit time integration methods are becoming widely used in the simulation of large-scale evolutionary PDEs (1.1). Implicit time stepping requires the solution of large nonlinear systems of equations, coupling all variables in the model, at each time step. The associated computation and communication costs constitute a major scalability bottleneck at best, and can quickly become infeasible for large systems.

In this paper we examine, and extend, the Rosenbrock class of integration methods. These are linearly-implicit methods which enjoy the benefit of requiring a fixed

---

<sup>†</sup>E-mail: ptranq@vt.edu

<sup>‡</sup>E-mail: sandu@cs.vt.edu

<sup>§</sup>Computational Science Laboratory, Department of Computer Science, Virginia Tech. 2202 Kraft Drive, Blacksburg, Virginia 24060.

number of linear solves, as opposed to non-linear solves in the case of implicit Runge-Kutta methods. A general  $s$ -stage Rosenbrock method reads [13, Section IV.7]

$$(1.2a) \quad y_{n+1} = y_n + \sum_{i=1}^s b_i k_i,$$

$$(1.2b) \quad (\mathbf{I}_{N \times N} - h \gamma_{i,i} \mathbf{A}_n) \cdot k_i = h f \left( t_n + \alpha_i h, y_n + \sum_{j=1}^{i-1} \alpha_{i,j} k_j \right) \\ + h \mathbf{A}_n \sum_{j=1}^{i-1} \gamma_{i,j} k_j + h^2 \gamma_i f_t(t_n, y_n), \quad i = 1, \dots, s.$$

Here  $y_n \approx y(t_n)$  and  $y_{n+1} \approx y(t_{n+1})$  are the numerical solutions computed by the method, and  $h = t_{n+1} - t_n$  is the current time step. The term  $f_t = \partial f / \partial t$  is the partial time derivative of  $f$  evaluated at the beginning of the current time step. The method coefficients  $\alpha_{i,j}$ ,  $\gamma_{i,j}$ , and  $b_i$  are chosen such as to ensure the desired accuracy and stability properties. For convenience of notation one also defines

$$\alpha_i = \sum_{j=1}^{i-1} \alpha_{i,j}, \quad \gamma_i = \sum_{j=1}^i \gamma_{i,j}, \quad \beta_{i,j} = \alpha_{i,j} + \gamma_{i,j}.$$

In classical Rosenbrock methods the matrix  $\mathbf{A}_n = \mathbf{J}_n \in \mathbb{R}^{N \times N}$ , where

$$\mathbf{J}_n = f_y(t_n, y_n)$$

is the Jacobian of  $f$  evaluated at the beginning of the current time step. The vectors  $k_i$  are the intermediate stage values; each of them is computed by solving an  $N \times N$  linear system. Typically all coefficients are chosen equal to each other,  $\gamma_{i,i} = \gamma$  for  $i = 1, \dots, s$ , such that all stages share the same LU decomposition. The order conditions of classical Rosenbrock methods rely on the assumptions that  $\mathbf{A}_n$  is the exact ODE Jacobian, and that each of the linear systems (1.2b) is solved exactly.

For many systems the exact Jacobian  $\mathbf{J}_n$  can be both costly and difficult to obtain, e.g., due to the size of the application and the use of complex spatial discretization schemes. The class of Rosenbrock-W methods [13, Section IV.7] has been derived for such situations. They have the same form (1.2), but the coefficients are selected such that the overall discretization order is preserved for any matrix  $\mathbf{A}_n$ , i.e., for arbitrary approximations of the Jacobian [34]. The role of the matrix is to ensure numerical stability, and its choice dictates the type and amount of implicitness used in (1.2).

Solving for the stage values  $k_i$  (1.2b) is the most expensive part of the integrator. For large-scale systems direct methods such as  $LU$  decomposition are not feasible, and iterative Krylov based methods, such as GMRES [36, 44], have been considered in the literature [46, 47, 45]. Krylov based methods use only Jacobian-vector products and do not require storage, or even knowledge of, the full Jacobian.

The use of Krylov(-Newton) methods is a natural approach to speed up the solutions of the linear/nonlinear systems of equations in implicit time integration [27, 28]. Such integration methods have been successfully employed in real-life applications [11, 40, 8]. Krylov space solvers have been used in the implementation of implicit Runge-Kutta [26, 7, 3], implicit linear multistep [22, 17, 8], and deferred correction [24] methods. Software for solving stiff ODEs and DAEs with this approach include

LSODKR [15], LSODPK [16], VODPK [17], and DASPK [5]. In addition, Krylov space methods have been used for the exact integration of linear ODEs with source terms [1], and to improve stability [2]. Krylov space techniques have been used to accelerate convergence of deferred correction methods [23, 25].

Of particular interest in this work is the use of Krylov methods in the context of Rosenbrock time integrators (1.2). Classical Rosenbrock integrators are poor matrix free methods due to the explicit presence of the exact Jacobian matrix, and the approximate nature of Krylov based methods. Rosenbrock-W methods are better suited for coupling with Krylov based solvers. A family of matrix-free Rosenbrock methods, named Krylov-ROW, has been proposed in the 1990's [45, 38, 33, 35, 41]. The control of linear solution accuracy in each stage is discussed in [38], and preconditioning in [39]. Order results for Krylov-ROW methods are studied in [45, 46]. A multiple Arnoldi process is proposed, where the Krylov space is enriched at each stage, such that the information from previous stages is reused, and all the right hand side vectors belong to the subspace. The order of the underlying Rosenbrock method is preserved under modest requirements on the Krylov space size, and independent of the dimension of the ODE system. The implementation of methods of order four is done in the code ROWMAP [35], where the error estimation and step size selection strategies are inherited from the underlying Rosenbrock method. The application of Krylov-ROW methods to index-1 DAEs [47] reveals that the Krylov space dimension needs to exceed the number of algebraic variables. Krylov-ROW methods are therefore attractive in the case where the number of algebraic constraints is small compared to the number of differential equations; or, by extension, where the dimension of the stiff subspace is small. Novati [32] presents a class of W-methods where the Jacobian is approximated using quasi-Newton-like rank one updates, based on solution and function values in previous time steps. Periodic restarts are needed for stability, as the Jacobian approximation deteriorates with time. A related family of methods are exponential integrators [18], which use matrix exponentials of the Jacobian as part of the solution process, and evaluate matrix exponential times vector products via Krylov space methods [21, 6, 42, 19, 20].

In this paper we develop a new family of Rosenbrock-Krylov time stepping methods characterized by the lowest possible degree of implicitness that ensures stability. The new algorithms are implicit in only the stiff subspace, which is captured by a Krylov space. Moreover, they perform only scalable operations such as Jacobian-vector products, and solve only small linear systems at each step. A naive implementation of a matrix free Rosenbrock-W method requires construction of a Krylov subspace for each stage equation. The Rosenbrock-K methods proposed herein extend the framework of Rosenbrock-W methods, accounting for the Krylov subspace approximation of the linear system when constructing the order conditions of the integrator. In this way we substantially reduce the number of required order conditions for a given order, thereby reducing the number of necessary stages of the method. The Rosenbrock-K methods require the construction of only a single Krylov subspace for the solution of all stage values. The dimension of this subspace need only be as large as the desired order of the method to ensure accuracy.

The new class of Rosenbrock-K methods differs from existing Rosenbrock schemes in several important aspects. Rosenbrock-K methods use approximate Jacobians similar to Rosenbrock-W family. The Jacobian approximations have to be based on a Krylov subspace approach, similar to the Krylov-ROW family. The primary benefit of Rosenbrock-K methods over Krylov-ROW stems from the integration of the

Krylov space properties into the Rosenbrock-K order condition theory; this elegant approach ensures the desired orders of accuracy with much smaller subspaces than those required by the Krylov-ROW approach. More importantly, a single Krylov space is used across all stages, and most operations are performed in this reduced space. The implementation of Rosenbrock-K is thus much simpler, and considerably more scalable, than the implementation of Krylov-ROW, where the subspace needs to be extended at each subsequent stage.

The paper is laid out as follows: in Section 2 we present the framework of the proposed class of methods as well as the Krylov subspace approximation of the Jacobian used, in Section 3 we extend the theory of order trees for Rosenbrock-w methods to our new Rosenbrock-Krylov methods as well as give details of how to construct these trees and the method order conditions from them, in Section 4 we construct two new Rosenbrock-Krylov integrators and outline a method for the solution of the order conditions to derive specific method coefficients, and in Section 5 we present some numerical results.

**2. Formulation of Rosenbrock-Krylov methods.** Rosenbrock-Krylov methods have the same form as Rosenbrock-W methods (1.2), but use a particular approximation  $\mathbf{A}_n$  of  $\mathbf{J}_n$ . We start the presentation with the case of autonomous systems.

Specifically, let  $f_n = f(y_n)$  and consider the  $M$ -dimensional Krylov space

$$(2.1) \quad \begin{aligned} \mathcal{K}_M(\mathbf{J}_n, f_n) &= \text{span} \{ f_n, \mathbf{J}_n f_n, \mathbf{J}_n^2 f_n, \dots, \mathbf{J}_n^{M-1} f_n \} \\ &= \text{span} \{ v_1, v_2, \dots, v_M \}. \end{aligned}$$

An orthogonal basis  $\{v_i\}_{i=1,\dots,M}$  for  $\mathcal{K}_M$  is constructed using a modified Arnoldi process [44]. The Arnoldi iteration returns two valuable pieces of information: a matrix  $\mathbf{V}_{n;M}$  whose columns are the orthonormal basis vectors of  $\mathcal{K}_M$ , and an upper Hessenberg matrix  $\mathbf{H}_{n;M}$ , such that

$$(2.2) \quad \mathbf{V}_{n;M} = [v_1, \dots, v_M] \in \mathbb{R}^{N \times M}; \quad \mathbf{H}_{n;M} = \mathbf{V}_{n;M}^T \mathbf{J}_n \mathbf{V}_{n;M} \in \mathbb{R}^{M \times M}.$$

*The Rosenbrock-Krylov matrix  $\mathbf{A}_n$  is the restriction of the full ODE Jacobian to the Krylov space:*

$$(2.3) \quad \mathbf{A}_n = \mathbf{V}_{n;M} \mathbf{H}_{n;M} \mathbf{V}_{n;M}^T = \mathbf{V}_{n;M} \mathbf{V}_{n;M}^T \mathbf{J}_n \mathbf{V}_{n;M} \mathbf{V}_{n;M}^T.$$

To obtain the stage vector  $k_i$  we decompose it in the component residing in  $\mathcal{K}_M$ , and the component orthogonal to  $\mathcal{K}_M$

$$(2.4) \quad k_i = \underbrace{\mathbf{V}_{n;M} \lambda_i}_{\in \mathcal{K}_M} + \underbrace{\mu_i}_{\in \mathcal{K}_M^\perp},$$

where the new vectors  $\lambda_i$  and  $\mu_i$  are defined by

$$\lambda_i = \mathbf{V}_{n;M}^T k_i \in \mathbb{R}^M, \quad \mu_i = (\mathbf{I}_{N \times N} - \mathbf{V}_{n;M} \mathbf{V}_{n;M}^T) k_i \in \mathbb{R}^N.$$

We consider also the projections of the function values in (1.2b) onto the Krylov space

$$F_i = f \left( y_n + \sum_{j=1}^{i-1} \alpha_{i,j} k_j \right), \quad \phi_i = \mathbf{V}_{n;M}^T F_i \in \mathbb{R}^M.$$

To construct a Rosenbrock-K method the Jacobian approximation (2.3) and the decomposition (2.4) are used in the stage formulation (1.2b) to obtain

$$(\mathbf{I}_{N \times N} - h\gamma \mathbf{V}_{n;M} \mathbf{H}_{n;M} \mathbf{V}_{n;M}^T) \cdot (\mathbf{V}_{n;M} \lambda_i + \mu_i) = h F_i + h \mathbf{V}_{n;M} \mathbf{H}_{n;M} \mathbf{V}_{n;M}^T \sum_{j=1}^{i-1} \gamma_{i,j} (\mathbf{V}_{n;M} \lambda_j + \mu_j).$$

Using the facts that  $\mathbf{V}_{n;M}^T \mathbf{V}_{n;M} = \mathbf{I}_{M \times M}$  and  $\mathbf{V}_{n;M}^T \mu_i = 0$  the equation can be written as

$$(2.5) \quad \mu_i + \mathbf{V}_{n;M} (\mathbf{I}_{M \times M} - h\gamma \mathbf{H}_{n;M}) \lambda_i = h F_i + h \mathbf{V}_{n;M} \mathbf{H}_{n;M} \sum_{j=1}^{i-1} \gamma_{i,j} \lambda_j.$$

Multiplying both sides of (2.5) by  $\mathbf{V}_{n;M}^T$  leads to the following reduced stage equation

$$(2.6) \quad (\mathbf{I}_{M \times M} - h\gamma \mathbf{H}_{n;M}) \lambda_i = h \phi_i + h \mathbf{H}_{n;M} \sum_{j=1}^{i-1} \gamma_{i,j} \lambda_j.$$

Multiplying both sides of (2.5) by  $\mathbf{I}_{N \times N} - \mathbf{V}_{n;M} \mathbf{V}_{n;M}^T$  gives

$$(2.7) \quad \mu_i = h (F_i - \mathbf{V}_{n;M} \phi_i).$$

The system (2.6) is of size  $M \times M$ , where  $M \ll N$ , and can be inverted through the use of a direct method. The full stage values can now be recovered from (2.4), (2.6), and (2.7) as

$$(2.8) \quad k_i = \mathbf{V}_{n;M} \lambda_i + h (F_i - \mathbf{V}_{n;M} \phi_i).$$

We next consider the formulation of Rosenbrock-K methods for non-autonomous problems (1.1). With the extended state and function

$$(2.9) \quad \hat{y}(t) = \begin{bmatrix} y(t) \\ t \end{bmatrix} \in \mathbb{R}^{N+1}, \quad \hat{f}(\hat{y}) = \begin{bmatrix} f(t, y) \\ 1 \end{bmatrix} \in \mathbb{R}^{N+1},$$

the general ODE (1.1) can be formulated as autonomous system

$$(2.10) \quad \frac{d\hat{y}}{dt} = \hat{f}(\hat{y}), \quad \hat{y}(t_0) = \begin{bmatrix} y \\ t_0 \end{bmatrix}, \quad t_0 \leq t \leq t_F, \quad \hat{y}(t), \hat{f}(\hat{y}).$$

Non-autonomous Rosenbrock-K integrators are constructed using this technique. The Jacobian of the extended right hand side function is

$$(2.11) \quad \hat{\mathbf{J}} = \begin{bmatrix} f_y & f_t \\ 0 & 0 \end{bmatrix}.$$

An extended Krylov space  $\mathcal{K}_M(\hat{\mathbf{J}}_n, \hat{f}_n)$  is constructed using matrix-vector products of the form

$$(2.12) \quad \hat{\mathbf{J}}_n [\zeta^T \ \xi]^T = \begin{bmatrix} \mathbf{J}_n \zeta + f_t(t_n, y_n) \xi \\ 0 \end{bmatrix}, \quad \zeta \in \mathbb{R}^N, \quad \xi \in \mathbb{R}.$$

The extended  $(N+1)$ -dimensional Arnoldi iterations produce the matrices  $\hat{\mathbf{V}}_{n;M}$  and  $\mathbf{H}_{n;M}$  such that

$$(2.13) \quad \hat{\mathbf{V}}_{n;M} = \begin{bmatrix} \mathbf{V}_{n;M} \\ w^T \end{bmatrix} \in \mathbb{R}^{(N+1) \times M}, \quad \mathbf{V}_{n;M} \in \mathbb{R}^{N \times M}, \quad w \in \mathbb{R}^M,$$

and

$$\mathbf{H}_{n;M} = \widehat{\mathbf{V}}_{n;M}^T \widehat{\mathbf{J}}_n \widehat{\mathbf{V}}_{n;M} = \mathbf{V}_{n;M}^T \mathbf{J}_n \mathbf{V}_{n;M} + \mathbf{V}_{n;M}^T f_t(t_n, y_n) w^T \in \mathbb{R}^{M \times M}.$$

This modified Arnoldi iteration proceeds as follows.

---

**Algorithm 1** Modified Arnoldi iteration

---

```

 $\beta = \left\| \begin{bmatrix} f_n^T & 1 \end{bmatrix}^T \right\|$ ;    $w_1 = 1/\beta$ ;    $v_1 = f_n/\beta$ .
for  $i = 1, \dots, M$  do
   $\zeta = \mathbf{J}_n v_i + f_t(t_n, y_n) w_i$ ;    $\xi = 0$ ;    $\tau = \|\zeta\|$ 
  for  $j = 1, \dots, i$  do
     $H_{j,i} = \langle \zeta, v_j \rangle + \xi w_j$ ;    $\zeta = \zeta - H_{j,i} v_j$ ;    $\xi = \xi - H_{j,i} w_j$ 
  end for
  if  $\left\| \begin{bmatrix} \zeta^T & \xi \end{bmatrix}^T \right\| / \tau \leq \kappa$  then
    for  $j = 1, \dots, i$  do
       $\rho = \langle \zeta, v_j \rangle + \xi w_j$ ;    $\zeta = \zeta - \rho v_j$ ;    $\xi = \xi - \rho w_j$ ;    $H_{j,i} = H_{j,i} + \rho$ 
    end for
  end if
   $H_{i+1,i} = \left\| \begin{bmatrix} \zeta^T & \xi \end{bmatrix}^T \right\|$ ;    $v_{i+1} = \zeta / H_{i+1,i}$ ;    $w_{i+1} = \xi / H_{i+1,i}$ ;
end for

```

---

The non-autonomous Rosenbrock-K integrator is obtained by applying the autonomous step (described above) to the extended system (2.10), and decoupling the state and time variables. The procedure is summarized in Algorithm 2. The autonomous version is obtained by letting  $w = 0$ .

---

**Algorithm 2** One step of a non-autonomous Rosenbrock-K integrator

---

- 1: Compute  $\mathbf{H}_{n;M}$ ,  $\mathbf{V}_{n;M}$ , and  $w$  (2.13) using the  $(N+1)$ -dimensional Arnoldi process.
- 2: **for**  $i = 1, \dots, s$  **do** ▷ For each stage, in succession

$$F_i = f \left( t_n + \alpha_i h, y_n + \sum_{j=1}^{i-1} \alpha_{i,j} k_j \right)$$

$$\phi_i = \mathbf{V}_{n;M}^T F_i + w$$

$$\lambda_i = (\mathbf{I}_{M \times M} - h\gamma \mathbf{H}_{n;M})^{-1} \left( h\phi_i + h\mathbf{H}_{n;M} \sum_{j=1}^{i-1} \gamma_{i,j} \lambda_j \right)$$

$$k_i = \mathbf{V}_{n;M} \lambda_i + h(F_i - \mathbf{V}_{n;M} \phi_i)$$

3: **end for**

$$4: y_{n+1} = y_n + \sum_{i=1}^s b_i k_i$$


---

**3. Order Conditions.** The accuracy theory is based on matching the Taylor series of the numerical solution and of the exact solution, up to some specified order. Butcher-trees [12, Section IV.7] are a well accepted method of representing individual terms in the Taylor series expansions. The derivation of order conditions for

Rosenbrock-K methods is an extension of the framework used to derive order conditions for Rosenbrock-W methods. The existing theory for Rosenbrock-W methods is based on the use of  $TW$ -trees, a subclass of  $P$ -trees [13, Section IV.7].  $P$ -trees themselves are an extension of the set  $T$  of Butcher-trees that allow for two different colors of the nodes. We have the following definition [13, Section IV.7]:

$$TW = \left\{ \begin{array}{l} P\text{-trees: end vertices are meagre, and} \\ \text{fat vertices are singly branched} \end{array} \right\}$$

In the context of Rosenbrock-K and Rosenbrock-W methods, a meagre node represents an actual derivative of  $f$  coming from the first term on the right of equation (1.2b), while a meagre node is an appearance of the approximate Jacobian  $\mathbf{A}_n$  coming from the second term on the right of equation (1.2b). Each tree represents a single elementary differential in the Taylor series of either the exact or numerical solutions of the ODE.

Figure 3.1 shows all  $TW$ -trees and Rosenbrock-W conditions for up to order three [13, Section IV.7]. The correspondence between the  $TW$ -trees and elementary differentials, and Rosenbrock-W order conditions, is summarized next:

- For the elementary differentials in Figure 3.1 superscripts represent component indices, and subscripts represent indices of variables with respect to which partial derivatives are taken. For example,  $f^J$  is the  $J$ -th component of  $f$ , and  $f_K^J f^K = \sum_K \partial f^J / \partial y^K \cdot f^K$ .
- A meagre node represents a derivative of  $f$ .
- The order of the  $f$  derivative equals the number of children the meagre node has.
- A fat node represents the appearance of the approximate Jacobian matrix,  $\mathbf{A}_n$ , in the elementary differential.

The correspondence between the  $TW$ -trees and Rosenbrock-W order conditions, is as follows:

- For the order conditions in Figure 3.1 the summations apply to all repeated indices in the expression.
- For Rosenbrock-W methods:
  - an edge originating from meagre node  $j$  to another node  $k$  gives  $\alpha_{j,k}$ ,
  - an edge originating from fat node  $j$  to another node  $k$  gives  $\gamma_{j,k}$ .
- For classical Rosenbrock methods
  - an edge connecting meagre node  $j$ , having multiple children, to a meagre node  $k$  gives  $\alpha_{j,k}$ ,
  - an edge connecting meagre node  $j$ , having a single child, to a meagre node  $k$  gives  $\beta_{j,k}$ .

The exact solution is represented by trees containing only meagre nodes, since the approximate Jacobian matrix never appears in its series expansion. For this reason Rosenbrock-W methods have two sets of order conditions: those arising from trees containing only meagre nodes, and those arising from trees containing at least one fat node. Trees containing fat nodes do not correspond to any trees in the exact solution and, as seen in Figure 3.1, the corresponding coefficients are set to zero [13].




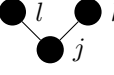




In order to build the relevant trees for Rosenbrock-K methods we need to have a closer look at the properties of the Jacobian approximation (2.3).

LEMMA 3.1 (Property of the Rosenbrock-Krylov approximate Jacobian (2.3)).  
For any  $0 \leq k \leq M - 1$  it holds that

$$\mathbf{A}_n^k f_n = \mathbf{J}_n^k f_n,$$



FIG. 3.1. *TW-trees and Rosenbrock-W conditions up to order three [13, Section IV.7].*

$a$		$f^J$	$\sum b_j = 1$
$b_1$		$f_K^J f^K$	$\sum b_j \alpha_{j,k} = 1/2$
$b_2$		$\mathbf{A}_{JK} f^K$	$\sum b_j \gamma_{j,k} = 0$
$c$		$f_{KL}^J f^K f^L$	$\sum b_j \alpha_{j,k} \alpha_{k,l} = 1/3$
$d_1$		$f_K^J f_L^K f^L$	$\sum b_j \alpha_{j,k} \alpha_{k,l} = 1/6$
$d_2$		$\mathbf{A}_{JK} f_L^K f^L$	$\sum b_j \gamma_{j,k} \alpha_{k,l} = 0$
$d_3$		$f_K^J \mathbf{A}_{KL} f^L$	$\sum b_j \alpha_{j,k} \gamma_{k,l} = 0$
$d_4$		$\mathbf{A}_{JK} \mathbf{A}_{KL} f^L$	$\sum b_j \gamma_{j,k} \gamma_{k,l} = 0$

where  $M = \dim(\mathcal{K}_M)$ .

*Proof.* Recall that  $\mathbf{V}_{n;M} \mathbf{V}_{n;M}^T$  is the orthogonal projector onto  $\mathcal{K}_M$ . If a vector  $u$  is in the Krylov space  $\mathcal{K}_M$ , its orthogonal projection onto  $\mathcal{K}_M$  is the vector itself:

$$u \in \mathcal{K}_M \quad \Rightarrow \quad \mathbf{V}_{n;M} \mathbf{V}_{n;M}^T u = u.$$

The proof of the Lemma is by finite induction. As the base case we have that

$$\mathbf{A}_n^0 f_n = \mathbf{J}_n^0 f_n = f_n.$$

Next we assume that  $\mathbf{A}_n^{i-1} f_n = \mathbf{J}_n^{i-1} f_n$  for some  $i \leq M-1$  and will show that  $\mathbf{A}_n^i f_n = \mathbf{J}_n^i f_n$ . By the definition of the approximate Jacobian (2.3) and our assumption it holds that

$$\mathbf{A}_n^i f_n = \mathbf{A}_n \cdot \mathbf{A}_n^{i-1} f_n = \mathbf{A}_n \cdot \mathbf{J}_n^{i-1} f_n = \mathbf{V}_{n;M} \mathbf{V}_{n;M}^T \mathbf{J}_n \mathbf{V}_{n;M} \mathbf{V}_{n;M}^T \cdot \mathbf{J}_n^{i-1} f_n.$$

Since  $M \geq i$  we have that  $\mathbf{J}_n^{i-1} f_n \in \mathcal{K}_M$ , and

$$\mathbf{V}_{n;M} \mathbf{V}_{n;M}^T \mathbf{J}_n^{i-1} f_n = \mathbf{J}_n^{i-1} f_n \quad \Rightarrow \quad \mathbf{A}_n^i f_n = \mathbf{V}_{n;M} \mathbf{V}_{n;M}^T \mathbf{J}_n \cdot \mathbf{J}_n^{i-1} f_n = \mathbf{V}_{n;M} \mathbf{V}_{n;M}^T \mathbf{J}_n^i f_n.$$

Since  $M \geq i+1$  we have that  $\mathbf{J}_n^i f_n \in \mathcal{K}_M$  and

$$\mathbf{V}_{n;M} \mathbf{V}_{n;M}^T \mathbf{J}_n^i f_n = \mathbf{J}_n^i f_n \quad \Rightarrow \quad \mathbf{A}_n^i f_n = \mathbf{J}_n^i f_n.$$

□

LEMMA 3.2 (Property of elementary differentials using the approximation (2.3)).  
 When the Rosenbrock-Krylov matrix approximation (2.3) is used, all linear TW-trees of order  $k \leq M$  correspond to a single elementary differential, regardless of the color of their nodes.

*Proof.* In a linear tree each node has only one child. A linear TW-tree with a fat root can be described by the sequence of its nodes, starting from the root. For example, the structure of a linear tree where the first  $\nu_1$  nodes from the root are fat, followed by  $\mu_1$  meagre nodes, etc. is described by the sequence  $(\circ^{\nu_1} \bullet^{\mu_1} \dots \circ^{\nu_p} \bullet^{\mu_p})$  with  $\mu_p > 1$  (since the leaf is a meagre node).

Consider now a tree of order  $k = \nu_1 + \mu_1 + \dots + \nu_p + \mu_p \leq M$ . The corresponding elementary differential has the form  $\mathbf{A}_n^{\nu_1} \mathbf{J}_n^{\mu_1} \dots \mathbf{A}_n^{\nu_p} \mathbf{J}_n^{\mu_p} f_n$ . Repeated applications of Lemma 3.1 reveal that

$$\mathbf{A}_n^{\nu_1} \mathbf{J}_n^{\mu_1} \dots \mathbf{A}_n^{\nu_p} \underbrace{\mathbf{J}_n^{\mu_p} f_n}_{=\mathbf{A}_n^{\mu_p} f_n} = \mathbf{A}_n^{\nu_1} \mathbf{J}_n^{\mu_1} \dots \underbrace{\mathbf{A}_n^{\nu_p + \mu_p} f_n}_{=\mathbf{J}_n^{\nu_p + \mu_p} f_n} = \dots = \mathbf{J}_n^{k-1} f_n.$$

Consequently, any linear TW-tree of order  $k \leq M$  has the same differential as the linear tree with only meagre nodes  $(\bullet^k)$ . □

An important consequence of Lemma 3.2 is that if a linear TW sub-tree with  $k \leq M$  nodes has a fat root, the corresponding differential is the same as for the linear tree with only meagre nodes  $(\bullet^k)$ . This observation allows us to essentially “recolor” linear TW sub-trees with a fat root (i.e., group them in classes of equivalence). This leads to the following.

DEFINITION 3.3 (TK-trees).

$$\begin{aligned} TK &= \{TW\text{-trees: no linear sub-tree has a fat root}\}; \\ TK(k) &= \{TW\text{-trees: no linear sub-tree of order} \\ &\quad \text{smaller than or equal to } k \text{ has a fat root}\}. \end{aligned}$$

For  $t \in TK$  let  $\rho(t)$  define the number of vertices. We denote by  $t = \bullet[t_1, \dots, t_m]$  a TK-tree tree with a meagre root linking the subtrees  $t_1, \dots, t_m$ , and by  $t = \circ[t_1]$  a TK-tree with a fat root to which the subtree  $t_1$  is connected. A special case is the single node tree  $\bullet[\ ]$ . The elementary differentials associated with TK-trees are the same as those of TW-trees, [13, Definition 7.5]. The TK-tree coefficients are constructed recursively as follows.

DEFINITION 3.4 (Coefficients of TK-trees). For  $t \in TK$

$$\phi_j(t) = \begin{cases} 1 & \text{if } t = \bullet[\ ] \\ \sum_{k_1, \dots, k_m} \alpha_{j, k_m} \phi_{k_1}(t_1) \dots \phi_{k_m}(t_m) & \text{if } t = \bullet[t_1, \dots, t_m], m \geq 2 \\ \sum_k \beta_{j, k} \phi_k(t_1) & \text{if } t = \bullet[t_1] \\ \sum_k \gamma_{j, k} \phi_k(t_1) & \text{if } t = \circ[t_1] \end{cases}$$

**3.1. Rosenbrock-K methods of type 1.** DEFINITION 3.5. A Rosenbrock-K method of type 1 is given by Algorithm 2 and uses an underlying Krylov subspace given by (2.1).

THEOREM 3.6 (Order conditions for Rosenbrock-K methods). A Rosenbrock-K method of type 1 has order  $p$  iff the underlying Krylov space (2.1) has dimension

$M \geq p$ , and the following order conditions hold:

$$(3.1a) \quad \sum_j b_j \phi_j(t) = \frac{1}{\gamma(t)} \quad \forall t \in T \quad \text{with } \rho(t) \leq p,$$

$$(3.1b) \quad \sum_j b_j \phi_j(t) = 0 \quad \forall t \in TK \setminus T \quad \text{with } \rho(t) \leq p.$$

Here  $\rho(t)$  is the number of vertices of the tree  $t$ , and  $\gamma(t)$  is the ‘‘product of  $\rho(t)$  and all orders of the trees which appear, if the roots, one after another, are removed from  $t$ ’’ [12, Section II.2].

*Proof.* The proof follows from our discussion and from the order conditions of Rosenbrock-W methods [13, Theorem 7.7].  $\square$

**THEOREM 3.7** (Order conditions for Rosenbrock-K methods with smaller Krylov space). *A Rosenbrock-K method of type 1 with an underlying Krylov space (2.1) of dimension  $M < p$  has order  $p$  iff condition (3.1a) holds, and, in addition:*

$$(3.1c) \quad \sum_j b_j \phi_j(t) = 0 \quad \forall t \in TK(M) \setminus T \quad \text{with } \rho(t) \leq p.$$

*Proof.* Follows immediately from Theorem 3.6.  $\square$

**REMARK 1.** *The number of required order conditions for Rosenbrock-K methods is substantially smaller than the number of order conditions for Rosenbrock-W methods.*

Figure 3.1 reveals that all TW-trees up to order three containing a fat root are linear, and so every tree containing a fat node can be recolored to contain only meagre nodes. Thus the order conditions for Rosenbrock-K methods are the same as those for classical Rosenbrock methods for up to order three (while Rosenbrock-W methods need four additional conditions). Figure 3.1 shows the TK-trees and order conditions for up to order four; Rosenbrock-K methods require only a single extra order condition for order four (while Rosenbrock-W methods require seventeen additional conditions). Finally, Figure 3.1 shows the four additional TK-trees and Rosenbrock-K conditions needed for order five.

### 3.2. Rosenbrock-K methods of type 2.

**DEFINITION 3.8.** *A Rosenbrock-K method of type 2 is given by Algorithm 2 and uses an enriched underlying Krylov subspace, where additional basis vectors are added to those in (2.1). The additional basis vectors are chosen such that different elementary differentials associated with trees in  $TK \setminus T$  are equal to those of similar trees in  $T$ . Consequently, the order conditions of a type 2 Rosenbrock-K method are the same as those of classical Rosenbrock methods.*








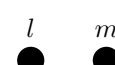

For example, consider the tree  $g_2$  in Table 3.1. The corresponding term in the Taylor series of the solution is  $\mathbf{A}_n \cdot f_{y,y}(f_n, f_n)$ . The application of the second derivative tensor  $f_{y,y}$  to a pair of function values results in the vector

$$(3.2) \quad u_{g_2}^k = (f_{y,y}(f_n, f_n))^k = \sum_{\ell, m=1}^N \frac{\partial^2 f^k}{\partial y^\ell \partial y^m} \Big|_{y=y_n} f^\ell(y_n) f^m(y_n), \quad k = 1, \dots, N.$$

To obtain a type 2 Rosenbrock-K method of order four the Krylov space (2.1) is extended as follows:

$$(3.3) \quad \begin{aligned} \mathcal{K}_{M+2}(\mathbf{J}_n, f_n) &= \text{span} \{ f_n, \mathbf{J}_n f_n, \dots, \mathbf{J}_n^{M-1} f_n, u_{g_2}, \mathbf{J}_n u_{g_2} \} \\ &= \text{span} \{ v_1, v_2, \dots, v_{M+2} \}. \end{aligned}$$

FIG. 3.2. TK-trees and Rosenbrock-Krylov conditions up to order four.

$a$		$f^J$	$\sum b_j = 1$
$b$		$f_K^J f^K$	$\sum b_j \beta_{j,k} = 1/2$
$c$		$f_{KL}^J f^K f^L$	$\sum b_j \alpha_{j,k} \alpha_{j,l} = 1/3$
$d$		$f_K^J f_L^K f^L$	$\sum b_j \beta_{j,k} \beta_{k,l} = 1/6$
$e$		$f_{KLM}^J f^K f^L f^M$	$\sum b_j \alpha_{j,k} \alpha_{j,l} \alpha_{j,m} = 1/4$
$f$		$f_{KM}^J f_L^K f^L f^M$	$\sum b_j \alpha_{j,k} \beta_{k,l} \alpha_{j,m} = 1/8$
$g_1$		$f_K^J f_{LM}^K f^L f^M$	$\sum b_j \alpha_{j,k} \alpha_{k,m} \alpha_{k,l} = 1/12$
$g_2$		$\mathbf{A}_{JK} f_{LM}^K f^L f^M$	$\sum b_j \gamma_{j,k} \alpha_{k,m} \alpha_{k,l} = 0$
$h$		$f_K^J f_L^K f_M^L f^M$	$\sum b_j \beta_{j,k} \beta_{k,l} \beta_{l,m} = 1/24$

The Jacobian approximation is  $\mathbf{A}_n = \mathbf{V}_{n;M} \mathbf{H}_{n;M} \mathbf{V}_{n;M}^T$  where  $\mathbf{V}_{n;M} \in \mathbb{R}^{N \times (M+2)}$ , and  $\mathbf{H}_{n;M} = \mathbf{V}_{n;M}^T \mathbf{J}_n \mathbf{V}_{n;M} \in \mathbb{R}^{(M+2) \times (M+2)}$  is no longer upper Hessenberg.

The construction (3.3) ensures that the elementary differential of the tree  $g_2 \in TW \setminus T$  coincides with the elementary differential of a regular Butcher tree:

$$\mathbf{A}_n f_{y,y}(f_n, f_n) = \mathbf{A}_n u_{g_2} = \mathbf{V}_{n;M} \mathbf{V}_{n;M}^T \mathbf{J}_n \underbrace{\mathbf{V}_{n;M} \mathbf{V}_{n;M}^T}_{=u_{g_2}} u_{g_2} = \mathbf{V}_{n;M} \mathbf{V}_{n;M}^T \mathbf{J}_n u_{g_2} = \mathbf{J}_n u_{g_2} = \mathbf{J}_n f_{y,y}(f_n, f_n).$$

We have the following interesting consequences.

REMARK 2. Any classical Rosenbrock method of order four (or higher) becomes a type 2 Rosenbrock-K method of order four when the Jacobian approximation (2.3) uses a six-dimensional extended Krylov space (more exactly, when the underlying Krylov space is (3.3) with  $M \geq 4$ ).

FIG. 3.3. Additional TK-trees and Rosenbrock-Krylov conditions for order five.

	$\mathbf{A}_{JK} f_{LMP}^K f^L f^M f^P$	$\sum \gamma_{j,k} \alpha_{k,l} \alpha_{k,m} \alpha_{k,p} = 0$
	$\mathbf{A}_{JK} f_{LKP}^K f_{LM}^L f^M f^P$	$\sum \gamma_{j,k} \alpha_{k,p} \alpha_{k,l} \alpha_{l,m} = 0$
	$\mathbf{A}_{JK} f_L^K f_{MP}^L f^P f^M$	$\sum \gamma_{j,k} \alpha_{k,l} \alpha_{l,m} \alpha_{l,p} = 0$
	$\mathbf{A}_{JK} \mathbf{A}_{KL} f_{MP}^L f^P f^M$	$\sum \gamma_{j,k} \gamma_{k,l} \alpha_{l,m} \alpha_{l,p} = 0$
	$f_K^J \mathbf{A}_{KL} f_{MP}^L f^P f^M$	$\sum \alpha_{j,k} \gamma_{k,l} \alpha_{l,m} \alpha_{l,p} = 0$

REMARK 3. General Rosenbrock-K methods of any order can be obtained by combining the type 2 and type 1 approaches. Specifically, some of the trees in  $TK \setminus T$  are recolored (to obtain the similar trees in  $T$ ) by extending the underlying Krylov space, i.e., by using a type 2 approach. The elementary differentials corresponding to the remaining trees in  $TK \setminus T$  are then cancelled by imposing additional type 1 order conditions.

**3.3. Implementation aspects.** The cost of Rosenbrock-K integration for large-scale systems is dominated by the cost of building the Krylov space at each step. The Arnoldi iteration requires  $M$  Jacobian-vector products, as well as vector operations totaling  $\mathcal{O}(M^2N)$  operations during orthogonalization [36]. Both Jacobian-vector products and vector operations can be efficiently parallelized.

Jacobian-vector can be obtained in several ways. Most straightforwardly the entire Jacobian matrix can be constructed and then a Jacobian-vector product can be calculated in the usual way. This process is expensive both in terms of storage and computation.

An alternative is to implement a routine that computes directly Jacobian-vector products without building the Jacobian matrix. Such a routine can be obtained

through forward-mode automatic differentiation [10] and its cost is similar to the ODE function computations. Large distributed applications rely on an infrastructure which partitions the solution vector  $y_n$  across nodes. The computation of the ODE function  $f_n$  is done in parallel. Data exchange of data among subdomains is needed in order to fulfill grid dependencies. Exact Jacobian-vector products  $J_n \cdot u$  are obtained by linearizing the ODE function  $f_n$  in the direction  $u$ . Therefore Jacobian-vector operations can be computed element by element, inherit the parallel structure of the ODE function calculation (e.g., parallelism obtained by domain decomposition), and can be implemented very efficiently using the same parallel software infrastructure. The same data partitioning can be used for both the solution  $y_n$  and the vector  $u$ . Note that successive Arnoldi iterations act on the distributed vectors  $u$  without any need for global communication (only local communication of the boundary elements is needed at each iteration).

Jacobian-vector products can also be approximated by finite differences of the form [28]

$$(3.4) \quad \mathbf{J}_n u \approx \frac{f(t_n, y_n + \delta u) - f(t_n, y_n)}{\delta}.$$

The increment  $\delta$  is related to machine precision. Equation (3.4) is sometimes referred to as a “matrix-free” approximation. For example, “Jacobian-free Newton-Krylov” methods [28] employ the approximation (3.4) within linear Krylov space solvers in the context of Newton iterations for nonlinear systems. Clearly the finite difference approximation (3.4) uses the same data partitioning for  $y_n$  and  $u$ , and inherits the parallel performance of the ODE function calculation.

Finite differences can also be used to approximate higher order derivatives. For example, the second derivative term (3.2) can be approximated by finite differences of Jacobian-vector products, as follows:

$$f_{y,y}(u, u) \approx \frac{f_y(t_n, y_n + \delta u) \cdot u - \mathbf{J}_n \cdot u}{\delta},$$

and where each Jacobian-vector product can also be approximated,

**3.4. Errors due to finite difference approximations.** An analysis of matrix-free Newton-Krylov methods is provided in [4, Theorem 2.3]. Assume that the Arnoldi process with the exact Jacobian-vector products produces  $\mathbf{H}_{n;M}$ ,  $\mathbf{V}_{n;M}$ , while the Arnoldi process using finite difference approximations (3.4) produces  $\tilde{\mathbf{H}}_{n;M}$  and  $\tilde{\mathbf{V}}_{n;M}$ . The errors in the finite difference approximations (3.4) during each Arnoldi iteration are

$$e_i = \frac{f(t_n, y_n + \delta_i \tilde{v}_i) - f(t_n, y_n)}{\delta_i} - \mathbf{J}_n \cdot \tilde{v}_i, \quad i = 1, \dots, M-1.$$

Collect these error vectors, together with  $e_0 = 0$  (the error in computing  $\tilde{v}_1 = f_n / \|f_n\|$ ), in the matrix

$$\mathbf{E} = [e_0, e_1, \dots, e_{M-1}] \in \mathbb{R}^{N \times M}.$$

According to [4, Theorem 2.3], the matrices  $\tilde{\mathbf{H}}_{n;M}$  and  $\tilde{\mathbf{V}}_{n;M}$  can be obtained by an application of the *exact* Arnoldi process (i.e., with exact Jacobian-vector products) to obtain a basis of the modified space

$$\mathcal{K}_M(\tilde{\mathbf{J}}_n, f_n) \quad \text{with} \quad \tilde{\mathbf{J}}_n := \mathbf{J}_n + \mathbf{E} \tilde{\mathbf{V}}_{n;M}^T.$$

According to Lemma 3.1, the matrix approximation  $\tilde{\mathbf{A}}_n = \tilde{\mathbf{V}}_{n;M} \tilde{\mathbf{H}}_{n;M} \tilde{\mathbf{V}}_{n;M}^T$  has the following property for any  $0 \leq k \leq M - 1$

$$\begin{aligned} \tilde{\mathbf{A}}_n^k f_n &= \tilde{\mathbf{J}}_n^k f_n \\ &= \left( \mathbf{J}_n + \mathbf{E} \tilde{\mathbf{V}}_{n;M}^T \right)^k f_n \\ &= \mathbf{J}_n^k f_n + \sum_{i=0}^{k-1} \binom{k}{i} \mathbf{J}_n^i (\mathbf{E} \tilde{\mathbf{V}}_{n;M}^T)^{k-i} f_n \\ &= \mathbf{J}_n^k f_n + \mathcal{O}(\|\mathbf{E}\|), \end{aligned}$$

where for the last equality we have made the assumption that the Jacobian powers are uniformly bounded.

When exact Jacobian-vector products are used no additional type 1 Rosenbrock-K order conditions are imposed for trees in  $TW \setminus TK$ . Similarly, when higher derivatives are computed exactly no additional order conditions are needed for type 2 methods. When finite difference approximations are used, however, the elementary differentials of trees in  $TW \setminus TK$  appear in the expansion of the numerical solution with nonzero coefficients of size  $\mathcal{O}(\|\mathbf{E}\|)$ , i.e., of the size of the *absolute* errors incurred in the finite difference approximations. If the finite difference approximations are not sufficiently accurate the order of the Rosenbrock-K methods may be lost.

For example, consider the tree  $d_2 \in TW \setminus TK$  in Figure 3.1. When finite differences are used, it contributes the following  $\mathcal{O}(\|\mathbf{E}\| h^2)$  term to the local error

$$(3.5) \quad h^2 \left( \sum_{j=1}^s b_j \gamma_{j,k} \right) (\mathbf{E} \tilde{v}_1) \|f_n\|.$$

In order to ensure that the Rosenbrock-K method preserves its order  $p$ , a sufficient condition is that the finite difference errors are bounded by

$$\|\mathbf{E}\| \leq C h^{p-1}.$$

Without assuming the uniform boundedness of the Jacobian a sufficient condition is  $\|\mathbf{J}_n^i (\mathbf{E} \tilde{\mathbf{V}}_{n;M}^T)^{k-i}\| \leq C h^{p-1}$ ,  $i = 0, \dots, k - 1$ . When the exact Jacobian-vector products are unavailable, and when the finite difference approximations cannot be computed this accurately, it may be advantageous to choose Rosenbrock-K methods whose coefficients satisfy the full set of Rosenbrock-W conditions.

**4. Construction of Rosenbrock-K methods of order four.** We now construct practical type 1 Rosenbrock-K methods of order four. We consider the case with  $\gamma_{i,i} = \gamma$  for all  $i$ , and denote

$$\beta_i = \sum_{j=1}^i \beta_{i,j} = \alpha_i + \gamma_i, \quad \beta'_i = \sum_{j=1}^{i-1} \beta_{i,j}.$$

We examine numerically the linear stability properties of the resulting methods when using the exact Jacobian so that  $\mathbf{A}_n = \mathbf{J}_n$ . Rosenbrock-K methods share the same stability function with classical Rosenbrock methods

$$(4.1) \quad R(z) = 1 + zb^T (\mathbf{I}_{s \times s} - z\beta)^{-1} \mathbf{1},$$

where  $\mathbf{1} \in \mathbb{R}^s$  is a vector of ones.

#### 4.1. ROK4a: a four stages, fourth order, L-stable Rosenbrock-K method.

We start with constructing a four stages, fourth order Rosenbrock-K method. The order conditions are as follows:

(4.2)

$$\begin{aligned}
(a) \quad b_1 + b_2 + b_3 + b_4 &= 1 \\
(b) \quad b_2\beta'_2 + b_3\beta'_3 + b_4\beta'_4 &= \frac{1}{2} - \gamma = p_{2,1}(\gamma) \\
(c) \quad b_2\alpha_2^2 + b_3\alpha_3^2 + b_4\alpha_4^2 &= \frac{1}{3} \\
(d) \quad b_3(\beta_{3,2}\beta'_2) + b_4(\beta_{4,2}\beta'_2 + \beta_{4,3}\beta'_3) &= \frac{1}{6} - \gamma + \gamma^2 = p_{3,2}(\gamma) \\
(e) \quad b_2\alpha_2^3 + b_3\alpha_3^3 + b_4\alpha_4^3 &= \frac{1}{4} \\
(f) \quad b_3\alpha_3\alpha_{3,2}\beta'_2 + b_4\alpha_4(\alpha_{4,2}\beta'_2 + \alpha_{4,3}\beta'_3) &= \frac{1}{8} - \frac{1}{3}\gamma = p_{4,2}(\gamma) \\
(g_1) \quad b_3\alpha_{3,2}\alpha_2^2 + b_4(\alpha_{4,2}\alpha_2^2 + \alpha_{4,3}\alpha_3^2) &= \frac{1}{12} \\
(g_2) \quad b_3\gamma_{3,2}\alpha_2^2 + b_4(\gamma_{4,2}\alpha_2^2 + \gamma_{4,3}\alpha_3^2) &= -\frac{1}{3}\gamma \\
(h) \quad b_4\beta_{4,3}\beta_{3,2}\beta'_2 &= \frac{1}{24} - \frac{1}{2}\gamma + \frac{3}{2}\gamma^2 - \gamma^3 = p_{4,4}(\gamma)
\end{aligned}$$

To solve (4.2) we follow the solution process outlined in [13, Section IV.7]. First we choose  $\gamma = 0.572816062482135$  so that  $R(\infty) = 0$ , where  $R(z)$  is the stability function of the method. We then treat equations (4.2.a), (4.2.c), and (4.2.e) separately, as a linear system in  $b_i$ 's. We make the arbitrary choices

$$b_3 = 0, \quad \alpha_2 = \frac{1}{2}, \quad \alpha_3 = 1, \quad \alpha_4 = 1,$$

and then solve the system

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & \alpha_2^2 & \alpha_4^2 \\ 0 & \alpha_2^3 & \alpha_4^3 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_4 \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1}{3} \\ \frac{1}{4} \end{bmatrix}$$

to obtain  $b_1, b_2$ , and  $b_4$ .

In order to allow for the existence of an embedded method of order three we require that the third order conditions are not satisfied uniquely. The following equation guarantees that by setting the determinant of the system of third order conditions to zero:

$$(4.3) \quad (\beta'_2\alpha_4^2 - \beta'_4\alpha_2)\beta_{3,2}\beta'_2 = (\beta'_2\alpha_3^2 - \beta'_3\alpha_2^2) \sum_{j=2}^3 \beta_{4,j}\beta'_j.$$

We now take  $\beta_{4,3}$  as a free parameter and compute  $\beta_{3,2}\beta'_2$  from (4.2.h) and  $(\beta_{4,2}\beta'_2 + \beta_{4,3}\beta'_3)$  from (4.2.d). Inserting these expressions into (4.3) yields a relation between  $\beta'_2, \beta'_3, \beta'_4$ . Eliminating  $(b_4\beta_{4,2} + b_3\beta_{3,2})$  from (4.2.d), and from  $\{(4.2.g_1) + (4.2.g_2)\}$ , yields a second relation. A third relation is obtained from (4.2.b), and this leads to the following system for  $\beta'_2, \beta'_3, \beta'_4$ :

$$\begin{bmatrix} \alpha_4^2 \frac{p_{4,4}}{b_4\beta_{4,3}} - \alpha_3^2 \frac{p_{3,2}}{b_4} & \alpha_2^2 \frac{p_{3,2}}{b_4} & -\alpha_2^2 \frac{p_{4,4}}{b_4\beta_{4,3}} \\ b_2 & b_3 & b_4 \\ b_4\beta_{4,3}\alpha_3^2 - p_{4,3} & -b_4\beta_{4,3}\alpha_2^2 & 0 \end{bmatrix} \begin{bmatrix} \beta'_2 \\ \beta'_3 \\ \beta'_4 \end{bmatrix} = \begin{bmatrix} 0 \\ p_{2,1} \\ -\alpha_2^2 p_{3,2} \end{bmatrix}.$$



$\gamma = 0.572816062482135$				
$\alpha_{2,1} =$		1	$\gamma_{2,1} =$	-1.91153192976055097824
$\alpha_{3,1} =$	0.10845300169319391758		$\gamma_{3,1} =$	0.32881824061153522156
$\alpha_{3,2} =$	0.39154699830680608241		$\gamma_{3,2} =$	0.0
$\alpha_{4,1} =$	0.43453047756004477624		$\gamma_{4,1} =$	0.03303644239795811290
$\alpha_{4,2} =$	0.14484349252001492541		$\gamma_{4,2} =$	-0.24375152376108235312
$\alpha_{4,3} =$	-0.07937397008005970166		$\gamma_{4,3} =$	-0.17062602991994029834
$b_1 =$	0.16666666666666666667	$\widehat{b}_1 =$	0.50269322573684235345	
$b_2 =$	0.16666666666666666667	$\widehat{b}_2 =$	0.27867551969005856226	
$b_3 =$	0.0	$\widehat{b}_3 =$	0.21863125457309908428	
$b_4 =$	0.66666666666666666667	$\widehat{b}_4 =$	0.0	

TABLE 4.1

Coefficients of ROK4a, a fourth order, L-stable, type 1 Rosenbrock-K method.

Here we make the arbitrary choice

$$\beta_{43} = -\frac{1}{4}$$

and compute  $\beta_{3,2}$  and  $\beta_{4,2}$  from

$$\beta_{3,2} = \frac{p_{4,4}}{b_4\beta_{4,3}\beta'_2}, \quad \beta_{4,2} = \frac{p_{3,2} - b_4\beta_{4,3}\beta'_3}{b_4\beta'_2}.$$

Next we impose directly equations (4.2.f), (4.2.g<sub>1</sub>), and (4.2.g<sub>2</sub>) along with the definition of  $\beta_{i,j}$ :

$$\begin{aligned} b_3\alpha_{3,2}\beta'_2 + b_4(\alpha_{4,2}\beta'_2 + \alpha_{4,3}\beta_3) &= p_{4,2} \\ b_3\alpha_{3,2}\alpha_2^2 + b_4(\alpha_{4,2}\alpha_2^2 + \alpha_{4,3}\alpha_3^2) &= \frac{1}{12} \\ b_3\gamma_{3,2}\alpha_2^2 + b_4(\gamma_{4,2}\alpha_2^2 + \gamma_{4,3}\alpha_3^2) &= -\frac{1}{3}\gamma \\ \gamma_{3,2} + \alpha_{3,2} &= \beta_{3,2} \\ \gamma_{4,2} + \alpha_{4,2} &= \beta_{4,2} \\ \gamma_{4,3} + \alpha_{4,3} &= \beta_{4,3} \end{aligned}$$

Finally  $\alpha_{i,1}$  and  $\beta_{i,1}$  follow immediately from the definition of  $\alpha_i$  and  $\beta'_i$  respectively. The coefficient values for this method, named ROK4a, are given in Table 4.1.

The choice of  $\gamma$  ensures that for the main method  $R(\infty) = 0$ . The embedded method has  $\widehat{R}(\infty) = -0.55$  Figure 4.1 shows the stability function values for both the main and embedded methods along the imaginary axis. We see that the absolute function values are below one, which implies that the main ROK4a method is L-stable, and the embedded method is strongly A-stable (i.e.,  $|\widehat{R}(\infty)| < 1$ ).

It is important to note that the stability results presented here apply to the case where a full Jacobian is used, and does not account for the impact of Krylov approximation. The impact of the Krylov approximation on the stability will be the subject of future work.

Exact stability requirements when making use of the Krylov approximation of the Jacobian are as yet undetermined, though a result by Wensch in [47] gives reason to believe that the size of the Krylov subspace must be as large as the number of stiff variables in the underlying problem.

$\gamma = 0.31$			
$\alpha_{2,1} =$	1.0	$\gamma_{2,1} =$	-22.824608269858540
$\alpha_{3,1} =$	0.5306333333333333	$\gamma_{3,1} =$	-69.343635255712726
$\alpha_{3,2} =$	-0.0306333333333333	$\gamma_{3,2} =$	-0.0306333333333333
$\alpha_{4,1} =$	0.8944444444444444	$\gamma_{4,1} =$	404.7106882480958
$\alpha_{4,2} =$	0.0555555555555556	$\gamma_{4,2} =$	0.0555555555555556
$\alpha_{4,3} =$	0.05	$\gamma_{4,3} =$	0.05
$\alpha_{5,1} =$	0.7383333333333333	$\gamma_{5,1} =$	-0.5716666666666667
$\alpha_{5,2} =$	-0.1216666666666667	$\gamma_{5,2} =$	-0.1216666666666667
$\alpha_{5,3} =$	0.3333333333333333	$\gamma_{5,3} =$	0.3333333333333333
$\alpha_{5,4} =$	0.05	$\gamma_{5,4} =$	0.05
$\alpha_{6,1} =$	-0.096929102825711	$\gamma_{6,1} =$	0.263595769492377
$\alpha_{6,2} =$	-0.1216666666666667	$\gamma_{6,2} =$	-0.1216666666666667
$\alpha_{6,3} =$	1.045582889789120	$\gamma_{6,3} =$	-0.378916223122453
$\alpha_{6,4} =$	0.173012879703258	$\gamma_{6,4} =$	-0.073012879703258
$\alpha_{6,5} =$	0.0	$\gamma_{6,5} =$	0
$b_1 =$	0.1666666666666667	$\hat{b}_1 =$	0.1666666666666667
$b_2 =$	-0.2433333333333333	$\hat{b}_2 =$	-0.2433333333333333
$b_3 =$	0.6666666666666667	$\hat{b}_3 =$	0.6666666666666667
$b_4 =$	0.1000000000000000	$\hat{b}_4 =$	0.1
$b_5 =$	0.0	$\hat{b}_5 =$	0.31
$b_6 =$	0.31	$\hat{b}_6 =$	0.0

TABLE 4.2

Coefficients of ROK4b, a fourth order, stiffly accurate, type 1 Rosenbrock-K method.

**4.2. ROK4b: a six stages, fourth order stiffly accurate Rosenbrock-K method.** Stiff accuracy is a desirable property when solving very stiff systems or index-1 differential algebraic equations. A stiffly accurate Rosenbrock method [13, Section IV.4] is characterized by the property

$$b_i = \beta_{s,i}, \quad i = 1, \dots, s.$$

We have derived a six-stage, stiffly accurate, fourth-order Rosenbrock-K method, named ROK4b. For brevity we do not show here the order conditions, nor we present the solution method. The coefficients have been obtained through a process similar to that outlined in Section 4.3. The ROK4b method coefficients are shown in Table 4.2. ROK4b has the additional benefit of both the main and embedded methods are L-stable. Figure 4.1 shows the stability functions of the main and embedded methods of ROK4b evaluated along the imaginary axis.

**4.3. ROK4p: a five stages, fourth order, parabolic Rosenbrock-K method.**

Due to their low stage order Rosenbrock methods can be marred by order reduction when solving initial value problems arising from the semi-discretization of PDEs. The following set of additional conditions guarantees the full order of convergence for Rosenbrock methods applied to semi-discrete *parabolic* PDEs [31, 32]:

$$(4.4) \quad b^T \beta^j (2\beta^2 \mathbf{1} - \alpha^2) = 0 \quad \text{for } p-2 \leq j \leq s-1 \text{ and } p \geq 3.$$

Here  $b = (b_i)_{i=1,\dots,s}$ ,  $\alpha = (\alpha_i)_{i=1,\dots,s}$ ,  $\beta = (\beta_i)_{i=1,\dots,s}$ ,  $p$  is the order of the method, and  $s$  is the number of stages. Multiplications are understood component-wise. We

will call a Rosenbrock method *parabolic* if it satisfies (4.4).

The order conditions for a five-stage, fourth-order, parabolic Rosenbrock-K are:

$$\begin{aligned}
(a) \quad & b_1 + b_2 + b_3 + b_4 + b_5 & = & 1 \\
(b) \quad & b_2\beta'_2 + b_3\beta'_3 + b_4\beta'_4 + b_5\beta'_5 & = & p_{2,1}(\gamma) \\
(c) \quad & b_2\alpha_2^2 + b_3\alpha_3^2 + b_4\alpha_4^2 + b_5\alpha_5^2 & = & \frac{1}{3} \\
(d) \quad & b_3\beta_{3,2}\beta'_2 + b_4(\beta_{4,2}\beta'_2 + \beta_{4,3}\beta'_3) \\
& \quad + b_5(\beta_{5,2}\beta'_2 + \beta_{5,3}\beta'_3 + \beta_{5,4}\beta'_4) & = & p_{3,2}(\gamma) \\
(e) \quad & b_2\alpha_2^3 + b_3\alpha_3^3 + b_4\alpha_4^3 + b_5\alpha_5^3 & = & \frac{1}{4} \\
(f) \quad & b_3\alpha_3\alpha_{3,2}\beta'_2 + b_4\alpha_4(\alpha_{4,2}\beta'_2 + \alpha_{4,3}\beta'_3) \\
& \quad + b_5\alpha_5(\alpha_{5,2}\beta'_2 + \alpha_{5,3}\beta'_3 + \alpha_{5,4}\beta'_4) & = & p_{4,2}(\gamma) \\
(4.5) \quad (g_1) \quad & b_3\alpha_3\alpha_{3,2}\alpha_2^2 + b_4(\alpha_{4,2}\alpha_2^2 + \alpha_{4,3}\alpha_3^2) \\
& \quad + b_5(\alpha_{5,2}\alpha_2^2 + \alpha_{5,3}\alpha_3^2 + \alpha_{5,4}\alpha_4^2) & = & \frac{1}{12} \\
(g_2) \quad & b_3\gamma_{3,2}\alpha_2^2 + b_4(\gamma_{4,2}\alpha_2^2 + \gamma_{4,3}\alpha_3^2) \\
& \quad + b_5(\gamma_{5,2}\alpha_2^2 + \gamma_{5,3}\alpha_3^2 + \gamma_{5,4}\alpha_4^2) & = & -\frac{1}{3}\gamma \\
(h) \quad & b_4\beta_{4,3}\beta_{3,2}\beta'_2 + b_5(\beta_{5,3}\beta_{3,2}\beta'_2 + \beta_{5,4}\beta_{4,2}\beta'_2 + \beta_{5,4}\beta_{4,3}\beta'_3) & = & p_{4,4}(\gamma) \\
(i) \quad & 2b_5\beta_{5,4}\beta_{4,3}\beta_{3,2}\beta'_2 - b_4\beta_{4,3}\beta_{3,2}\alpha_2^2 - b_5\beta_{5,3}\beta_{3,2}\alpha_2^2 \\
& \quad - b_5\beta_{5,4}\beta_{4,2}\alpha_2^2 - b_5\beta_{5,4}\beta_{4,3}\alpha_3^2 & = & \pi_1(\gamma) \\
(j) \quad & b_5\beta_{5,4}\beta_{4,3}\beta_{3,2}\alpha_2^2 & = & \pi_2(\gamma) \\
(k) \quad & 0 & = & \pi_3(\gamma)
\end{aligned}$$

where the polynomials  $p_{i,j}(\gamma)$  are defined in (4.2), and

$$\begin{aligned}
\pi_1(\gamma) &= 2\gamma p_{4,3} - 8\gamma p_{4,4}(\gamma) + \frac{1}{3}\gamma^2 - 12\gamma^2 p_{3,2}(\gamma) - 8\gamma^3 p_{2,1}(\gamma) - 2\gamma^4, \\
\pi_2(\gamma) &= 3\gamma\pi_1(\gamma) - 3\gamma^2 p_{4,3}(\gamma) + 20\gamma^2 p_{4,4}(\gamma) - \frac{1}{3}\gamma^3 + 20\gamma^3 p_{3,2}(\gamma) \\
& \quad + 10\gamma^4 p_{2,1}(\gamma) + 2\gamma^5, \\
\pi_3(\gamma) &= -4\gamma\pi_2(\gamma) + 6\gamma^2\pi_1(\gamma) - 4\gamma^3 p_{4,3}(\gamma) + 40\gamma^3 p_{4,4}(\gamma) - \frac{1}{3}\gamma^4 \\
& \quad + 30\gamma^4 p_{3,2}(\gamma) + 12\gamma^5 p_{2,1}(\gamma) + 2\gamma^6.
\end{aligned}$$

The approach to solve the system of equations (4.5) is similar to that used for (4.2). A sequence of linear systems is constructed, and for each system arbitrary choices are made for the values of some parameters. A numerical genetic optimization algorithm is employed to select free parameter values which lead to method coefficients of acceptable magnitudes. The coefficients of the resulting method, named ROK4p, are given in Table 4.3.

The choice of  $\gamma$  ensures that for the main method  $R(\infty) = 0$ . The embedded method has  $\widehat{R}(\infty) = 0.24$ . Figure 4.1 shows the stability function values for both the main and embedded methods along the imaginary axis. We see that the absolute function values are below one, which implies that the main ROK4p method is L-stable, and the embedded method is strongly A-stable.

$\gamma = 0.572816062482135$					
$\alpha_{2,1}$	=	0.7579000000000000	$\gamma_{2,1}$	=	-0.7579000000000000
$\alpha_{3,1}$	=	0.1704000000000000	$\gamma_{3,1}$	=	-0.295086678808293
$\alpha_{3,2}$	=	0.8211000000000000	$\gamma_{3,2}$	=	0.1789000000000000
$\alpha_{4,1}$	=	1.196218621274069	$\gamma_{4,1}$	=	-1.836333117783808
$\alpha_{4,2}$	=	0.2977000000000000	$\gamma_{4,2}$	=	-0.2477000000000000
$\alpha_{4,3}$	=	-1.433618621274069	$\gamma_{4,3}$	=	1.681409044712106
$\alpha_{5,1}$	=	-0.010650410785863	$\gamma_{5,1}$	=	-0.197089800872483
$\alpha_{5,2}$	=	0.1421000000000000	$\gamma_{5,2}$	=	-0.684644029868020
$\alpha_{5,3}$	=	-0.129349589214137	$\gamma_{5,3}$	=	0.166330242942910
$\alpha_{5,4}$	=	0.3928000000000000	$\gamma_{5,4}$	=	0.0000000000000000
$b_1$	=	0.0560000000000000	$\hat{b}_1$	=	-0.186875355621256
$b_2$	=	0.116601238130482	$\hat{b}_2$	=	-0.250433793031115
$b_3$	=	0.1603000000000000	$\hat{b}_3$	=	0.326360736478684
$b_4$	=	-0.031109354304222	$\hat{b}_4$	=	0.110948412173687
$b_5$	=	0.698208116173739	$\hat{b}_5$	=	1.0000000000000000

TABLE 4.3

Coefficients of ROK $_4p$ , a fourth order, parabolic, type 1 Rosenbrock-K method.

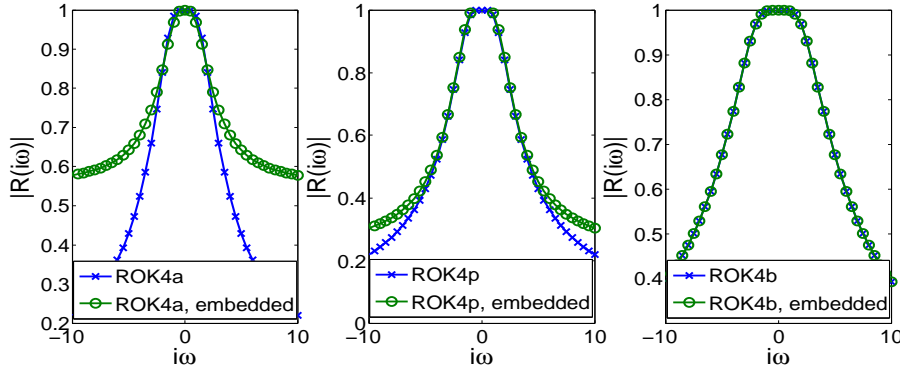


FIG. 4.1. Stability functions for the main and embedded methods of ROK4a, ROK4p, and ROK4b.

**5. Numerical Results.** Here we present some results from numerical experiments verifying the properties of the methods discussed above, as well as comparing performance of Rosenbrock-Krylov methods with several standard classical Rosenbrock and Rosenbrock-W methods. RANG3 is a third order Rosenbrock-W method [34], RODAS4 is a fourth order, stiffly accurate classical Rosenbrock method [13, Section IV.10], and ROS4 is a fourth order, L-stable, classical Rosenbrock method [13, Section IV.10].

**5.1. Lorenz 96.** The nonlinear test is carried out with the Lorenz-96 model [30]. This chaotic model has  $N = 40$  states, periodic boundary conditions, and is described by the following equations:

$$(5.1) \quad \begin{aligned} \frac{dy_j}{dt} &= -y_{j-1} (y_{j-2} - y_{j+1}) - y_j + F, \quad j = 1, \dots, N, \\ y_{-1} &= y_{N-1}, \quad y_0 = y_N, \quad y_{N+1} = y_1. \end{aligned}$$

The forcing term is  $F = 8.0$ , with  $t \in [0, 0.3]$ .

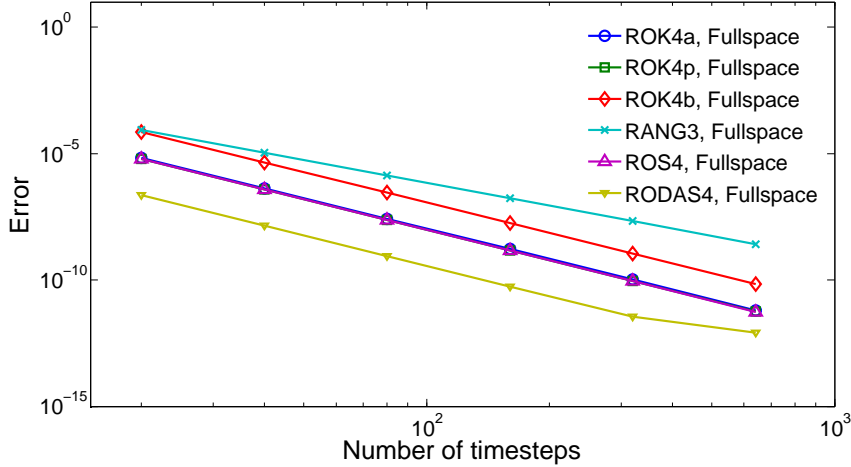


FIG. 5.1. Precision diagram for Lorenz 96, showing convergence order of methods using a full Jacobian.

	RANG3	ROS4	RODAS4	ROK4a	ROK4p	ROK4b
M = N	2.99	4.01	3.99	4.01	3.99	3.99
M = 4	2.99	3.03	3.05	4.01	3.98	3.99

TABLE 5.1  
Convergence Rates on Lorenz 96.

Table 5.1 shows the convergence orders of all methods applied to the Lorenz-96 system, using both the full Jacobian as well as a four dimensional Krylov approximation of the Jacobian. Figure 5.1 verifies numerically the theoretical order results for all methods using the full Jacobian.

Recall that all methods satisfying the classical Rosenbrock order conditions are also Rosenbrock-K methods of at least order three. Table 5.1 shows this property, where the third order method RANG3 maintains its order and both fourth order methods, ROS4 and RODAS4, reduce to third order while using the approximate Jacobian.

**5.2. Dissipative Burger's equation.** We apply the newly derived methods to an ODE system coming from a semi-discretization of a partial differential equation using the method of lines. The dissipative Burger's equation is a one-dimensional PDE described by

$$(5.2) \quad \frac{du}{dt} + \frac{d}{dx} \left( \frac{1}{2} u^2 \right) = \varepsilon \frac{d^2 u}{dx^2}, \quad x \in [0, 10], \quad t \in [0, 0.5], \quad \varepsilon = .001,$$

with homogeneous boundary conditions, and initial condition

$$u(x, t = 0) = \frac{1}{6} \sin^2 \left( \frac{1}{5} \pi x \right) (1 - x^2), \quad \varepsilon = .001.$$

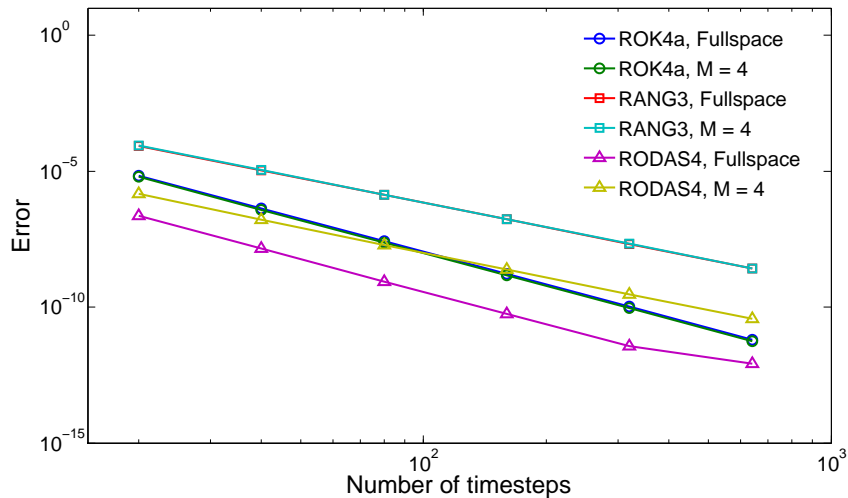


FIG. 5.2. Precision diagram showing the convergence order of ROK4a, RANG3, and RODAS4 using the full and Krylov approximated Jacobian.

The spatial discretization is a Nodal Discontinuous Galerkin method using equispaced fourth-order elements, making use of the code base provided for [14].

Figures 5.3 and 5.4 show a performance comparison of the newly proposed methods with both ROS4 and RODAS4. The figures show that for problems of even modest size, Rosenbrock-Krylov methods have comparable efficiency with previously existing methods. The increase in relative efficiency between Rosenbrock-K methods and the classical Rosenbrock methods as the problem size increases is a good indicator that Rosenbrock-K methods are likely to be much more efficient than full space methods as problem size increases, and the benefits of solving a reduced system become more pronounced.

**5.3. CBM-IV.** Here we give some results for ROK methods applied to a stiff system of ODEs coming from a KPP MATLAB implementation of the CBM-IV model [9]. This problem is based on the Carbon Bond Mechanism IV (CBM-IV), consisting of 32 chemical species involved in 70 thermal and 11 photolytic reactions [37].

While CBM-IV is a perfect example of a problem for which Rosenbrock-K methods are a poor choice, due to its small size and relatively large number of stiff variables, it does allow us to illustrate numerically the relationship between stability and choice of Krylov basis size. Figure 5.5 shows the number of timesteps, normalized to a full space solution of the respective method, required to obtain a reasonable solution in a single day simulation of the CBM-IV model. The number of timesteps required for a full space solution are given in Table 5.2.

ROK4a	ROK4p	ROK4b
1301	10270	261

TABLE 5.2

Number of required timesteps in a full space solution with tolerances of  $10^{-2}$ .

We see from this figure that for small Krylov basis sizes, Rosenbrock-K methods

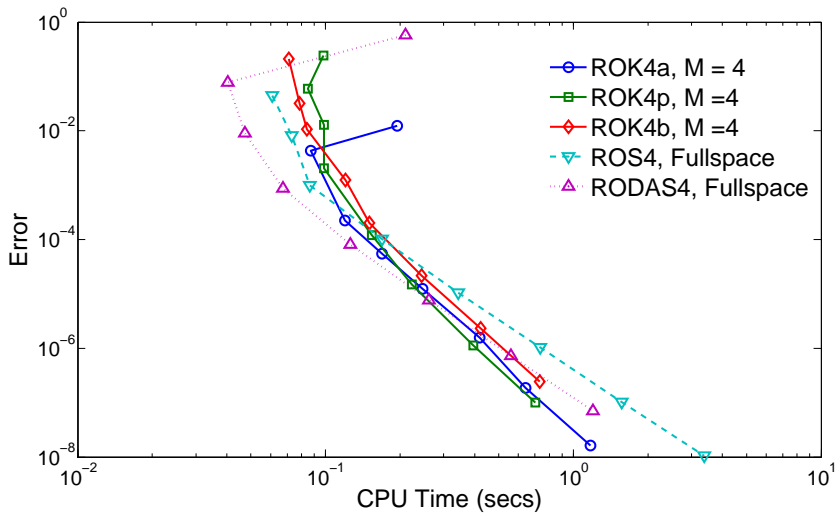


FIG. 5.3. Precision diagram for Burger's equation using 10 fourth order elements.

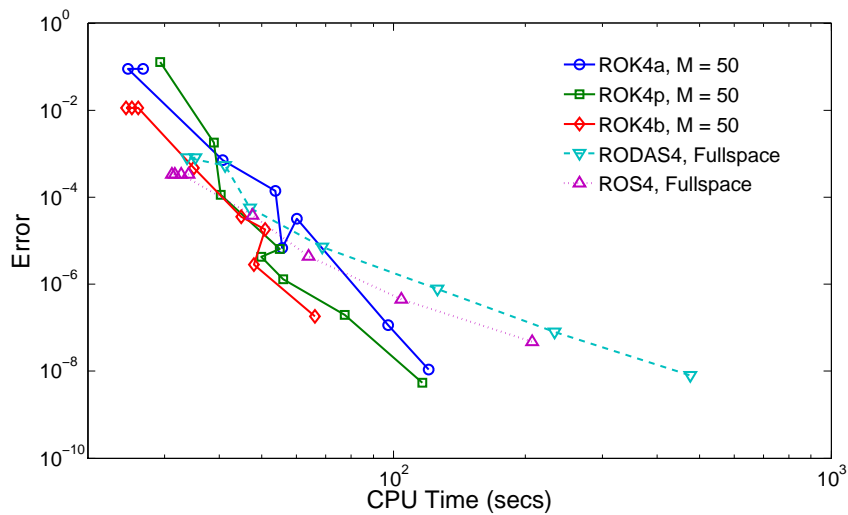


FIG. 5.4. Precision diagram for Burger's equation using 400 fourth order elements.

are unstable. However, as the size of the Krylov space nears the size of the full space the behavior of the Rosenbrock-K methods approaches that of the full space method.

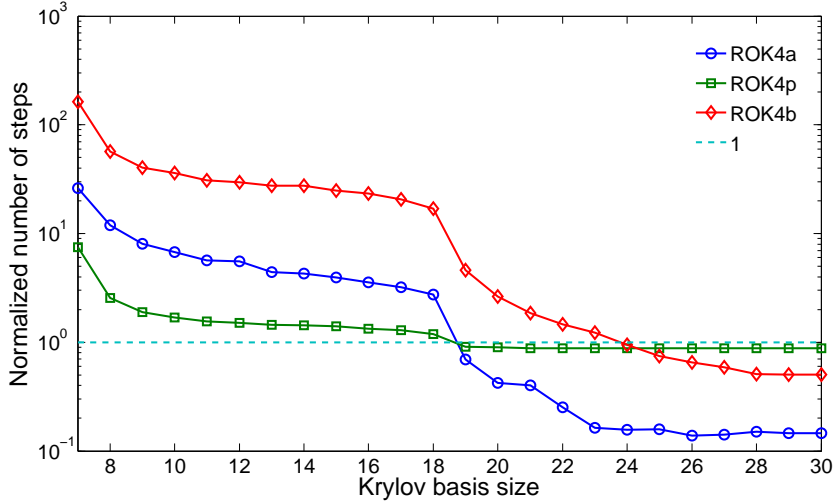


FIG. 5.5. Number of accepted steps in a full space solution with tolerances of  $10^{-2}$  for CBM-IV.

**5.4. Shallow water equations.** We examine the relative performance of the methods on the shallow water equations [29].

$$(5.3a) \quad \frac{\partial}{\partial t} h + \frac{\partial}{\partial x} (uh) + \frac{\partial}{\partial y} (vh) = 0$$

$$(5.3b) \quad \frac{\partial}{\partial t} (uh) + \frac{\partial}{\partial x} \left( u^2 h + \frac{1}{2} g h^2 \right) + \frac{\partial}{\partial y} (uvh) = 0$$

$$(5.3c) \quad \frac{\partial}{\partial t} (vh) + \frac{\partial}{\partial x} (vuh) + \frac{\partial}{\partial y} \left( v^2 h + \frac{1}{2} g h^2 \right) = 0,$$

with reflective boundary conditions, where  $u(x, y, t)$ ,  $v(x, y, t)$  are the flow velocity components and  $h(x, y, t)$  is the fluid height. After spatial discretization using centered finite differences on a  $32 \times 32$  grid the system (5.3) is brought to the standard ODE form (1.1) with

$$y = [u \ v \ h]^T \in \mathbb{R}^N, \quad f_y(t, y) = \mathbf{J} \in \mathbb{R}^{N \times N}, \quad N = 3072.$$

For all experiments we report only the time discretization errors, calculated against a reference solution computed by MATLAB's ODE15s solver with absolute and relative tolerances set to  $10^{-12}$ .

Figure 5.7 gives an efficiency comparison of the Rosenbrock-K and classical Rosenbrock methods, all methods make use of a sparse Jacobian matrix. This problem illustrates the scalability of the Rosenbrock-K methods, when the stiff subspace of the problem is kept relatively small. Here there are 3072 state variables, but the Rosenbrock-K methods require only eight basis vectors for stability, and so the cost of computing the Krylov space and solving the small system is much smaller than solving the linear system in the full space.

Table 5.3 shows the convergence rates of the three ROK methods applied with a constant timestep to solve the shallow water problem. We consider the exact Jacobian implementation as well as reduced space approximations of the Jacobian. We compute



	ROK4a	ROK4p	ROK4b
Fullspace	3.85	3.87	3.94
Exact Jacobian	3.86	3.88	3.94
Finite-difference	3.86	3.88	3.94
Low accuracy finite-difference	1.99	0.99	2.01

TABLE 5.3  
Convergence rates for shallow water equations.

the Krylov space approximations using exact Jacobian-vector products and using finite difference approximations. In order to assess the impact of the finite difference errors we consider both an accurate implementation of the Jacobian-vector products, where the increment  $\delta$  in (3.4) is carefully chosen, as well as an inaccurate implementation where a large, fixed value of  $\delta$  is used.

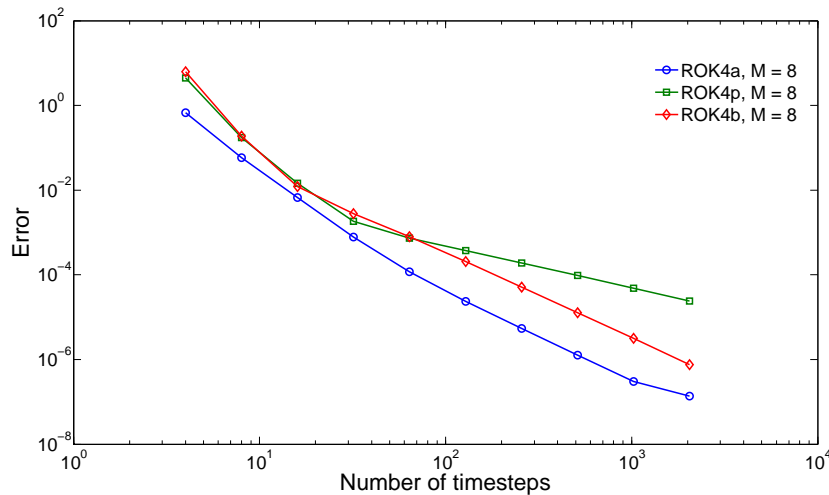


FIG. 5.6. Precision diagram for different methods on the shallow water equations problem. The lower order convergence rates are due to the low accuracy finite-difference approximation of Jacobian-vector products.

To verify the results presented in section 3.4 we perform a convergence study for the three ROK methods using a low accuracy finite-difference approximation of the Jacobian-vector products applied to the shallow water equations. Results are presented in Figure 5.6. The three curves show a distinct change in slope. When the finite-difference error is small compared to the timestep the methods have a convergence orders 3.4–3.7. When the finite-difference error becomes large compared to the timestep ROK4a and ROK4b show second-order and ROK4p shows only first-order. The difference in order when the finite-difference approximation is poor can be explained as follows. ROK4a and ROK4b satisfy the second-order W condition of Figure 3.1 and have the error term (3.5) equal nonzero, while ROK4p does not satisfy this additional condition. Rosenbrock-W methods are preferred when finite-differences are the only option for obtaining Jacobian-vector products, and these products cannot be obtained accurately.

With the small basis size requirements in mind we explore in Figure 5.8 the

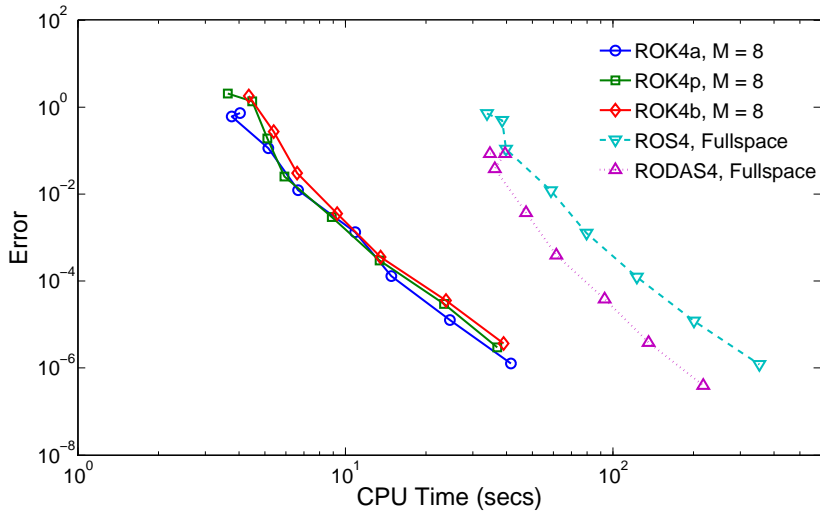


FIG. 5.7. Precision diagram for the shallow water equations.

relative difference in cost, measured by the number of right hand side evaluations, between an explicit Runge-Kutta method and matrix-free Rosenbrock-K methods. Figure 5.8 shows the number of function evaluations on the  $x$ -axis and the Error of the resulting solution on the  $y$ -axis. The number of function evaluations for ROK4a includes those required to compute the matrix-free Jacobian-vector products in the Arnoldi iteration. For the shallow water equations we see that Rosenbrock-K methods perform well against the explicit Runge-Kutta method, when low accuracy is desired and the CFL condition begins to constrain the explicit method.

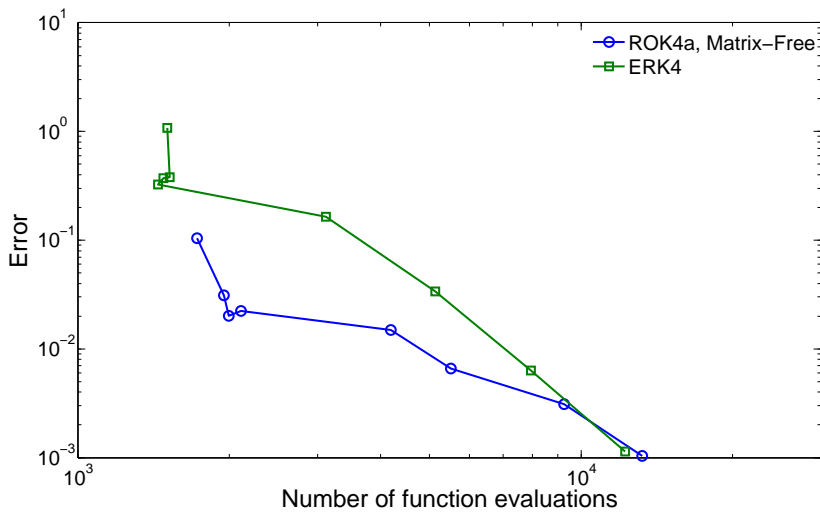


FIG. 5.8. Comparison of ROKa and ERK4 on the shallow water equations.

**6. Conclusions and future work.** In this work we have developed a new class of Rosenbrock like integrators, along with a corresponding order condition theory. We consider the ODE integrator and linear solver as a single computational process to develop methods with the least possible amount of implicitness.

The Rosenbrock-K order conditions remove the requirement for accurate solution of the linear systems which constrain the use of approximate methods in classical Rosenbrock integrators. For accuracy of the integration process, the size of the Krylov approximation of the Jacobian need be only as large as the desired order of the method. Stability considerations give stricter requirements on the size of the Krylov basis used, though the exact nature of these requirements is not yet entirely understood and will be the focus of future work. Some numerical investigation, and a result by Wensch [47], give reason to believe that the required size of the Krylov subspace is related to the number of stiff variables in the underlying problem.

The Rosenbrock-K methods developed here have many favorable properties over similar integrators. Rosenbrock-K methods have substantially fewer order conditions than Rosenbrock-W methods, requiring only a single extra order condition for order four methods as opposed to four extra conditions for order three methods in the case of Rosenbrock-W. The reduced number of order conditions allows for methods of higher order, we have given conditions up to order five, or for methods with fewer stages. Further, the structure of Rosenbrock-K methods allows for the computation of a single Krylov subspace at each timestep without the requirement of enriching this space for each internal stage, as is the case for Krylov-ROW methods.

The efficiency of Rosenbrock-K integrators applied to a specific problem is dependent on the stability requirements, and therefore on the stiffness of the underlying problem. We have illustrated this in [43] with the help of a chemical kinetic test problem. For this reason Rosenbrock-K methods are best suited to very large problems in which there is a relatively small number of stiff variables. However, Rosenbrock-K methods are expected to perform at least as well as Krylov-ROW methods in all cases.

**Acknowledgements.** This work has been supported in part by NSF through awards NSF OCI-8670904397, NSF CCF-0916493, NSF DMS-0915047, NSF CMMI-1130667, NSF CCF-1218454, AFOSR FA9550-12-1-0293-DEF, AFOSR 12-2640-06, and by the Computational Science Laboratory at Virginia Tech.

#### REFERENCES

- [1] M. BOTCHEV, Block Krylov subspace exact time integration of linear ODE systems. part 1: algorithm description. <http://arxiv.org/abs/1109.5100>, 2011.
- [2] M. BOTCHEV, G. SLEIJPEN, AND H. VAN DER VORS, Low-dimensional Krylov subspace iterations for enhancing stability of time-step integration schemes, tech. rep., University Utrecht, 2001.
- [3] A. BOUHAMIDI AND K. JBILOU, Stein implicit Runge–Kutta methods with high stage order for large-scale ordinary differential equations, *Applied Numerical Mathematics*, 61 (2011), pp. 149–159.
- [4] P. BROWN, H. WALKER, R. WASYK, AND C. WOODWARD, On using approximate finite differences in matrix-free Newton-Krylov methods, *SIAM Journal on Numerical Analysis*, 46 (2008), pp. 1892–1911.
- [5] P. N. BROWN, A. C. HINDMARSH, AND L. R. PETZOLD, Using Krylov methods in the solution of large-scale differential-algebraic systems, *SIAM Journal on Scientific Computing*, 15 (1994), pp. 1467–1488.
- [6] M. CALIARIA AND A. OSTERMANN, Implementation of exponential Rosenbrock-type integrators, *Applied Numerical Mathematics*, 3–4 (2009), pp. 568–581.
- [7] K. DEKKER, Partitioned Krylov subspace iteration in implicit Runge–Kutta methods, *Linear Algebra and its Applications*, 431 (2009), pp. 488–494.

- [8] K. EVANS AND D. KNOLL, Temporal accuracy analysis of phase change convection simulations using the JFNK-SIMPLE algorithm, *International Journal for Numerical Methods in Fluids*, 55 (2007).
- [9] M. W. GERY, G. Z. WHITTEN, J. P. KILLUS, AND M. C. DODGE, A photochemical kinetics mechanism for urban and regional scale computer modeling, *Journal of Geophysical Research: Atmospheres*, 94 (1989), pp. 12925–12956.
- [10] A. GRIEWANK, Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, no. 19 in *Frontiers in Appl. Math.*, SIAM, Philadelphia, PA, 2000.
- [11] I. GROTOWSKYA AND J. BALLMANNA, Efficient time integration of Navier–Stokes equations, *Computers & Fluids*, 28 (1996), pp. 243–263.
- [12] E. HAIRER, S. NORSETT, AND G. WANNER, Solving Ordinary Differential Equations I: Nonstiff Problems, Springer, 2008.
- [13] E. HAIRER AND G. WANNER, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer, 2002.
- [14] J. S. HESTHAVEN AND T. WARBURTON, Nodal Discontinuous Galerkin Methods, *Texts in Applied Mathematics*, Springer, 2008.
- [15] A. HINDMARSH, LSODKR, stiff ordinary differential equations (ODE) system solver with Krylov iteration with rootfinding, 1991.
- [16] A. HINDMARSH AND P. BROWN, LSODPK, ordinary differential equations solver for stiff and nonstiff system with Krylov corrector iteration, 1991.
- [17] A. HINDMARSH, P. BROWN, AND G. BYRNE, VODPK: Variable-coefficient ordinary differential equation solver with the Preconditioned Krylov method GMRES for the solution of linear systems. <http://www.netlib.org/ode/vodpk.f>, 2002.
- [18] M. HOCHBRUCK AND C. LUBICH, On Krylov subspace approximations to the matrix exponential operator, *SIAM Journal on Numerical Analysis*, 34 (1997), pp. 1911–1925.
- [19] M. HOCHBRUCK, C. LUBICH, AND H. SELHOFER, Exponential integrators for large systems of differential equations, *SIAM Journal of Scientific Computing*, 19 (1998), pp. 1552–1574.
- [20] M. HOCHBRUCK AND A. OSTERMANN, Exponential integrators, *Acta Numerica*, 19 (2012), pp. 209–286.
- [21] M. HOCHBRUCK, A. OSTERMANN, AND J. SCHWEITZER, Exponential Rosenbrock-type methods, *SIAM Journal on Numerical Analysis*, 47 (2009), pp. 786–803.
- [22] S. HOSSEINI AND G. HOJJATI, Matrix free MEBDF method for the solution of stiff systems of ODEs, *Mathematical and Computer Modelling*, 29 (1999), pp. 67–77.
- [23] J. HUANG, J. JIA, AND M. MINION, Accelerating the convergence of spectral deferred correction methods, *Journal of Computational Physics*, 214 (2006), pp. 633–656.
- [24] J. HUANG, J. JIA, AND M. MINION, Arbitrary order Krylov deferred correction methods for differential algebraic equations, *Journal of Computational Physics*, 221 (2007), pp. 739–760.
- [25] J. HUANG, J. JIA, AND M. MINION, Arbitrary order Krylov deferred correction methods for differential algebraic equations, *Journal of Computational Physics*, 221 (2007), pp. 739–760.
- [26] L. JAY, A parallelizable preconditioner for the iterative solution of implicit Runge–Kutta-type methods, *Journal of Computational and Applied Mathematics*, 111 (1999), pp. 63–76.
- [27] C. KELLEY, I. KEVREKIDIS, AND L. QIAO, Newton-Krylov solvers for time-steppers. [arXiv:math/0404374v1](https://arxiv.org/abs/math/0404374v1), 2004.
- [28] D. KNOLL AND D. KEYES, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, *Journal of Computational Physics*, 193 (2004), pp. 357–397.
- [29] R. LISKA AND B. WENDROFF, Composite schemes for conservation laws, 1997.
- [30] E. N. LORENZ, Predictability – a problem partly solved, in *Predictability of Weather and Climate*, Cambridge University Press, 2006.
- [31] C. LUBICH AND A. OSTERMANN, Linearly implicit time discretization of nonlinear parabolic equations, *IMA Journal on Numerical Mathematics*, 15 (1995), pp. 555–583.
- [32] P. NOVATI, Some secant approximations for Rosenbrock W-methods, *Applied Numerical Mathematics*, 58 (2008), pp. 195–211.
- [33] H. PODHAISKY, R. WEINER, AND B. SCHMITT, Numerical experiments with Krylov integrators, *Applied Numerical Mathematics*, 28 (1997), pp. 413–425.
- [34] J. RANG AND L. ANGERMANN, New Rosenbrock W-methods of order 3 for partial differential algebraic equations of index 1, *BIT Numerical Mathematics*, 45 (2005), pp. 761–787.
- [35] R. WEINER, B. SCHMITT, AND H. PODHAISKY, ROWMAP—a ROW-code with Krylov techniques for large stiff ODEs, *Applied Numerical Mathematics*, 25 (1997), pp. 303–319.
- [36] Y. SAAD, Iterative methods for sparse linear systems, PWS Pub. Co., Boston, 1996.
- [37] A. SANDU, J. VERWER, M. V. LOON, G. CARMICHAEL, F. POTRA, D. DABDUB, AND J. SE-

- INFELD, Benchmarking stiff ode solvers for atmospheric chemistry problems-i. implicit vs explicit, Atmospheric Environment, 31 (1997), pp. 3151 – 3166. ic:title;EUMAC: European Modelling of Atmospheric Constituents.
- [38] B. SCHMITT AND R. WEINER, Matrix free W-methods using a multiple Arnoldi iteration, Applied Numerical Mathematics, 18 (1995), pp. 307–320.
- [39] ———, Polynomial preconditioning in Krylov-ROW-methods, Applied Numerical Mathematics, 28 (1998), pp. 427–437.
- [40] J. C. SCHULZE, P. J. SCHMID, AND J. L. SESTERHENN, Exponential time integration using Krylov subspaces, International Journal for Numerical Methods in Fluids, 60 (2008), pp. 561–609.
- [41] K. STREHMEL, R. WEINER, AND M. BÜTTNER, Order results for Rosenbrock type methods on classes of stiff equations, Numerische Mathematik, 59 (1991), pp. 723–737.
- [42] M. TOKMAN, Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods, Journal of Computational Physics, 213 (2006), pp. 748–776.
- [43] P. TRANQUILLI AND A. SANDU, Rosenbrock-Krylov methods for large systems of differential equations, ArXiv e-prints, (2013).
- [44] H. A. VAN DER VORST, Iterative Methods for Large Linear Systems, Cambridge University Press, 2003.
- [45] R. WEINER AND B. SCHMITT, Consistency of Krylov-W-methods in initial value problems, Tech. Rep. 14, Fachbereich Mathematik und Informatik, Martin-Luther-Universität Halle-Wittenberg, 1995.
- [46] ———, Order results for Krylov-W methods, Computing, 61 (1998), pp. 69–89.
- [47] J. WENSCH, Krylov-ROW methods for DAEs of index 1 with applications to viscoelasticity, Applied Numerical Mathematics, 53 (2005), pp. 527–541.