Numerical experiments with Krylov integrators

H.Podhaisky, R. Weiner and B.A. Schmitt

Abstract

We discuss the use of preconditoning in Krylov-W-methods. The preconditioning is based on different operator splitting schemes for reaction-diffusion problems. Comparison of various Krylov codes show that the preconditioned version of ROWMAP works efficient and reliable, especially for large dimensions. Furthermore, parallelism can easily be exploited in the preconditioning.

Key words. ROW-methods, stiff initial value problems, Krylov subspaces, preconditioning.

AMS(MOS) subject classification. 65L06, 65F10

1 Introduction

For the numerical solution of stiff initial value problems

$$\begin{aligned} y'(t) &= f(t, y(t)), \\ y(t_0) &= y_0 \in \mathbb{R}^n, \end{aligned} \tag{1.1}$$

implicit or linearly implicit methods have to be used due to stability requirements. For large dimensions n these methods spend most of their computing time for the evaluation of the Jacobian and in the solution of the linear equations. Recently, Krylov methods have been used extensively in implicit methods for stiff ordinary differential equations to accelerate the linear algebra in these integration schemes, e.g. [2, 3, 4, 15]. An alternative approach for large stiff equations are methods containing matrix exponentials as exact solutions of linearized equations. An ODE integrator based on Krylov approximations for such matrix functions multiplied by a vector is discussed in [7].

We will study the use of Krylov techniques in ROW- and W-methods. ROW-methods have shown to be very efficient solution methods for stiff problems. The code ROS4 of Hairer and Wanner [6] performs well for low and medium accuracy requirements.

For simplicity we restrict the following considerations to autonomous problems. A ROW-method for autonomous systems (1.1) is given by

$$(I - h\gamma A)\left(k_i + \sum_{j=1}^{i-1}\frac{\gamma_{ij}}{\gamma}k_j\right) = f_i + \sum_{j=1}^{i-1}\frac{\gamma_{ij}}{\gamma}k_j =: w_i, \tag{1.2}$$
$$f_i = f(u_{m+1}^{(i)}),$$
$$u_{m+1}^{(i)} = u_m + h\sum_{j=1}^{i-1} a_{ij}k_j,$$
$$u_{m+1} = u_m + h\sum_{i=1}^{s} b_i k_i,$$

where

$$A = f_y(u_m) \tag{1.3}$$

For the computation of the approximate solution u_{m+1} in (1.2) s linear systems

$$(I - h\gamma A)x_i = w_i, \quad x_i := k_i + \sum_{j=1}^{i-1}\frac{\gamma_{ij}}{\gamma}k_j$$

with the same coefficient matrix have to be solved. In Krylov–W-methods these equations are solved iteratively by Arnoldi's method. A corresponding code ROWMAP, which is based on the 4-stage ROW-methods of order 4 of ROS4, was extensively tested on large test-problems obtained by MOL-semidiscretization of 2- and 3-dimensional parabolic equations (cf. [15]). Comparisons with the multistep code VODPK [4] have shown the efficiency and reliability of ROWMAP.

The computational effort of the Arnoldi process increases for large dimensions n quadratically with the Krylov dimension. We are therefore interested in keeping these dimensions small. In this context the idea of preconditioning is attractive. In this paper we will consider preconditioning strategies, which exploit the structure of the given problem, especially for reaction-diffusion problems.

In section 2 we describe left preconditioning for Krylov–W-methods and discuss the choice of a preconditioner based on operator splitting for a class of reaction-diffusion equations.

In section 3 we present numerical results for the preconditioned ROWMAP applied to several large stiff problems obtained by MOL-discretization of parabolic equations. Comparisons with VODPK and EXP4 show the efficiency of the code.

2 Preconditioning

For the iterative solution of (1.2) we consider here the equivalent preconditioned systems

$$P^{-1}(I - h\gamma A)x_i = P^{-1}w_i, \quad i = 1, \ldots, s \tag{2.1}$$

with an arbitrary nonsingular matrix $P = P(u_m)$. We expect that the preconditioned problem can be solved more efficiently than the original problem provided the linear systems $Pz = c$ are easy to solve, and P is in some sense close to $(I - h\gamma A)$.

Denote $M := P^{-1}(I - h\gamma A)$ and $v_i := P^{-1}w_i$. We will use a special Multiple Arnoldi Process described in the following to construct sequences of orthonormal vectors $q_1, \ldots, q_{\kappa_i}$ that constitute a basis for the subspaces $\mathcal{K}_i = \text{span}\{q_1, \ldots, q_{\kappa_i}\}$. We determine an approximate solution $\tilde{x}_i$ for (2.1) by

(i) $\tilde{x}_i \in \mathcal{K}_i$

(ii) $res_i \perp \mathcal{K}_i, \quad res_i := M\tilde{x}_i - v_i.$

With the notation $Q_i = (q_1, \ldots, q_{\kappa_i})$, (i) implies that $\tilde{x}_i = Q_i l_i$ for a certain $l_i \in \mathbb{R}^{\kappa_i}$. From (ii) we get a linear system for l_i

$$H_i l_i = Q_i^T v_i, \quad H_i := Q_i^T M Q_i \tag{2.2}$$

which has dimension κ_i only.

We consider as basic integration method the 4-stage ROW-method GRK4T [9] of order 4 characterized by $\gamma = 0.231$, $b_3 = 0$, $f_4 = f_3$, which is $A(\alpha)$-stable with $\alpha = 89.3°$.

At the first stage we use the classical Arnoldi method [10] to construct the orthonormal basis $Q_1 = (q_1, \ldots, q_{\kappa_1})$ of the Krylov subspace

$$\mathcal{K}_1 = \mathcal{K}(M, v_1, \kappa_1) := \text{span}\{v_1, Mv_1, \ldots, M^{\kappa_1 - 1}v_1\}. \tag{2.3}$$

Of practical importance is that H_1 is in Hessenberg form and that the residual res_1 can be computed form the subspace data alone by

$$res_1 = M\tilde{x}_1 - v_1 = (I - Q_1 Q_1^T) M q_{\kappa_1} e_{\kappa_1}^T l_1$$

without knowing $k_1 = Q_1 l_1$.

We stop the Arnoldi process at the first stage if $\kappa_1 \geq 4$ and

$$||res_1||_{WRMS} := \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{res_{1,i}}{atol_i + rtol_i|u_{m,i}|}\right)^2} \leq \frac{\theta}{h}$$

holds or an upper bound (due to memory restrictions) $\kappa_1 = \kappa_1^{max}$ is reached. In the following computations we used $\theta = 0.1$. Here, $atol_i$ and $rtol_i$ are the components of an absolute and an relative tolerance vector.

Remark 2.1 Throughout our investigations we used left preconditioning. This is a natural choice, because we are not interested in the residuals, but in the errors and left preconditioning transforms the residuals in such manner, that their norms should be of the same magnitude as the errors

$$\begin{aligned} res_1 &= P^{-1}(I - h\gamma A)\tilde{x}_1 - P^{-1}v_1 \\ &\approx \tilde{x}_1 - x_1. \end{aligned}$$

This requires P^{-1} to be regular, which may not be the case in other schemes like polynomial preconditioning. $\square$

At higher stages, $i > 1$ we employ a special Multiple Arnoldi process (cf. [13, 15]) to re-use the information about M contained in the subspace $\mathcal{K}_1$. We use the Gram-Schmidt orthogonalization to construct the bases Q_i for the subspaces

$$
\begin{aligned}
\mathcal{K}_2 &= \mathcal{K}_1 \oplus \operatorname{span}\{v_2, Mq_{\kappa_1}, Mv_2\} \\
\mathcal{K}_3 &= \mathcal{K}_2 \oplus \operatorname{span}\{v_3\} \\
\mathcal{K}_4 &= \mathcal{K}_3 \oplus \operatorname{span}\{M^2 q_{\kappa_1}, M^2 v_2, Mv_3\}.
\end{aligned}
\tag{2.4}
$$

With these subspaces (2.3), (2.4) the Krylov–W-method has order 4 like the underlying ROW-method.

Theorem 2.1 *The method ROWMAP with preconditioning and MAP (2.3), (2.4) based on GRK4T is of order 4.*

Proof. Analogous to the proof of Theorem 2.1 in [15]. $\blacksquare$

Remark 2.2 1. To summarize, the first dimension κ_1 is at least 4 and determined by accuracy requirements for the residual res_1, but is restricted by κ_1^{max}. The other dimensions grow by at most $\kappa_2 = \kappa_1 + 3$, $\kappa_3 = \kappa_2 + 1$, $\kappa_4 = \kappa_3 + 3$.

2. The Krylov–W-method can be adaped to nonautonomous systems (cf. [15]). With slightly larger Krylov dimensions

$$
\kappa_1 \geq 4, \quad \kappa_2 = \kappa_1 + 5, \quad \kappa_3 = \kappa_2 + 1, \quad \kappa_4 = \kappa_3 + 5
$$

the order of consistency remains 4. $\square$

We will now study the choice of the preconditioner P for ODE systems arising in MOL-treatment for a class of reaction-diffusion problems. We consider systems of PDEs in d spatial dimensions in the domain $\Omega = [\alpha_1, \beta_1] \times, \cdots \times [\alpha_d, \beta_d]$ of the type

$$
\frac{\partial u^j}{\partial t} = \sum_{i=1}^{d} f_i^j(u^j, u_{x_i}^j, u_{x_i x_i}^j, x) + g^j(u, x),
\tag{2.5}
$$

$$
j = 1, \ldots, r, \qquad u = (u^1, \ldots, u^r) \in \mathbb{R}^r, \qquad x = (x_1, \ldots, x_d)
$$

with initial and boundary conditions. By introducing a $m_1 \times \cdots \times m_d$ grid and substituting the derivatives in the f_i by finite differences we get an ODE system

$$
\frac{dU}{dt} = \sum_{i=1}^{d} F_i(U) + G(U), \quad U \in \mathbb{R}^n, \quad n = r \prod_{i=1}^{d} m_i.
\tag{2.6}
$$

Here, U contains the solution $u^1, \ldots, u^r$ in all grid points.

For preconditioning we need an approximation to the Jacobian of the system (2.6). We can take into consideration

(a) the part of the Jacobian due to $F := \sum_{i=1}^{d} F_i$,

(b) the part from G,

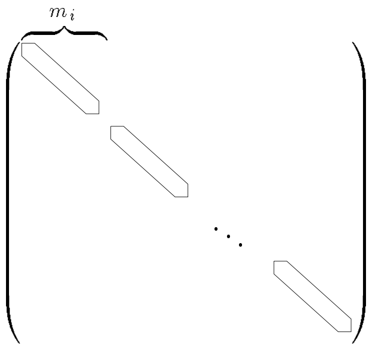
(c) both.

In case (a) we assume for simplicity that standard first and second order differences have been used for semidiscretization. We can split the Jacobian of F with respect to the d spatial dimensions. Therefore, we have

$$\frac{\partial F(U)}{\partial U} = D_1 + \cdots + D_d, \quad D_i := \frac{\partial F_i}{\partial U},$$

where the matrices D_i can be transformed to tridiagonal matrices $\tilde{D}_i$, i.e. there exist permutations $S_i \in \mathbb{R}^{n,n}$ with

(2.7) $$S_i D_i S_i^{-1} = \tilde{D}_i =$$



.

The preconditioner P in (2.1) can be choosen as proposed by van der Houwen et.al. [14]

$$P = P_D := (I - h\gamma D_d) \cdots (I - h\gamma D_1)$$

We see from (2.7) that the solution of the systems

$$(I - h\gamma \tilde{D}_i) z = c \tag{2.8}$$

can be computed in blocks of size m_i independently. We have therefore a great potential for parallel solution in the case of preconditioning. In our numerical tests we have tested this by solving (2.8) in parallel with 4 processors giving a speedup of about 2.

If the stiffness of the problem (2.6) is mainly due to the reaction terms given by G we can use for preconditioning

$$P = P_R := (I - h\gamma R), \quad R := \frac{\partial G}{\partial U}.$$

With an appropriate ordering of the components in U, the matrix P_R has block-diagonal structure

$$P_R = diag\{R_1, \ldots, R_{n/r}\}, \quad R_i \in \mathbb{R}^{r,r}.$$

In (c) we use the splitting

$$A = \frac{\partial F}{\partial U} + \frac{\partial G}{\partial U}$$

and set

$$P = P_{RD} := P_R P_D = (I - h\gamma R)(I - h\gamma D_d)\cdots(I - h\gamma D_1).$$

3 Numerical Experiments

We have added the possibility of preconditioning to our code ROWMAP with the Multiple Arnoldi Process in `FORTRAN 77`. In the new version the user has the possibility to supply two subroutines which calculate the preconditioning matrix P and solve the linear systems $Px = c$. The code can be obtained from http://www.mathematik.uni-halle.de/institute/numerik/software. In this code the preconditioning has to be done by the user as described in the preceding section. A version with polynomial preconditioning, which is done by the code automatically, is discussed in [12].

We compare ROWMAP with the code VODPK [4] (version of May 1994) which is based on the BDF-code VODE [1] and uses GMRES(m) [11] for the solution of the linear systems occurring in the Newton iteration and with the code EXP4 [7] (version of March 1997) based on a Krylov subspace method calculating a function related to the exponential of the Jacobian multiplied by a vector.

We present results for the following 3 parabolic problems which are semidiscretized by the method-of-lines with central differences with uniform spatial discretization in x and y leading to first order systems of ordinary differential equations. The actual dimensions n are given in the diagrams below.

- HEAT: The simple 2-dimensional heat equation [4]

$$\begin{aligned} u_t &= u_{xx} + u_{yy}, \\ u(0,x,y) &= 16x(1-x)(1-y), \\ u(t,x,y) &= 0 \quad \text{on} \quad \partial\Omega, \qquad \Omega = [0,1]\times[0,1], \quad t \in [0, 10.24] \end{aligned}$$

  For preconditioning we use $P = P_D$.

- NILIDI: The nonlinear diffusion equation [13]

$$u_t = e^u(u_{xx} + u_{yy}) + u(18e^u - 1),$$

  on $[0,\pi/3]\times[0,\pi/3]$, $\quad t \in [0,1]$.
  Initial and boundary values are taken from the exact solution: $u = e^{-t}\sin(3x)\sin(3y)$.
  The preconditioner $P = P_D$ is based on the splitting

$$u_t = \left(e^u u_{xx} + \frac{u}{2}(18e^u - 1)\right) + \left(e^u u_{yy} + \frac{u}{2}(18e^u - 1)\right) = f_1 + f_2.$$

- VPOL: The van der Pol equation with diffusion [8]

$$
\begin{aligned}
u_t &= u_{xx} + u_{yy} + v \\
v_t &= 0.05(v_{xx} + v_{yy}) + 100(1-u^2)v - u \\
u(0,x,y) &= 1 - \cos(\pi x)\cos(2\pi y) \\
v(0,x,y) &= 1 + 2\cos(2\pi x)\cos(\pi y) \\
\frac{\partial u}{\partial n} = \frac{\partial v}{\partial n} &= 0 \quad \text{on} \quad \partial\Omega, \quad \Omega = [0,1]\times[0,1], \quad t \in [0,1000]
\end{aligned}
$$

  For preconditioning we use $P = P_{RD}$ corresponding to the splitting $(u_t, v_t)^T = f_1 + f_2 + g$ with

$$
\begin{aligned}
f_1 &= (f_1^1, f_1^2)^T = (u_{xx}, 0.05 v_{xx})^T \\
f_2 &= (f_2^1, f_2^2)^T = (u_{yy}, 0.05 v_{yy})^T \\
g &= (g^1, g^2)^T = (v, 100(1-u^2)v - u)^T.
\end{aligned}
$$

The computations were performed on a HP-Convex Exemplar S-Class-Server. This machine has 16 PA-8000 processors with 180 MHz and 3 GB global shared memory. The codes are written for sequential computation. So, our experiments were run on one processor only, the investigation of parallel Krylov methods will be a topic of further work. However, for comparison we have already parallelized the preconditioning as described in section 2. Therefore we will include the results with four processors here.

ROWMAP and VODPK are `FORTRAN 77` codes. EXP4 is written in C but uses Fortran subroutines for the linear algebra. In our tests EXP4 uses the same Fortran subroutines for the right hand sides as the other codes. All three codes use the highly optimized BLAS-library for this machine. For solving the linear systems for the preconditioner we employed an optimized LAPACK [5] implementation.

For all options and parameters in the codes we used the default values except for the parameter `maxl` in VODPK. In the multistep code the Krylov method GMRES(`maxl`) is used to solve the linear equations arising in the Newton method. When the accuracy is not sufficient after `maxl` steps a restart of GMRES is performed. The default value is `maxl`=5. For our test problems `maxl`=10 usually has given better results than `maxl`=5.

All problems were solved for the seven tolerances $tol = 10^{-1}, \ldots, 10^{-7}$ with $atol = rtol = tol$. For details of error estimation and stepsize control we refer to [15]. The marks in the figures show the computing time (in seconds, logarithmic scale) plotted against the logarithm of the relative accuracy ERR achieved at the endpoint, where

$$
ERR = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{u_i - uex_i}{1 + |uex_i|}\right)^2}
$$

The different marks in each series of tolerances are connected by lines. Marks for cruder tolerances may be missing if the code fails there. The reference solution uex for the semidiscretized system was set equal zero for HEAT and was computed with VODPK and $tol = 10^{-12}$ for VPOL, $n = 800$, $n = 20000$ and NILIDI, $n = 10000$ and with the preconditioned ROWMAP, $tol = 10^{-8}$ for NILIDI, $n = 99856$.

In the computations corresponding to the Figures 1-6 all Jacobian-vector products were approximated by finite differences

$$Av = f_y(u_m)v = \frac{1}{\delta}\left(f(u_m + \delta v) - f(u_m)\right) + \mathcal{O}(\delta||v||^2).$$

So, the exact computation of the Jacobian is not necessary ("matrix free methods").

In the figures we use the following notations:

ROWMAP	—	ROWMAP without preconditioning
ROWMAP-PC	—	ROWMAP with preconditioning
ROWMAP-PAR	—	ROWMAP with parallelized preconditioning (four processors)
ROWMAP-J	—	ROWMAP with exact Av subroutine
VODPK	—	VODPK without preconditioning
VODPK-PC	—	VODPK with preconditioning
VODPK-PAR	—	VODPK with parallelized preconditioning (four processors)
EXP4	—	EXP4
EXP4-J	—	EXP4 with exact Av subroutine

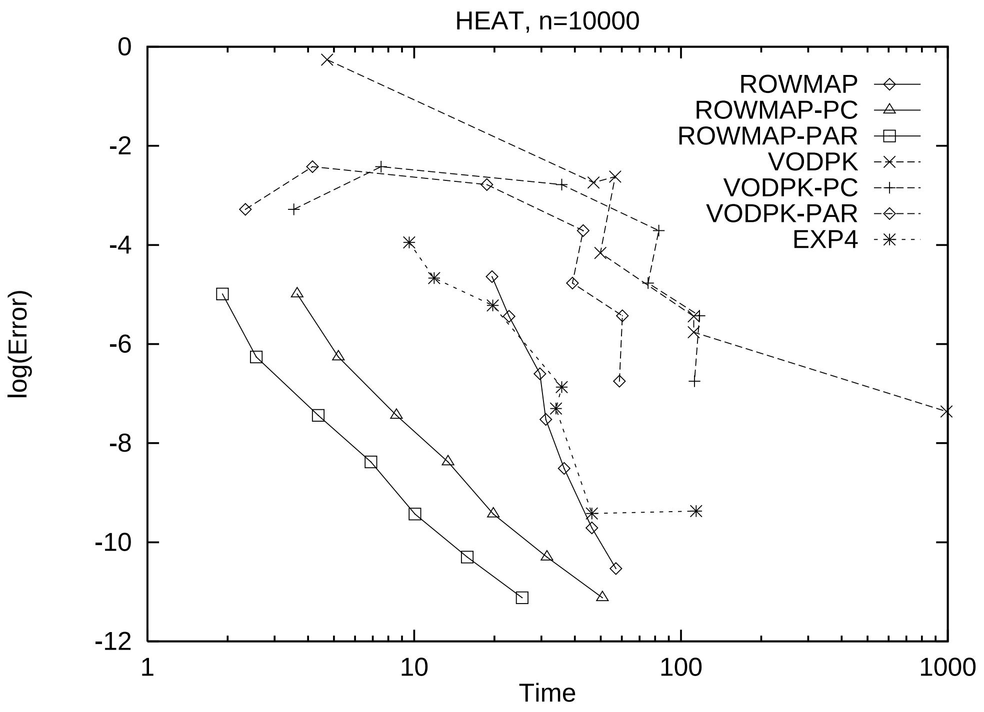


Figure 1: Work-precision diagram for HEAT, $n = 10000$.

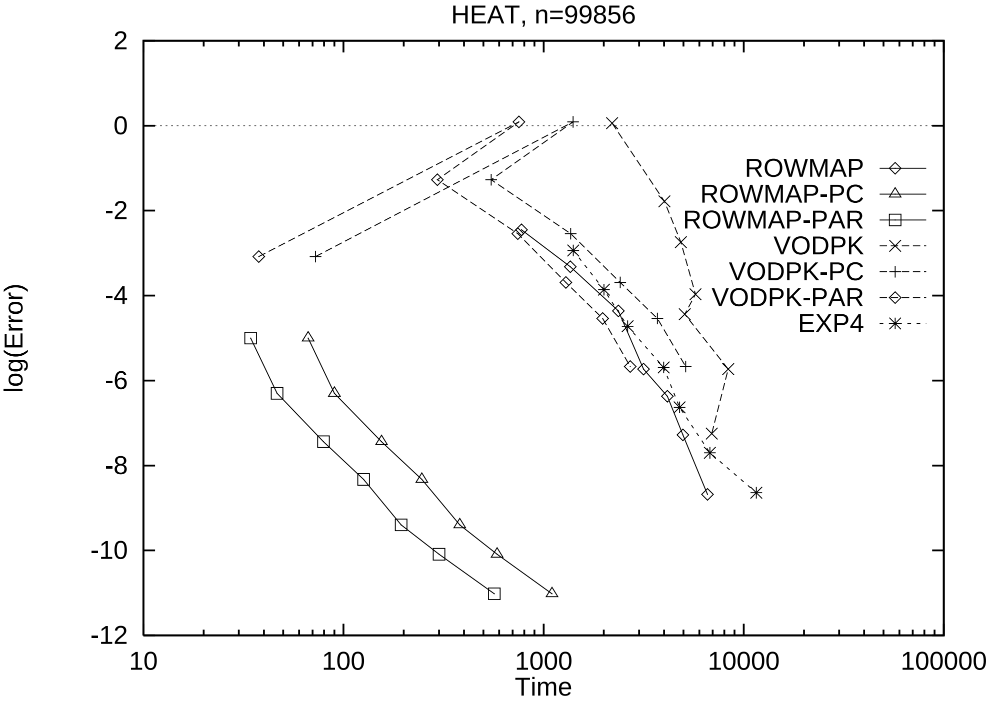


Figure 2: Work-precision diagram for HEAT, $n = 99856$.

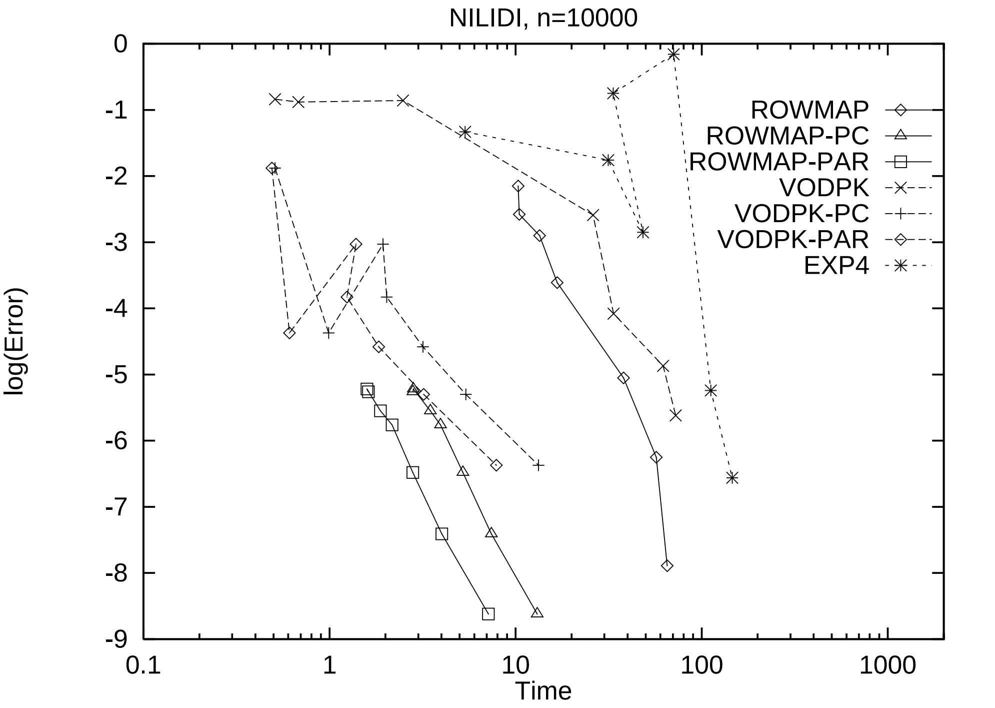


Figure 3: Work-precision diagram for NILIDI, $n = 10000$.

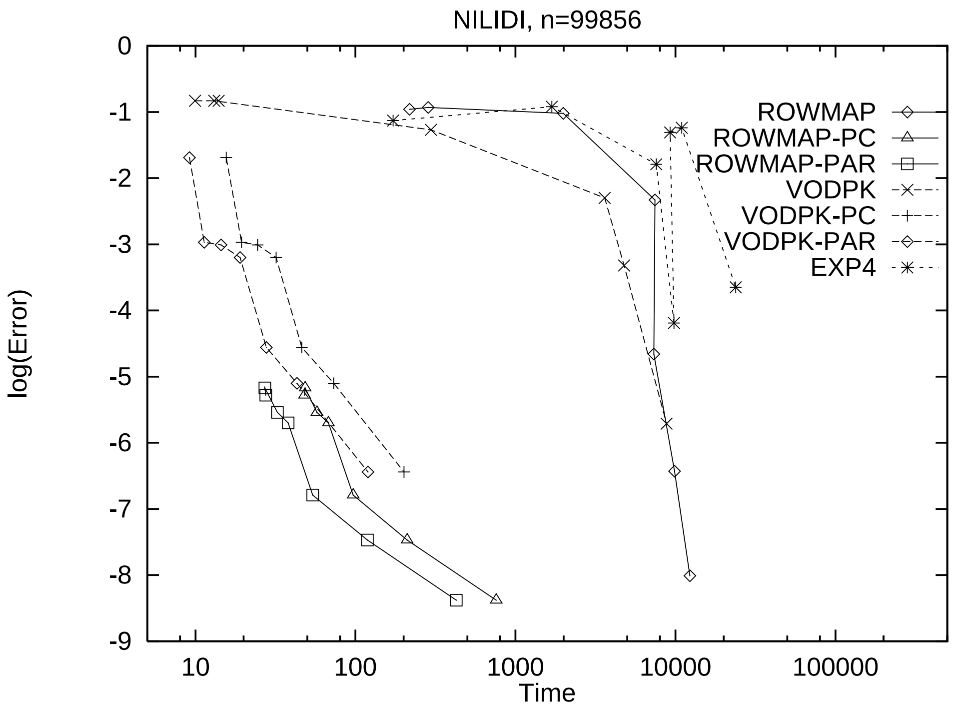


Figure 4: Work-precision diagram for NILIDI, $n = 99856$.

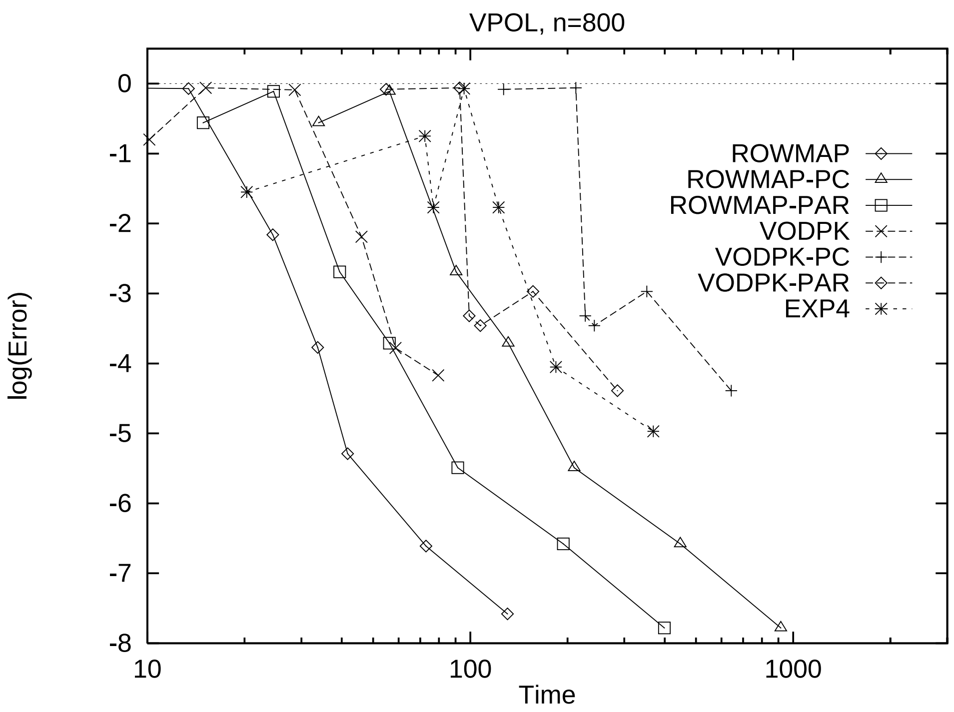


Figure 5: Work-precision diagram for VPOL, $n = 800$.

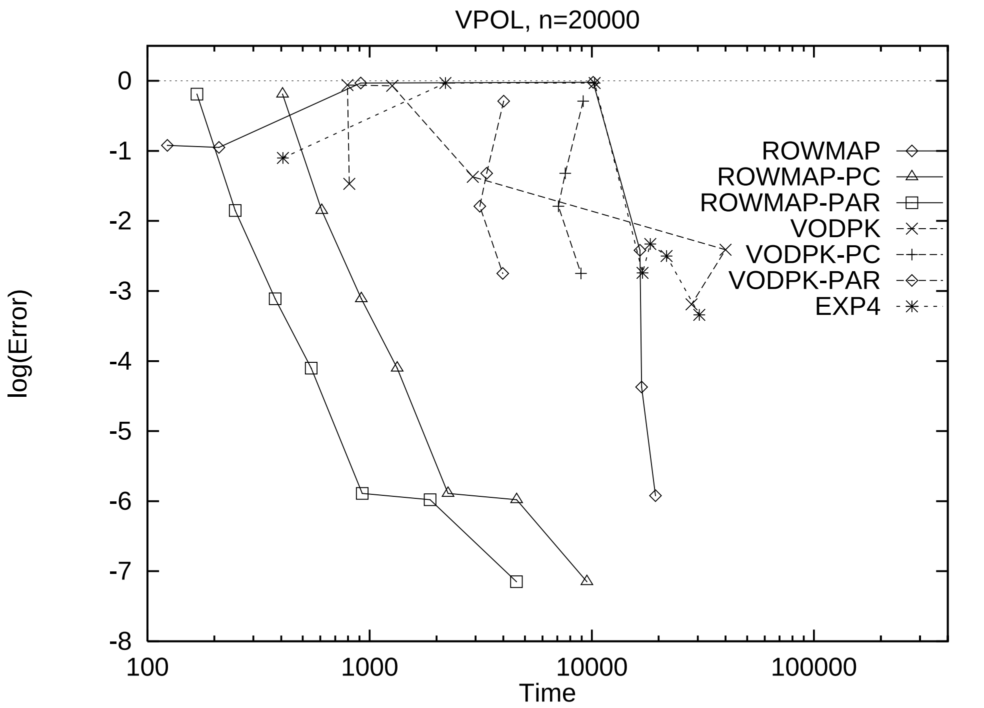


Figure 6: Work-precision diagram for VPOL, $n = 20000$.

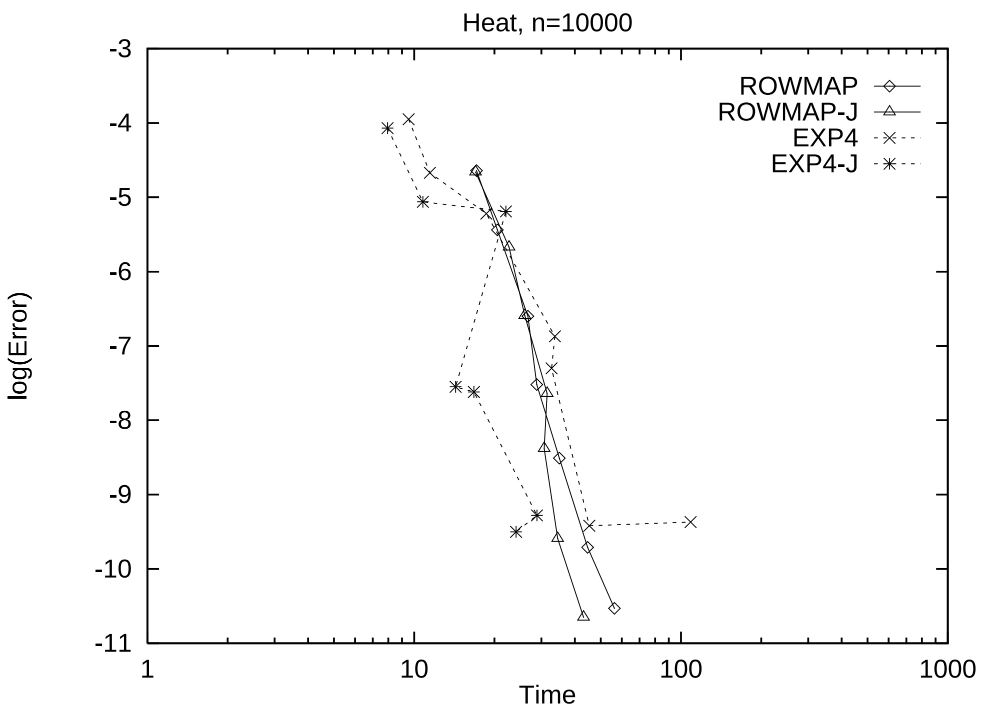


Figure 7: HEAT with different computations of Av.

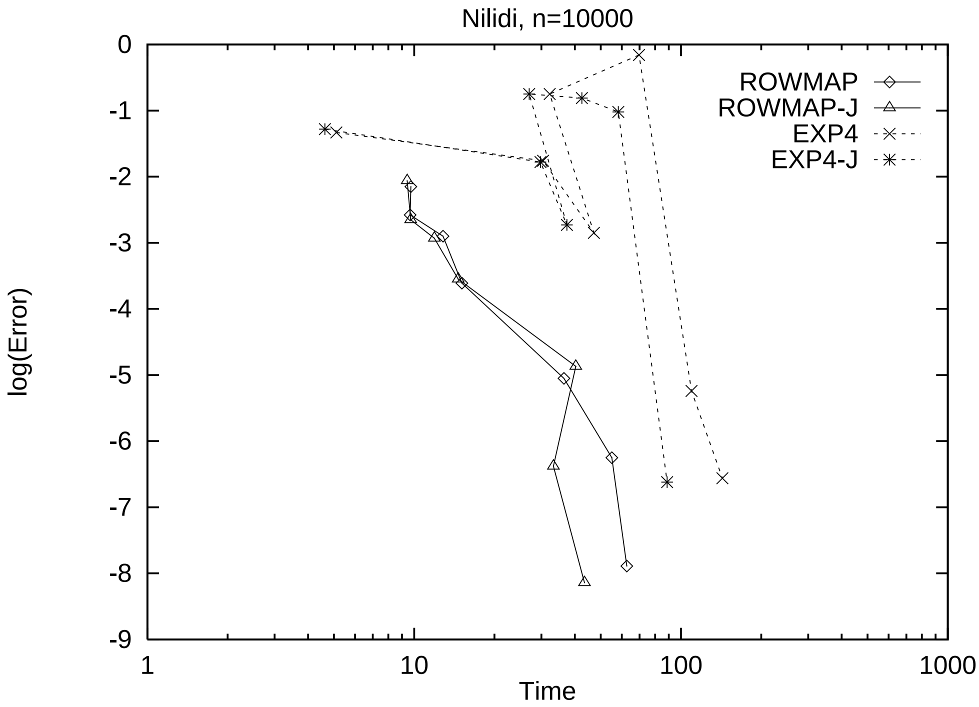


Figure 8: NILIDI with different computations of Av.

For Heat for the coarse grid the performance of the multistep code VODPK is much worse than that of the onestep codes. Unlike ROWMAP it does not profit from preconditioning. If we use the finer grid we observe, as in the other examples too, that the preconditioned methods become more efficient compared with the unpreconditioned.

For Nilidi ROWMAP and VODPK are clearly superior to EXP4, which shows a highly nonregular behaviour. Preconditioning redudes the computing times drastically. This is due to the large Krylov dimensions needed for this problem in the basic codes.

For the Van der Pol equation ROWMAP is the fastest and by far the most accurate code. For the small problem ROWMAP uses only small Krylov dimensions. The code without preconditioning is therefore faster than with preconditioning, the same is true for VODPK. This changes for the high-dimensional problem. The preconditioned codes are not only faster, but we also observe that the codes without preconditioning are very inaccurate for crude tolerances, for stringent tolerances only ROWMAP becomes sufficiently accurate. The preconditoned ROWMAP performs excellent.

At all problems the parallel solution of the linear systems for the preconditioner gives a speedup of ca. 2, for both VODPK and ROWMAP.

The last two figures show the influence of the approximation of the products Av by finite differences. Both ROWMAP and EXP4 allow the user to supply a subroutine computing the exact product Av. We see that for the linear problem HEAT the code EXP4 gains more from the exact Av-computation than ROWMAP. But in the nonlinear example NILIDI this does not remain true, both codes profit in the same magnitude from the exact Jacobian for stringent tolerances.

4 Conclusion

The presented numerical results and many other computations performed for semidiscretized 2- and 3-dimensional parabolic equations lead to the following conclusions:

- Krylov techniques are very useful in the numerical solution of large stiff equations.
- For large dimensions preconditioning, which exploits the problem structure, can reduce the computing time strongly. However, such a preconditioning requires additional effort by the user, especially for systems of parabolic equations the implementation is complicated. Our tests show that in these cases ROWMAP without preconditioning works also reliable.
- When using preconditioning there is a large potential of parallelism that can be exploited very easily.

References

[1] P.N. Brown, G.D. Byrne, and A.C. Hindmarsh, *VODE: a variable-coefficient ODE solver*, SIAM J. Sci. Statist. Comput. **10** (1989), 1038–1051.

[2] P.N. Brown and A.C. Hindmarsh, *Matrix-free methods for stiff systems of ODE's*, SIAM J. Numer. Anal. **23** (1986), 610–638.

[3] ______, *Reduced storage matrix methods in stiff ODE systems*, J. Appl. Math. Comp. **31** (1989), 40–91.

[4] G.D. Byrne, *Pragmatic experiments with Krylov methods in the stiff ODE setting*, Computational Ordinary Differential Equations (Clarendon Press, Oxford) (J.R. Cash and I. Gladwell, eds.), 1992, pp. 323–356.

[5] E. Anderson et al., *LAPACK users' guide*, SIAM, Philadelphia, 1995.

[6] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, Springer, 1996.

[7] M. Hochbruck, C. Lubich, and H. Selhofer, *Exponential integrators for large systems of differential equations*, Tech. report, Mathematisches Institut Universität Tübingen, 1995, Submitted to SIAM J. Sci. Comp.

[8] K.R. Jackson and W.L. Seward, *Adaptive linear equation solvers in codes for large stiff systems of ODEs*, SIAM J. Sci. Comput. **14** (1993), 800–823.

[9] P. Kaps and P. Rentrop, *Generalized Runge-Kutta methods of order four with stepsize control for stiff ordinary differential equations*, Numer. Math. **38** (1979), 55–68.

[10] Y. Saad, *Krylov subspace methods for solving large unsymmetric linear systems*, Math. Comp. **37** (1981), 105–126.

[11] Y. Saad and M.H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput. **7** (1986), 856–869.

[12] B.A. Schmitt and R. Weiner, *Polynomial preconditioning in Krylov-ROW-methods*, submitted for publication.

[13] ———, *Matrix-free W-methods using a multiple Arnoldi iteration*, Appl. Numer. Math. **18** (1995), 307–320.

[14] P.J. van der Houwen, B.P. Sommeijer, and J. Kok, *The iterative solution of fully implicit discretization of three-dimensional transport models*, Tech. Report 21, CWI Amsterdam, 1996.

[15] R. Weiner, B.A. Schmitt, and H. Podhaisky, *ROWMAP – a ROW-code with Krylov techniques for large stiff ODEs*, Appl. Numer. Math. **25** (1997).

Helmut Podhaisky, Rüdiger Weiner
FB Mathematik und Informatik
Martin-Luther-Universität
Halle-Wittenberg
Postfach
D-06099 Halle/Saale
Germany
e-mail: rw@mail.mathematik.uni-halle.de

Bernhard A. Schmitt
Fachbereich Mathematik
Universität Marburg
Lahnberge
D-35032 Marburg
Germany
e-mail: schmitt@mathematik.uni-marburg.de