PRECONDITIONING AND PARALLEL IMPLEMENTATION OF IMPLICIT RUNGE-KUTTA METHODS.

LAURENT O. JAY *

Abstract. A major problem in obtaining an efficient implementation of fully implicit Runge-Kutta (IRK) methods applied to systems of differential equations is to solve the underlying systems of nonlinear equations. Their solution is usually obtained by application of modified Newton iterations with an approximate Jacobian matrix. The systems of linear equations of the modified Newton method can actually be solved approximately with a preconditioned linear iterative method. In this article we present a truly parallelizable preconditioner to the approximate Jacobian matrix. Its decomposition cost for a sequential or parallel implementation can be made equivalent to the cost corresponding to the implicit Euler method. The application of the preconditioner to a vector consists of three steps: two steps involve the solution of a linear system with the same block-diagonal matrix and one step involves a matrix-vector product. The preconditioner is asymptotically correct for the Dahlquist test equation. Some free parameters of the preconditioner can be determined in order to optimize certain properties of the preconditioned approximate Jacobian matrix.

Key words. GMRES, implicit Runge-Kutta methods, inexact modified Newton iterations, linear iterative methods, nonlinear equations, ordinary differential equations, parallelism, preconditioning, stiffness

AMS subject classifications. 65F10, 65H10, 65L05, 65L06, 65Y05

1. Introduction.

We consider the numerical solution of implicit systems of differential equations by fully implicit Runge-Kutta (IRK) methods. A major problem in obtaining an efficient implementation of IRK methods lies in the numerical solution of the underlying systems of nonlinear equations. These nonlinear equations are usually solved by modified Newton iterations. This requires at each iteration the solution of a system of linear equations with an approximate Jacobian matrix. Instead of being solved exactly, these systems of linear equations can actually be solved approximately and iteratively with the help of a preconditioner to the approximate Jacobian matrix. In this paper we present a new preconditioner whose decomposition cost for a sequential or parallel implementation can be made equivalent to the cost corresponding to the implicit Euler method. Therefore, this preconditioner is of interest for both sequential and parallel computers. The application of the preconditioner to a vector consists of three steps: two involving the solution of a linear system with the same block-diagonal matrix and one involving a matrix-vector product. Each of these steps can be executed in parallel with some communication in-between. Not only the decomposition of the block-diagonal matrix can be executed in parallel, but of course also the solution of the linear systems corresponding to each block. This makes this preconditioner truly parallel compared to the one proposed in [19, 21]. The preconditioner is asymptotically correct for the Dahlquist test equation. Moreover, some free parameters of the preconditioner can be determined in order to optimize certain properties of the preconditioned approximate Jacobian matrix in relation, for example, with the convergence properties of the preconditioned linear iterative method.

In section 2, the class of implicit systems of ODEs considered in this article is presented together with a definition of the application of IRK methods to these equations.

* Department of Mathematics, 14 MacLean Hall, The University of Iowa, Iowa City, IA 52242-1419, USA. E-mail: `ljay@math.uiowa.edu` and `na.ljay@na-net.ornl.gov`. This material is based upon work supported by the National Science Foundation under Grant No. 9983708.

In section 3, we motivate the use of inexact modified Newton iterations to solve the systems of nonlinear equations of IRK methods. A detailed presentation of the new preconditioner is given in section 4. In section 5, the free parameters of the preconditioned Jacobian matrix corresponding to the Dahlquist test equation are optimized with respect to various criteria. In section 6, we present a numerical experiment for a reaction-diffusion problem, showing that the new preconditioner is effective. A short conclusion is given in section 7.

2. The implicit system of ODEs and IRK methods. We consider an implicit system of ordinary differential equations (ODEs)

$$\frac{d}{dt}a(t,y) = f(t,y), \tag{1}$$

where $y = (y^1, \dots, y^n)^T \in \mathbf{R}^n$. When $a(t,y) \equiv y$ we obtain a standard system of ODEs $\frac{d}{dt}y = f(t,y)$. We suppose that an initial value y_0 at t_0 is given and we assume that

$$a_y(t,y) := \frac{\partial}{\partial y}a(t,y) \quad \text{is invertible} \tag{2}$$

in a neighborhood of the solution of the initial value problem. Applying the chain rule to the left-hand side of (1) and then inverting $a_y(t,y)$, we obtain an explicit system of ODEs

$$\frac{d}{dt}y = a_y^{-1}(t,y)\left(f(t,y) - a_t(t,y)\right), \tag{3}$$

where $a_t(t,y) := \frac{\partial}{\partial t}a(t,y)$. In this paper we consider IRK methods applied directly to (1), not to (3). This has the advantage of not requiring the exact computation of $a_t(t,y)$ and $a_y^{-1}(t,y)$. It is assumed that the system of ODEs presents some stiffness, thus justifying the application of implicit methods.

The direct application of IRK methods to the implicit system of ODEs (1) can be defined as follows [19]:

Definition 2.1. *One step $y_0 \mapsto y_1$ from t_0 to t_0+h of an s-*stage implicit Runge-Kutta (IRK) method *applied to (1) with initial value y_0 at t_0 and stepsize h is defined implicitly by*

$$a(t_0 + c_i h, Y_i) - \left(a(t_0, y_0) + h\sum_{j=1}^{s} a_{ij} f(t_0 + c_j h, Y_j)\right) = 0 \quad \text{for } i = 1, \dots, s, \tag{4a}$$

$$a(t_0 + h, y_1) - \left(a(t_0, y_0) + h\sum_{j=1}^{s} b_j f(t_0 + c_j h, Y_j)\right) = 0. \tag{4b}$$

The RK coefficients are given by $(b_j)_{j=1,\dots,s}$, $(c_j)_{j=1,\dots,s}$, and $A := (a_{ij})_{i,j=1,\dots,s}$. The equations (4a) define a nonlinear system of dimension $s{\cdot}n$ to be solved for the s *internal stages* Y_i for $i = 1, \dots, s$. The numerical approximation y_1 at $t_0 + h$ is then given by the solution of the n-dimensional implicit system (4b). In this paper we concentrate the discussion on the solution of the system of nonlinear equations (4a). The value y_1 can then be obtained by standard application of modified Newton iterations to (4b) when $a(t,y)$ is nonlinear. It can also be obtained explicitly in certain situations. For example when the IRK method is *stiffly accurate*, i.e., when $a_{sj} = b_j$ for $j = 1, \dots, s$, this value is directly given by $y_1 = Y_s$.

3. Modified Newton iterations for the internal stages. The system of nonlinear equations (4a) for the s internal stages is usually solved by modified Newton iterations with approximate/modified Jacobian matrix

$$L := I_s \otimes M - hA \otimes J \quad \text{where} \quad M \approx a_y(t_0, y_0), \quad J \approx f_y(t_0, y_0). \tag{5}$$

Here, the symbol $\otimes$ denotes the matrix tensor product and I_s is the identity matrix in $\mathbf{R}^s$. Modified Newton iterations read

$$L\Delta Y^k = -F(Y^k), \qquad Y^{k+1} = Y^k + \Delta Y^k, \qquad k = 0, 1, 2, \ldots, \tag{6}$$

where $Y := (Y_1^T, \ldots, Y_s^T)^T$ is a vector collecting the s internal stages and $F(Y)$ corresponds to the left-hand side of (4a). Hence, each modified Newton iteration requires the solution of an $(s \cdot n)$-dimensional system of linear equations. A direct decomposition of the approximate Jacobian matrix L is generally inefficient when $s \geq 2$. However, the computational cost of its decomposition can be greatly reduced by exploiting its special structure. For example by diagonalizing the RK coefficient matrix A

$$SAS^{-1} = \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_s),$$

the approximate Jacobian matrix L can be transformed into a block-diagonal matrix

$$(S \otimes I_n)L(S^{-1} \otimes I_n) = I_s \otimes M - h\Lambda \otimes J = \begin{pmatrix} M - h\lambda_1 J & & O \\ & \ddots & \\ O & & M - h\lambda_s J \end{pmatrix}. \tag{7}$$

This kind of transformation can dramatically reduce the number of arithmetic operations when decomposing L and it allows for parallelism. Unfortunately, all eigenvalues (except one when s is odd) of most standard IRK methods, such as Radau IIA, Gauss, and Lobatto IIIC methods, arise as conjugate complex eigenpairs with nonzero imaginary parts. This impairs parallelism and significantly increases the decomposition cost of the transformed approximate Jacobian matrix (7) compared to the situation where all eigenvalues are real [16, Section IV.8]. Moreover, if several distinct IRK methods are used in a partitioned/additive way, such as for SPARK methods [18, 20, 26], this diagonalization technique cannot be applied since distinct RK matrices have in general distinct eigenvectors. Ideally, the decomposition cost of the approximate Jacobian matrix for s-stage IRK methods should be reduced to at most s independent decompositions of matrices of dimension n.

Various iteration schemes to solve the nonlinear equations of IRK methods have been suggested [6, 7, 11, 12, 13, 14, 17, 27]. The schemes of [6, 17, 27] can be interpreted as the direct application of modified Newton iterations with a modified Jacobian. Actually it could certainly be valuable to use such modified Jacobians as preconditioners to the system of linear equations of the modified Newton method (6) with approximate Jacobian matrix (5). The schemes of [7, 11, 12, 13, 14] do not have the same interpretation. They have an inherent sequential structure and can be interpreted as nonlinear Gauss-Seidel iterations. All aforementioned schemes require the solution of a sequence of s linear systems of dimension n and are therefore not truly fully parallelizable compared to the iterations developed in this paper. The various coefficients introduced in these *ad hoc* iterations are generally tuned for the numerical IRK solution y_1 (4) of the Dahlquist test equation

$$y' = \lambda y, \qquad \mathrm{Re}(\lambda) \leq 0. \tag{8}$$

Unfortunately, when $\mathrm{Re}(\lambda) \to -\infty$, none of the aforementioned methods is asymptotically correct for the internal stages Y_i. In contrast, the preconditioner introduced in this paper is asymptotically correct by construction, as is the one presented in [19, 21].

To reduce the amount of computations, instead of solving at each modified Newton iteration (6) the linear system exactly, we can solve it approximately by application of a preconditioned linear iterative method, as was already proposed in [19, 21]. Hence, we obtain a sequence of iterates $\widetilde{Y}^k$ with a residual error $r_k := L\Delta\widetilde{Y}^k + F(\widetilde{Y}^k)$ after each iteration. Theoretical and practical conditions to ensure convergence of such *inexact modified Newton iterations* to the solution of the nonlinear system of equations (4a) are given in [19] in a general framework. The use of linear iterative methods for the solution of implicit integration methods was also considered in [2, 5, 9], with an emphasis on preconditioning in [3]. Inexact Newton-type methods are generally considered to be amongst the most efficient ways to solve nonlinear system of equations [8, 23].

In this paper we present a preconditioner to the approximate Jacobian matrix (5) requiring s independent decompositions of matrices of dimension n. Each matrix to be decomposed depends on a free and distinct parameter. If these parameters are all chosen to be equal, only one matrix of dimension n needs to be decomposed. This is of high interest when using serial computers since the decomposition cost is thus really minimal and equivalent to the decomposition cost corresponding to the implicit Euler method. In fact a major interest of this new preconditioner is that not only matrix decompositions can be executed in parallel, but also the solution of the systems of linear equations. This makes this preconditioner truly parallel compared to the one developed in [19, 21] which entails the sequential solution of systems of linear equations and which is based on the W-transformation of the RK coefficients and an approximate block-LU decomposition.

4. Preconditioning the linear systems.

For the sake of generality we consider a linear transformation $T \otimes I$ of the approximate Jacobian matrix L (5). At each modified Newton iteration we obtain a system of linear equations

$$Kx = b \tag{9a}$$

with matrix

$$K = (T \otimes I)L(T^{-1} \otimes I) = I \otimes M - hTAT^{-1} \otimes J \tag{9b}$$

To solve the linear system (9) we consider the application of linear iterative methods, such as GMRES [10, 15, 22, 24, 25], with a preconditioner $Q \approx K^{-1}$. We take Q of the form

$$Q := H^{-1}GH^{-1},$$

where

$$H := I_s \otimes M - h\Gamma \otimes J, \qquad G := I_s \otimes M - h\Omega \otimes J, \tag{10}$$

with coefficients matrices Γ and Ω still to be determined. We choose the coefficients matrix Γ to be diagonal

$$\Gamma := \operatorname{diag}(\gamma_1, \gamma_2, \ldots, \gamma_s)$$

so that H is block-diagonal

$$H = \begin{pmatrix} H_1 & & & O \\ & H_2 & & \\ & & \ddots & \\ O & & & H_s \end{pmatrix} \tag{11}$$

with blocks given by

$$H_i := M - h\gamma_i J \quad \text{for } i = 1, \ldots, s.$$

The matrices H_i are independent, hence they can be decomposed in parallel. Solving a linear system with matrix H can also be done in parallel since it is block-diagonal. This is the main advantage of this preconditioner compared to the one presented in [19, 21]. Assuming at least s processors on a parallel computer, the local cost on the ith processor of computing the matrix-vector product QKv consists essentially of:

- one decomposition of matrix H_i;
- solving two linear systems with the decomposed matrix H_i;
- two matrix-vector products with matrix M;
- two matrix-vector products with matrix J;
- some communication with other processors according to the nonzero coefficients of the ith row of matrices TAT^{-1} and Ω.

The coefficients of matrices Γ and Ω remain to be fixed to some values. Assuming the coefficients γ_i for $i = 1, \ldots, s$ to be given, it is natural to determine the coefficients Ω_{ij} for $i, j = 1, \ldots, s$ such that the preconditioner Q is asymptotically correct when considering the Dahlquist test equation (8). Denoting $z := h\lambda$ we obtain

$$Q(z) = H^{-1}(z)G(z)H^{-1}(z), \qquad K(z) = T(I_s - zA)T^{-1} \tag{12a}$$

where

$$H(z) = I_s - z\Gamma, \qquad G(z) = I_s - z\Omega. \tag{12b}$$

Defining $J(z) := Q(z)K(z)$ we have

$$J(z) = H^{-1}(z)G(z)H^{-1}(z)T(I_s - zA)T^{-1} \tag{13}$$

At $z = 0$ we have $J(0) = I_s$. We determine Ω such that

$$J(z) \longrightarrow I_s \quad \text{for} \quad |z| \longrightarrow \infty. \tag{14}$$

From (13) we easily obtain the following result.

THEOREM 4.1. *If the RK matrix A is invertible and the coefficients γ_i for $i = 1, \ldots, s$ satisfy $\gamma_i > 0$ then condition (14) holds if and only if*

$$\Omega = \Gamma T A^{-1} T^{-1} \Gamma. \tag{15}$$

When the RK matrix A is not invertible, an expression similar to (15) can be obtained for certain classes of IRK methods such as the Lobatto IIIA and Lobatto IIIB methods.

From now on we assume for simplicity that the matrix A of RK coefficients is invertible. The coefficients γ_i for $i = 1, \ldots, s$ remain free, they are only required to satisfy $\gamma_i > 0$ which is a natural assumption to ensure the invertibility of the matrices H_i. When all these coefficients are equal, only one matrix decomposition of dimension n is needed. This is quite advantageous on a serial computer compared to some other implementations of implicit Runge-Kutta methods [16, 17]. This new approach can also be extended to SPARK methods [18, 26] applied to differential-algebraic equations [20].

EXAMPLE 4.2. *Consider the* 2*-stage Radau IIA method whose RK matrix is given by*

$$A = \begin{pmatrix} 5/12 & -1/12 \\ 3/4 & 1/4 \end{pmatrix}. \tag{16}$$

Assuming $T = I$ *and* $\Gamma = \gamma I$, *we obtain* $\Omega = \gamma^2 A^{-1}$ *and*

$$J(z) = \frac{1}{(1-\gamma z)^2} \begin{pmatrix} 1 - (5/12 + 3\gamma^2/2)z + \gamma^2 z^2 & (1/12 - \gamma^2/2)z \\ (9\gamma^2/2 - 3/4)z & 1 - (1/4 + 5\gamma^2/2)z + \gamma^2 z^2 \end{pmatrix}.$$

Ideally we would like $J(z)$ *to be as close as possible to the identity matrix. By taking* $\gamma := 1/\sqrt{6} \approx 0.408$ *the matrix* $J(z)$ *becomes diagonal with double eigenvalue*

$$\lambda(z) = 1 + z\frac{\sqrt{2/3} - 2/3}{(1 - z/\sqrt{6})^2}.$$

Hence, the condition number of $J(z)$ *satisfies* $\kappa(J(z)) = 1$ *and is therefore minimal. This choice of* γ *is not only natural, but also optimal in other ways which will be stated precisely in the next section 5.*

5. Optimal choices of coefficients γ_i**.** For simplicity we assume the matrix A of RK coefficients to be invertible. The coefficients Ω_{ij} for $i, j = 1, \ldots, s$ are given by (15). The coefficients $\gamma_i > 0$ for $i = 1, \ldots, s$ remain to be determined. We consider the preconditioned matrix $J(z) := Q(z)K(z)$ given in (13) corresponding to the Dahlquist test equation (8). This matrix $J(z)$ depends on the coefficients γ_i of Γ. Ideally we would like to have $J(z) = I_s$ and this is of course generally not possible. Hence, we must define some criterion to determine the remaining coefficients γ_i. Here we present a few criteria based on the solution of optimization problems

$$\min_{\gamma_1 > 0, \ldots, \gamma_s > 0} \phi(\gamma_1, \ldots, \gamma_s) \tag{17}$$

for diverse functions ϕ in relation with the convergence properties of preconditioned linear iterative methods.

A first choice for ϕ is given by

$$\phi_\infty(\gamma_1, \ldots, \gamma_s) := \max_{\mathrm{Re}(z) \le 0} \left(\max_{i=1,\ldots,s} |\lambda_i(J(z)) - 1| \right) \tag{18}$$

where $\lambda_i(J(z))$ for $i = 1, \ldots, s$ are the eigenvalues of matrix $J(z)$. The goal is to obtain a good clusterization of the eigenvalues of the preconditioned approximate Jacobian to the value 1 to ensure a rapid convergence of the preconditioned linear iterative method. This is justified since the convergence of most linear iterative methods, such as GMRES [24, 25], is dictated by the eigenspectrum of the preconditioned matrix [4, 15, 22, 24]. We have

$$\phi_\infty(\gamma_1,\ldots,\gamma_s) = \max_{i=1,\ldots,s}\left(\max_{\mathrm{Re}(z)\le 0}|\lambda_i(J(z))-1|\right) = \max_{i=1,\ldots,s}\left(\max_{\mathrm{Re}(z)=0}|\lambda_i(J(z))-1|\right)$$

from the maximum principle. Finding a solution to the global optimization problem (17)-(18) is certainly difficult and remains an open problem. Nevertheless, for example for the 2-stage Radau IIA (16) and $T = I$ we have obtained numerically $\gamma_{1,\min} \approx 0.25, \gamma_{2,\min} \approx 0.66$ and $\phi_{\infty,\min} \approx 0.04$ in (18). In the situation where we assume additionally that all parameters γ_i are equal to a unique value γ, i.e., $\Gamma = \gamma \cdot I_s$, we have

$$|\lambda_i(J(z))-1| = \left|\frac{\gamma z}{(1-\gamma z)^2}\right|\left|\frac{\mu_i}{\gamma}\left(1-\frac{\gamma}{\mu_i}\right)^2\right|$$

where μ_i for $i = 1,\ldots,s$ are the eigenvalues of A. The maximum value of $|\gamma z/(1-\gamma z)^2|$ for $\mathrm{Re}(z) \le 0$ is independent of γ and is equal to $1/2$ (value taken at $z = \pm\mathrm{i}/\gamma$). Hence, we need to determine γ as the optimal solution of

$$\min_{\gamma>0}\left(\max_{i=1,\ldots,s} f_i(\gamma)\right) \tag{19a}$$

where

$$f_i(\gamma) := \left|\frac{\mu_i}{\gamma}\left(1-\frac{\gamma}{\mu_i}\right)^2\right| = \frac{|\mu_i|}{\gamma} + \frac{\gamma}{|\mu_i|} - 2\cos(\theta_i) \tag{19b}$$

with $\theta_i := \arg(\mu_i)$. The minimum value of $f_i(\gamma)$ holds at $\gamma = |\mu_i|$.

THEOREM 5.1. *The solution $\gamma_{\min}$ of the optimization problem (19) satisfies either $\gamma_{\min} \in \{|\mu_1|,\ldots,|\mu_s|\}$ or lies at the intersection of two curves f_i, i.e., $f_j(\gamma_{\min}) = f_k(\gamma_{\min})$ for two distinct indices j and k.*

Proof. The functions $f_i(\gamma)$ are convex and satisfy

$$\lim_{\gamma\to 0^+} f_i(\gamma) = +\infty, \quad \lim_{\gamma\to+\infty} f_i(\gamma) = +\infty.$$

Hence, the result follows. ◻

For $s = 2$ and $\mu_1 = \overline{\mu}_2$ we obtain $\gamma_{\min} = |\mu_1|$. For example, for the 2-stage Radau IIA method (16) we have $\gamma_{\min} = 1/\sqrt{6}$ and $\phi_{\infty,\min} = 1 - \sqrt{6}/3 \approx 0.184$ in (18).

A second choice for ϕ similar to (18) is given by

$$\phi_1(\gamma_1,\ldots,\gamma_s) := \max_{\mathrm{Re}(z)\le 0}\left(\sum_{i=1}^{s}|\lambda_i(J(z))-1|\right). \tag{20}$$

Finding a solution to the global optimization problem (17)-(20) is also an open problem. If we assume in addition that all parameters γ_i are equal to a unique value γ, we obtain

$$\phi_1(\gamma,\ldots,\gamma) = \max_{\mathrm{Re}(z)\le 0}\left|\frac{\gamma z}{(1-\gamma z)^2}\right|\sum_{i=1}^{s}\left(\frac{|\mu_i|}{\gamma} + \frac{\gamma}{|\mu_i|} - 2\cos(\theta_i)\right).$$

Hence, we need to determine γ as the optimal solution of

$$\min_{\gamma>0} f(\gamma) \tag{21a}$$

where

$$f(\gamma) := \frac{1}{\gamma}\sum_{i=1}^{s}|\mu_i| + \gamma\sum_{i=1}^{s}\frac{1}{|\mu_i|} - 2\sum_{i=1}^{s}\cos(\theta_i). \tag{21b}$$

The minimum value of this optimization problem holds at

$$\gamma_{\min} = \sqrt{\frac{\sum_{i=1}^{s}|\mu_i|}{\sum_{i=1}^{s}1/|\mu_i|}}.$$

For $s = 2$ and $\mu_1 = \overline{\mu}_2$ we obtain again $\gamma_{\min} = |\mu_1|$. For example, for the 2-stage Radau IIA method (16) we obtain again $\gamma_{\min} = 1/\sqrt{6}$.

A third natural choice for ϕ in (17) is given by

$$\phi_F(\gamma_1, \ldots, \gamma_s) := \max_{\mathrm{Re}(z)\leq 0} \|J(z) - I_s\|_F \tag{22}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The goal here is to have the preconditioned approximate Jacobian as close to the identity matrix as possible. From the maximum principle we have

$$\max_{\mathrm{Re}(z)\leq 0} \|J(z) - I_s\|_F = \max_{\mathrm{Re}(z)=0} \|J(z) - I_s\|_F.$$

Finding a solution to the global optimization problem (17)-(22) is also an open problem. Nevertheless, for example for the 2-stage Radau IIA (16) and $T = I$ we have obtained numerically $\gamma_{1,\min} \approx 0.32, \gamma_{2,\min} \approx 0.60$ and $\phi_{F,\min} \approx 0.16$ in (22). Assuming $\Gamma = \gamma \cdot I_s$, we have

$$\|J(z) - I_s\| = \left|\frac{\gamma z}{(1-\gamma z)^2}\right| \|\gamma A^{-1} + \gamma^{-1}A + 2I_s\|.$$

Hence, we need to determine γ as the optimal solution of

$$\min_{\gamma>0} g(\gamma) \tag{23a}$$

where

$$g(\gamma) := \|\gamma A^{-1} + \gamma^{-1}A + 2I_s\|_F^2. \tag{23b}$$

The solution $\gamma_{\min}$ of the optimization problem (23) must be one of the roots of $g'(\gamma)$. For example, for the 2-stage Radau IIA method (16) we obtain

$$g(\gamma) = 29\gamma^2 + 16\gamma + \frac{11}{2} + \frac{8}{3\gamma} + \frac{29}{36\gamma^2}.$$

We have

$$g'(\gamma) = \frac{58}{\gamma^3}\left(\gamma - \frac{1}{\sqrt{6}}\right)\left(\gamma + \frac{1}{\sqrt{6}}\right)\left(\gamma^2 + \frac{8}{29}\gamma + \frac{1}{6}\right) \text{ and } g''(\gamma) = 58 + \frac{16}{3\gamma^3} + \frac{29}{6\gamma^4}.$$

Therefore, the minimum of $g(\gamma)$ is once again attained at $\gamma_{\min} = 1/\sqrt{6}$.

6. A numerical experiment. We consider a reaction-diffusion problem, the *Brusselator* system in one spatial variable, see [16],

$$\begin{aligned}\frac{\partial u}{\partial t} &= A + u^2 v - (B+1)u + \alpha \frac{\partial^2 u}{\partial x^2},\\ \frac{\partial v}{\partial t} &= Bu - u^2 v + \alpha \frac{\partial^2 v}{\partial x^2}\end{aligned}$$

where $x \in [0,1]$ and $\alpha \geq 0, A,$ and B are constant parameters. The boundary conditions for u and v are $u(0,t) = 1 = u(1,t), v(0,t) = 3 = v(1,t), u(x,0) = 1 + \sin(2\pi x), v(x,0) = 3$. We apply the method of lines by discretizing the diffusion terms using finite differences on a grid of N points $x_i = i/(N+1)$ for $i = 1,\ldots,N$, $\Delta x = 1/(N+1)$. We consider the value $N = 500$ and parameters $A = 1, B = 3, \alpha = 0.02$. We obtain a system of $2N = 1000$ differential equations

$$\frac{du_i}{dt} = 1 + u_i^2 v_i - 4u_i + \frac{0.02}{(\Delta x)^2}\left(u_{i-1} - 2u_i + u_{i+1}\right), \tag{24a}$$

$$\frac{\partial v_i}{\partial t} = 3u_i - u_i^2 v_i + \frac{0.02}{(\Delta x)^2}\left(v_{i-1} - 2v_i + v_{i+1}\right) \tag{24b}$$

where $u_0(t) = 1 = u_{N+1}(t)$, $v_0(t) = 3 = v_{N+1}(t)$, $u_i(0) = 1 + \sin(2\pi x_i)$, and $v_i(0) = 3$ for $i = 1,\ldots,N$. We consider the time interval $[t_0, t_{\text{end}}] = [0,10]$ and the $s = 3$-stage Radau IIA method. The eigenvalues of the Jacobian matrix J have a wide spectrum. The largest negative eigenvalue of J is close to -20000, so the system is stiff. By ordering the variables as $y = (u_1, v_1, u_2, v_2, u_3, v_3, \ldots)$, matrices $I - \gamma hJ$ have a bandwidth of 2. They are decomposed using the routine `DGBTRF` of LAPACK for banded matrices [1]. We have taken the absolute and relative error tolerances for each component equal to a certain error tolerance TOL. We give some statistics obtained with the code SPARK3 on this problem in Table 1. The *TOL-norm* $\|\cdot\|_{\text{tol}}$ is a scaled 2-norm which depends on absolute and relative error tolerances for each component, $ATOL_i$ and $RTOL_i$ respectively, to be specified by the user

$$\|y\|_{\text{tol}} := \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i}{D_i}\right)^2}, \qquad D_i := ATOL_i + RTOL_i|y_i|. \tag{25}$$

We consider the preconditioner with $\gamma_i = \gamma$ for $i = 1,\ldots,s$ both for the sequential and parallel versions in order to obtain identical results but different timings in order to compare these versions. In the parallel version we decompose the matrix $M - \gamma hJ$ on each processor. For the results in Table 1, the preconditioner is applied to the system of linear equations only once per modified Newton iteration. For the results in Table 2 we consider the application of one preconditioned Richardson iteration as preconditioned linear iterative solver, corresponding to two applications of the preconditioner per modified Newton iteration. We do not observe a significant gain compared to Table 1. This demonstrates numerically the quality of the preconditioner. The numerical experiments discussed in this section were made on a SGI Power Challenge using 3 processors for the parallel version.

7. Conclusion. We have considered the application of IRK methods to implicit systems of ODEs. The major difficulty and computational bottleneck for an efficient

Table 1
Results for the 3*-stage Radau IIA method on the Brusselator equations (24) with* 1 *preconditioner solve per modified Newton iteration.*

Error tolerance TOL	10^{-3}	10^{-6}	10^{-9}
Measured error in TOL-norm $\|\cdot\|_{\text{tol}}$	0.59	0.39	0.19
CPU-time [s] - sequential	1.19	2.38	6.35
CPU-time [s] - parallel	0.70	2.02	4.12
number of steps	24	49	190
number of rejected steps	0	5	1
number of function evaluations	243	531	1524
number of modified Newton iterations	75	176	508
number of J evaluations	6	1	1
number of P decompositions	21	29	28
number of P solves	81	177	508
number of matrix-vector products	0	0	0

Table 2
Results for the 3*-stage Radau IIA method on the Brusselator equations (24) with* 2 *preconditioner solves per modified Newton iteration.*

Error tolerance TOL	10^{-3}	10^{-6}	10^{-9}
Measured error in TOL-norm $\|\cdot\|_{\text{tol}}$	0.11	0.81	0.11
CPU-time [s] - sequential	1.44	4.50	11.01
CPU-time [s] - parallel	0.82	2.60	6.61
number of steps	18	50	187
number of rejected steps	0	3	1
number of function evaluations	183	606	1533
number of modified Newton iterations	58	202	511
number of J evaluations	3	1	1
number of P decompositions	18	32	24
number of P solves	117	401	1019
number of matrix-vector products	56	199	508

implementation of these numerical integration methods is to solve the resulting nonlinear systems of equations. Linear systems of the modified Newton method can be solved approximately with a preconditioned linear iterative method. In this paper we have developed a preconditioner which is asymptotically exact for stiff systems. The preconditioner requires m independent matrices of the type $M - \gamma_i hJ$ ($\gamma_i > 0$, $i = 1, \dots, m$) to be decomposed. The integer parameter m of the number of decompositions satisfies $1 \leq m \leq s$ and is free, $m = 1$ being particularly advantageous for serial computers. Not only the decompositions parts of the preconditioner are parallelizable, but the solution parts involved when applying the preconditioner are also parallelizable, making the preconditioner truly parallel. This preconditioner has been implemented in the code SPARK3 and is shown to be effective on a problem with diffusion.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK User's Guide*, SIAM, Philadelphia, Second Edition, 1995.

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

[3] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold, *Using Krylov methods in the solution of large-scale differential-algebraic systems*, SIAM J. Sci. Comput., 15 (1994), pp. 1467–1488.

[4] S. L. Campbell, I. C. F. Ipsen, C. T. Kelley, and C. D. Meyer, *GMRES and the minimal polynomial*, BIT, 36 (1996), pp. 32–43.

[5] T. F. Chan and K. R. Jackson, *The use of iterative linear equation solvers in codes for large systems of stiff IVPs for ODEs*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 378–417.

[6] G. J. Cooper and J. C. Butcher, *An iteration scheme for implicit Runge-Kutta methods*, IMA J. Numer. Anal., 3 (1983), pp. 127–140.

[7] G. J. Cooper and R. Vignesvaran, *A scheme for the implementation of implicit Runge-Kutta methods*, Computing, 45 (1990), pp. 321–332.

[8] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[9] C. W. Gear and Y. Saad, *Iterative solution of linear equations in ODE codes*, SIAM J. Sci. Stat. Comput., 4 (1983), pp. 583–601.

[10] G. Golub and C. F. van Loan, *Matrix computations*, Johns Hopkins Studies in the Math. Sci., The Johns Hopkins Univ. Press, Baltimore and London, Third Edition, 1996.

[11] S. González-Pinto, C. González-Concepción, and J. I. Montijano, *Iterative schemes for Gauss methods*, Comput. Math. Applic., 27 (1994), pp. 67–81.

[12] S. González-Pinto, J. I. Montijano, and S. Pérez-Rodríguez, *On the numerical solution of stiff IVPs by Lobatto IIIA Runge-Kutta methods*, J. Comput. Appl. Math., 82 (1997), pp. 129–148.

[13] ———, *Implementation of high-order implicit Runge-Kutta methods*, Comput. Math. Applic., 41 (2001), pp. 1009–1024.

[14] S. González-Pinto, J. I. Montijano, and L. Rández, *Iterative schemes for three-stage implicit Runge-Kutta methods*, Appl. Numer. Math., 17 (1995), pp. 363–382.

[15] A. Greenbaum, *Iterative methods for solving linear systems*, vol. 17 of Frontiers in Applied Math., SIAM, Philadelphia, 1997.

[16] E. Hairer and G. Wanner, *Solving ordinary differential equations II. Stiff and differential-algebraic problems*, vol. 14 of Comput. Math., Springer, Berlin, Second Revised Edition, 1996.

[17] W. Hoffmann and J. J. B. de Swart, *Approximating Runge-Kutta matrices by triangular matrices*, BIT, 37 (1997), pp. 346–354.

[18] L. O. Jay, *Structure preservation for constrained dynamics with super partitioned additive Runge-Kutta methods*, SIAM J. Sci. Comput., 20 (1999), pp. 416–446.

[19] ———, *Inexact simplified Newton iterations for implicit Runge-Kutta methods*, SIAM J. Numer. Anal., 38 (2000), pp. 1369–1388.

[20] ———, *Iterative solution of nonlinear equations for SPARK methods applied to DAEs*, Numer. Algor., (2002). To appear.

[21] L. O. Jay and T. Braconnier, *A parallelizable preconditioner for the iterative solution of implicit Runge-Kutta type methods*, J. Comput. Appl. Math., 111 (1999), pp. 63–76.

[22] C. T. Kelley, *Iterative methods for linear and nonlinear equations*, vol. 16 of Frontiers in applied mathematics, SIAM, Philadelphia, 1995.

[23] W. Rheinboldt, *Methods for solving systems of nonlinear equations*, SIAM Classics in Appl. Math., SIAM, Philadelphia, Second Revised and Expanded Edition, 1998.

[24] Y. Saad, *Iterative methods for sparse linear systems*, PWS Publ. Co., Boston, MA, 1996.

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

[26] M. Schaub and B. Simeon, *Blended lobatto methods in multibody dynamics*, ZAMM, 7 (2001), pp. 0–10.

[27] P. J. van der Houwen and J. J. B. de Swart, *Triangularly implicit iteration methods for ODE-IVP solvers*, SIAM J. Sci. Comput., 18 (1997), pp. 41–55.