ADAPTIVE ITERATION TO STEADY STATE OF FLOW PROBLEMS

KARL HÖRNELL[1] and PER LÖTSTEDT[2] *

[1]*Dept of Information Technology, Scientific Computing, Uppsala University, SE-75104 Uppsala, Sweden. email: karl@tdb.uu.se*

[2]*Dept of Information Technology, Scientific Computing, Uppsala University, SE-75104 Uppsala, Sweden. email: perl@tdb.uu.se*

Abstract

Runge-Kutta time integration is used to reach the steady state solution of discretized partial differential equations. Continuous and discrete parameters in the method are adapted to the particular problem by minimizing the residual in each step, if this is possible. Algorithms for parameter optimization are devised and analyzed. Solutions of the nonlinear Euler and Navier-Stokes equations for compressible flow illustrate the methods.

Keywords: steady state solution, flow problems, Runge-Kutta iteration, adapted parameters

AMS subject classification: 65F10, 65M20, 65N22

1 Introduction

A common method to compute the time-independent solution of a partial differential equation (PDE) is to integrate the equation in time until the time-derivatives vanish or are sufficiently small. Then we have the solution to the steady state problem. The time stepping method is often an explicit Runge-Kutta (RK) method. The process can be accelerated by a multigrid algorithm. This approach is the standard method in computational fluid dynamics for the Euler and Navier-Stokes equations in compressible flow. Examples of RK time-stepping to steady state are found in [9], [10], [11], [12], [15], [16], [17], [22], [23].

*Financial support has been obtained from the Swedish Institute for Applied Mathematics.

Assume that we are interested in the solution u of a system of nonlinear equations

$$r(u) = 0,\ u, r \in \mathbb{R}^N,$$

obtained after space discretization of a PDE. Add a time-derivative and discretize in time by an explicit m-stage RK method. Let $u^{(n)}$ be the numerical solution at time step n. Then one iteration from n to $n+1$ is

$$\begin{array}{rcl} u^0 & = & u^{(n)} \\ u^1 & = & u^0 + \alpha_1 \kappa T\, r(u^0) \\ & \vdots & \\ u^m & = & u^0 + \alpha_m \kappa T\, r(u^{m-1}) \\ u^{(n+1)} & = & u^m\,. \end{array} \tag{1}$$

The diagonal matrix T in (1) contains the time steps, κ is the scalar CFL number chosen to minimize the number of iterations and $\alpha_k, k = 1, \ldots, m$, are the parameters defining a particular RK scheme where α_m usually is 1. The iteration is terminated when $\|r(u)\|$ is sufficiently small in a suitable norm. The iteration is often robust and only two solution vectors have to be stored.

The time-steps Δt_j in T are determined locally in each cell j of the grid by the cell size and the local properties of the solution in the cell [11], [18], [22]. Let J be the Jacobian of r so that $J = \partial r/\partial u$. The time steps are such that the largest eigenvalues λ_j of TJ are of $\mathcal{O}(1)$. The Courant-Friedrichs-Lewy (CFL) number κ has its name from the classical paper reprinted in [3] and is usually taken as large as possible without incurring instability. The parameters m and α_j are constant for each RK scheme. The maximum κ depends on the chosen RK method.

Because Δt_j is usually derived from an approximation of $\lambda(TJ)$, and because the largest possible actual time steps are not necessarily the ones that result in the fastest convergence, finding a suitable κ is a non-trivial problem. A priori estimates of κ based on properties of r are not sufficiently sharp in particular for nonlinear problems. Most often a CFL number is used that has worked reasonably well for similar problems based on the user's experience, or it has to be determined by trial and error. A sufficiently small κ usually guarantees convergence but the efficiency may be low. If good convergence properties are desired, without the need to run a series of experiments, it will be necessary to use some form of adaptive technique based on optimization. The purpose of this paper is to suggest such techniques to calculate close to optimal κ, number of stages m, and number of levels l in the multigrid method.

For linear equations, the iterative method (1) is known as Richardson iteration. Richardson methods are developed in [2], [19], [20], [21]. If the size of the spectrum of the system matrix is known then optimal methods are suggested and analyzed in [2] and [20]. An upper bound of the eigenvalues of a linearized flow problem with periodic boundary conditions can be estimated and is used to calculate Δt_j in local time-stepping [18]. To determine a lower bound different from zero is much more difficult. The spectrum is approximated by GMRES iterations and then the adapted coefficients α_j in (1) are computed in [19] and [21]. Nonlinear equations are analyzed in [4] and solved in [2]. The choice of α_j for flow problems is discussed in [12], [15], [22]. There the parameters are determined by an optimal behavior for the Fourier transform of a model problem with constant coefficients.

Our paper is organized as follows. In Sect. 2, the adaptive method for selection of the CFL number κ is described and analyzed for linear systems. The adaptation of the number of RK stages m and multigrid levels l is discussed in Sect. 3. The performance of the methods is demonstrated with the solution of the Euler and Navier-Stokes equations in the final section. The norm is the Euclidean vector norm and the subordinate spectral matrix norm.

2 Continuous adaptation

A method for adaptation of the continuous CFL parameter κ in (1) is developed in this section. The algorithm is based on local minimization of the residual and is first discussed and analyzed for systems of linear equations and then generalized to nonlinear equations.

The system of N linear equations

$$Au = b \tag{2}$$

is solved with the RK method (1). Then

$$\begin{aligned} u^{(n+1)} &= u^{(n)} + \textstyle\sum_{j=1}^m \beta_j \kappa^j (TA)^j u^{(n)} - \sum_{j=1}^m \beta_j \kappa^j (TA)^{j-1} Tb \\ &= u^{(n)} - \textstyle\sum_{j=1}^m \beta_j \kappa^j (TA)^{j-1} T r^{(n)}, \end{aligned} \tag{3}$$

where $\beta_j = \prod_{i=m-j+1}^m (-\alpha_i)$ and the residual $r^{(n)} = b - Au^{(n)}$, or equivalently, expressed in terms of $r^{(n)}$

$$r^{(n+1)} = r^{(n)} + \sum_{j=1}^m \beta_j \kappa^j (AT)^j r^{(n)} = p(\kappa AT) r^{(n)}. \tag{4}$$

The polynomial p in (4) satisifes $p(0) = \beta_0 = 1$ and $p'(0) = \beta_1 = -\alpha_m = -1$. For convergence, the eigenvalues of $p(\kappa AT)$ have to reside inside the unit circle or if AT is diagonalizable an equivalent condition is

$$|p(\kappa\lambda_j)| < 1,\ \lambda_j = \lambda_j(AT),\ j = 1 : N. \tag{5}$$

In the theory of the numerical solution of ordinary differential equations [7], the stability region defines the domain $\mathcal{S}$ in which the integration method is stable for $du/dt = \mu u$. If $\mu \in \mathcal{S}$, then $|p(-\mu)| \leq 1$.

In Table 1, four different RK schemes (1) are given.

Table 1: *Coefficients for standard two-, three-, four-, and five-stage RK methods*

Stages	α_1	α_2	α_3	α_4	α_5
2	1	1	-	-	-
3	3/5	3/5	1	-	-
4	1/4	1/3	1/2	1	-
5	1/4	1/6	3/8	1/2	1

These three-, four- [22] and five-stage [16], [17], [23], coefficients are typical choices for flow problems—in particular the Euler and Navier-Stokes equations. The corresponding stability regions $\mathcal{S}$ are shown in Fig. 1. It can be shown by Taylor expansion of $|p(\mu)|$ with $\mu = iy$ and small y that part of the imaginary axis around 0 is in $\mathcal{S}$ for these methods. If $\Re\lambda_j \geq 0$, for all j, then by decreasing κ we will eventually have stability.

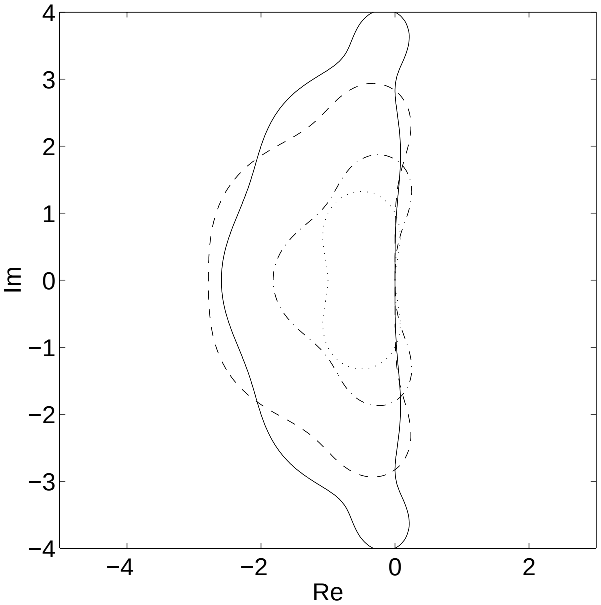


Figure 1: *Stability regions for standard two- (..), three- (-.), four- (–) and five-stage (solid) RK methods*

If a_j are the coefficients of each eigenvector v_j in $r^{(n)}$, the residual after another

q steps will be

$$r^{(n+q)} = p(\kappa AT)^q r^{(n)} = \sum_{j=1}^{N} a_j p(\kappa\lambda_j)^q v_j \, . \tag{6}$$

The eigenvectors are scaled such that $\|v_j\| = 1$. The convergence $r^{(n)} \to 0$ can be separated into two phases: a transient phase and an asymptotic phase. In the transient part the performance depends on the original composition of eigenvectors in $r^{(n)}$. The fastest decay of $r^{(n)}$ may be achieved by a κ with $|p(\kappa\lambda_j)| > 1$ for some j if a_j is small in (6). Eventually, when q is large, the convergence will be dominated by the eigenvectors with the largest $|p(\kappa\lambda_j)|$, the corresponding a_j-coefficients have about the same modulus, and we have entered the asymptotic phase. We can associate this with an asymptotic convergence factor

$$\eta = \max_j |p(\kappa\lambda_j)| \tag{7}$$

and, similarly, the optimal convergence factor of the iterative process for any *constant* κ

$$\eta_{opt} = \min_\kappa \max_j |p(\kappa\lambda_j)| \, . \tag{8}$$

If a single eigenvalue λ_j satisfies $\eta_{opt} = |p(\kappa\lambda_j)|$ for the optimal κ^*, long-term convergence becomes simple, but in the general case the optimum will be found where two or more curves $|p(\kappa\lambda_j)|$ as functions of κ intersect. Let κ^* be the CFL number satisfying $\eta(\kappa^*) = \eta_{opt}$ and λ_1 and λ_2 be the smallest and largest eigenvalues for which $|p(\kappa^*\lambda_j)| = \eta_{opt}$. No *constant* κ can outperform κ^* in the asymptotic phase.

When κ varies, $\kappa\lambda_j$ moves on rays given by λ_j from the origin in $\mathcal{S}$. If we let $\kappa > \kappa^*$, usually $|p(\kappa\lambda_2)| > \eta_{opt} > |p(\kappa\lambda_1)|$ with the stability regions in Fig. 1, and if we let $\kappa < \kappa^*$, we have $|p(\kappa\lambda_2)| < \eta_{opt} < |p(\kappa\lambda_1)|$.

Three cases can be distinguished: λ_1 and λ_2 are complex, either λ_1 or λ_2 is real and the other one is complex, and λ_1 and λ_2 are real. We analyze the first case here. The other two can be treated similarly.

The major contribution to the residual in (6) in the asymptotic phase comes from the eigenvectors v_1 and v_2 and their conjugates $\bar{v}_1$ and $\bar{v}_2$. We split the residual into the dominant part r_D and a remainder term r_R

$$\begin{aligned}
r^{(n+q)} &= r_D^{(n+q)} + r_R^{(n+q)}, \\
r_D^{(n+q)} &= p(\kappa\lambda_1)^q \left(a_1 v_1 + \bar{a}_1 \theta_1^q \bar{v}_1 + a_2 \theta_2^q v_2 + \bar{a}_2 \theta_3^q \bar{v}_2\right), \\
r_R^{(n+q)} &= \textstyle\sum_{j=5}^{N} a_j p(\kappa\lambda_j)^q v_j = p(\kappa\lambda_1)^q \left(\sum_{j=5}^{N} a_j \left(\frac{p(\kappa\lambda_j)}{p(\kappa\lambda_1)}\right)^q v_j\right), \\
\theta_1 &= p(\kappa\bar{\lambda}_1)/p(\kappa\lambda_1),\ \theta_2 = p(\kappa\lambda_2)/p(\kappa\lambda_1),\ \theta_3 = p(\kappa\bar{\lambda}_2)/p(\kappa\lambda_1).
\end{aligned} \tag{9}$$

If two eigenvalues define the optimal point, we have with the optimal κ, $|\theta_j| = 1,\ j = 1, 2, 3$, and $|p(\kappa\lambda_j)/p(\kappa\lambda_1)| < 1$. Hence, for large q the convergence of $r^{(n+q)}$ is determined by $r_D^{(n+q)}$ in (9) and $r_R^{(n+q)} \to 0$ faster than $|p(\kappa\lambda_j)|^q$.

Let V be the eigenvector matrix $V = (v_1, \bar{v}_1, v_2, \bar{v}_2)$ and $W = V^H V$. In the asymptotic phase, starting with the initial residual, $n = 0$, the residual is well approximated by the four vectors in V and its norm is

$$\begin{aligned}\|r_D^{(q)}\|^2 = \;& 2|p(\kappa\lambda_1)|^{2q}|a_1|^2 + 2|p(\kappa\lambda_2)|^{2q}|a_2|^2 \\ & +2\Re(a_1^2 W_{21} p(\kappa\lambda_1)^{2q} + a_2^2 W_{43} p(\kappa\lambda_2)^{2q}) \\ & +4\Re(a_1\overline{a}_2 W_{31} p(\kappa\lambda_1)^q p(\overline{\kappa\lambda_2})^q + a_1 a_2 W_{41} p(\kappa\lambda_1)^q p(\kappa\lambda_2)^q).\end{aligned} \tag{10}$$

In (10) the symmetries in W have been utilized to simplify the expressions. By $\Re z \leq |z|$, the notation $b_j = |a_j||p(\kappa\lambda_j)|^q,\ j = 1, 2$, and $b_1 b_2 \leq 0.5(b_1^2 + b_2^2)$ we derive an upper bound on $\|r_D^{(q)}\|^2$ in (10)

$$\|r_D^{(q)}\|^2 \leq 2(b_1^2(1 + |W_{21}| + |W_{31}| + |W_{41}|) + b_2^2(1 + |W_{43}| + |W_{31}| + |W_{41}|)).$$

Since $|W_{ij}| \leq 1$ and asymptotically $b_1 = b_2$, we obtain

$$\|r_D^{(q)}\| \leq 4|a_1||p(\kappa\lambda_1)|^q. \tag{11}$$

Each term will decrease exponentially on average in (10), but the four last ones may also oscillate. If the arguments of $p(\kappa\lambda_1)$ and $p(\kappa\lambda_2)$ are ϕ_1 and ϕ_2 respectively, the frequencies will be $2\phi_1$, $2\phi_2$, $\phi_1 - \phi_2$ and $\phi_1 + \phi_2$. Near-parallel eigenvectors causing iteration methods to temporarily deviate from the behavior one would expect by looking at the eigenvalues alone is a well-known problem and has motivated the study of so-called pseudospectra when solving certain PDEs [5], [24]. Here the effect is that the residual norm may increase in individual steps although the chosen κ is stable, but a bound on the oscillations is given by (11).

The convergence will be monotone if $W_{ij},\ i \neq j$, and $|p(\kappa\lambda_1)|$ are sufficiently small. In the simpler, special case with two dominant vectors v_1 and $\bar{v}_1$ the convergence is monotone if

$$\|r_D^{(q)}\|^2 = 2|a_1|^2|p(\kappa\lambda_1)|^{2q} + 2\Re(a_1^2 W_{21} p(\kappa\lambda_1)^{2q}) < \|r_D^{(q-1)}\|^2. \tag{12}$$

Let

$$\begin{aligned}& p(\kappa\lambda_1) = r_1 \exp(i\phi_1),\ a_1 = r_a \exp(i\xi),\ W_{21} = r_W \exp(i\psi), \\ & \delta = 2(q-1)\phi_1 + 2\xi + \psi,\end{aligned}$$

and insert into (12). Then (12) is satisfied if

$$2r_1^{2q} r_a^2(1 + r_W \cos(\delta + 2\phi_1)) < 2r_1^{2q-2} r_a^2(1 + r_W \cos(\delta)),$$

or

$$r_1^2 + r_W \cos(\delta)(r_1^2 \cos(2\phi_1) - 1) - r_1^2 r_W \sin(\delta)\sin(2\phi_1) < 1. \tag{13}$$

The left hand side of (13) is bounded by $r_1^2 + r_W + 2r_1^2 r_W$. Hence, the asymptotic convergence with one dominant vector will be monotone if

$$|p(\kappa\lambda_1)|^2 + |W_{21}| < 1.$$

Note that oscillations will never arise for normal matrices. If all eigenvectors v_j are orthogonal then $W_{jk} = 0,\ j \neq k$ in (10). The farther W is from a normal matrix, the greater the risk is of an oscillatory behavior. For normal matrices, we have simply

$$\|r^{(q)}\|^2 = \|p(\kappa AT)^q r^{(0)}\| = \|\sum_{j=1}^{N} a_j v_j p(\kappa\lambda_j)^q\|^2 = \sum_{j=1}^{N} |a_j|^2 |p(\kappa\lambda_j)|^{2q}, \qquad (14)$$

and assuming κ is stable, all factors $|p(\kappa\lambda_j)|^{2q}$ will decrease in each step.

Suppose that we have the solution at step n and want to select a κ so that $\|r^{(n+1)}\|$ is minimized. From (4) we derive

$$\|r^{(n+1)}\|^2 = (p(\kappa AT)r^{(n)})^T(p(\kappa AT)r^{(n)}) \equiv \rho(\kappa)\,. \qquad (15)$$

The polynomial $\rho(\kappa)$ is of degree $2m$ with coefficients γ_k that can be computed as

$$\begin{aligned} \gamma_k &= \textstyle\sum_{i+j=k} \beta_i\beta_j S_{ij},\ k = 0, \ldots, 2m,\ 0 \le i,\ j \le m, \\ S_{ij} &= ((AT)^i r^{(n)})^T((AT)^j r^{(n)}). \end{aligned} \qquad (16)$$

The κ that minimizes $\rho(\kappa)$, normally with the convention that $\kappa > 0$, corresponding to stepping forward in time, is then used to compute $u^{(n+1)}$ in (3).

A necessary and sufficient condition for convergence of $r^{(n)}$ is that $\rho(\kappa)$ has no global minimum at $\kappa = 0$. If that is the case then $\|r^{(n+1)}\| = \|r^{(n)}\|$ and no progress is made in the step from n to $n+1$ by minimizing $\rho(\kappa)$ and the iteration stagnates. Otherwise, there always exists a time step (though not necessarily positive) which will reduce the residual norm. A sufficient condition for the minimum to be different from 0 is that the first-order coefficient γ_1 of $\rho(\kappa)$ is different from 0, ensuring that $\rho'(\kappa)$ only has non-zero roots.

It follows from (16) that

$$\begin{aligned} \gamma_1 &= -2(r^{(n)})^T \mathrm{symm}(AT) r^{(n)}, \\ \mathrm{symm}(AT) &= 0.5((AT)^T + (AT)). \end{aligned} \qquad (17)$$

If the symmetric part $\mathrm{symm}(AT)$ of AT is definite in (17), then $\gamma_1 \neq 0$. Since $\rho'(0) = \gamma_1$, there is a minimum for $\kappa < 0$ if $\mathrm{symm}(AT)$ is negative definite and for $\kappa > 0$ if $\mathrm{symm}(AT)$ is positive definite. An example when $\mathrm{symm}(AT) = 0$ is a discretization by centered differences of a first order differential equation on a Cartesian grid with periodic boundary conditions.

If we return to the asymptotic phase with four dominant eigenvalues as in (9), then γ_1 is approximated by

$$\begin{aligned}\gamma_1 = -4\Re \quad & (\lambda_1(|a_1|^2 + a_1^2 W_{21} + a_1 \overline{a}_2 W_{31} + a_1 a_2 W_{41}) + \\ & +\lambda_2(|a_2|^2 + a_2^2 W_{43} + \overline{a}_1 a_2 \overline{W}_{31} + a_1 a_2 W_{41}))\,.\end{aligned} \tag{18}$$

Suppose that v_1 and $\overline{v}_1$ are pairwise orthogonal to v_2 and $\overline{v}_2$ so that $W_{31} = W_{41} = 0$. Then

$$\gamma_1 = -4\Re(\lambda_1(|a_1|^2 + a_1^2 W_{21}) + \lambda_2(|a_2|^2 + a_2^2 W_{43}))\,. \tag{19}$$

One can show [9] that if $\tan^{-1}(|\Re(\lambda_j)/\Im(\lambda_j)|) \geq \sin^{-1}(|v_j^H \overline{v}_j|)$ for $j = 1, 2$, then γ_1 cannot vanish, e.g. if $\Re(\lambda_j)$ is sufficiently large compared to $\Im(\lambda_j)$. Otherwise, combinations of a_1, a_2, and W are possible such that $\gamma_1 = 0$ in both (18) and (19). We have found in (10) that there are situations when $r^{(n)}$ will increase and $\rho(\kappa) \geq 1$. The conclusion is that the global minimum of $\rho(\kappa)$ can be at $\kappa = 0$ or a κ very close to 0. Such values have to be avoided since they will lead to stagnation or very slow convergence.

Based on the analysis, the algorithm is as follows. Compute the global minimizer κ^* of $\rho(\kappa)$. The κ^*-value is chosen if $\rho(\kappa^*) < \|r^{(n)}\|^2$ and $\kappa^* \in \mathcal{I}$, for some interval $\mathcal{I}$ not including the origin. An alternative is to accept κ^* if $\rho(\kappa^*) \leq C\|r^{(n)}\|^2$, where some $C < 1$ is a desired performance factor or a moving average of the past convergence rate. If the convergence is too slow according to the first or second criterion, a “safe” CFL number κ_s (one which is stable but not necessarily optimal) is used instead. Optimization is then turned off and not resumed until the residual norm has dropped below the point where the safe value took over. A flow chart with the second alternative is shown in Fig. 2. A flag variable called *optimization* determines whether to search for an optimal κ or just use κ_s. The variable *endvalue* holds the value the residual norm has to pass below before optimization of κ is resumed. The following proposition guarantees the convergence.

Proposition 1 *If the safe CFL number κ_s has the property that*

$$\|r^{(n+q)}\| < \|r^{(n)}\|$$

for all n and some $q \geq 1$, then as n grows the algorithm generates a subsequence of $r^{(n)}$ such that $\|r^{(n)}\| \to 0$.

Proof Either *optimization*=ON at the low 'Iterate RK' in Fig. 2 and then

$$\|r^{(n+1)}\|^2 = \rho(\kappa^*) < \|r^{(n)}\|^2,$$

after 'Iterate RK' or *optimization*=OFF for q time steps with $\kappa = \kappa_s$ and then

$$\|r^{(n+q)}\|^2 < \|r^{(n)}\|^2.$$

Hence, there is a strictly decreasing subsequence of $r^{(n)}$ bounded below by 0. ■

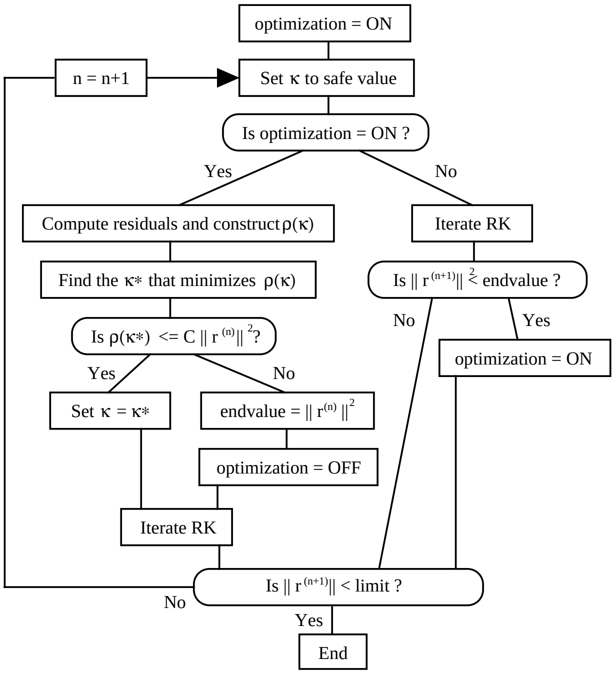

optimization = ON
n = n+1
Set κ to safe value
Is optimization = ON ?
Yes
No
Compute residuals and construct ρ(κ)
Iterate RK
Find the κ* that minimizes ρ(κ)
Is || r $^{(n+1)}$|| ^{2} < endvalue ?
Is ρ(κ*) <= C || r $^{(n)}$|| ^{2}?
No
Yes
optimization = ON
Yes
No
Set κ = κ*
endvalue = || r$^{(n)}$ ||^2
optimization = OFF
Iterate RK
Is || r$^{(n+1)}$|| < limit ?
No
Yes
End

Figure 2: *Optimization algorithm*

In order to determine the coefficients of the polynomial $\rho(\kappa)$ we need all scalar products $S_{ij},\ i,j = 0,\ 1, \dots, m$, in (16). Taking into account that $S_{ij} = S_{ji}$, this requires approximately $N(m^2 + 3m + 2)$ arithmetic operations. The evaluation cost of $\rho(\kappa)$ is then $\mathcal{O}(m)$. The polynomial is evaluated on a discrete set of points and then the search was refined around the discrete minumum using the golden section (GS) method [14]. The cost of the optimization process is typically negligible compared to the rest. If we are using an m-stage method and, for example, start by examining $\rho(\kappa)$ at 20 equidistant points and then apply GS to shrink the resulting uncertainty interval down to 1%, this will cost us roughly $120m + 40$ floating-point operations. For comparison, just evaluating the Euler equation operators costs about 100 operations per grid cell in 3D—and for the Navier-Stokes equations this figure is about 250.

For discretized linear, hyperbolic PDEs, experiments in [9] indicate that the result of local κ optimization is similar to or better than anything that can be achieved with a constant κ.

For nonlinear systems the residual at $n+1$ is approximated by a linearization around $r^{(n)}$, resulting in (cf. (4))

$$r^{(n+1)} \approx r^{(n)} + \sum_{j=1}^{m} \beta_j \kappa^j (JT)^j r^{(n)}. \tag{20}$$

The approximation has the same form as (4) and the algorithm is applicable. Although the Jacobian J could be obtained analytically, a simpler approach is to approximate it as a linearization and compute $(JT)^j r^{(n)}$ recursively using

$$Jv \approx (r(u^{(n)} + \sigma v) - r^{(n)})/\sigma \,, \tag{21}$$

where σ is some small number [1] and v is arbitrary.

The linearized and optimized RK (20) requires one more residual evaluation than the ordinary RK, plus a few scalar products. While the latter are often negligible for larger problems, the former unavoidably increases the workload of an m-stage RK method by a factor of $(m+1)/m$. This is not the case for linear problems. Efficient optimization should hopefully compensate for that for nonlinear equations, at least when compared to the result of picking a safe parameter value. An advantage of the complete algorithm in Fig. 2 is of course that this extra cost does not occur during the steps in which optimization is turned off. A less important drawback, given the improved capacity of present computers, is the need to store all $(JT)^j r^{(n)}, \;\; j = 0, \ldots, m$, vectors in memory.

3 Discrete adaptation

Two parameters that have not yet been discussed are the number of RK stages m and, when multigrid iteration is involved, the number of multigrid levels l. As usual we are interested in achieving the greatest residual reduction per unit of computational effort. Normally, the convergence rate per iteration increases with more stages and grid levels, but so does the workload.

The number of iterations n_{iter} to reduce the initial residual $\|r^{(0)}\|$ below ε is

$$\|r^{(n_{iter})}\| = \zeta^{n_{iter}} \|r^{(0)}\| \leq \varepsilon, \; \zeta = \|r^{(j+1)}\| / \|r^{(j)}\|,$$

assuming ζ to be constant. If w_{iter} is the computational work in each iteration, then the total work w_{tot} for a convergent solution is

$$w_{tot} = n_{iter} w_{iter} = \log(\|r^{(0)}\|/\varepsilon) w_{iter} / \log(1/\zeta). \tag{22}$$

Thus, we want to maximize θ, the quotient between the successive residuals per computational work unit

$$\theta = \log(\|r^{(j)}\|/\|r^{(j+1)}\|)/w_{iter}. \tag{23}$$

The damping factor ζ_0 for eigenvalues λ_j close to the origin with $\Re\lambda_j > 0$ is approximately (see (4))

$$\zeta_0 \approx 1 - \kappa \Re \lambda_j. \tag{24}$$

Since η_{opt} in (8) fulfills $\eta_{opt} \geq \zeta_0$ and there is a possibility to reduce η_{opt} by increasing κ in (24), RK methods with large stability regions $\mathcal{S}$ allowing large κ are preferred. The same conclusion holds for eigenvalues with $\Re\lambda_j = 0$ (or very small). This is discussed in [13]. The work per iteration is proportional to $m+1$ for an m stage method applied to a nonlinear problem. For the methods in Table 1 and Fig. 1, the maximum κ seems to behave like const $\cdot\, m$. Suppose that the small eigenvalues bound the convergence rate. Then

$$\theta \approx \log(1/(1-\kappa \min_j \Re\lambda_j))/w_{iter} \approx c_{RK} m \min_j \Re\lambda_j/(m+1), \tag{25}$$

for some constant c_{RK}. In (25), θ grows slowly with m making RK methods with more stages more attractive.

The multigrid acceleration of the iteration is implemented as a V-cycle in a standard manner, see e.g. [6] or [13], with smoothing at every level using the three-stage RK scheme in Table 1. It is shown in [13] that for the smallest eigenvalues as in (24) we have

$$\zeta_0 \approx 1 - \kappa(2^l - 1)\Re\lambda_j. \tag{26}$$

The work per full V-cycle in 2D with a fixed m in the RK iteration is proportional to

$$1 + \sum_{j=2}^{l} 4^{1-j} = 4(1-4^{-l})/3 \approx 4/3.$$

Then as in (25)

$$\theta \approx c_{mg}(2^l - 1)\min_j \Re\lambda_j \tag{27}$$

for some constant c_{mg}. It is clearly advantageous to use many grid levels.

The algorithm for the selection of a discrete parameter is as follows. Assume that we have k different parameter values $\phi_1, \ldots, \phi_k$, and k associated convergence factor estimates $\theta_1, \ldots, \theta_k$. We assign "quotas" q_i to the parameter values, chosen in such a way that

$$q_1 \geq q_2 \geq q_3 \geq \ldots \geq q_k > 0, \quad \text{and} \quad \sum_{i=1}^{k} q_i = 1\,. \tag{28}$$

We will also need a set of counter variables $c_1, \ldots, c_k$, where c_i keeps track of how many times we have used the i:th best parameter value ϕ_i. After each iterative step, we first add 1 to the c_i that corresponds to the last used parameter value. Then its performance value g_i is updated by

$$g_i^n = \xi\theta_i^n + (1-\xi)g_i^{n-1}, \tag{29}$$

where ξ satisfies $0 < \xi < 1$ and blends the past values with the most recent measured θ_i^n in (23). For $j \neq i$, we take $g_j^n = g_j^{n-1}$. If g_i is updated in every step and $g_i^1 = \theta_i^1$, then

$$g_i^n = (1-\xi)^{n-1}\theta_j^1 + \sum_{k=2}^{n}(1-\xi)^{n-k}\xi\theta_i^k.$$

We find that old measurements are successively losing their influence. Then we sort our ϕ_j by their g_j^n and obtain a list of indices from best to worst. Finally we check, for $j = 2, \ldots, k$, in turn whether $c_j/n < q_j$. If so, we stop checking and use ϕ_j for our next iteration. If no comparison holds true, we use ϕ_1.

The result is that we test and update the parameter value on average a fraction q_j of all steps. The remaining time is devoted to the best estimate g_1. After a large number of steps $n \gg k$ with almost constant g_j^n, we have obtained an average performance

$$\bar{\theta} = \frac{1}{n}\sum_{i=0}^{n-1}\theta^{(i)} \approx \frac{1}{n}\sum_{j=1}^{k} nq_j g_j = g_1 - \sum_{j=2}^{k} q_j(g_1 - g_j) \leq g_1\,. \tag{30}$$

The performance of all parameter values needs to be updated regularly in a neighborhood of the currently best estimate. To measure the objective we have to use the corresponding parameter value, even if it is sub-optimal according to our current estimates. Thus, we can never expect fully optimal performance, see (30). The process becomes a trade-off between maintaining up-to-date performance estimates and using the best parameter value as often as possible. Very poor values leading to divergent iterations with $g_J > 1$ will be discovered and can be avoided by allowing $q_j,\ j = 1, \ldots, k$, to change and letting $q_J \to 0$.

4 Nonlinear examples

The methods proposed in the previous sections are here applied to the steady state solution of inviscid and viscous flow problems.

4.1 Continuous optimization

The steady state solutions of the Euler equations of compressible, inviscid flow and the Navier-Stokes equations of compressible, viscous flow are computed in a channel with a bump. For a definition and discussion of these equations see e.g. [8]. The upper and lower walls are solid and inflow boundary conditions are specified to the left and outflow conditions to the right. The computational domain is covered by a structured grid. In Fig. 3, two grids at a coarse multigrid level are depicted. The initial solution $u^{(0)}$ is freestream flow in the channel. The resolution on the finest grid is 129 by 65 points.

The equations are discretized in space according to Jameson [10], [11]. In the examples, the flow is subsonic with a Mach number $M = 0.5$ and the Reynolds number in the Navier-Stokes problem is 10^4. All numerical experiments have been run on a Sun SPARCstation Ultra.

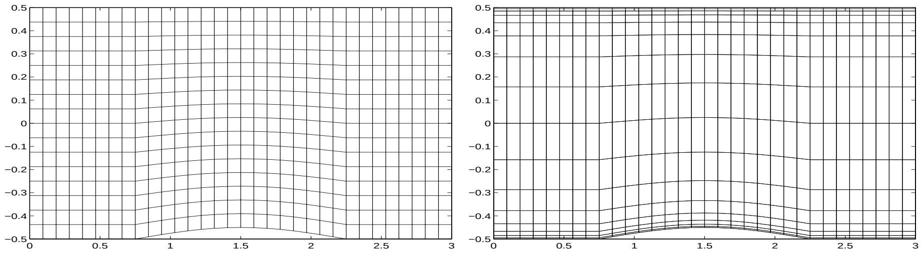
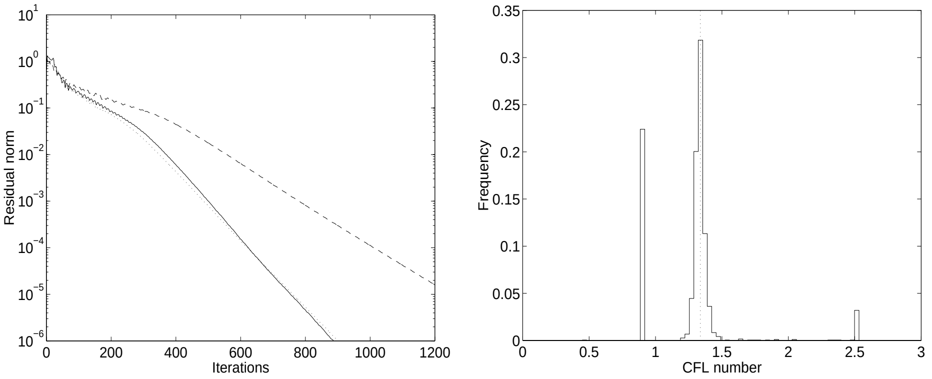


Figure 3: *Coarse computational grids for the Euler (left) and Navier-Stokes equations (right).*

A typical residual curve for the Euler equations solved by the three-stage RK in Table 1 is shown in Fig. 4, where optimization is compared to using a safe CFL value κ_s and the "best choice" κ^*—determined by repeatedly running the iteration process from beginning to end, gradually narrowing down the search interval.

In these experiments, the first strategy mentioned in Sect. 2 was used. The optimal values were selected in the interval $\mathcal{I} = [0.2, 2.5]$ and $\kappa_s = 0.9$.



(a) *Residual curves. Local optimization (solid), globally optimal κ (dotted) and safe value (dashed).*

(b) *Distribution of optimized CFL numbers. The dotted line shows the experimentally determined global optimum.*

Figure 4: *The Euler equations on one grid.*

Fig. 4(a) shows a plot of the residual norms in each iteration step. After a difficult start with many local increases in the residual norm, the optimized method proceeds at the same convergence rate as the globally optimal curve. As expected, Fig. 4(b) shows the locally optimal CFL values distributed in a neighborhood of κ^*. The peak at 0.9 is due to the safe value being picked often during the start, and the part around 2.5 is an "overshoot" that often occurs when optimization is resumed and the optimizer tries to compensate for the period of low CFL values.

In Fig. 5 we can see the result of two solution processes for the Euler and Navier-Stokes equations, with and without multigrid acceleration, after the optimization cost has been included. The convergence is better than with no optimization at all, but it does not quite match the global optimum asymptotically.

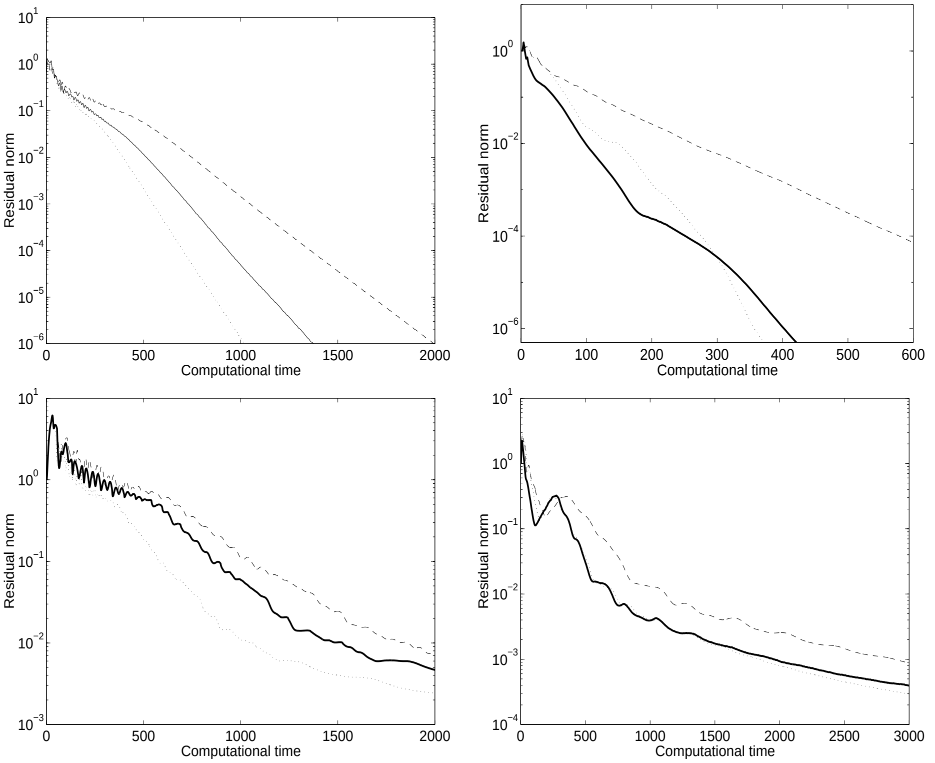


Figure 5: *Optimized (solid), safe (dashed) and best (dotted) CFL numbers for the Euler equations (upper) and the Navier-Stokes equations (lower) using single grid iteration (left) and three level multigrid iteration (right).*

4.2 Discrete optimization

We now repeat the experiments with the Euler and Navier-Stokes equations with discrete optimization as in Sect. 3 over the number of RK stages and multigrid levels. For a given number of stages and grid levels, our previously derived method of continuous CFL optimization is used. The behavior of the residual on the fine grid level determines when optimization is turned on and off for all orders and grid levels. Here the forgetting factor in (29) is $1-\xi = 0.75$ and the quotas are $q_2 = 1/6,\ q_3 = q_4 = 1/7$. The result is shown in Figs. 6 and 7.

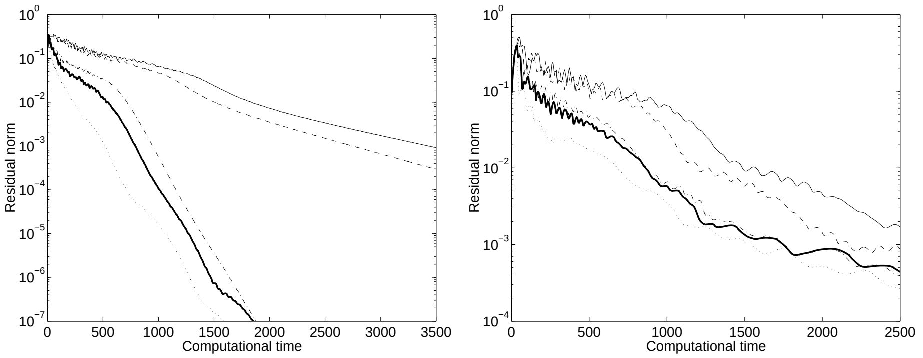


Figure 6: *Euler (left) and Navier-Stokes (right) equations solved with optimized RK stages (thick) compared to 2 (..), 3 (-.), 4 (–) and 5 (solid)*

The general trend here is that more grid levels result in faster convergence per unit of work in accordance with (27). In contrast, more RK stages seem to just increase the workload on a per-step basis. This behavior is presumably dependent on the specific problem, since four- or five-stage RK iteration tends to be the recommended choice in the literture. The experiments are in disagreement with the analysis in (25), but the relation between the maximum κ and m is oversimplified there (cf. Fig. 1). In our case, RK(2) is unexpectedly the most efficient method and the optimization algorithm discovers that.

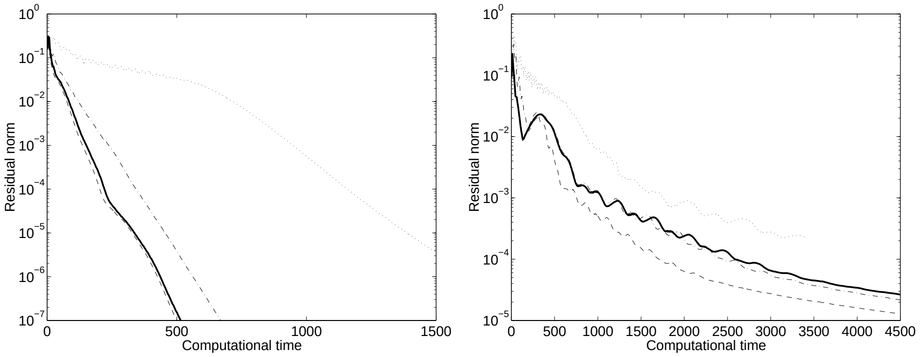


Figure 7: *Euler (left) and Navier-Stokes (right) equations solved with optimized multigrid levels (thick) compared to 1 (..), 2 (-.) and 3 (–)*

5 Conclusions

We have developed a way of adapting the Runge-Kutta method with respect to the CFL number for nonlinear problems, at a relatively low extra cost. However, the real benefit lies in not having to manually select an efficient CFL value. One that works will be enough and the optimization algorithm will achieve performance close to the global optimum anyway measured in number of iterations. This is important, as a safe guess may be possible to determine analytically from studying an approximating problem, but the real optimum can usually only be selected by repeated testing, i.e. solving the problem over and over.

We have shown the CFL optimization for RK developed in Sect. 2 to be a stable platform from where to add further optimization with respect to the discrete parameters: number of stages and number of multigrid levels. To eliminate a discrete parameter, we have applied a simple algorithm on top of the optimized RK. Performance is close to optimal.

References

[1] P. N. Brown and Y. Saad. Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Stat. Comput.*, 11(3):450–481, 1990.

[2] C. Chiu and D. A. Kopriva. An optimal Runge-Kutta method for steady-state solutions of hyperbolic systems. *SIAM J. Numer. Anal.*, 29(2):425–438, 1992.

[3] R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *IBM J.*, 11:215–234, 1967.

[4] G. H. Golub and R. Kannan. Convergence of a two-stage Richardson process for nonlinear equations. *BIT*, 26(2):209–216, 1986.

[5] A. Greenbaum. *Iteration Methods for Solving Linear Systems.* SIAM, Philadelphia, PA, 1997.

[6] W. Hackbusch. *Multigrid Methods and Applications.* Springer-Verlag, Berlin, Heidelberg, 1985.

[7] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems*, 2nd ed., Springer-Verlag, Berlin, 1996.

[8] C. Hirsch. *Numerical Computation of Internal and External Flows*, Vols 1 and 2, Wiley, New York, 1990.

[9] K. Hörnell. *Runge-Kutta Time Step Selection for Flow Problems.* PhD thesis, Dept. of Scientific Computing, Uppsala University, Uppsala, Sweden, 1999.

[10] A. Jameson. Solution of the Euler equations for two-dimensional transonic flow by a multigrid method. *Appl. Math. Comput.*, 13:327–355, 1983.

[11] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the Euler equations by the finite volume method using Runge-Kutta time-stepping schemes. *AIAA paper* 81-1259, 1981.

[12] B. van Leer, W.-T. Lee, P. L. Roe, K. G. Powell, and C.-H Tai. Design of optimally smoothing multistage schemes for the Euler equations. *Comm. Appl. Numer. Methods*, 8:761–769, 1992.

[13] P. Lötstedt. Improved convergence to the steady state of the euler equations by enhanced wave propagation. *J. Comput. Phys.*, 114(1):34–44, 1994.

[14] D. G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, Reading, Mass., 1973.

[15] J. F. Lynn and B. van Leer. Multi-stage schemes for the Euler and Navier-Stokes equations with optimal smoothing. *AIAA paper* 93-3355, 1993.

[16] L. Martinelli and A. Jameson. Validation of a multigrid method for the Reynolds averaged equations. *AIAA paper* 88-0414, 1988.

[17] D. J. Mavriplis and A. Jameson. Multigrid solution of the the Navier-Stokes equations on triangular meshes. *AIAA J.*, 28(8):1415–1425, 1990.

[18] B. Müller. Linear stability condition for explicit Runge-Kutta methods to solve the compressible Navier-Stokes equations. *Math. Meth. Appl. Sci.*, 12:139–151, 1990.

[19] N. M. Nachtigal, L. Reichel, and L. N. Trefethen. A hybrid GMRES algorithm for nonsymmetric linear systems. *SIAM J. Matrix Anal. Appl.*, 13(3):796–825, July 1992.

[20] G. Opfer. Richardson's iteration for nonsymmetric matrices. *Lin. Alg. Appl.*, 58:343–361, 1984.

[21] P. E. Saylor and D. C. Smolarski. Implementation of an adaptive algorithm for Richardson's method. *Lin. Alg. Appl.*, 154/156:515–646, 1991.

[22] R. C. Swanson and E. Turkel. A multistage time-stepping scheme for the Navier-Stokes equations. *AIAA paper* 85-0035, 1985.

[23] R. C. Swanson and R. Radespiel. Cell centered and cell vertex multigrid schemes for the Navier-Stokes equations. *AIAA J.*, 29(5):697–703, 1991.

[24] S. C. Reddy and L. N. Trefethen. Stability of the method of lines. *Numer. Math.*, 62:235–267, 1992.