Parameter-free Krylov subspace iteration methods for the Euler and Navier-Stokes equations

by

Karl Hörnell

Report No. 184/1996

UPPSALA UNIVERSITET
Institutionen för
teknisk databehandling

UPPSALA UNIVERSITY
Department of
Scientific Computing

Parameter-free Krylov subspace iteration methods for the Euler and Navier-Stokes equations *

Karl Hörnell †

September 26, 1996

Abstract

This paper presents a way of selecting optimal values for the CFL parameter of the Runge-Kutta method. Initial and asymptotic convergence behavior are analyzed for linear problems, with both single and multi-grid iterations, and the optimized RK is combined with other Krylov subspace methods to eliminate the fixed coefficients as well. Finally, the resulting methods are tested and evaluated for a few test cases with the real Euler and Navier-Stokes equations.

*This work was supported by a two-year scholarship from the Swedish Institute of Applied Mathematics, Chalmers Teknikpark, S-412 88 Göteborg, Sweden.

†Dept. of Scientific Computing, Uppsala University, S-751 04 Uppsala, Sweden

1 Introduction

Discretization of certain partial differential equations, such as the Euler or Navier-Stokes equations, lead to large nonlinear systems of equations, which can be linearized and expressed as

$$Ax = b$$

where A is a $\nu \times \nu$ matrix with eigenvalues λ_j, $j = 1, \ldots, \nu$ and b is a vector of length ν.

A popular way of solving these equations iteratively is to consider x the steady-state solution $u(\infty)$ of the ordinary differential equation

$$\frac{du}{dt} = r$$

with r being the residual of an approximate solution u, $r = b - Au$, and integrate numerically over time until the residual vanishes.

This approach makes it natural to analyze the results of the solution process as a course of events in time. Components of the solution, residual or error can be thought of as traveling across the computational domain at certain "wave speeds". Rapid convergence towards the steady state requires that the parts not belonging to the solution leave the area as quickly as possible, and that high-frequency modes resulting from the problem discretization become "damped".

If a Runge-Kutta method is used for the integration, we get an iterative method that depends on the size of the time step — normally expressed in terms of the ratio of the time step Δt and the spatial discretization Δx, the so-called CFL number κ. (This is a special case of the Krylov subspace methods, which form a new approximate solution from u and the set $[r, Ar, A^2r, \ldots, A^{m-1}r]$.) RK methods have some additional parameters which will, initially, be regarded as fixed at standard values for similar problems.

As was shown by, for example, Manteuffel in [4], determining optimal parameters for long-term convergence of this type of iteration leads to a mini-max problem of making the largest point value of a polynomial p on any of the eigenvalues λ_j of A as small as possible.

$$\gamma_{opt} = \min_{\kappa} \max_{j} |p(\lambda_j, \kappa)|$$

Given some assumptions about the properties of the system matrix A, it was also shown that the only eigenvalues which have to be considered in such an optimization are the ones spanning the convex hull of the whole set $\{\lambda_j\}$.

In section 2, however, it will be demonstrated that if a particular short-term optimization is used instead, there is no need to compute or estimate γ_{opt} or the corresponding κ. They will both eventually emerge as by-products and no particular assumptions about the structure of A need to be made. Under some circumstances, the results also apply, and in fact give the optimized method further advantages, when the iterations are carried out on multiple grids.

We will also attempt to combine this optimized RK with other Krylov methods in order to obtain even better convergence properties and eliminate the need for the remaining RK parameters.

Previous attempts at selecting "optimal" CFL numbers have mainly involved some form of search algorithm for finding or rejecting values based on results from previous iteration steps. One example is the automatic time stepping described in [11]. It searches among a discrete set of CFL numbers with exponentially increasing distance and eventually stops at the largest one within the stability region. However, the resulting CFL number is not necessarily optimal in any sense, and although a parameter is formally eliminated, the outcome will depend, to some extent, on the initial guess. The optimization process described here has no such drawbacks and can be expected to consistently perform better.

It should be mentioned that some insight can often be gained by modifying and simplifying the original problem in such a way that it lends itself to Fourier transformation and analysis (e.g. assuming periodic boundary conditions). But this is usually not reliable enough to base any optimal parameter estimation upon.

In addition, the fixed coefficients of an RK method need not be particularly well suited for the problem to which it is applied. Then the time step will have to be made very small in order to maintain stability — resulting in unnecessarily long computational times. Therefore an ideal optimized iteration method would have the ability to modify *all* its parameters and coefficients to improve the speed of convergence. A common one is the Generalized Minimum RESidual [1], which will be discussed in the following sections. It does however have some disadvantages that motivate the construction of alternative methods.

Asymptotically, nonlinear problems will resemble linear ones, with the matrix A corresponding to the Jacobian of the system. This is also the basis for the linearization that will let us extend these iteration methods back to the original class of nonlinear problems. This is the main objective of this research.

The formulas and results are presented in section 3. It turns out the optimized RK method works well for the Euler and Navier-Stokes equations, even on multiple computational grids, and that the hybrid methods developed in section 2 are more robust alternatives to parameter-free methods like GMRES.

2 Optimization of Krylov Methods

2.1 CFL optimization for RK

Let $u^{(n)}$ be the numerical solution to $du/dt = r(u)$ at time step n. The general m-stage Runge-Kutta iteration can then be described as follows:

$$
\begin{aligned}
u^0 &= u^{(n)} \\
u^1 &= u^0 + a_1 \Delta t\, r(u^0) \\
&\vdots \\
u^m &= u^0 + a_m \Delta t\, r(u^{m-1}) \\
u^{(n+1)} &= u^m \,.
\end{aligned}
$$

At the nth step, the residual produced by an m-stage RK iteration for solving the linear problem $Au = b$ will be

$$r^{(n+1)} = r^{(n)} + \sum_{j=1}^{m} \alpha_j \kappa^j A^j r^{(n)} = p(A) r^{(n)} \,, \tag{1}$$

where $p(x) = 1 + \alpha_1 \kappa x + \ldots + \alpha_m \kappa^m x^m$. The square of the ℓ_2-norm of the residual is then

$$\|r^{(n+1)}\|^2 = (p(A) r^{(n)})^H (p(A) r^{(n)}) = \rho(\kappa) \,. \tag{2}$$

The polynomial $\rho(\kappa)$ is consequently of degree 2m, non-negative and will have a global minimum in the region of interest (usually the half-line $\kappa > 0$ to ensure the proper direction of the wave speed — it will be assumed from here on, for simplicity, that such a region exists and corresponds to positive values of κ, otherwise this can be obtained by simply switching the signs of A and b). Its coefficients β_k can be computed as

$$\beta_k = \sum_{i+j=k,\, 0 \le i,j \le m} \alpha_i \alpha_j (A^i r^{(n)})^H (A^j r^{(n)}) \,, \qquad k = 0, 1, \ldots, m \,. \tag{3}$$

The κ that minimizes $\rho(\kappa)$ can be found with fairly simple and well-known numerical optimization algorithms. Sometimes it is more convenient to compare the values of $\rho(\kappa)$ at stationary points (the roots of $\rho'(\kappa)$, computed e.g. with the *C02AGF NAG* Fortran Library routine, or — for small-scale numerical experiments — the MATLAB function *roots()* and interval limits. That κ is then used to obtain

$$u^{(n+1)} = u^{(n)} + \sum_{j=1}^{m} \alpha_j (\kappa)^j A^{j-1} r^{(n)} \,. \tag{4}$$

The complete algorithm can be expressed as

$v_0 = u^{(n)}$
for $i = 1 \ldots m$
    $v_i = r(v_{i-1})$
endfor
for $i = 0 \ldots m$

for $j = 0 \ldots m$

$\beta_{i+j} = \beta_{i+j} + < v_i, v_j > \alpha_i \alpha_j$

endfor

endfor

$\kappa = findMinimum(p(\kappa))$

for $i = 1 \ldots m$

$v_0 = v_0 + \kappa \alpha_i v_i$

endfor

$u^{(n+1)} = v_0$.

Although this CFL optimization requires the explicit evaluation of $A^m r^{(n)}$ which is not needed in (4), the formula (1) gives $r^{(n+1)}$ as a combination of

$$[r^{(n)}, Ar^{(n)}, \ldots, A^m r^{(n)}], \tag{5}$$

so the total number of new Krylov vectors to compute (from the second iteration and on) will be the same as for a non-optimized RK.

For certain types of system matrices A it is possible to draw some conclusions about the long-term behavior of κ. Consider the case with a normal matrix A. Let the coefficients of each eigenvector in $r^{(n)}$ be a_j. Then the square of the residual norm in step n+1 is

$$\|r^{(n+1)}\|^2 = \sum_{j=1}^{\nu} |a_j p(\lambda_j)|^2 . \tag{6}$$

If we assume, for simplicity, that κ is constant from that step until n+q, the residual norm squared will then be

$$\|r^{(n+q)}\|^2 = \sum_{j=1}^{\nu} |a_j (p(\lambda_j))^q|^2 . \tag{7}$$

The normal way of analyzing the stability of an RK method applied to a problem is in terms of the *stability domain* of the method itself. That is the set of $z \in \mathcal{C}$ for which the first iteration step to solve the model problem

$$\begin{cases} y' = zy \\ y(0) = 1 \end{cases} \tag{8}$$

results in $|y^{(1)}| \leq 1$. Examples of stability domains for RK(2), RK(3) and RK(4) with the α_j coefficients used in this report are shown in figure 1. We have chosen the values in table 1.

For the three- and four-stage methods we have used standard RK coefficients for flow problems, with the convention that α_1 is normalized.

However, for a better comparison with other Krylov methods that might not have constant coefficients, we will begin with a slightly different perspective.

Stages	α_1	α_2	α_3	α_4
2	-3/5	9/25		
3	-1	3/5	-9/25	
4	-1	1/2	-1/15	1/60

Table 1: *RK coefficients*

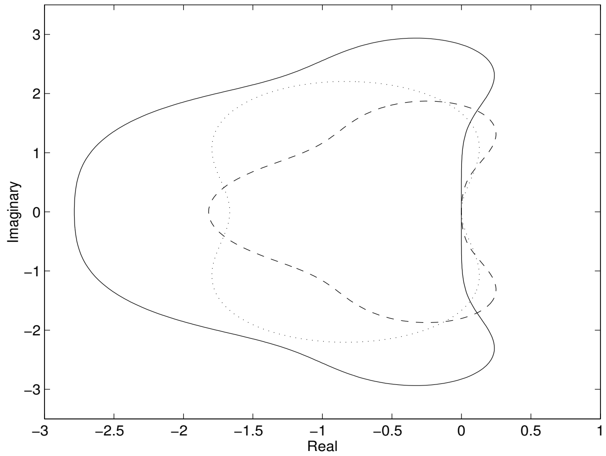


Figure 1: *Stability domains for second-order (dotted), third-order (dashed) and fourth-order (solid) RK methods with standard coefficients.*

Convergence will be discussed in terms of the eigenvalues of the iteration matrix $p(A)$.

If κ lies within the stability region, all $|p(\lambda_j)| < 1$ and the value of $\|r^{(n+q)}\|^2$ as q increases will eventually be dominated by the largest of $|p(\lambda_j)|$. Thus the asymptotic convergence factor as the number of iterations goes to infinity is

$$\max_j |p(\lambda_j)| \,. \tag{9}$$

The optimal convergence factor with a *constant* κ, as the number of iterations goes to infinity, is then given by

$$\gamma_{opt} = \min_\kappa \max_j |p(\lambda_j, \kappa)| \,. \tag{10}$$

Assuming not all eigenvalues are identical, this optimum will be found at a point where two or more curves $|p(\lambda_j, \kappa)|$ intersect, as illustrated in fig 2.

In the particular case when all λ_j can be written as $c_j(a \pm ib)$, $0 < c_j < \infty$, i.e. they all have the same proportion of real and imaginary parts — possibly with different signs on the latter — all eigenvalues will follow the same curve $p(a+ib,t)$, $t \in [0,\infty)$ in the complex plane (or $p(a-ib,t)$, which is just mirrored in the real axis). If $|p(a+ib,t)|$ is unimodal at least while $p(a+ib,t)$ lies inside

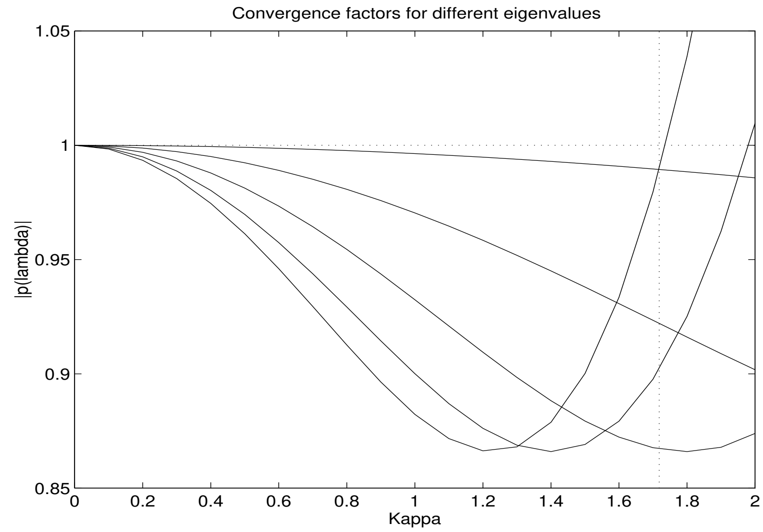


Figure 2: *Example of convergence curves for the eigenvectors of a small matrix. The stability region is where all curves are below the horizontal line. The vertical line marks the κ corresponding to the optimal convergence factor.*

the unit circle, then the optimal constant CFL number κ^* will be given by

$$|p(\lambda_{min}, \kappa^*)| = |p(\lambda_{max}, \kappa^*)|, \quad |p(\lambda_{max}, \kappa^*)| < 1\,, \tag{11}$$

where λ_{min} and λ_{max} are the eigenvalues with the smallest and largest absolute values respectively. One such example is the two-stage RK iteration

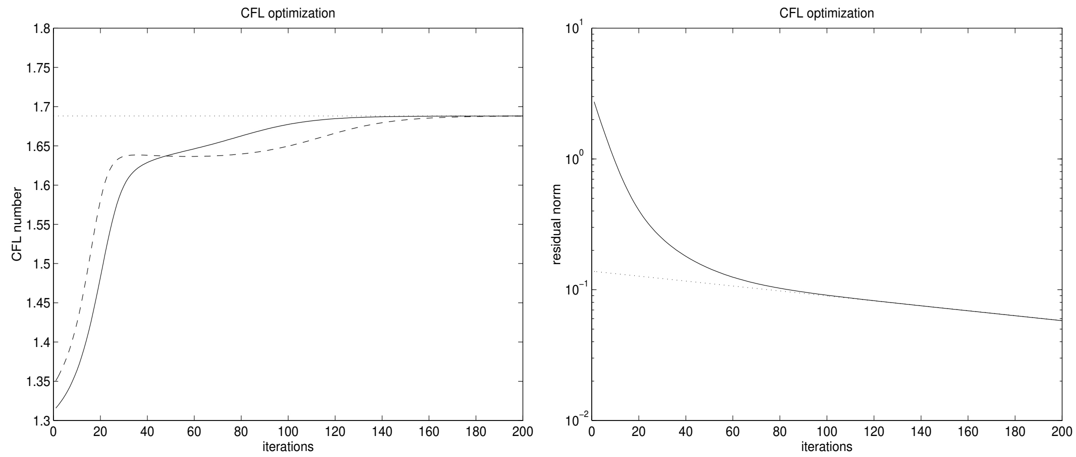
$$r^{(n+1)} = r^{(n)} - 0.6\kappa A r^{(n)} + 0.36\kappa^2 A^2 r^{(n)} \tag{12}$$

applied to the model problem

$$\begin{cases} u_x = 0\,, & 0 \leq x \leq 1 \\ u(0) = 0\,, u(1) = 0 \end{cases} \tag{13}$$

discretized with a system matrix corresponding to a one-dimensional centered difference operator. We will use the convention, from here on, that such system matrices have $1/h$ factored out, to be included in κ. This is to make the matrix elements of similar (unity) order and independent of problem size. In the current case, A consequently has 0.5 along the first upper diagonal and -0.5 along the first lower diagonal. (Note that this PDE is not a well-posed problem, since there is an over-specification of boundary values. The extra condition will however match the exact solution, $u(x) = 0$, and has been added to make the system matrix easier to analyze.) Then the eigenvalues are

$$\lambda_j = i\cos(\pi j/(\nu+1)), \quad j = 1, 2, \ldots, \nu\,. \tag{14}$$



(a) *The solid curve corresponds to an initial state where all eigenvectors have equal representation in $r^{(0)}$. For the dashed curve the initial state is random. The dotted line shows κ^*.*

(b) *The residual norm corresponding to the solid line in the previous graph. The dotted line shows the asymptotic expected convergence rate with κ^*.*

Figure 3: *Locally optimal CFL numbers with 16x16 system.*

They are all imaginary and, assuming ν is even, non-zero. The eigenvalues λ_{min} and λ_{max} become $\pm i\, cos((\pi\nu/2)/(\nu+1))$ and $\pm i\, cos(\pi/(\nu+1))$ respectively and $p(\lambda, \kappa)$ is simply $1 - 0.6\kappa\lambda + 0.36\kappa^2\lambda^2$. Solving (11) then gives us an optimal constant CFL number of

$$\kappa^* = \frac{5/3}{\sqrt{cos^2(\pi/(\nu+1)) + cos^2((\pi\nu/2)/(\nu+1))}}, \tag{15}$$

which approaches 5/3 as ν increases. The corresponding optimal convergence factor is

$$\gamma_{opt} = \frac{\sqrt{\left(cos^2(\frac{\pi}{\nu+1}) + cos^2(\frac{\pi\nu/2}{\nu+1})\right)^2 - cos^2(\frac{\pi}{\nu+1})cos^2(\frac{\pi\nu/2}{\nu+1})}}{cos^2(\frac{\pi}{\nu+1}) + cos^2(\frac{\pi\nu/2}{\nu+1})}, \tag{16}$$

which will approach 1.

Since no *constant* κ can outperform κ^*, for the local RK CFL optimization method to be *globally* optimal as well, if it eventually approaches some constant value, it has to be κ^*. That is in fact what happens, as is shown in figure 3. Regardless of the initial state κ will converge towards κ^*, although that convergence is relatively slow and decreases with expanding system size.

Exactly the same behavior results for the corresponding problem in two dimensions, with the system matrix A being the difference operator $D_{0x} + \alpha D_{0y}$, except that there exists no closed-form expression for λ_{min} for general values of α. Nevertheless, if we have λ_{min} and λ_{max} we can compute both κ^* and γ_{opt}.

In the general case we cannot expect convergence for κ, but it is a reasonable assumption that the CFL number, in the long run, will not be far from the global optimum and at least oscillate around it. We will denote this by the *asymptotic phase,* as opposed to the initial *transient phase* possibly leading up to it.

An alternative way of analyzing the different stages of convergence is to use the classifications *sublinear*, *linear* and *superlinear* convergence phases, as in [8]. For these types of Krylov subspace methods, with a limited $m \ll \nu$, however, the final superlinear phase will never occur and it is more convenient to make the simpler above distinction.

If the one-dimensional difference operator in our example is augmented with artificial viscosity

$$0.005h^4(D_+D_-)u^n \tag{17}$$

the eigenvalues will no longer be purely imaginary or follow the same curve of κ in the complex plane. However, it is still easy to identify a pair of eigenvalues λ_{min} and λ_{max} that play the same role in determining the optimal CFL number and convergence speed.

Let us make the following assumption: $p(\lambda_{min}, \kappa)$ is closest to the unit circle for low values of κ and $p(\lambda_{max}, \kappa)$ is the first one to leave *the unit circle (or equivalently, λ_{max} is the first eigenvalue to leave the stability region for the current RK method) as κ increases.*

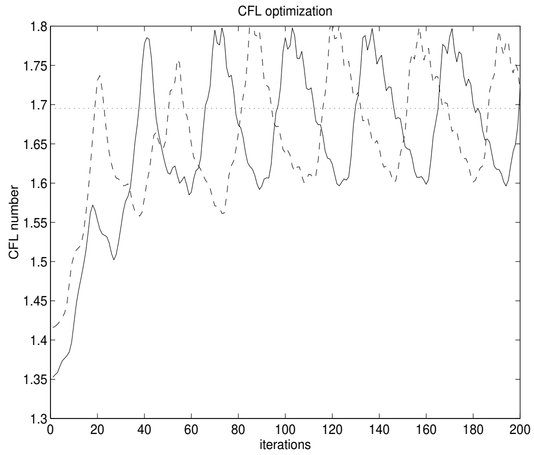
Solving (11) and computing γ_{opt} get slightly more complicated here. The result of this modified problem is shown in figure 4. Due to the eigenvectors no longer being orthogonal (since A is no longer antisymmetric), the expression for the residual norm (2) will contain (possibly negative) terms relating the coefficients of various eigenvectors in $u^{(n)}$ to each other. Then there is no obvious way of telling what will happen *in a single step* just by looking at the eigenvalues. Although neither κ nor γ completely converge, the general trend is fairly clear. Long-term behavior can still be predicted using the same technique as in the original simplified problem.

Finally, we consider a more realistic case, namely the linearized Euler equations in two dimensions, with three variables:

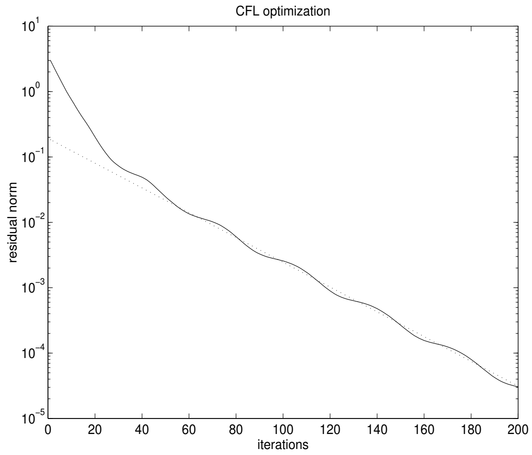
$$\begin{bmatrix} 1 & c & 0 \\ c & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} u_x + \begin{bmatrix} 0 & 0 & c \\ 0 & 0 & 0 \\ c & 0 & 0 \end{bmatrix} u_y = 0, \qquad c > 0,\ x \in [0,3],\ y \in [0,1]\,. \tag{18}$$

If $c > 1$, the equation models subsonic flow and if $c < 1$ it is supersonic. We will use $c = 2.0$, a space discretization of 20 by 60 points, centered differences with some artificial viscosity, the boundary conditions

$$\begin{aligned} &u_1(0,y) = -u_2(0,y), \quad &&u_3(0,y) = 0, \quad &&u_1(3,y) = u_2(3,y), \\ &u_3(x,0) = 0, &&u_3(x,1) = 0 && \end{aligned} \tag{19}$$



(a) *The solid curve corresponds to an initial state where all eigenvectors have equal representation. For the dashed curve the initial state is random. The dotted line shows κ^*.*



(b) *The residual norm corresponding to the solid line in the previous graph. The dotted line shows the asymptotic expected convergence rate with κ^*.*

Figure 4: *Locally optimal CFL numbers with 16x16 system and artificial viscosity.*

and solve the problem with CFL optimization of the standard *three-stage* RK iterations

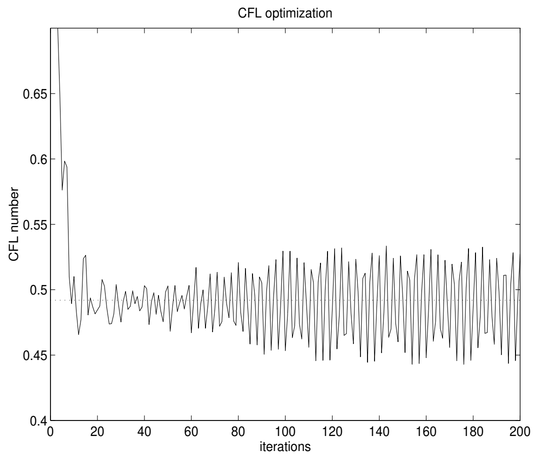
$$r^{(n+1)} = r^{(n)} - \kappa A r^{(n)} + 0.6\kappa^2 A^2 r^{(n)} - 0.36\kappa^3 A^3 r^{(n)} . \tag{20}$$

In this case, a numerical computation gives $|\lambda_{max}| \approx 3.82$. Since A is the combination of several centered differences and artificial viscosities, we can expect its eigenvalues to have a distribution approximately similar to the previous model problem. The ones with the largest absolute values will be mainly imaginary with small, positive real parts. If we assume that something like 99.5% of λ_{max} consists of the imaginary part and let $\lambda_{max} = 3.82(\sqrt{1-0.995^2} + 0.995i)$, we can solve

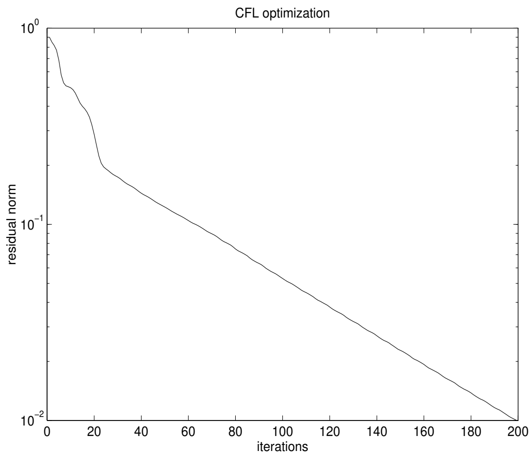
$$|p(\lambda_{max}, \kappa)| = |1 - \lambda_{max}\kappa + 0.6\lambda_{max}^2\kappa^2 - 0.36\lambda_{max}^3\kappa^3| = 1 \tag{21}$$

to get an upper bound for κ. The largest solution in our region of interest, $\kappa > 0$, is $\kappa \approx 0.49$ and the actual κ^* should be slightly lower than that, to allow for convergence.

As can be seen in figure 5, the behavior of the average locally optimal κ is similar to those of the model problems. We also get a constant average slope for the residual norm as the oscillations of κ stabilize around a fixed equilibrium.



(a) *The solid curve is the locally optimized κ. The dotted line shows the approximate limit of the stability region.*



(b) *The corresponding residual norm.*

Figure 5: *Locally optimal CFL numbers with the linearized Euler equations.*

2.2 Optimization for other Krylov methods

Consider the nth iteration step of a general Krylov method

$$r^{(n+1)} = r^{(n)} + \sum_{j=1}^{m} \alpha_j A^j r^{(n)} = p(A) r^{(n)} . \tag{22}$$

In the particular case of the Generalized Minimum RESidual iteration [1], minimization of the residual norm

$$\|r^{(n)}\|^2 = (p(A)r^{(n)})^H (p(A)r^{(n)}) = \rho(\alpha_1, \ldots, \alpha_m) \tag{23}$$

is carried out over the coefficients $\alpha_1, \ldots, \alpha_m$ and is in each step a least-squares problem. Ideally, GMRES would not stop until ν terms, but since each one requires explicit storage of yet another residual, for practical purposes it is necessary to introduce a limit $m \ll \nu$.

If we are to analyze the long-term behavior of (the re-started) GMRES(m) in the same way as we did with RK in section 2.1 we once again start by assuming that all eigenvectors of the system matrix are orthogonal and all α_j are constant from step n. There will be a similar bound for the convergence factor:

$$\max_j |p(\lambda_j, \alpha_1, \ldots, \alpha_m)| . \tag{24}$$

Just like the optimal constant κ^* was found where two or more curves crossed in the plane, the optimal choice of GMRES parameters $\alpha_1^*, \ldots, \alpha_m^*$ will generally

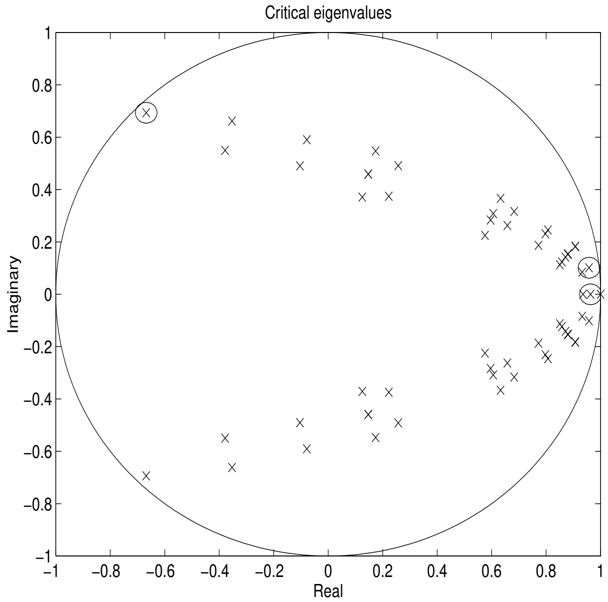


Figure 6: *Eigenvalues of $p(A)$ for second-order GMRES. The distribution is symmetric with respect to the real axis.*

correspond to $m+1$ or more surfaces $|p(\lambda_j, \alpha_1, \ldots, \alpha_m)|$ crossing in $(m+1)$-dimensional space.

Consider the partial differential equation

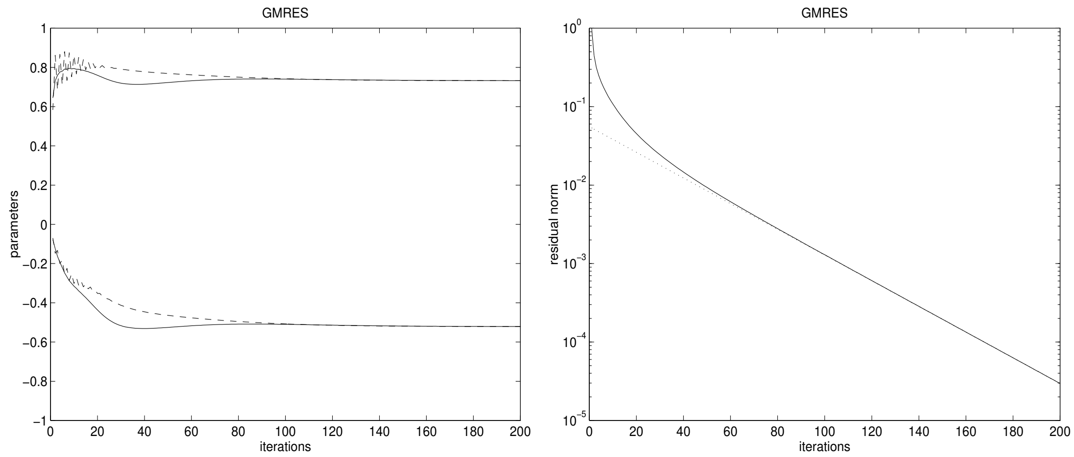
$$\left\{ \begin{array}{ll} u_x + 0.5u_y = 0 & x, y \in \Re \\ u(x,y) = u(x+j, y+k) & j, k \in \mathcal{Z} \end{array} \right. \tag{25}$$

approximated with central differences and the artificial viscosity (17). The eigenvectors of the system matrix will be the Fourier vectors and thus orthogonal. The eigenvalues of the system matrix A are given by

$$\begin{array}{l} \lambda_{j,k} = 0.08\left(sin^4(\frac{\pi j}{\nu}) + sin^4(\frac{\pi k}{\eta})\right) + i\left(sin(\frac{2\pi j}{\nu}) + 0.5sin(\frac{2\pi k}{\eta})\right) \\ j = 0, \ldots, \nu - 1, \quad k = 0, \ldots, \eta - 1\,. \end{array} \tag{26}$$

One eigenvalue $\lambda_{0,0}$ is identically zero, so from (24) we would expect the "optimal" long-term convergence factor to be 1. However, from our definition of the residual as $r^{(n)} = Au^{(n)}$, we see that the coefficient corresponding to $\lambda_{0,0}$ in $r^{(n)}$ is already gone, so that eigenvalue will not be noticed in any optimization of $\|r^{(n)}\|$.

A two-stage restarted GMRES(2) will distribute $p(\lambda_{j,k})$ approximately as shown in figure 6 within the unit circle. The ones that limit the long-term convergence factor (i.e. the ones with absolute values closest to 1) in this particular case are circled. Especially the critical eigenvalues near (1, 0) may vary with ν, η and the parameters α_1 and α_2, but there is always some optimal parameter



(a) *The solid curve corresponds to an initial state where all eigenvectors have equal representation. For the dashed curve the initial state is random.*

(b) *The residual norm corresponding to the solid line in the previous graph. The dotted line has a constant slope for comparison.*

Figure 7: *GMRES for the two-dimensional flow equation discretized on an 8x8 grid.*

configuration that, under the right circumstances, will govern the behavior of the method as the number of iterations grows large. Figure 7 shows the parameters and residual norm for the problem on an 8x8 grid.

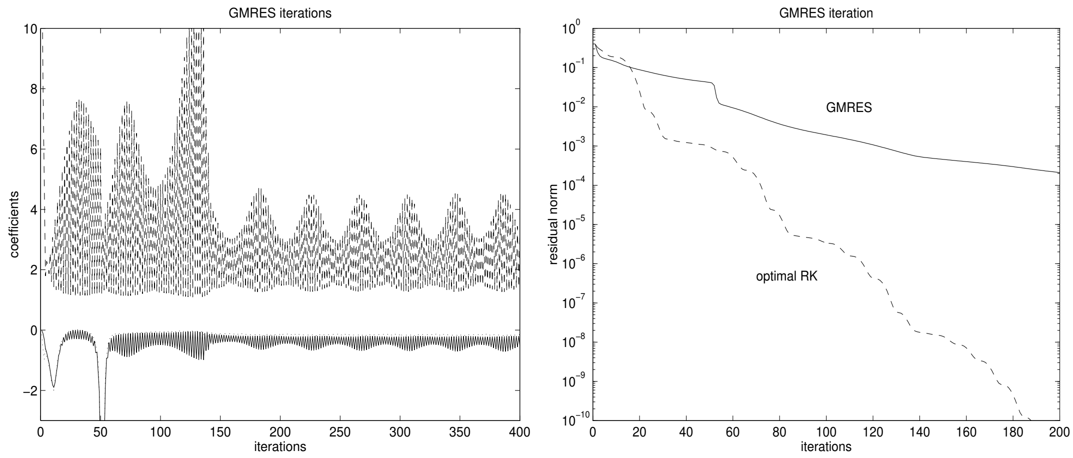
Since GMRES can adjust *all* coefficients rather than only one as the optimized RK, one would expect it to have superior convergence properties — at least asymptotically — but this is not always true.

Consider, for example, the equation

$$\begin{cases} u_x = 0\,, & 0 \leq x \leq 3\,, 0 \leq y \leq 1\,, \\ u(3,y) = 0\,, & u(x,0) = 0\,, \end{cases} \tag{27}$$

discretized with the centered difference formula, but without artifical viscosity, and linear extrapolation at the free end. Initial data is a Gaussian pulse. The result of GMRES(3) is shown in figure 8. Next to it is the curve for an optimized standard RK with the same number of stages.

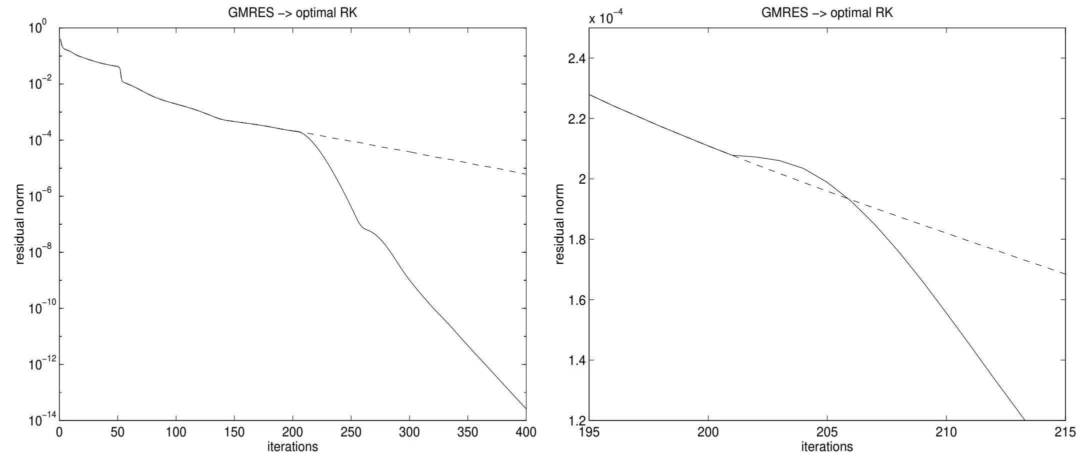
Apparently, the parameters do not oscillate around fixed convergence limits. GMRES is "stuck" in a cyclic process that will never obtain optimal long-term properties. The local optimality of the method does not allow it to escape this cycle, as that turns out to require a short-term sacrifice of residual reduction. Figure 9 shows what happens when the iteration method is suddenly switched to Runge-Kutta of the same order and with a fixed CFL number $\kappa = 1.77$ (which is approximately the average κ of the optimized RK in the previous figure).



(a) *The three parameters* α_1, α_2, α_3 *(solid, dashed, dotted).*

(b) *The solid line is the residual norm with GMRES. An optimized Runge-Kutta has been put in for comparison.*

Figure 8: *GMRES for an "unsuitable" problem.*



(a) *The solid curve is what happens with a switch to fixed RK after 200 iterations.*

(b) *Close-up. The RK method initially leads to slower convergence.*

Figure 9: *GMRES for an "unsuitable" problem.*

In that example, the convergence factor of the RK method is approximately 0.90. Yet, the actual residual reduction for the first few steps after the switch is considerably less efficient than that (and virtually none in the initial step). It is the presence of negative terms in the residual norm (23) that makes it possible to reduce all eigenvector coefficients by at least 0.90 and still not accomplish any noticeable residual reduction. If the eigenvectors of the system matrix are orthogonal (due to, for example, periodicity) this phenomenon cannot occur. Then the residual is reduced by at least γ in *each* step.

$$\begin{aligned} \|r^{(n+1)}\|^2 = (p(A)r^{(n)})^H(p(A)r^{(n)}) = \textstyle\sum_{j=1}^{\nu} |a_j p(\lambda_j)|^2 \leq \\ \leq \max_k |p(\lambda_k)|^2 \textstyle\sum_{j=1}^{\nu} |a_j|^2 = \max_k |p(\lambda_k)|^2 \|r^{(n)}\|^2 = \gamma^2 \|r^{(n)}\|^2 \end{aligned} \tag{28}$$

But in the long run, the average slope of the residual curve will always match the convergence factor.

For flow problems, the first coefficient of a Krylov method will be, in some sense, proportional to the speed at which waves in the error travel and leave the computational domain. The complete freedom of parameter optimization of GMRES allows the first coefficient to become too small. As figure 8 (a) shows, even the lowest dips of the solid line are far from the corresponding ≈ -1.77 obtained with an optimal RK, once the cyclic behavior starts. (Further problems with GMRES will be discussed in the section about non-linear PDEs.)

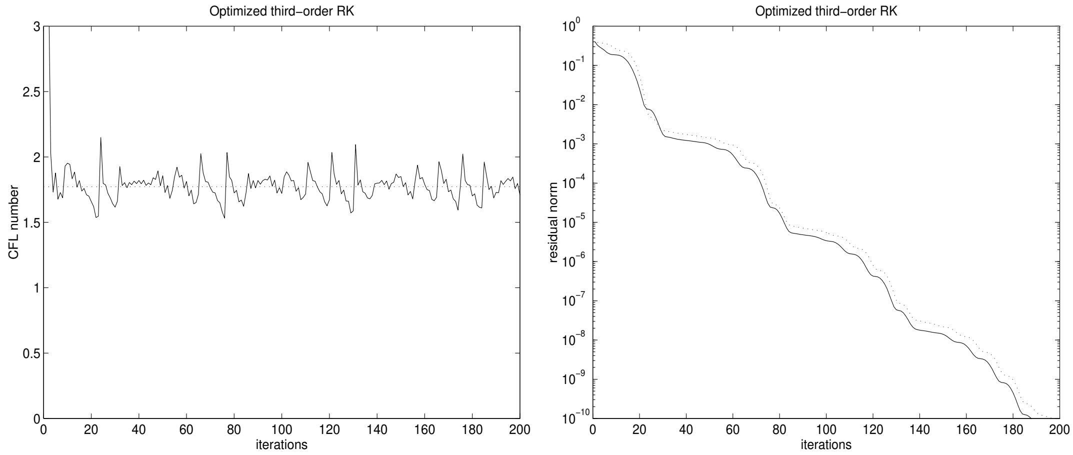
This motivated the construction of a modified GMRES method [2], where the first coefficient is fixed at some nonzero value, guaranteeing proper wave speeds, while the rest are optimized like before.

2.3 Relative performance

We continue the study of (27). In figure 10 an optimized standard third-order RK is compared to the same method with a fixed κ equal to the average of the asymptotic optimal CFL numbers — roughly 1.77.

Clearly, the curves are almost identical. The only difference is an initial advantage for the optimal RK that the constant RK can never compensate. The optimization makes it possible to adjust the coefficients of the eigenvectors in $r^{(n)}$, using even unstable CFL numbers, to produce a smaller norm. Generally, the optimal RK will decide to suppress the a_j belonging to the largest eigenvalues more, since they are also weighted with the largest $p(\lambda_j)$ in the residual.

One example is equation (13). A two-stage RK method with a globally optimal CFL number will reduce the coefficients belonging to the extreme eigenvalues equally. If all eigenvectors in $u^{(0)}$ have the same distribution, the coefficients in the *residual* will eventually look like in figure 11 (a). The right end dominates due to its greater weights. Let S be an orthonormal matrix of eigenvectors of A



(a) *The solid curve is the locally optimal CFL number.*

(b) *Close-up. Optimal RK (solid) and RK with a constant κ (dotted).*

Figure 10: *Locally optimal versus constant-CFL RK.*

and $w^{(n)} = S^{-1}u^{(n)}$ the eigenvector distribution in $u^{(n)}$ at each time step. Then

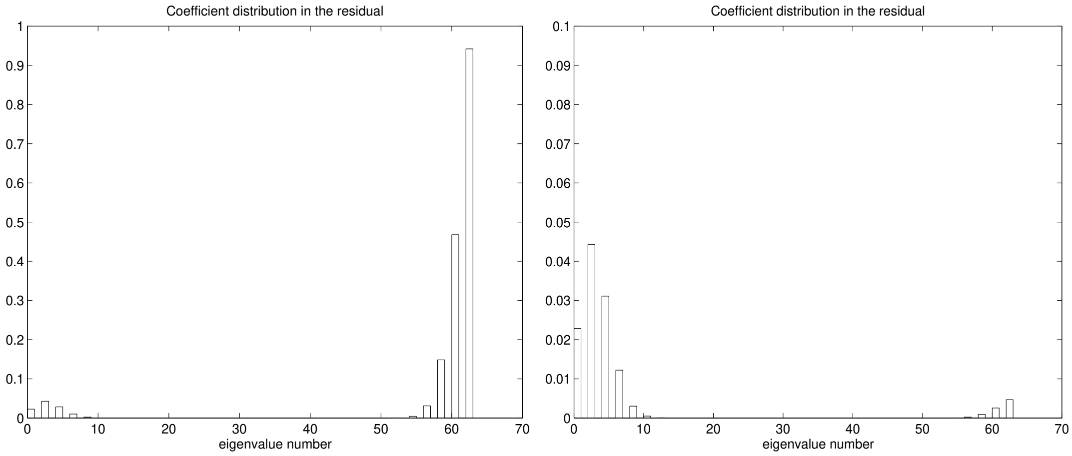
$$\|r^{(1)}\|^2 = \min_{\kappa} \|p(A,\kappa)r^{(0)}\|^2 = \min_{\kappa} \|p(\Lambda,\kappa)\Lambda w^{(0)}\|^2 , \tag{29}$$

$$where\ \Lambda = S^{-1}AS = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_\nu \end{bmatrix} . \tag{30}$$

The iteration step is a least-squares problem over $p(\lambda_j, \kappa)$, weighted by $\lambda_j w^{(0)}$. The coefficients belonging to the larger eigenvalues around λ_{max} will then be more important on average.

An optimal method would work harder on reducing those coefficients, if possible. That is what the optimal RK does, initially. By picking CFL numbers that will decrease the dominant parts, at the cost of less reduction at the other end, it gives the extremes more similar orders of magnitude. When no more can be gained from this approach the CFL value will stabilize at κ^*. Figure 11 (b) shows the result with the same settings but CFL optimization. (Note the different scales.)

For any *finite* number of iterations n, the globally optimal κ^* need not be the best choice for a constant CFL number. In the general, practical case, such a κ will only be available through repeated experiments, and educated guesses or rules of thumb will have to be used when solving flow problems. A fully optimal CFL number is normally just of theoretical importance, as the solution itself is what is desired and repeating the computational process in order to find a way to solve the same problem faster seems pointless.



(a) *The eigenvector coefficients in the residual after 200 iterations with a constant κ^*.*

(b) *The eigenvector coefficients with a locally optimal CFL number after 200 iterations.*

Figure 11: *Residual coefficients of a 64x64 system.*

Although local optimization works and eliminates the need to choose the parameter κ, it is far from evident that it would also perform well enough to match the best possible guess.

However, at least for linear cases, experiments indicate that the result of local RK optimization is similar to or better than anything that can be achieved with a constant κ for any given n. Figure 12 shows the residual norm for the same discretized equation with different constant CFL numbers, compared to the result of local optimization.

Consider a model problem with the system matrix

$$A = \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix} \tag{31}$$

to which we want to apply the second-order RK metod (12). Since there are only two eigenvalues, determining $\lambda_{max} = 3$ and $\lambda_{min} = 1$ is trivial. The globally optimal CFL number κ^* is then given by (11), which becomes

$$|1 - \frac{9}{5}\kappa^* + \frac{81}{25}(\kappa^*)^2| = |1 - \frac{3}{5}\kappa^* + \frac{9}{25}(\kappa^*)^2| \tag{32}$$

or $\kappa^* = 5/12$. The resulting globally optimal convergence factor γ_{opt} becomes $|p(\lambda_{max}, \kappa^*)| = 13/16 = 0.8125$. This is in fact the weakest possible convergence optimized RK can achieve in a single step for *any* vector $u^{(n)}$. Intuitively, this makes sense — in the asymptotic phase all advantages of local CFL optimization have been exhausted. There is nothing more to gain. Figure 13 shows the optimal

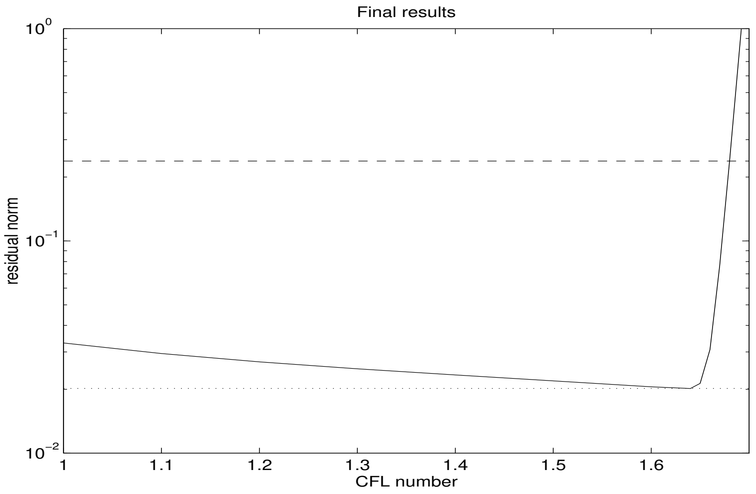


Figure 12: *The solid line shows the residual norms after 200 iterations with different fixed κ values. The dotted line is the level reached with local optimization and the dashed line the result with κ^*.*

convergence factors $\|r^{(n+1)}\|/\|r^{(n)}\|$ for different vectors $u^{(n)} = v_1 + a_2 v_2$, where $v_1 = [1, 0]^T$ and $v_2 = [-\sqrt{1/2}, \sqrt{1/2}]^T$ are the eigenvectors of A. The solid curve represents positive values of a_2 and the other one is for negative a_2.

There are two directions for which the convergence factor is exactly γ_{opt}, represented by dotted lines in figure 13, and experiments show that $u^{(n)}$ *always* converges to either of those. In figure 14 the same thing is illustrated as a function of direction for $u^{(n)} = a_1 v_1 + a_2 v_2$ in the $a_1 a_2$ plane. The inner solid curve is the convergence factor, limited by the dashed circle with a radius of γ_{opt}. The dotted lines are the region in $\Re^2$ to which $u^{(n)}$ converges when local CFL optimization is used. Analytical computation of that subspace even for this simple model problem requires finding the largest real root of a third-degree polynomial, and is presumably not feasible.

With a *constant* CFL number κ, $u^{(n)}$ will behave as follows:

- If $|p(\lambda_A, \kappa)| > |p(\lambda_B, \kappa)|$, $u^{(n)}$ converges towards the line μv_A, $\mu \in \Re$, since $u^{(n)} = p(\lambda_A, \kappa)^n v_A + p(\lambda_B, \kappa)^n v_B \approx p(\lambda_A, \kappa)^n v_A$ as n increases.

- If $\kappa = \kappa^*$, then $u^{(n)} = p(\lambda_A, \kappa)^n u^{(0)} = p(\lambda_B, \kappa)^n u^{(0)}$ and $u^{(n)}$ remains on the line $\mu u^{(0)}$, $\mu \in \Re$.

Provided the assumption about λ_{max} and λ_{min} for general cases in section 2.1 is correct, this behavior model will hold, asymptotically, for *any* number of dimensions ν. If $\kappa = \kappa^*$, then $|p(\lambda_{max}, \kappa)| = |p(\lambda_{min}, \kappa)|$ and these eigenvalues dominate $p(A)$. The eigenvectors in $u^{(n)}$ corresponding to all other λ_j will shrink more rapidly and the solution vector approaches the linear subspace spanned by v_{max} and v_{min}.

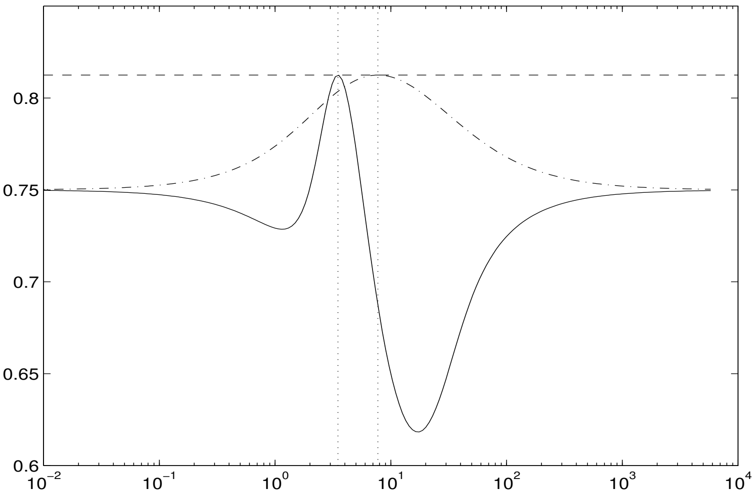


Figure 13: *Convergence factors for different proportions of eigenvectors in* $u^{(n)}$.

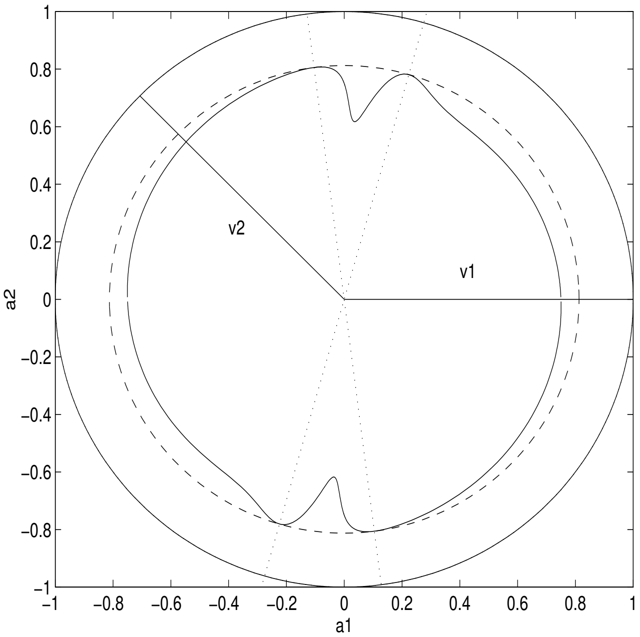
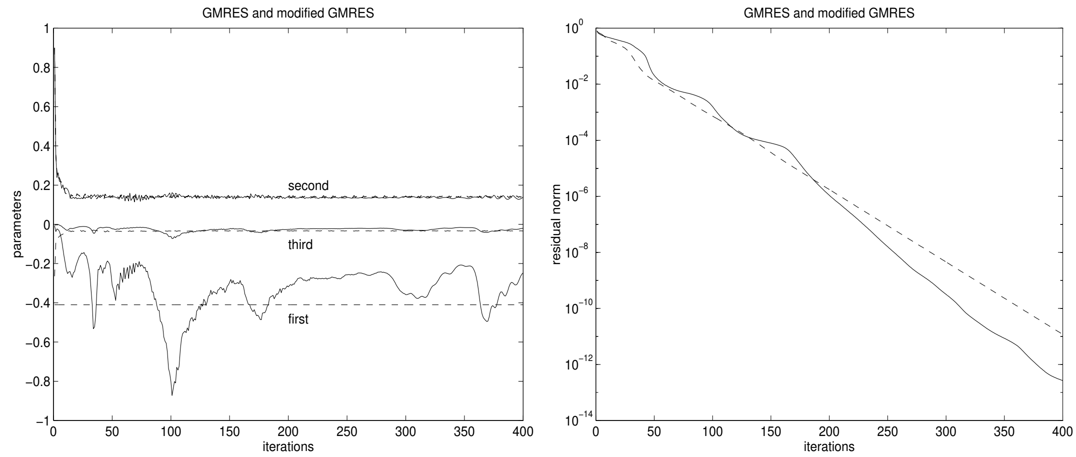


Figure 14: *Convergence factors for different directions of* $u^{(n)}$.



(a) *The parameters. The solid lines belong to GMRES and the dashed ones to mGMRES.*

(b) *Convergence curves. The solid line represents GMRES and the dashed line mGMRES.*

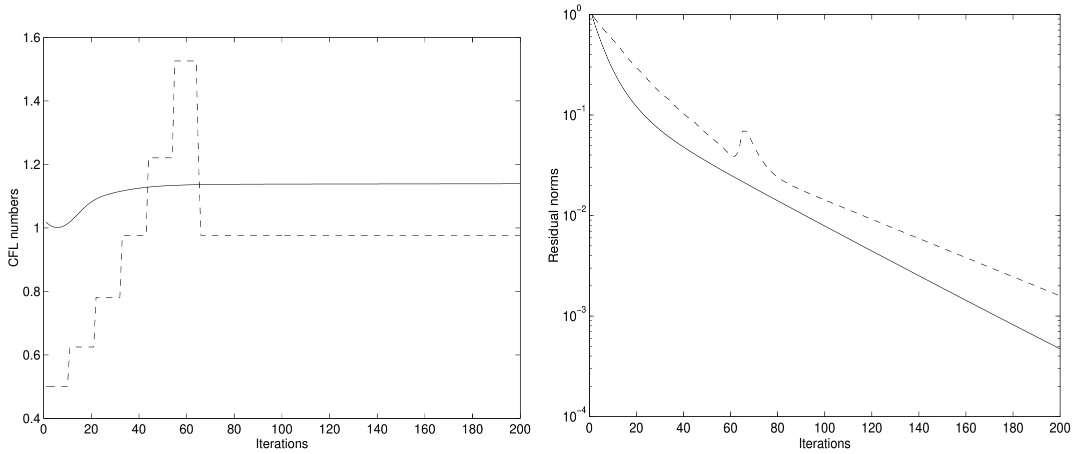
Figure 15: *GMRES(3) and mGMRES(3) for the linearized Euler equations.*

Local optimization makes sure $u^{(n)}$ eventually gets aligned in such a way that the optimal convergence factor (on average) is γ_{opt}. This subspace in turn will move towards the plane spanned by v_{max} and v_{min} as the remaining eigenvector coefficients in $u^{(n)}$ get proportionally smaller.

It is possible that if the system matrix A is sufficiently far from being normal, there can be variations from this model. As was indicated in 2.2, an iteration method based on local optimization can miss the global optimum if the path to it requires a short-term sacrifice of residual reduction. However, no such cases were found for linear equations and CFL optimization, indicating that it is fairly robust.

The modified GMRES method was introduced in [2] to ensure proper wave speed for GMRES and can outperform it in many cases. However, when low wave speed does not inhibit the convergence of GMRES or does not occur at all, the relationship between GMRES and mGMRES is somewhat analogous to that between optimized and constant-CFL RK described above. The first parameter in mGMRES, α_1, being constant will result in slower convergence, on the average, for a finite number of iterations. The convergence rate may match that of GMRES asymptotically, but only if the "correct" value is chosen for the constant parameter.

Figure 15 shows the three-stage GMRES and mGMRES applied to the linearized Euler equations (18) with boundary conditions (19). Here, the modified GMRES has been given a non-optimal constant parameter. The other two stabilize quickly and the slope of the convergence curve is approximately constant, but



(a) *CFL numbers* (b) *Resulting convergence history*

Figure 16: *RK optimization (solid) compared to automatic time stepping (dashed).*

considerably weaker than that of the ordinary GMRES. Much could be gained from an alternative to GMRES with improved wave speed properties, but without the dependence on the parameter choice of mGMRES. The successful use of local optimization for RK suggests that a hybrid method combining that and the modified GMRES might work well.

Finally, we will make one performance comparison with the automatic time stepping from [11]. Figure 16 shows the results of both methods when applied to (25). The automatic time stepping is a discrete optimization and generally cannot pick the asymptotically optimal CFL value, which means its residual curve will have less slope. It is also unable to take advantage of the eigenvector coefficient distribution in the initial phase.

2.4 Hybrid methods

Although the optimized Runge-Kutta is parameter-free in the sense that it selects its own CFL number, it needs to get the RK coefficients specified in advance. For flow problems, there exist standard coefficients for order two through at least five. Unless otherwise specified, the those in table 1 are the ones used in the experiments. They work, but they will not always be optimal.

The modified GMRES method, on the other hand, optimizes all parameters except the first one. A regular interchange of parameters between the methods should then produce the same degree of freedom as for GMRES — at least in the long run.

One approach, which we will denote by *hybrid 1*, is to check, in each step, which method of optimized RK and mGMRES would cause the greatest residual reduction, use that method and save its optimized parameter(s) as a starting point for the next step. I.e. if optimized RK is chosen, its CFL number will be taken as the free parameter of mGMRES next time, and if mGMRES is chosen, the optimized RK will then use the coefficients

$$\alpha_j^{(RK)} = \frac{\alpha_j^{(mGMRES)}}{\kappa^j}, \qquad j = 2, \dots, n. \tag{33}$$

This is numerically feasible because if either method is already being used, the marginal cost of trying out the other one is minimal. They both require computation of the same Krylov subspace vectors, along with pairwise scalar products thereof. Depending on which algorithm is used for mGMRES (see e.g. [2], [3]), they may work with different vector bases, but going from one to the other only requires work of the order of inverting an $(n+1)\times(n+1)$ matrix. For small values of n, numerical stability might not have to be considered and the coefficients can then be obtained from the normal equations without the need for any transformation between the optimization problems.

We know the residual norm caused by a possible application of the optimized RK, for any CFL number κ, is given by $\sqrt{\rho(\kappa)}$. Let S be an $(n+1)\times(n+1)$ matrix with components s_{ij} given by the scalar products of the basis vectors v_i and v_j in the Krylov subspace (5). Then if $\alpha_1, \dots, \alpha_n$ are the parameters, the residual norm that would result from applying mGMRES is

$$\sqrt{[1, \alpha_1, \dots, \alpha_n]\, S\, [1, \alpha_1, \dots, \alpha_n]^H}\,.$$

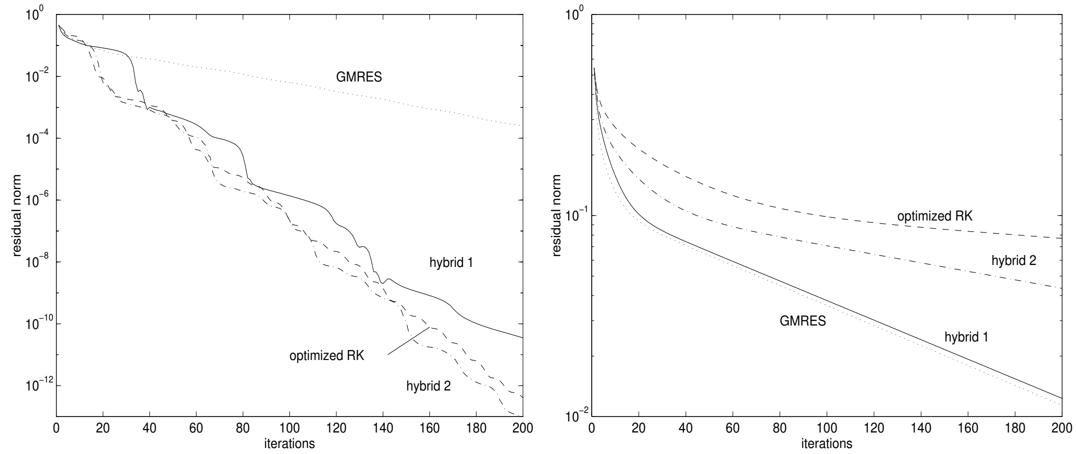
A second approach, which we will denote by *hybrid 2*, is to — in each step — start with the optimized RK, but just to get a CFL number κ to be used as the first parameter of mGMRES. After that iteration, the other parameters are then converted into RK coefficients to be used at the start of the next step, according to (33), and so on.

Both *hybrid 1* and *hybrid 2* need some form of parameter specification prior to the first step, but what values are given is relatively unimportant. All parameters will adjust themselves as the iterations proceed.

In order to determine the performance of either method, we will begin by looking at two extreme cases:

- When GMRES works poorly due to low wave speeds
- When wave speeds are unimportant and the ordinary GMRES outperforms other methods

In the first case, we return to the equation (27). The residual norms are shown in figure 17 (a). As expected, *hybrid 1* inherits some of the poor qualities of GMRES, while *hybrid 2* behaves more like the optimized RK and converges faster than any



(a) *When GMRES has a problem with low wave speeds.*

(b) *When GMRES works well because wave speed is not essential for convergence.*

Figure 17: *Comparisons between different iteration methods.*

of the others. In the second case, we work with the equation (13). Figure 17 (b) shows the results. Here, the optimized RK does not even come close to the efficiency of as GMRES. Thus, *hybrid 2* is not much better, while *hybrid 1* keeps up with GMRES fairly well.

In most realistic situations, we would expect something in between. Figure 18 (a) shows the same test with the equation (18) and boundary conditions (19). GMRES and *hybrid 1* are fairly equal in terms of performance. The more RK-related *hybrid 2* keeps up with those methods during the transient phase, but eventually evens out towards a lower convergence rate. It inherits some of the poor behavior of any ordinary RK for that problem.

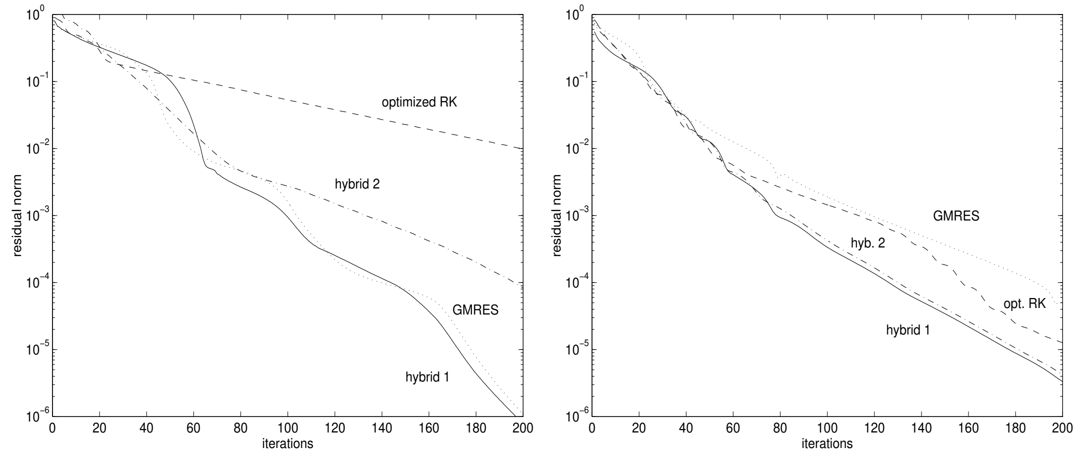
The last example is the equation

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} u_x + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} u_y = 0, \quad x \in [0,3],\, y \in [0,1]\,, \tag{34}$$

with the boundary conditions

$$u_1(0,y) = 0, \quad u_2(3,y) = 0, \quad u_1(x,0) = u_2(x,0), \quad u_1(x,1) + u_2(x,1) = 0\,, \tag{35}$$

artificial viscosity and linear extrapolation at the remaining boundaries. The result is in figure 18 (b). Here, both hybrids are superior to optimized RK and GMRES. Even though *hybrid 1* has, basically, the same parameter freedom as GMRES, its different optimization process appears to make it less prone to select too low wave speeds.



(a) *The linearized Euler equations.* (b) *The two-dimensional wave problem.*

Figure 18: *Comparisons between different iteration methods.*

Both hybrid methods have the practical advantage over the modified GMRES of not needing any parameters (save for the actual number of stages in the method). That can often be worth some sacrifice of performance and a comparison between them with the latter having an experimentally chosen optimal parameter is not completely fair, but we will make one nevertheless, to see how much worse or better the hybrid methods are likely to perform. Figure 19 shows the result.

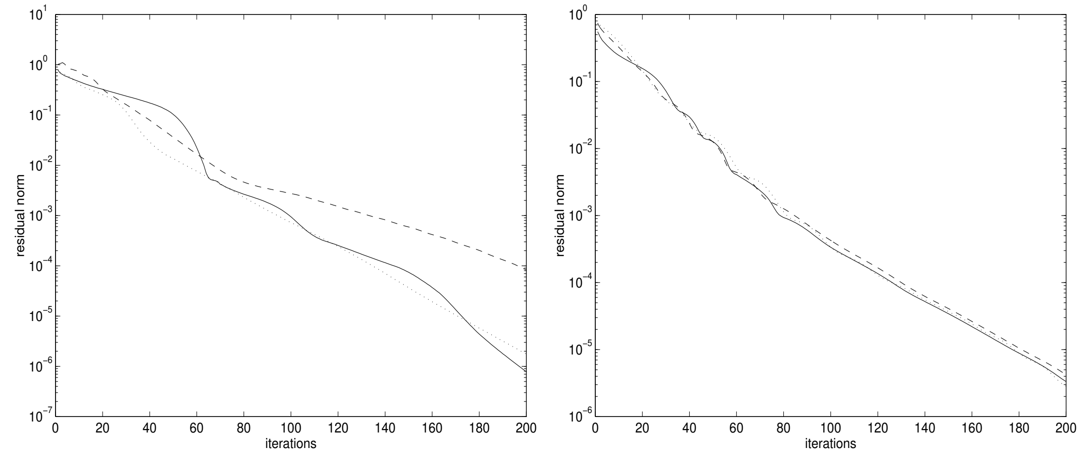
For the linearized Euler equations there is not much difference between mGMRES and *hybrid 1* (or GMRES — see figure 18). For the other case, all three methods behave almost identically.

The experimental results suggest that *hybrid 1* is a conceivable parameter-free alternative to mGMRES, and that *hybrid 2* is also a possible choice for those problems where the ordinary GMRES method does not work well.

2.5 Multigrid experiments

So far, we have seen that one of the limits on convergence rate for first-order systems is caused by the smallest eigenvalues, which will necessarily correspond to convergence factors very close to 1 within the stability region of, for example, the CFL parameter. Even optimization cannot alter this fact and the asymptotic convergence rate will decrease as the problem size grows (when the discretization becomes finer). This makes the trade-off between efficiency and accuracy of the problem-solving iterations even more acute.

A fairly recent way of dealing with this are the so-called multigrid methods. Using coarser auxiliary levels of discretization, low and high frequencies each get



(a) *The linearized Euler equations.* (b) *The two-dimensional wave problem.*

Figure 19: *hybrid 1 (solid), hybrid 2 (dashed) and mGMRES (dotted).*

reduced on the levels where this can be done the most efficiently. The smooth components of the residual — generally belonging to the smallest eigenvalues — can be sufficiently well approximated on a coarser grid, where the solution can be obtained with less computational effort. The corresponding problem on the coarse grid is $Av = r$, where r is the "restricted" residual from the fine grid. The solution u is then updated by adding a "prolonged" copy of v.

In most practical cases the exact solution on the coarse grid is not computed. Instead some iterative method is used, often in combination with computing another correction term on an even coarser grid, and so on, by applying the two-grid algorithm recursively — thus the name multigrid. A more complete description is given in e.g. [7].

For our type of problems, let R_i denote the restriction operator that maps a grid function from grid i to the coarser grid $i+1$, and P_i the prolongation operator that interpolates a grid function from grid $i+1$ to i, where $i = 0, 1, 2, \ldots$. Further, let A_i denote the system matrix on grid i. The multigrid algorithm can then be described the following way:

1. Perform l_1 iterations with some method to get an approximate solution $\bar{u}$ to $A_i u = f_i$, starting with u_i.

2. Compute the new defect $d_i = f_i - A_i \bar{u}$.

3. Solve $A_{i+1} v = R_i d_i$ on the coarser grid $i + 1$, starting with $v_{i+1} = 0$.

4. Use the coarse-grid solution as a correction term, $\tilde{u} = \bar{u}_i + P_i v$.

5. Perform l_2 iterations with some method to solve $A_i u = f_i$, starting with $\tilde{u}$.

(with some stopping criterion for i). This is also known as a "V cycle", due to the way the computations are moved from finer to coarser grids and back again. Other alternatives involve multiple coarse-grid references on some intermediate level, with smoothing iterations in between. Then we get so-called "W cycles". Those, however, will not be further discussed here.

The method used will be the two-stage RK (12) on two grids with only one pre-smoothing iteration on each. A cycle is then carried out as:

$$\begin{aligned} u_0 &= u^{(n)} - 0.6\kappa r(u^{(n)}) + 0.36\kappa^2 A_0 r(u^{(n)}) \\ r_1 &= R_0 r(u_0) \\ u_1 &= -0.6\kappa r_1 + 0.36\kappa^2 A_1 r_1 \\ u^{(n+1)} &= u_0 + P_0 u_1\,. \end{aligned}$$

The combined iteration matrix M can be written as

$$M = (I + P_0 q(A_1) R_0 A_0) p(A_0) \tag{36}$$

where $q(A_i) = -0.6\kappa I + 0.36\kappa^2 A_i$, i.e. $p(A_i) = I + q(A_i)A_i$.

The matrix M, however, will not be used for optimization purposes. It can be shown that obtaining the coefficients of the total polynomial $\rho(\kappa) = \|Mr^{(n)}\|^2$ would require more residual evaluations than performing the ordinary CFL optimization on each grid individually, so we will try the latter approach.

For the multigrid case the cost of the extra Krylov subspace vector evaluation required in CFL optimized RK (1), as opposed to *ordinary* RK (4), cannot always be compensated for by not having to evaluate $r^{(n+1)}$ explicitly in the next step. After the last post-smoothing operation before returning to a finer grid, this information is not used anyway. Thus, $A^m r^{(n)}$ is "wasted", but since this only occurs on coarse grids the relative cost is small.

We begin by studying the one-dimensional test problem

$$\begin{cases} u_x = 0\,, & 0 \le x \le 1 \\ u(0) = 0 \end{cases} \tag{37}$$

discretized with a central difference operator, artificial viscosity and linear extrapolation at the free end. The coarse grid is obtained simply by skipping every other point. The restriction operator R_0 transfers a grid function v_j to a corresponding $\tilde{v}_j$ according to

$$\tilde{v}_j = \frac{1}{4}v_{2j-1} + \frac{1}{2}v_{2j} + \frac{1}{4}v_{2j+1}\,. \tag{38}$$

Similarly, the prolongation operator P_0 mapping $\tilde{v}_j$ onto the fine grid is

$$v_j = \begin{cases} \tilde{v}_{j/2} & j\ even \\ (\tilde{v}_{(j-1)/2} + \tilde{v}_{(j+1)/2})/2 & j\ odd \end{cases} \tag{39}$$

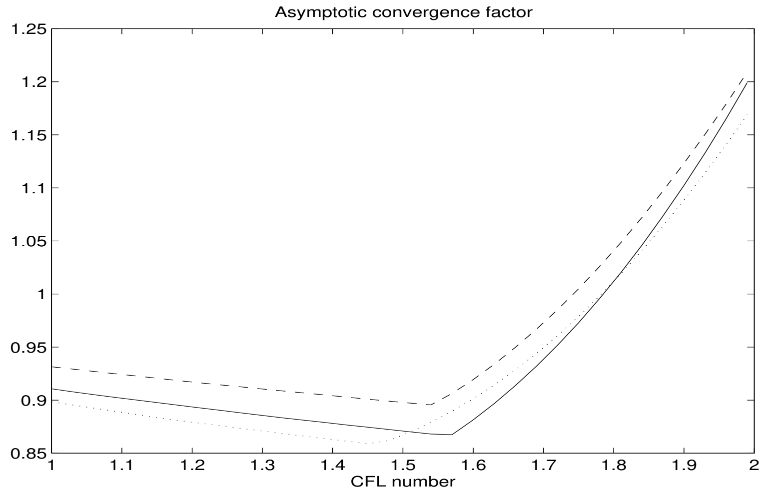


Figure 20: *Asymptotic convergence rates on the combined (solid), fine (dashed) and coarse (dotted) grid.*

with constant extrapolation at the boundaries. (The initial function chosen is a sine wave.)

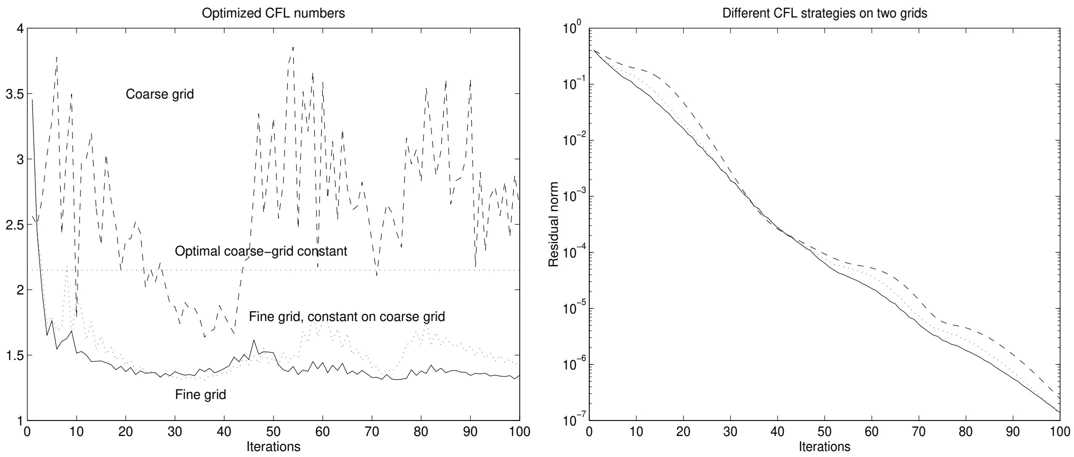
Figure 20 shows the asymptotic convergence rates for ordinary RK(2) on either grid or the two combined as functions of κ. Apart from having better convergence properties than just the fine-grid operator alone, the combined operator allows for (and is optimal at) higher CFL numbers, thus guaranteeing better wave propagation properties.

Individual local optimization on both grids should hopefully prove competitive with setting the CFL number to an experimentally determined globally optimal value. Another theoretical option would be to use local optimization on the fine grid only, keeping the CFL number on the coarse grid equal to a global optimum.

Tests indicate that the best choice of κ for both grids and a limited number of iterations ($\approx$ 100) is 1.55, and the best choice for the coarse grid alone is $\kappa = 2.15$. The behavior of the locally optimized κ is shown in figure 21 (a) and figure 21 (b) shows the corresponding convergence curves.

There is, apparently, not much difference in performance between the strategies. Local optimization on both grids individually works well and creates a slight initial advantage, as we have come to expect for the single-grid case. However, it can be shown that the asymptotic convergence rate of that method does not quite match those obtained with globally optimal values of κ.

The fluctuations aside, the optimized CFL number on the fine grid stays on average significantly lower than the asymptotically best choice for that grid itself (see figure 20). This is a result of adding the correction term from the coarse grid. It helps reduce the low-frequency contributions to $r^{(n+1)}$, i.e. the a_j in (6) corresponding to the smallest λ_j, which from the optimizer's point of view will



(a) *Optimized CFL numbers.*

(b) *Residuals with optimization on both grids (solid), fine grid only (dotted) and on neither grid (dashed).*

Figure 21: *Two-grid results for model problem.*

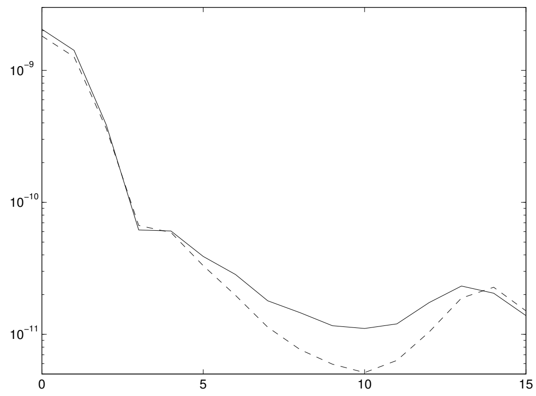
be indistinguishable from a reduction of $|p(\lambda_j)|$ and thus cause a different, lower κ^* in (11).

One might assume that the very high values of κ on the coarse grid are similarly due to a reduction of the high-frequency coefficients. However, in this case it is primarily caused by the shape of the coarse-grid residual vector not being sufficiently preserved from one iteration step to the next. Carrying out the local CFL optimization on the coarse grid by itself would eventually force the coefficients a_{min} and a_{max} into the proportions for which κ^* is optimal. But the prolongation, fine-grid smoothing and restriction create a disturbance of such magnitude that this will never happen. The coarse-grid optimization essentially starts from scratch in every step.

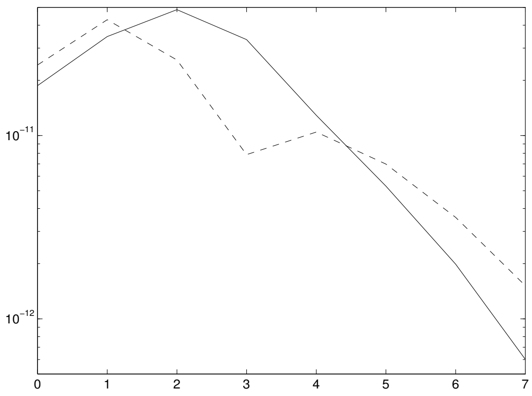
Figures 22 (a) and (b) show the distribution of eigenvector coefficients on the fine and coarse grid, respectively, before and after a full iteration step some 50 cycles from the start. On the fine grid the important proportion, i.e. between the left and right ends, stays more or less the same.

Adding one or more grids after that does not change the overall behavior. Optimization on even coarser grids essentially follows the same pattern as on the second one. Figure 23 shows the locations of the combined iteration matrix eigenvalues for one, two and three grids, with the constant CFL number 1.5.

Computation on more than one grid makes analysis of CFL optimization much more difficult. In this first model problem the resulting convergence was not much better than that of the best possible constant CFL number, and asymptotically even slightly inferior. But it appears equally possible that CFL optimization



(a) *Fine grid.*



(b) *Coarse grid.*

Figure 22: *Coefficient distribution before (solid) and after (dashed).*

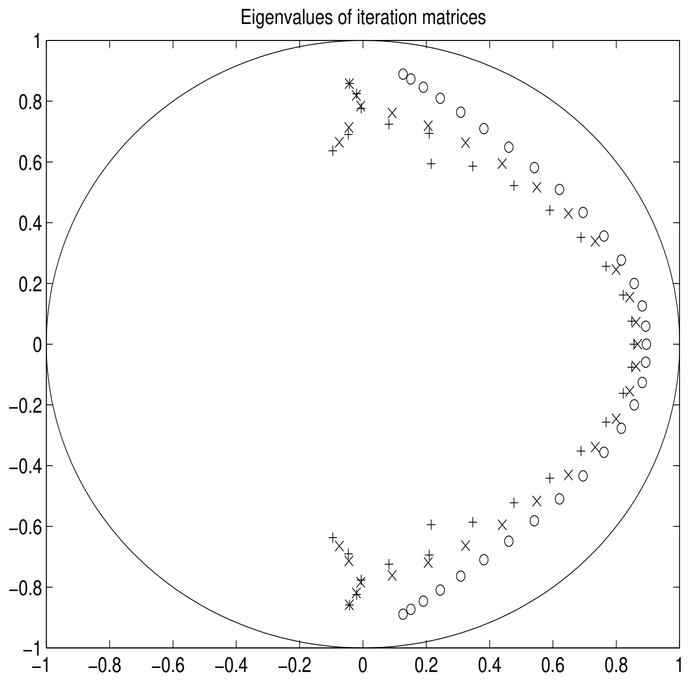


Figure 23: *Eigenvalues of the combined iteration matrix on one (o), two (x) and three (+) grids.*

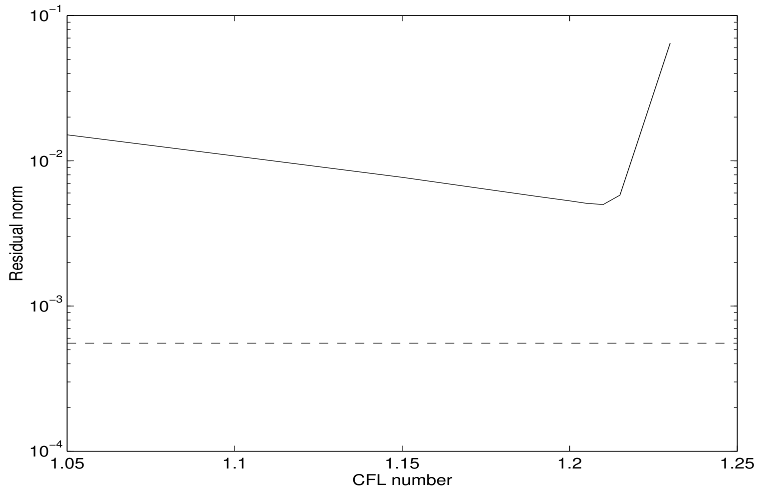


Figure 24: *Residual norm after 200 iterations as a function of CFL number. The dashed line marks the level reached with optimization.*

would perform even better relative to the fixed RK on multiple grids. After all, individual CFL optimization on each grid has more degrees of freedom.

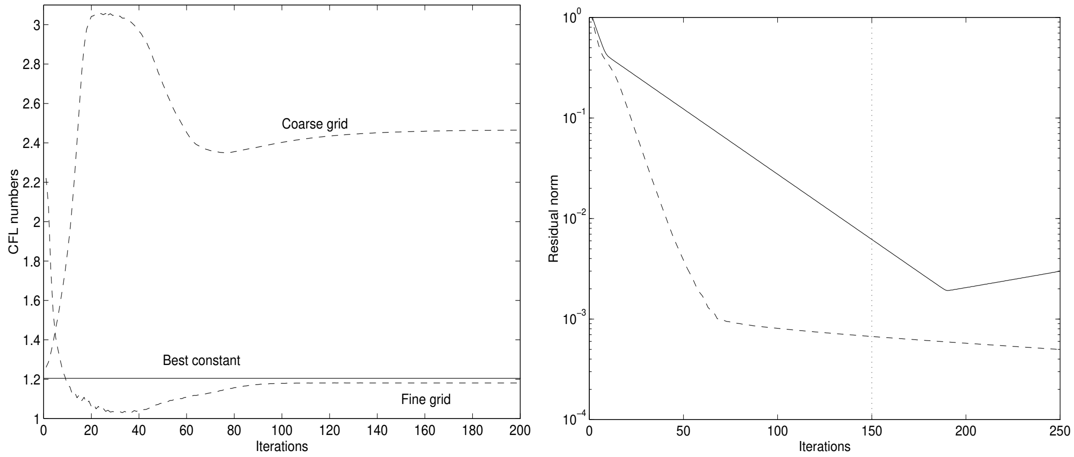
One case where significantly better results are obtained is the model problem (25). A cell-centered approach is used and both the restriction and prolongation operators get the point values from bilinear interpolation. (The fine grid is of size 14×14, with a centered Gaussian pulse as initial data.)

In figure 24 the residual norms after 200 two-grid iterations with different constant CFL values are shown, compared to the result of using locally optimized CFL numbers instead. A full order of magnitude is gained.

The optimized CFL numbers are shown in figure 25 (a). The solid line is the best constant choice for a limited iteration period of 150 steps. Here, *both* optimized values converge. Figure 25 (b) shows the corresponding residual curves. When κ begins to stabilize, the local optimization will converge relatively slowly. It seems as if the best constant choice of κ (which is slightly higher and will thus cause somewhat better wave properties) might soon outperform the optimization. The slope of its curve looks more promising. However, it turns out that such values are only stable during a finite number of iterations and the process will diverge before the residual norm comes close to the optimization curve.

Previous results have strongly indicated that if there is such a thing as an asymptotically optimal CFL value κ^*, local optimization will eventually locate it. If such values exist and are *different* on each grid, there is of course no way a single, constant CFL value could match the convergence rate of the locally optimized iterations. Optimization will thus have a strong advantage in cases where the shape of the residual is roughly preserved, on each grid level, by the restriction and prolongation operators from one iteration step to the next.

A comparative test of all methods mGMRES(2), optimized RK(2), GM-



(a) *CFL numbers.* (b) *Convergence curves.*

Figure 25: *2D model problem on two grids.*

RES(2), *hybrid 1 and* hybrid 2 (where mGMRES has been given an experimentally determined optimal parameter value) is shown in figure 26 (a). The problem is (37) on two grids. The hybrid methods roughly match the behavior of GMRES, which does poorly compared to the RK method for this problem.

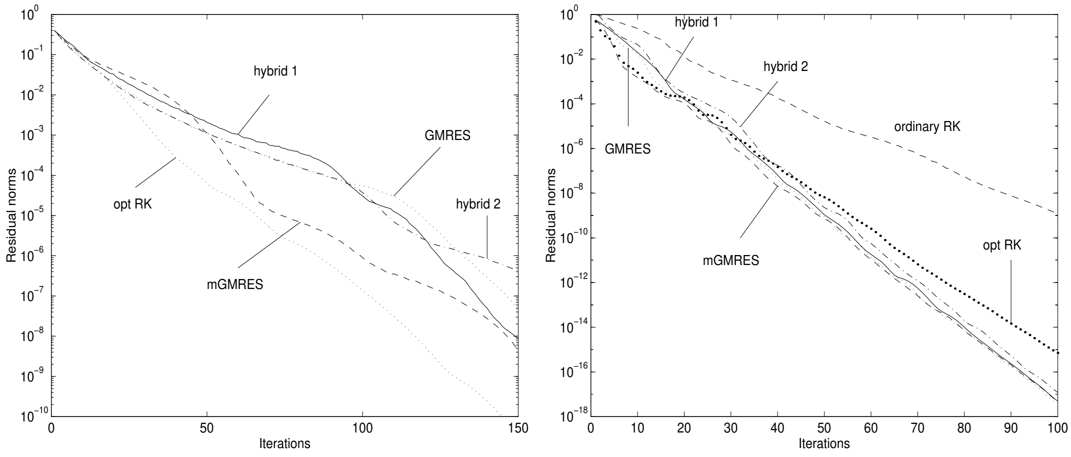
Similarly, a test of three-stage versions of the methods, plus an ordinary RK(3) with the best constant CFL number, applied to problem (34) on three grids is shown in figure 26 (b). Here the performance of the hybrids, GMRES and mGMRES is essentially the same, while optimized RK lags a little behind, but is still a significant improvement over the best constant-CFL RK.

3 Optimized methods for nonlinear PDEs

In this section we will apply the previously discussed optimization strategies to a few nonlinear cases, which are specific but in some sense representative of actual modern problems.

Two natural choices for nonlinear flow are the Euler and Navier-Stokes equations. It has been shown in e.g [9] and [10] that they can be efficiently solved using multigrid methods. In two dimensions the Euler equations can be written as

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0$$



(a) *1D problem solved on two grids.* (b) *2D problem solved on three grids.*

Figure 26: *Performance comparison of various methods on multiple grids.*

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \; F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (\rho E + p) u \end{bmatrix}, \; G = \begin{bmatrix} \rho v \\ \rho u v \\ \rho u^2 + p \\ (\rho E + p) v \end{bmatrix}$$

$$p = (\gamma - 1)\rho e$$

$$E = \frac{1}{2}(u^2 + v^2) + e,$$

and the variables u and v are the velocity components in the x and y directions, ρ is the density, p is the pressure, E is the total energy and γ is the ratio of specific heats.

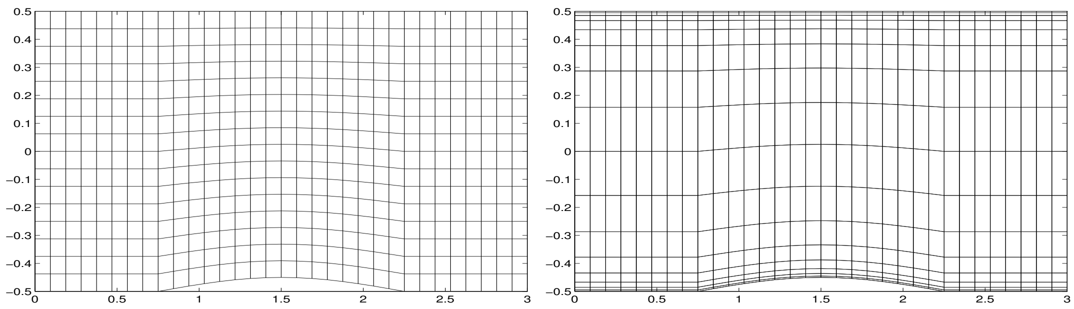
The Navier-Stokes equations, with temperature dependence and body forces neglected, become

$$\frac{\partial W}{\partial t} + \frac{\partial F^c}{\partial x} + \frac{\partial G^c}{\partial y} = \frac{\partial F^v}{\partial x} + \frac{\partial G^v}{\partial y}$$

where $W = (\rho \;\; \rho u \;\; \rho v \;\; \rho E)^T$, F^c and G^c, the so-called convective fluxes, are the same as F and G above,

$$p = (\gamma - 1)\rho(E - \frac{1}{2}(u^2 + v^2)),$$

$$F^v = \begin{bmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ u\sigma_{xx} + v\sigma_{xy} \end{bmatrix}, \; G^v = \begin{bmatrix} 0 \\ \sigma_{xy} \\ \sigma_{yy} \\ u\sigma_{xy} + v\sigma_{yy} \end{bmatrix},$$



(a) *Euler grid.*

(b) *Navier-Stokes grid (coarse level).*

Figure 27: *Computational grids for non-linear model problems.*

and

$$
\begin{aligned}
\sigma_{xx} &= 2\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) \\
\sigma_{yy} &= 2\mu\frac{\partial v}{\partial y} - \frac{2}{3}\mu(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) \\
\sigma_{xy} &= \mu(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y})
\end{aligned}
$$

with μ specified by way of the Reynolds' number $Re = \rho_\infty U_\infty L/\mu$, where ρ_∞ and U_∞ are the reference density and velocity and L is a characteristic length.

The computational domain is a rectangular channel with flow from left to right and a slight inward bump in the lower wall. The discretization depends on which equations are to be solved. For the Euler equations all grid elements are of approximately equal size, while the Navier-Stokes equations require better resolution in the normal direction near the walls. Figure 27 shows one of the coarser grids for each problem. Initial data is a constant equal to the boundary inflow given at the left end. The resolution on the finest grid is 129 by 65 points.

After discretization, the resulting equations have been given second- and fourth-order artificial viscosity following Jameson [9]. Unless otherwise specified, the Navier-Stokes problem has a Reynolds' number of 10^4.

For subsonic flow, the solution to the Euler equations looks like figure 28.

3.1 The Runge-Kutta method

For nonlinear systems the RK iteration equivalent to (1) can be written

$$r^{(n+1)} = r(u^{(n)} + \beta_1\kappa r(u^{(n)} + \beta_2\kappa r(\ldots + \beta_m\kappa r(u^{(n)}))\ldots)) \qquad (40)$$

where β_j would relate to the linear-case coefficients according to $\alpha_j = \prod_{i=1}^{j}\beta_i$. In order to carry out the same type of CFL optimization as in the previous sections,

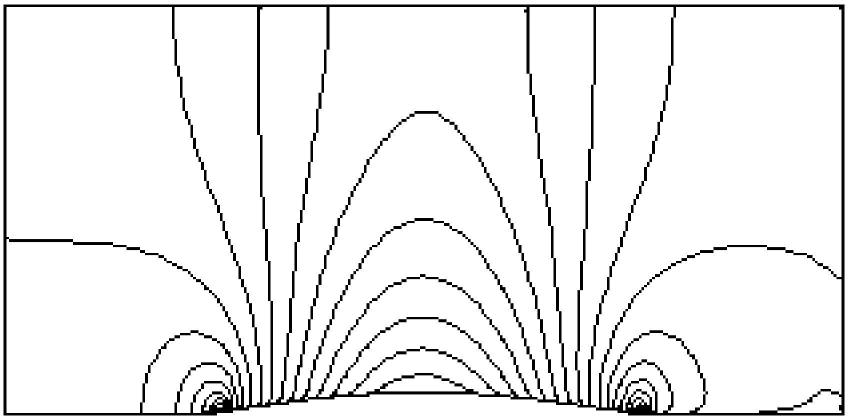


Figure 28: *The Euler equations: level curves for ρ.*

we need to express (40) as the sum of a set of vectors corresponding to successive residual evaluations. The standard approach for that is to linearize the equations. We get

$$r^{(n+1)} \approx r^{(n)} + \sum_{j=1}^{m} \alpha_j J^j r^{(n)}\,, \qquad J = \frac{\partial r^{(n)}}{\partial u^{(n)}}\,. \tag{41}$$

While the Jacobian J could be obtained analytically, given an expression for $r(u)$, a much simpler approach is to approximate that too as a linearization

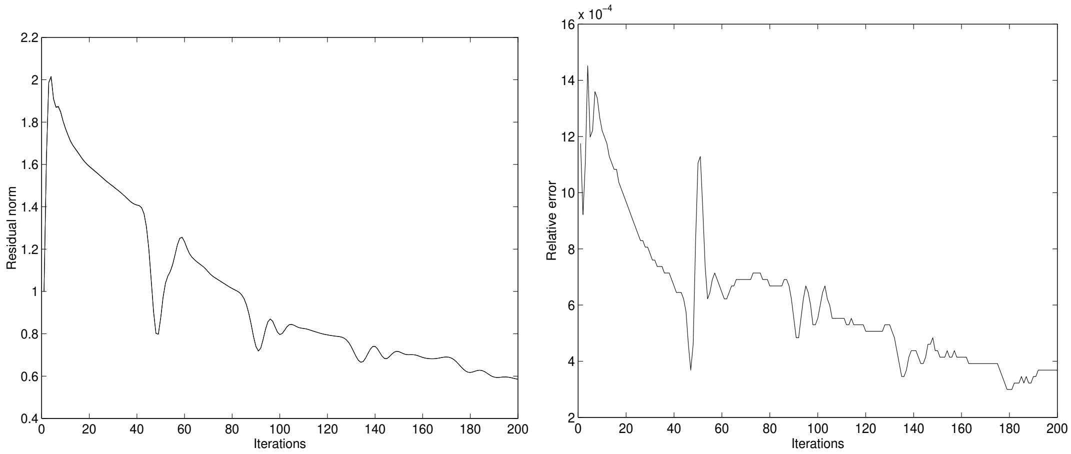
$$J^j r^{(n)} \approx (r(u^{(n)} + \sigma J^{j-1} r^{(n)}) - r^{(n)})/\sigma\,, \quad j = 1, \ldots, m \tag{42}$$

where σ is some small number. The value of $J^{j-1} r^{(n)}$ is not very sensitive to σ. In the experiments, $\sigma = 10^{-4}$ has been used.

Just like in the linear case, the vectors used in the optimization require one more residual evaluation than an ordinary RK iteration. That the linearized expression for $u^{(n+1)}$ works as a sufficiently accurate approximation in most cases can be verified experimentally. Figure 29 shows the linearized RK compared to the genuine one in (40) for the Euler equations with a Mach number of 0.5.

We would expect that the approximation of $r^{(n+1)}$ we can get as a by-product is good enough to be used in the next iteration step. This is important, as non-linear residual evaluations are generally much more expensive relative to vector sums and scalar products. Unfortunately, there is no way of "recycling" $r^{(n+1)}$ in the next step. Successive Krylov subspace vectors are computed using difference approximations in (42) rather than matrix products as in (5), which will cause possible approximation errors to grow by a factor of $1/\sigma$. Thus CFL optimization of RK will lead to an extra cost. Later on we will study the influence of this cost on overall performance.

Locally optimized RK proved fairly robust for linear problems, where there exists a constant system matrix, and a resulting iteration matrix whose eigenvalues are functions of the CFL number alone. However, it is quite possible for a



(a) *200 iterations for the Euler equations. The residual curves practically coincide.*

(b) *Difference in residual norms.*

Figure 29: *Ordinary RK compared to linearized RK.*

nonlinear problem to make the RK optimization break down and diverge unless some proper precautions are taken.

As we will see, in particular the initial phase is a sensitive period for the model problems studied in this section. The iterations do not respond to optimization the way one would expect in the long run.

A typical residual curve for the Euler equations solved by a standard RK(3) is shown in figure 30. Convergence is slow. The general trend is decreasing, and the CFL number has been experimentally shown to be stable, but there are relatively long periods where the residual *grows*. Under such conditions optimization will not perform well.

Figure 31 shows the objective function $\rho(\kappa)$ (normalized to make comparisons easier) for CFL numbers in an approximate region of interest. The same RK(3) method is used. We need to avoid $\kappa \leq 0$ to ensure the right direction of wave propagation. In the first iteration step, $\rho(\kappa)$ would have no local minimum for positive values. In fact, the higher κ the more the residual norm would increase. It is not until, approximately, the 15th iteration step that optimization in the sense that has been discussed so far would do any good. Here, there exists a local minimum on the right side of zero, but the curve still does not look anything like its linear-case counterparts. By the 25th iteration step, however, we get a $\rho(\kappa)$ that does. All positive values of κ up to a certain limit now produce a residual decrease.

For the linear cases there always existed at least a local minimum for $\kappa > 0$, which could be found either with some numerical search algorithm or by comparing $\rho(\kappa)$ at the positive roots of $\partial\rho/\partial\kappa$. In this case, the former method would

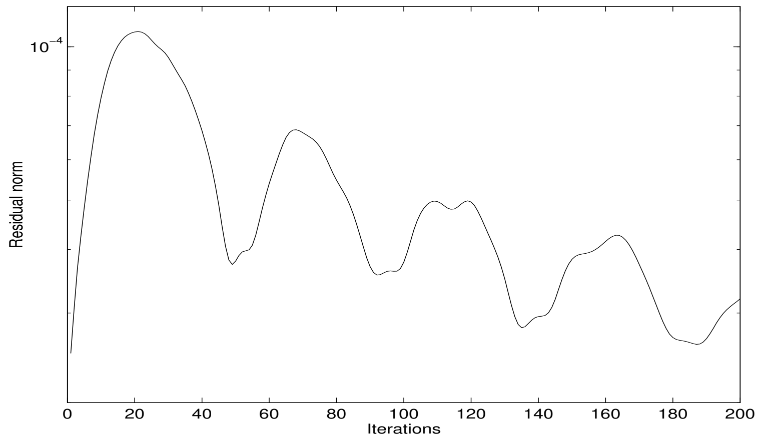


Figure 30: *Residual curve for the Euler equations on one grid.*

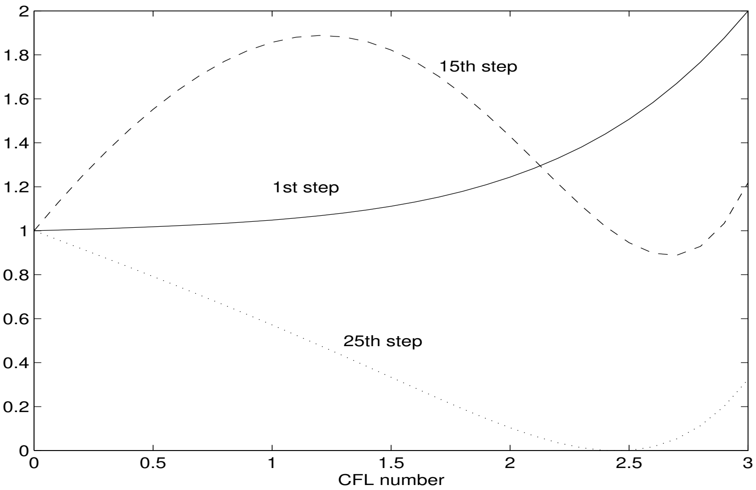


Figure 31: *Residual norms as functions of κ in different iteration steps.*

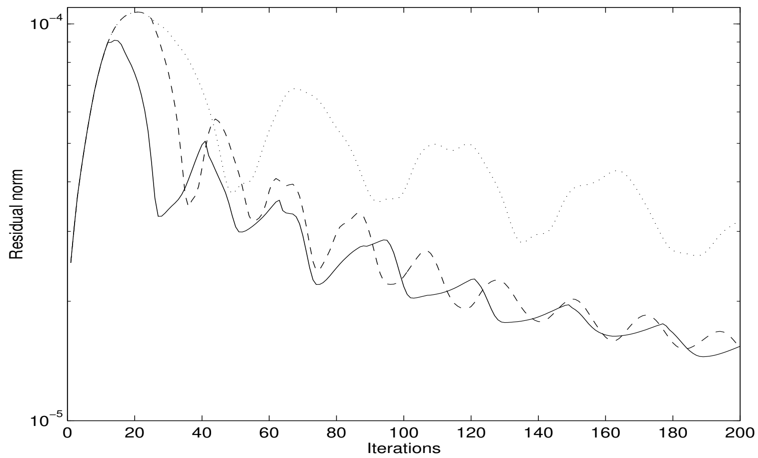


Figure 32: *Residual curve for the Euler equations on one grid. Local optimization (solid), globally optimal κ (dashed) and first guess (dotted).*

produce a result as close as possible to zero in the first step, which would produce virtually no change in the solution and we would remain at the initial state. The latter one would not work at all.

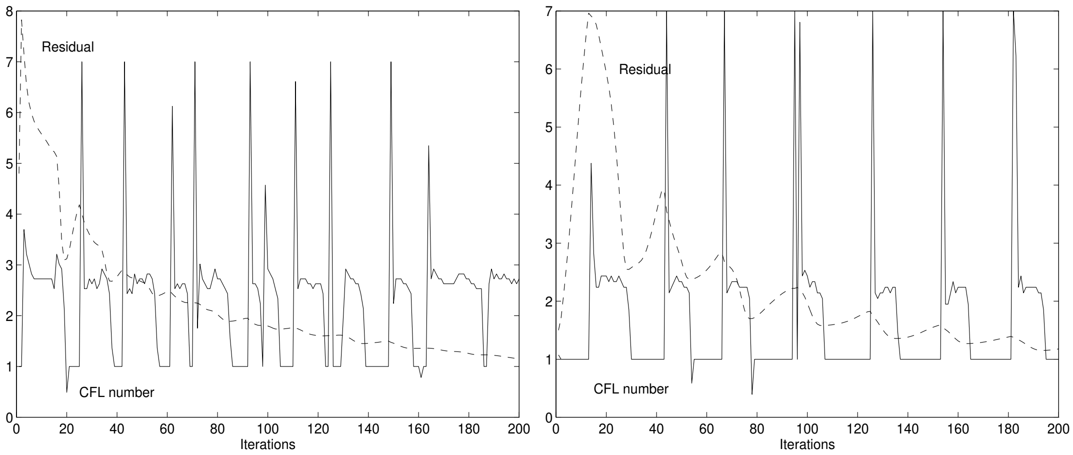
One possible remedy is to ignore the "optimal" CFL value unless there exists a local minimum for κ strictly greater than zero, and instead use a fixed, reasonably problem-independent value that at least will not cause divergence in the long run and can get the iterations over the hill, to where local optimization starts functioning again. We will choose as this constant, once and for all, $\kappa = 1.0$.

Figure 32 shows the performance of this modified optimization versus the result in figure 30 and what would be obtained using an experimentally determined globally optimal CFL number.

When the optimized CFL numbers are plotted next to (scaled) residual curves, their behavior turns out as expected. Brief periods of growing residuals are characterized by CFL numbers of 1 and as soon as the opportunity to decrease the residual norm returns, the optimization becomes active again. Figures 33 (a) and (b) show these curves for the first 200 iterations of the Euler and Navier-Stokes equations respectively (both with Mach 0.5).

3.2 Relative performance of optimized RK

As previously mentioned, the linearized and optimized RK requires one more residual evaluation than the ordinary RK, as well as a few scalar products of Krylov subspace vectors. While the latter are often negligible for larger problems, the former unavoidably hampers the performance of an m-step RK method by a factor of $m/(m+1)$. Efficient optimization should hopefully compensate for that, at least when compared to the result of picking a "generic" or "safe" parameter



(a) *The Euler equations.* (b) *The Navier-Stokes equations.*

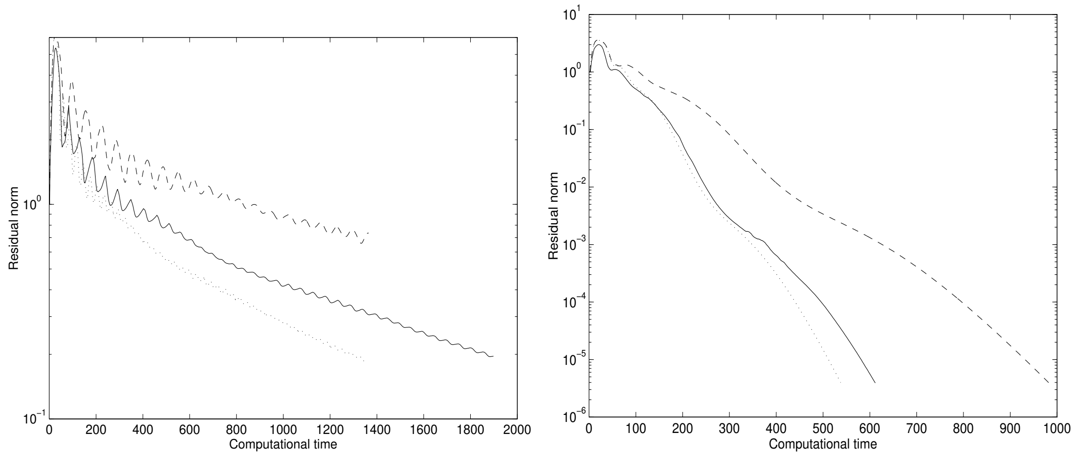
Figure 33: *Optimized CFL numbers next to convergence curves.*

value.

This appears to be the case for variations of the model problems. Figure 34 (a) shows the residual norms versus computational time for three different approaches — using linearized and locally optimized RK, ordinary RK with a standard CFL number (1.0) and the same with an experimentally determined CFL number for best convergence (2.15). After a while an approximately linear phase begins and the iterations produce a more or less constant average slope. All three approaches have been given the same number of iteration steps and the first and last are almost equal in terms of convergence per step, but clearly the optimized RK cannot match the ordinary RK with the best constant CFL number when the cost of the extra residual evaluation and the scalar products are weighed in. It does however outperform the second approach, locally and asymptotically.

Slightly better is the result shown in figure 34 (b), for a problem with multigrid iterations on *three* grids. The best constant CFL value here is 1.90. In spite of its handicap, optimization keeps up with that best constant until the residual norm has been reduced to approximately 10^{-3}. In this case the solution processes were interrupted when the residuals reached a specified limit value. With local CFL optimization it took 224 multigrid cycles, whereas the best constant CFL number needed 279 and the generic CFL number required as many as 500.

The actual values of the locally optimized CFL numbers have been plotted in figure 35 (a) and (b). The first one is consistent with our previous results for both linear and nonlinear problems on a single grid — an increasing residual norm makes the optimizer pick the safe CFL number 1.0, whereas the rest of the time the value appears concentrated very close to the experimentally determined



(a) *Subsonic flow (Mach 0.5) on one grid.* (b) *Supersonic flow (Mach 1.5) on three grids.*

Figure 34: *Optimized (solid), standard (dashed) and best (dotted) CFL numbers for the Euler equations and RK(3).*

best constant. If the extra cost of optimization had been disregarded in figure 34 (a), the solid and dotted curves would have approximately the same asymptotic slopes.

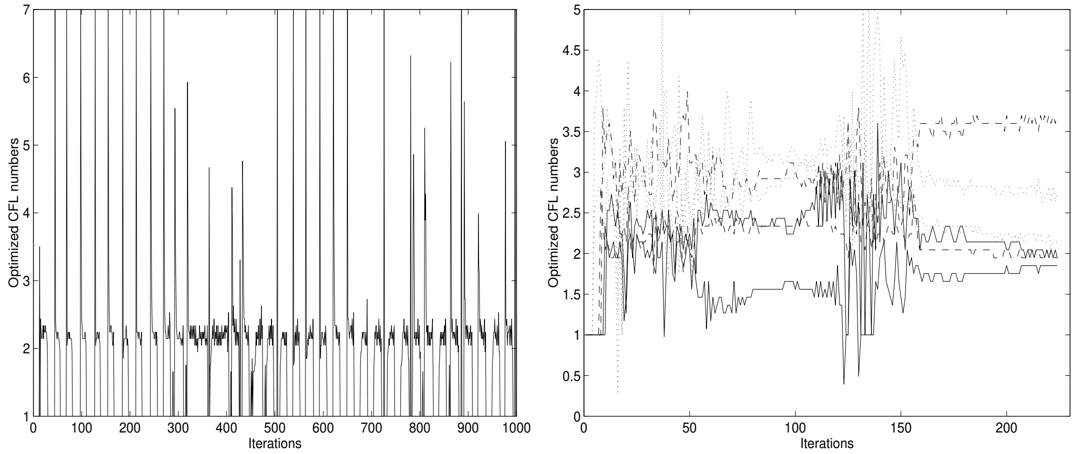
There is a similar consistency with linear multiple-grid results in figure 35 (b). Asymptotically, the CFL numbers all seem to converge (albeit to different values). That means the optimization has located the asymptotic optima, and since most of them are different from the best constant, optimization will achieve better convergence. Without the extra cost, the optimized curve in figure 34 (b) has the highest slope.

For these experiments a different optimization algorithm has been used. The "best" κ is chosen from a discrete and evenly spaced set of values in the allowed range. This, however, does not affect the results. The optimization has proved to be practically insensitive to the discretization when the step size is about 0.05 or smaller. If $\Delta\kappa$ is the desired accuracy, the cost of such an algorithm is essentially $\mathcal{O}(m/\Delta\kappa)$.

3.3 Other Krylov methods

Once having made the linearization (41) in section 3.1, we can just as easily apply any other Krylov subspace methods. The question is whether they will work for these new types of problems.

The least restricted and potentially the most efficient alternative is the GMRES method. When minimizing the norm $\|r^{(n+1)}\|$ in (41) it is clear that at least



(a) *Subsonic flow on one grid.*

(b) *Supersonic flow on three grids, pre- and post-smoothing, fine grid (solid), medium (dashed) and coarse grid (dotted).*

Figure 35: *Optimized CFL numbers for the Euler equations.*

one of $J^j r^{(n)}$, $j = 1, \dots, m$, needs to have a component in the subspace spanned by $r^{(n)}$ for there to be any parameter choice except the trivial $\alpha_1 = \alpha_2 = \ldots = \alpha_m = 0$, which does not produce any change in the solution vector $u^{(n)}$ at all.

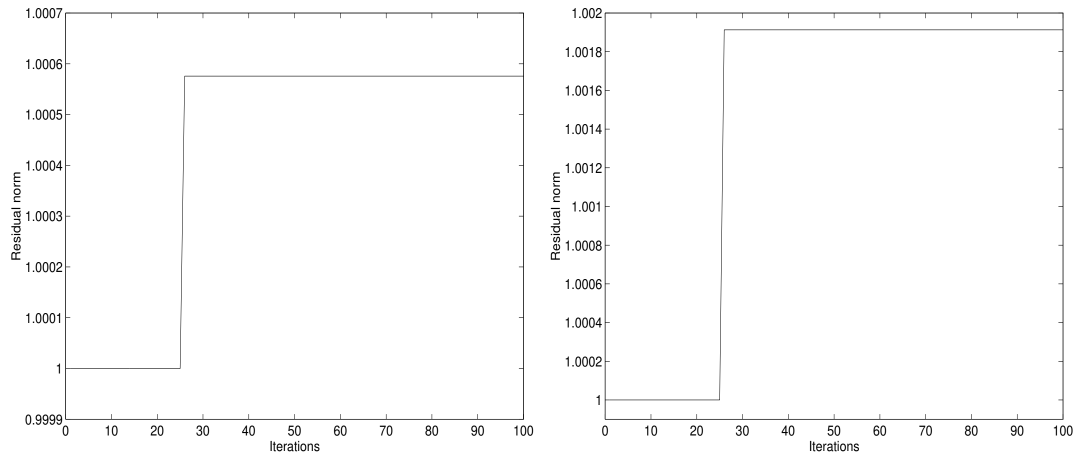
This is a consideration that did not have to be made for any of the linear test problems in the previous section. In this particular case, however, it turns out that an ordinary GMRES simply will not work. Figures 36 (a) and (b) show what happens for the Euler and Navier-Stokes equations (on a single grid) when GMRES(3) is used.

$Jr^{(n)}$, $J^2 r^{(n)}$ and $J^3 r^{(n)}$ very quickly become orthogonal to $r^{(n)}$, after which the residual will remain constant except for numerical disturbances. The following table shows the (normalized) largest scalar product between $r^{(n)}$ and any $J^i r^{(n)}$ in the first six iterations. GMRES(m) with $m > 3$ essentially gives the same results as long as $m \ll \nu$.

Problem	1	2	3	4	5	6
Euler	1.0	$2.4 \cdot 10^{-2}$	$3.1 \cdot 10^{-5}$	$8.9 \cdot 10^{-8}$	$1.6 \cdot 10^{-10}$	$1.1 \cdot 10^{-11}$
Navier-Stokes	1.0	$1.1 \cdot 10^{-1}$	$2.0 \cdot 10^{-3}$	$4.4 \cdot 10^{-5}$	$1.0 \cdot 10^{-6}$	$2.3 \cdot 10^{-8}$

Another risk is the possibility of the Krylov subspace vectors becoming linearly dependent, but this can be countered by performing a singular value decomposition prior to solving the normal equations.

The modified GMRES method works much better, due to its fixed wave speed.



(a) *The Euler equations.* (b) *The Navier-Stokes equations.*

Figure 36: *Residual curves resulting from GMRES(3), subsonic flow.*

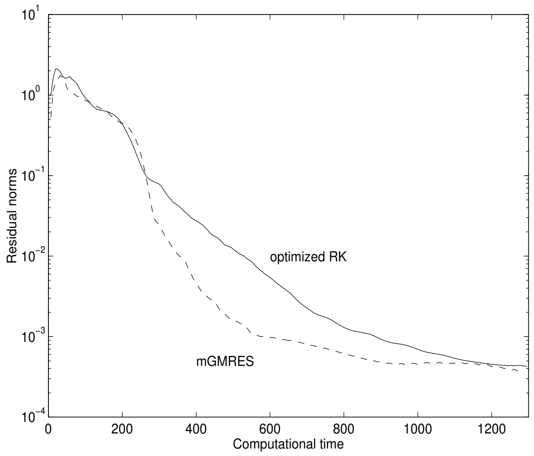
That gets it past temporarily increasing residuals. A comparison between mGMRES(3) and optimized RK(3) on three grids for the subsonic (Mach 0.5) Euler and Navier-Stokes equations respectively is shown in figure 37.

Apparently, mGMRES can keep up with the optimized RK and even outperform it, but for that it needs an experimentally determined (roughly) optimal wave speed number. The hybrid methods, on the other hand, are parameter-independent. A test of those under the same conditions is shown in figure 38. *Hybrid 1,* which has much in common with GMRES, displays the slowest convergence, while *hybrid 2,* which is more similar to mGMRES, gets a slightly better result. It is conceivable, though, that *hybrid 1* would perform better than the others if the GMRES method had worked well.

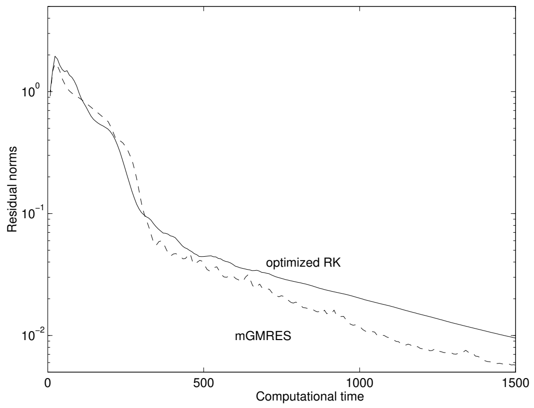
4 Conclusions

We have developed a way of optimizing the Runge-Kutta method with respect to the CFL number, at a relatively small cost. This optimization combines long-term optimality with rapid convergence during an initial phase and nearly always outperforms the original, even if the best constant CFL number happens to be known in advance (which normally is not the case).

Combinations with the modified Generalized Minimum RESidual method have been introduced to avoid the need for determining the remaining coefficients and result in performance often on par with the best of the commonly used Krylov subspace methods — even when these are supplied with experimentally determined optimal parameters.

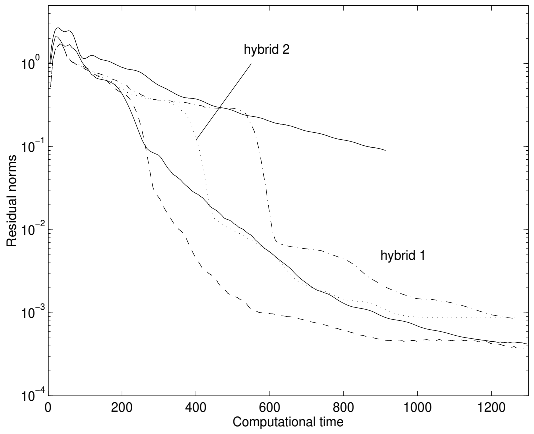


(a) *The Euler equations*

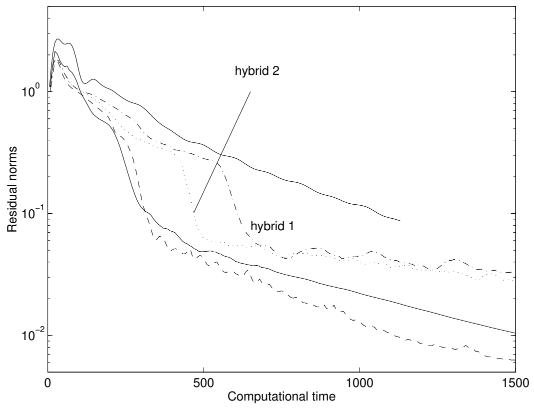


(b) *The Navier-Stokes equations*

Figure 37: *Multi-grid iterations — optimized RK (solid) versus the modified GMRES method (dashed).*



(a) *The Euler equations*



(b) *The Navier-Stokes equations*

Figure 38: *Multi-grid iterations ordinary RK (solid, upper), optimized RK (solid,lower) and modified GMRES (dashed) compared to the hybrid methods.*

The hybrid methods work for nonlinear cases as well and can be used when safe elimination of parameters is worth the cost of less than optimal performance. Their advantage over the usual parameter-free alternative GMRES is robustness. Non-zero wave speeds are guaranteed.

Similarly, the optimized RK (with the CFL limit modification) can be used in place of the ordinary RK when the stability region is unknown. Although the cost of optimization — dominated by one extra residual plus a few scalar products — generally prevents it from matching the performance of the ordinary RK with an optimal CFL number, it is likely to outperform "safe" guesses, especially on multiple grids, where it has the advantage of greater flexibility.

Acknowledgments

The nonlinear numerical experiments were made using Euler and Navier-Stokes equation solver software developed by Rickard Enander, Dept. of Scientific Computing, Uppsala University.

References

[1] Y. Saad and M. Schultz, GMRES: a generalized minimum residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* 7 (1986) 865-869.

[2] B. Gustafsson and P. Lötstedt, The GMRES method improved by securing fast wave propagation, *Appl. Num. Math.* 12 (1993) 135-152.

[3] P. Lötstedt, Improved Convergence to the Steady State of the Euler Equations by Enhanced Wave Propagation, *J. Comput. Phys.,* Vol. 114, No. 1 (1994) 34-44.

[4] T. A. Manteuffel, The Tchebyshev Iteration for Nonsymmetric Linear Systems, *Numer. Math.* Vol. 28, No. 3 (1977) 307-328.

[5] P. N. Brown and Y. Saad, Convergence Theory of Nonlinear Newton-Krylov Algorithms, *SIAM J. Opt.,* Vol. 4, No. 2 (1994) 297-330

[6] P. N. Brown and Y. Saad, Hybrid Krylov Methods for Nonlinear Systems of Equations, *SIAM J. Sci. Stat. Comput.,* Vol. 11, No. 3 (1990) 450-481

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

[8] O. Nevanlinna, Convergence of iterations for linear equations, Birkhäuser, 1993.

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

[10] B. Koren, Multigrid and defect correction for the steady Navier-Stokes equations, *J. Comput. Phys.,* 87 (1990), 25-46.

[11] W. Couzy, Note on Damping Outflow Conditions, Automatic Time Stepping and Upwind Discretization, *Dept. of Mech. Eng., École Polytechnique Fédérale de Lausanne,* Report T-96-11, Lausanne, March 1996.