INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2009; **60**:591–609
Published online 15 September 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1902

Exponential time integration using Krylov subspaces

J. C. Schulze[1,*,†], P. J. Schmid[2] and J. L. Sesterhenn[1]

[1]*Department for Numerical Methods in Aero/Astronautics, University of the German Armed Forces, Munich, Germany*
[2]*Laboratoire d'Hydrodynamique (LadHyX), Ecole Polytechnique, Palaiseau, France*

SUMMARY

The application of exponential integrators based on Krylov techniques to large-scale simulations of complex fluid flows with multiple time-scales demonstrates the efficiency of these schemes in reducing the associated time-step restrictions due to numerical stiffness. Savings of approximately 50% can be achieved for simulations of the three-dimensional compressible Navier–Stokes equations while still maintaining a truncation error typical of explicit time-stepping schemes. Exponential time integration techniques of this type are particularly advantageous for fluid flows with a wide range of temporal scales such as low-Mach number, reactive or acoustically dominated flows. Copyright © 2008 John Wiley & Sons, Ltd.

Received 3 December 2007; Revised 15 July 2008; Accepted 15 July 2008

KEY WORDS: Krylov subspaces; exponential integration

1. INTRODUCTION

The presence of physical processes acting on vastly different temporal scales is a commonplace phenomenon in many applications in science and engineering. The propagation of acoustic waves in a highly turbulent compressible flow, the treatment of low-Mach number flows, or the combustion of reactive gases initiated by a premixed shear flow present only three of many applications where processes with a fast characteristic speed coexist with fluid motion on a significantly reduced temporal scale. It is needless to say that this type of flow configurations poses great challenges for computational resources as well as numerical methods. The implicit treatment of multi-scale processes is often prohibitively expensive, while an explicit temporal evolution is plagued by severe time-step restrictions in order to maintain numerical stability.

*Correspondence to: J. C. Schulze, Department for Numerical Methods in Aero/Astronautics (LRT1), University of the German Armed Forces (UniBw) Munich, Werner-Heisenberg-Weg 39, D-85579 Neubiberg, Germany.
†E-mail: jan.schulze@unibw.de

Contract/grant sponsor: Alexander-von-Humboldt Foundation

As a representative example of flows with a vastly varying range of characteristic velocities the simulation of low-Mach number flows will be considered. For this case, the acoustic waves determine the time-step resulting in a critical CFL-number that scales inversely proportional to the (low) Mach number: CFL$\sim 1/Ma$. For example, for a given *computational* CFL-number of one, the simulation of a low-Mach number flow with $Ma = 10^{-3}$ would require an *effective* CFL-number of the order of 10^{-3}. With a time-step restriction this severe, the demands on computational resources and CPU-times increase drastically. Any improvement of this restriction, even by a modest factor, would be welcome and yields a more efficient time-evolution scheme.

In many cases, the source of the temporal stiffness lies in the linear terms of the governing equations. In this case, a more accurate description of the linear solution operator (the matrix exponential for the autonomous case) has the potential to greatly alleviate the time-step restrictions. Due to the large number of degrees of freedom in high-performance fluid simulations, the explicit calculation (and formation) of the matrix exponential is impractical; rather, a significantly lower-dimensional approximation based on iterative techniques is sought. This approximation represents the matrix exponential in a low-dimensional Krylov space using a polynomial ansatz. The Krylov subspace is formed by a repeated application of the linear (or linearized) system matrix to a given starting vector. In this Krylov basis, the matrix exponential can then be computed by direct methods.

The method of integrating a general system of ordinary differential equations (ODEs) using Krylov subspaces techniques is by no means new. It has been shown [1] that various techniques can be devised that provide approximations of different accuracy and error bounds. A comprehensive review of the techniques and their implementation is given in [2–4]. Specific applications to fluid dynamic equations, in particular, the incompressible Navier–Stokes equations, have been reported by e.g. [5] or [6] where the linear system matrix has been extracted from a finite-elements discretization combined with a projection method. Numerical software for the implementation of exponential integrators based on Krylov subspaces is also available: the *EXPOKIT*-package provides various implementation and error estimates [4].

Although the general techniques have been developed and implemented for general systems of ODEs and for simplified (incompressible) flow situations, the exponential integration technique based on Krylov subspaces has not yet found wide-spread use in more challenging (compressible) fluid simulations where multi-scale effects combined with commonly used explicit techniques represent the bottleneck for more efficient large-scale simulations. This paper introduces and showcases exponential time-integration to challenging large-scale fluid simulations and demonstrates its ability to overcome inherent time-step restrictions. Special emphasis will be put on a fair comparison of gained efficiency versus exerted effort.

As large-scale simulations of complex fluid motion on high-performance computers become not only feasible but also more common, the improvement of stiffness constraints by exponential time integration represents an important component in extending the envelope of achievable numerical investigations and, therefore, a tool for treating more realistic flow configurations.

The organization of the paper is as follows: in the second section, we give a brief description of the underlying theory of exponential time-stepping and its practical implementation; the third section will comprise applications in increasing complexity, starting with a simple one-dimensional linear and nonlinear equation, followed by a two-dimensional simulation of vortex merging in a compressible shear layer, culminating in a fully three-dimensional computation of a subsonic, viscous jet.

DOI: 10.1002/fld

2. GOVERNING THEORY

The exponential time integration based on Krylov subspaces is based on a low-dimensional approximation of the matrix exponential which represents the exact evolution operator for an autonomous linear system. For inhomogeneous problems minor modifications have to be implemented, as will be shown later. To start, however, we will consider the simplest case: the linear and autonomous problem.

2.1. Linear equations

The solution of the linear and autonomous problem

$$\frac{\mathrm{d}\mathbf{u}(t)}{\mathrm{d}t} = \mathbf{A}\mathbf{u}(t) + \mathbf{q}(t) \tag{1a}$$

$$\mathbf{u}(0) = \mathbf{v} \tag{1b}$$

with system matrix $\mathbf{A}$, state vector $\mathbf{u}$, initial condition $\mathbf{v}$ and external driving $\mathbf{q}$ can formally be written as

$$\mathbf{u}(t) = \mathrm{e}^{t\mathbf{A}}\mathbf{v} + \int_0^t \mathrm{e}^{(t-s)\mathbf{A}}\mathbf{q}(s)\,\mathrm{d}s \tag{2}$$

The first term involves the matrix exponential and represents the homogeneous part of the solution, whereas the second term consists of a convolution integral, again involving the matrix exponential, and represents the particular part of the full solution.

We will first consider the homogeneous case, i.e. the case where $\mathbf{q}=0$, for which we need to compute the matrix exponential applied to a vector, $\mathrm{e}^{t\mathbf{A}}\mathbf{v}$. If $\mathbf{A}$ is a bounded operator, which is the case for most fluid dynamic equations, the matrix exponential can be defined by a Taylor power series. For an unbounded operator, like an elliptic operator, the meaning of $\mathbf{A}$ comes from the mathematical theory of semigroups. A detailed discussion of this topic can be found in [7]. Since the present research project focuses on fluid dynamic equations, the solution of the matrix exponential will be understood as an expansion series for the remainder of this discussion, like

$$\mathrm{e}^{t\mathbf{A}}\mathbf{v} = \sum_{n=0}^{\infty} \frac{1}{n!}(t\mathbf{A})^n\mathbf{v} \tag{3}$$

A straightforward approach to construct $\mathrm{e}^{t\mathbf{A}}$ is thus a truncated form of (3). This is in fact the underlying principle of a Runge–Kutta (or any linear multi-step) scheme; in general, Runge–Kutta schemes approximate the first few terms of the above Taylor series expansion, but are not a recommended method—according to [8]—to approximate the matrix exponential. Better approximations can be found if we seek a general polynomial approximation of the product of the matrix exponential and a given vector of the form

$$\mathrm{e}^{t\mathbf{A}}\mathbf{v} \approx p_{m-1}(t\mathbf{A})\mathbf{v} \tag{4}$$

with p_{m-1} denoting a general $(m-1)$-degree polynomial.

This approximation problem can be recast by introducing a m-dimensional Krylov subspace based on the starting vector $\mathbf{v}$ and the system matrix $\mathbf{A}$

$$\mathscr{K}_m = \mathrm{span}\{\mathbf{v}, \mathbf{A}\mathbf{v}, \ldots, \mathbf{A}^{m-1}\mathbf{v}\} \tag{5}$$

DOI: 10.1002/fld

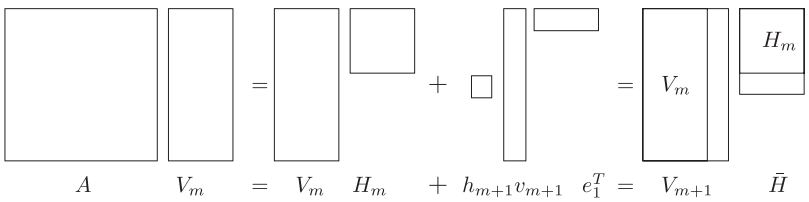

Figure 1. Arnoldi decomposition.

ALGORITHM 1: ARNOLDI DECOMPOSITION

$\beta = \|\mathbf{v}\|_2$
$\mathbf{v}_1 = \mathbf{v}/\beta$
do $j = 1, m$
  $\mathbf{p} = \mathbf{A}\mathbf{v}_j$
  do $i = 1, j$
    $h_{i,j} = \mathbf{v}_i^{\mathrm{T}}\mathbf{p}$
    $\mathbf{p} = \mathbf{p} - h_{i,j}\mathbf{v}_i$
  end do
  $h_{j+1,j} = \|\mathbf{p}\|_2$
  $\mathbf{v}_{j+1} = \mathbf{p}/h_{j+1,j}$
end do

Our problem thus reduces to finding an element in the Krylov subspace $\mathscr{K}_m$ that best approximates the solution to the homogeneous initial-value problem. To efficiently manipulate the elements of the Krylov subspace we generate an orthonormal basis $\mathbf{V}_m = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$ of $\mathscr{K}_m$. This orthonormal basis is obtained by employing the well-known Arnoldi algorithm starting with the initial vector $\mathbf{v}_1 = \mathbf{v}/\beta$ where $\beta = \|\mathbf{v}\|_2$. Besides the orthonormal basis $\mathbf{V}_m$, the Arnoldi algorithm also produces an upper $m \times m$ Hessenberg-matrix $\mathbf{H}_m$ which satisfies the relation (see also Figure 1):

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + h_{m+1,m}\mathbf{v}_{m+1}\mathbf{e}_1^{\mathrm{T}} \tag{6}$$

where for a given k, $\mathbf{e}_k$ denotes the kth unit vector belonging to $\mathbb{R}^m$, and h_{ij} stands for an element of the matrix $\mathbf{H}_m$.

The complete Arnoldi algorithm is outlined in Algorithm 1. For the computation of the matrix $\mathbf{H}_m$ and the orthonormalized basis $\mathbf{V}_m$ only matrix–vector and scalar products have to be evaluated. As an input the algorithm requires the dimension of the Krylov subspace m, the initial vector $\mathbf{v}$ and the system matrix $\mathbf{A}$. Alternatively, the algorithm can be rewritten in a matrix-free form such that only the matrix–vector product $\mathbf{A}\mathbf{v}$ is used without explicitly forming or evaluating the system matrix $\mathbf{A}$.

Since we assume an orthonormal basis $\mathbf{V}_m$ of $\mathscr{K}_m$, we get $\mathbf{H}_m \approx \mathbf{V}_m^{\mathrm{T}}\mathbf{A}\mathbf{V}_m$; in other words, $\mathbf{H}_m$ is a projection of the (large) system matrix $\mathbf{A}$ onto the Krylov subspace $\mathscr{K}_m$ using the orthonormalized basis $\mathbf{V}_m$. We can then form the vector $\mathbf{u}_{\mathrm{opt}} = \mathbf{V}_m\mathbf{V}_m^{\mathrm{T}}\mathrm{e}^{t\mathbf{A}}\mathbf{v}$ which is a projection of $\mathrm{e}^{t\mathbf{A}}\mathbf{v}$ onto $\mathscr{K}_m$. This vector is the best approximation of $\mathrm{e}^{t\mathbf{A}}\mathbf{v}$ among the elements of the Krylov subspace $\mathscr{K}_m$.

DOI: 10.1002/fld

Furthermore, a reformulation of the optimal vector, using the definition of the orthonormal basis $\mathbf{V}_m$, leads to

$$\mathbf{u}_{\text{opt}} = \mathbf{V}_m \mathbf{V}_m^{\mathrm{T}} \mathrm{e}^{t\mathbf{A}} \mathbf{v} = \beta \mathbf{V}_m \mathbf{V}_m^{\mathrm{T}} \mathrm{e}^{t\mathbf{A}} \mathbf{V}_m \mathbf{e}_1 \tag{7}$$

The optimal vector still depends on the matrix exponential $\mathrm{e}^{t\mathbf{A}}$ based on the (large) matrix $\mathbf{A}$, which is in practice unobtainable. However, owing to the orthogonality of the basis $\mathbf{V}_m$, we can introduce the following approximation

$$\mathbf{V}_m^{\mathrm{T}} \mathrm{e}^{t\mathbf{A}} \mathbf{V}_m \approx \mathrm{e}^{t\mathbf{H}_m} \tag{8}$$

and finally rewrite the vector $\mathbf{u}_{\text{opt}}$ as

$$\mathbf{u}_{\text{opt}} \approx \beta \mathbf{V}_m \mathrm{e}^{t\mathbf{H}_m} \mathbf{e}_1 \tag{9}$$

or, with the improved approximation including the residual in Equation (6) with $h_{m+1,m}$,

$$\mathbf{u}_{\text{opt}} \approx \beta \mathbf{V}_{m+1} \mathrm{e}^{t\mathbf{H}_{m+1}} \mathbf{e}_1 \tag{10}$$

The original problem—which was to compute $\mathrm{e}^{t\mathbf{A}}\mathbf{v}$ with $\mathbf{A} \in \mathbb{R}^{n\times n}$—has been replaced by evaluating $\mathrm{e}^{t\mathbf{H}_m}\mathbf{v}$ with $\mathbf{H}_m \in \mathbb{R}^{m\times m}$ and $m \ll n$. For the reduced problem, the matrix exponential can be computed using highly accurate Padé methods [8].

It should be mentioned that the Arnoldi algorithm merely requires the *product* of $\mathbf{A}$ and $\mathbf{v}$ to compute $\mathbf{V}_m$ and $\mathbf{H}_m$; we thus do not explicitly need to know the system matrix $\mathbf{A}$—which is the case for many technical applications.

Contrary to the formulation above, the solution $\mathbf{u}(t)$ at a given time t is not computed directly, but rather advanced from the initial condition $\mathbf{v}$ via a time-stepping algorithm. The problem to be treated by the exponential time integration thus reduces to an initial-value problem over one time-step $[t_n, t_n + \tau]$

$$\mathbf{u}(0) = \mathbf{v} \tag{11a}$$

$$\mathbf{u}(t_n + \tau) = \mathrm{e}^{(t_n+\tau)\mathbf{A}} \mathbf{v} = \mathrm{e}^{\tau\mathbf{A}} \mathbf{u}(t_n) \tag{11b}$$

For various flow configurations the system of governing equations is nonhomogeneous, i.e. the source term $\mathbf{q}$ in (1a) is nonzero. Examples include reacting fluids where the combustion terms are modeled as temperature-dependent source terms, or numerical boundary conditions such as an outflow sponge term that applies a highly dissipative force near the boundary which ensures that no (or negligible) energy is scattered back into the computational domain. The general solution for this problem is given in (2) and can be rewritten using the solution of the integral term

$$\mathbf{u}(t) = \mathrm{e}^{t\mathbf{A}} \mathbf{v} + t\,\varphi(t\mathbf{A})\mathbf{q} \tag{12}$$

where $\varphi(z) = (\mathrm{e}^z - 1)/z$. The resulting method

$$\mathbf{u}_{n+1} = \mathrm{e}^{\tau\mathbf{A}} \mathbf{u}_n + \tau\varphi(\tau\mathbf{A})\mathbf{q}(t_n), \quad t_{n+1} = t_n + \tau \tag{13}$$

is often called the (explicit) exponential Euler method. Exponential time-integration methods of Runge–Kutta type are studied in [9].

DOI: 10.1002/fld

If we use the following manipulation:

$$\begin{aligned}\mathbf{u}(t_n+\tau) &= \mathrm{e}^{\tau\mathbf{A}}\mathbf{u}(t_n)+\tau\varphi(\tau\mathbf{A})\mathbf{q}(t_n)\\ &= (\tau\mathbf{A}\varphi(\tau\mathbf{A})+\mathbf{I})\mathbf{u}(t_n)+\tau\varphi(\tau\mathbf{A})\mathbf{q}(t_n)\\ &= \tau\varphi(\tau\mathbf{A})(\mathbf{A}\mathbf{u}(t_n)+\mathbf{q}(t_n))+\mathbf{u}(t_n)\end{aligned} \tag{14}$$

the only remaining expensive operation for the inhomogeneous case is to evaluate $\varphi(\tau\mathbf{A})\mathbf{v}$ which can be approximated in a similar manner as $\mathrm{e}^{\tau\mathbf{A}}\mathbf{v}$. We therefore obtain

$$\varphi(\tau\mathbf{A})\mathbf{v}\approx\beta\mathbf{V}_m\varphi(\tau\mathbf{H}_m)\mathbf{e}_1 \tag{15}$$

The Krylov subspace approximation to $\varphi(\tau\mathbf{A})\mathbf{v}$ converges toward the exact solution as fast as $\mathrm{e}^{\tau\mathbf{A}}\mathbf{v}$ converges to its exact equivalent, see [10].

Following the formulation in [3], the function $\varphi(\mathbf{Z})$ of a matrix argument $\mathbf{Z}$—or, more precisely, the product $\varphi(\mathbf{Z})\mathbf{e}_1$—can be computed using only one matrix exponential

$$\exp\begin{pmatrix}\mathbf{Z} & \mathbf{e}_1\\ \mathbf{0} & 0\end{pmatrix}=\begin{pmatrix}\exp(\mathbf{Z}) & \varphi(\mathbf{Z})\mathbf{e}_1\\ \mathbf{0} & 1\end{pmatrix} \tag{16}$$

This expression can be obtained by explicitly deriving the matrix exponential by a Taylor series expansion (cf. also [3]). As shown in [2], the approximation error can be bounded by

$$\|\mathrm{e}^{\tau\mathbf{A}}\mathbf{v}-\beta\mathbf{V}_m\mathrm{e}^{\tau\mathbf{H}_m}\mathbf{e}_1\|_2\leqslant 2\beta(\tau\rho)^m\phi(\mu(\tau\mathbf{A})) \tag{17}$$

where ρ stands for the two-norm of $\mathbf{A}$, i.e. $\rho=\|\mathbf{A}\|_2$, and $\mu(.)$ denotes the logarithmic norm defined as

$$\mu(\mathbf{B})=\lim_{h\to 0+}\frac{\|\mathbf{I}+h\mathbf{B}\|_2-1}{h} \tag{18}$$

The function ϕ in the above expression is given by

$$\phi(\eta)=\frac{1}{\eta^m}\left(\mathrm{e}^{\eta}-\sum_{k=0}^{m-1}\frac{\eta^k}{k!}\right) \tag{19}$$

As we can see, the error depends on the step size τ and will, therefore, grow with the chosen CFL-number. Some further error bounds are reported, for instance, in [1, 3]. The mathematical basis for these Krylov approximations to the matrix exponential has been documented in [1, 2].

2.2. Error bounds and error control

Since the error of the Krylov subspace time-integration increases with the time-step (CFL-number), we would like to have an *a posteriori* error estimate. Different error estimates have been suggested in [4], all of them depending on the last term in Equation (6) or the last entry of the Hessenberg matrix $\tilde{\mathbf{H}}$, i.e. $h_{m+1,m}$. This entry represents the residual of the projection onto the Krylov subspace:

$$\mathrm{err}=\beta|h_{m+1,m}\tau\mathbf{e}_m^{\mathrm{T}}\varphi(\tau\mathbf{H}_m)\mathbf{e}_1| \tag{20}$$

The above error bound can be computed with little extra cost since it contains a side-product of the Arnoldi process as well as the matrix exponential. Once a maximum error threshold is reached,

DOI: 10.1002/fld

two possible interventions are conceivable. Most straightforward is the reduction of the time-step, since this reduces the error proportionately. This choice, however, affects the CFL-number (which we prefer to keep constant). The second possibility is to increase the dimensionality of the Krylov subspace until the error falls below a predefined threshold. This second option, which is used for all the numerical examples in Section 3, has to take into account the available computer memory to avoid overflow.

2.3. Nonlinear equations

The exponential time-integration technique is based on a linear problem given by the system matrix $\mathbf{A}$. However, nonlinear governing equations are well within the reach of this technique. In particular, we want to apply the Krylov time-integration method to the nonlinear compressible Navier–Stokes equations, which calls for some small modifications. First, the equations have to be linearized about the current state vector, and, secondly, we need an approximation for the resulting Jacobian matrix $\mathbf{A}$.

We consider the general nonlinear initial-value problem

$$\frac{\mathrm{d}\mathbf{u}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{u}(t)) \tag{21}$$

A Taylor expansion of the right-hand side about a given state $\mathbf{u}(t_n)$ yields

$$\mathbf{f}(\mathbf{u}(t)) = \mathbf{f}(\mathbf{u}(t_n)) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{u}(t_n))(\mathbf{u}(t) - \mathbf{u}(t_n)) + \mathbf{r}(\mathbf{u}(t)) \tag{22}$$

with the remainder term $\mathbf{r}(\mathbf{u}(t))$ [11] and the Jacobian matrix

$$\mathbf{A} \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{u}(t_n)) \tag{23}$$

The nonlinear governing equation can then be rewritten in the form

$$\frac{\mathrm{d}\mathbf{u}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{u}(t_n)) + \mathbf{A}(\mathbf{u}(t) - \mathbf{u}(t_n)) + \mathbf{r}(\mathbf{u}(t)) \tag{24}$$

Introducing an integrating factor $\mathrm{e}^{-t\mathbf{A}}$ in Equation (24) and subsequently integrating over the interval $t \in [t_n, t_n + \tau]$, the solution $\mathbf{u}(t_n + \tau)$ at the new time-step yields

$$\mathbf{u}(t_n + \tau) = \mathbf{u}(t_n) + (\mathrm{e}^{\tau\mathbf{A}} - \mathbf{I})\mathbf{A}^{-1}\mathbf{f}(\mathbf{u}(t_n)) + \int_{t_n}^{t_n+\tau} \mathrm{e}^{(t_n+\tau-s)\mathbf{A}}\mathbf{r}(\mathbf{u}(s))\,\mathrm{d}s \tag{25}$$

Equation (25) still depends on $\mathbf{A} \in \mathbb{R}^{n\times n}$ and represents an exact solution to problem (21). The integral term in Equation (25) has to be evaluated numerically. Different methods based on quadrature rules like Runge–Kutta and multi-step methods are presented, e.g. in [1, 11]. Solving this integral term numerically will enhance the accuracy of the time-integration method at the expense of an increase in computational time. More precisely, depending on the specific case and the chosen approximation method for the integral, the computational time for the solution of the integral roughly matches the computational time for the approximated matrix exponential [11]. This raises the question whether the inclusion of this integral term is necessary for our specific fluid dynamic equations. Our numerical experiments in Section 3 will show that the error caused by omitting

DOI: 10.1002/fld

the integral is comparable to the error induced by the standard explicit Runge–Kutta three-step method. Methods of this type are referred to as *exponentially fitted Euler methods* [1] or *exponential Rosenbrock–Euler methods* [12].

Using the same φ-function from the linear, nonhomogeneous case, $\varphi(z)=(e^z-1)/z$, and setting the integral to zero, we can rewrite Equation (25) and give an approximation for the solution at the new time-step as

$$\mathbf{u}(t_n+\tau)\approx\mathbf{u}(t_n)+\tau\varphi_1(\tau\mathbf{A})\mathbf{f}(\mathbf{u}(t_n)) \tag{26}$$

Employing the approximation of the matrix exponential in the φ-function, we obtain

$$\mathbf{u}(t_n+\tau)\approx\mathbf{u}(t_n)+\tau\beta\mathbf{V}_{m+1}\varphi_1(\tau\bar{\mathbf{H}}_{m+1})\mathbf{e}_1 \tag{27}$$

including the improved approximation with $\bar{\mathbf{H}}_{m+1}=[\bar{\mathbf{H}}\ \ 0]$.

The Jacobian $\mathbf{A}=(\partial\mathbf{f}/\partial\mathbf{u})(\mathbf{u}(t_n))$ of the system can be approximated using a first Fréchet derivative. However, we choose to implement the Arnoldi algorithm in a matrix-free form; we thus only need the matrix–vector product $\mathbf{Av}$ (see Algorithm 1) which can straightforwardly be approximated by

$$\mathbf{Av}\approx\frac{\mathbf{f}(\mathbf{u}+\varepsilon\mathbf{v})-\mathbf{f}(\mathbf{u})}{\varepsilon} \tag{28}$$

A higher-order approximation for this first derivative (for example, a second-order accurate central difference) could also be implemented. Our numerical tests, however, showed that a second-order accurate approximation of the Jacobian did not increase the overall accuracy of the simulation. On the contrary, the evaluation of an additional right-hand side causes extra computational costs. For this reason, the numerical examples in the following section use a first-order accurate derivative to approximate the Jacobian.

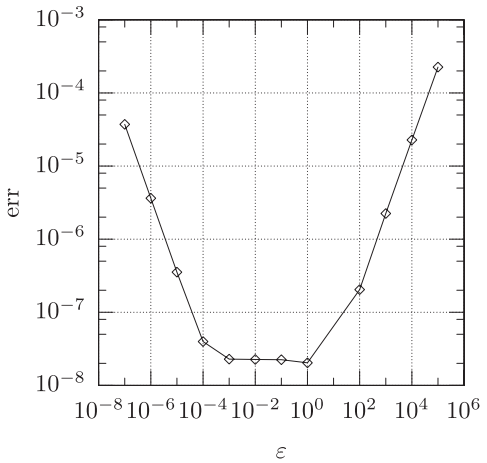

Figure 2. Dependence on ε of the error in evaluating the Jacobian via a finite-difference approximation, tested for a two-dimensional direct numerical simulation of the compressible Navier–Stokes equations with approximately 150 000 grid points.

DOI: 10.1002/fld

The choice of ε in Equation (28) can be influential on the stability of the simulation, in particular, for large systems, and has to be adapted judiciously for each case. For the numerical simulations presented in the next section, we chose the parameter ε according to

$$\varepsilon = \|\mathbf{u}\|_2 \sqrt{\varepsilon_{\text{machine}}} \tag{29}$$

which yielded satisfactory results. Figure 2 displays the typical dependence of the error on the choice of ε. For large values of ε, the error is dominated by the truncation error associated with Equation (28), whereas for exceedingly small values of ε round-off errors prevail.

To conclude this section, it should be mentioned that we focus in the present study on the compressible Navier–Stokes equations where the presence of a mass matrix $\mathbf{M}$ is not relevant ($\mathbf{M}\mathrm{d}\mathbf{u}(t)/\mathrm{d}t = \mathbf{A}\mathbf{u}(t)$). In the case of the incompressible Navier–Stokes equations where the mass matrix is not invertible, the equations need to be rewritten. Examples for these strategies can be found e.g. in [5] or [13].

3. NUMERICAL APPLICATIONS

The following section applies the Krylov time integration to three test cases of varying degrees of difficulty. We start with the one-dimensional, nonlinear, viscous Burgers equation, where we showcase the results of the exponential time-integration and compare them to other commonly used explicit time-marching methods. The special case of a linearized Burgers equation, i.e. a simple advection–diffusion equation, is also included in this section. Next, we investigate the performance of the Krylov time-integration to fluid dynamic problems. These results are obtained by solving the two-dimensional compressible Navier–Stokes equations. Finally, we present results of a fully three-dimensional and turbulent compressible Navier–Stokes computation of a subsonic jet including its acoustic near-field.

3.1. One-dimensional Burgers equation

Our first test problem deals with the homogeneous, one-dimensional, viscous Burgers equation given by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} = 0 \tag{30}$$

with $Re = 20$ and the initial condition $u(x,0) = \alpha \sin(2\pi x/L_x) + u_0$.

We start by evaluating the error-norm $\text{err} = \|\mathbf{u}_{\text{method}} - \mathbf{u}_{\text{ref}}\|_2$ versus the CFL-number. As reference solution $\mathbf{u}_{\text{ref}}$ we take the solution of the Runge–Kutta method for a small CFL-number of $\text{CFL} = 0.1$. Three different explicit time-integration methods are investigated:

- first-order explicit Euler method,
- three-step Runge–Kutta method,
- Krylov-based exponential time-integration.

The results are shown in Figure 3. Due to numerical instabilities as the CFL-number is increased, the Euler method as well as the Runge–Kutta method cannot be continued beyond a CFL-number of about 0.4 and 0.6, respectively.

DOI: 10.1002/fld

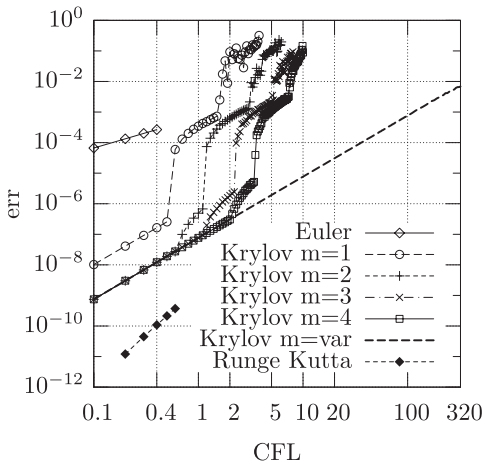

Figure 3. Local error versus the CFL-number for the Euler method, three-step Runge–Kutta method and the exponential time-stepping method using Krylov bases of varying dimensionality (fixed $m = 1, 2, 3, 4$ and adaptive $m = \text{var}$).

Since we do not have the exact solution for our nonlinear equations, a numerically obtained reference solution is used; for the remainder of this study a standard explicit Runge–Kutta three-step method with a small CFL-number will be used to compute the reference solution. In above-mentioned error-norm, the measured error of the respective method under investigation will initially always be higher than the one of the reference solution.

The Krylov solutions appear to be much more stable despite a monotonically increasing truncation error. Using at least a two-dimensional subspace, the error of the Krylov time-integration falls on a straight line up to a certain CFL-number. Increasing the CFL-number beyond this point increases the error in a characteristic 'step pattern' up to a CFL-number beyond which the solution can no longer be computed. If we allow an expanding Krylov subspace (increasing its dimensionality as needed) the error continues on a straight line up to very high CFL-numbers. In the present study we computed converged solution for CFL-numbers of up to 320.

A representative solution of the Burgers equation is presented in Figure 4. For the case of a high CFL-number (CFL = 320), a deviation from the reference solution is clearly visible; it should, however, be kept in mind that this solution has been obtained in only seven time-steps. The reference solution, on the other hand, required more than 20 000 time-steps.

An evaluation of the expended run-time versus the CFL-number is depicted in Figure 5(a). Whereas for small CFL-numbers a large number of time-steps is necessary to advance the initial condition from $t = 0$ to T, at large CFL-numbers the bulk of the computational effort is consumed by forming and projecting onto the growing Krylov subspace. An optimal CFL-number is established at CFL ≈ 15 which presents the most favorable balance between the permissible time-step and the dimensionality of the necessary Krylov space. Figure 5(b) displays the dimensionality of the Krylov subspace versus the chosen CFL-number; we observe a nearly direct proportionality between the two quantities.

Figure 6(a) shows the evolution of the residual over time. For a fixed dimension of the Krylov subspace (chosen as $m = 3$) a residual of about 10^{-6} can be maintained over the entire integration

DOI: 10.1002/fld

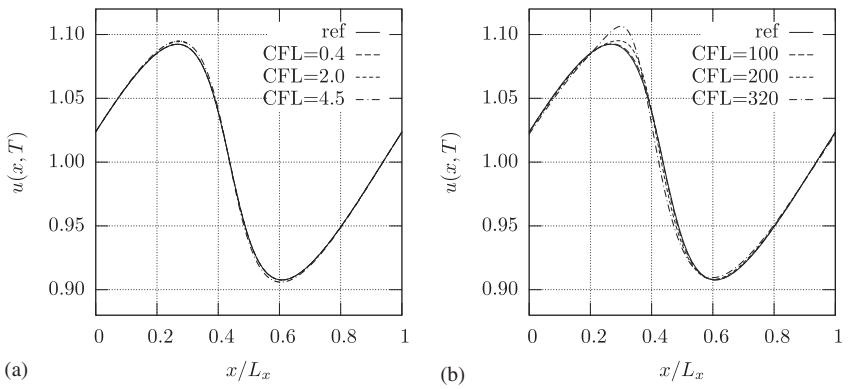

Figure 4. (a) Solution of the Burgers equation for a fixed number of Krylov subspaces ($m=3$) and different CFL-numbers. The choice of the CFL-number is based on the error (cf. 'error-steps' in Figure 3). (b) Solution of the Burgers equation for a variable number of Krylov subspaces (depending on the error); high CFL-numbers are considered.

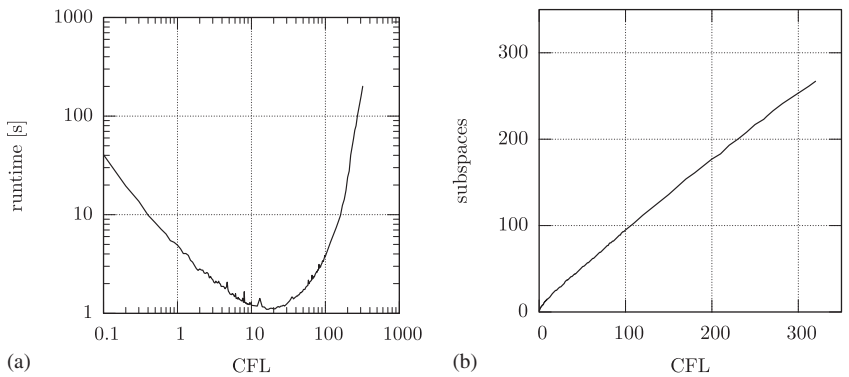

Figure 5. (a) Computational run-time for the integration of the Burgers equation over the interval $t \in [0, T]$ using an adaptive Krylov basis to maintain a specified error bound. (b) Maximal dimension of the Krylov subspaces for the integration of the Burgers equation over the interval $t \in [0, T]$ versus the CFL-number.

interval $[0\ T]$. Increasing the dimensionality of $\mathscr{K}$ to six yields a significantly lower residual ($<10^{-11}$). An adaptive strategy is also included in the plot, where a given threshold for the residual is prescribed (10^{-9}) and the dimensionality of the Krylov subspace is adjusted accordingly to maintain the residual levels below this threshold. The corresponding residual curve (the dashed line in Figure 6(a)) shows characteristic discontinuities when the Krylov space $\mathscr{K}$ has been augmented.

Approximate eigenvalues of the evolution operator are readily available from the Arnoldi process as the spectrum of the Hessenberg matrix **H** (also known as the Ritz values) represent an approximation of the spectrum of **A**. In the case of the Burgers equation the spectrum (see Figure 6(b)) shows a characteristic parabolic shape reminiscent of the typical spectrum of advection–diffusion operators. The real part of the eigenvalues lie in the left half-plane indicating a stable evolution

DOI: 10.1002/fld

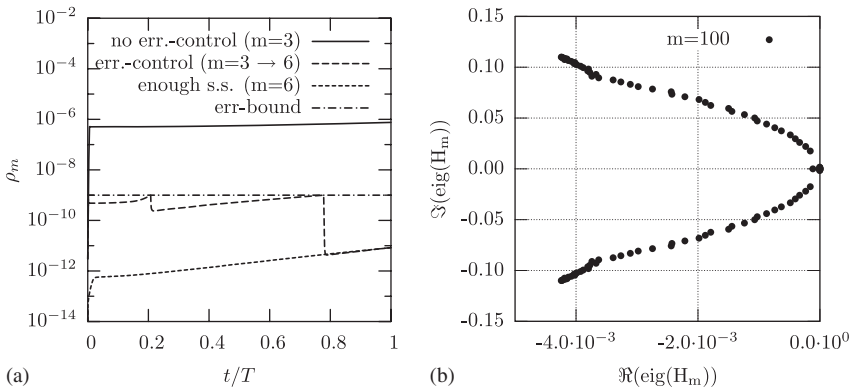

Figure 6. (a) Time evolution of the residual for fixed and adaptive Krylov bases. (b) Eigenvalues (Ritz values) of the Hessenberg matrix H_m for a Krylov subspace of 100 elements.

operator. This method will only find the largest eigenvalues of the system **A** and will lose accuracy if $\mathbf{V_m}$ is not properly orthogonalized. In the present project a modified Gram–Schmidt algorithm is implemented for the orthogonalization of $\mathbf{V_m}$ which is numerically stable, but especially for large dimensions of the Krylov subspace orthogonality up to machine precision can no longer be guaranteed. An improved orthogonality can be achieved with e.g. an iterative modified Gram-Schmidt algorithm at the expense of performance on parallel computers [14, 15].

3.1.1. One-dimensional linear convection–diffusion equation. Since the Krylov time-integration method gives exact results for linear governing equations we show some results for the linear, one-dimensional convection–diffusion equation, which is easily obtained from the Burgers equation by linearizing the convective term (which introduces a constant advection velocity λ). We obtain

$$\frac{\partial u}{\partial t}+\lambda\frac{\partial u}{\partial x}-\frac{1}{Re}\frac{\partial^2 u}{\partial x^2}=0 \tag{31}$$

With the same settings as in the nonlinear case we again compute the error versus the CFL number. As apparent from Figure 7 the error of the Krylov time-integration scheme is smaller than the error accumulated by the three-step Runge–Kutta method and is, by contrast, not growing as the time-step is increased. Furthermore, the exponential integration method seems to be extremely stable which allowed simulations with CFL-numbers of up to CFL = 1024. In this extreme case, the computation used only a single time iteration to integrate the governing equation from $t=0$ to T.

3.2. Two-dimensional Navier–Stokes equations—a counter-rotating vortex pair

The second, more complex case is based on the two-dimensional, compressible Navier–Stokes equations. The direct numerical simulation code solves for the pressure p, the velocity components $\mathbf{u}=(u,v)^{\mathrm{T}}$ and the entropy s using a decomposition into plane characteristic waves (see [16]). For the spatial discretization a central compact difference scheme (see [17]) with sixth-order accuracy is implemented and solved on an orthogonal grid with approximately 150 000 points.

DOI: 10.1002/fld

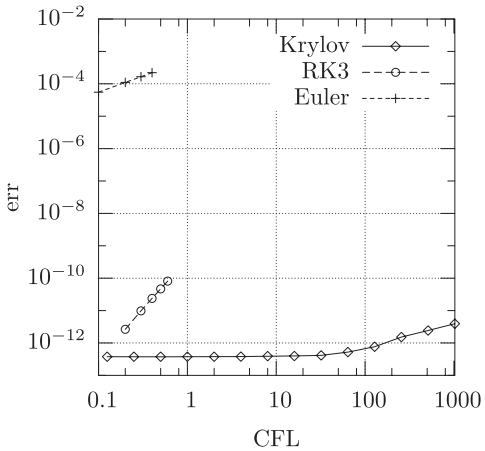

Figure 7. Local error versus the CFL-number for the linearized Burgers equation (advection–diffusion equation) using the Euler method, the three-step Runge–Kutta method and exponential (Krylov) time-stepping.

In the present case, we simulate the evolution of a counter-rotating vortex pair in a two-dimensional setup. As an initial condition we use a shear layer and superimpose on it a small disturbance in the center of the computational domain. In the shear direction we assume an infinite domain with periodic boundary conditions. The Mach number is, respectively, set to $Ma = 0.2$ and $Ma = -0.2$ for the flows separated by the shear layer. Integrating this initial condition in time causes a vortex roll-up followed by a vortex pairing. In Figure 10(a) this vortex pairing process is visualized by the entropy variable evaluated at the terminal time T. For this simulation we used the Krylov time-integration with a five-dimensional subspace and a CFL-number of 2.75. As a reference solution we again use the result obtained by a three-step Runge–Kutta integration.

In Figure 9(a) the computational work (measured in floating point operations) to integrate the Navier–Stokes equations up to a terminal time T for various CFL-numbers, is presented. As a main result we demonstrate that the exponential time-integration based on Krylov subspaces can be up to twice as fast as a standard explicit Runge–Kutta method. Due to the loss of time-stability the reference solution failed for CFL-numbers larger than one. On the contrary, CFL-numbers of up to 50 could be reached for the exponential time-integration using a maximally 106-dimensional Krylov subspace. Increasing the CFL-number even further caused an overflow of the available computer memory (1 GB). Owing to the error-dependent use of subspaces, we do not observe a linear decay of the computational time with the CFL-number, as is the case for standard explicit methods. As before, there appears to be an optimum concerning the computational work for CFL-numbers around 15 (see also Figure 5(a)).

The local error as a function of the chosen time-step (or CFL-number) is shown in Figure 8(a) for the exponential time integration method with variable-size Krylov subspaces and the reference Runge–Kutta method (up to its stability limit). Although the error is growing nearly linear with the CFL-number it is still reasonably small. The accuracy of the Krylov scheme for moderate to large CFL-numbers is confirmed by plotting the pressure solution in a horizontal slice through the centerline of the computational domain. The three Krylov solutions (for $\mathrm{CFL} = 10, 20$ and 50)

DOI: 10.1002/fld

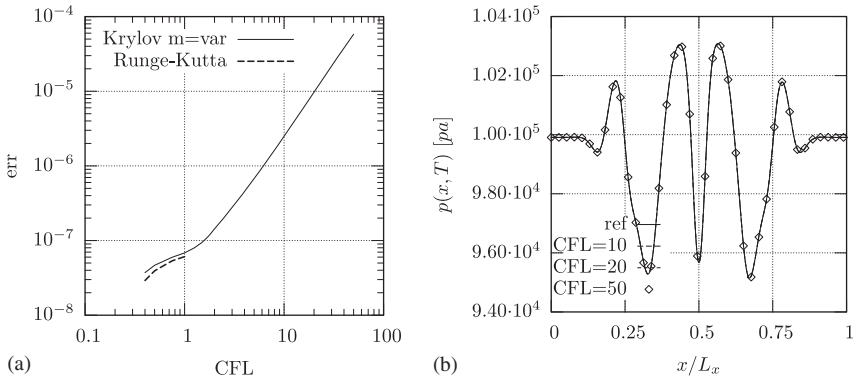

Figure 8. (a) Local error versus the CFL-number for the two-dimensional compressible Navier–Stokes equations. The three-step Runge–Kutta method is contrasted with an adaptive Krylov technique. (b) Horizontal slice through the computational domain (middle of the mixing layer). Shown is the pressure-field using the Krylov and Runge–Kutta time-integration technique with different CFL-numbers. Even for a Krylov solution with CFL = 50, no deviation from the Runge–Kutta reference solution (obtained with CFL = 0.25) is visible.

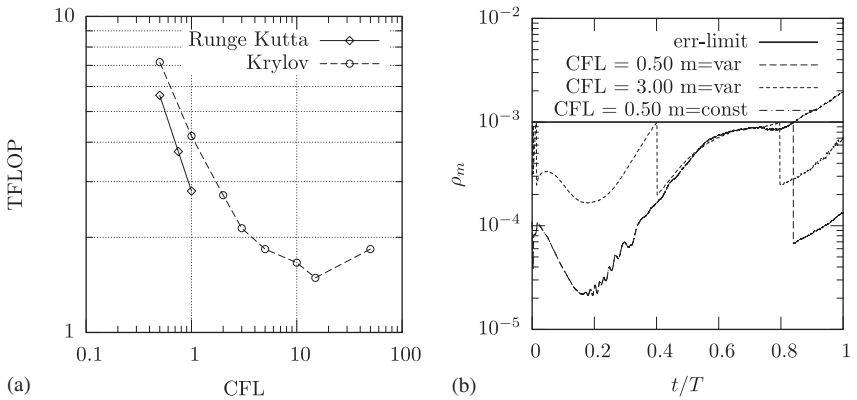

Figure 9. (a) Computational work (in Tera Floating Point Operations) versus the CFL-number for the two-dimensional compressible Navier–Stokes equations using a Krylov exponential time-stepping technique with at most a 27-dimensional Krylov subspace. The computational work for a three-step Runge–Kutta method is included for reference. (b) Time evolution of the residual for adaptive and fixed Krylov techniques and different CFL-numbers.

together with the reference Runge–Kutta solution are shown in Figure 8(b). The four curves coincide to within plotting accuracy; even for the highest CFL-number case (CFL = 50) no deviation from the reference solution is visible.

Similar to the previous case, an evaluation of the exerted computational effort versus the CFL-number reveals a sharp drop (see 9(a)) until more effort has to be expended due to the formation and projection onto the increasing Krylov basis. A minimum is reached—as before—for a CFL-number of approximately 15. The evolution of the residual is displayed in Figure 9(b). For a variable-size

DOI: 10.1002/fld

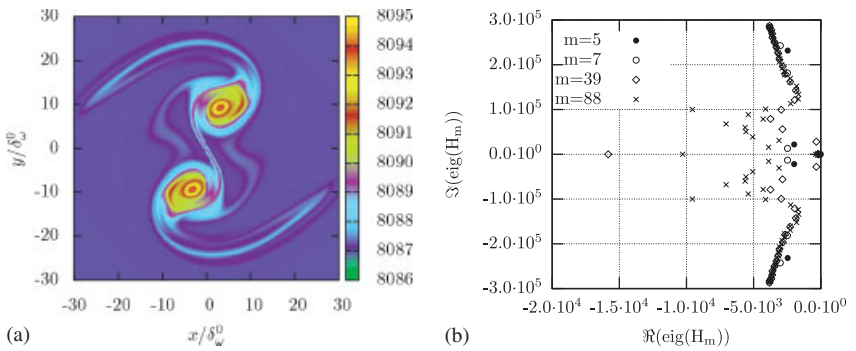

Figure 10. (a) Counter-rotating vortex pair visualized by the entropy variable (δ_ω^0 defines the initial shear-layer thickness). The solution has been computed with a CFL-number of CFL = 10 using a 25-dimensional Krylov subspace. (b) Eigenvalues of the Hessenberg matrix H_m (Ritz values) for different Krylov subspaces.

Krylov basis and a prescribed residual threshold of 10^{-3} we observe familiar discontinuities in the residual curve as the dimensionality of the Krylov subspace is adjusted in order to maintain the residual below the threshold. For a fixed Krylov basis, the residual eventually exceeds the 10^{-3}-level (in our case, near $t/T \approx 0.8$).

The spectrum of the fluid motion—more precisely, the eigenvalues of the Hessenberg matrix **H**—is shown in Figure 10(b). We observe a symmetric spectrum that is dominated by convective (acoustic) processes signified by the branches that extend in the positive and negative imaginary direction. The fluid motion on a slower time-scale, including the vortex pairing process, is contained in the eigenvalues closer to the origin. This spectrum illustrates the vast range of characteristic speeds present in the flow that poses a challenge to any explicit time-stepping scheme. Furthermore, it can be concluded from Figure 10(b) that the flow itself is stable, since the real parts of all eigenvalues lie in the stable half-plane.

3.3. Three-dimensional Navier–Stokes equations—a subsonic planar jet

The last and most challenging example is again based on the compressible Navier–Stokes equations but now includes the third velocity component w to obtain a fully three-dimensional flow. As a physical problem we investigate a three-dimensional planar and subsonic jet which is perfectly expanded. The Navier–Stokes equations are solved on an orthogonal grid stretched in both the streamwise and the normal direction. In the spanwise direction, periodicity and statistical homogeneity are assumed. The spatial derivatives are approximated using a sixth-order compact difference scheme, and the dependent variables are formulated in terms of characteristic wave components. The computational mesh contains more than 20 million grid points with 768 points in the streamwise direction, 512 points in the normal direction and 64 points in the (periodic) spanwise direction. Based on the height of the jet, a Reynolds number of $Re = 11\,300$ has been chosen.

The large number of grid points for this three-dimensional case makes the use of parallel computing techniques imperative. To this end, a domain decomposition approach combined with a message passing interface (MPI) has been implemented. The current setup uses 256 CPUs. It should also be mentioned that the Arnoldi decomposition has to be parallelized as well, since

DOI: 10.1002/fld

the Hessenberg matrix has to be computed for the entire computational domain. Parallelizing the Arnoldi-algorithm, however, is easily accomplished since only scalar data—the norm β and the scalar product $h_{i,j}$; see Algorithm 1—have to be exchanged between the processors. The associated additional amount of computational time, which is not present for standard explicit methods, is however, negligible compared to the evaluation of the right-hand side.

Several numerical boundary conditions have been implemented. To avoid large structures to interact with the outflow boundary, a sponge term is applied to the governing equations as a highly dissipative force. An additional low-pass filter (applied spatially) and a grid stretching in the streamwise direction aid in removing high-frequency oscillations. Finally, a boundary wave-acceleration term is implemented to avoid acoustic reflections from the downstream and normal boundaries. At the inflow the rectangular nozzle is modeled by a laminar tanh-profile. The resulting shear layers are forced slightly downstream of the inflow boundary with a model spectrum including the most unstable frequency of the shear layer together with its first subharmonic with a stochastic phase-shifting.

Figure 11 shows a result of the three-dimensional computation. It displays the absolute velocity ($|\mathbf{u}|$) in a streamwise-normal cutting plane as well as isosurfaces of the streamwise vorticity ($((\nabla \times \mathbf{u})^{\mathrm{T}}\mathbf{e_3}$). This computation has been performed with a CFL-number of CFL$=5$ and used a 25-dimensional Krylov subspace. A simulation with a CFL-number of CFL$=20$ (about 100 Krylov bases) used approximately 20% of the available computer memory. Based on this information and the assumption that the memory requirements scale linearly with the CFL-number, it is possible to estimate the maximum possible CFL-number. For our configuration this would amount to CFL-numbers of up to CFL≈ 120 (cf. Figure 12).

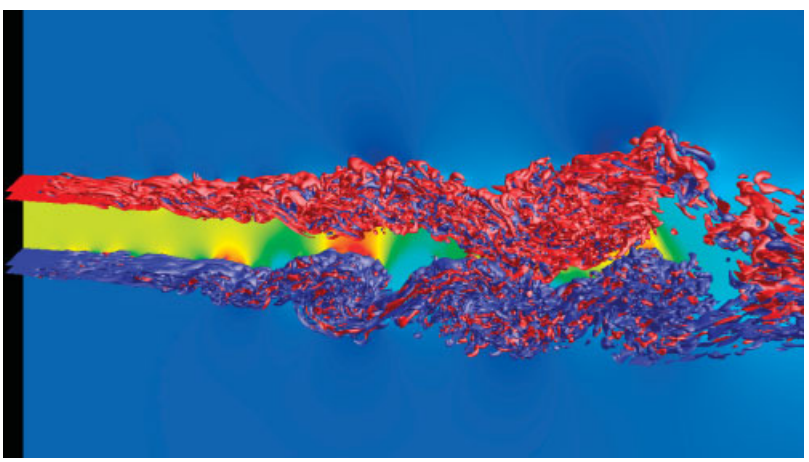

Figure 11. Instantaneous flow field (visualized by the isosurface of the streamwise vorticity (blue -1000, red 1000) and the magnitude of the absolute velocity) of a three-dimensional planar subsonic jet obtaining by the exponential (Krylov) time-integration technique. The simulation has been performed with a CFL-number of five and used at most a 25-dimensional Krylov subspace. The computational mesh consists of approximately 20×10^6 grid points; the simulation has been run on 256 processes using MPI.

DOI: 10.1002/fld

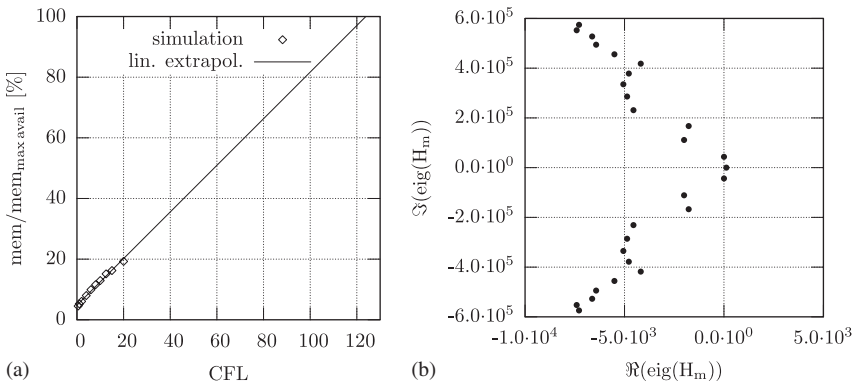

Figure 12. (a) Memory usage (in percent of the maximally available memory) for three-dimensional simulations of the compressible Navier–Stokes equations with different CFL-numbers (subspaces), diamonds ◇. The solid line is a linear extrapolation (least-squares fit of the simulation data) to provide an estimate for the maximally possible CFL-number. (b) Eigenvalues of the Hessenberg matrix H_m (Ritz values).

We also display again the Ritz values, i.e. the eigenvalues of the Hessenberg matrix **H**, at the terminal time T in Figure 12(b). These eigenvalues represent an approximation of the evolution operator **A** propagating the subsonic jet with more than 100×10^6 degrees of freedom. Similar to the case of the two-dimensional vortex pairing, we again identify a wide range of time-scales and a dominant convective behavior. In contrast to the previous cases, we observe two complex conjugate eigenvalues with a small but positive real part.

The efficiency of exponential time integration over commonly used Runge–Kutta schemes is substantiated in Figure 13 where the ratio of wall-clock times between explicit Runge–Kutta and Krylov-based integration is presented. We observe a speed-up by a factor of up to two for moderate to large CFL-numbers. This speed-up is quite significant in high-performance supercomputing applications where explicit time-stepping schemes are pushing the envelope of current resources and algorithms. The local error using exponential time integration is displayed in Figure 13(b) which demonstrates that error-norms comparable with standard explicit schemes can be expected.

4. SUMMARY AND CONCLUSIONS

In the present paper the method of exponentially integrating ordinary differential equations (ODEs) using Krylov subspaces has been presented and applied to different showcases with a focus on large-scale fluid-dynamic simulations containing a wide range of temporal scales. Our numerical experiments showed that the exponential Krylov time integration is up to twice as fast as a standard Runge–Kutta three-step method when the wall-clock time is considered as a measure of performance. This speed-up factor is considerable and important for large-scale high-performance computations where the CPU hours are limited and expensive. As an example, the three-dimensional simulation in Section 3.3, performed on 256 processors, required a total of 1000 CPU hours to integrate the initial condition up to the terminal time T using a standard explicit Runge–Kutta scheme. The same case could be completed in half the time using the presented exponential time integration scheme based on Krylov subspaces. Furthermore, it could be shown that despite a

DOI: 10.1002/fld

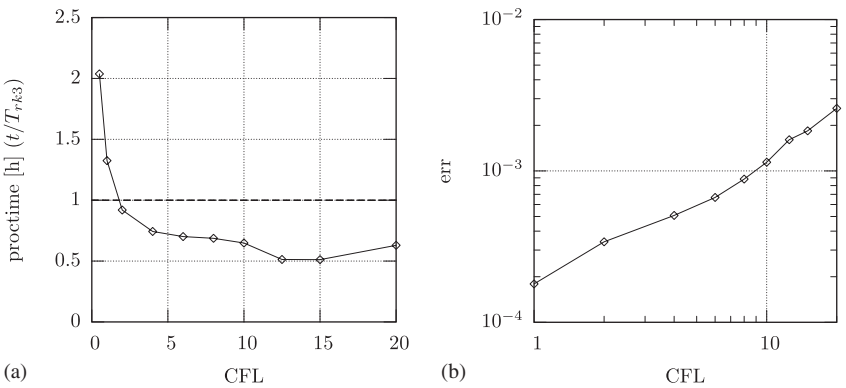

Figure 13. (a) Performance of the Krylov time-integration; for CFL-numbers larger than 2, the Krylov method is faster than a standard explicit Runge–Kutta three-step integration and can be up to twice as fast for CFL-numbers of about 10. (b) Local error as a function of the CFL-number which demonstrates that the exponential time-stepping technique exhibits errors typical of standard explicit schemes while allowing significantly larger time-steps.

nearly linear increase of the time-stepping error with the CFL-number, the error remained well within the levels of more common, explicit schemes. Exponential time-integration methods thus allow a more detailed and efficient exploration of fluid flow phenomena that are dominated by a wide range of temporal scales, such as low-Mach number flows or combustion applications, which we will leave for a future effort.

ACKNOWLEDGEMENTS

The second author wishes to acknowledge the generous support of the Alexander-von-Humboldt Foundation.

REFERENCES

1. Hochbruck M, Lubich Ch, Selhofer H. Exponential integrators for large systems of differential equations. *SIAM Journal on Scientific Computing* 1998; **19**(5):1552–1574.
2. Gallopoulos E, Saad Y. Efficient solutions of parabolic equations by Krylov approximation methods. *SIAM Journal on Scientific and Statistical Computing* 1992; **13**(5):1236–1264.
3. Saad Y. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis* 1992; **29**(1):209–228.
4. Sidje R. EXPOKIT: a software package computing matrix exponentials. *ACM Transactions on Mathematical Software* 1998; **24**(1):130–156.
5. Saad Y. Application of Krylov exponential propagation to fluid dynamic equations. *RIACS Technical Report 91.06*, 1991.
6. Newman Ch. Exponential integrators for the incompressible Navier–Stokes equations. *Ph.D. Thesis*, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2003.
7. Trefethen LN. *Spectra and Pseudospectra*: *The Behavior of Nonnormal Matrices and Operators*. Princeton University Press: Princeton, NJ, 2005.
8. Moler C, Van Loan C. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review* 2003; **45**(1):3–49.

DOI: 10.1002/fld

9. Hochbruck M, Ostermann A. Explicit exponential Runge–Kutta methods for semilinear parabolic problems. *SIAM Journal on Numerical Analysis* 2005; **43**(3):1069–1090.
10. Hochbruck M, Lubich Ch. On Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis* 1997; **34**:1911–1925.
11. Tokman M. Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods. *Journal of Computational Physics* 2006; **213**:748–776.
12. Caliari M, Ostermann A. Implementation of exponential Rosenbrock-type integrators. *Applied Numerical Mathematics* 2008; DOI: 10.1016/j.apnum.2008.03.021.
13. Edwards W, Tuckerman L, Friesner R, Sorensen D. Krylov methods for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 1994; **110**:82–102.
14. Faryseé V, Giraud L, Kharraz-Aroussi H. *On the Influence of the Orthogonalization Scheme on the Parallel Performance of GMRES*. Lecture Notes in Computer Science, vol. 1470. Springer: Berlin, 1998; 751–762.
15. Lehoucq R, Salinger A. Large-scale eigenvalue calculations for stability analysis of steady flows on massively parallel computers. *International Journal for Numerical Methods in Fluids* 1999; **36**:309–327.
16. Sesterhenn J. A characteristic-type formulation of the Navier–Stokes equations for high-order upwind schemes. *Computers and Fluids* 2001; **30**:37–67.
17. Lele S. Compact finite difference schemes with spectral like resolution. *Journal of Computational Physics* 1992; **103**:16–42.