Computers and Fluids 159 (2017) 53–63

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

A numerical investigation of matrix-free implicit time-stepping methods for large CFD simulations

Arash Sarshar[a,*], Paul Tranquilli[a], Brent Pickering[b], Andrew McCall[b], Christopher J. Roy[b], Adrian Sandu[a]

[a] Department of Computer Science and Applications, VirginiaTech, United States
[b] Department of Aerospace and Ocean Engineering, VirginiaTech, United States

ARTICLE INFO

Article history:
Received 19 August 2016
Revised 14 September 2017
Accepted 18 September 2017
Available online 20 September 2017

JEL classification:
65L05
65L07

Keywords:
Initial value problems
Computational fluid dynamics
Implicit time integration

ABSTRACT

This paper is concerned with development and testing of advanced time-stepping methods for large unsteady CFD problems in the method of lines approach, where the semi-discretization in space is performed first. The performance of several time discretization methods is studied numerically with regards to computational efficiency, order of accuracy, and stability, as well as the ability to effectively treat stiff problems. We consider matrix-free implementations, a popular approach for time-stepping methods applied to large CFD applications due to its adherence to scalable matrix-vector operations and a small memory footprint. We compare explicit methods with matrix-free implementations of implicit, linearly-implicit, as well as Rosenbrock–Krylov methods. We show that Rosenbrock–Krylov methods are competitive with existing techniques excelling for a number of problem types and settings.

1. Introduction

While flow problems are inherently unsteady, computer flow simulations have traditionally focused mainly on steady-state flow problems because they reduce the computational effort dramatically. Nevertheless, in many practical applications it is important to quantify the impact of unsteady flow phenomena on the forces and moments exerted on a body. These phenomena impact performance characteristics such as the lift and drag of a body, or the dynamic response of a control system. Historically, in aircraft design, these unsteady effects have required additional analyses to mitigate undesirable aeroelastic effects such as wing flutter and undesirable stall characteristics, among other issues [1–3]. Sometimes the unsteady effects are beneficial, e.g., when using leading edge extensions (LEX) to improve high angle of attack performance [4]. Studies of low Reynolds number unsteady flows have become much more relevant today with the development of micro air vehicles (MAV) [5–11], with typical sizes as small as 15 cm. At these low Reynolds numbers viscous forces dominate the flow characteristics, leading to unsteady viscous effects such as laminar separation and von Kármán vortices, as commonly demonstrated in the flow over a cylinder [12].

The use of CFD allows for preliminary analyses of these designs to determine whether any undesirable unsteady effects will be present, before committing to the expensive development and testing of a physical system. However, a major limitation of unsteady flow analysis using CFD is the prohibitive amount of computational time required to simulate a time-accurate solution with the time integration schemes commonly used to solve the Navier–Stokes equations. This coupled set of nonlinear partial differential equations has to be solved iteratively to determine the solution for each time step. A fine mesh resolution typically required to capture the length and time scales of the flow. For making the computations feasible it is necessary to use a time discretization scheme that maximizes convergence speed without prohibitively restricting the time steps due to stability constraints. It is also important that the time integration scheme provides an accurate solution at every time step.

Explicit time integration methods have been used for time-accurate solutions of unsteady flow problems due to their low computational cost per step and moderate memory requirements. For example, in [13] a number of embedded high-order explicit Runge–Kutta methods with minimal memory storage have been developed for the compressible Navier–Stokes equations based on van der Houwen's technique [14] for stage memory storage

* Corresponding author.
E-mail addresses: sarshar@vt.edu (A. Sarshar), ptranq@vt.edu (P. Tranquilli), bpickeri@vt.edu (B. Pickering), mandrew9@vt.edu (A. McCall), cjroy@vt.edu (C.J. Roy), sandu@cs.vt.edu (A. Sandu).

http://dx.doi.org/10.1016/j.compfluid.2017.09.014
0045-7930/

reduction. However, stability constraints restrict the maximum time steps that explicit methods can employ. Sengupta et al. [15,16] show that explicit Runge–Kutta methods restricted to small time-steps can better predict physical instabilities of a flow problem than multistep Adams–Bashforth methods. Bhaumik et al. [17] investigate numerical dispersion and phase-shift errors associated with Runge–Kutta methods and different spatial discretizations. These issues are addressed in [18–21] by developing a class of explicit Runge–Kutta methods that, when coupled with accurate spatial discretization schemes, optimize wave properties in advection-dominated problems.

Due to their better stability, implicit time-stepping methods can use very large time steps, however, the computational cost per step is also greater. The overall computational efficiency of a method is determined by the trade off between the computational cost per step and the total number of steps required to carry out the simulation.

The main cost of implicit methods is associated with solving a large system of nonlinear equations at each step. Newton type methods for the solution of nonlinear systems are commonly used in the CFD literature in conjunction with preconditioned Krylov-based solvers for the inherent linear systems. The popular Jacobian-free Newton Krylov (JFNK) methods employ finite difference approximations of the Jacobian-vector products required by Krylov solvers [22]. Studies of JFNK methods applied to Navier–Stokes equations [23] have shown that error tolerances of Krylov space solvers need to be carefully optimized for performance and accuracy. Interested readers may also refer to [24] for detailed experiments quantifying the effects of Krylov solver tolerance on the convergence and efficiency of high order Rosenbrock methods in CFD applications. Furthermore, development of matrix-free space-time implicit methods for DG discretizations are reported in [25–27].

There is considerable interest in developing numerical schemes that provide a suitable level of implicitness for time integration of stiff flow problems, such as to allow relatively large time steps while keeping the cost per time step comparable to that of explicit methods.

This paper studies the efficiency of matrix-free time stepping schemes, applied to ODE systems arising from the method-of-lines treatment of spatial dependencies in Navier–Stokes equations. In such cases, DNS approach to flow problems often leads to ODE systems of very large dimensionality. We examine several practical aspects of using Jacobian-free methods and their effects on the computational cost and the accuracy of the solutions. In addition to standard techniques, we examine a new class of lightly-implicit time integration schemes, called Rosenbrock–Krylov methods, which are particularly well suited to employ approximate Jacobian-vector products.

The remaining part of this paper is structured as follows. Section 2 reviews numerical methods accessible for the time integration of large systems of ordinary differential equations. The numerical methods investigated here and their implementation are presented in Section 3. Section 4 applies these methods to a number of test problems and studies their effectiveness in terms of their numerical accuracy, stability, and computational efficiency in case of high dimensional problems. Conclusions and future work directions are discussed in Section 5.

2. Numerical time integration for CFD applications

Consider the autonomous initial value problem:

$$\frac{dy}{dt} = f(y), \quad y(t_0) = y_0, \quad t_0 \le t \le t_F, \quad y(t) \in \mathbb{R}^N, \quad f : \mathbb{R}^N \to \mathbb{R}^N. \tag{1}$$

In this paper Eq. (1) represents the system of ODEs resulting from the spatial semi-discretization of the Navier–Stokes equations for flow problems in the method-of-lines framework. The system is considered autonomous without loss of generality: any system can be written in autonomous form by appending the time variable to the solution vector. With only time derivatives remaining in Eq. (1), it is the choice of time-stepping method that determines the stability, accuracy, and efficiency of the numerical solution as the solution is propagated in time.

We next review several important classes of numerical time integration algorithms.

2.1. Runge–Kutta methods

The historically well-known time integration schemes attributed to Runge and Kutta are well-studied [28,29] and extensively utilized in flow applications [30,31]. Let $y_n \approx y(t_n)$ be a numerical approximation of the solution of the system (1). An s-stage Runge–Kutta method(advances the numerical solution to the next time step $t_{n+1} = t_n + h$ as follows:

$$k_i = f\left(y_n + h\sum_{j=1}^{s} a_{i,j}\, k_j\right), \quad i = 1, \ldots, s; \tag{2a}$$

$$y_{n+1} = y_n + h\sum_{j=1}^{s} b_j\, k_j. \tag{2b}$$

The method coefficients

$$a = [a_{i,j}]_{1\le i,j\le s} \quad b = [b_i]_{1\le i\le s} \quad c = [c_i]_{1\le i\le s},$$

are determined such that the method (2a)and (2b) has the desired accuracy and stability properties [32, II.1].

Explicit Runge–Kutta (ERK) methods are characterized by coefficients $a_{i,j} = 0$ for any $j \le i$. This means that each stage value k_i (2a) depends only on previously stage vectors $k_1, \ldots, k_{i-1}$. This leads to the convenient result that explicit Runge–Kutta methods need only one ODE right-hand-side function evaluation per stage, and no linear or nonlinear systems of equations are solved in the process. The stability requirements due to CFL conditions limit the step size h, and therefore impact the efficiency of the method.

Singly diagonally implicit Runge–Kutta methods (SDIRK) [33, IV.6] are characterized by coefficients $a_{i,j} = 0$ for any $j < i$, and $a_{i,i} = \gamma > 0$ for all stages $i = 1, \ldots, s$. Solving for the stage vector k_i requires the solution of a nonlinear system of equations at each stage

$$F_i(k_i) = k_i - f(\xi_i + h\gamma k_i) = 0 \quad \text{for} \quad i = 1, \ldots, s, \tag{3}$$

which makes the computational cost per step significantly larger than for ERK. However, this also leads to improved stability properties and the ability to use much larger time steps. The nonlinear Eq. (3) is solved using Newton-type iterations:

$$\Delta k_i^{\{\ell\}} = -\left(\frac{\partial F_i}{\partial k_i}\right)^{-1} F_i\left(k_i^{\{\ell\}}\right), \quad k_i^{\{\ell+1\}} = k_i^{\{\ell\}} + \Delta k_i^{\{\ell\}}, \quad \ell = 0, 1, \ldots \tag{4}$$

where

$$\frac{\partial F_i}{\partial k_i} = \mathbf{I}_N - h\,\gamma\,\mathbf{J}_n, \tag{5}$$

and $\mathbf{J}_n$ is the Jacobian of the ODE right-hand-side function:

$$\mathbf{J}_n = \left.\frac{\partial f(y)}{\partial y}\right|_{y=y_n}. \tag{6}$$

The fact that $a_{i,i} = \gamma$ for all stages allows re-using the LU decomposition of (5) in the solution of linear systems appearing in Eq. (4) for all stage vectors $i = 1, \ldots, s$.