Computers and Mathematics with Applications 69 (2015) 170–179

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

On the error propagation of semi-Lagrange and Fourier methods for advection problems☆

Lukas Einkemmer*, Alexander Ostermann

Department of Mathematics, University of Innsbruck, Austria

ARTICLE INFO

Article history:
Received 8 June 2014
Received in revised form 11 November 2014
Accepted 5 December 2014
Available online 26 December 2014

Keywords:
Semi-Lagrange methods
FFT
Error propagation
High-precision computations

ABSTRACT

In this paper we study the error propagation of numerical schemes for the advection equation in the case where high precision is desired. The numerical methods considered are based on the fast Fourier transform, polynomial interpolation (semi-Lagrangian methods using a Lagrange or spline interpolation), and a discontinuous Galerkin semi-Lagrangian approach (which is conservative and has to store more than a single value per cell).

We demonstrate, by carrying out numerical experiments, that the worst case error estimates given in the literature provide a good explanation for the error propagation of the interpolation-based semi-Lagrangian methods. For the discontinuous Galerkin semi-Lagrangian method, however, we find that the characteristic property of semi-Lagrangian error estimates (namely the fact that the error increases proportionally to the number of time steps) is not observed. We provide an explanation for this behavior and conduct numerical simulations that corroborate the different qualitative features of the error in the two respective types of semi-Lagrangian methods.

The method based on the fast Fourier transform is exact but, due to round-off errors, susceptible to a linear increase of the error in the number of time steps. We show how to modify the Cooley–Tukey algorithm in order to obtain an error growth that is proportional to the square root of the number of time steps.

Finally, we show, for a simple model, that our conclusions hold true if the advection solver is used as part of a splitting scheme.

1. Introduction

The accurate numerical simulation of advection dominated problems is an important problem in many scientific applications. However, due to the non-dissipative nature of the equations considered, care has to be taken to obtain a stable numerical scheme (see, for example, [1]). A large body of research has been accumulated that describes finite difference, finite volume, and finite element discretizations of such problems. However, for advection-dominated problems so-called semi-Lagrangian methods offer a competitive alternative. These methods integrate the characteristics back in time and consequently have to use some interpolation scheme to reconstruct the desired value at the grid points. Strictly speaking, semi-Lagrangian methods can only be applied to systems of first-order differential equations. However, in many instances, first-order systems arise from the splitting of more complicated equations or constitute the linear part of an evolution equation (which is then treated separately from the nonlinearity). Consequently, semi-Lagrangian methods have been used

☆ This work is supported by the Austrian Science Fund (FWF) – project id: P25346.

* Corresponding author.
E-mail addresses: lukas.einkemmer@uibk.ac.at (L. Einkemmer), alexander.ostermann@uibk.ac.at (A. Ostermann).

http://dx.doi.org/10.1016/j.camwa.2014.12.004

extensively in applications ranging from fluid dynamics to plasma physics (see e.g. [2,3]). Such an approach is especially promising, if the characteristics (of a sub-problem) can be computed analytically; this can be done, for example, in context of the Vlasov–Poisson equations. In addition, semi-Lagrangian methods do not impose a Courant–Friedrichs–Lewy (CFL) condition.

In some problems, methods based on the FFT (fast Fourier transform) can also be employed; this is the case for tensor product domains (see e.g. [4]). Compared to FFT based methods, the semi-Lagrangian methods provide a local approximation (which is important in the context of parallelization). They are more easily applicable to non-periodic boundary conditions (due to the absence of Gibbs' phenomenon) and usually are better suited to handle nonlinearities. Using the fast Fourier transform, on the other hand, allows us to solve the linear advection equation exactly (in infinite precision arithmetics).

In most scientific applications a tolerance of say 10^{-3} is sufficient. In this case, the main research goal is to construct more efficient algorithms and to implement better step size control mechanism. Also methods that preserve certain invariants of the continuous system are of interest in that context.

However, a number of applications have been identified where double precision floating point numbers are not sufficient. A proposed remedy is to (selectively) use 128-bit floating point numbers. Note, however, that this procedure is accompanied by a significant reduction in performance. Examples of such problems range from the investigation of vortex sheet roll-ups in fluid dynamics to electromagnetic scattering phenomena (for an excellent review article see [5]).

Also, concern has been raised in recent years with regard to the reproducibility of numerical simulations; especially if such simulations are conducted on different computer systems. In [6], for example, it is demonstrated that climate codes show significantly different results depending on the number of processors that are employed in the simulation. One popular choice of numerical methods for atmospheric modeling are semi-Lagrangian methods.

Furthermore, due to the diminishing gain in per core CPU (central processing unit) performance, massively parallel computing architectures, such as GPUs and the Xeon Phi, have become more and more common. Usually in such situations the memory per core is significantly smaller than in more traditional cluster systems. In addition, single precision floating point performance is usually faster than double precision floating point performance (for example, the CUDA FFT single precision implementation achieves a speedup of about 2.5 compared to the double precision implementation [7]). Such considerations make the use of single precision floating point numbers attractive for some applications.

Therefore, our goal in this paper is to study the error propagation in a context where results close to machine precision are of interest or where the error has to be tightly controlled. We will limit ourselves to the advection equation (on a finite spatial interval)

$$\partial_t u(t,x) + v\partial_x u(t,x) = 0, \tag{1}$$

where u is a continuously differentiable function and v is a given constant.

For this model problem, we consider the time evolution of the interplay of round-off and discretization errors for semi-Lagrangian and FFT based methods. The discretization of (1) is important in itself as it is a building block for many more involved schemes (for example, in the context of splitting methods or for methods where the linear part is treated differently). Such schemes are applied, for example, in fluid dynamics or to solve the Vlasov equation. Note, however, that problems where v is a function of space and time can be treated as well within the semi-Lagrangian approach. In this case the characteristics have to be integrated backward in time by a suitable ordinary differential equation solver. In certain situations a function v which depends on the unknown function u can be handled as well (see, for example, [8]).

For future reference, we note that the analytic solution of (1) can be easily written down as

$$u(t,x) = u(0, x - vt),$$

where in this paper we always assume that v is a constant independent of x and t.

2. Description and error bounds

In this section we will describe how to use the semi-Lagrangian method (Section 2.1) and the fast Fourier transform (Section 2.2). Furthermore, we will discuss some theoretical results concerning the discretization error of these schemes.

2.1. The semi-Lagrangian method

A time step in the semi-Lagrangian method for the ith grid point is computed as follows:

$$u_n(x_i) = u_{n-1}\left(X_{x_i}(\tau)\right),$$

where u_n is the numerical solution after n time steps, τ is the time step size, and the characteristics of Eq. (1) are given by $X_x(\tau) = x - v\tau$. Note that $X_{x_i}(\tau)$ does not necessarily coincide with any grid point. Therefore, a sufficiently accurate interpolation procedure has to be used in order to extend the values stored at the grid to the entire domain. Both (continuous) spline interpolation as well as (discontinuous) Legendre or Lagrange interpolation are popular choices. Furthermore, similar to discontinuous Galerkin methods, multiple coefficients can be stored for each cell (which yields a local reconstruction at the expense of additional memory demands).