Applied Numerical Mathematics ••• (••••) •••–•••

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

An adaptive step size controller for iterative implicit methods

Lukas Einkemmer

Department of Mathematics, University of Innsbruck, Austria

ARTICLE INFO

Article history:
Received 17 October 2017
Received in revised form 7 June 2018
Accepted 7 June 2018
Available online xxxx

Keywords:
Adaptive step size selection
Implicit time integration
Iterative methods

ABSTRACT

The automatic selection of an appropriate time step size has been considered extensively in the literature. However, most of the strategies developed operate under the assumption that the computational cost (per time step) is independent of the step size. This assumption is reasonable for non-stiff ordinary differential equations and for partial differential equations where the linear systems of equations resulting from an implicit integrator are solved by direct methods. It is, however, usually not satisfied if iterative (for example, Krylov) methods are used.

In this paper, we propose a step size selection strategy that adaptively reduces the computational cost (per unit time step) as the simulation progresses, constraint by the tolerance specified. We show that the proposed approach yields significant improvements in performance for a range of problems (diffusion–advection equation, Burgers' equation with a reaction term, porous media equation, viscous Burgers' equation, Allen–Cahn equation, and the two-dimensional Brusselator system). While traditional step size controllers have emphasized a smooth sequence of time step sizes, we emphasize the exploration of different step sizes which necessitates relatively rapid changes in the step size.

1. Introduction

Numerically solving time dependent differential equations is an important task in many fields of science and engineering. Crucial to that process is an efficient time integrator. Since the early advent of computers such methods have been used to first solve ODEs (ordinary differential equation) and then PDEs (partial differential equations).

Numerical simulations can be run with a constant time step size. However, most modern software packages automatically select an appropriate step size given a desired tolerance (which is specified by the user). To accomplish this so-called (automatic) step size controllers are used in conjunction with an error estimator. Such an approach provides a range of advantages. First, it frees the user from selecting an appropriate step size and, ideally, from verifying the accuracy of the simulation (by numerical convergence studies or similar means). Second, a good step size controller is not only able to provide an estimate of the error made, but also to detect the onset of numerical instabilities and to reduce the step size in order to prevent them. This is of particular importance for explicit methods which, for example, can not operate with step sizes above the Courant–Friedrichs–Lewy (CFL) limit and for implicit methods that are not A-stable. Last, but certainly not least, step size controllers can increase the computational efficiency by allowing the software to adaptively increase and decrease the step size during the course of the simulation. This is usually done in response to an error estimate, where errors significantly below the specified tolerance indicate the possibility to increase the time step.

E-mail address: lukas.einkemmer@uibk.ac.at.

https://doi.org/10.1016/j.apnum.2018.06.002
0168-9274/

Please cite this article in press as: L. Einkemmer, An adaptive step size controller for iterative implicit methods, Appl. Numer. Math. (2018), https://doi.org/10.1016/j.apnum.2018.06.002

Step size controllers require an error estimate. Fortunately, estimating the error can often be accomplished with only a minor increase in the computational cost. One approach commonly used are so-called embedded Runge–Kutta methods. These schemes consist of a pair of Runge–Kutta methods with different order that share most or even all internal stages (see, for example, [12, Chap. II.4] or [4,5,20]). For multistep methods a comparison with extrapolated values is often used (see, for example, [14]). Alternatively, Richardson extrapolation can be used to obtain an error estimate, but this approach is usually more demanding from a computational point of view.

Almost all step size controllers are predicated on the assumption that the largest possible step size should be selected. Thus, the step size is chosen such that the error committed exactly matches the tolerance specified by the user. This is a reasonable assumption for explicit Runge–Kutta methods, where the computational cost is independent of the step size. Now, let us assume that our error estimator provides an estimate ϵ^k for the kth step (note that in accordance with much of the PDE literature we use superscripts to denote the time indices). The local error of a numerical method with order p is modeled as $e^k = D(\tau^k)^{p+1}$, where τ denotes the time step size and D is a constant (which for the purpose of simplicity is assumed to be independent of k; in most situations this is a reasonable assumption as the error constant only varies slowly on $\mathcal{O}(\tau)$ timescales). Then, to determine the optimal step size we set

$$\text{tol} = e^{k+1} = D(\tau^{k+1})^{p+1},$$

where tol is the user specified tolerance. This is not a particular useful constraint to determine τ^{k+1} as D is unknown. Thus, we consider

$$\frac{e^{k+1}}{e^k} = \left(\frac{\tau^{k+1}}{\tau^k}\right)^{p+1}$$

which can be solved for τ^{k+1}

$$\tau^{k+1} = \tau^k \left(\frac{\text{tol}}{e^k}\right)^{1/(p+1)}. \tag{1}$$

Equation (1) allows us to estimate the optimal time step τ^{k+1} based on the previous error estimate e^k and the previous time step size τ^k. This then results in a sequence of times t^k at which a numerical approximation is obtained. In practice this is a dangerous approach as even very small errors in the error estimate can result in a time step sizes that exceeds the prescribed tolerance (leading to step size rejection). Thus, usually a safety factor is incorporated. For more details we refer the reader to [12, Chap. II.4] and [19].

This simple formula can be interpreted as a P controller. The mathematical analysis is in fact based on this observation (see, for example, [9,8,21,22]). Consequently, PI controllers have been introduced [9], which for some integrators and problems show an increase in performance. Certainly, these PI controllers increase the smoothness of the step size sequence (i.e. the change in step size behaves less erratic). These ideas have been enhanced in a variety of directions. The importance of changing strategies when operating close to the stability limit for explicit methods has also been recognized [13].

Although some work has been conducted in estimating global errors (see, for example, [19]), the local step size controllers described above, with some modifications to avoid excessively large step size changes, still form the backbone of most time integration packages. For example, the RADAU5 code [11, Chap. IV.8] employs a variant of the PI controller, while the multistep based CVODE codes [14] and [6] use a variant of the P controller. As a result, the described step size controllers have been extensively tested and used in a range of applications, both for ordinary as well as for partial differential equations.

The desire for solving partial differential equations with ever increasing grid sizes and more accurate physical models, however, calls into question the validity of the assumptions made. In both the RADAU5 and CVODE code mentioned above implicit numerical methods are employed to solve the stiff ODEs resulting from the space discretization of the PDE under consideration. These implicit methods require the solution of a linear system of equations which is now routinely done by iterative numerical methods (such as the conjugate gradient method or GMRES). However, the number of iterations required is quite sensitive to the linear system solved. In particular, smaller time step sizes reduce the magnitude of the largest eigenvalue of the matrix, which in turn reduces the number of iterations required per time step. This means that reducing the time step size below what is dictated by the specified tolerance, i.e. according to equation (1), can actually result in an increase in performance.

Many implementations do not exploit this fact. However, the issue at hand has been recognized in [15] and [24]. Both of these approaches limit the size of the Krylov subspace. In [15] both lower and upper bounds are specified. If the Krylov dimension falls within those bounds, the time step size is chosen according to the step size controller. If this is not the case, the step size is adjusted. In [24] a multiple Arnoldi process is used. In that context the increase of the Krylov dimension in the higher stages is limited by a fixed value. The downside of this approach is that it is usually not known a priori how the bounds should be chosen; the corresponding value is most likely highly problem dependent. It should also be noted that in the context of ODEs the importance of considering variations in cost (as a function of the time step size) has been recognized in [10]. There are analytically derived estimates of the cost and, in line with the control theoretic approach to

Please cite this article in press as: L. Einkemmer, An adaptive step size controller for iterative implicit methods, Appl. Numer. Math. (2018), https://doi.org/10.1016/j.apnum.2018.06.002