Computers and Mathematics with Applications 71 (2016) 1137–1150

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Parallel exponential Rosenbrock methods

Vu Thai Luan [a,*], Alexander Ostermann [b]

[a] School of Natural Sciences, University of California, Merced, 5200 North Lake Road, Merced, CA 95343, USA

[b] Institut für Mathematik, Universität Innsbruck, Technikerstr. 13, A-6020 Innsbruck, Austria

ARTICLE INFO

Article history:
Received 23 September 2015
Received in revised form 27 December 2015
Accepted 23 January 2016
Available online 15 February 2016

Keywords:
Exponential Rosenbrock methods
Parallel exponential Rosenbrock integrators
Stiff order conditions
Multiprocessors
Parallel implementation
Speedup factor

ABSTRACT

Exponential Rosenbrock integrators were shown to be very efficient in solving large systems of stiff ordinary differential equations. So far, such exponential methods have been derived up to order 5. The aim of this paper is to construct new integrators of orders 4, 5, and 6. In contrast to the existing schemes, the new schemes, which are called parallel exponential Rosenbrock integrators, can be implemented on a multi-processor system or parallel computers. The new schemes of orders 4 and 5 require the same number of stages as the old schemes of the same orders of accuracy. However, while the parallel integrator of order 4 can be implemented with the same cost as a 2-stage method, the ones of orders 5 and 6 can be implemented at the cost of a 3-stage method only. This offers a significant improvement over the old schemes in terms of computational time when implemented in parallel. The numerical experiments show the efficiency of the new integrators as well as the comparative performance with the old ones.

1. Introduction

Discretizing a time dependent partial differential equation in space often yields a large system of stiff ordinary differential equations in the form of an initial value problem (IVP)

$$u'(t) = F(u(t)) = Au(t) + g(u(t)), \quad u(t_0) = u_0. \tag{1.1}$$

As of today, there are many numerical methods for solving the IVP (1.1). Due to the technological development of computer architectures, much attention has also been given to parallel methods for the numerical solution of IVPs. To the best of our knowledge, the paper by Nievergelt [1] is one of the earliest work on parallel methods for IVPs. In the last thirty years this area of research has received much interest. Many efficient methods were developed; see, for example, [2–11]. Most of the existing parallel methods for integrating IVPs are based on standard time integration techniques such as explicit or implicit Runge–Kutta methods, multistep methods, and general linear methods.

In this paper we are concerned with the construction of new parallel methods based on exponential Rosenbrock methods (see [12]). They belong to the class of exponential integrators. It is worth mentioning that in recent years exponential integrators turned out to be very competitive in solving (1.1) in the context of stiff problems, see for instance [13–19]. On the one hand, they can overcome the step size restrictions for explicit methods which are caused by stiffness. On the other hand, as they treat the nonlinearity in a fully explicit way, they do not require solving large nonlinear systems at every integration step like standard implicit methods. As they make use of the variation-of-constants formula to integrate the linear part of (1.1) exactly, they need to compute the exponential and related functions of A. In the case when A has a large

* Corresponding author.
E-mail addresses: vluan@ucmerced.edu (V.T. Luan), alexander.ostermann@uibk.ac.at (A. Ostermann).

http://dx.doi.org/10.1016/j.camwa.2016.01.020
0898-1221/

norm or is even an unbounded operator, this task was considered as a computational bottleneck of exponential integrators. This situation, however, has changed when Hochbruck and Lubich [13] published their landmark paper in 1997 where they analyzed Krylov subspace methods for efficiently computing the matrix exponential operator with large norm. Since then exponential integrators have become an active field of research in numerical analysis (see the review paper by Hochbruck and Ostermann [20]).

Exponential Rosenbrock methods are constructed by applying exponential Runge–Kutta methods [15] to a continuous linearization of the vector field F along the numerical solution u_n of (1.1). The linearized problem is given by

$$u'(t) = J_n u(t) + g_n(u(t)), \tag{1.2}$$

where J_n denotes (an approximation to) the Jacobian of F evaluated at u_n, and g_n is the nonlinear remainder, i.e.,

$$J_n = \frac{\partial F}{\partial u}(u_n), \qquad g_n(u) = F(u) - J_n u. \tag{1.3}$$

The idea of linearizing the vector field as (1.2) was first proposed by Pope [21]. The benefits of solving (1.2) in every integration step are the following. First, it offers a better stability when integrating the nonlinearity. Second, it can significantly simplify the order conditions. (Since $\frac{\partial g_n}{\partial u}(u_n) = 0$, the number of order conditions is much less than that of exponential Runge–Kutta methods applied directly to (1.1)). In [12] methods up to order 4 have been constructed. The implementation of exponential Rosenbrock methods was discussed in [22]. Recently we introduced a new theory for deriving stiff order conditions for exponential Rosenbrock methods of arbitrary order, see [23]. In that paper we gave the stiff order conditions for methods of order up to 6. For the construction of methods of order 5, we refer to [19]. Like the existing exponential schemes, however, the new methods of [19] do not allow parallelism across the stages and use a single processor only. It should be mentioned that there are only a few works focusing on parallel implementation of matrix functions for exponential integrators, see [24,25]. However, none of them deals with the construction of parallel exponential integrators.

Our idea for constructing parallel exponential Rosenbrock methods is to investigate the sparsity structure of the coefficient matrix of the method. When solving the order conditions one typically has to work with coefficients that can be taken as free parameters. By choosing the right ones and then zeroing them, we are able to derive several internal stages of methods of orders 4, 5, and 6 that are independent of each other. As a consequence, such stages can be implemented in parallel using multiprocessors leading to parallel methods. For example, we have constructed a 3-stage parallel scheme of order 4 that can be implemented on two processors with the same cost as 2-stage methods. For methods of order 5, two parallel schemes are constructed using 4 and 5 stages which can be implemented on two and three processors with the same cost as 3-stage methods. Finally, we derived a sixth-order 7-stage parallel scheme that can be implemented like 3-stage methods by using four processors. As we can see, the parallel schemes use the same number of stages as the non-parallel schemes which have the same orders. Therefore, when implemented in parallel, clearly the parallel schemes can compute the solution much faster than the non-parallel ones.

The remainder of the paper is organized as follows. In Section 2 we briefly describe the class of exponential Rosenbrock methods. Our motivation and ideas for constructing parallel exponential Rosenbrock schemes are presented in Section 3. In Section 4 we recall the stiff order conditions that allow us to construct high order methods. Section 5 contains the main results of the paper. Namely, we derive parallel integrators of orders 4, 5, and 6: `pexprb43`, `pexprb54s4`, `pexprb54s5`, and `pexprb65s7`. Moreover, we prove that there does not exist an exponential Rosenbrock method of order 6 with less than or equal to 6 stages (Theorem 5.1). Section 6 is devoted to numerical results. We show that the parallel schemes of orders 4 and 5, when implemented in serial, perform similarly as their corresponding non-parallel schemes of the same orders `exprb43`, `exprb54s4`, `exprb54s5` which were constructed in [19]. When implemented in parallel, however, the parallel schemes are shown to be much more efficient than the corresponding non-parallel ones in terms of total CPU time.

2. Exponential Rosenbrock schemes

Consider the numerical solution of (1.1) by (explicit) exponential Rosenbrock methods. These schemes can be represented in the following form (see [12, Section 2.2]):

$$U_{ni} = u_n + c_i h_n \varphi_1(c_i h_n J_n) F(u_n) + h_n \sum_{j=2}^{i-1} a_{ij}(h_n J_n) D_{nj}, \tag{2.1a}$$

$$u_{n+1} = u_n + h_n \varphi_1(h_n J_n) F(u_n) + h_n \sum_{i=2}^{s} b_i(h_n J_n) D_{ni} \tag{2.1b}$$

with

$$D_{ni} = g_n(U_{ni}) - g_n(u_n), \quad 2 \le i \le s. \tag{2.1c}$$

Here $u_n \approx u(t_n)$, c_i are the nodes, s is the number of stages ($s-1$ internal stages and one external one), $U_{ni} \approx u(t_n + c_i h_n)$, J_n and g_n are given in (1.3), $h_n = t_{n+1} - t_n > 0$ denotes the time step. The coefficients $a_{ij}(z)$ and $b_i(z)$ are usually chosen as