Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

Workshop innovative time integrators; (Amsterdam, October 30 - November 1, 1996)

J.G. Verwer, J.M. Sanz-Serna (eds.)

Department of Numerical Mathematics

NM-N9602 1996

Report NM-N9602
ISSN 0169-0388

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Workshop Innovative Time Integrators

CWI, Amsterdam, October 30 - November 1, 1996
Organized by CWI and the University of Valladolid

1 Themes and Goal

Traditionally, time-stepping for ODEs has been carried out by general black boxes of wide applicability. There are general solvers and stiff solvers, but, except for this distinction, time-stepping is as a rule performed with methods that are problem independent. However many applications require special, problem-dependent methods. Some applications refer to situations where different physical phenomena, involving very different time scales, occur simultaneously. In other cases, like in many molecular dynamics simulations, the time-stepping is performed to obtain an idea of an average long-term behaviour and no accuracy in the conventional sense is required. New classes of problems, like Hamiltonian problems, are receiving much attention. It is clear that conventional software is not meant to cater for all these special situations and new solution techniques are required.

These considerations have a direct impact on PDEs. For time-dependent PDE problems, a flexible and general approach is given by the Method of Lines, where the discretization of spatial operators and the numerical time integration are considered separately. This approach has made it possible to time-step in PDEs, notably in parabolic problems, with the ODE packages. Due to the rapid growth in computer resources, an automatic black box solution is now possible for a variety of systems of nonlinear, one-space dimensional PDE problems. In more space dimensions, however, the automatic solution becomes much more intricate. Apart from grid generation difficulties, the use of implicit integration formulas requires much more attention than in one dimension. As a rule, one has to resort to iterative methods for the obtention of the implicitly defined approximations. Additional difficulties are due to the presence of several time-scales in many physical phenomena or for the need for long-term integrations. For large systems of PDEs, describing combinations of different phenomena, e.g. advection, diffusion and chemistry, it readily becomes more advisable to resort to special techniques tailored to the problem at hand. Examples are provided by implicit-explicit formulas, time or operator splitting techniques, special purpose iterative solvers connected with domain decomposition, etc.

The goal of this workshop is to bring together numerical analysts working on numerical algorithms using nonstandard ODE and PDE time-stepping techniques, to determine the state of the art and to exchange latest research results. Keywords for lectures may be classified into two main categories:

- Aspects of special purpose time integration methods for systems of multi-space dimensional time-dependent PDE problems, and

- Long time integration issues, dynamical systems and multiple time-scales.

2 Organization and Funding

The workshop is organized by CWI and the University of Valladolid, Spain. The organizers are Dr. J.G. Verwer (Department of Numerical Mathematics, CWI, email: janv@cwi.nl, Tel. +31205924095) and Prof. dr. J.M. Sanz-Serna (Dpto. Matematica Aplicada y Computacion, Facultad de Ciencias, University of Valladolid, email: sanzserna@cpd.uva.es, Tel. +3483423183). Mrs. Simone Panka (secretary CWI, email: simone@cwi.nl, Tel. +31205924189) takes care of administrative tasks and local organizational matters. The organizers gratefully acknowledge financial support from

- The European Union through the HCM Program "The Equations of Fluid Mechanics and Related Subjects", Nr. CHRX-CT-93-0407,
- CWI and
- The University of Valladolid.

3 Registered Participants and Lectures

The number of registered participants equals 39; 34 participants will present a lecture. The titles of lectures and author names are given by (first author) alphabetical order. In case of more than one author, the name of the lecturer is underlined. Abstracts with full addresses can be found in the indicated sections of Section 5.

Section 5.1: A.L. Araújo and J.M. Sanz-Serna, University of Valladolid, Spain
On symplectic additive methods

Section 5.2: U. Ascher, University of British Columbia, Vancouver, Canada
Implicit-explicit methods for time-dependent PDEs

Section 5.3: M. Berzins, J.M. Ware and I. Ahmad, University of Leeds, England
Innovative time integration methods for solving convection and convection reaction problems using the M.O.L.

Section 5.4: L. Brugnano and D. Trigiante, University of Firenze, Italy
Packages BVMpack and BVMpack-p for ODEs

Section 5.5: M.P. Calvo, A. Iserles and A. Zanna, University of Valladolid, Spain
Conservative integrators for the Toda lattice equations

Section 5.6: J.R. Cash, Imperial College, London, England
Innovative time integrators

Section 5.7: Ch. Engstler and Ch. Lubich, University of Tübingen, Germany
Multirate extrapolation methods for differential equations with different time scales

Section 5.8: G. Fairweather, Colorado School of Mines, Colorado, U.S.A.
Alternating direction orthogonal spline collocation methods for initial/boundary value problems

Section 5.9: J.A. Ferreira, University of Coimbra, Portugal
On moving finite difference methods for partial differential equations on two-dimensional domains

Section 5.10: J.E. Flaherty, Rensselaer Polytechnic Institute, Troy, New York, U.S.A.
Adaptive and parallel computational techniques with applications in materials science

Section 5.11: J. de Frutos and J. Novo, University of Valladolid, Spain
Postprocessing spectral methods in the time integration of dissipative partial differential equations

Section 5.12: I. Gladwell, K. Bouas-Dockery and R.W. Brankin,
Southern Methodist University, Dallas, U.S.A.
A Fortran 90 mechanical system solver

Section 5.13: S.K. Gray, Argonne National Laboratory, Argonne, U.S.A.
Symplectic integrators for quantum dynamics

Section 5.14: E. Hairer, Université de Genève, Switzerland
Variable step size integration with symplectic methods

Section 5.15: D. Higham, University of Dundee, Scotland
Adaptive time-stepping and long term dynamics

Section 5.16: M. Hochbruch and Ch. Lubich, University of Tübingen, Germany
Exponential integrators for large systems of ODEs

Section 5.17.: P.J. van der Houwen, CWI, Amsterdam, The Netherlands
Solving Implicit Differential Equations on Parallel Computers

Section 5.18: W. Hundsdorfer, CWI, Amsterdam, The Netherlands
Stability results for implicit-explicit linear 2-step methods

Section 5.19: R.I. McLachlan, Massey University, Palmerston North, New Zealand
Composition of maps as integrators

Section 5.20.: V.I. Lebedev, Institute of Numerical Mathematics, Moscow, Russia
Explicit difference schemes with variable time steps for solving stiff systems of equations

Section 5.21: M.A. López-Marcos, J.M. Sanz-Serna and R.D. Skeel, University of Valladolid, Spain
Improving the accuracy of symplectic integrators

Section 5.22: A. Kværnø and S.P. Nørsett, Trondheim University, Norway
GODESS, an ODE-solver framework

Section 5.23: D.I. Okunbor, University of Missouri-Rolla, Rolla, U.S.A.
Efficient integration methods for many-particle problems

Section 5.24: G.R.W. Quispel, La Trobe University, Melbourne, Australia
Numerical integration of ODE's, while preserving volume, symmetries, or first integrals

Section 5.25: S. Reich, Konrad-Zuse Center, Berlin, Germany
Molecular dynamics simulations and statistical mechanics

Section 5.26: P. Rentrop, TH Darmstadt, Germany
The DAE aspect of the charge cycle in a combustion engine

Section 5.27: T. Schlick, Chemistry Department and Courant Institute of Mathematical Sciences,
New York University and Howard Hughes Medical Institute, New York, U.S.A.
Long-time integration schemes for molecular dynamics

Section 5.28: B.A. Schmitt and R. Weiner, Martin-Luther-University, Halle, Germany
Krylov-W methods for large stiff equations

Section 5.29: R.D. Skeel, University of Illinois at Urbana-Champaign, U.S.A.
Cheap implicit symplectic integrators

Section 5.30: B.P. Sommeijer, CWI, Amsterdam, The Netherlands
Vector/parallel time integration methods for 3D bio-chemical transport in shallow water

Section 5.31: M.N. Spijker, University of Leiden, The Netherlands
Linear stability analysis for discretizations of initial value problems

Section 5.32: K. Strehmel, University of Halle, Germany
Linear partial differential-algebraic equations and their indices

Section 5.33: S. Vandewalle, K.U. Leuven, Belgium
Waveform relaxation methods for time-dependent partial differential equations: recent developments

Section 5.34: J. Yen and L.R. Petzold, University of Minnesota, Minneapolis, U.S.A.
Numerical methods for nonlinear oscillatory multibody dynamic systems

The other registered participants are

35. J.M. Hyman, Los Alamos National Laboratory, M/S B284, Group T-7, Los Alamos, NM 87545, U.S.A.
36. P. de Oliveira, University of Coimbra, Department of Mathematics, 3000 Coimbra, Portugal
37. NN, University of Coimbra, Department of Mathematics, 3000 Coimbra, Portugal
38. J.M. Sanz-Serna, University of Valladolid, Spain (organizer)
39. J.G. Verwer, CWI, Amsterdam, The Netherlands (organizer)

4 Time Table

All lectures have an equal length of 30 minutes including discussion. Bold names in the time table are names of lecturers. Bracketed numbers following names of lecturers are corresponding section numbers of Section 5 where full addresses and abstracts can be found. The organizers have taken the freedom to designate section leaders. Their names are also listed in the time table with the abbreviation SL (Section Leader) in front. If someone is not willing or not able to act as a section leader, then please inform the organizers timely.

Session	Time	Wednesday Oct. 30	Thursday Oct. 31	Friday Nov. 1
I	09.00	Registration	SL Lubich **Gladwell** (5.12) **Noersett** (5.22)	SL Ascher **Flaherty** (5.10) **Ferreira** (5.9)
Break	10.00	Opening (Jan)	Coffee break	Coffee break
II	10.15	SL Sanz-Serna **Van der Houwen** (5.17) **Spijker** (5.31)	SL Strehmel **Skeel** (5.29) **Gray** (5.13)	SL Schlick **McLachlan** (5.19) **Araújo** (5.1)
Break	11.15	Coffee break	Coffee break	Coffee break
III	11.30	SL Verwer **López – Marcos** (5.21) **Okunbor** (5.23)	SL Gladwell **Trigiante** (5.4) **De Frutos** (5.11)	SL Flaherty **Berzins** (5.3) **Sommeijer** (5.30)
Break	12.30	Lunch break	Lunch break	Lunch break
IV	14.00	SL Spijker **Hairer** (5.14) **Strehmel** (5.32)	SL Rentrop **Ascher** (5.2) **Hundsdorfer** (5.18)	SL Hairer **Lubich** (5.7) **Cash** (5.6)
Break	15.00	Tea break	Tea break	Tea break
V	15.15	SL Van der Houwen **Petzold** (5.34) **Rentrop** (5.26)	SL Skeel **Reich** (5.25) **Schlick** (5.27)	SL Berzins **Calvo** (5.5) **Quispel** (5.24)
Break	16.15	Coffee break	Coffee break	Coffee break
VI	16.30	SL Petzold **Lubich** (5.16) **Weiner** (5.28)	SL Noersett **Lebedev** (5.20) **Vandewalle** (5.33)	SL Cash **Higham** (5.15) **Fairweather** (5.8)
	17.30	Reception		Closure (Chus)
	19.30			Workshop dinner

Transport

For transport from the hotel to CWI and back we have arranged a bus. The bus will leave from the hotel on Wednesday morning (October 30) at 08.45 hours; the bus will bring you back to the hotel at 18.45 hours (after the reception). On Thursday and Friday the bus will leave at 08.30 hours from the hotel and will bring you back to the hotel at 17.45 hours.

5 Abstracts

The abstracts are listed in the same by (first author) alphabetical order as the titles in Section 2. The name of the lecturer is here underlined and given in bold face in the time table of Section 4.

5.1 On symplectic additive methods

A.L. Araújo
Departamento de Matemática, Universidade de Coimbra, 3000 Coimbra, Portugal
E-mail: alma@mat.uc.pt
J.M. Sanz-Serna
Departamento de Matemática Aplicada y Computación, Universidad de Valladolid, España
E-mail: sanzserna@cpd.uva.es

Introduction

We consider D-dimensional systems of differential equations

$$\frac{dy}{dt} = f(y), \tag{1}$$

where f can be decomposed in the form

$$f(y) = \sum_{\nu=1}^{N} f^{[\nu]}(y). \tag{2}$$

We are concerned with numerical methods for (1) that take advantage of the decomposition (2). An important application is given by *splitting methods*: if the systems

$$\frac{dy}{dt} = f^{[\nu]}(y), \qquad \nu = 1, \dots, N, \tag{3}$$

are integrable in closed form, it is possible to integrate (1) by combining the individual solutions of (3). Splitting methods are often used in time-dependent partial differential equations; the different $f^{[\nu]}$ may correspond to different spatial variables or to different physical contributions. Cooper and Sayfy [CS80] have considered the class of such *additive methods*, including (N-part) additive Runge-Kutta (ARK$_N$) methods. A step $y_n \mapsto y_{n+1}$ of the s-stage ARK$_N$ method specified by

$$\begin{aligned} Y_{n,i} &= y_n + h \sum_{\nu=1}^{N} \sum_{j=1}^{s} a_{ij}^{[\nu]} f^{[\nu]}(Y_{n,j}), & (4) \\ y_{n+1} &= y_n + h \sum_{\nu=1}^{N} \sum_{i=1}^{s} b_i^{[\nu]} f^{[\nu]}(Y_{n,i}). & (5) \end{aligned}$$

The aim of this work is the study of numerical methods based on decompositions (2) used for the integration of differential systems (1) of Hamiltonian type [SSC93].

We are specially interessed in the study of numerical methods that integrate differential systems of Hamiltonian type by taking advantage of a decomposition of the right-hand side. The main result of this work is a necessary and sufficient condition for an NB-series to be a symplectic method. This result will be particularized for the general class of additive Runge-Kutta methods. An algorithm for the construction of symplectic methods of the mentioned class is also included. The results presented on this work are essentially included in [AMSS95].

Preliminaries

An N-tree is a tree where each vertex has been assigned, out of a choice of N, a colour or a type. The terms tree and N-tree are always assumed to refer to *rooted* graphs. We respectively denote by NT, $\overline{NT}$ the sets of N-trees and nonempty N-trees. If $u \in \overline{NT}$ and $v \in NT$, we denote by $u \cdot v$ their Butcher product.

Let us consider a mapping $\mathbf{c}$ that assigns to each $u \in NT$ a real number $\mathbf{c}(u)$. An NB-series relative to the decomposition (2) is a formal power series

$$NB(\mathbf{c}, y) = \sum_{u \in NT} \frac{h^{\rho(u)}}{\sigma(u)} \mathbf{c}(u) F(u)(y(t)), \tag{6}$$

where $\rho(u)$ is the order of the N-tree, $\sigma(u)$ the number of symmetries and $F(u)(y)$ is the elementary differential.

Symplectic NB-series

Assume that (1) is a Hamiltonian system, i.e., that D is even $D = 2d$ and that there exists a real-valued function $H(y)$ such that

$$f(y) = J^{-1} \frac{\partial H}{\partial y}(y), \tag{7}$$

where $\partial H/\partial y = (\partial H/\partial y^1, \dots, \partial H/\partial y^{2d})^T$ and J is the $2d \times 2d$ matrix

$$J = \begin{bmatrix} 0_d & I_d \\ -I_d & 0_d \end{bmatrix}.$$

A transformation $\psi : \mathcal{R}^{2d} \to \mathcal{R}^{2d}$ is symplectic if its Jacobian ψ' satisfies

$$\psi'(y)^T J \psi'(y) = J. \tag{8}$$

We are going to find conditions on the coefficients $\mathbf{c}(u)$ of an NB-series $NB(\mathbf{c}, y)$ to be symplectic. Suppose f is Hamiltonian, so that (7) holds, but the parts $f^{[\nu]}$ themselves are not assumed to be Hamiltonian. We have the following theorem which generalizes the results in [CSS94] and [Mu94] for the case of B-series and P-series, respectively.

Theorem 1 *Consider a sequence of coefficients* $\mathbf{c}$ *with* $\mathbf{c}(\emptyset) = 1$. *The NB-series* $NB(\mathbf{c}, y)$ *is canonical for all Hamiltonian systems (1), (7) and all decompositions (2) in not necessarily Hamiltonian parts if and only if, for each pair of nonempty* N*-trees* w, z

$$\mathbf{c}(w \cdot z) + \mathbf{c}(z \cdot w) = \mathbf{c}(w)\mathbf{c}(z), \tag{9}$$

and for each pair of nonempty N*-trees* u, v *that only differ in the colour of their roots*

$$\mathbf{c}(u) = \mathbf{c}(v). \tag{10}$$

Now, let us consider the situation where the individual parts

$$f^{[\nu]}(y) = J^{-1} \frac{\partial H^{[\nu]}}{\partial y}(y),$$

with $H(y) = H^{[1]}(y) + \cdots + H^{[N]}(y)$, are themselves Hamiltonian.

Theorem 2 *Consider a sequence of coefficients* $\mathbf{c}$ *with* $\mathbf{c}(\emptyset) = 1$. *Then, the corresponding NB-series is symplectic for arbitrary Hamiltonian problems (1), (7) arbitrarily decomposed in Hamiltonian parts, if and only if for each pair of nonempty trees,* $\mathbf{c}(u \cdot v) + \mathbf{c}(v \cdot u) = \mathbf{c}(u) + \mathbf{c}(v)$.

Application to Additive Runge-Kutta methods

We begin by a sufficient condition for an ARK$_N$ method of s stages, applied to the Hamiltonian system (1), (7) to be symplectic.

Theorem 3 *(i) Assume that the system (1) is Hamiltonian with a general decomposition (2), with f, $f^{[\nu]}$ of class C^1 in an open subset of $\mathcal{R}^{2d}$. Assume that, for a given h, the formulae (4), (5) define a mapping $\psi : y_{n+1} \mapsto y_n$ in an open subset of $\mathcal{R}^{2d}$. Then the conditions*

$$b_i^{[\nu]} = b_i^{[\mu]}, \qquad 1 \leq i \leq s, \quad 1 \leq \nu, \mu \leq N \tag{11}$$

and

$$b_i^{[\nu]} a_{ij}^{[\mu]} + b_j^{[\mu]} a_{ji}^{[\nu]} - b_i^{[\mu]} b_j^{[\nu]} = 0, \qquad 1 \leq i,j \leq s, \quad 1 \leq \nu, \mu \leq N, \tag{12}$$

are sufficient for ψ to be a symplectic transformation. (ii) If in (i) the parts $f^{[\nu]}$ are themselves Hamiltonian, then (12), on its own, is sufficient for ψ to be symplectic.

Following an NB-series approach, we may state the next result that gives us the conditions the NB-series associated with an ARK$_N$ method must verify to be symplectic.

Theorem 4 *Assume that the ARK$_N$ method (4)–(5) is S-irreducible. Then the necessary and sufficient condition (9) for the symplecticness of the correspondent NB-series for arbitrary Hamiltonian systems (1), (7) decomposed in Hamiltonian parts is equivalent to (12). If the parts are nonHamiltonian, the condition (10) is equivalent to (11).*

Finally we will find a class of ARK$_N$ methods (4)– (5) that are symplectic when applied to Hamiltonian systems with a Hamiltonian decomposition. We will show that, for this class, the relations (12) admit $\dfrac{s(s-1)}{2}$ free parameters which are the values $a_{ji}^{[N]}$, with $i < j$. Observe that this result generalizes the result obtained in [SSM92] for the standard Runge-Kutta case.

References

[AMSS95] A.L. Araújo, A. Murua and J.M. Sanz-Serna (1995) *Symplectic methods based on decompositions*, Applied Mathematics and Computation Reports, Report 1995/10, Universidad de Valladolid, Spain.

[CSS94] M.P. Calvo and J.M. Sanz-Serna (1994), *Canonical B-series*, Numer. Math. 67, 161-175.

[CS80] G.J. Cooper and A. Sayfy (1980), *Additive methods for the numerical solution of ordinary differential equations*, Math. Comput. 35, 1159-1172.

[Jo92] J.C. Jorge (1992), Los Métodos de Pasos Fraccionarios para la Integración de Problemas Lineales: Formulación General, Análisis de la Convergencia y Diseño de Nuevos Métodos, PhD thesis, Universidad de Zaragoza, Zaragoza, España.

[Mu94] A. Murua (1994), Métodos Simpléticos Desarrollables en P-series, PhD thesis, Universidad de Valladolid, Valladolid, España.

[SSM92] S. Saito, H. Sugiura and T. Mitsui (1992), *Family of symplectic implicit Runge-Kutta formulae*, BIT 32, 539-543.

[SSC93] J.M. Sanz-Serna and M.P. Calvo (1993), Numerical Hamiltonian Problems, Chapman and Hall, London.

5.2 Implicit-explicit methods for time-dependent PDEs

Uri Ascher
Department of Computer Science and Institute of Applied Mathematics, University of British Columbia,
Vancouver, B.C., Canada V6T 1Z4
E-mail: ascher@cs.ubc.ca

Various methods have been proposed to integrate systems of ordinary differential equations arising from spatially discretized time dependent PDEs. For problems with terms of different types, implicit-explicit (IMEX) schemes have been used, especially in conjunction with spectral methods. For convection-diffusion problems, for example, one would use an explicit scheme for the convection term and an implicit scheme for the diffusion term. Reaction-diffusion problems can also be approximated in this manner. In this work we analyze the performance of such schemes and experiment with them to confirm theory's predictions.

For the prototype linear advection-diffusion equation, a stability analysis for first, second, third and fourth order multistep IMEX schemes is performed. Stable schemes permitting large time steps for a wide variety of problems and yielding appropriate decay of high frequency error modes are identified.

Numerical experiments demonstrate that weak decay of high frequency modes can lead to extra iterations on the finest grid when using multigrid computations with finite difference spatial discretization, and to aliasing when using spectral collocation. When this behaviour occurs, use of weakly damping schemes such as the popular combination of Crank-Nicolson with second order Adams-Bashforth is discouraged.

Our findings are demonstrated on several examples.

The talk will mostly cover results from the paper by U. Ascher, S. Ruuth and B. Wetton, “Implicit-explicit methods for time-dependent PDE's”, SIAM J. Numer. Anal. 32 (1995), 797-823. Further results will be discussed as well.

Joint work with Steve Ruuth and Brian Wetton

5.3 Innovative Time Integration Methods for Solving Convection and Convection Reaction Problems Using the M.O.L.

M. Berzins, J.M. Ware and I. Ahmad
School of Computer Studies, The University of Leeds, Leeds LS2 9JT, United Kingdom
E-mail: martin@dcre.leeds.ac.uk

Introduction

The method of lines for solving time dependent p.d.e.s may be described as 'discretize in space and then integrate in time', usually using o.d.e. initial value software. The talk described by this abstract will address the issue of how to solve large problems arising from combustion and atmospheric dispersion problems using a method of lines type approach. The main focus of the talk will be to show that it is possible to modify a standard method of lines type approach to include an error control which takes account of the spatial error already present and to use a modest amount of problem-dependent information to increase the efficiency of the solver by an order of magnitude. This last increase in efficiency is achieved by combining an operator splitting approach with a novel time integration method.

Spatial and Temporal Discretisation Methods

The finite volume spatial discretization methods used here are commonly employed for the semi-discretisation of fluid flow equations. Examples of the types of methods often applied to regular meshes are given by [1] and in the references therein. For problems with very general spatial geometries one important approach is to consider methods which are based on unstructured triangular or tetrahedral meshes. Although finite element and finite volume schemes based on unstructured triangular meshes have been used for many years, only recently have a number of high-order cell-centred finite volume schemes been developed for convection dominated problems, see [4]. The precise details of this scheme are not particularly important but the form of the p.d.e. problem class addressed is critical to the development of efficient methods. Considering for example the class of scalar p.d.e.s:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(x,y,u,\frac{\partial u}{\partial x},\frac{\partial u}{\partial y}) + \frac{\partial}{\partial y} g(x,y,u,\frac{\partial u}{\partial x},\frac{\partial u}{\partial y}) = h(x,y,u) \tag{1}$$

where f and g are the flux functions in x and y respectively, the source term is h, which contains the chemical reaction terms for atmospheric dispersion and combustion problems, and appropriate boundary and initial conditions are supplied.

The spatial discretization scheme of [4] results in a system of differential equations, which can be written as the initial value problem:

$$\dot{\underline{U}} \;=\; \underline{F}_N \;\;(\; t,\; \underline{U}(t) \;)\;\;,\; \underline{U}(0) \;\; \text{given}\;, \tag{2}$$

where the vector, $\underline{U}(t)$, is defined by $\underline{U}(t) \;=\; [U(x_1,y_1,t),...,U(x_N,y_N,t)\;]^T$. The point x_i, y_i is the centre of the i th cell and $U_i(t)$ is a numerical approximation to the exact solution to the p.d.e. evaluated at the centroid i.e. $u(x_i,y_i,t)$. A method of lines approach is used to numerically integrate equation (2) thus generating an approximation, $\underline{V}(t)$, to the vector of exact p.d.e. solution values at the mesh points, $\underline{u}(t)$. At present two time integration methods are used: Theta method with functional and/or Newton iteration or the backward differentiation formula method using Newton Krylov methods [7]. Both codes allow automatic control of the local error. The Theta method code, see [1, 2], defines the numerical solution at $t_{n+1} = t_n + k$, where k is the time step size, as denoted by $\underline{V}(t_{n+1})$, by

$$\underline{V}(t_{n+1}) \;=\; \underline{V}(t_n) \;+\; (1-\theta)k\;\dot{\underline{V}}(t_n) + \theta\;k\;\underline{F}_N(t_{n+1}, \underline{V}(t_{n+1})), \tag{3}$$

in which $\underline{V}(t_n)$ and $\dot{\underline{V}}(t_n)$ are the numerical solution and its time derivative at the previous time t_n and $\theta = 0.55$. This system of equations is solved by either using functional iteration or a Newton Krylov method or by automatically switching between the two methods, [2]. Suppose that the local error is denoted by $\underline{l}_{n+1}(t_{n+1})$ and that a standard local error control i.e. $\|\; \underline{l}_{n+1}(t_{n+1}) \;\| < TOL$ is used. In this case it is difficult to establish a relationship between the accuracy tolerance, TOL , and the global time error. It is thus important to use an error control which reflects the spatial and temporal nature of the method of lines approach.

Space-Time Error Control

The global error in the numerical solution can be expressed as the sum of the spatial discretization error, $\underline{e}(t) = \underline{u}(t) - \underline{U}(t)$, and the global time error, $\underline{g}(t) = \underline{U}(t) - \underline{V}(t)$. That is,

$$\begin{aligned} \underline{E}(t) = \underline{u}(t) - \underline{V}(t) &= (\underline{u}(t) - \underline{U}(t)) + (\underline{U}(t) - \underline{V}(t)) \\ &= \underline{e}(t) + \underline{g}(t). \end{aligned} \tag{4}$$

Efficient time integration requires that the spatial and temporal are roughly the same order of magnitude. The need for spatial error estimates to be unpolluted by temporal error requires the spatial error to be the larger of the two errors. One approach for achieving this is described by Berzins [1] who balances the spatial and temporal errors by controlling the local time error to be a fraction of the local growth in the spatial discretization error. The local-in-time spatial error, $\hat{\underline{e}}(t_{n+1})$, for the timestep from t_n to t_{n+1} is defined as the spatial error at time t_{n+1} given the assumption that the spatial error, $\underline{e}(t_n)$, is zero. A local error balancing approach is then:

$$\|\; \underline{l}_{n+1}(t_{n+1}) \;\| \;<\; \epsilon \,\|\; \hat{\underline{e}}(t_{n+1}) \;\|, \quad 0 < \epsilon < 1. \tag{5}$$

The error $\hat{\underline{e}}(t_{n+1})$ is estimated by the difference between the computed solution and the first-order solution which satisfies a modified o.d.e. system:

$$\dot{\underline{v}}_{n+1}\;(t) = \underline{G}_N(t, \underline{v}_{n+1}(t)), \tag{6}$$

where $\underline{v}_{n+1}(t_n) = \underline{V}(t_n)$, $\dot{\underline{v}}_{n+1}(t_n) = \underline{G}_N(t, \underline{V}(t_n))$ and where $\underline{G}_N(.,.)$ is obtained is obtained by using a different order (higher) spatial discretisation method. The local-in-time space error is then estimated by

$$\hat{\underline{e}}(t_{n+1}) = \underline{V}(t_{n+1}) - \underline{v}_{n+1}(t_{n+1}) \tag{7}$$

and is computed by applying the Theta method, with one functional iteration or one backsolve, to equation (6) which thus requires only one evaluation of $\underline{G}_N(.,.)$ per timestep. One effect of using this error estimate is that the source term is not taken into account as the same solution value is applied to both the o.d.e. systems used in the error estimate. Hence an extension of the existing approach is needed to cope with this situation. This extension is work in progress which will be reported at the meeting.

Operator Splitting

Although the method of lines approach may be applied to the large o.d.e. systems that result from the discretisation of combustion or atmospheric dsipersion problems (at least 50,000 equations typically) the c.p.u. times seem to be excessive even when using Newton-Krylov methods to solve the nonlinear equations. One approach which overcomes this is to use a form of operator splitting based on a decomposition of the p.d.e. s into a set of flow terms and a source reactive term. A number of such splitting methods are described by Chung [5]. Consider the o.d.e. function defined by equation (9) and decompose it into two parts:

$$\underline{F}_N \ (t, \ \underline{U}(t) \ ) \ = \ \underline{F}_N^f \ (t, \ \underline{U}(t) \ ) \ + \ \underline{F}_N^s \ (t, \ \underline{U}(t) \ ) \qquad (8)$$

where $\underline{F}_N^f \ (t, \ \underline{U}(t) \ )$ represents the discretisation of the convective flux terms f and g in equation (1) and $\underline{F}_N^s \ (t, \ \underline{U}(t) \ )$ represents the discretization of the of the source term h in the same equation. In order to apply operator splitting consider now the equations to be solved for the correction to the solution $\underline{\Delta V}$ for the $p+1$ th iteration of the modified Newton iteration used with the Theta method:

$$[I - k\theta J] \ \underline{\Delta V} \ = \ \underline{r} \, (t_{n+1}^p) \qquad (9)$$

where

$$\underline{r} \, (t_{n+1}^p) \ = \ -\underline{V}(t_{n+1}^p) + \underline{V}(t_n) + (1-\theta)k\dot{\underline{V}}(t_n) - \theta k \underline{F}_N(t_{n+1}, \underline{V}(t_{n+1}^p)),$$

$$J \ = \frac{\partial \ \underline{F}_N}{\partial \underline{U}} \ , \quad \text{and} \ \ \underline{\Delta V} \ = \ \left[\underline{V}(t_{n+1}^{p+1}) - \underline{V}(t_{n+1}^p)\right]. \qquad (10)$$

A standard splitting approach, [5], is to employ the following approximation to the Jacobian matrix used by the Theta method within a Newton iteration:

$$I - k\theta J \ \approx \ [I - k\theta \ J_f] \ \ [I - k\theta \ J_s \ ] \ ) \ \ + \ \ O(k^2). \qquad (11)$$

where

$$J_f \ = \ \frac{\partial \ \underline{F}_N^f}{\partial \underline{U}} \ , \ \ J_s \ = \ \frac{\partial \underline{F}_N^s}{\partial \underline{U}}$$

The disadvantage of this is that it introduces a second-order splitting error, which in the case of the Theta method is of the same order as the local error. It is however possible to estimate the effect of this splitting error, see [3]. The matrix $I - k\theta J_s$ which is the Jacobian of the discretization of the time derivatives and the source term h. The matrix $I - k\theta J_s$ is thus block diagonal and each block may be inverted (or the equations solved) independently. The remaining part of the Jacobian $I - k\theta J_f$ has only flow equation terms from the Euler equations and so may be approximated by the identity matrix, [2]. The new iteration may then be formally written as

$$\underline{\Delta V^*} \ = \ [I - k\theta J_s]^{-1} \ \underline{r} \, (t_{n+1}^p) \qquad (12)$$

where $\underline{\Delta V^*}$ is the operator splitting approximation to $\underline{\Delta V}$. The inversion of the source-term matrix is easy for some combustion problems, see [3], but for atmospheric dispersion problems rather than inverting the matrix it is easier to solve the appropriate equations, either by direct or iterative methods, see [8] . The advantage of this is that functional iteration may be used to solve the nonlinear equations provided that the residuals on each triangle are multiplied by the inverse of the matrix $I - k\theta J_s$, or the appropriate equations solved. This modified form of functional iteration has been implemented as a new linear algebra module inside the software and has been found to give increases in speed-up by a factor of five over using a Newton-Krylov method for the combustion problem by [3]. Experiments are currently being conducted to apply the same method to atmospheric dispersion problems. The convergence properties of this iteration may be analyzed using standard theory (e.g. Ortega and Rheinboldt).

In the case of atmospheric dispersion problems it is not clear that the theta method is the most appropriate time integration method. The talk will therefore also consider modifications of the approach of Verwer [8] as the method of integration and present the results of experiments on realistic atmospheric dispersion problems.

Conclusions

The talk will show how a traditional method of lines approach has been modified to include novel methods for spatial and temporal error balancing and operator splitting. Early results suggest that the method is more reliable than fixed space and time step methods and provided that the splitting approach is used does not have as high a computational cost as a traditional m.o.l. code.

References

[1] M. Berzins, Temporal error control for convection-dominated equations in two space dimensions, *SIAM Jour. of Sci. Comp.* **16** (1995) 558–581.

[2] M. Berzins and R.M. Furzerland, An adaptive theta methods for the solution of stiff and non-stiff differential equations, *Applied Numerical Mathematics* **9** (1992), 1–19.

[3] M. Berzins and J. Ware, Solving Convection and Convection Reaction Problems using M.O.L., Paper submitted to *Applied Numerical Mathematics*

[4] M. Berzins and J.M. Ware, Positive discretization methods for hyperbolic equations on irregular meshes, *Applied Numerical Mathematics* **16** (1995), 417–438.

[5] T.J. Chung (ed), Numerical Modeling in Combustion, Taylor and Francis, Washington D.C. (1993).

[6] J. Lawson, M. Berzins and P.M. Dew, Balancing Space and Time Errors for Parabolic Equations, *SIAM J. Sci. Stat. Comput.* **12** (1991) 573–594.

[7] W.L. Seward and K.R. Jackson, Adaptive linear equations solvers in codes for large stiff systems of o.d.e.s, *SIAM Jour. of Sci. Comp.* **14** (1993) 800–823.

[8] J.G. Verwer, Gauss Seidel Iteration for Stiff ODEs from Chemical Kinetics, *SIAM Jour. of Sci. Comp.* **15** (1994) 1243–1250.

5.4 PACKAGES BVMpack AND BVMpack-p FOR ODEs

L. Brugnano and D. Trigiante
Dipartimento di Energetica, Via Lombroso 6/17, 50134 Firenze, Italy
E-mail: mbxtrigiante@mail.cnuce.cnr.it

Introduction

The packages BVMpack and BVMpack-p are packages designed to solve ODEs on sequential and parallel computers, respectively. The current prototypes are able to treat efficiently the following problems:

1. classical dissipative problems;
2. Hamiltonian problems;
3. boundary value problems;
4. differential-algebraic problems.

The extension to delay ODEs will be considered as next step.

Mathematical background

The codes are based on symmetric LMF formulae, *symmetric schemes* hereafter, used as Boundary Value Methods (BVMs). It has been proved that, when used as BVMs with (k_1, k_2)-*boundary conditions*, such formulae may be $A_{k_1 k_2}$-*stable*, [1] with the boundary loci coinciding with the imaginary axis.

In the current implementation, BVMs are used as composite schemes, by introducing appropriate *additional methods*. In this way, the problem of providing the additional conditions for the k-step formula is avoided. Moreover, in order to gain efficiency, a block implementation of such methods has been considered. In more details, given the integration interval $[t_0, T]$, this is discretized by using two meshes:

- a coarser one, $\tau_i = t_0 + i\hat{h}_i$, $i = 1, \ldots, m$, and
- a finer one, $t_{ji} = \tau_{i-1} + jh_i, \quad j = 1, \ldots, s, \qquad h_i = \hat{h}_i/m.$

Then, the considered BVM is used with *internal stepsize* h_i over the subinterval $[\tau_{i-1}, \tau_i]$. In this way, they are similar to Runge-Kutta methods where the additional methods play a role similar to the stages. However, they differ substantially from Runge-Kutta schemes. For example, on each point of the finer mesh, the solution provided by the BVM has the same accuracy.

The above strategy allows the efficient implementation of BVMs on parallel computers. In fact, by assigning each subinterval in the coarse mesh to a parallel processor, one may obtain almost perfect speed-up, even on small problems.

The codes use an innovative mesh selection strategy, not based on the estimates of the local errors. To be more specific without being too long, we shall refer to the linear case. The *mesh vector* $\mathbf{h} = (\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_m)$ is chosen such that the continuous quantities

$$\gamma_c = \frac{1}{T - t_0} \int_{t_0}^{T} \|\Phi(t, t_0)\| dt, \qquad \kappa_c = \max_{t_0 \leq t \leq T} \|\Phi(t, t_0)\|,$$

and their discrete approximations $\gamma_d(\mathbf{h})$ and $\kappa_d(\mathbf{h})$ are as close as possible. In the previous expression, $\Phi(t, t_0)$ is the fundamental matrix associated with the given ODE, and $[t_0, T]$ is the integration interval.

We describe shortly the expected theoretical performances on each of the above mentioned class of problems, as well as the obtained results based on the experience made so far.

[1] $A_{k_1 k_2}$-stability is the straightforward generalization of the usual concept of A-stability.

Contractive problems

Due to the stability properties of the methods, studied in a series of papers, and to the mentioned mesh selection strategy, the classical contractive initial value problems are easily solved, even though in the non stiff case they may result more costly with respect to the existing codes. We expect to gain efficiency on parallel computers. In the stiff case, however, we have already experimental evidence of the good performance of both codes.

Hamiltonian Problems

So far, the theoretical analysis has been carried out only in the linear case. It has been shown that the used methods are able to maintain the values of the constants of the motion on the coarse mesh for all values of the stepsizes. On the finer mesh, however, such quantities differ from the exact ones by a quantity of the order $O(h^p)$, where p is the order of the method. Considering that for our methods p can easily reach high values, it turns out that, in practice, the constants of motion are *essentially preserved.* Concerning the non linear case, the performances of the codes are as good as in the linear case.

Boundary value problems

Symmetric schemes are particularly suited for approximating continuous boundary value problems. This essentially because they provide solutions having the time reversal symmetry. Moreover, as BVMs, they are the natural candidates to solve such problems, being themselves (discrete) boundary value problems. This permits to avoid the *unnatural* transformation of the problem in a set of initial value problems, as the shooting method does. It is well known, in fact, that well conditioned BVPs give rise to ill conditioned IVPs. The codes have been extensively tested on very difficult traditional problems in this area such as Trench problem, singular perturbation problems, etc. In all cases, the performances are very good.

Differential-Algebraic problems

Although the theoretical analysis has been carried out only for unsymmetric BVMs with very promising results, the experiences made with the code are very satisfactory.

State of the art

Consider the following steps:

1. choice of the methods and study of their essential properties;
2. definition of an appropriate mesh selection strategy;
3. implementation on sequential computer and testing on the classes of problems mentioned above;
4. comparison with existing codes;
5. implementation on parallel computers and evaluation of the performances.

The current state of art is then summarized by the following table.

steps	1	2	3	4	5
% of the (estimated) total time	100	80	80	50	60

Numerical tests

To give an idea of the obtainable performances, we now report some numerical tests on linear problems, obtained by using the same fourth order symmetric scheme. We divide the tests in two parts, each representative of significant situations. In the first one, we report numerical results obtained with the parallel code, running on a network of transputers, with a constant stepsize. In the second part, we report the results obtained by the scalar code, using the above mentioned mesh selection strategy, on singularly perturbed BVPs.

Part 1 Consider the Hamiltonian problem:

$$y' = \begin{pmatrix} & -I_5 \\ I_5 & \end{pmatrix} \begin{pmatrix} 10 & 9 & \dots & 1 \\ 9 & 10 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 9 \\ 1 & \dots & 9 & 10 \end{pmatrix} y, \qquad y(0) = y_0.$$

The speed-ups s obtained on p processors are reported in the following table, evidentiating the scalability of the methods.

p	1	2	4	8	16
s	1.00	1.97	3.90	7.66	15.11

Consider now the following boundary value problem

$$\begin{aligned} \varepsilon u'' &= u, & u(-1) = -y(-1) = 1, \\ \varepsilon y'' &= t + \tfrac{t}{2}u' - \tfrac{t}{2}y' - \varepsilon\pi^2\cos(\pi t) - \tfrac{t}{2}\pi\sin(\pi t), & u(1) = y(1) = e^{-2/\sqrt{\varepsilon}}. \end{aligned}$$

where $\varepsilon = 10^{-3}$. If p is the number of used processors, the problem is approximated with constant stepsize $h = 2/(40p)$. This means that when the stepsize is halved, and consequently the size of the discrete problem is doubled, it is also doubled the number of the used parallel processors. As consequence, we expect the execution time to remain approximately constant, even if the accuracy of the solution improves, due to the use of an halved stepsize. In the following table we report the measured execution times, expressed in units of time (*ticks*), each corresponding to $64\mu sec$.

p	1	2	4	8	16
time	4217	3955	4110	4176	4265

It is remarkable to observe that the execution times on multiple processors are often smaller than the execution times on 1 processor. This is due to the fact that, as the stepsize is decreased (i.e. p increases), the discrete problem changes, and the LU factorization algorithm (the used linear solver) requires less permutations for pivoting.

Part 2 We now consider the application of the same symmetric scheme, but with variable stepsize, to the following two ill-conditioned singular perturbation BVPs ($\varepsilon = 10^{-4}$):

1. $\begin{array}{ll} \varepsilon y'' = 2ty', & -1 < t < 1, \\ y(-1) = 1, & y(1) = 2, \end{array}$

2. $\begin{array}{ll} \varepsilon y'' = ty' - y, & -1 < t < 1, \\ y(-1) = 1, & y(1) = 2. \end{array}$

In both cases, the method starts from a uniform mesh. The final solution is obtained by using the new mesh selection strategy previously mentioned. Next table summarizes the results.

problem	points in the final mesh	estimated error
1	360	$2 \cdot 10^{-5}$
2	480	$7 \cdot 10^{-5}$

Observe that most of the known solvers for BVPs fail to obtain the correct solution for both problems.

References

1. P. Amodio, L. Brugnano. Parallel Implementation of Block Boundary Value Methods for ODEs, *(submitted)*.

2. P. Amodio, W.L. Golik, F. Mazzia. Variable Step Boundary Value Methods Based on Reverse Adams Schemes and their Grid Redistribution, *Appl. Num. Math. (to appear)*.

3. P. Amodio, F. Mazzia. Boundary Value Methods for the Solution of Differential-Algebraic Equations, *Numer. Math.* **66** (1994) 411–421.

4. A.O.H. Axelsson, J.G. Verwer. Boundary Value Techniques for Initial Value Problems in Ordinary Differential Equations, *Math. Comp.* **45** (1985) 153–171.

5. L. Brugnano. Symplectic Boundary Value Methods for Linear Hamiltonian Systems, *(submitted).*

6. L. Brugnano, D. Trigiante. Convergence and Stability of Boundary Value Methods for Ordinary Differential Equations, *J. CAM (to appear).*

7. L. Brugnano, D. Trigiante. Block Boundary Value Methods for Linear Hamiltonian Systems, *(submitted).*

8. L. Brugnano, D. Trigiante. High Order Multistep Methods for Boundary Value Problems, *Appl. Numer. Math. (to appear).*

9. L. Brugnano, D. Trigiante. A New Mesh Selection Strategy for ODEs, *(submitted).*

10. P. Marzulli, D. Trigiante. Stability and Convergence of Boundary Value Methods for Solving ODE, *Jour. of Difference Eq. and Appl.* **1** (1995) 45–55.

5.5 Conservative integrators for the Toda lattice equations

M.P. Calvo
Departamento de Matemática Aplicada y Computación, Universidad de Valladolid, Prado de la Magdalena s/n, 47005 Valladolid, Spain
E-mail: maripaz@mac.cie.uva.es
A. Iserles
Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW, England
E-mail: ai@amtp.cam.ac.uk
A. Zanna
Newnham College, University of Cambridge, Silver Street, Cambridge CB3 9EW, England
E-mail: A.Zanna@amtp.cam.ac.uk

In the last years there has been a growing interest in studying methods for the numerical integration of systems of ordinary differential equations that preserve some characteristic features of the exact flow of the system. An instance of such methods are the well-known *symplectic* integrators [7] which preserve the symplectic structure of phase space for Hamiltonian systems. A second example, that will be considered in this talk, are *isospectral* methods [1]. Both special integrators become relevant in the numerical integration of the Toda lattice equations. In the original variables this Hamiltonian problem reads as

$$\begin{aligned}
\frac{d}{dt}q_i &= p_i, \qquad 1 \le i \le d\\
\frac{d}{dt}p_i &= 2\left(\exp\left(2(q_{i-1}-q_i)\right) - \exp\left(2(q_i - q_{i+1})\right)\right), \qquad 2 \le i \le d-1,\\
\frac{d}{dt}p_1 &= -2\exp\left(2(q_1-q_2)\right), \qquad \frac{d}{dt}p_d = 2\exp\left(2(q_{d-1}-q_d)\right).
\end{aligned}$$

This differential system models the motion of a finite number of particles on the line with exponential interaction between nearest neighbours. Symplectic integrators have revealed themselves very efficient for the numerical integration of Hamiltonian systems [7]. On the other hand, after introducing the new variables [5]

$$\begin{aligned}
a_i &= -p_i, \qquad 1 \le i \le d,\\
b_i &= \exp\left(q_i - q_{i+1}\right), \qquad 1 \le i \le d-1,
\end{aligned}$$

the differential system may be rewritten as

$$\frac{d}{dt}L(t) = \left[B(L(t)), L(t)\right], \quad t \ge 0 \tag{1}$$

where $L(t)$ is the symmetric tridiagonal matrix with elements $L_{i,i}(t) = a_i(t)$, $1 \le i \le d$ and $L_{i,i+1}(t) = L_{i+1,i}(t) = b_i(t)$, $1 \le i \le d-1$, $B(L) = L_+ - L_-$ is the difference between the upper and lower triangular parts of the matrix L and $[\cdot,\cdot]$ denotes the commutator of two matrices. This corresponds to the Lax representation of the Toda Lattice equations. It has been shown that the problem (1) is equivalent to solving

$$\begin{aligned}
\frac{d}{dt}U(t) &= B(L(t))U(t), \quad t \ge 0\\
U(0) &= Id
\end{aligned} \tag{2}$$

in tandem with

$$L(t) = U(t)L(0)(U(t))^T, \quad t \ge 0. \tag{3}$$

As $B(L(t))$ is skew-symmetric, the solution $U(t)$ of (2) is an orthogonal matrix and relation (3) implies that $L(t)$ is orthogonally similar to $L(0)$. This means that there is a set of d integrals of motion [8] which are the eigenvalues of the symmetric matrix $L(0)$ (i.e. the flow of (1) is isospectral). Isospectral methods are numerical integrators for (1) which preserve the eigenvalues of the solution. In [1], [2] the authors show that isospectrality is lost when numerical integration of (1) is performed with a Runge-Kutta method if $d > 2$. They also propose a general approach which allows to produce an isospectral solution for (1) with arbitrarily high order of accuracy. The basic idea is to integrate (2) with a *unitary* integrator (Gauss methods have this property [4]), subsequently using the discrete counterpart of (3) in order to get a numerical approximation to the solution of (1). The schemes obtained in this way are implicit.

In the talk new isospectral methods will be derived, based on Flaschka's formalism but using more general integrators for (2). Instead of Gauss methods for the numerical integration of (2) we propose Taylor series methods. In this way semi-explicit isospectral integrators are obtained. Furthermore, these new methods are also valid for the numerical integration of differential systems like (1) where $L(0)$ is not necessarily symmetric and, as a consequence, $U(t)$ is not orthogonal. The exact flow for this more general system is isospectral, (2) is still valid but (3) must be replaced by

$$L(t) = U(t)L(0)(U(t))^{-1}, \quad t \geq 0.$$

A numerical comparison among different isospectral methods and standard ODE integrators will be reported for the Toda lattice equations. We pay special attention to the error growth with time for these methods. It is found that for isospectral schemes the error decays exponentially with time but for conventional integrators the error is almost constant with time. These experimental findings will be explained by theoretical analysis.

Moser in [6] has shown that as t goes to $+\infty$, the flow $L(t)$ tends to a diagonal matrix whose elements are the eigenvalues of $L(0)$ while Deift, Nanda and Tomei in [3] have shown that the convergence to the asymptotically stable equilibrium point is exponential. We prove that, under certain assumptions on the numerical method, the same property holds for the numerical solution. This means that the sequence $\{L_n\}_{n=0}^{+\infty}$ given by the numerical approximation to the solution of (1) converges to a diagonal matrix L_∞. Only for isospectral methods the elements of L_∞ are exactly the eigenvalues of $L(0)$. This analysis explains the different behaviour of the error for the numerical methods considered in the numerical experiments.

References

[1] M.P. Calvo, A. Iserles & A. Zanna, *Numerical solution of isospectral flows*, DAMTP Tech. Rep. 1995/NA03.

[2] M.P. Calvo, A. Iserles & A. Zanna, *Runge-Kutta methods on manifolds*, to appear in *Numerical Analysis: A.R. Mitchell 75th Birthday Volume* (G.A. Watson & D.F. Griffiths, eds.), World Scientific, Singapore.

[3] P. Deift, T. Nanda & C. Tomei, *Ordinary differential equations and the symmetric eigenvalue problem*, SIAM J. Numer. Anal. 20 (1983), 1–22.

[4] L. Dieci, R.D. Russell & E.S. van Vleck, *Unitary integrators and applications to continuous orthonormalization techniques*, SIAM J. Numer. Anal. 31 (1994), 261–281.

[5] H. Flaschka, *The Toda lattice*, Phys. Rev. B 9 (1974), 1924–1925.

[6] J. Moser, *Finitely many mass points on the line under the influence of an exponential potential – An integrable system*, in *Dinamical Sistems Theory and Applications*, (J. Moser, ed.) Springer-Verlag (1975), 467–497.

[7] J.M. Sanz-Serna & M.P. Calvo, *Numerical Hamiltonian problems*, Chapman & Hall, London, 1994.

[8] M. Toda, *Theory of Nonlinear Lattices*, Springer-Verlag, Berlin, Heidelberg, New York, 1981.

5.6 Innovative Time Integrators

J.R. Cash
Department of Mathematics, Imperial College, London SW7 2BZ, United Kingdom
E-mail: j.cash@ic.ac.uk

A very popular way of numerically solving time dependent partial differential equations of the form

$$c(x,t)\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} r\left(x,t,u,\frac{\partial u}{\partial x}\right) + f\left(x,t,u,\frac{\partial u}{\partial x}\right) \tag{1}$$

$$(x,t) \in \Omega \equiv [a,b] \times [0,T]$$

is by the numerical method of lines [MOL]. To avoid any problems with singular behaviour we assume that there exist constants c_1, c_2 such that

$$0 < c_1 < c(x,t) < c_2, \qquad \forall (x,t) \in \Omega. \tag{2}$$

We will further assume that the boundary conditions are given in the form

$$g_a(t, u(a,t)) = g_b(t, u(b,t)) = 0 \tag{3}$$

and that the initial condition is given as

$$u(x,0) = u_0(x), \qquad x \in [a,b]. \tag{4}$$

The extension of the MOL approach to systems of PDEs, or to problems with Neumann or mixed boundary conditions, is straightforward [15]. The MOL approach is an extremely powerful and versatile one which has been very widely used in general purpose software for the solution of (1)–(4) and in what follows we will briefly describe the general ideas behind it.

In the method of lines for the solution of (1), the space domain is first divided into a finite, discrete mesh. The spatial derivatives appearing in (1) are then approximated at each point in the mesh, typically by the use of finite differences, spectral methods or finite element techniques. This results in a system of initial value problems taking one of the two forms

$$\frac{du}{dt} = f(t, u(t)), \qquad u(0) = u_0 \tag{5}$$

or

$$A(t,u)\frac{du}{dt} = g(t, u(t)), \qquad u(0) = u_0. \tag{6}$$

Systems of the form (5) typically occur when finite differencing of finite elements with mass matrix lumping is used to discretize the space variables and we will be solely concerned with systems of this form for the present.

Typical MOL software consists of two distinct components, one which discretizes the space derivatives to produce (5) and the other which integrates (5) in time to give $u(t)$. In the earliest MOL software the user was typically required to specify a spatial mesh at the start of the integration and normally this mesh was held constant for all time. More recent work has focussed on the balancing of temporal and spatial errors [2,5,16,17] and some of the more modern packages, such as SPRINT [4], attempt to modify the spatial mesh dynamically as the time integration proceeds.

Very often the system of equations (5) arising from the use of MOL is stiff and, as a result, the BDF are one of the most widely used classes of formulae for integrating (5). Often the form of the boundary conditions gives rise to a system of differential/algebraic equations and in this case a suitable modification of LSODE such as LSODI[11] or DASSL[18] should be used instead.

Recently some competitors to BDF for the numerical solution of stiff initial value problems have arisen. In particular, results presented for Modified Extended Backward Differentiation Formulae, MEBDF,

[1] have shown that they are competitive with BDF over a large class of problems. In particular MEBDF are generally much superior to BDF on problems of the form (5) where some eigenvalues of the Jacobian matrix $\frac{\partial f}{\partial u}$ are complex, large in modulus and lie close to the imaginary axis. Examples of such problems for which BDF are notoriously inefficient, are advection-diffusion problems where advection dominates. Some examples of the performance of BDF and MEBDF will be given for a particular problem of this class namely the chemical flooding problem [1]. The other major classes of problems where MEBDF have performed very well compared with BDF is when high accuracy is required (MEBDF use formulae up to order 9 while BDF use order up to order 5) and for problems where function or Jacobian evaluations are very expensive compared with other operations. Some numerical examples of this will be given for heat conduction type problems.

A major advantage of MEBDF methods, and the reason for their marked superiority over BDF methods for advection dominated problems, are that they are A-stable up to order 4 while BDF methods are A-stable up to order 2. This is achieved by using one super future point. A natural extension to this approach is to use two super future points. This has been done and algorithms which are A-stable up to order 6 have been derived. (Conjecture: by using k super future points we can get A-stability up to order $2(k+1)$). It should be a simple matter to incorporate these new algorithms into existing MEBDF software and a discussion of their potential for MOL will be given.

[1] Anonymous FTP: to cato.ma.ic.ac.uk (155.198.192.82). Codes are in /pub/jeff and the user should first get readme.
WWW:http://www.ma.ic.ac.uk/-baxter/jeff/readme.html

5.7 Multirate extrapolation methods for differential equations with different time scales

Ch. Engstler and Ch. Lubich
Universität Tübingen, Mathematisches Institut, Auf der Morgenstelle 10, 72076 Tübingen, Germany
E-mail: lubich@na.uni-tuebingen.de

Introduction

When integrating systems of differential equations whose components evolve at different time scales, one would like to use numerical methods that do not expend unnecessary numerical work on slow solution components. The first article to treat such problems appears to be a paper by Rice (1960), who proposes 'split' Runge-Kutta methods (which would nowadays be called 'multirate' Runge-Kutta methods) that use different time steps to integrate fast and slow solution components. Such an approach has been developped further by Gear & Wells (1984), Günther & Rentrop (1993), and Skelboe & Andersen (1989). In a closely related approach, known as 'multiple time stepping', one does not split solution components, but instead the right-hand side function as a sum of fast and slowly changing functions which are then evaluated with different frequency.

A difficulty with the existing multirate techniques is that they assume a clear-cut partition of the system into fast and slow components, which has to be known before performing a macro-step. A 'slowest first' or 'fastest first' strategy has to be employed, whose success depends on the coupling between components.

In the present paper, we propose a multirate method based on Richardson extrapolation. The basic idea is to stop building up the extrapolation tableau for components that have been recognized to be already sufficiently accurate. Since local error estimates are available at all extrapolation levels, this provides an inexpensive strategy for dynamic partitioning into several classes of slow to fast components. In contrast to previous multirate methods, slow and fast components are integrated simultaneously over a macro-step. However, slow components are inactivated at early extrapolation levels, and the refinement of the solution by high-order Richardson extrapolation focusses on ever faster components. The method proposed here has been implemented in MURX, a multirate extrapolation code written in Fortran.

We begin by recalling classical Richardson extrapolation. Multirating for the extrapolated Euler scheme is described, and also a defect control mechanism to verify the inactivation strategy. A simple extension of the techniques to the situation of 'multiple time stepping' for split right-hand sides is given. Finally, we present numerical results of MURX applied to an example from astrophysics.

Classical extrapolation

When the initial value problem

$$y' = f(t,y) \ , \qquad y(t_0) = y_0$$

is discretized by the explicit Euler method with step size h, this gives at $t_{n+1} = t_n + h$ a first-order approximation to $y(t_{n+1})$ by

$$y_{n+1} = y_n + hf(t_n, y_n) \ , \qquad n \geq 0 \ ,$$

which we denote

$$y(t,h) = y_n \quad \text{for} \ \ t = t_0 + nh \ .$$

Given a macro-step size H, one constructs Euler approximations to $y(t_0 + H)$ with step sizes $h_j = H/n_j$, where $\{n_j\}$ is the step number sequence. With these values one computes the extrapolation tableau

$$\begin{array}{llll} T_{11} & & & \\ T_{21} & T_{22} & & \\ T_{31} & T_{32} & T_{33} & \\ \vdots & \vdots & \vdots & \ddots \end{array}$$

An important property of the extrapolation tableau is that numerical error estimators are available at all levels: The subdiagonal difference $T_{k,k} - T_{k,k-1}$ is taken as an approximation to the error at level k. This is used for step size and order selection. Another very useful option of extrapolation methods is their continuous solution approximation.

Multirate extrapolation

We will see that multirating is easy and natural within extrapolation methods. Let us assume that we have already computed the entries T_{11}, T_{21}, T_{22} of the extrapolation tableau. We use the following simple strategy: If the ith component of the error estimator is below the given tolerance,

$$|T_{22}^i - T_{21}^i| < 0.1 \cdot \mathrm{tol} \ , \tag{1}$$

then we accept T_{22}^i as solution approximation,

$$T_{22}^i \approx y^i(t_0 + H) \ ,$$

and make component i *inactive*, meaning that we are not willing to make any further evaluations of f^i, the ith component of the right-hand side function. To reduce the risk of accidental cancellation of error terms in $T_{22}^i - T_{21}^i$, one might check (1) for small groups of components rather than single components.

At the next level of extrapolation, we then have to compute T_{31} for the remaining (=*active*) components. We split the solution vector and the function into their active and inactive components,

$$y = \begin{pmatrix} y^A \\ y^I \end{pmatrix} \ , \qquad f = \begin{pmatrix} f^A \\ f^I \end{pmatrix} \ ,$$

where f^I is not to be evaluated any more. To compute T_{31}^A, we would have to perform Euler steps with $h = H/3$:

$$\begin{array}{rl} y_1^A & = y_0^A + hf^A(t_0, y_0) \\ y_2^A & = y_1^A + hf^A(t_1, y_1^A, \underline{y_1^I}) \\ T_{31}^A = y_3^A & = y_2^A + hf^A(t_2, y_2^A, \underline{y_2^I}) \ . \end{array} \tag{2}$$

At this stage we have the problem of how to obtain the values y_1^I and y_2^I. There are several possibilities, of which we favour the following:

In (2), the correct values would be Euler approximations, which we now try to approximate without, however, using the inactivated function components f^I. We start from the asymptotic expansion

$$y^I(t,h) = y^I(t) + he_1^I(t) + \dots \ .$$

We have already an accurate approximation $P_2^I(t)$ to $y^I(t)$. To approximate $e_1^I(t)$, we note that up to terms of size $C^I \cdot H^3$ with a constant C^I of the same type as previously, we have for $h = H/2$ and $t = t_0 + H$ by (2.6)

$$T_{21}^I \approx T_{22}^I + H/2 \cdot e_1^I(t_0 + H) \ ,$$

and for $h = H/2$ and $t = t_0 + H/2$

$$y^I(t_0 + H/2, H/2) \approx P_2^I(t_0 + H/2) + H/2 \cdot e_1^I(t_0 + H/2) \ .$$

We now replace $e_1^I(t)$ by the quadratic polynomial $q_2^I(t)$ which vanishes at t_0 and for which the above two relations become an equality:

$$q_2^I(t_0 + \theta H) = (4a - b)\theta + (2b - 4a)\theta^2 \approx e_1^I(t_0 + \theta H) \ ,$$

where $a = 2/H \cdot \left(y^I(t_0 + H/2, H/2) - P_2^I(t_0 + H/2)\right)$ and $b = 2/H \cdot \left(T_{21}^I - T_{22}^I\right)$. By construction we have

$$|q_2^I(t) - e_1^I(t)| \leq C^I \cdot H^3 \ ,$$

where C^I is nearly a constant of the previously encountered type, except that it now depends also on derivatives of $e_1^I(t)$. (Again, only the inactive – or slow – components enter into C^I.) This gives us the approximated Euler steps

$$P_2^I(t) + hq_2^I(t) \approx y^I(t,h) \ , \tag{3}$$

which are to be used with $h = H/3$ and $t - t_0 = H/3$, $2H/3$ in (3.2), and with $h = H/4, H/5, \ldots$ at higher levels of the extrapolation tableau. Provided that (3) is sufficiently accurate, we work with the correct asymptotic expansions in continuing the extrapolation tableau.

This approach can be extended to higher levels, which will be described in the talk. Numerical experiments with the code MURX (MUlti-Rate eXtrapolation) will be presented for a problem from astrophysics.

5.8 Alternating Direction Implicit Orthogonal Spline Collocation Methods for Initial/Boundary Value Problems

G. Fairweather
Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO 80401-1887, U.S.A.
E-mail: gfairwea@mines.colorado.edu
B. Bialecki
Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO 80401-1887, U.S.A.
R. I. Fernandes
Department of Mathematics and Statistics, Mississippi State University, Mississippi 39762, U.S.A.
B. Li
Department of Mathematics, University of Kentucky, Lexington, Kentucky 40506-0027, U.S.A.

(Supported by the U.S. Army Research Office grant DAAH04-G-0344.)

We discuss the formulation, analysis and implementation of new efficient algorithms for the solution of linear initial/boundary value problems in two space variables on rectangles. The basic approach is to discretize in space using orthogonal spline collocation with C^1 piecewise polynomials of degree ≥ 3, and advance in time using second order accurate alternating direction implicit methods.

Orthogonal spline collocation (OSC) methods, that is, methods based on spline collocation at Gauss points, have proved to be exceedingly effective for the approximate solution of a broad class of problems. Like many finite element Galerkin methods, OSC provides approximations to the solution and its spatial derivatives at all points of the domain of the problem and allows for arbitrarily high order accuracy in the spatial discretization. In addition, it has been observed experimentally that the basic OSC approach yields superconvergent approximations. The obvious advantage of OSC methods over finite element Galerkin methods is that the calculation of the coefficients in the linear system of equations determining the approximate solution is very fast since no integrals need be evaluated nor approximated.

Alternating direction implicit (ADI) techniques, introduced in the context of finite differences for canonical parabolic and elliptic problems by Peaceman and Rachford (1955), have been used extensively over the last four decades in the numerical solution of time-dependent problems. ADI finite element Galerkin methods for parabolic and hyperbolic problems in two space variables were first formulated and analyzed by Douglas and Dupont (1971). For problems with variable coefficients, their methods were based on a "Laplace-modified" approach in which there appears a parameter which must be chosen appropriately for stability. ADI techniques are of much current interest because they reduce the solution of a multi-dimensional problem to a set of independent one-dimensional problems and are thus well-suited for parallel computation.

By mimicking ADI Galerkin methods, various authors formulated and implemented ADI OSC schemes for parabolic problems to solve some practical problems. However, no rigorous convergence analyses of such schemes were derived until Fernandes and Fairweather (1993) examined ADI OSC methods for the inhomogeneous heat and wave equations on the unit square. For variable coefficient parabolic equations in which the principal part of the spatial operator is in selfadjoint form, Bialecki and Fernandes (1993) analyzed ADI OSC Laplace modified schemes, and also an ADI OSC method, based on a Crank-Nicolson technique, which has no finite element Galerkin counterpart.

In this paper, we overview the development of ADI OSC methods and describe recent progress in their formulation and analysis. In particular, we describe an ADI OSC method for non-selfadjoint parabolic problems and also methods for second order hyperbolic problems, again techniques with no finite element Galerkin counterparts. An ADI OSC solution of Schrödinger-type problems based on a similar approach is also presented.

With standard choices of basis functions, each of the ADO OSC methods discussed leads to independent one-dimensional problems which require the solution of almost block diagonal linear systems. These systems can be solved effectively using existing software packages such as COLROW (Diaz, Fairweather and Keast, 1983).

5.9 On moving finite difference methods for partial differential equations on two-dimensional domains

J.A. Ferreira
Departamento de Matemática, Universidade de Coimbra, 3000 Coimbra - Portugal
E-mail: ferreira@matuc2.mat.uc.pt

(This work was supported by Junta Nacional de Investigacão Científica e Tecnológica- INVOTAN, CMUC and European project- Humain Capital and Mobility ERBCHRXCT: 930407)

Introduction

For parabolic problems defined on two dimensional domains with solutions presenting moving large local gradients the use of a fixed spatial grid fixed throughout entire calculation of a numerical approximation can be computationally inefficient. The use of moving grid methods enable us to obtain, with an acceptable degree of accurate, a numerical approximation to the solution of such partial differential problems while avoiding the excessive number of grid points.

For problems defined on two-dimensional domains, there are basically two different approaches on moving grid methods. While on the first one the choice of the nonuniform grid points depends one some function of the solutions and the grid points movement continuously in time, on the second one the grid is locally uniform and these grid points are frequently replaced at least at local level. We remark that the first approach is followed frequently on unidimensional problems, as for instance [2]-[5], [9] but also for two-dimensional problems (in [11]), the second approach is often considered for time dependent or time independent two- dimensional partial differential problems as for instance in [1], [8], [10].

The subject of this paper belongs to the field of moving grid methods that follow the mentioned first approach. Following the procedure introduced in [6] and considering a moving rectangular grid, we present a theoretical convergence study for the semi-discrete approximation to the solution of a class of parabolic two-dimensional problems. This semi-discrete approximation is obtained considering a finite difference spatial discretization and some smoothness conditions about the time evolution of the grid points.

The theoretical results are established for a general time dependent finite difference operator that discretize the Lagrangian formulation of the parabolic problem

$$\left\{\begin{array}{rcll} \dfrac{\partial u}{\partial t}(x,t) + Lu(x,t) & = & f(x,t), & (x,t) \in \Omega \times (0,T], \\ u(x,0) & = & u_0(x), & x \in \overline{\Omega}, \\ \mathcal{B}u(x,t) & = & g(x,t), & (x,t) \in \Gamma \times [0,T], \end{array}\right. \tag{1}$$

where L is the spatial differential operator defined by

$$Lu(x,t) = \sum_{\ell=1}^{2} \frac{\partial}{\partial x_\ell}(-a_{\ell\ell}(x)\frac{\partial u}{\partial x_\ell}(x,t)) + a_\ell(x)\frac{\partial u}{\partial x_\ell}(x,t) + a_0(x)u(x,t),$$

$\Omega = (0,a) \times (0,b)$ and $\mathcal{B}u(x,t)$ defines the boundary conditions.
Being $x_{i,j}(t) = (x_{1,i}(t), x_{2,j}(t)), i = 1, ..., N, j = 1, ..., M$, the grid points and

$$h_i(t) = x_{1,i}(t) - x_{1,i-1}(t),\ k_j(t) = x_{2,j}(t) - x_{2,j-1}(t),$$

we suppose that

$$\frac{dh_i}{dt}(t) = C_{i,\ell}(t)h_i(t),\ {\scriptstyle i=1(1)N},\ t \in (0,T], \quad \text{for} \quad \text{some } \ell, \tag{2}$$

$$C_{i+1,\ell}(t) = C_{i,\ell}(t) + \epsilon_{i,\ell}(t),\ \epsilon_{i,\ell}(t) = O(h_{max}(t)),\ {\scriptstyle i=1(1)N-1}, \tag{3}$$

$$\frac{dk_j}{dt}(t) = G_{\imath,j}(t)k_j(t), \;\; {\scriptstyle j=1(1)M}, \; t \in (0,T], \quad \text{for} \quad \text{some} \;\; \imath, \tag{4}$$

$$G_{\imath,j+1}(t) = G_{\imath,j}(t) + \beta_{\imath,j}(t), \; \beta_{\imath,j}(t) = O(k_{max}(t)), \;\; {\scriptstyle j=1(1)M-1}, \tag{5}$$

where $C_{i,\ell}$, $G_{\imath,j}$ are such that $\max\{|C_{i,\ell}|, \, |G_{\imath,j}|\} \leq C_M$ for all i, ℓ and $\imath, j$.

From the assumptions (2) and (4) and using the energy method, we prove, in section 2, for the semi-discrete approximation defined by a general class of finite difference operators two convergence results. In the first one it is established an estimate to the discrete Sobolev norm of the global spatial discretization error using the norm of the truncation error. This estimate is improved on the second theorem proving that the discrete Sobolev norm of the global spatial discretization error is bounded by the correspondent norm of the global error of the discretization of the stationary problem associated with (1) and its evolution in time. The results obtained in section 2 are specified in section 3 to a discretization of (1) considering Dirichlet boundary conditions. In this section using the results obtained in [7] and the conditions (3), (5), we establish that the discrete Sobolev norm of the global semi-discretization error is of second order while the correspondent truncation error presents only first order.

In section 5 we analyse the relation between the theoretical assumptions considered in section 2 and which allowed to the convergence results for the semi-discrete approximation and a practical criterion. We introduce an extension of the equidistribution principle followed on the unidimensional to the two-dimensional case - $\Omega = (0,1) \times (0,1)$. We consider for the movement grid points the following equation

$$\int_{x_{1,i}(t)}^{x_{1,i+1}(t)} \int_{x_{2,j}(t)}^{x_{2,j+1}(t)} M(\sigma,\tau,t) d\sigma d\tau = \frac{1}{NM} \int_0^1 \int_0^1 M(\sigma,\tau,t) d\sigma d\tau, \tag{6}$$

for all i and j, where the function M depends on the derivatives of the solution. Assuming some smoothness on the monitor function M we prove that the conditions on the evolution in time of the grid points are consequence of (6) and so, the convergence results, previously established in the section 2, hold.

Considering M as an extension of the arclenght and the curvature, that is, M given by

$$M(x_1,x_2,t) = \sqrt{1 + (\frac{\partial u}{\partial x_1})^2 + (\frac{\partial u}{\partial x_2})^2}, \tag{7}$$

and

$$M(x_1,x_2,t) = \sqrt{1 + (\frac{\partial^2 u}{\partial x_1^2})^2 + (\frac{\partial^2 u}{\partial x_2^2})^2}, \tag{8}$$

we analyse the performance of the criterion induced by (6) looking, on several examples, to the behaviour of the grid points.

References

[1] D.C. Arney, J.E. Flaherty, 1989, An adaptive local mesh refinement method for time-dependent partial differential equations, Appl. Num. Math. 5, 257-274.

[2] M. Bieterman and I. Babuska, 1986, An adaptive method of lines with error control for parabolic equations of the reaction-diffusion type, J. Comp. Phys. 63, pp.33-66.

[3] J.G. Blom, J.M. Sanz-Serna, J.G. Verwer, 1988, On simple moving grid methods for one dimensional partial differential equations, J. Comp. Phys. 74, pp.191-213.

[4] J.G. Blom, J.M. Sanz-Serna, J.G. Verwer, 1989, A Lagrangian moving grid scheme for one-dimensional evolutionary partial differential equations, IMACS Annals on Computing and Applied Mathematics, Vol.1.1, ed. W. Ames, pp. 247-255.

[5] J.G. Blom, J.G. Verwer, 1987, On the use of arclength and curvature monitor in a moving grid method which is based on the method of lines, Note NM-N8902, Centre for Mathematics and Computer Science (CWI), Amsterdam.

[6] J.A. Ferreira, 1993, The second order initial/boundary value problem: supraconvergence of regridding methods, Appl. Num. Math. 13, 69-81.

[7] J.A. Ferreira and R.D. Grigorieff, On the supraconvergence of finite difference discretizations of stationary partial differental problems with Neumann or Dirichlet boundary contitions, submitted.

[8] R.A. Trompert, J.G. Verwer, 1991, A static regridding method for two-dimensional parabolic partial differential equations, Appl. Num. Math. 8, 65-90.

[9] J.G. Verwer, J.G. Blom, J.M. Sanz-Serna, 1989, An adaptive moving grid method for one-dimensional system of partial differential equations, J.Comp.Phys. 82, pp. 454-486.

[10] J.G. Verwer, R.A. Trompert, 1993, Analysis of local uniform grid refinement, Appl. Num. Math. 13.

[11] P.A. Zegeling, 1992, Moving finite element solution of time dependent partial differential equations in two space dimensions, Report NM-R9206, Centre for Mathematics and Comuter Science (CWI), Amsterdam.

5.10 Adaptive and Parallel Computational Techniques with Applications in Materials Science

Joseph E. Flaherty
Department of Computer Science and Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, New York 12180-3590, U.S.A.
E-mail: flaherje@cs.rpi.edu

Introduction

Adaptive computational techniques have advanced to the point where they are having a dramatic impact on the way that scientists and engineers solve problems involving partial differential equations. Progress and use will accelerate as problems become more complex and the need to automate a greater portion of the solution process increases. Specific adaptive strategies enrich preliminary solutions obtained on coarse meshes using combinations of mesh refinement and coarsening (h-refinement), variation of method order (p-refinement), and mesh motion (r-refinement). A posteriori estimates of discretization errors control adaptive enrichment and provide an important measure of solution reliability. Without such error estimates, computations must either be repeated with, e.g., finer meshes or solution accuracy must be inferred using independent physical knowledge of the anticipated solution. The former approach is inefficient and impractical with truly demanding problems while the latter approach provides qualitative information at best and is becoming less-and-less possible with the complexities of modern engineering systems. Error estimates obtained by order embedding (p-refinement) are asymptotically correct and have expenses ranging from one-third to one-half of the solution cost.

Like adaptivity, parallel computation is being used to an increasing degree as multi-dimensional, nonlinear, and other realistic features are included in mathematical models. However, adaptivity and parallelism are at odds. The most efficient adaptive procedures utilize hp-refinement to achieve methods having very high convergence rates. Optimality occurs through complex logic and heuristics that are difficult to parallelize.

We are developing parallel adaptive techniques to automate the solution of problems involving partial differential equations. The computational approach emphasizes high-order adaptive techniques with spatial and temporal hp- and hpr-refinement. Finite element meshes are automatically generated, refined, and coarsened. With an emphasis on three-dimensional transient systems, computational research is aiming at developing (i) adaptive procedures for massively-parallel distributed-memory systems; (ii) strategies for balancing spatial and temporal errors; (iii) optimal enrichment and (iv) a posteriori error estimation techniques for specific applications; and (v) software tools to help scientists and engineers address related problems.

Although the software system has general applicability, we have been working with materials scientists and engineers to model techniques for manufacturing ceramic composites, coating ceramic fibers, and fabricating indium-phosphide crystals. These and similar materials-processing problems offer several challenges that make adaptive software a desirable if not essential solution tool. Chemical reactions occur in narrow layers that propagate as material is transformed; pores and voids between solid particles fill as reactions progress; phase transitions are present; the geometry evolves; strong convective currents are present; and complex surface and bulk reactions are involved.

Computational Framework

We have developed a framework for adaptive and parallel computation that is capable of (i) generating three-dimensional unstructured meshes of tetrahedral elements, (ii) automatically refining and coarsening these meshes, (iii) partitioning the computation into subdomains that may be processed in parallel, and (iv) maintaining a balanced parallel computation. Mesh generation, done in collaboration with Mark Shephard, is supported by a hierarchical database in which spatial regions are linked to their bounding faces, which, in turn, are linked to their bounding edges, which are linked to their endpoints. The database is connected to geometrical modeling procedures and to a parallel library having capabilities for processor scheduling and reassignment. The data structures used in the library provide fast query and update of information on the boundaries between subdomains (partitions) belonging to different processors. Mesh refinement and coarsening are edge based with coarsening done by deleting all elements connected to one end vertex and connecting the faces of the polyhedral cavity to the other end vertex. Shape optimization, needed to prevent element quality degradation, is currently done by edge swapping procedures within element faces. These techniques require no history of previous mesh modifications and are, thus, fast and space efficient.

Adaptive enrichment introduces an imbalance in a balanced parallel computation and, thus, a need for dynamic load balancing. Existing partitioning and balancing procedures for static-mesh computation are too expensive for use with adaptive methods because they (i) have high unit costs, (ii) require a global re-partitioning, and (iii) do not account for the locality of adaptive enrichment. Migration, a strategy for the local exchange of information between neighboring processors overcomes these difficulties and is much better suited to an adaptive environment. Our migration techniques use linked structures on partition boundaries for the efficient exchange of elemental information. While fast, these procedures are only moderately effective at controling communications volume. Superior volume control may be obtained by using a strategy where the principal axes of inertia of the communications volume indicate partitions having large surface areas relative to their volumes. A second strategy uses the underlying octree of the mesh generation procedure to compute parallel partitions. All techniques execute in parallel, use the same migration software to interchange data, and support h-, p-, and p-refinement.

Temporal integration utilizes a method-of-lines formulation with singly implicit Runge-Kutta (SIRK) methods. Temporal error estimates of all Runge Kutta stages may be computed for the cost of an additional function evaluation. Temporal errors are coupled to spatial error estimates so as to control global errors and maintain a balance between time and space errors. The SIRK procedures execute in parallel within the adaptive parallel framework. Spatial variation of the SIRK order is supported; however, time step variation, at the moment, is global.

The solution of the resulting linear algebraic equations is accomplished by iterative techniques using generalized and quasi-minimal residual methods. Multigrid methods are used to construct preconditionings. In a spirit similar to algebraic multigrid techniques, coarser grids are constructed from the hierarchical data structure without need of explicit grid generation.
This software is being used to address problems in materials science involving chemical vapor deposition and infiltration as well as crystal growth.

Recent Publications and Preprints

1. C. Ozturan, H.L. deCougny, M.S. Shephard, and J.E. Flaherty, "Parallel Adaptive Mesh Refinement and Redistribution on Distributed Memory Computers," *Comp. Meths. Appl. Mech. Engng.*, **119** (1994), 123-137.

2. S. Adjerid, J.E. Flaherty, M. Shephard, Y. Wang, J. Hudson, W. Hillig, and N. Patibandla, "Adaptive Numerical Techniques for Reactive Vapor Infiltration," *Ceram. Engng. and Sci. Proc.*, **15** (1994), 924-931.

3. S. Adjerid, M. Aiffa, J.E. Flaherty, "High-Order Finite Element Methods for Singularly-Perturbed Elliptic and Parabolic Problems," *SIAM J. Appl. Math.*, **55** (1995), 520-543.

4. C.L. Bottasso, H.L. DeCougny, M. Dindar, J.E. Flaherty, C. Ozturan, Z. Rusak, and M.S. Shephard, "Compressible Aerodynamics Using a Parallel Adaptive Time-Discontinuous Galerkin Least-Squares Finite Element Method," AIAA Paper 94-1888, 12th AIAA Appl. Aerodyn. Conf., June 20-22, 1994, Colorado Springs.

5. S. Adjerid, J.E. Flaherty, W. Hillig, J. Hudson, and M.S. Shephard, "Modeling and the Adaptive Solution of Reactive Vapor Infiltration Problems," *Model. and Simul. in Mat. Sci. Engng.*, 1995, to appear.

5.11 Postprocessing spectral methods in the time integration of dissipative Partial Differential Equations

Julia Novo, Javier de Frutos
Dpto. de Matemática Aplicada y Computación, Universidad de Valladolid, Prado de la Magdalena s/n, 47005 Valladolid, Spain
E-mail: frutos@mac.cie.uva.es
Bosco García-Archilla
Dpto. de Matemáticas, Universidad Autónoma de Madrid, Madrid, Spain

The time integration of dissipative partial differential equations have received large attention in recent years. In this contribution we introduce a postprocessed spectral-Galerkin method based on Legendre or Chebyshev polynomials. The method is shown to be of higher accuracy than the nonpostprocessed method while the computational cost is kept nearly the same.

Many dissipative equations can be written as a differential equation:

$$\frac{du}{dt} + Au + R(u) = f \tag{1}$$

in a suitable choosen Hilbert space H. Here A is a densily defined,unbounded, nonnegative and selfadjoint, operator with compact inverse, defined in $D(A) \subset H$, and R is a nonlinear operator in H with $D(A) \subset D(R)$. The linear operator A is meant to involve the higher order spatial derivatives, while R gathers nonlinear terms and perhaps lower order spatial derivatives. Nonlinear Galerkin methods [13] are novel discretization methods for (1) that where introduced in connection with inertial manifolds [3].
In an inertial manifold, once the solution of (1) is expanded in terms of the eigenfunctions $\{\varphi_1, \varphi_2, \ldots\}$ of A associated with eigenvalues $\lambda_1 \le \lambda_2 \le \ldots$, the higher modes are given explicitly in terms of the lower modes, that is, if P_M is the orthogonal projection onto $H_M = \mathrm{span}\{\varphi_1, \varphi_2, \ldots, \varphi_M\}$, $q_M = u - P_M u = \Phi(P_M u)$. This allows us to write (1) in the equivalent finite dimensional form:

$$\frac{dp}{dt} + Ap + P_M R(p+q) = P_M f, \quad p \in H_M, \quad q = \Phi(p). \tag{2}$$

Since one does not usually know the exact form of the function Φ, the nonlinear Galerkin method replace it by some approximation Φ_{app}, so that the solution is found by numerically integrating the system

$$\frac{dy_M}{dt} + Ay_M + P_M R(y_M + \Phi_{app}(y_M)) = P_M f, \tag{3}$$

with $\Phi_{app}(y_M) \in H_N - H_M$, for some $N > M$. Among several alternatives for Φ_{app} the following one, first introduced in [2], is of common use,

$$\Phi_{app}(y_M) = A^{-1}\{-(P_N - P_M)R(y_M) + (P_N - P_M)f\}. \tag{4}$$

The main advantage of the nonlinear Galerkin method over the standard Galerkin method based on the eigenfunctions of the operator A is its larger order of convergence. Depending on the regularity of the solution and under some general hypotheses, the error in the standard method is of order $1/\lambda_{M+1}^{1+\alpha}$ whereas in the nonlinear approach the error can be shown to be of order $1/\lambda_{M+1}^{\beta+\alpha}$, with, typically, $\beta = 3/2$ (see for example [1]).

Based on the superior accuracy of nonlinear Galerin methods, many papers on the subject, see [9] and the references cited therein, assert that they perform more efficiently than traditional standard spectral discretizations. Although some numerical test [6] seem to corroborate this assertion, the superior performance of the nonlinear Galerkin has been questioned in two very recent papers, [5], [10] where it is shown, by means of exahustive numerical experiments, that due to t he extra computational cost of the evaluation of the approximate inertial manifold associated with the nonlinear method, standard spectral or pseudospectral discretizations when used in conjunction with appropriate (and standard) time integrators are far more efficient than the more accurate nonlinear Galerkin.

However, the idea of using approximated inertial manifolds in the time integration of dissipative partial differential equations can be used to enhance the standard method, reaching exactly the same order of convergence than the nonlinear method. In [4] a postprocessed Galerkin method has been introduced. The postprocessed method uses the same approximate inertial manifold as the nonlinear Galerkin but it is evaluated only once, after the time integration of the equation is completed. The new method is proved to posses the same order as the nonlinear Galerkin method but at a computational cost virtually equal that of standard method.

A serious drawack in both the postproccesed and the nonlinear Galerkin methods is that they require knowing of the exact eigenvalues and eigenfunctions of the operator A. Furthermore even if these where known, the global accuracy of the method is limited not only by the regularity of the solution, but also by compatibility requirements at the boundary.

In this talk we show, both theoretically and numerically, that the same ideas can be used with spectral methods based on Legendre or Chebyshev polynomials overcoming the dificulties previously mentioned. We consider, as a model, the Burgers' equation with homogeneous Dirichlet boundary conditions,

$$\begin{array}{rcll} u_t - \nu u_{xx} + u u_x & = & f(x,t), & -1 < x < 1,\ t \geq 0, \\ u(-1,t) = u(1,t) & = & 0, & t \geq 0, \\ u(x,0) & = & u_0(x), & -1 < x < 1. \end{array} \qquad (5)$$

We introduce a postproccesed spectral Galerkin method for (5) and present some numerical analysis and experimental evidence of the superior perfomance of the new method when compared with standard spectral discretizations. The method is based on a partition of the solution on the analogous to the higher and lower modes [8].

More precisely, let ω the Chebyshev or Legendre wheigt and introduce the Sobolev spaces H^m_ω of the functions in $L^2_\omega = \{v \mid \int_{-1}^1 v^2 \omega dx < \infty\}$ with m distributional derivatives in L^2_ω. We make use of the space $H^1_{\omega,0} = \{v \in H^1_\omega \mid v(-1) = v(1) = 0\}$. Let $\mathbb{P}^M_0$ the space of polynomials of degree $\leq M$ that satisfy the boundary conditions. The spectral Galerkin approximation $p^M \in \mathbb{P}^M_0$ to Burgers' equation is defined by the equations

$$(\frac{dp^M}{dt}, \phi)_\omega + a_\omega(p^M, \phi) + (p^M p^M_x, \phi)_\omega = (f, \phi)_\omega, \forall \phi \in \mathbb{P}^M_0. \qquad (6)$$

where $a_\omega(\varphi, \phi) = \nu \int_{-1}^1 \varphi_x (\phi\omega)_x dx$. Note that, in the Chebyshev case, the linear operator is no longer selfadjoint.

For $N > M$, we define the space of polynomals $\mathbb{Q}^N_M$ by means of $\mathbb{P}^N_0 = \mathbb{P}^M_0 \oplus \mathbb{Q}^N_M$. Once the time integration of (6) is completed until time T, using any stiff time-integrator, for the postprocessed step we take $p^M = p^M(T)$, the last step in (6), and we look for a polynomial $q^N \in \mathbb{Q}^N_M$ solution of the following auxilary elliptic problem:

$$a_\omega(q^N, \phi) = -(p^M p^M_x, \phi)_\omega + (f, \phi)_\omega, \forall \phi \in \mathbb{Q}^N_M. \qquad (7)$$

The complete solution at time T is then $u^N = p^M + q^N$.

The error in the standard Galerkin approximation is of order $1/M^s$ for solutions of (5) in H^s_ω whereas the error following the postprocessed step can be shown to be of order $1/M^{s+1}$, for N big enough.

References

[1] C. Devulder, M. Marion and E.S. Titi, *On the rate of convergence of Nonlinear Galerkin method*, Math. Comput., 60 (1993), pp. 495-514.

[2] C. Foias, O. Manley and R. Temam, *Modelling of the interaction of small and large eddies in two dimensional turbulent flows*, RAIRO M^{2}AN, 22 (1988), pp. 93-114.

[3] C. Foias, G.R. Sell and R. Temam, *Inertial manifolds for non-linear evolutionary equations*, J. Diff. Eqq., 73 (1988), pp. 309-353.

[4] B. García-Archilla, J. Novo and E.S. Titi, *Postprocessing the Galerkin method: A novel approach to Approximate inertial manifolds*, unpublished report.

[5] B. García-Archilla and J. de Frutos, *Time integraton of the nonlinear Galerkin method*, IMA J. Numer. Anal., 15 (1995), pp. 221-244.

[6] F. Jauberteau, C. Rosier, and R. Temam, *A non-linear Galerkin method for the Navier-Stokes equations*, Comput. Methods Appl. Mech. Eng., 8 (1990), pp. 245-260.

[7] M. Marion and R. Temam, *Non-linear Galerkin methods*, SIAM J. Numer. Anal., 26 (1989), pp. 1139-1157.

[8] J. Shen and R. Temam, *Nonlinear Galerkin method using Chebyshev and Legendre polynomials I. The one dimensional case*, SIAM j. Numer. Anal., 1 (1995), pp. 215-234.

[9] R. Temam, *Stability analysis of the non-linear Galerkin method*, Math. Comput., 196 (1991), pp. 477-505.

[10] R. Wallace and D. Sloan, *Numerical solution of a nonlinear dissipative system using a pseudospectral method and inertial manifolds*, SIAM J. Sci. Comput., 16 (1995), pp. 1049-1070.

5.12 A Fortran 90 Mechanical System Solver

I. Gladwell[1], *K. Bouas-Dockery*[2]
Department of Mathematics, Southern Methodist University, Dallas, Texas 75275, U.S.A.
E-mail: h5nr1001@vm.cis.smu.edu
R. W. Brankin[1]
Numerical Algorithms Group Ltd., Wilkinson House, Jordan Hill Road, Oxford OX2 8DR, United Kingdom

[1] Partially supported by NATO Collaborative Research Grant 920037
[2] Partially supported by a NASA/Texas Space Grant Fellowship 1995-6

Computing the solution, $y \in \text{Re}^{\,n}$, of the initial value problem for the special second order ordinary differential system,

$$y'' = f(x,y),\ a \leq x,\ y(a) = y_a,\ y'(a) = y'_a, \tag{1}$$

is and has been a major problem in (Hamiltonian) mechanics problems. More or less independently methods have been developed by computational mathematicians and by engineers, chemists and celestial mechanists. The aim of computational mathematicians has been to develop accurate, stable methods which can be implemented in a standard way in general purpose software. In contrast, practitioners in applications' fields have developed methods which are sometimes problem specific and usually are designed to preserve physical quantities such as energy and angular momentum. These methods are often implemented in such a way as to be less accurate than the computational mathematician's counterpart algorithms but, at least in the sense of the conserved quantities, give a more realistic solution in a long time integration.

Recently, with the development of special methods (such as those popularly labeled symplectic or time-reversible) the computational mathematics community has come closer to providing physically acceptable solutions. At the same time mathematical analysis of popular "engineering" methods (such as the leapfrog-like method of Verlet and the methods associated with Newmark and Wilson) has demonstrated why they have been so successful in practice.

Two of the authors were involved in the development of a Fortran 77 explicit Runge-Kutta-Nyström (RKN) algorithm for the solution of (1), [1]. This algorithm gives a choice of methods, allows the user control of the integration and permits the user to query the code for integration statistics. To achieve this required a novel structural design. This design has been developed further in the popular code RKSUITE, [2], for the standard first order system

$$y' = f(x,y),\ a \leq x,\ y(a) = y_a, \tag{2}$$

which has recently been translated and redesigned into a Fortran 90 code, RKSUITE_90, [3]. The redesign permits much easier use of the code, such as can only be made available through a higher level language like Fortran 90. This code also permits a choice of algorithm and method similar to the choices in the RKN code. As in that code, the choice between methods is on the basis of their accuracy and stability and hence efficiency for the task at hand. We are just completing a Fortran 90 revision of the Fortran 77 RKN code in [1]. This revision gives it essentially the feel and ease-of-use of RKSUITE_90 but does not add significant functionality.

We are pursuing a variety of extensions to increase the functionality of this code for long-time integration and for conservation of physical properties. These fall into three categories:

1. Provision of a selection of popular formulas of symplectic and/or time-reversible type.
2. Provision of new "error control" facilities based on conservation.
3. Extension of the problem to be solved from problems that may be written directly in the form (1) to a "general" separable Hamiltonian mechanics problem; that is for the Hamiltonian differential equations

$$Mq'' = -\nabla V(q),\ Mq'(a) \equiv p(a) = p_a,\ q(a) = q_a, \tag{3}$$

corresponding to the separable Hamiltonian

$$H(p,q) = \frac{1}{2}p^T M^{-1} p + V(q), \tag{4}$$

where M is a symmetric positive definite "mass matrix" in a "kinetic energy" term and V represents "potential energy".

Corresponding to the above categories:

1. Presuming we are to use the new methods in fixed-step form, how should we determine how well they are performing and should we attempt to interfere with the user's choice of stepsize? Should we attempt a variable step implementation and, if so, should it be a conventional local error control or possibly of a type which is itself time-reversible?

2. It is attractive to control stepsize in either the conventional or new methods employing a user (or problem) defined error monitor. Particularly, using stepsize choice to control changes in an invariant of the differential system (such as energy) is attractive and will be tested; possibly a change of variables (such as to action-angle variables) will be required, [4].

3. Fortran 90 permits us significant flexibility in specifying different forms for M and V in (3) without the user having to negotiate a maze of interfaces and arguments. Particularly we will be able to deal with different forms of M (possibly factored) transparently. Similarly the user could be permitted to specify an (*a priori* unknown) number of invariants to be checked, and possibly for error control. However, Fortran 90 does not provide differentiation facilities to derive the Hamiltonian differential equations (3) from (4) nor to derive any solution invariants. We are investigating whether automatic differentiation codes which take Fortran 90 as data can be used in this application. Fortunately there are three which are also written in Fortran 90. So, there is a realistic chance of producing a portable code where the user is required only to specify the Hamiltonian, initial conditions and the usual integration control and display arguments.

In addition, we intend to exploit the availability of several low accuracy, good conservation property, and high accuracy, poor conservation property integrators in the same framework by permitting the user to switch simply between the integrators and between fixed step and error controlled integration. We have in mind here permitting accurate, local explorations during an "inaccurate", but physically realistic, long-time integration. Demonstrating that we can do this seamlessly should position us to build a robust package for the integration of Hamiltonian systems.

The first author designed what is believed to have been the first variable order, variable step code for the linear damped vibration problem, [6],

$$My'' + Cy' + Ky = f, \ a \le x, \ y(a) = y_a, \ y'(a) = y'_a. \tag{5}$$

This code, which was based on extensions of Newmark and Wilson-θ methods, [5], to provide variable stepsize and order, has been exciting some interest recently. A major design challenge was to compute the solution in small (or no) damping cases, in the presence of small components (noise) in the directions of the high-frequency modes. Experience of designing the Hamiltonian system solver above may encourage us to revisit this problem.

References

[1] R.W. Brankin, J.R. Dormand, I. Gladwell, P. Prince and W.L. Seward, ALGORITHM 670: A Runge–Kutta–Nyström Code, *ACM Trans. Math. Softw.* **15** (1989) 31–40.

[2] R.W. Brankin, I. Gladwell and L.F. Shampine, RKSUITE: A Suite of Explicit Runge–Kutta Codes, pp. 41–53 of *Contributions to Numerical Mathematics* (ed. R.P. Agarwal), *WSSIAA* **2**, World Scientific Press, 1993; and R.W. Brankin, I. Gladwell and L.F. Shampine, RKSUITE: a Suite of Runge–Kutta Codes for the Initial Value Problem for ODES. SMU Math. Softreport 92–S1, 1992 (also available on Netlib and in NAG Fortran 77 library).

[3] R.W. Brankin and I. Gladwell, A Fortran 90 Version of RKSUITE: An ODE Initial Value Solver. *Annals of Numerical Mathematics* **1** (1994) 363–375; and R.W. Brankin and I. Gladwell, RK-SUITE_90: Software for ODE IVPs NAG Tech. Report, TR6/94, 1994 (software available on Netlib and in NAG Fortran 90 library).

[4] I. Gladwell, G. Reddien and J. Wang, Energy Superconvergence of One-Step Methods for Separable Hamiltonian Sytems, *Phys. Lett. A* to appear.

[5] I. Gladwell and R.M. Thomas, Stability Properties of the Newmark, Houbolt and Wilson θ Methods, *Int. J. Numer. Anal. Math. Geomechanics* **4** (1980) 143–158; and I. Gladwell and R.M. Thomas, A Qualitative Analysis of Two– and Three–step Methods for Stable Second Order Systems, *SIAM J. Numer. Anal.* **22** (1985) 535–560.

[6] R.M. Thomas and I. Gladwell, Variable Order Variable Step Algorithms for Second Order Systems, I: The Methods, *Int. J. Numer. Meth. in Engng.* **26** (1988) 39–53; and I. Gladwell and R.M. Thomas, Variable Order Variable Step Algorithms for Second Order Systems, II: The Codes, *Int. J. Numer. Meth. in Engng.* **26** (1988) 55–80.

5.13 Symplectic Integrators for Quantum Dynamics

Stephen K. Gray
Theoretical Chemistry Group, Chemistry Division, Argonne National Laboratory, Argonne, Illinois 60439, U.S.A.
E-mail: gray@tcg.anl.gov

With spatial discretizations and/or basis set expansions, the partial differential equations of quantum mechanics can be described by large systems of ordinary differential equations. Numerical integration of such equations is important in many areas of physics and chemistry. In discrete form, both the time-dependent and (radial) time-independent Schrödinger equations can be written as *classical Hamiltonian* systems [1, 2]. Symplectic integrators (SIs)[3-6] developed for approximating classical problems can then be used for quantum ones. It is also possible to develop *new* SIs tailored to quantum dynamics[7].

The discrete time-dependent Schrödinger equation is ($\hbar = 1$)

$$\frac{d}{dt}\mathbf{\Psi}(t) \;=\; -i\;\mathbf{H}\cdot\mathbf{\Psi}(t) \quad , \tag{1}$$

where $\mathbf{\Psi}$ is an N-component *complex* vector and $\mathbf{H}$ is an N by N Hermitian matrix. Define two N-component *real* vectors $\mathbf{q} = \sqrt{2}Re\mathbf{\Psi}$ and $\mathbf{p} = \sqrt{2}Im\mathbf{\Psi}$. Eq. (1) is equivalent to the classical Hamiltonian equations

$$\frac{d}{dt}\mathbf{q} = \frac{d}{d\mathbf{p}}h \;,\; \frac{d}{dt}\mathbf{p} = -\frac{d}{d\mathbf{q}}h \quad , \tag{2}$$

where h is a certain classical Hamiltonian function. In the case of a real symmetric and time-independent $\mathbf{H}$,

$$h(\mathbf{q},\mathbf{p}) \;=\; \frac{1}{2}(\mathbf{p}^T\cdot\mathbf{H}\cdot\mathbf{p} \;+\; \mathbf{q}^T\cdot\mathbf{H}\cdot\mathbf{q}) \;=\; \frac{1}{2}\sum_{ij}H_{ij}(p_ip_j \;+\; q_iq_j). \tag{3}$$

Since h is of the "separable" form $h = T(\mathbf{p}) + V(\mathbf{q},t)$, explicit SIs[3-6] may be directly applied. (V in the present example does not depend on t.) These explicit SIs are all of the following form. Let $\mathbf{f}(\mathbf{q},t) = -dV(\mathbf{q},t)/d\mathbf{q}$ and $\mathbf{g} = dT(\mathbf{p})/d\mathbf{p}$. If $\mathbf{p}_0$ and $\mathbf{q}_0$ are the momenta and coordinates at time t_0, then the iterations

$$\begin{aligned} \mathbf{p}_k &= \mathbf{p}_{k-1} + b_k\tau\;\mathbf{f}(\mathbf{q}_{k-1},t_{k-1}) \\ \mathbf{q}_k &= \mathbf{q}_{k-1} + a_k\tau\;\mathbf{g}(\mathbf{p}_k) \\ t_k &= t_{k-1}+a_k\tau \quad , \end{aligned} \tag{4}$$

for $k = 1,...,m$ lead to $\mathbf{p}_m$ and $\mathbf{q}_m$, the approximation to the momenta and coordinates at time $t_0 + \tau$. The $2m$ coefficients $(\mathbf{a},\mathbf{b})$ may be chosen such that the integration is accurate to some order in the time step τ, and many such SIs (for m ranging from 1 to 6) are available. Recently, an analysis of these SIs in the context of large scale (classical) molecular dynamics simulations has been made[8]. Eq. (4) is simple to program and involves minimal storage requirements.

It is possible to develop explicit SIs just for Eq. (1). The essential point is that it is easy to write down the formal *exact* propagator for this problem in terms of cosine and sine operators acting on $\mathbf{H}\tau$. The Taylor series coefficients of the corresponding approximate propagator that results from Eq. (4) can also be deduced. The coefficients $(\mathbf{a},\mathbf{b})$ such that the Taylor series of the approximate and exact propagators agree, to some order, can then be obtained. Simulated annealing methods[9] are used to find the coefficients. Unlike general SIs, it is always possible to find SIs with m stages that are accurate to m'th order in the time step for *all* m. Significant improvements in accuracy (for fixed numerical effort) can be achieved with this approach. SIs accurate up to 12'th order in τ, with *reasonable* stability properties, can be obtained.

Another application of SI techniques is to the multichannel radial Schrödinger equation of *time-independent* scattering theory[2],

$$\frac{d^2}{dr^2}\mathbf{\Psi}(r) \;=\; \mathbf{W}(r)\mathbf{\Psi}(r) \quad , \tag{5}$$

where $\mathbf{W}(r)$ and $\mathbf{\Psi}(r)$ are N by N matrices, E is a fixed total energy and r is the radial variable. Such equations arise for many of the same problems that lead to Eq. (1). (Time-dependent and time-independent quantum mechanics are mathematically equivalent: solutions within one approach are related, via Fourier transforms, to solutions in the other approach. Depending on information desired, problem size, and available numerical methods, one approach might be more efficient than the another.) In many applications the matrices involved are real, and $\mathbf{W}$ is symmetric. Eq. (5) can be put in Hamiltonian form. Define *matrices* $\mathbf{q} = \mathbf{\Psi}$ and $\mathbf{p} = d\mathbf{\Psi}/dr$. Associate "time" with r. Eq. (5) is equivalent to Hamilton's equations, Eq. (2), with t replaced by r, and

$$h(\mathbf{p},\mathbf{q},r) \;=\; \frac{1}{2}tr(\mathbf{p}^T\cdot\mathbf{p} - \mathbf{q}^T\cdot\mathbf{W}(r)\cdot\mathbf{q}) \quad . \tag{6}$$

Eqs. (2) and (6) are solved from an intial condition at small r, out to large r. Many applications involve "classically forbidden" regions, or closed channel components such that $W_{ii}(r) > 0$, leading to strong instabilities in direct stepping algorithms such as Eq. (4). An invariant imbedding technique that avoids these problems is to solve for the log derivative matrix $\mathbf{y} = \mathbf{p}\cdot\mathbf{q}^{-1}$ instead of $\mathbf{p}$ and $\mathbf{q}$. A remarkable feature of the SI algorithm Eq. (4) is that it leads to a simple algorithm for propagating $\mathbf{y}$. Several applications are also discussed.

References

[1] S.K. Gray and J.M. Verosky, J. Chem. Phys. **100**, 5011 (1994).

[2] D.E. Manolopoulos and S.K. Gray, J. Chem. Phys. **102**, 9214 (1995).

[3] R.D. Ruth, IEEE Trans. Nucl. Sci. **NS-30**, 2669 (1983).

[4] J. Candy and W. Rozmus, J. Comput. Phys. **92**, 230 (1991).

[5] R.I. McLachlan and P. Atela, Nonlinearity **5**, 541 (1991).

[6] J.M. Sanz-Serna and M.P. Calvo, *Numerical Hamiltonian Problems*, (Chapman and Hall, London, 1994).

[7] S.K. Gray and D.E. Manolopoulos, *in preparation.*

[8] S.K. Gray, D.W. Noid, and B.G. Sumpter, J. Chem. Phys. **101**, 4062 (1994).

[9] E.g., W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes, 2nd Edition*, (Cambridge University Press, Cambridge, 1992).

5.14 Variable step size integration with symplectic methods

Ernst Hairer
Université de Genève, Section de Mathématiques, CP 240, 1211 Genève 24, Switzerland
E-mail: hairer@divsun.unige.ch

We consider the numerical treatment of Hamiltonian systems

$$\begin{aligned} p' &= -H_q(p,q) \\ q' &= H_p(p,q). \end{aligned} \tag{1}$$

It is well-known that a constant step size implementation of symplectic methods has favourable properties for an integration over long times. This can be explained by a backward error analysis argument.

If the system (1) has widely varying time scales in different regions of the phase space, an efficient integration needs variable step sizes. It became clear after numerical experiments of Calvo & Sanz-Serna (1992, 1993) that the standard use of variable step sizes destroys the good long-time behaviour of the constant step size implementation of symplectic methods. Attempts to find symplectic integrators $y_{n+1} = \Phi_h(y_n)$ and step size strategies $h_n = hs(y_n, h)$, for which the mapping $y_n \mapsto \Phi_{hs(y_n,h)}(y_n)$ is symplectic, have been vainly. Stoffer (1988, 1995) and Skeel & Gear (1992) have shown that this can be achieved only for (essentially) constant step size functions $s(y, h)$.

In this talk we present a new modification of standard symplectic integrators such that

- approximations (p_n, q_n) to $\big(p(t_n), q(t_n)\big)$ are obtained, where

  $$t_{n+1} = t_n + hs(p_n, q_n, h)$$

  for some given (non constant) function $s(p, q, h)$.
- the numerical solution can be (formally) interpreted as the exact solution of a perturbed Hamiltonian system on a non-uniform grid (backward error analysis), i.e.,

  $$p_n = \widetilde{p}(\widetilde{t}_n), \qquad q_n = \widetilde{q}(\widetilde{t}_n),$$

  where $\big(\widetilde{p}(t), \widetilde{q}(t)\big)$ is the exact solution of the perturbed system and $\widetilde{t}_n \approx t_n$. This implies that the algorithm is well-suited for long-time integrations (nearby conservation of energy over exponentially long times, linear error growth for integrable systems, ...).
- in the case, where the Hamiltonian is given by

  $$H(p,q) = \frac{1}{2} p^T M^{-1} p + V(q),$$

  explicit symplectic integrators of arbitrarily high order and with variable step sizes can be designed.

If the system (1) is also reversible, step size strategies that retain the good long-time behaviour have been designed by Hut, Makino & McMillan (1993), Stoffer (1995), and Huang & Leimkuhler (1997). We shall present several numerical experiments, including comparisons with these reversible step size strategies.

References

Calvo, M.P., Sanz-Serna, J.M.: *The development of variable-step symplectic integrators, with application to the two-body problem*, SIAM J. Sci. Comput. 14, 936-952 (1993).

Calvo, M.P., Sanz-Serna, J.M.: *Variable steps for symplectic integrators*, In: Numerical Analysis 1991 (D.F. Griffiths and G.A. Watson, eds), Pitman Research Notes in Mathematics Series 260, 34-48 (1992).

Hut, P., Makino, J., McMillan, S.: *Building a better leapfrog*, Astrophysics preprint series 93/74, Inst. Advanced Study, Princeton (1993).

Huang, W., Leimkuhler, B.: *The adaptive Verlet method*, Special Gear issue of SIAM J. Sci. Comput. (1997).

Skeel, R.D., Gear, C.W.: *Does variable step size ruin a symplectic integrator?*, Physica D 60, 311-313 (1992).

Stoffer, D.: *On reversible and canonical integration methods*, Research Report 88-05, SAM, ETH Zürich (1988).

Stoffer, D.: *Variable steps for reversible integration methods*, Computing 55, 1-22 (1995).

5.15 Adaptive Time-Stepping and Long Term Dynamics

Desmond J. Higham
Department of Mathematics and Computer Science, University of Dundee, Dundee DD14 1KN, Scotland
E-mail: dhigham@mcs.dundee.ac.uk

Background

It is now widely appreciated that discretizations of nonlinear evolutionary equations can give rise to **spurious** solutions; that is, solutions of the numerical map that are not related to the underlying continuous problem. Most attention has been paid to formulas for ordinary differential equations (ODEs) implemented with fixed time-steps. In particular, it is known that choosing a time-step based on a standard linearised stability analysis does not guarantee that spurious behaviour will be avoided.

For the autonomous initial value ODE system

$$y'(t) = f(y(t)), \quad t > 0, \quad y(0) = y_0, \tag{1}$$

an explicit, one-step, formula with time-step Δt updates $y_n \approx y(n\Delta t)$, to $y_{n+1} \approx y((n+1)\Delta t)$, according to some map $y_{n+1} = y_n + \Delta t \Phi(\Delta t, y_n)$. If $\Phi(\Delta t^*, y^*) = 0$ but $f(y^*) \neq 0$, then y^* is a spurious fixed point (of period one) for the time-step Δt^*. Many authors have investigated this phenomenon, and particular attention has been paid to the existence and stability of spurious fixed points as $\Delta t \to 0$ and also as Δt increases beyond the linearised stability limit for a true fixed point. Related work has been done for semi-discretised partial differential equations of reaction-diffusion type.

Hairer, Iserles and Sanz-Serna [HISS90] looked at Runge-Kutta (RK) formulas for (1) and defined a formula to be **regular** if it could be guaranteed never to introduce a spurious fixed point when f is continuous. They showed that Euler's Method is the only regular **explicit** RK formula and gave a recursive test for regularity of implicit RK formulas.

Although this potential for spurious solutions with fixed time-steps is worrying, it should be noted that most standard time-stepping software is adaptive. Hence, it is of interest to investigate spuriosity for variable time-stepping algorithms that use local error control.

Aves, Griffiths and Higham [3] considered algorithms based on explicit RK pairs. Here, the time-step is variable, so that $\Delta t_n := t_{n+1} - t_n$, and two approximations are computed:

$$\begin{aligned} y_{n+1} &= y_n + \Delta t_n \Phi(\Delta t_n, y_n), \\ \widehat{y}_{n+1} &= y_n + \Delta t_n \Psi(\Delta t_n, y_n). \end{aligned}$$

The difference $\|y_{n+1} - \widehat{y}_{n+1}\|$ is used for two related purposes: error control and time-step selection. It is shown in [3] that, for small values of the user-supplied tolerance, spurious fixed points of the algorithm arise when the individual formulas have simultaneous spurious fixed points; that is, when $\Phi(\Delta t^*, y^*) = 0 = \Psi(\Delta t^*, y^*)$ and $f(y^*) \neq 0$. It is argued in [3] that this condition is quite likely to arise for **scalar** problems, and hence spurious fixed points for adaptive algorithms cannot be ruled out. The stability of such fixed points is also investigated in [3]. On the positive side, it is shown that stable spurious fixed points only arise in special circumstances—essentially the intersection of the fixed point branches $\Phi(\Delta t, y) = 0$ and $\Psi(\Delta t, y) = 0$ must be tangential. On the other hand, however, by defining f to be locally piecewise linear, stability, with a non-trivial basin of attraction, is possible.

New Material

The talk will cover new results that concern the the avoidance of spurious fixed points in adaptive time-stepping. First, we will look at **regular Runge-Kutta pairs**; that is, pairs which, together, guarantee that the standard adaptive algorithm will not produce a spurious fixed point (for sufficiently small tolerances). Second, we will describe some alternative approaches for error control and time-step selection that will 'regularise' any Runge-Kutta pair.

Regular Runge-Kutta Pairs

Following the analysis in [3] it can be argued that, for small tolerances, a Runge-Kutta pair will not admit a spurious fixed point if the individual formulas do not admit a simultaneous spurious fixed point. We emphasise that this is a weaker demand than asking for at least one of the individual formulas to be regular. For example, consider the 2nd and 3rd order explicit pair defined by

$$\begin{aligned}
\xi_1 &= f(y_n), \\
\xi_2 &= f(y_n + h_n \xi_1), \\
\xi_3 &= f(y_n + h_n(\xi_1 + \xi_2)/4), \\
y_{n+1} &= y_n + h_n(\xi_1 + \xi_2 + 4\xi_3)/6, \\
\widehat{y}_{n+1} &= y_n + h_n(\xi_1 + \xi_2)/2.
\end{aligned}$$

(Note that this pair is used in the `ode23.m` ODE-solving function of Matlab [Mat].) The individual formulas are not regular—this follows from Corollary 4 of [HISS90], where it is proved that any explicit Runge-Kutta formula with order greater than one cannot be regular. However, any fixed point that is shared by the formulas must satisfy

$$\frac{1}{6}\xi_1 + \frac{1}{6}\xi_2 + \frac{4}{6}\xi_3 = 0 \quad \text{and} \quad \frac{1}{2}\xi_1 + \frac{1}{2}\xi_2 = 0 \quad \Longrightarrow \quad \xi_3 = 0.$$

Also,

$$z_3 = y + \frac{1}{4}\xi_1 + \frac{1}{4}\xi_2 = y.$$

So, $f(y) = f(z_3) = \xi_3 = 0$. Hence, any simultaneous fixed point is not spurious.

This example shows that certain pairs of formulas, used in a standard error control process, can work together to eliminate spurious fixed points. The theory in [HISS90] can be extended to the case of pairs of formulas. In particular, we will show that there is a recursive algorithm that can be used to test whether a given pair of formulas is regular. Work is currently in progress to derive further examples of regular pairs of formulas, and to obtain order bounds on this type of regularity.

New Time-Stepping Schemes

An alternative way to avoid spuriosity is to alter the error control and time-step changing process. By adding extra conditions to the test that determines whether a step is accepted or rejected, it is possible to derive an algorithm that, for **any** RK pair, guarantees that no spurious fixed points are computed. As a simple example, the condition

$$\frac{\|y_{n+1} - y_n - h_n f(y_n)\|}{h_n \|f(y_n)\|} \leq \frac{1}{2}$$

eliminates spurious fixed points. More sophisticated tests are possible, some of which also eliminate period two solutions. Since the standard error control policy has been found to work well on a large variety of ODEs, it is important to make changes that do not affect the behaviour of the algorithm when it is not near a spurious solution. Although it is relatively easy to strengthen the error control criteria so that a pair is 'regularised', a key difficulty is to modify the time-step changing mechanism so that the algorithm remains efficient.

This work on new time-stepping schemes is being performed jointly with Tony Humphries (Bristol) and Richard Wain (Dundee).

References

HISS90 E. Hairer, A. Iserles and J.M. Sanz-Serna, *Equilibria of Runge-Kutta methods*, Num. Math. 58, 1990, 243–254.

Mat The MathWorks, Inc., *MATLAB User's Guide*, Natick, Massachusetts, 1992.

Relevant, recent work of the author

1 *The dynamics of a discretised nonlinear delay differential equation*, by D.J. Higham, in the *Proceedings of the 15th Biennial Conference on Numerical Analysis, Dundee, 1993*, Editors, G.A. Watson and D.F. Griffiths, Pitman Research Notes in Mathematics, Volume 303, pages 167–179, Longman Scientific and Technical, Essex.

2 *Existence and stability of fixed points for a discretised nonlinear reaction-diffusion equation with delay*, by D.J. Higham and T. Sardar, Appl. Num. Math. 18, (1995), 155–173.

3 *Does error control suppress spuriosity?*, by M.A. Aves, D.F. Griffiths and D.J. Higham, to appear in SIAM J. Numer. Anal.

4 *Non-normality effects in a discretised, nonlinear, reaction-convection-diffusion equation*, by D.J. Higham and B. Owren, Department of Mathematical Sciences Preprint No. 5/1994, NTH (Norway) University. To appear in J. Comp. Phys.

5 *Analysis of the dynamics of local error control via a piecewise continuous residual*, by D.J. Higham and A.M. Stuart, Stanford University Technical Report SCCM-95-03, 1995.

6 *Time-stepping and preserving orthonormality*, by D.J. Higham, Numerical Analysis Technical Report NA/162, University of Dundee, 1995.

5.16 Exponential integrators for large systems of ODEs

M. Hochbruck and Ch. Lubich
Universität Tübingen, Mathematisches Institut, Auf der Morgenstelle 10, 72076 Tübingen, Germany
E-mail: lubich@na.uni-tuebingen.de

The idea to use the exponential function of the Jacobian in a numerical integrator is by no means new, but so far it has mostly been regarded as a rather impractical approach. Our recent study of Krylov subspace approximations to the action of the matrix exponential operator has, however, led us to revive such an approach. We showed that Krylov approximations to $\exp(\tau A)v$ converge substantially faster than those for the solution of linear systems $(I-\tau A)x = v$, which usually arise in the numerical integration of stiff systems of ODEs, at least unless a good preconditioner for the linear system is available.

We therefore consider the following class of methods for the integration of large ODE systems $y' = f(y)$, with A denoting (an approximation to) the Jacobian:

$$\begin{aligned} k_i &= \varphi(\gamma h A)\Big(f(y_0 + h\textstyle\sum_{j=1}^{i-1}\alpha_{ij}k_j) + h\sum_{j=1}^{i-1}\gamma_{ij}k_j\Big) \qquad i = 1,\dots,s \\ y_1 &= y_0 + h\textstyle\sum_{i=1}^{s} b_i k_i \ . \end{aligned}$$

For $\varphi(z) = 1/(1-z)$, this would just be a standard Rosenbrock-Wanner method. We choose instead $\varphi(z) = (e^z - 1)/z$. With this choice, the method coefficients can be chosen such that the method is exact for constant-coefficient linear problems $y' = Ay + b$. Only a finite (small) number of conditions have to be satisfied for this property. We remark that Krylov subspace approximations to $\varphi(\tau A)v$ converge about as fast as for $\exp(\tau A)v$.

For example, we have constructed the following method, which could be reformulated to fit into the above class:

$$\begin{aligned}
l_1 &= \varphi(1/3 \cdot hA)f(y_0) \\
l_2 &= \varphi(2/3 \cdot hA)f(y_0) \\
l_3 &= \varphi(hA)f(y_0) \\
w_4 &= -7/300 \cdot l_1 + 97/150 \cdot l_2 - 37/300 \cdot l_3 \\
u_4 &= y_0 + hw_4 \\
d_4 &= f(u_4) - f(y_0) - hAw_4 \\
l_4 &= \varphi(1/3 \cdot hA)d_4 \\
l_5 &= \varphi(2/3 \cdot hA)d_4 \\
l_6 &= \varphi(hA)d_4 \\
w_7 &= 59/300 \cdot l_1 - 7/75 \cdot l_2 + 269/300 \cdot l_3 + 2/3 \cdot (l_4 + l_5 + l_6) \\
u_7 &= y_0 + hw_7 \\
d_7 &= f(u_7) - f(y_0) - hAw_7 \\
l_7 &= \varphi(1/3 \cdot hA)d_7 \\
y_1 &= y_0 + h(l_3 + l_4 - 4/3 \cdot l_5 + l_6 + 1/6 \cdot l_7)
\end{aligned}$$

This method has classical order 4, is exact for linear problems $y' = Ay + b$, and has order 3 for index-1 differential-algebraic systems and singularly perturbed problems. It has order 2 when used with inexact Jacobian A, and order 3 when A is $O(h)$-close to the true Jacobian. An embedded method for step size control and a dense output are also available.

When one uses a Krylov method to compute $\varphi(\tau A)v$, then the main work is in the computation of $\varphi(hA)f(y_0)$. The computations of $\varphi(1/3 \cdot hA)f(y_0)$ and $\varphi(2/3 \cdot hA)f(y_0)$ are obtained along the way, using simple recursions for $\varphi(2z)$ and $\varphi(3z)$. The φ-multiplications with d_4 and d_7 are usually cheaper, because d_4 and d_7 are typically rather small (they vanish for linear problems) and hence need only few iterations.

For long-time computations with conservative problems, we have constructed a symmetric two-step scheme of order 2 that is exact for linear constant-coefficient problems, and another such scheme for second-order equations.

We present numerical experiments with the proposed methods applied to reaction-diffusion equations (with stiff and non-stiff chemistry), to Schrödinger equations and to wave equations. The experiments show both the scope and the limitations of the approach.

5.17 Solving Implicit Differential Equations on Parallel Computers

P.J. van der Houwen and W.A. van der Veen
CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
E-mail: senna@cwi.nl

We consider initial value problems (IVPs) for systems of implicit differential equations (IDEs)

$$\varphi(\dot{\mathbf{y}}(t), \mathbf{y}(t)) = \mathbf{0}, \qquad \mathbf{y}, \varphi \in R^d. \tag{1}$$

It will always be assumed that the initial conditions for $\mathbf{y}(t_0)$ and $\dot{\mathbf{y}}(t_0$ are consistent and that the IVP has a unique solution. The class of Runge-Kutta methods for solving (1.1) can be represented in the form (see e.g. [2, p.406])

$$\mathbf{y}_{n+1} = (\mathbf{e}_s^T \otimes I)\mathbf{Y}, \mathbf{R}(\mathbf{Y}) = \mathbf{0}, \mathbf{R}(\mathbf{Y}) := \Phi((h^{-1}A^{-1} \otimes I)(\mathbf{Y} - \mathbf{e} \otimes \mathbf{y}_n), \mathbf{Y}). \tag{2}$$

Here, A denotes a nonsingular s-by-s matrix, I is the d-by-d identity matrix, h is the stepsize $t_{n+1} - t_n$, and $\otimes$ denotes the Kronecker product. The s vector components $\mathbf{Y}_i$ of the sd-dimensional solution vector $\mathbf{Y}$ represent numerical approximations to the exact solution vectors $\mathbf{y}(\mathbf{e}t_n + \mathbf{c}h)$, $\mathbf{c}$ being the abscissa vector with $c_s = 1$ and $\mathbf{e}$ representing the s-dimensional vector with unit entries. Furthermore, $\mathbf{e}_s$ is the sth unit vector and $\mathbf{y}_n$ is the numerical approximation to $\mathbf{y}(t_n)$. In (2), $\Phi(\mathbf{U}, \mathbf{V})$ contains the values $(\varphi(\mathbf{U}_i, \mathbf{V}_i))$ for any pair of vectors $\mathbf{U} = (\mathbf{U}_i)$ and $(\mathbf{V}_i)$.

The usual approach for solving the (nonlinear) implicit equation in (2) is by the modified Newton method

$$N_0(\mathbf{Y}^{(j)} - \mathbf{Y}^{(j-1)}) = -(hA \otimes I)\mathbf{R}(\mathbf{Y}^{(j-1)}), N_0 := I \otimes K - A \otimes hJ, j = 1, \ldots, m, \tag{3}$$

where $K := \varphi_{\mathbf{U}}(\mathbf{u}, \mathbf{v})$ and $J := -\varphi_{\mathbf{V}}(\mathbf{u}, \mathbf{v})$ (in the definition of J, the minus sign is inserted so that *explicit* differential equations with $\varphi = \mathbf{y}' - \mathbf{f}(t, \mathbf{y})$ yield the familiar formula $J = \partial \mathbf{f}/\partial \mathbf{y}$). Each iteration in (3) requires the solution of a sd-by-sd dimensional linear system. Already for moderate values of d, this is quite expensive. For example, when direct methods are used, the LU- decomposition of the sd-by-sd matrix N_0 requires as many as $\frac{2}{3}s^3d^3$ arithmetic operations. The computational work can be reduced by reducing (3) to 'block-diagonal form' by means of a Butcher transformaion such that the matrix of coefficients of the transformed linear systems has (real) diagonal blocks, each block corresponding to an eigenvalue of A (cf. [1]). If an eigenvalue is real, then the associated diagonal block is of order d, otherwise it has order $2d$. The LU-costs of these blocks are reduced to $\frac{2}{3}d^3$ and $\frac{16}{3}d^3$ operations, respectively. Assuming that the corrector (1) is based on a Radau IIA formula, and because such methods are defined by matrices A with atmost one real eigenvalue, the LU-costs are $\frac{2}{3}(4s-3)d^3$ operations for s odd and $\frac{2}{3}(4s)d^3$ operations for s even. However, since the LU-decompositions can be computed concurrently on a parallel computer system, the Butcher transformation reduces the *effective* computational costs to atmost $\frac{16}{3}d^3$ operations, irrespective the value of s. In this contribution, we propose a parallel iterative linear system method for solving the linear systems in (3) requiring s LU-decompositions of matrices of order d. The LU-decompositions can be computed concurrently, so that the effective LU-costs are only $\frac{2}{3}d^3$ operations, irrespective the value of s, resulting in a reduction by a factor 8. This parallel iterative linear system method is defined by

$$N(\mathbf{Y}^{(j,\nu)} - \mathbf{Y}^{(j,\nu-1)}) = -N_0\mathbf{Y}^{(j,\nu-1)} + \mathbf{C}^{(j-1)}, N := I \otimes K - B \otimes hJ, \tag{4}$$

$$\nu = 1, 2, \ldots, r,$$

$$\mathbf{C}^{(j-1)} := N_0\mathbf{Y}^{(j-1)} - (hA \otimes I)\mathbf{R}(\mathbf{Y}^{(j-1)}),$$

where $\mathbf{Y}^{(j,0)} = \mathbf{Y}^{(j-1,r)}$ and where B is a nondefective matrix with *positive* eigenvalues. The positive spectrum of B enables us to diagonalize the linear system, implying that only LU-decompositions of d-by-d matrices are involved. Note that also on sequential computers, the LU-costs (amounting to $\frac{2}{3}sd^3$ operations) compare favourably with those associated with a direct transformation of (3). For example, if $s = 3$, the LU costs associated with the transformed forms of (3) and (4) respectively are $6d^3$ and $2d^3$ operations.

However, the price to be paid is the introduction of the (inner) iteration process (4) requiring r forward-backward substitutions. If r is large, then the advantage of the reduced LU costs is easily lost. Thus, the task is to construct a matrix B such that (4) is a fast converging iteration process. The convergence of (4) can be studied by deriving the (exact) error recursion for $\mathbf{Y}^{(j,\nu)} - \mathbf{Y}^{(j)}$, i.e.

$$\mathbf{Y}^{(j,\nu)} - \mathbf{Y}^{(j)} = M(\mathbf{Y}^{(j,\nu-1)} - \mathbf{Y}^{(j)}), M := (I \otimes K - B \otimes hJ)^{-1}((A - B) \otimes hJ). \tag{5}$$

The construction of suitable matrices B can be achieved along the lines of [3] and [4]. Experiments show that in most problems, the linear systems in (3) are solved within a few (inner) iterations.

References

1. J.C. Butcher (1976): On the implementation of implicit Runge-Kutta methods, BIT 16, 237-240.
2. E. Hairer and G. Wanner (1991): Solving ordinary differential equations, II. Stiff and differential-algebraic problems, Springer-Verlag, Berlin.
3. P.J. van der Houwen and B.P. Sommeijer (1991): Iterated Runge-Kutta methods on parallel computers, SIAM J. Sci. Stat. Comput. 12, 1000-1028.
4. P.J. van der Houwen and J.J.B. de Swart (1995): Triangularly implicit iteration methods for ODE-IVP solvers, Preprint NM-R9510, CWI, Amsterdam (to appear in SIAM J. Sci. Comput.).

5.18 Stability results for implicit-explicit linear 2-step methods

Willem Hundsdorfer
CWI, P.O.Box 94079, 1090 GB Amsterdam, The Netherlands
E-mail: willem@cwi.nl

Note. The material for this talk is based on joint work with Jason Frank and Jan G. Verwer.

Frequently in applications one is confronted with problems having both stiff and nonstiff components. For example, in atmospheric chemistry one may have a nonstiff advection term and a stiff reaction-diffusion term. In such cases it is desirable to treat the stiff part implicitly and the nonstiff part explicitly. We consider initial value problems of the form

$$w'(t) = F(t, w(t)) + G(t, w(t)) \quad (0 \le t \le T), \qquad w(0) = w_0, \tag{1}$$

where F represents the nonstiff part and G stands for the stiff part of the system. The general form of an implicit-explicit (IMEX) linear multistep method is

$$\sum_{j=0}^{k} \alpha_j w_{n+1-j} = \tau \sum_{j=1}^{k} \beta_j F(t_{n+1-j}, w_{n+1-j}) + \tau \sum_{j=0}^{k} \gamma_j G(t_{n+1-j}, w_{n+1-j}). \tag{2}$$

Here $\tau > 0$ denotes the time step and the vectors w_n approximate the exact solution at $t_n = n\tau$. Methods of this type were introduced in 1980 by Crouzeix [2] and Varah [4]. A natural way to derive such a method is to start with an implicit method that is known to posess favourable stability properties, and then replace the term $F(t_{n+1}, w_{n+1})$ by a linear combination of explicit terms using extrapolation [3].

It is necessary to determine under what conditions such schemes are stable. We consider the scalar test equation

$$w'(t) = \lambda w(t) + \mu w(t), \tag{3}$$

where λ is of moderate size and μ is a stiff term. It will be assumed that λ and μ are not coupled. As a simple example consider the first order IMEX Euler method

$$w_{n+1} - w_n = \tau F(t_n, w_n) + \tau G(t_{n+1}, w_{n+1}). \tag{4}$$

To simplify the notation we will make in the following the substitutions $\lambda \to \tau\lambda$ and $\mu \to \tau\mu$. Then application of this method to the test equation gives

$$w_{n+1} = \frac{1+\lambda}{1-\mu} w_n,$$

and it easily follows that the method is stable whenever λ lies in the stability region of the explicit Euler method, $|1+\lambda| \le 1$, and μ is in the stability region of the implicit Euler method, $|1-\mu| \ge 1$. As we shall see, this is an exeptional situation. Usually, stability of the individual explicit and implicit methods does not guarantee stability of the combined IMEX method.

In this talk we consider several second order IMEX 2-step methods, where the implicit method is A-stable. We shall address two questions:

- Suppose that λ lies in the stability region of the explicit method. What restrictions are to be placed on the location of μ to have stability?
- What additional restrictions, if any, may be imposed on the location of λ to ensure that the IMEX method is stable for all μ in the left half-plane?

References

[1] U.M. Ascher, S.J. Ruuth, B.Wetton, *Implicit-Explicit Methods for Time-Dependent PDE's*, SIAM J. Numer. Anal. 32 (1995), pp. 797-823.

[2] M. Crouzeix, *Une méthode multipas implicite-explicite pour l'approximation des équations d'évolution paraboliques.* Numer. Math. 35 (1980), pp. 257-276.

[3] W. Hundsdorfer, J.G. Verwer, *A note on splitting errors for advection-reaction equations.* Appl. Num. Math. 18 (1995), pp. 191-199.

[4] J.M. Varah, *Stability restrictions on second order, three-level finite-difference schemes for parabolic equations.* SIAM J. Numer. Anal. 17 (1980), pp. 300-309.

[5] J.G. Verwer, J.G. Blom, W. Hundsdorfer, *An implicit-explicit approach for atmospheric transport-chemistry problems*, CWI Report NM-R9501, 1995.

5.19 Composition of maps as integrators

Robert I. McLachlan
The Sir Isaac Newton Institute, Cambridge, England
Permanent address:
Mathematics, Massey University, Palmerston North, New Zealand
E-mail: R.McLachlan@massey.ac.nz

It is sure to be a theme of this workshop that *qualitative* features of solutions must be emphasized when striving for good long-time integrations; and that the best way to achieve this is to study the geometric and structural properties of the system at hand, such as those preserved under composition. Each property's independent effect on systems' long-time dynamics, and its interaction with other properties, must be explored.

I will take these for granted and survey first the analysis of splitting and composition methods, returning later to the question of possible new viewpoints for numerical integration.

Analysis of composition methods

To fix the notation, we are working with a system of differential equations

$$\dot{x} = X(x)$$

with solution

$$x(t) = \exp(tX)(x(0)).$$

It often happens that

$$X(x) = A(x) + B(x)$$

with A and B integrable. If so, we say that the original vector field X has been 'split,' and can consider *composition methods* of the form

$$\varphi_t(X) = \prod_{i=1}^{n} \exp(ta_i A)\exp(tb_i B)$$

The advantage of such methods is that many properties of the true solution can be automatically inherited by the integrator, such as symplecticity, volume preservation, some integrals, reversibility, and so on. (Reversibility is not a group, but a 'pseudogroup' property—see later.)

There are two ways to study such a composition. We can look at the map $\varphi_t(x)$ ($\approx \exp(tX)$) itself, or we can write

$$\varphi_t(X) = \exp(tX_1 + t^2X_2 + \ldots) = \exp(t\tilde{X})$$

and look at $t\tilde{X}$, the autonomous vector field with time-1 flow $\varphi_t(X)$. The *map* method is traditional in numerical analysis and has been adapted to the present case and highly refined by many people, most notably Sanz-Serna. The *vector field* method is a standard technique in, e.g., celestial mechanics (Lie theory), and numerical analysis (backward error analysis). The two fields' techniques are now being fruitfully merged.

I believe that both approaches should be developed in parallel, and that even more representations should be explored, until the full correspondence is clear. Studying the vector field has led to methods of greater generality and easily counts determining equations, but so far has not encompassed, e.g., the symplectic Runge-Kutta methods. Conversely, the map approach has not been extended to 'symmetric compositions' (similar to diagonally implicit symplectic Runge-Kutta), but the determining equations themselves can be written down more easily. (I hope that symbolic manipulation and Suzuki's new operator calculus may help the vector field method here.)

For compositions, the terms in the perturbed vector field $\tilde{X}$ are Lie brackets of A and B, and hence can be studied using the theory of *free Lie algebras*. This makes it easy to count the determining equations at each order under various special cases. It also allows one to show that some methods have much greater applicability than they were first derived for: (i) methods derived in the separable case $\dot{x} = A = f(y)$, $\dot{y} = B = g(x)$, in fact work for all A, B; (ii) methods composing exact flows are equivalent to methods composing an arbitrary map; (iii) methods designed for classical mechanical systems $\dot{q} = A = p$, $\dot{p} = B = V'(q)$, in fact work (almost always!) for all A, B with $[B,[B,[B,A]]] = 0$.

Other cases where composition methods are unusually effective are near-integrable systems (because they preserve the near-integrability) and in the trace formulas of quantum statistical mechanics. They also lend themselves well to *corrector* methods, of the form

$$\varphi = CKC^{-1}$$

which amounts to integrating the system in different variables, reminiscent of classical normal form theory. In this case, though, the change of variables C must be expressed covariantly with respect to the vector field, so the simplifications are not quite as dramatic.

Here if φ has order p then Butcher (who first introduced this idea) said that K has *effective order* p. The corrector C does not need to be evaluated or even known—surely any phenomenon worth studying is invariant under coordinate changes! One should compare corrected methods of effective order p to corrected methods of standard order p, since correctable errors aren't important anyway. By this standard John Butcher and I have shown that, for two-stage symplectic Runge-Kuttas, no improvement is possible over standard methods. Nevertheless, at higher order there are some gains: for example, one can now achieve effective order 8 with 9 stages instead of 15.

Extensions have also been made by many people to classical mechanical systems, to multiderivative methods, and to singly implicit methods; to my knowledge effective order has not been explored to high order for general Runge-Kuttas.

What is the space of numerical methods?

I would like to bring up a a few general points which have been nagging me and which I have barely been able to detect in the literature. In studying structural properties of integrators, we should clearly separate effects due to the particular form of a method (e.g., Runge-Kutta) and those which are due to it belonging to some wider class of functions (e.g., symplectic maps). To do this some representation of the space of integrators is needed. What should it be? 'Maps' is one answer, and this is enough for many purposes. However, this class does not include the dependence of the integrator on the vector field, which can come in many forms.

What is the difference between explicit and implicit methods and exact flows? They are all just maps on phase space. The difference is in the algorithm by which the maps are evaluated. This complexity/computability question, the 'computer science' dimension of numerical analysis, has hardly begun to be studied (even with the urgings and contributions of Smale). Almost all of numerical analysis is in fact analysis rather than computer science. Amongst all sequences of N operations which approximate $\exp(tX)$ for a given t and X, is it likely that a Runge-Kutta[/Newton] method is the best? Unfortunately there are few examples of what I mean here. The arithmetic-geometric mean algorithm for elliptic functions is one ('elementary') example, which suggests considering fixed points of general iterations.

It would also be nice to incorporate abstractly the dependence of the method on the information it is given, such as the vector field, its derivative, any first integrals, etc. But without the correct framework we are restricted to writing down, e.g., Obreshkov methods, and studying these. Care is required because even the exact solution $\exp(tX)$ 'depends on' only X.

Flows have many nice properties, many of which are desirable in an integrator as well. Some of these properties are defined for all vector fields, such as covariance under transformation groups (changing variables in v.f. = changing variables in map), closure under differentiation, and time symmetry. Others are defined only for the flows of a some vector fields, such as symplecticity for Hamiltonian systems, closure under restriction to closed subsystems, and energy conservation. It is natural to try to 'lift' these properties so that they are naturally defined for all vector fields. For example, Ge's definition of invariance with respect to symplectic transformations is only defined for Hamiltonian systems, but it can also be seen as a special case of general covariance. Such lifts can't be unique; can they be natural?

Finally, a tenuous link back to composition methods. (Incidentally, composition is a pathetically simple algorithm, which irritates me.) Let φ and ψ be methods. To preserve symplecticity, say, the composition $\varphi\psi$ suffices; but to preserve time-symmetry ($\varphi_t = \varphi_{-t}^{-1}$), we must take $\varphi\psi\varphi$. So call time-symmetry a 'pseudogroup' property. Another example: let φ and ψ be orthogonal (respectively symmetric) matrices. Recently, Reinout Quispel and I have explored this slightly trivial point in the context of groups of automorphisms and antiautomorphisms of a group. For integrators, we take the group generated by maps associated to vector fields, e.g., $\exp(tX)(1+tY)$ is in the group. This point of view elucidates the relationships between properties such as time-symmetry and reversibility, which are independent, although related in some special cases considered previously. We also see how a map covariant under a larger group than that strictly necessary for symmetry preservation can be more flexible under composition, leading to systematic constructions of (for example) volume-preserving, time-symmetric, reversing-symmetry-group-preserving methods. It remains to be seen whether the algebraic structures used are important in their own right.

5.20 Explicit Difference Schemes with Variable Time Steps for Solving Stiff Systems of Equations

V.I. Lebedev
Institute of Numerical Mathematics, Russian Academy of Sciences, Leninsky prospect 32A, 117993 Moscow, Russia
E-mail: nucrect@post.inm.ras.ru

Problem

In applications often occurs a need to solve Cauchy problems for stiff differential equations or equations derived by the method of lines, when the use of implicit schemes is sophisticated, and the step in time explicit schemes use is too short.

The core
To solve Cauchy problems for stiff differential equations or equations derived from the method of lines, were proposed [1,2,3] explicit stable difference schemes with the steps, varying in time. The conditions of stability of the optimal algorithm of choice of steps were investigated. The algorithm provides a drastic improvement compared with known explicit schemes (up to 30 000 000). A special algorithm is used for interior and boundary layers. Algorithms are based on the properties of T-sequence of roots of Tchebyscheff polynomials.

The advantages of the proposed method
The methods are convenient to implement on parallel and pipeline computers, is suitable to solve highly multidimensional problems, problems with nonlinear, nonsymmetric, nondefinite operators, for unconstrained optimization. The use of explicit stable difference schemes gives us a possibility almost absolutely parallelize and vectorize computations.

The code
The method is used in FORTRAN code DUMKA. Computations were conducted on CYBER, ELBRUS, BESM and different PCs.

The method is quite flexible. The use in applications is simple, since **DUMKA** user needs the just to write a subroutine for right-hand side of equation and a subroutine, estimating the greatest in modulus negative eigenvalue of Jacobean.

It seems advantageous to use DUMKA for neutron transport problems such as multidimensional nonlinear heat transfer problems and unstationary cinetic problems with delayed neutrons.

Explicit Stable Method for Stiff Systems

Let's write Cauchy problem for system of ordinary differential equations as

$$\frac{dU}{dt} = f(U,t) \tag{1}$$

$$U|_{t=t_0} = U_0 \tag{2}$$

where $U = (U_1, U_2, \ldots, U_n), f = (f_1, f_2, \ldots, f_n)$.
Let

$$J = \| \frac{df_i}{dU_j} \| \tag{3}$$

be Jacobean matrix, t_k - time-mesh, h_k- steps in time.

Let first the spectrum of J be real and λ be an upper estimate of modulus of negative eigenvalue of Jacobean matrix; the figure

$$COU = 2/\lambda \tag{4}$$

we will call courant.

The approximate solution $U_k = U(t_k)$ of the Cauchy problem (1) we will find by the following explicit difference rule

$$\begin{aligned}
&y_{k+1/2} = U_k + h_{k+1} f(U_k, t_k),\\
&t_{k+1/2} = t_k + h_{k+1},\\
&y_{k+1} = y_{k+1/2} + h_{k+1} f(y_{k+1/2}, t_{k+1/2}),\\
&U_{k+1} = y_{k+1} + \gamma_{k+1} h_{k+1} (f(U_k, t_k) - f(y_{k+1/2}, t_{k+1/2}))\\
&t_{k+1/2} = t_{k+1/2} + h_{k+1},\\
&k = 0, 1, \ldots, N-1
\end{aligned} \tag{5}$$

where h_k, γ_k are parameters chosen in a certain way.

In explicit Euler's rule with the constant step h:

$$U_{k+1} = U_k + hf(U_k, t_k), \tag{6}$$

on the step h by the stability condition the constrain is imposed

$$h \leq COU \tag{7}$$

The method (5) is implemented in the code **DUMKA**.

The code **DUMKA** from given tolerance EPS derives a series of steps $h_{k+1}, h_{k+2}, \ldots, h_{k+N}$, which guarantee stability and the given tolerance. So the transition operator of the difference rule after N steps forms out the operator Tchebysheff polynomial of the first kind of the $2N$th degree [1,2,3] and Zolotarev polynomial [4].

Let $g = h/cou$,

$$h^N = \frac{1}{N} \sum_{i=1}^{N} h_{k+i} (N = 2^k * 3^m, k = 0, 1, \ldots, 14, m = 0, 1, \ldots, 6) \tag{8}$$

be the average step. The values of N, h^N depend on the smoothness of the solution, the tolerance EPS and g.

So are the abilities of **DUMKA** code, that $N \leq 47\,775\,744$, and

$$h^N \leq 3 * 10^7 * COU. \tag{9}$$

The equality in (9) can be reached on the intervals of integration where the solution $U(t)$ is asymptotically linear in t. From the comparison of (7) and (9) follows, that **DUMKA** will solve the Cauchy problem by explicit rule (4) with steps in time $0.4358 * 10^7$ times greater in average then those of explicit Euler's rule (6).

Let $z = x + iy$. The algorithm implemented is able to account on some specific complex spectrum domain configurations of J: ellipsis with focus on the real axis with (0,0) belonging to the boundary of the ellipsis and with smaller semiaxis l; domain in the neighborhood of imaginary axis

$$x = 0, \quad -b \leq y \leq b, \quad b > 0 \tag{10}$$

the neighborhood of two intervals

$$y = 0, \quad x \in [0, -\frac{\lambda}{2}(1-b)], \bigcup [-\frac{\lambda}{2}(1+b), -\lambda], \quad 0 \leq b < 1, \tag{11}$$

the neighborhood of the "crux"

$$\{y = 0, -\lambda \leq x \leq 0\} \bigcup \{x = -\frac{\lambda}{2}, -b \leq y \leq b\}, \quad 0 \leq b \leq 2, \tag{12}$$

the neighborhood of the "crux"

$$\{y=0, \|x+c\| \leq \delta\} \bigcup \{x=-c, -b \leq y \leq b\}, \quad 0<\delta \leq c<b, \tag{13}$$

or of two vertical intervals

$$\{x=-c, \delta \leq \|y\| \leq b\}, \quad b>c, \quad 0 \leq \delta. \tag{14}$$

the neighborhood of two the "cruxs"

$$\begin{gathered}\{y=0; -\lambda \leq x \leq 0\} \bigcup \{x=-\frac{\lambda}{2}, -\lambda \leq y \leq \lambda\} \bigcup \\ \{-\frac{\lambda}{2}\Big(1 \pm \exp{(\pm\frac{i\pi}{4})r}\Big), \quad 0 \leq r \leq b \leq \sqrt{2}\}.\end{gathered} \tag{15}$$

These possibilities let us successfully integrate by explicit rules much of unstationary problems of mathematical physics.

Testing of the Code

To test and analyze the code was used the standard testing set [5,6,7]. The problems were divided into six groups by the sort of spectrum (real, complex), and the sort of equations (linear, nonlinear). The results of testing confirmed the perspectivity of using **DUMKA** for integration of stiff systems.

Additionally were solved some problems,not included into the sets [5,6,7], but of interest in investigation of the ability of the code to solve neutron transport problems, and also some problems from [3] (linear heat transfer equation for three - dimensional domain with a large number of nodes, a system of multigroup, one-dimentional cinetic equations with delayed neutrons, Babuska's example, elasticity equation and so on). These results confirmed the assumptions about the method's features and advantages of the code in solution of stiff systems.

References

1. V.I. Lebedev. Explicit difference schemes with time-variable steps for solution of stiff systems of equations. Preprint DNM AS USSR No.177, 1987.

2. V.I. Lebedev. Explicit difference schemes with time-variable steps for solution of stiff systems of equations. Sov. J. Numer. Anal. Math. Modelling. Vol.4, No.2, pp.111–135(1989).

3. V.I. Lebedev. How to solve stiff systems of equations by explicit difference schemes. Numerical methods and applications, Ed. G.I.Marchuk, CRC Press, Boca Raton, Ann Arbor, London, Tokyo, 1994, pp.45–80.

4. V.I. Lebedev. Zolotarev polynomials and extremum problems. Russ. J. Numer. Anal. Math. Modelling, Vol.9, No.3, pp.231–263(1994).

5. W.H. Enright, J.D. Pryce. Two FORTRAN Packages for Assesing Initial Value Methods. ACM Trans. Math. Soft., Vol.13, No.1, pp.1–27(1987).

6. G.D. Byrne, A.C. Hindmarsh. Stiff ODE Solvers:A Review of Current and Coming Attractions. J. of Comp. Physics. Vol.70, pp.1–62(1987).

7. E. Hairer, G. Wanner. Solving Ordinary Differential Equations, Vol.11. Springer-Verlag, 1991.

5.21 Improving the Accuracy of Symplectic Integrators

M.A. López-Marcos, J. M. Sanz-Serna
Departamento de Matemática Aplicada y Computación, Facultad de Ciencias, Universidad de Valladolid, Valladolid, Spain
E-mail: lopezmar@cpd.uva.es
Robert D. Skeel
Department of Computer Science and Beckman Institute, University of Illinois at Urbana-Champaign, 1304 West Springfield Avenue, Urbana, Illinois 61801-2987, U.S.A.
E-mail: skeel@a.cs.uiuc.edu

Introduction

It is widely known that many physical systems from mechanics, optics, etc. are often described by a Hamiltonian system [1]. We restrict our atention to systems with Hamiltonian function of the form

$$H(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^t M^{-1}\mathbf{p} + V(\mathbf{q}),$$

where the potential V is a smooth function and M is a constant, invertible, symmetric matrix. The Hamiltonian system corresponding to this Hamiltonian function is given by

$$\frac{d}{dt}\mathbf{q} = M^{-1}\mathbf{p}, \qquad \frac{d}{dt}\mathbf{p} = -V_{\mathbf{q}}(\mathbf{q}).$$

The notation $V_{\mathbf{q}}$ means the gradient of V with respect to $\mathbf{q}$.

Evidence of investigations suggests that symplectic methods are suitable for the numerical integration of Hamiltonian systems [6]. These methods preserve the symplectic structure of phase space, thus reproducing the main feature of Hamiltonian dynamics.

The recent literature on symplectic integration of Hamiltonian systems is very large. Numerous methods have been derived and analysed. We show how to increase the accuracy of symplectic methods by cheaply processing the results given by the numerical integrator being used [3].

Processing

In the symplectic context the starting point is to consider a numerical method for the integration of the Hamiltonian system. A one-step integrator is given by a transformation $\psi_{h,H}$ that provides the numerical solution corresponding to a time level from the solution at the previous level. When processing is used, there are two sets of variables being considered. On the one hand, the values computed with the method $\psi_{h,H}$, that we denote by capital letters $(\mathbf{Q},\mathbf{P})$. On the other hand, the values represented as $(\mathbf{q},\mathbf{p})$ are connected with the first ones through a transformation $(\mathbf{Q},\mathbf{P}) = \chi_{h,H}(\mathbf{q},\mathbf{p})$. These new variables are seen as the processed numerical approximations to the solutions of the Hamiltonian system.

The processed method requires the following steps:

1. *Preprocessing:* Find, from the initial values $(\mathbf{q}(0),\mathbf{p}(0))$ the starting values for time-stepping $(\mathbf{Q}_0,\mathbf{P}_0) = \chi_{h,H}(\mathbf{q}(0),\mathbf{p}(0))$.
2. *Time-stepping:* Compute $(\mathbf{Q}_{n+1},\mathbf{P}_{n+1}) = \psi_{h,H}(\mathbf{Q}_n,\mathbf{P}_n)$, $n = 0,1,\ldots$
3. *Postprocessing:* If output at time $t = nh$ is desired, then calculate $(\mathbf{q}_n,\mathbf{p}_n) = \chi^{-1}_{h,H}(\mathbf{Q}_n,\mathbf{P}_n)$, wich provides the numerical aproximation to $(\mathbf{q}(nh),\mathbf{p}(nh))$.

The processed solutions can be interpreted as unprocessed solutions computed with the method

$$\hat{\psi}_{h,H} = \chi^{-1}_{h,H} \circ \psi_{h,H} \circ \chi_{h,H}.$$

Processing is interesting if the processed method $\hat{\psi}_{h,H}$ is a more accurate method than $\psi_{h,H}$ and the cost of processing is negligible.

The idea of processing appeared first in [2] where $\psi_{h,H}$ and $\chi_{h,H}$ are Runge-Kutta methods. Butcher shows that $\hat{\psi}_{h,H}$ may possess a higher order than $\psi_{h,H}$; the order of $\hat{\psi}_{h,H}$ is called the effective order of $\psi_{h,H}$.

When $\psi_{h,H}$ and $\chi_{h,H}$ are symplectic transformations the processed integrator is symplectic too, and then provides approximations that solve exactly a perturbed Hamiltonian system. The aim of our processing is, given a symplectic method $\psi_{h,H}$, to find a symplectic transformation $\chi_{h,H}$ so that the perturbed Hamiltonian associated with the processed method is as close as possible to the Hamiltonian that drives the true solution.

Integrators using Hessian-vector products

Processing permits construct integrators with a high effective order. A notable example is the method introduced by Rowlands in which the force in the second order Störmer/leapfrog/Verlet method is modified by the addition of a Hessian-vector product to yield an effective order of four [5].

Methods than incorporate higher derivatives of the Hamiltonian may be seen as the limiting case when abscissas coalesce of methods that evaluate the force at distinct but closely spaced abscissas. These methods are interesting because using a Hessian may introduce savings in cost per step while only increasing slightly the error constants [4].

We determine some processors and illustrate the effect of processing in different known methods, in addition to constructing new integrators using Hessian-vectors products.

References

[1] V.I. Arnold. *Mathematical Methods of Classical Mechanics*, 2nd. edition. Springer, New York, 1989.

[2] J. Butcher. *The effective order of Runge-Kutta methods.* In A. Dold, Z. Heidelberg, and B. Eckmann, eds., *Conference on the Numerical Solution of Differential Equations, Lecture Notes in Mathematics*, volume 109, pages 133-139. Springer-Verlag, New York, 1969.

[3] M.A. López-Marcos, J.M. Sanz-Serna and R.D. Skeel. *Cheap enhancement of symplectic integrators.* In *Proceedings, 1995 Dundee Conference on Numerical Analysis.* To appear.

[4] M.A. López-Marcos, J.M. Sanz-Serna, and R.D. Skeel. *Explicit symplectic integrators using Hessian-vector products.* SIAM J. Sci. Comput. To appear.

[5] G. Rowlands. *A numerical algorithm for Hamiltonian systems.* J. Comput. Phys. 97, 235-239, 1991.

[6] J.M. Sanz-Serna and M.P. Calvo. *Numerical Hamiltonian Problems.* Chapman and Hall, London, 1994.

5.22 GODESS, an ODE-solver framework

Anne Kværnø and Syvert P. Nørsett
UNIT/NTH, Institutt for Numerisk Matematikk, Alfred Getz vei 1, 7034 Trondheim-NTH, Norway
E-mail: norsett@imf.unit.no

For each program for solving an ODE system one normally has to go through all the phases from method development to implementation of a computer code. This approach has often made it a difficult and tedious task to develop new solvers for new numerical methods. Using the GODESS framework, the work of implementing new methods has been simplified significantly. GODESS is an object oriented system, written in C++, for solving initial value problems for ordinary differential equations (ODEs) and differential-algebraic equations (DAEs). The system contains the necessary structures to facilitate the implementation of new numerical methods for these classes of problems, as well as for developing special-purpose solvers and testing methods under true ceteris paribus conditions.

In this talk we will focus on the developement of a set of SDIRK methods. The method construction is more or less classical, we only have to formulate what properties we want our methods to satisfy.

It has been known for some time that SDIRK methods will have problems due to order reduction and poor stability. To reduce these problems, we want our methods to be:

1. Stiffly accurate,
2. with a stiffly accurate error estimating method.
3. $L(\alpha)$ (L-stable if possible),
4. with an $A(\alpha)$ error estimating method.
   (L-stable if possible.)
5. of high stage order.

In our case we choose methods with an explicit first stage, so that the first stage is the value of the last computed step point. The new SDIRK methods are called **ESDIRK**s. We have constructed methods of order 2 to 5. Hence a collection of suitable methods in the SDIRK family is found.

For the code different implementation aspects have to be considered. We will discuss items as

- Continuous version of the method.
- Stopping criteria for the Newton method.
- Starting value for the Newton iteration.

For testing our new methods we start with the classical testset in DETEST. GODESS has been used as testing environment. It turns out that our new methods compare favourably with more established methods in the RADAU family, in particular for low tolerances.

The GODESS program is not only useful for testing purposes, it can also be used to solve real-life problems. To test this capability, the time integrator of a simulator of electrical circuits has been replaced with GODESS. The results, which will be presented in the talk, are very promising.

5.23 Efficient Integration Methods for Many-Particle Problems

Daniel I. Okunbor
Department of Computer Science, University of Missouri-Rolla, Rolla, MO 65401, U.S.A.
E-mail: okunbor@cs.umr.edu

Introduction

This research focuses on the numerical integration of the Newton's equations of motion for N particles in two or three-dimensional space. The Hamiltonian (which in this case corresponds to the energy) of this motion is

$$H = \frac{1}{2} p^{\mathrm{T}} M^{-1} p + U(q), \tag{1}$$

where q is a vector of generalized coordinates, p is a vector of generalized momenta, $U(q)$ is the total potential energy and M is a diagonal matrix of masses. The first term on the right of (1) is the kinetic energy, which is quadratic. The corresponding equations of motion are

$$\dot{q} = M^{-1} p, \; \dot{p} = -\frac{\partial U}{\partial q}(q)$$

Many-body and many-particle systems are encountered in a number of applications, examples are in studies of dynamics of biomolecules, in drug design, in the simulation of plasmas, in the vehicle and space-structure and in astronomy. Computer simulation of particles started approximately thirty years ago. There are basically two techniques for the computer simulations of particles, namely, molecular dynamics (MD) and Monte-Carlo (MC) methods. MD involves the study of the time evolution of the dynamical state of a system of interacting particles or bodies by the computation of trajectories of particles based on the numerical integration of the Newton's equations of motion. Monte-Carlo techniques are based on statistical and sampling procedures and although, they may be suitable for studying equilibrium properties, they certainly do not have the concept of "time" as in molecular dynamics. The surge in computational resources has made molecular dynamics methods more useful than ever. Time averages are calculated over sufficiently long trajectories to provide results equivalent to ensemble averaging over independent systems. Many results obtained using molecular dynamics have been found to be in agreement with experimental results, see for example [8]. Conceptually, the algorithm for the molecular dynamics simulation of particles can be written as

```
**Compute-Solution**($t_i, z_i$){
    $f_i$ := *force*($z_i$);
    $z_{i+1}$ :=*update*($z_i, f_i, t_i$);
}
```

where z_i is vector of solution consisting of positions and momenta at time t_i and z_{i+1} is the solution at time $t_{i+1} = t_i + h_i$. The *update* consists of the integration algorithm being used. Approximately ninety percent of the compute time is spent on the force calculations, whose time complexity is $O(N^2)$ placing a limitation on the size of simulated system. Typically, the number of particles ranges from hundreds to tens of thousands. The breakthrough in parallel computing and a significant number of improved algorithms has recently allowed for the simulation of more than one million particles as reported by D. G. Trublar in the Winter 1994 issue of IEEE Computational Science & Engineering.

We will discuss different strategies to improve the performance of molecular dynamics for large systems. Symplectic integrators recently developed for Hamiltonian system are very promising. Symplectic integrators have been proven to be stable for long time integrations, thus, permitting larger time steps with sufficient accuracies. Since the system of concern have rapidly changing components, there is subtlety in the application of larger time steps which may result in resonances and instabilities. Other advances to improve the performance of MD simulation such as distance cut-off, multiple time-stepping, the hierarchical-based algorithms will be described. Straightforward distance cut-off approach has been known to create discontinuities in the problem thereby giving rise to inaccurate temperature and energies of the system. Multiple time-stepping algorithms are developed to alleviate the problems associated with distance cut-off by partitioning the force functions and applying different timesteps to each partition. As it turns out, there is no systematic way of partitioning the force function and some cases, there are limitations on the how far the force function can be partitioned to avoid instabilities.

Numerical algorithms

In biomolecular modeling, the Newton's equations of motion are commonly solved using the Störmer/Verlet method. This integration method is simple and comes in different, but equivalent forms. We favor the one that reduces round-off error. This method is given by

$$p^I_{n+\frac{1}{2}} = p^I_n - \frac{\Delta t}{2} M^{-1} \frac{\partial U}{\partial q}(q_n), \;\; q^I_{n+1} = q^I_n + \Delta t p^I_{n+\frac{1}{2}}$$

$$p^I_{n+1} = p^I_{n+\frac{1}{2}} - \frac{\Delta t}{2} M^{-1} \frac{\partial U}{\partial q}(q_{n+1})), \quad I = 1, 2, \ldots, N$$

where p^I and q^I are, respectively, the momentum vector and position vector of the I-th atom and Δt is the timestep. Other numerical integrators currently favored for MD simulations are high-order methods due to Beeman (see, e.g., [5]) and Gear (which are Störmer methods in Nordsieck form). What makes the Störmer/Verlet method popular in MD simulations stems from its simplicity, time reversibility, minimal storage requirements and stability. Although, these qualities may be important requirements for integration methods for MD simulations, the reason why the Störmer/Verlet method performs relatively well may be because is is symplectic, a property that makes the method stable during long runs and preserves volume in phase space. These property is not present in high-order Beeman and Gear integration methods. The Störmer/Verlet method is only second-order accurate, a condition that may place an undesirable restriction on the timestep in order to achieve sufficient accuracy. Higher-order methods may allow for large timesteps and still give qualitative behavior that is close to that of the original system. In most MD simulations, in order to obtain high precision in the transport coefficients, it is required to perform simulations for a very long period of time. Symplectic integrators may be appropriate in this regards, since they have been shown to have better long-time dynamics than non-symplectic schemes (see [7, 6, 9, 3, 8, 1, 2]).

Multiple time-stepping

Variable steps implementations of symplectic integrator degrade the symplecticness of the method as reported in the outstanding work of Calvo and Sanz-serna[10]. In biomolecular modeling, what is generally applicable as an alternative for variable steps is the multiple time-stepping. Multiple time steps were proposed in 1978[12], for molecular dynamics. In its simplest form, the MTS technique involves the computation of forces less frequently, that is, making use of current forces for some number of time steps. Force update is done only when the distances between particles satisfy specified criteria. One example of multiple time-stepping methods is the "distance class" algorithm developed by Grubmüller *et el.* [4]. It involves separating the forces into classes. For illustration, let us separate the forces into two classes at time t_n:

$$f = f^{\text{near}} + f^{\text{far}}$$

The far force is evaluated only once for the next two timesteps. Commonly this is done by re-using f^{far}_n at time t_{n+1}. Such a method is not canonical and its order of accuracy is only one. To maintain second-order accuracy, some modifications are done to the discrete equations obtained from the SV. Skeel and Biesiadecki[11] proposed a method for making the distance class algorithm using SV to be symplectic by suitably changing the discrete equations.

References

[1] K. Feng. Difference schemes for Hamiltonian formalism and symplectic geometry. *J. Comput. Math.*, 4(3):279–289, 1986.

[2] K. Feng, H.-M. Wu, and M.-Z. Qin. Symplectic difference schemes for linear Hamiltonian canonical systems. *J. Comput. Math.*, 8(4):371–380, 1990.

[3] B. Gladman and M. Duncan. Symplectic integrators for long-term integrations in celestial mechanics. *Celestial Mech. Dynamical. Astron.*, 52:221–240, 1991.

[4] H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten. Verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Molecular Simulation*, 6:121–142, 1991.

[5] J. A. McCammon and S. C. Harvey. *Dynamics of Proteins and Nucleic Acids.* Cambridge Univ. Press, 1987.

[6] D. Okunbor. Canonical methods for Hamiltonian systems: Numerical experiments. *Physica D*, 60:314–322, 1992.

[7] D. Okunbor and R. D. Skeel. Explicit canonical methods for Hamiltonian systems. *Math. Comp.*, 59(200):439–455, 1992.

[8] A. Rahman. Correlation in the motion of atoms in liquid argon. *Phys. Rev. A*, 136:405–411, 1964.

[9] J. M. Sanz-Serna. Symplectic integrators for hamiltonian problem: an overview. *Acta Numerica*, page 243, 1992.

[10] J. M. Sanz-Serna and M. P. Calvo. *Numerical Hamiltonian Problems.* Chapman & Hall, London, 1994.

[11] R. D. Skeel, J. Biesiadecki, and D. Okunbor. Symplectic integration for macromolecular dynamics. In *Proceedings of the International Conference Computation of Differential Equations and Dynamical Systems*, World Scientific Publishing Co., 1992. To appear.

[12] W. B. Streett, D.J. Tildesley, and G. Saville. Multiple time step methods in molecular dynamics. *Mol. Phys.*, 35:639–648, 1978.

[13] M. Toda. *Nonlinear Waves and Solitons.* Kluwer Academic Publisher, 1989.

5.24 Numerical Integration of ODE's, While Preserving Volume, Symmetries, or First Integrals

G.R.W. Quispel
La Trobe University, Bundoora, Melbourne 3083, Australia
E-mail: mataw@lure.latrobe.edu.au

Three approaches are used to study ordinary differential equations (ODE's): exact analytic solution, geometric methods, and numerical approximations. Most ODE's, however, can not be solved exactly on a computer. Although many ingenious integration schemes have been invented, they can only ever be approximations to the true quantitative behaviour of ODE's. Very recently, however, there has been a lot of research on numerical integration schemes in which certain qualitative features are exactly preserved (while simultaneously quantitative features are not neglected). This is especially relevant in the case of long integration times.

Examples of such qualitative features that can be preserved are: Hamiltonian structure, the presence of symmetries, conservation of phase-space volume, and conservation of first integrals. Preservation of these features is important for the long term stability of dynamical systems, because they ensure the existence of stabilising KAM tori and/or integral surfaces. These features also determine which bifurcations generally occur. As the preservation of Hamiltonian (symplectic) structure is at present fairly well understood, we will concentrate on the three other qualitative features.

The main aim of this talk is to study how volume, first integrals and reversing symmetries can be preserved in numerical simulations.

This is currently of great importance in astronomy (the evolution of the solar system; trajectories of comets and satellites), physics (particle accelerators, statistical mechanics), and computational mathematics.

References

[1] G.R.W. Quispel, Volume-preserving integrators, Phys. Lett. 206A (1995) 26-30.
[2] G.R.W Quispel, R.I. McLachlan and G.S. Turner, Numerical integrators that preserve symmetries and reversing symmetries, preprint.
[3] G.R.W. Quispel and H.W. Capel, The numerical solution of ODE's, while preserving a first integral, in preparation.
[4] G.R.W Quispel and H.W. Capel and C. Scovel, The numerical solutions of ODE's while preserving one or more first integrals, in preparation.

5.25 Molecular Dynamics Simulations and Statistical Mechanics

Sebastian Reich
Konrad-Zuse Center, Heilbronner Str. 10, 10711 Berlin-Willemsdorf, Germany
E-mail: reich@zib-Berlin.de

Over the last 2-3 years, I have been involved in the development of advanced simulation techniques for macromolecular systems. The focus is in particular on time-stepping methods that allow one to simulate the long-time behavior of molecules. There are several obstacles to overcome to achieve this goal. The most important ones are the following three:

- highly oscillatory motions in the bonded interactions such as bond stretching and bond-angle bending,
- frequent occurrence of collisions between atoms due to the Lennard-Jones potentials,
- high energy barriers in the torsion angles.

So far I have been mainly working on the first two problems.

The highly oscillatory motions in the bonded interactions are due to the strong, harmonic potentials in the bond-lengths, bond-angles respectively. In [1], I have shown how to average over those degrees of freedom to obtain equations of motion in the "slow" variables (typically the torsion angles of a molecule). This requires the computation of proper correction terms in the potential energy functions as well as the introduction of proper constraints for the fast bonded interactions (soft constraints). The resulting constrained equations of motion can be discretized by suitable modifications of the SHAKE/RATTLE algorithms. This project as been carried out in close collaboration [2] with Bernie Brooks from the NIH, Bethesda.
Based on this work, a student of mine and I are currently working on a torsion-angle formulation that would allow one to directly compute the torsion-dynamics of macromolecules.

There seem to be various approaches to resolve Lennard-Jones collisions. I have, in particular, developed an implicit energy-momentum method that is based on local modifications of the implicit midpoint rule [3]. The idea is to poorly resolve collisions but to reproduce the dynamics correctly overwise. Since collisions between specific pairs of atoms are rare events in terms of the Boltzmann distribution function, the numerical trajectories will sufficiently well reproduce the statistical behavior of the overall system. In other words, one gives up the idea of approximately following exact trajectories, but instead devices numerical integrators that reproduce the statistical behavior of molecular systems (in thermal equilibrium, for example). This approach requires a completely different error concept then the standard one based on the local discretization error and stability.
Other approaches include regularization techniques and time-reversible variable step-size methods. I am currently collaborating with Ben Leimkuhler on these ideas.

I have also started working on methods that might help overcome the barrier problem in the torsion potentials. For example, assuming that one wants to compute ensemble averages, one can, for example, combine molecular dynamics simulations with techniques related to umbrella sampling in Monte Carlo simulations. These methods are attractive, in particular, for free energy calculations.

Another project of mine, closely related to the ones described above, is the simulation of quantum mechanical systems in thermal equilibrium. This can be done, for example, through discretization of the corresponding Feynmann path integral formulation. Here the idea is to generalize the idea of multiple-time-stepping to the Feynmann path integral. The path integral formulation is attractive because it allows one to combine classical and quantum calculations in the context of molecular dynamics in a straightforward way.

In my lecture I would focus on smoothing techniques and various approaches to resolve the Lennard-Jones collisions. Since some of these techniques (such as smoothing, multiple time-stepping, etc.) are also important tools for the numerical solution of Feynmann path integrals in thermal equilibrium quantum mechanics, I would also briefly discuss the Feynmann path integral formulation and its discretization.

References

[1] S. Reich, Smoothed dynamics of highly oscillatory Hamiltonian systems, *Physica D*, to appear.

[2] B.R. Brooks, S. Reich, J. Zhou, in preparation.

[3] S. Reich, An enhanced energy conserving method, *BIT*, to appear.

5.26 The DAE–Aspect of the Charge Cycle in a Combustion Engine

Th. Neumeyer and P. Rentrop
FB Mathematik, TH Darmstadt, Schloßgartenstr. 7, D-64289 Darmstadt, Germany
E-mail: rentrop@mathematik.th-darmstadt.de

In car industry the performance index of a combustion engine is used as one of the important arguments to convince a customer. In order to raise moments and power of an engine or to diminish the air pollution and fuel consumption, intensive experiments were carried out. Especially the charge cycle can be controlled by gas pressure oscillations in the inlet and outlet pipes, leading to a better performance. Therefore different diameters and lengths of the pipes were tested experimentally. With the extending improvements of the engines, only such an experiment-based strategy was too expensive and less successful. Therefore numerical simulations for possible engine configurations were necessary. This step demands two basic levels: the development of a mathematical model of the charge cycle and the choice of algorithms for the numerical simulation.

In the beginning of the talk a network formulation of an engine is presented. The mathematical model describes the gas flow in the various parts of an engine. A system of ordinary differential equations in time represents conservation of mass and energy in the cylinder. Inside the cylinder and for all connections like nozzles or bifurcations a semi-empirical approach is chosen. The theoretical gas flow values are lowered by experimental gas flow coefficients. In the straight pipes the gas flow is described by the unsteady one-dimensional Euler gas equations.
The numerical algorithms are based on low order multistep schemes inside the cylinder and on ENO schemes for the pipes. The velocity of the gas flow in the pipe is always beyond the sound velocity, but due to opening and closing of the valves, sharp discontinuities for example in the temperature happened. Therefore in car industry the traditionally used Lax-Wendroff scheme produce spurious numerical oscillations. The coupling of Eulers gas equations and the cylinder yields nonlinear equations, for example using characteristic values at the boundaries. After space discretization a differential-algebraic system – shortly DAE – is obtained (in time).

The new approach is demonstrated for a mathematical one cylinder benchmark and for a realistic six cylinder engine. Successful runs on new engine generations are presented.

References

1. G. Engl, P. Rentrop: Gas flow in a single cylinder internal combustion engine. Int. J. Num. Meth. Heat Fluid Flow 2 (1992) 63-78
2. G. Engl, Th. Neumeyer, P. Rentrop: Numerical benchmark for the charge cycle in a combustion engine. J. Appl. Numer. Math. 18 (1995) 293-305
3. K. A. Görg, Th. Neumeyer, P. Rentrop: Zur numerischen Simulation des Ladungswechsels im Verbrennungsmotor. Preprint Nr. 1810 (1996), FB Mathematik, TH Darmstadt
4. A. Harten: ENO schemes with subcell resolution. J. Comp. Phys. 83 (1989) 148-184
5. Ch. Hirsch: Numerical computation of internal and external flows. Vol 1, 2 John Wiley Sons, Chichester (1988)
6. R. J. LeVeque: Numerical methods for conservation laws. Birkhäuser, Basel (1992)

5.27 Long-Time Integration Schemes for Molecular Dynamics

Tamar Schlick
The Howard Hughes Medical Institute and New York University, Chemistry Department and Courant Institute of Mathematical Sciences, 251 Mercer Street, New York, New York 10012, U.S.A.
E-mail: schlick@nyu.edu

Many important problems in chemistry are solved by numerical integration of very large coupled ODE systems. Two examples are molecular (or Langevin) dynamics simulations of biomolecules [1] and solution of the time-dependent Schrödinger equation when expanded in a basis set. The technology of numerical integrators for solving ODEs has a long history with significant interplay between mathematics, physics and chemistry. Many of the earliest integrators, such as Runge-Kutta and predictor-corrector integrators, are still in common use [2], but there have also been recent advances, such as symplectic integrators [3,4], driven in part by the need for methods that can treat multiple-timescale systems and have greater stability for the large-scale coupled nonlinear oscillators commonly found in molecular dynamics (MD) of polymers and biological macromolecules. In applications to Hamiltonian systems, symplectic integrators have the property of building in Liouville's theorem, whereby areas in phase space are preserved as the system evolves in time. This strong conservation property translates into stability over long-time integrations, an important property in MD calculations involving millions and more steps. One consequence of this for constant-energy MD simulations is that except for fluctuations, symplectic integrators at small timesteps conserve energy for very long times; non-symplectic integrators, in contrast, typically introduce a systematic drift in the total energy.

Symplectic integrators may be *implicit* or *explicit.* Most attention has focused on explicit methods for MD, since they generally involve simple propagation formulae that use modest memory and computation. Implicit methods, in contrast, are more complex in algorithmic structure but may be more powerful for treating systems with disparate timescale dynamics. Indeed, the commonly explicit, symplectic Verlet method for MD [5] imposes a severe constraint on the maximum timestep possible: less than about 1 fs = 10^{-15} s. This step size is determined by the period associated with high frequency modes present in all macromolecules, and it contrasts with the much longer timescales (up to 10^2 s) which govern key conformational changes (e.g., folding) in macromolecules. This disparity in timescales urges development of new methods which increase the feasible timestep

Standard techniques of effectively freezing the fast vibrational modes by a constrained formulation [6] increase the timestep by a small factor such as two, still with added complexity at each step. The multiple-timestep approaches for updating the slow and fast forces provide additional speedup [7], though some stability issues are also involved [8].

There are well known numerical techniques for solving differential equations describing physical processes with multiple timescales [9]. Various implicit formulations are available that balance stability, accuracy, and complexity. However, the standard implicit techniques used by numerical analysts [10] have not been directly applicable to MD simulations of macromolecules, for the following reasons.

First, such implicit schemes are often designed to suppress the rapidly-decaying component of the motion. This is a valid approach when the contribution of these components becomes negligible for sufficiently long times. This situation does not hold for biomolecular systems because of the intricate vibrational coupling. It is well recognized that concerted conformational transitions (e.g., in hinge-bending proteins) require a *cooperative* mechanism driven by small-scale fluctuations to make possible a large-scale collective displacement. Thus, while the damping of the high-frequency modes may not by itself be a severe problem, the lower *energies* associated with these modes may be undesirable for proteins and nucleic acids, as cooperative motions among the correlated vibrational modes may require energy transfer.

Second, implicit schemes with known high-stability (e.g., implicit-Euler) can introduce numerical damping [11]. Therefore, such discretizations of the classical Newton equations of motion will eventually damp out the energy of a system. This has prompted the application of such implicit schemes to the *Langevin* dynamics formulation, which involves frictional and Gaussian random forces in addition to the systematic force to mimic molecular collisions and therefore a thermal reservoir. This stabilizes implicit discretizations and can be used to quench higher-frequency vibrational modes [12], but unphysical increased rigidity can result [11]. Therefore, more rigorous approaches are required to resolve the subdynamics correctly, such as by combining normal-mode techniques with implicit integration [11]; significant linear-algebra work in the spectral decomposition is necessary so that these methods become feasible for macromolecular systems. There has also been some work on implicit schemes that do not have inherent damping, but preliminary experience suggests that for nonlinear systems desirable energy conservation properties can only be obtained up to moderate timesteps [13]. Resonance problems, in particular [14], are a limitation.

Third, implicit schemes increase complexity, since they involve solution of a nonlinear system or minimization of a nonlinear function at each timestep. Therefore, very efficient implementations of these additional computations are necessary and, even then, computational gain (with respect to standard "brute-force" integrations at small timesteps) can be realized only at very large timesteps.

Thus, it appears that the optimal algorithms for MD will require a combination of methods and strategies discussed above, including symplectic and implicit numerical integration schemes that have minimal intrinsic damping, and correct resolution of the subdynamics of the system by some other technique (e.g., normal-mode analysis). In tandem, development of an appropriate theoretical framework for evaluation of error in MD trajectories [15] will become essential. Undoubtedly, high-performance implementations will make possible a gain of several orders of magnitude in the simulation times, and there are certainly additional gains to be achieved by clever programming strategies.

References

[1] J.A. McCammon and S.C. Harvey, *Dynamics of Proteins and Nucleic Acids*, Cambridge University Press, Cambridge (1987).

M.P. Allen and D.J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, New York (1990).

[2] A.M. Stuart and A.R. Humphries, "Model Problems in Numerical Stability Theory for Initial Value Problems", *SIAM Review* **36**, 226 (1994).

[3] R.D. Ruth, "A Canonical Integration Technique", *IEEE Trans. Nucl. Sci.* **NS-30**, 2669 (1983).

K. Feng, "Difference Schemes for Hamiltonian Formalism and Symplectic Geometry", *J. Comput. Math.* **4**, 279 (1986).

J. Candy and W. Rozmus, "A Symplectic Integration Algorithm for Separable Hamiltonian Functions", *J. Comp. Phys.* **92**, 230 (1991).

R.I. McLachlan and P. Atela, "The Accuracy of Symplectic Integrators", *Nonlinearity* **5**, 541 (1992).

M.P. Calvo and J.M. Sanz-Serna, "The Development of Variable-step Symplectic Integrators, with Application to the Two-Body Problem", *SIAM J. Sci. Comput.* **14**, 936 (1993).

D.I. Okunbor and R.D. Skeel, "Canonical Numerical Methods for Molecular Dynamics Simulations", *Math. Com.* **59**, 439 (1992).

G.J. Martyna and M.E. Tuckerman, "Symplectic Reversible Integrators: Predictor-Corrector Methods", *J. Chem. Phys.* **102**, 8071 (1995).

[4] P.J. Channell and J.C. Scovel, "Symplectic Integration of Hamiltonian Systems", *Nonlinearity* **3**, 231 (1990).

J. De Frutos and J.M. Sanz-Serna, *An Easily Implementable Fourth-Order Method for the Time Integration of Wave Problems, J. Comput. Phys.* **103**, 160 (1992).

[5] L. Verlet, "Computer 'Experiments' on Classical Fluids. i. Thermodynamical Properties of Lennard-Jones Molecules", *Phys. Rev.* **159**, 98 (1967).

[6] W.F. van Gunsteren, "Constrained Dynamics of Flexible Molecules", *Mol. Phys.* **40**, 1015 (1980).

W.F. van Gunsteren and H.J.C. Berendsen, "Algorithms for Macromolecular Dynamics and Constraint Dynamics", *Mol. Phys.* **34**, 1311 (1977).

J.P. Ryckaert, G. Ciccotti, and H.J.C. Berendsen, "Numerical Integration of the cartesian Equations of Motion of a System with Constraints: Molecular Dynamics of n-alkanes", *J. Comp. Phys.* **23**, 327 (1977).

S. Miyamoto and P.A. Kollman, "SETTLE: An Analytical Version of the SHAKE and RATTLE Algorithm for Rigid Water Models", *J. Comp. Chem.* **13**, 952 (1992).

[7] W.B. Streett, D.J. Tildesley and G. Saville, "Multiple Time Step Methods in Molecular Dynamics", *Mol. Phys.* **35**, 639 (1978).

M.E. Tuckerman and B.J. Berne, "Molecular Dynamics in Systems with Multiple Time Scales: Systems with Stiff and Soft Degrees of Freedom and with Short and Long Range Forces", *J. Comp. Chem.* **95**, 8362 (1992).

D.E. Humphreys, R.A. Friesner, and B.J. Berne, "A Multiple-Time-Step Molecular Dynamics Algorithm for Macromolecules", *J. Phys. Chem.* **98**, 6885 (1994).

H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten, "Generalized Verlet Algorithm for Efficient Molecular Dynamics Simulations with Long-Range Interactions", *Mol. Sim.* **6**, 121 (1991).

M. Watanabe and M. Karplus, "Dynamics of Molecules With Internal Degrees of Freedom by Multiple Time-Step Methods", *J. Chem. Phys.* **99**, 8063 (1993).

[8] J.J. Biesiadecki and R.D. Skeel, "Dangers of Multiple-time-step Methods", *J. Comp. Phys.* **109**, 318 (1993).

[9] C.W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Englewood Cliffs, New Jersey (1971).

G. Dahlquist and Å. Björck, *Numerical Methods*, Prentice Hall, New Jersey (1974).

[10] H.-O. Kreiss, "Problems with Different Time Scales", *Acta Numerica* **1**, 101 (1991).

[11] G. Zhang and T. Schlick, "LIN: A New Algorithm to Simulate the Dynamics of Biomolecules by Combining Implicit-Integration and Normal Mode Techniques", *J. Comp. Chem.* **14**, 1212 (1993).

[12] C.S. Peskin and T. Schlick, "Molecular Dynamics by the Backward-Euler Method", *Comm. Pure App. Math.* **42**, 1001 (1989).

T. Schlick and C.S. Peskin, "Can Classical Equations Simulate Quantum-mechanical Behavior? A Molecular Dynamics Investigation of a Diatomic Molecule with a Morse Potential", *Comm. Pure App. Math* **42**, 1141 (1989).

[13] J.C. Simo, N. Tarnow, and K.K Wong, "Exact Energy-Momentum Conserving Algorithms and Symplectic Schemes for Nonlinear Dynamics", *Computer Methods in Applied Mechanics and Engineering* **100**, 63 (1992).

G. Zhang and T. Schlick, "Implicit Discretization Schemes for Langevin Dynamics", *Mol. Phys.* **84**, 1077 (1995).

[14] M. Mandziuk and T. Schlick, "Resonance in the Dynamics of Chemical Systems Simulated by the Implicit-Midpoint Scheme", *Chem. Phys. Lett.* **237**, 525 (1995).

[15] B. Mishra and T. Schlick, "Error in Numerical Integration of the Langevin Equation: 1. Linear Analysis for Five Explicit and Implicit Schemes", sumbitted August 1995 to *J. Chem. Phys.*

5.28 Krylov-W-Methods for Large Stiff Equations

B.A. Schmitt (Marburg) and R. Weiner
FB Mathematik und Informatik, Martin-Luther-Universität, Theodor-Lieser-Str. 5, 06120 Halle (Saale), Germany
E-mail: rw@numsun1.mathematik.uni-halle.de

For the numerical solution of stiff initial value problems

$$\begin{aligned} y'(t) &= f(t, y(t)) \\ y(t_0) &= y_0 \in \mathcal{R}^n, \end{aligned} \tag{1}$$

due to stability requirements implicit or linearly implicit methods have to be used. In practical computations linearly-implicit Runge-Kutta methods have shown good performance. They include ROW- and W-methods ([8], [10]) and adaptive Runge-Kutta methods ([11]).

The nonautonomous form of an s-stage ROW-method is given by [6], [12]

$$\begin{aligned} (I - h\gamma A)(k_i + \sum_{j=1}^{i-1} \frac{\gamma_{ij}}{\gamma} k_j) &= f_i + \sum_{j=1}^{i-1} \frac{\gamma_{ij}}{\gamma} k_j + hd_i f_t =: w_i, \quad i = 2, \ldots, s \\ f_i &= f(t_m + c_i h, u_m + h \sum_{j=1}^{i-1} a_{ij} k_j) \\ u_{m+1} &= u_m + h \sum_{j=1}^{s} b_i k_i \end{aligned} \tag{2}$$

with

$$f_t = f_t(t_m, u_m), \quad d_i = \gamma + \sum_{j=1}^{i-1} \gamma_{ij}, \quad c_i = \sum_{j=1}^{i-1} a_{ij}, \quad A = f_y(t_m, u_m).$$

W-methods do not use the partial derivative f_t and the Jacobian A is replaced by an arbitrary matrix.

For large systems, arising for instance by application of the method of lines to 2- or 3-dimensional parabolic equations, these methods spend most of their computing time for the evaluation of the Jacobian and for the solution of the linear equations.

Krylov-W-methods reduce this effort per step by replacing the matrix $A = f_y(t_m, u_m)$ in (2) at the i-th stage by the low rank-approximation

$$T_i = Q_i Q_i^T A, \tag{3}$$

where $Q_i = \{q_1, ..., q_{\kappa_i}\} \in \mathcal{R}^{n,\kappa_i}$ is an orthogonal matrix computed by an Arnoldi process. Note that the exact computation of the Jacobian is not necessary, since it occurs in the form of matrix-vector products Av only, which can be approximated by the difference quotient

$$f_y(t_m, u_m)v = \frac{1}{\delta}(f(t_m, u_m + \delta v) - f(t_m, u_m)) + \mathcal{O}(\delta \|v\|^2).$$

Due to this feature these methods are often called "matrix-free".

Remark

Krylov techniques have been used also in multistep codes for the solution of the arising nonlinear systems of algebraic equations, e.g. Gear and Saad [5], Brown and Hindmarsh [1], Byrne [4]. Recently another generalization of W-methods was proposed by Hochbruck and Lubich [7].

With (3) the formula for the computation of k_i becomes

$$\begin{aligned} k_i &= (I - h\gamma T_i)^{-1} w_i - \sum_{j=1}^{i-1} \frac{\gamma_{ij}}{\gamma} k_j \\ &= [I + h\gamma Q_i (I - h\gamma H_i)^{-1} Q_i^T A] w_i - \sum_{j=1}^{i-1} \frac{\gamma_{ij}}{\gamma} k_j, \end{aligned} \tag{4}$$

with $H_i = Q_i^T A Q_i$. So we have to solve equations of dimension κ_i only.

There are different ways in applying the Arnoldi-method:

(A) The stages are treated separately and the iteration is started for every system anew.

(B) The information, i.e. the matrices Q_i, from lower stages are exploited at higher stages, the Arnoldi process is continued in some appropriate way.

We concentrate on the second approach. A first attempt for general linearly-implicit Runge-Kutta methods is dicussed in [2], for W-methods in [3]. There the Arnoldi process from the first stage is simply continued, we have

$$\operatorname{span}\{q_1, \ldots, q_{\kappa_i}\} = \operatorname{span}\{q_1, \ldots, A^{\kappa_i - 1} q_1\}. \tag{5}$$

This implies that H_i is of Hessenberg form. For dissipative linear autonomous systems

$$y' = Ay + g$$

in [2], [3] contractivity results in an energy norm were obtained. Estimates for the required Krylov dimensions κ_i, ensuring that the norm of the difference of the solution of the Krylov-W-method and of the solution with $A = f_y$ is below some prescribed tolerance, were given. Although in numerical tests this method performs surprisingly well for different test problemed, it has the drawback that w_i, $i \geq 2$, is in general no longer contained in the Krylov subspace. This can lead to high Krylov dimensions for some problems. Furthermore in the residuals arise terms of the form $(I - Q_i Q_i^T) w_i$ which are difficult to handle in theoretical investigations. Also the construction of methods of order $p > 3$ is rather difficult.

We will present here another approach of continuing the Arnoldi process. First theoretical and numerical results for such a *multiple Arnoldi-process* were obtained by Schmitt and Weiner [9]. It re-uses information from previous stages but avoids dubious error contributions by incorporating the new right-hand sides w_i into the subspace range Q_i. In this case (5) simplifies to

$$k_i = Q_i (I - h\gamma H_i)^{-1} Q_i^T w_i - \sum_{j=1}^{i-1} \frac{\gamma_{ij}}{\gamma} k_j, \tag{6}$$

and a large part of the computations can be performed with vectors of dimension κ_i only.

We will present a special multiple Arnoldi process, which guarantees the Krylov-W-method to be of the same consistency order p as the underlying method with $A = f_y(t_m, u_m)$. This can be achieved with relatively small Krylov dimensions. We give bounds for the required Krylov dimensions κ_i which are independent of the dimension of the differential system n. We consider various examples, e.g. Krylov-W-methods for the widely used ROW-methods with $s = p = 4$. In contrast to the simple continuation (5) the Hessenberg form of the matrices H_i is lost. However, they are of generalized Hessenberg structure, with a few additional subdiagonals.

Furthermore we present stability investigations for dissipative semi-linear problems $y' = Ay + g(t, y)$ and give numerical results for several 2-dimensional parabolic problems. They illustrate that the Krylov-W-method with multiple Arnoldi process is much more efficient than solving the linear systems at each stage with a suitable iteration method, e.g. with BiCGSTAB. On the other hand we compare our method w ith the multistep Krylov-code VODPK [4], which uses a variant of GMRES. It turns out that the considered Krylov-W-method with multiple Arnoldi process is competitive for moderate tolerances.

References

[1] P.N. Brown and A.C. Hindmarsh, *Matrix-free methods for stiff systems of ODEs*, SIAM J. Numer. Anal. **23** (1986), 610–638.

[2] M. Büttner, B.A. Schmitt, and R. Weiner, *Automatic partitioning in linearly-implicit Runge-Kutta methods*, Applied Numerical Mathematics **13** (1993), 41–55.

[3] ———, *W-methods with automatic partitioning by Krylov techniques for large stiff systems*, SIAM J. Numer. Anal. **32** (1995), 260–284.

[4] G.D. Byrne, *Pragmatic experiments with Krylov methods in the stiff ODE setting*, Computational ordinary differential equations, Clarendon Press, Oxford, 1992, pp. 323–356.

[5] C.W. Gear and Y. Saad, *Iterative solution of linear equations in ODE codes*, SIAM J. Sci. Stat. Comp. **4** (1983), 583–601.

[6] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, Springer-Verlag, 1991.

[7] M. Hochbruck and Ch. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, Preprint no. 5, Mathematisches Institut Tübingen, 1995.

[8] P. Kaps and P. Rentrop, *Generalized Runge-Kutta methods of order four with stepsize control for stiff ordinary differential equations*, Numer. Math. **38** (1979), 55–68.

[9] B.A. Schmitt and R. Weiner, *Matrix-free W-methods using a multiple Arnoldi iteration*, (1995), to appear in APNUM.

[10] T. Steihaug and A. Wolfbrandt, *An attempt to avoid exact Jacobian and nonlinear equations in the numerical solution of stiff ordinary differential equations*, Math. Comp. **33** (1979), 521–534.

[11] K. Strehmel and R. Weiner, *Behandlung steifer Anfangswertprobleme gewöhnlicher Differentialgleichungen mit adaptiven Runge-Kutta-Methoden*, Computing **29** (1982), 153–165.

[12] ———, *Linear-implizite Runge-Kutta Methoden und ihre Anwendung*, Teubner-Verlag Stuttgart-Leipzig, 1992.

5.29 Cheap Implicit Symplectic Integrators

Robert D. Skeel [2]
Department of Computer Science, (and Beckman Institute) University of Illinois at Urbana-Champaign,
1304 West Springfield Avenue, Urbana, Illinois 61801-2987, U.S.A
E-mail: skeel@a.cs.uiuc.edu

The high frequency modes of Hamiltonian systems tend to have small amplitudes. Hence for moderately accurate integration of such problems, by say, the leapfrog method, the time step tends to be limited by stability restrictions rather than accuracy restrictions. Conventional implicit methods, on the other hand, have less severe stability restrictions if any at all but the cost of solving large nonlinear systems with dense Jacobian matrices is probably too high to make them worthwhile. To bring down the cost of implicit methods, we propose (i) linearly implicit and (ii) mixed implicit-explicit methods that retain the property of being symplectic. Motivation for this work is its application to molecular dynamics (MD) although the ideas may be more suitable for other applications like structural dynamics and wave propagation, which are not so nonlinear as MD.

We begin by describing a family of simple implicit methods of conventional type for the system

$$\dot{x} = v, \quad \dot{v} = F(x)$$

where $F(x) = -U_x(x)$ and $U(x)$ is potential energy [2]. At the beginning of a typical step we have available the collective positions X_n, the collective velocities V_n, and a collective force vector F_n. The last of these has been obtained by solving

$$F_n = F(X_n + \alpha\Delta t^2 F_n)$$

where α is the parameter for the family of methods. We advance one step as follows:

$$\begin{aligned}
V_{n+1/2} &= V_n + \frac{\Delta t}{2} F_n, \\
X_{n+1} &= X_n + \Delta t V_{n+1/2}, \\
F_{n+1} &= F(X_{n+1} + \alpha\Delta t^2 F_{n+1}) \\
V_{n+1} &= V_{n+1/2} + \frac{\Delta t}{2} F_{n+1}.
\end{aligned}$$

The case $\alpha = 0$ is the computationally inexpensive leapfrog/Verlet method. The other choices of α can be viewed as a modification of the force vector which attempts to compensate for the finite size of Δt. That is, we use a force $F_{\Delta t}(X)$ which is defined implicitly by

$$F_{\Delta t} = F(X + \alpha(\Delta t)^2 F_{\Delta t}).$$

What is needed for symplecticness is merely that the Jacobian matrix of $F_{\Delta t}(X)$ be symmetric, which it is. The value $\alpha = \frac{1}{4}$, which yields essentially the implicit midpoint scheme, is special in that it guarantees that the total energy is exactly conserved for linear problems. (Also $\frac{1}{4}$ is the smallest value of α that guarantees unconditional stability for any linear problem.)

This way of viewing the method suggests modifications that can significantly reduce the cost. The idea is to seek finite-Δt forces $F_{\Delta t}(X)$ which are cheap to compute and have a symmetric Jacobian matrix.

For example, we can write

$$F = F^1 + F^2 + \cdots + F^m$$

where the splitting of collective force vectors is induced by a splitting of the potential energy function. Then define

$$F_{\Delta t} = F^1_{\Delta t} + F^2_{\Delta t} + \cdots + F^m_{\Delta t}$$

[2] This work is joint with Meiqing Zhang.

where

$$F_{\Delta t}^k = F^k(q + \alpha_k(\Delta t)^2 F_{\Delta t}^k).$$

A fairly obvious scheme would be handle the "fast" forces as a single term in the splitting, with say $\alpha = \frac{1}{4}$, and the "slow" forces explicitly. This idea of mixing implicit and explicit differencing is used effectively in partial differential equations, such as reaction–diffusion equations. This splitting is, in principle, straightforward for MD if the fast forces are chosen to include only bonded forces. However, nonbonded forces due to nearby atoms can be strong, if only temporarily, and full benefit of implicitness requires that these forces be treated implicitly. One way to do this is by a splitting of the nonbonded interaction into a slow long-range part and a fast short-range part. The short-range part is chosen to vanish beyond a certain cutoff and it is treated implicitly. For an example of such a splitting see [1].

To solve the implicit equations of such a method requires an iteration, which is costly even if only some forces F^1 are handled implicitly. If we do only one Newton iteration for forces F^1, the method would be much cheaper but we would sacrifice symplecticness. It would be desirable to have a symplectic linearly implicit method which reduces to some given implicit method for linear forces. Such a method is obtained by defining $F_{\Delta t}^1$ as the negative gradient of

$$U^1 - \frac{1}{2}\alpha_1(\Delta t)^2(U_x^1)^{\mathrm{T}}(I + \alpha_1(\Delta t)^2 U_{xx}^1)^{-1}U_x^1.$$

When differentiated this yields one Newton iteration of the original implicit method plus a "correction term" involving the third derivative of U^1, which is small for the mildly nonlinear bonded forces but would be large for the highly nonlinear van der Waals forces. If the third derivative term is omitted from the force, the method is is still time reversible even though it is not symplectic, so in practice it may work as well.

References

[1] R. D. Skeel and J. J. Biesiadecki. Symplectic integration with variable stepsize. *Annals of Numer. Math.*, 1:191–198, 1994.

[2] R. D. Skeel, G. Zhang, and T. Schlick. A family of symplectic integrators: stability, accuracy, and molecular dynamics applications. *SIAM J. Sci. Comput.*, 18, 1997, to appear.

5.30 Vector/parallel time integration methods for 3D bio-chemical transport in shallow water

B.P. Sommeijer, P.J. van der Houwen and J. Kok
CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
E-mail: bsom@cwi.nl

The mathematical model describing transport processes of salinity, pollutants, etc. in water, combined with their bio-chemical interactions, is defined by an initial-boundary value problem for the system of 3D advection-diffusion-reaction equations

$$\frac{\partial c_i}{\partial t} + \frac{\partial}{\partial x}(uc_i) + \frac{\partial}{\partial y}(vc_i) + \frac{\partial}{\partial z}(wc_i) = \frac{\partial}{\partial x}\left(\varepsilon_x \frac{\partial c_i}{\partial x}\right) + \frac{\partial}{\partial y}\left(\varepsilon_y \frac{\partial c_i}{\partial y}\right) + \frac{\partial}{\partial z}\left(\varepsilon_z \frac{\partial c_i}{\partial z}\right) + \quad (1)$$
$$g_i(t, x, y, z, c_1, \ldots, c_m), i = 1, \ldots, m,$$

where c_i denotes the unknown concentrations of the contaminants. The local fluid velocities u, v, w (to be provided by a hydrodynamical model) and the diffusion coefficients $\varepsilon_x, \varepsilon_y, \varepsilon_z$ are assumed to be given functions. The equations in (1) are mutually coupled by means of the functions g_i, which model the (concentration-dependent) bio-chemical reactions and emissions from sources. The definition of the physical domain and of the initial and boundary conditions completes the model.

Following the Method of Lines approach, equation (1), together with the initial condition and the boundary conditions, is converted into the semi-discrete initial value problem

$$\frac{d\mathbf{C}(t)}{dt} = \mathbf{F}(t, \mathbf{C}(t)) := \mathbf{H}(t, \mathbf{C}(t)) + \mathbf{G}(t, \mathbf{C}(t)), \mathbf{C}(t_0) = \mathbf{C}_0. \quad (2)$$

Here, $\mathbf{C}$ is a vector of dimension mN containing the m concentrations c_i at the total number of $N := N_x.N_y.N_z$ grid points (N_x, N_y, and N_z denote the number of grid points in the various spatial directions, respectively). The term $\mathbf{H}(t, \mathbf{C}(t))$ originates from the discretization of the advection-diffusion terms (including the boundary conditions), and $\mathbf{G}(t, \mathbf{C}(t))$ is the discrete analogue of the reaction terms and emissions. Finally, $\mathbf{C}_0$ contains the initial values.

Since the functions $\mathbf{H}$ and $\mathbf{G}$ have quite a different origin, they give rise to a completely different coupling of the unknowns: in $\mathbf{H}$ the concentrations of the various species are uncoupled, but there is of course a coupling in space, due to the underlying spatial differential operators. In $\mathbf{G}$, on the other hand, we have in each grid point a local coupling of the concentrations. Another observation is that $\mathbf{H}$ is linear in $\mathbf{C}$, whereas $\mathbf{G}$ is usually nonlinear.
These observations should be taken into account in selecting a suitable time integration method. In this context, 'suitable' means that the method should have the following properties:

- *sufficient stability*; in the present application, we are primarily concerned with transport in shallow seas, resulting in small values for the mesh size in the vertical direction. As a consequence, stiffness is introduced into the discrete system (2). This observation excludes the use of fully explicit methods, since the stability requirements would force the method to take unrealistically small time steps. One possibility to avoid these stability problems is to select a fully implicit method. However, the different nature of $\mathbf{H}$ and $\mathbf{G}$, as well as the fact that we are dealing with three spatial dimensions, result in a complicated coupling in the right-hand side function $\mathbf{F}$, and hence in the corresponding Jacobian. As a result, the linear algebra problem to solve the implicit relations is extremely large. This aspect leads us to the second requirement:

- *manageable level of computational effort*; based on the above observation, we strive for a reduction of the amount of implicitness, while maintaining sufficient stability. Especially, the coupling in the systems that have to be solved should be modest.
  For this item it is also relevant to mention that good vectorization and parallelization properties are indispensable to reduce the computation time.

- *realistic accuracy*; in this PDE context, high precision results (e.g., produced by high order methods) are usually not necessary. On the other hand, since predictions over long time periods are an essential part in these kind of simulations, first order accuracy is, in our opinion, too low. Therefore, we restrict our attention to methods that are second order in time.

- *storage economy*; although present-day computers are equipped with large memories, the nature of flow problems, especially in *three* dimensions, still necessitates a careful selection of an algorithm with respect to its storage requirements. A situation in which we are dealing with $N = 10^6$ grid points and $m = 10$ or 20 species, is certainly not unusual.
- *domain decomposition*; in many practical situations, different resolutions in space will be required in various regions of the domain. For example, near the coasts and in estuaries a fine grid is unavoidable to capture the physical phenomena. A natural way to efficiently cope with this demand is to apply a domain decomposition approach, in which the various subdomains are discretized with an appropriate resolution. Then the (sub)problems on the various subdomains can be solved in parallel. However, to obtain an efficient process for the overall solution, the coupling of these subproblems should not be too tight, since in that case many iterations would be necessary to match the interface conditions on the boundaries of these subdomains. Therefore, we are aiming at methods that are 'loosely coupled in the horizontal direction'.

To cope with these requirements, a splitting method could be a possibility. In [1,3] several splitting methods have been designed for this special application.
Another option is to start with an implicit ODE method (e.g., a BDF2), the solution of which has to be found iteratively. At CWI, a special iteration scheme has been designed in order to reduce the amount of linear algebra work involved [2].

In this presentation we will discuss the advantages and disadvantages of these approaches. Furthermore, we will present performance results obtained on a multi-processor vector machine.

1. P.J. van der Houwen and B.P. Sommeijer, *Splitting methods for three-dimensional transport models with interaction terms*, CWI Report NM-R9516.
2. P.J. van der Houwen, B.P. Sommeijer and J. Kok, *The iterative solution of implicit discretizations of three-dimensional transport models*, CWI Report NM-R96XX.
3. B.P. Sommeijer and J. Kok, *Splitting methods for three-dimensional bio-chemical transport*, CWI Report NM-R9607.

5.31 Linear Stability Analysis for Discretizations of Initial Value Problems

M.N. Spijker
Dept. of Mathematics and Computer Science, Leiden University, P.O. Box 9512, 2300 RA Leiden, The Netherlands
E-mail: spijker@wi.leidenuniv.nl

This talk deals with the theoretical analysis of numerical methods for evolutionary problems. A crucial question in the step–by–step solution of such problems is whether the method will behave *stably* or not. Here we use the term stable to designate that any numerical error, introduced at some stage of the calculations, is propagated in a mild fashion – i.e. does not blow up in the subsequent steps of the method.

In the stability analysis of the numerical methods one is often faced with the problem of estimating the norm of the n-th powers of given $s \times s$ matrices B. Stable numerical processes may be distinguished by the property that the norm of B^n – as a function of n – does not grow at a faster rate than some power of n, say n^α. The case $\alpha = 0$ corresponds to what sometimes is called *strong stability*, and $\alpha > 0$ to *weak stability.*

A classical way to assess the growth of B^n, a-priori, consists in an inspection of the eigenvalues of B. One might e.g. require that the modulus of all eigenvalues of B be less than 1, in order that (strong) stability is present. However, this *eigenvalue condition* is notorious for being deceptive – for matrices B which fail to be normal.

The purpose of the talk is to review (recent) reliable conditions on B guaranteeing that the norm of B^n behaves actually as is required for stability. The conditions involve the so-called *resolvent* $(\zeta I - B)^{-1}$ of B. Here ζ is a complex variable, and I denotes the $s \times s$ identity matrix. In formulating these resolvent conditions we denote by $\|\cdot\|$ a matrixnorm which is induced by a given vectornorm $|\cdot|$ on $\mathbf{C}^s$. Further, W stands for a subset of the *unit disk* $D = \{\zeta : \zeta \in \mathbf{C} \text{ with } |\zeta| \leq 1\}$, and $\mathrm{d}(\zeta, W) = \inf\{|\zeta - \xi| : \xi \in W\}$ denotes the *distance* from $\zeta \in \mathbf{C}$ to W.

Our conditions on the resolvent of B require that a constant L exists such that

(a) $$\zeta I - B \text{ is invertible, and } \|(\zeta I - B)^{-1}\| \leq L \, \mathrm{d}(\zeta, W)^{-1} \text{ for all } \zeta \in \mathbf{C} \backslash W.$$

We refer to this as the *Kreiss resolvent condition with respect to W, with constant L.* This terminology is used because one of the equivalent statements in the reputed Kreiss matrix theorem is of the form (a) with W equal to the unit disk D.

Under condition (a), stability estimates of the form

(b) $$\|B^n\| \leq \gamma \cdot L \cdot n^\alpha \qquad (s \geq 1, n \geq 1)$$

and

(c) $$\|B^n\| \leq \gamma \cdot L \cdot s^\beta \qquad (s \geq 1, n \geq 1)$$

are discussed, following Spijker & Straetemans (1996). Here $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$.

The above, general stability considerations are illustrated in the numerical solution of a diffusion-convection-reaction problem.

Reference

Spijker M.N., Straetemans F.A.J. (1996): 'Stability estimates for families of matrices of nonuniformly bounded order'. To appear in Lin. Alg. Appl.

5.32 Linear partial differential-algebraic equations and their indices

W. Lucht, K. Strehmel and C. Liebenow (Halle)
Institut für Numerische Mathematik, FB Mathematik und Informatik, Martin-Luther-Universität Halle-Wittenberg, 06099 Halle (Saale), Germany
E-mail: strehmel@mathematik.uni-halle.d400.de

The differential index ([3]) and the perturbation index ([4], [5]) play a fundamental role in the analysis and development of numerical methods for differential-algebraic equations (DAEs).

In this lecture we extent the concept of both indices to partial differential-algebraic equations (constrained partial differential equations). We show that the situation is more complex for such systems.

We consider linear partial differential-algebraic equations (PDAEs) of second order with constant coefficients

$$Au_t(t,x) + Bu_{xx}(t,x) + Cu(t,x) = f(t,x) \tag{1}$$

with $t \in (t_0, t_e)$, $x \in (0, L)$, $A, B, C \in \mathbf{R}^{n\times n}$, $u, f : [t_0, t_e] \times [0, L] \to \mathbf{R}^n$. The matrices A and B are assumed to be positive semidefinit and negative semidefinit, respectively. Furthermore there are given homogeneous boundary conditions

$$\begin{aligned} R_1u(t,x) &:= \alpha_1 u(t,0) - \beta_1 u_x(t,0) = 0 \\ R_2u(t,x) &:= \alpha_2 u_(t;L) + \beta_2 u_x(t,L) = 0 \end{aligned}$$

with $\alpha_i, \beta_i \in \mathbf{R}$, α_i, $\beta_i \geq 0, \alpha_i + \beta_i > 0$ and the initial condition

$$u(t_0, x) = g(x).$$

Suppose that the compatibility condition

$$R_i g(x) = 0 \text{ for } i = 1, 2$$

holds.
To characterize a differential time index and a differential space index for the PDAE (1) we use as traditional semidiscretization methods the method of lines (MOL) and the Rothe method.
For the space discretization we use finite difference approximations on an equidistant grid $\{x_i = ih, i = 1(1)N + 1, h = \frac{L}{N+1}\}$ covering the interval $0 \leq x \leq L$. Replacing u_{xx} by its finite difference approximation of second order we get the continuous time semi-discrete problem of N decoupled differential-algebraic equations of first order

$$\begin{aligned} A\omega_k'(t) + (\lambda_k B + C)\omega_k(t) &= \varphi_k(t), \qquad k = 1, \cdots, N \\ \omega_k(t_0) &= \gamma_k \end{aligned} \tag{2}$$

where λ_k denotes the k-th eigenvalue of a suitable defined semi-discretization matrix P of the continuous operator $\frac{\partial^2}{\partial^2 x}$. The vector functions $\omega_k(t)$, $k = 1, \cdots, N$ are coefficients in a series expansion of the MOL-approximation U (for u) with respect to the eigenvectors of P.

The application of Rothes method yields a sequence of differential-algebraic equations of second order

$$\begin{aligned} Bu_j''(x) + (\frac{1}{\tau}A + C)u_j(x) &= f_j(x) + \frac{1}{\tau}Au_{j-1}(x), \qquad j = 1, 2, \ldots \\ R_i u(t,x) &= 0, \ i = 1, 2 \end{aligned} \tag{3}$$

where τ denotes the time-stepsize.

The N DAEs (2) allow us to define an uniform differential time index for the PDAE (1). Among others this uniform time index determines the number of consistence conditions between the initial conditions and the functions $\varphi_k(t)$ on the right hand side of the DAEs (2). For special matrices A, B, C the regular matrix pencil $\{A, \lambda_k B + C\}$ does not have the same differential time index for all k in the limit $h \to 0$. It can be different for finite many $k \in \mathbf{N}$, that means we have an index jump.

We discuss this index jump and show that problems with an index jump are qualitatively different from problems with an uniform differential time index.

With the sequence (3) of DAEs we can define a differential space index for the PDAE (1). It gives the number of consistence conditions between the boundary conditions $R_i(t, x) = 0$, $i = 1, 2$ and the functions on the right hand side of the DAEs (3). This index is identical with the algebraic x-index of Campbell/Marszalek ([2]).

The sequence of DAEs (3) can also be characterized by a space perturbation index. For linear PDAEs (1) this index is identical with the differential space index. Finally we give some numerical examples and determine the different indices.

References

[1] K.E. Brenan, S.L. Campbell, and L.R. Petzold; Numerical Solution of Initial-Value Problems in Differential-Algebraic equations, SIAM, 1996.

[2] S.L. Campbell and W. Marszalek, The Index of an Infinite Dimensional Implicit System, Preprint 1996.

[3] C.W. Gear, Differential-algebraic equations, indices, and integral algebraic equations, SIAM J. Nemer., Anal., vol.27, 1990.

[4] E. Hairer, C. Lubich, and M. Roche, The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods, Springer-Verlag, New-York, 1989.

[5] E. Hairer and G. Wanner, Solving Ordinary Differential Equations II, Springer-Verlag, 1991.

5.33 Waveform Relaxation Methods for Time-Dependent Partial Differential Equations: Recent Developments

Stefan Vandewalle
K.U. Leuven, Dept. Computer Science, Celestijnenlaan 200 A, 3030 Heverlee, Belgium
E-mail: stefan@euler.cs.kuleuven.ac.be

Standard time-stepping

Numerical methods for the simulation of time-evolution processes usually employ *time-stepping.* Although well understood and widely used in practice, time-stepping schemes suffer from drawbacks which may limit their effectiveness on sequential and parallel computers. A first problem stems from the global nature of the discretisation, which precludes the exploitation of multi-rate behaviour. A second problem originates from the inherently sequential nature of the time-stepping process. Unless the problem to be solved in each time-step is very large compared to the number of processors, parallel overheads easily dominate the calculation.

Space-time grid methods

Both problems are addressed and possibly alleviated by the use of so-called *space-time grids.* The time-dependent problem is then discretised and solved on several or on many time-levels simultaneously, or on grids extending in space and time. In other words, an N-dimensional time-dependent problem is solved as a problem on an $N+1$-dimensional grid.

Several such methods have been developed, analysed and implemented recently; they are based on applying the multigrid method to the discrete system of equations associated with the space-time grid: multigrid waveform relaxation, time-parallel multigrid and space-time multigrid [1, 6]. The parallel performance of these methods has been compared to the parallel performance of standard time-stepping methods, and was shown to be a (possibly) orders of magnitude improvement [2, 7].

Waveform relaxation

Waveform relaxation methods solve the time-dependent partial differential equation along a continuous time-window. Their discrete-time variants operate on space-time grids. They can be parallelised easily and offer the potential of multi-rate integration (although the latter has rarely been exploited so far).

After spatial discretisation, a time-dependent partial differential equation is transformed into a system of ODEs. In the linear case, those are of the form $B\dot{u}(t) + Au(t) = f(t)\ ,\quad u(0) = u_0$. The basic waveform relaxation method solves the system by applying a splitting to the coefficient matrices B and A: $B = M_B - N_B$ and $A = M_A - N_A$, and by iterating as

$$\left(M_B \frac{d}{dt} + M_A\right) u^{(\nu)}(t) = \left(N_B \frac{d}{dt} + N_A\right) u^{(\nu-1)}(t) + f(t)\ ,\ u^{(\nu)}(0) = u_0\ .$$

The resulting iteration is convergent under mild conditions that are easily satisfied in practice. Convergence is rather slow however, and an acceleration of the basic scheme is called for.

Recent Developments

Two of the acceleration schemes (SOR and multigrid) have been investigated for several years now. In [4, 5] we have studied different aspects of the multigrid acceleration. In [3] we concentrated on SOR acceleration schemes. It has been shown that the multigrid waveform acceleration and the SOR waveform acceleration (in particular the convolution SOR variant) lead to a similar convergence improvement for *time-dependent* problems, as the standard multigrid and SOR methods do for *time-independent* problems.

Other acceleration schemes have been suggested only recently; in some cases analysis is just starting and numerical experience is mostly lacking: Krylov subspace acceleration methods based on a CGR or

GMRES technique (Andrew Lumsdaine), acceleration by a Toeplitz matrix-splitting (Sigitas Keras), multi-splitting methods (Bert Pohl), waveform preconditioning schemes (Kevin Burrage), domain decomposition and Schwarz methods (Hong Zhang), etc.

In Leuven, we are currently analysing and implementing some of these methods in the framework of the research project *New numerical methods for the solution of time-dependent partial differential equations*, funded by the Catholic University of Leuven and the Belgian National Fund for Scientific Research (NFWO). At present, we are looking at the Chebyshev acceleration and Krylov subspace techniques. In this talk I will survey these developments, and I plan to compare the performance and applicability of the different acceleration schemes.

References

[1] G. Horton and S. Vandewalle. A space-time multigrid method for parabolic P.D.E.s. *SIAM J. Sci. Comput.*, 16(4):848–864, July 1995.

[2] G. Horton, S. Vandewalle, and P. Worley. An algorithm with polylog parallel complexity for solving parabolic partial differential equations. *SIAM J. Sci. Comput.*, 16(3):531–541, May 1995.

[3] J. Janssen and S. Vandewalle. On sor waveform relaxation methods. Technical Report CRPC-95-4, California Institute of Technology, September 1995. (submitted to SIAM J. Num. Anal.).

[4] J. Janssen and S. Vandewalle. Multigrid waveform relaxation on spatial finite element meshes: the discrete-time case. *SIAM J. Sci. Comput.*, 17(1), January 1996. (to appear).

[5] J. Janssen and S. Vandewalle. Multigrid waveform relaxation on spatial finite element meshes: the continuous-time case. *SIAM J. Num. Anal.*, 33(2), April 1996. (to appear).

[6] S. Vandewalle. *Parallel Multigrid Waveform Relaxation for Parabolic Problems.* B.G. Teubner Verlag, Stuttgart, 1993.

[7] S. Vandewalle and E. Van de Velde. Space-time concurrent multigrid waveform relaxation. *Annals of Numerical Mathematics*, 1(1-4):347–363, April 1994.

5.34 Numerical Methods for Nonlinear Oscillatory Multibody Dynamic Systems

Jeng Yen
Army High Performance Computing Research Center, University of Minnesota, Minneapolis, MN 55415, U.S.A.
Linda R. Petzold
Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, U.S.A.
E-mail: petzold@cs.umn.edu

The work of Jeng Yen was partially supported by ARO contract number DAAL03-89-C-0038 with the University of Minnesota Army High Performance Computing Research Center, and by ARO contract number DAAH04-94-G-0208.
The work of Linda Petzold was partially supported by ARO contract number DAAL03-89-C-0038 with the University of Minnesota Army High Performance Computing Research Center, and by ARO contract number DAAL03-92-G-0247, and by DOE contract number DE-FG02-92ER25130, and by the Minnesota Supercomputer Institute.

One of the outstanding problems in the numerical simulation of mechanical systems is the development of efficient methods for dealing with highly oscillatory systems. These types of systems arise for example in vehicle simulation in modelling the suspension system or tires, in some models for contact and impact, and in flexible body simulation from vibrations in the structural model. Simulations involving high frequency vibration can take a huge number of time steps, often as a consequence of oscillations which are not physically important. On the other hand, the components causing the oscillations cannot usually be eliminated from the model because in some situations they are critical to the simulation. In this lecture, we will first examine in some detail the structure of multibody and flexible systems, and implications for efficient numerical solution. Then we will explore two types of methods. The first class of methods damps out the oscillation via highly stable implicit methods. Even in this relatively simple approach, problems may arise for Newton iteration convergence, due to the nonlinearities. The second class of methods involves linearizing the system around the smooth solution. The linearized system can be solved rapidly via a number of different methods.

Much recent work has been focused on the development of numerical methods and underlying theory for the solution of multibody dynamic systems (MBS) consisting of fast and slow subsystems [9, 7]. These types of systems occur frequently as initial value problems in the computer-aided design and modeling of constrained mechanical systems, molecular dynamics, and in many other applications [15]. It is well-known that the characteristics of fast or slow solution is determined not only by the modeling aspects, e.g., the coefficients of stiffness and damping, but also via the initial conditions and events that may excite stiff components in the system during the simulation. The governing equations of motion of a mechanical system of stiff or highly oscillatory force devices may be written as differential-algebraic equations (DAE) [2]:

$$\begin{aligned} M(q)\ddot{q} + G^T(q)\lambda - (f^s + f^n) &= 0 \\ g(q) &= 0 \end{aligned} \tag{1}$$

where $q = [q_1, ..., q_n]^T$ is the generalized coordinate, $\dot{q} = \frac{dq}{dt}$ is the generalized velocity, $\ddot{q} = \frac{d^2q}{dt^2}$ is the acceleration and $\lambda = [\lambda_1, ..., \lambda_m]^T$ is the Lagrange multiplier. The stiff or oscillatory force is $f^s = \sum_i^{n_f} f_i^s$, and f^n includes all the field forces and the external forces which are non-stiff compared to the stiff components, e.g.,

$$\|\frac{\partial f^s}{\partial q}\| \gg \|\frac{\partial f^n}{\partial q}\|. \tag{2}$$

The kinematic constraints is g, and $G = \frac{\partial g}{\partial q}$, and M is the mass-inertia matrix. For the stiff force components in (1), we assume that

$$f_i^s = -B_i(q)(K_i\eta_i(q) + C_i\frac{d\eta_i}{dt}) \tag{3}$$

where η_i is smooth $\forall i \in 1, ..., n_f$ and $B_i = \frac{\partial \eta_i}{\partial q}^T$, and K_i, C_i are the associated stiffness and damping factors. For some generalized coordinate sets, the function η may be linear or even identities, e.g., for instance $\eta_i = q_{i_l}$ for some $i \in \{1, ..., n_f\}$ and $i_l \in \{1, ..., n\}$. When the components of the coefficient matrices K_i and C_i become large, these force components may cause rapid decay or high frequency oscillation in the solution of (1). In this notation, the stiff force term in (1) can be written as

$$f^s = -B(q)(K\eta(q) + CB(q)v) \tag{4}$$

To demostrate the problem of oscillation and the recent developments in this area, we will present two examples: a stiff pendulum and a 2D *bushing* problem. The former is a very simple example of a type of system often seen in modeling molecular dynamic systems, and the latter is a general form of modeling force devices in multibody mechanical systems. There are several observations that we can make immediately from these examples. First, the choice of global coordinates strongly determines the form and local eigenstructure of the system. Second, the eigenstructure of the local Jacobian matrix varies rapidly, on the scale of the high-frequency oscillation, in a neighborhoold of the smooth solution. We will use the fact that the eigenstructure varies slowly along the smooth solution to construct efficient numerical methods and accurate approximate solutions.

Given the possibility of a rapidly changing local eigenvalue structure, perhaps the simplest strategy is to consider damping the oscillation whenever it is not important via highly stable implicit numerical methods. In recent work [12] we have considered the solution of mechanical systems with high frequency vibrations via this type of technique. In experiments with the bushing problem solved directly by low-order BDF methods, we found that the methods experienced severe problems with Newton convergence. To overcome these problems, we proposed a *coordinate-split* (CS) formulation of the equations of motion, and a Newton-type iteration for solving the coordinate-split equations at each time step. The coordinate-split formulation eliminates problems due to obtaining an accurate predictor for the Lagrange multiplier variables because these variables are no longer present in the computation. We found that the coordinate-split formulation worked well for several test problems involving mechanical systems with high frequency oscillations. However, for problems with very high-frequency oscillations, there are still difficulties with Newton convergence.

The Jacobian matrix for solving the nonlinear equations of the coordinate-split formulation at each time step involves several terms which are complicated to compute and which are small at the solution of the nonlinear system. These are terms of second-order which correspond to the derivative of the projection operator onto the constraints. Away from the solution, these terms are highly oscillatory. By neglecting these terms, we found that the resulting Newton-type method converged much faster for oscillating test problems like the bushing problem. We called the resulting method the *modified coordinate-split*, or CM method. The modified coordinate-split (CM) method performed extremely well in numerical experiments described in [12] and in other experiments we performed subsequently. In [13], convergence of the CM iteration is analyzed, and the improved convergence for oscillatory multibody systems is explained. Intuitively, by neglecting these terms the CM-iteration approximates the Jacobian along the smooth solution, thus yielding more reliable Newton directions.

Often in multibody systems the components exhibiting high frequency oscillation result from the potential forces induced by material deformations. In flexible body multibody dynamic systems, for example, there are usually nonlinear transformations applying to the internal and inertia forces of the flexible structure to obtain equations of motion in the generalized coordinate space [14]. The resulting system of equations contains nonlinear high frequency oscillatory forces.

In these types of problems, it is desirable to have more control over any damping. One approach to solving the high frequency oscillation problem which has been used in structural analysis is to carry out modal analysis and then eliminate the higher modes, since lower modes may preserve the slowly varying part of the solution [5]. For example, the extreme high modes of a structure are often rejected in modeling flexible effects of mehanisms, since the details of the oscillating solution are not so important as the long-term solution behavior. A similar approach has been developed in recent work on molecular dynamics simulation [15, 16]. However, due to nonlinear oscillatory forces in the multibody formulation, the modal analysis needs to be carried out at each time step to resolve the rapidly varying local eigenvalue structure of the system, resulting in very costy computations. Noting that the local eigenstructure varies slowly along the smooth solution, we will describe a class of methods which is based on modal analysis via linearizing the oscillating components around the smooth solution, and present some examples from flexible systems.

References

[1] N. N. Bogoliubov and Y. A. Mitropolski, *Asymptotic Methods in the Theory of Nonlinear Oscillations*, Hindustan Publishing Corp., Delhi, India, 1961.

[2] K. E. Brenan, S. L. Campbell and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Elsevier Science Publishers, 1989.

[3] A. Cardonna and M. Geradin, *Time integration of the equations of motion in mechanism analysis*, Comput. and Struct. 33 (1989), pp. 801-810.

[4] C.W. Gear, G.K. Gupta and B.J. Leimkuhler, *Automatic integration of the Euler-Lagrange equations with constraints*, J. Comp. Appl. Math., vol. 12 & 13, 1985, 77-90.

[5] T. J. R. Hughes, *The Finite Element Method*, Prentice-Hall, Inc., Englewood Cliffs, 1987.

[6] P. Leger and E. L. Wilson, *Generation of load dependent Ritz transformation vectors in structural dynamics*, Eng. Comput. 4 (1987), 309-318.

[7] Ch. Lubich, *Integration of stiff mechanical systems by Runge-Kutta methods*, ZAMP, Vol. 44, 1022-1053, 1993.

[8] B. Nour-Omid, *Applications of the Lanczos algorithm*, Comp. Phy. Comm., 53, 1989.

[9] S. Reich, *Numerical integration of highly oscillatory hamiltonian systems using slow manifolds*, Beckman Institute, University of Illinois, UIUC-BI-TB-94-06, 1994.

[10] H. C. Chen and R. L. Taylor, *Using Lanczos vectors and Ritz vectors for computing dynamic responses*, Eng. Comput. 6 (1989), 151-157.

[11] E. L. Wilson, M. Yuan and J. M. Dickens, *Dynamic analysis by direct superposition of Ritz vectors*, Earthquake Engineering and Structural Dynamics 10 (1982), 813-821.

[12] J. Yen and L. Petzold, *On the numerical solution of constrained multibody dynamic systems*, University of Minnesota AHPCRC 94-038, 1994.

[13] J. Yen and L. Petzold, *Convergence of the iterative methods for coordinate-splitting formulation in multibody dynamics*, TR 95-052, Tech. Rept., Dept. of Computer Science, Univ. of Minnesota, July 1995.

[14] W. S. Yoo and E. J. Haug, *Dynamics of articulated structures. Part I. Theory*, J. Struct. Mech., 14 (1986), 105-126.

[15] G. Zhang and T. Schlick, *The Langevin/implicit Euler/normal-mode scheme for molecular dynamics at large time steps*, J. Chem. Phys. 101 (1994), 4995-5012.

[16] G. Zhang and T. Schlick, *LIN: A new algorithm to simulate the dynamics of biomolecules by combining implicit-integration and normal mode techniques*, J. Comp. Chemistry 14 (1993), 1212-1233.