CHALMERS
FINITE ELEMENT CENTER

PREPRINT 2001–20

Topics in adaptive computational methods for differential equations

Claes Johnson, Johan Hoffman and Anders Logg

Chalmers Finite Element Center
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg Sweden 2001

CHALMERS FINITE ELEMENT CENTER

Preprint 2001–20

Topics in adaptive computational methods for differential equations

Claes Johnson, Johan Hoffman and Anders Logg

CHALMERS

Chalmers Finite Element Center
Chalmers University of Technology
SE–412 96 Göteborg Sweden
Göteborg, October 2001

Topics in adaptive computational methods for differential equations
Claes Johnson, Johan Hoffman and Anders Logg
NO 2001–20
ISSN 1404–4382

Chalmers Finite Element Center
Chalmers University of Technology
SE–412 96 Göteborg
Sweden
Telephone: +46 (0)31 772 1000
Fax: +46 (0)31 772 3595
www.phi.chalmers.se

Printed in Sweden
Chalmers University of Technology
Göteborg, Sweden 2001

TOPICS IN ADAPTIVE COMPUTATIONAL METHODS FOR DIFFERENTIAL EQUATIONS

CLAES JOHNSON, JOHAN HOFFMAN, AND ANDERS LOGG

ABSTRACT. We discuss two topics of adaptive computational methods for differential equations: (i) individual time-stepping and (ii) subgrid modeling, and we present some applications including the computability and predictability of the Solar System and aspects of subgrid modeling in convection-diffusion-reaction systems.

1. INTRODUCTION

The two fundamental aspects of science of *formulating equation* (modeling) and *solving equation* (computation), are today interacting on a new level in the field of *Computational Mathematical Modeling*. Particular focus is put on adaptive methods with feed-back from the computational process to reach goals of efficiency and control of modeling and computational errors. Over the years we have in cooperation with in particular the Rannacher group in Heidelberg developed a general methodology for adaptive computational methods for differential equations based on duality techniques and finite element discretization in space-time, with applications to a variety of problems in fluid and solid mechanics, electromagnetics, and reactive flow (see [1] and [13] and references therein). The methodology is being integrated into an applied mathematics education program from beginning undergraduate to graduate level, which is presented in the series of books ([2], [3] and [4]). Our research and educational programs are presented on `www.phi.chalmers.se`.

In this note we present recent results on the following two topics of adaptivity: (i) individual time-stepping with application to the computability and predictability of the Solar System and the Lorenz system, and (ii) subgrid modeling with applications to convection-diffusion-reaction systems with fractal solutions.

We first present a general picture of the basic problems of formulating and solving equations. We consider a mathematical model of the form $A(u) = f$, where A is a *differential operator*, f is given *data*, and u is the *solution*. The model is subject to perturbations from *data* represented by $\hat{f}$, *modeling* represented by $\hat{A}$, and *computation* represented by U viewed as an approximate solution to a perturbed problem $\hat{A}(\hat{u}) = \hat{f}$ with exact solution $\hat{u}$. We say that the *data/modeling error* is equal to $u - \hat{u}$ and the *computational error* is

Date: October 4, 2001.

Key words and phrases. Multi-adaptivity, individual time-stepping, subgrid modeling.

Department of Mathematics, Chalmers University of Technology, SE–412 96 Göteborg, Sweden, *email*: claes@math.chalmers.se

Invited lecture at Cedya 2001: Congreso de Ecuaciones Diferenciales y Aplicaciones, Salamanca Sept 24-28, 2001.

equal to $\hat{u} - U$, and that the total error, $u - U = u - \hat{u} + \hat{u} - U$, thus has a contribution from data/modeling and a contribution from computation. The model perturbation $\hat{A}$ may contain coefficients representing subgrid models in the form of e.g. turbulent viscosities.

An *adaptive method* for solving $A(u) = f$ includes a feed-back process, where the quality of computed solutions U of perturbed models $\hat{A}(\hat{u}) = \hat{f}$, are investigated with the objective of decreasing the modeling error $u - \hat{u}$ by improving the model $\hat{A}$, and/or the computational error $\hat{u} - U$, typically by appropriately modifying the local mesh size. Typically an adaptive method is based on *posteriori error* estimates estimating the modeling or computational errors in terms of computable residuals such as $f - A(U)$ or $\hat{f} - \hat{A}(U)$, or estimated residuals $f - A(\hat{u})$.

Adaptive methods for solving differential equations form a rapidly expanding area with important applications. One may argue that the activity of formulating and solving equations is inherently adaptive with feed-back between computation and modeling. Adaptivity of modeling and computation can be viewed as one aspect of *optimization* with the objective of improving the model or the computation. Adding also aspects of optimization of solutions, which is often the main objective, one gets a full picture of optimization of model, computation and solution.

An adaptive method typically involves a *stopping criterion* guaranteeing that the error in some norm is less than a given tolerance, and a *modification strategy* to be applied if the stopping criterion is not satisfied. A modification strategy may concern quantities related to computation such as the local mesh size, or quantities related to modeling such as a turbulent viscosity. The modification strategy for the local mesh size is often based on *equidistribution* with the objective of satisfying the stopping criterion with a minimal number of degrees of freedom, or largest possible local mesh size. Both the stopping criterion and modification criterion may be based on *a posteriori* error estimates involving the residuals $f - A(U)$ and $\hat{f} - \hat{A}(U)$ of computed solutions, and/or *a priori* error estimates involving estimates of the exact solution $\hat{u}$ or u.

2. G^2: The General Galerkin method

2.1. **Features.** As computational method we use a class of finite element methods, which we refer to as the General Galerkin method, or G^2 in short, with the following features:

- piecewise polynomial approximation in space/time,
- residual of trial function orthogonal to test functions (Galerkin orthogonality),
- continuous trial functions of degree q: cG(q)
- discontinuous trial/test functions of degree q: dG(q),
- Eulerian form or Lagrangean form depending on the orientation of the space/time mesh,
- least-squares modification/stabilization,
- residual-based artificial viscosity,
- multi-adaptivity: e.g. individual time-stepping.

Applying G^2 to a specific problem leads to a finite-dimensional (discrete) system of equations involving certain integrals expressing the Galerkin orthogonality. The integrals are evaluated using quadrature and the resulting discrete system of algebraic equations is solved by some iterative method such as fixed point iteration or Newton's method. The total computational error in G^2 has contributions from (i) Galerkin discretization error, (ii) quadrature error, (iii) discrete solution error.

2.2. **An *a posteriori* error estimate for** cG(1)**.** We now derive a prototypical *a posteriori* error estimate for G^2 applied to an initial value problem for a system of differential equations of the form

$$\begin{aligned} u_t + f(t,u) &= 0 \qquad \text{for } 0 < t < T, \\ u(0) &= u_0, \end{aligned} \tag{2.1}$$

where $f(t,\cdot) : R^N \to R^N$ is a given mapping, $u_t = \frac{du}{dt}$, and T is a final time. As a particular case of G^2, we consider the cG(1)-method based on continuous piecewise linear trial functions and discontinuous piecewise constant test functions on a partition $0 = t_0 < t_1 < t_2 < .... < t_N = T$ into intervals $I_n = (t_{n-1}, t_n)$ with corresponding time-steps $k(t) = k_n = t_n - t_{n-1}$, $t \in I_n$.

In cG(1) we seek $U(t)$ continuous piecewise linear satisfying

$$\int_{I_n} (U_t + \hat{f}(t,U))dt = 0, \quad n = 1, 2, ..., N, \quad U(0) = u_0,$$

corresponding to testing against piecewise constants. Choosing different quadratures, such as midpoint quadrature with

$$\hat{f}(t,U) = f(\frac{t_{n-1}+t_n}{2}, \frac{U_n + U_{n-1}}{2}) \quad \text{on } I_n,$$

we obtain different fully discrete schemes where the quadrature error may be viewed as a part of the perturbation $\hat{f}$ of f.

We now derive an *a posteriori error estimate* for the error at final time $\|e(T)\| = \|u(T) - U(T)\|$ where $\|\cdot\|$ is the Euclidean R^N norm. We introduce the continuous dual "backward" linearized problem:

$$-\phi_t + A(t)^*\phi = 0 \quad \text{in} \quad (0,T), \qquad \phi(T) = e(T),$$

where

$$A(t) \equiv \int_0^1 f'(t, su(t) + (1-s)U(t))\, ds,$$

and $f'(t,\cdot)$ is the Jacobian of $f(t,\cdot)$. Using that

$$A(t)e = \int_0^1 f'(t, su(t) + (1-s)U(t))e\, ds$$

$$= \int_0^1 \frac{d}{ds} f(t, su(t) + (1-s)U(t))\, ds = f(t,u(t)) - f(t,U(t)),$$

we obtain the following error representation:

$$\|e(T)\|^2 = \|e(T)\|^2 + \int_0^T e \cdot (-\phi_t + A^*\phi)\, dt$$

$$= \int_0^T (e_t + A(t)e) \cdot \phi \, dt + e(0) \cdot \phi(0)$$

$$= \int_0^T (u_t + f(t,u)) \cdot \phi \, dt - \int_0^T (U_t + \hat{f}(t,U)) \cdot \phi \, dt$$

$$+ \int_0^T (\hat{f}(t,U) - f(t,U)) \cdot \phi \, dt + e(0) \cdot \phi(0)$$

$$= - \int_0^T (U_t + \hat{f}(t,U)) \cdot \phi \, dt + \int_0^T (\hat{f}(t,U) - f(t,U)) \cdot \phi \, dt + e(0) \cdot \phi(0),$$

where we used that $u_t + f(t,u) = 0$.

For the local mean value $P_k\phi$ of ϕ on the partition, we have

$$\|(\phi - P_k\phi)(t)\| \leq \int_{I_n} \|\frac{d\phi}{ds}(s)\| \, ds, \quad t \in I_n.$$

Using the Galerkin orthogonality, we can write the error representation in the form

$$\|e(T)\|^2 = - \int_0^T (U_t + \hat{f}(t,U)) \cdot (\phi - P_k\phi) \, dt$$

$$+ \int_0^T (\hat{f}(t,U) - f(t,U)) \cdot \phi dt + e(0) \cdot \phi(0).$$

We can use this estimate for *a posteriori* error estimation by first computing an approximation of the dual solution ϕ and then computing $\phi - P_k\phi$. We can also derive different upper estimates by introducing norms and using Cauchy's inequality. A typical such *a posteriori* error estimate takes the form

$$\|e(T)\| \leq S^{[1]}(T) \max_{0 \leq t \leq T} \|k(t)R(U,t)\|$$

$$+ S^{[0]}(T) \max_{0 \leq t \leq T} \|\hat{f}(t,U) - f(t,U)\| + S_d(T)\|e(0)\|,$$

where $R(U,t) = U_t + \hat{f}(U,t)$, $\hat{f}(t,U) - f(t,U)$, and $e(0)$ act as residuals connected to computation, modeling and data, and

$$S^{[1]}(T) \equiv \frac{\int_0^T \|\phi_t\| dt}{\|\phi(T)\|}, \; S^{[0]}(T) \equiv \frac{\int_0^T \|\phi(s)\| ds}{\|\phi(T)\|}, \; S_d(T) \equiv \frac{\|\phi(0)\|}{\|\phi(T)\|},$$

are the corresponding stability factors. Note the presence of the time-step factor $k(t)$ in the term $S_c(T)\|k(t)R(U,t)\|$ resulting from the difference $\phi - P_k\phi$.

The stability factors may be computed by linearizing around a computed solution $U = U_h$ and estimating the data $e_H(T)$ on a coarser mesh H by $U_h - U_H$, and computing the corresponding dual solution ϕ and its stability factors.

The size of the stability factors vary with the model, the particular solution and the error norm. Obviously, the presence of large stability factors indicates that the problem is sensitivity to small perturbations, and pertinent questions concern the predictability (small data/modeling error) and computability (small computational error). We discuss the issues of predictability and computability below in the context of the Solar System and the Lorenz System.

3. Multi-adaptivity: Individual time-steps

In this section we give a brief overview of the multi-adaptive Galerkin methods mcG(q) and mdG(q), presented in [11, 12], including two key applications,

3.1. **The** mcG(q) **method.** To formulate the mcG(q) method, we partition the interval $(0,T)$ individually for the different components with individual time-intervals $\{I_{ij}\}_j$ and time-steps $\{k_{ij}\}_j$ for every individual component $U_i(t)$.

The mcG(q) method for (2.1) reads: Find $U \in V$ with $U(0) = u_0$, such that

$$\int_0^T (\dot{U}, v)\, dt + \int_0^T (f(U,\cdot), v)\, dt = 0 \;\forall v \in W, \tag{3.1}$$

where

$$\begin{array}{rcl} V & = & \{v \in C([0,T]) : v_i|_{I_{ij}} \in \mathcal{P}^{q_{ij}}(I_{ij}),\; j = 1,\dots,M_i,\; i = 1,\dots,N\}, \\ W & = & \{v : v_i|_{I_{ij}} \in \mathcal{P}^{q_{ij}-1}(I_{ij}),\; j = 1,\dots,M_i,\; i = 1,\dots,N\}, \end{array} \tag{3.2}$$

and where $\mathcal{P}^q(I)$ denotes the linear space of polynomials of degree $\leq q$ on I. The trial functions in V are thus continuous piecewise polynomials, locally of degree q_{ij}, and the test functions in W are discontinuous piecewise polynomials that are locally of degree $q_{ij} - 1$.

Noting that the test functions are discontinuous, we can rewrite the global problem (3.1) as a number of successive local problems for each component: For $i = 1,\dots,N$, $j = 1,\dots,M_i$, find $U_i|_{I_{ij}} \in \mathcal{P}^{q_{ij}}(I_{ij})$ with $U_i(t_{i,j-1})$ given, such that

$$\int_{I_{ij}} \dot{U}_i v\, dt + \int_{I_{ij}} f_i(U,\cdot) v\, dt = 0 \;\forall v \in \mathcal{P}^{q_{ij}-1}(I_{ij}). \tag{3.3}$$

We notice the presence of the vector $U(t) = (U_1(t),\dots,U_N(t))$ in the local problem for U_i on I_{ij}. If thus component $U_{i_1}(t)$ couples to component $U_{i_2}(t)$ through f, this means that in order to solve the local problem for component $U_{i_1}(t)$ we need to know the values of component $U_{i_2}(t)$ and vice versa. The solution is thus implicitly defined by (3.3).

Making an Ansatz for every component $U_i(t)$ on every local interval I_{ij} in terms of a nodal basis for $\mathcal{P}^{q_{ij}}(I_{ij})$, we can rewrite (3.3) as

$$\xi_{ijm} = \xi_{ij0} + \int_{I_{ij}} w_m^{[q_{ij}]}(\tau_{ij}(t))\, (-f_i(U(t),t))\, dt,\; m = 1,\dots,q_{ij}, \tag{3.4}$$

where $\{\xi_{ijm}\}_{m=0}^{q_{ij}}$ are the nodal degrees of freedom for $U_i(t)$ on the interval I_{ij}, $\{w_m^{[q]}\}_{m=1}^{q} \subset \mathcal{P}^{q-1}(0,1)$ are corresponding polynomial weight functions and τ_{ij} maps I_{ij} to $(0,1]$: $\tau_{ij}(t) = (t - t_{i,j-1})/(t_{ij} - t_{i,j-1})$. Here we assume that the solution is expressed in terms of a nodal basis with the end-points included, so that by the continuity requirement $\xi_{ij0} = \xi_{i,j-1,q_{i,j-1}}$.

Finally, evaluating the integral in (3.4) using nodal quadrature, we obtain a fully discrete scheme in the form of an implicit Runge-Kutta method: For $i = 1, \ldots, N$, $j = 1, \ldots, M_i$, find $\{\xi_{ijm}\}_{m=0}^{q_{ij}}$, with ξ_{ij0} given by the continuity requirement, such that

$$\xi_{ijm} = \xi_{ij0} + k_{ij} \sum_{n=0}^{q_{ij}} w_{mn}^{[q_{ij}]} \; (-f_i(U(\tau_{ij}^{-1}(s_n^{[q_{ij}]})), \tau_{ij}^{-1}(s_n^{[q_{ij}]}))), \;\; m = 1, \ldots, q_{ij}, \tag{3.5}$$

for certain weights $\{w_{mn}^{[q]}\}$, and certain nodal points $\{s_n^{[q]}\}$ (see [11]).

Using standard techniques [11] we can prove that the mcG(q) method is of order $2q$ (locally of order $2q_{ij}$). Assuming that the same time-steps are used for corresponding position and velocity components for a Hamiltonian system, we can also prove that the mcG(q) method conserves energy at common nodal points.

3.2. **The** mdG(q) **method.**

Generalizing the standard discontinuous Galerkin method in the same way as above for the continuous method, we obtain the mdG(q) method:

$$\xi_{ijm} = \xi_{ij0}^- + k_{ij} \sum_{n=0}^{q_{ij}} w_{mn}^{[q_{ij}]} \; (-f_i(U(\tau_{ij}^{-1}(s_n^{[q_{ij}]})), \tau_{ij}^{-1}(s_n^{[q_{ij}]}))), \;\; m = 0, \ldots, q_{ij}, \tag{3.6}$$

with new nodal points and nodal weights.

Using standard techniques we can prove that the mdG(q) method is of order $2q + 1$ (locally of order $2q_{ij} + 1$), and that it is B-stable [11].

3.3. **A Posteriori Error Analysis and Adaptivity.**

To estimate the global error at final time, and to adaptively determine the individual time-steps, we derive a posteriori error estimates for the total error, including the Galerkin error, computational errors (including round-off error) and quadrature errors (including modeling error):

$$\|e(T)\| \leq E_G + E_C + E_Q. \tag{3.7}$$

The a posteriori error estimates contain information of the sizes of different *individual* residuals and stability factors (or time-dependent stability weights) obtained from solving the dual problem, compare section 2.2. For the mcG(q) method, we obtain the following error estimate:

$$\|e(T)\| \leq \sum_{i=1}^{N} \left[S_i^{[q_i]} \max_{[0,T]} \{C_{q_i - 1} k_i^{q_i} r_i\} + \bar{S}_i^{[0]} \max_{[0,T]} |\mathcal{R}_i^{\mathcal{C}}| + \bar{S}_i^{[0]} \max_{[0,T]} |\mathcal{R}_i^{\mathcal{Q}}| \right], \tag{3.8}$$

and for the mdG(q) method we have

$$\|e(T)\| \leq \sum_{i=1}^{N} \left[S_i^{[q_i+1]} \max_{[0,T]} \left\{C_{q_i} k_i^{q_i+1} \bar{r}_i\right\} + \bar{S}_i^{[0]} \max_{[0,T]} |\mathcal{R}_i^{\mathcal{C}}| + \bar{S}_i^{[0]} \max_{[0,T]} |\mathcal{R}_i^{\mathcal{Q}}| \right], \tag{3.9}$$

where the $S_i^{[q]}$ are stability factors for the individual components of the dual solution. The true global error may thus be estimated in terms of computable stability factors and residuals.

Focusing on the Galerkin error (controlling other contributions below say 10% of the total error), the goal of the adaptive of the algorithm is to produce an approximate solution within a given tolerance TOL. Simplifying the notation, we then want to determine for each individual component $U_i(t)$ a sequence of time-steps $\{k_{ij}\}_j$ such that

$$\sum_{i=1}^{N} S_i \max_j k_{ij}^{q_{ij}} r_{ij} = \mathrm{TOL}, \tag{3.10}$$

which is achieved by

$$k_{ij}^{q_{ij}} r_{ij} = \mathrm{TOL}/(N S_i). \tag{3.11}$$

The adaptive algorithm may then be expressed as follows: Given a tolerance TOL > 0, make a preliminary guess for the stability factors and then

(1) Solve the primal problem with time-steps based on (3.11).
(2) Solve the dual problem and compute stability factors and stability weights.
(3) Compute an error estimate E.
(4) If $E \leq$ TOL then stop, and if not go back to (1).

3.4. **Computability of the Lorenz system.** We consider the Lorenz system,

$$\begin{cases} \dot{x} &= \sigma(y - x), \\ \dot{y} &= rx - y - xz, \\ \dot{z} &= xy - bz, \end{cases} \tag{3.12}$$

with the usual data $(x(0), y(0), z(0)) = (1, 0, 0)$, $\sigma = 10$, $b = 8/3$ and $r = 28$. The solution $u(t) = (x(t), y(t), z(t))$ is very sensitive to perturbations and is often described as being "chaotic". With our perspective this is reflected by stability factors with rapid growth in time.

The computational challenge is to solve the Lorenz system accurately on a time interval $[0, T]$ with T as large as possible. We investigate the computability of the Lorenz system by solving the dual problem and computing stability factors, to find out the maximum value of T.

To begin with, we examine solutions of the Lorenz system for a number of different methods. In Figure 1 we plot solutions for the x-component of the Lorenz system on $[0, 40]$. The time-step is the same for all components and constant equal to $k = 0.001$. Some time after $t = 20$, the second-order accurate cG(1) solution is no longer correct. (A "good" thing about the Lorenz system is that when the solution once starts to be incorrect, it quickly becomes very inaccurate also in picture-norm.)

Increasing the order of the method until we finally reach the 20:th and 21:st order methods cG(10) and dG(10), we get more and more accurate solutions, and are able to solve until $T = 40$. Increasing the order further does not increase the accuracy of the solution, since the error is now dominated by the computational error caused by finite precision arithmetic round-off error.

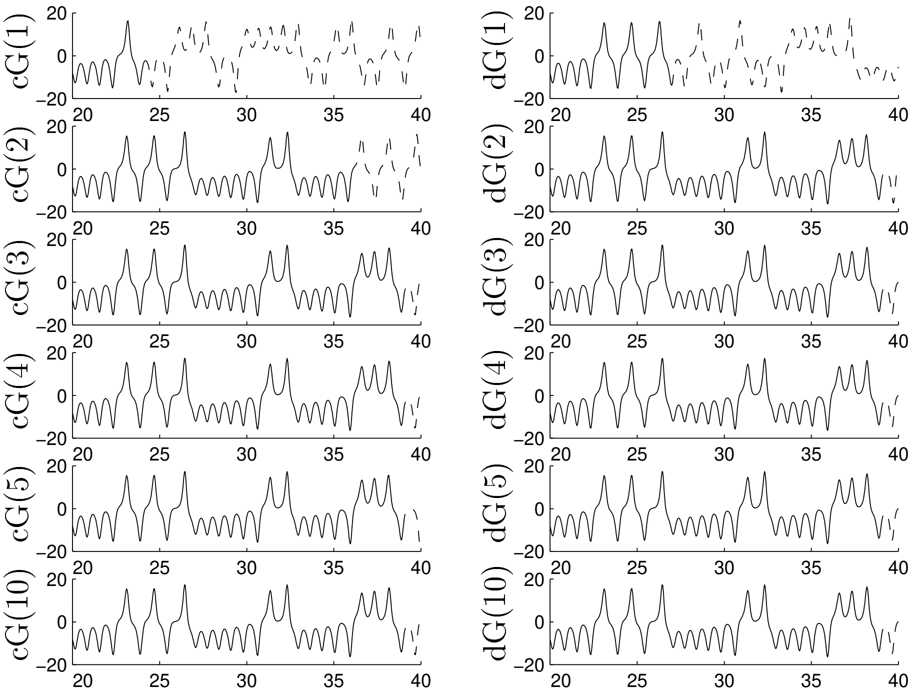

FIGURE 1. These plots show solutions, computed with different methods with constant time-step $k = 0.001$, for the x-component of the Lorenz system from time $t = 20$ to final time $T = 40$.

3.4.1. *Reaching further.* To reach further than to $T = 40$, we must decrease the computational error. A simple view of this is that at every time-step we will make a relative error of at least 10^{-16} with double precision arithmetic. These errors will accumulate at a rate determined by the stability properties of the dual. To decrease this error we must thus not decrease the time-step, but instead *increase the time-step*! In this way we will take fewer time-steps and so decrease the computational error. Since increasing the time-step we will increase the Galerkin error, we also have to increase the order of the method to keep the Galerkin error small.

In Figure 2 we present solutions with $k = 0.1$. We are now able to solve until about $T = 50$, using as few as 500 time-steps. The higher-order methods all agree until some point close to $t = 48$, and not even the 30:th-order method mcG(15) is able to get any further. This indicates that again we have reached a limit.

3.4.2. *Stability factors.* We now turn to the computation of stability factors for the Lorenz system, such as

$$S^{[q]}(T) = \max_{\|v\|=1} \int_0^T \|\varphi^{(q)}(t)\| \, dt, \tag{3.13}$$

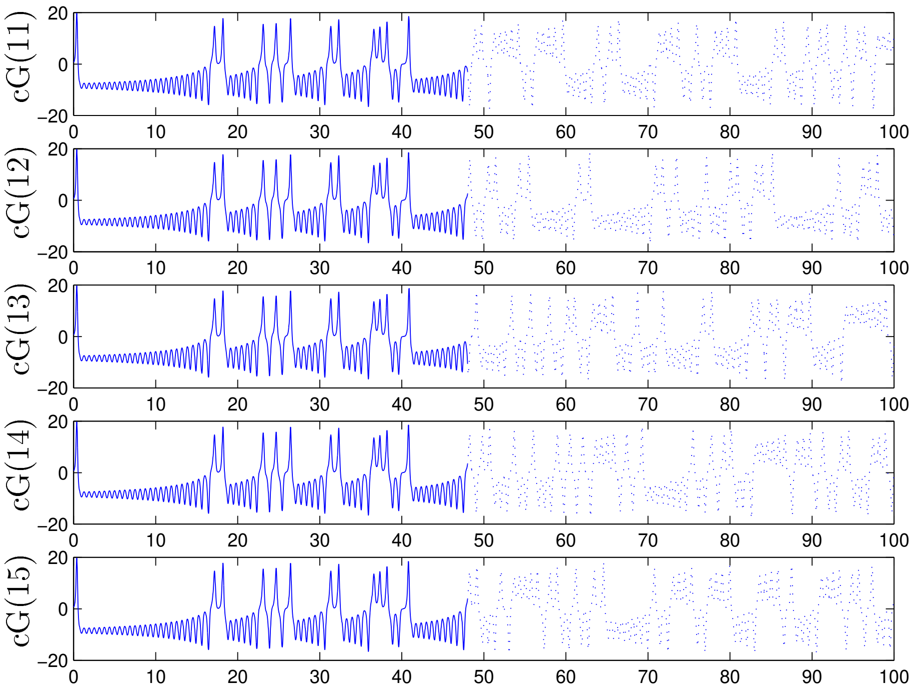

FIGURE 2. Solutions for the x-component of the Lorenz system with methods of different order, using a constant time-step $k = 0.1$.

where φ is an (approximate) solution of the dual problem with $\varphi(T) = v$. Letting Φ be an (approximate) fundamental solution of the dual problem, we have

$$(3.14) \qquad \max_{\|v\|=1} \int_0^T \|\Phi^{(q)}(t)v\|\, dt \le \int_0^T \max_{\|v\|=1} \|\Phi^{(q)}(t)v\|\, dt = \int_0^T \|\Phi^{(q)}(t)\|\, dt = \bar{S}^{[q]}(T),$$

and thus computing $\bar{S}^{[q]}(T)$ gives a bound for $S^{[q]}(T)$, which for Lorenz system turns out to be quite sharp. By computing the fundamental solution we avoid computing the maximum in (3.13).

We compute the variation of the stability factors on $[0, 50]$, see Figure 3, where we plot the stability factor for $q = 0$, corresponding to computational and quadrature errors. The stability factors grow exponentially with time, but not as fast as indicated by an a priori error estimate. An a priori error estimate indicates that the stability factors grow as

$$(3.15) \qquad S^{[q]}(T) \sim e^{\mathcal{A}T},$$

where $\mathcal{A}$ is some bound for the Jacobian of the right-hand side for the Lorenz system. Making a simple estimate, we take $\mathcal{A} = 50$, so that already at $T = 1$ we have $S(T) \approx 10^{22}$. In view of this, we would not be able to compute even to $T = 1$, and certainly not to $T = 50$ for which $S(T) \approx 10^{1000}$. The point is that although the stability factors grow very rapidly at some occasions, such as nearby the first flip at $T = 18$, they do not grow monotonically, and thus as an average grow at a moderate exponential rate.

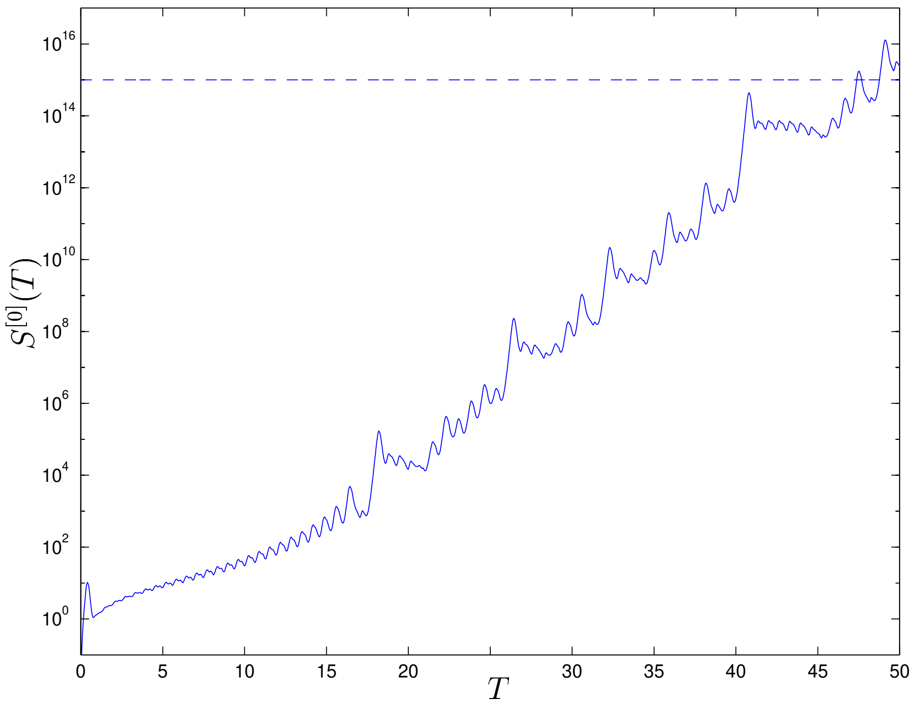

FIGURE 3. The stability factor for the computational and quadrature errors, as function of time for the Lorenz system.

3.4.3. *Conclusions.* We now make simple estimates for the growth rate of the stability factors, in order to predict how far along it is possible to compute. Fitting simple functions to the stability factors as function of time, we have the following approximations:

$$\tilde{S}^{[q]}(T) \approx 4 \cdot 10^{(q-3)+0.37T}, \tag{3.16}$$

or, simpler but not as good,

$$\tilde{S}^{[q]}(T) \approx 10^{q+T/3}. \tag{3.17}$$

From the a posteriori error estimates presented in [11], we find that the computational error can be estimated as

$$E_C \approx S^{[0]}(T) \max_{[0,T]} \|\mathcal{R}^{\mathcal{C}}\|, \tag{3.18}$$

where the computational residual $\mathcal{R}^{\mathcal{C}}$ is defined as

$$\mathcal{R}_i^{\mathcal{C}}(t) = \frac{1}{k_{ij}} \left(U(t_{ij}) - U(t_{i,j-1}) - \int_{I_{ij}} f_i(U,\cdot)\ dt \right) \tag{3.19}$$

for the mcG(q) method and similarly for the discontinuous method.

With 16 digits of precision, we cannot expect to have a computational residual smaller than about $\frac{1}{k_{ij}}10^{-16}$, so that, with the approximation above, we have

$$E_C \approx 10^{T/3} \frac{1}{\min k_{ij}} 10^{-16} = 10^{T/3-16} \frac{1}{\min k_{ij}}, \tag{3.20}$$

so that with $k_{ij} = 0.1$ as above we have

$$E_C \approx 10^{T/3-15}. \tag{3.21}$$

With time-steps $k_{ij} = 0.1$ we thus cannot expect to do much better than $T = 50$, since then we will have an error larger than unity, which we take as a criterion for an incorrect solution. Increasing the time-step further, to say $k_{ij} = 10$, we can compute a little further, but only a couple of time units, since the stability factors grow exponentially. Increasing the time-step even further we will soon have a time-step that is greater than the length of the whole interval, i.e. $k > T$.

Our conclusion is thus that by examining the stability factors, we can say that our computation is probably correct until right before $T = 50$, and that we cannot get much further with 16 digits of precision. (With 32 digits of precision we would reach $T = 100$, and so on.)

3.5. **Computability and predictability of the Solar System.** We now consider the Solar System, including the Sun, the Moon, and the nine planets, which is a particular n-body problem of fundamental importance:

$$m_i \ddot{x}_i = \sum_{j \neq i} \frac{G m_i m_j}{|x_j - x_i|^3}(x_j - x_i), \tag{3.22}$$

where $x_i(t) = (x_i^1(t), x_i^2(t), x_i^3(t))$ denotes the position of body i at time t, m_i is the mass of body i, and G is the gravitational constant.

As initial conditions we take the values at 00.00 GMT on January 1:st 2000, obtained from the US Naval Observatory with initial velocities obtained by fitting a high-degree polynomial to the values of December 1999. The initial data should be correct to five or more digits, which is similar to the available precision for the masses of the planets. We normalize length and time to have the space coordinates per astronomical unit, AU, which is (approximately) the mean distance between the Sun and Earth, the time coordinates per year, and the masses per solar mass. With this normalization, the gravitational constant is $4\pi^2$.

3.5.1. *Predictability.* Investigating the *predictability* of the Solar System, the question is how far we can accurately compute the solution, given the precision in initial data. In order to predict the accumulation rate of errors, we solve the dual problem and compute stability factors. Assuming the initial data is correct to five or more digits, we find that the Solar System is computable on the order of 500 years. Including also the Moon, we cannot compute more than a few years. The dual solution grows linearly backward in time, see Figure 4, and so errors in initial data grow linearly with time. For every extra digit of increased precision, we thus reach ten times further.

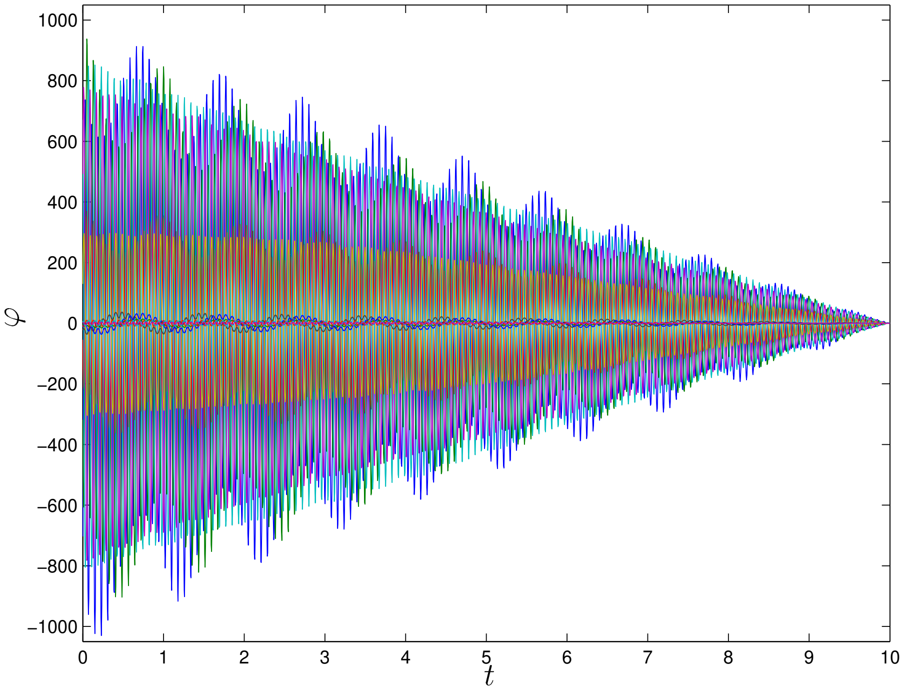

FIGURE 4. Part of the dual of the Solar System with data chosen for control of the error in position of the Moon at final time.

3.5.2. *Computability.* To touch briefly the fundamental question of the *computability* of the Solar System, concerning how far the system is computable with correct initial data and correct model, we compute the trajectories for Earth, the Moon and the Sun over a period of 50 years, comparing different methods. Since errors in initial data grow linearly, we expect numerical errors, as well as stability factors, to grow quadratically.

In Figure 5 we plot the errors for the 18 components of the solution, computed for $k = 0.001$ with cG(1), cG(2), dG(1) and dG(2). This figure contains much of information. To begin with, we see that the error seems to grow linearly for the cG methods. This is in accordance with earlier observations [9] for periodic Hamiltonian systems, recalling that the (m)cG(q) methods conserve energy [11]. The stability factors, however, grow quadratically and thus overestimate the error growth for this particular problem. In an attempt to give an intuitive explanation of the linear growth, we may think of the error introduced at every time-step by an energy-conserving method as a pure phase error, and so at every time-step the Moon is pushed slightly forward along its trajectory (with the velocity adjusted accordingly). Such errors do not accumulate but stay constant, and so repeatedly making such an error, the resulting total error grows linearly.

Examining the solutions obtained with the dG(1) and dG(2) methods, we see that the error grows quadratically as we expect. For the dG(1) solution, the error reaches a maximum level of ~ 0.5 for the velocity components of the Moon. The error in position for the Moon is much smaller. This means that the Moon is still in orbit around Earth, the

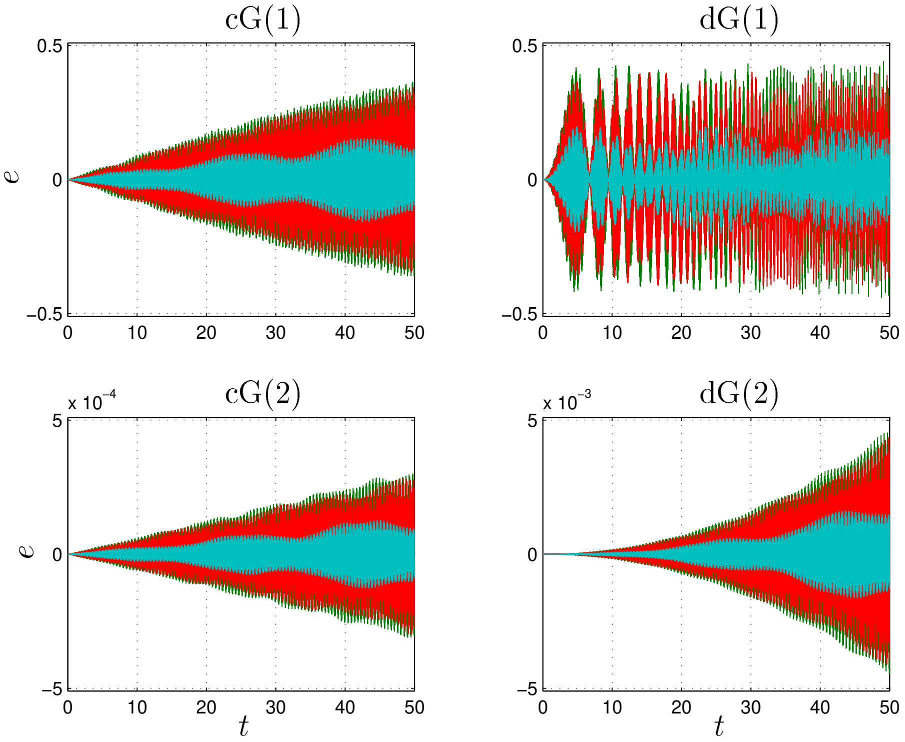

FIGURE 5. The growth of the error over 50 years for the Earth-Moon-Sun system as described in the text.

position of which is still very accurate, but the position relative to Earth is incorrect and thus also the velocity. The error thus grows quadratically until it reaches a limit. This effect is also visible for the error of the cG(1) solution; the linear growth flattens out as the error reaches the limit. Notice also that even if the higher-order dG(2) method performs better on a short time-interval, it will be outrun on a long enough interval by the cG(1) method with its linear accumulation of errors (for this particular problem).

Solving with the multi-adaptive method mcG(2), see Figure 6, the error grows quadratically. We saw in [11] that in order for the mcG(q) method to conserve energy, we require that corresponding position and velocity components use the same time-steps. Computing with different time-steps for all components, as here, we thus cannot expect to have linear error growth. Keeping $k_i^2 r_i \leq$ tol with $tol = 10^{-10}$ as here, the error grows as $10^{-4}T^2$ and we are able to reach $T \sim 100$. Decreasing tol to say 10^{-18}, we could instead reach $T \sim 10^6$.

4. Computational Subgrid Modeling

In recent years methods of *dynamic subgrid modeling* have been proposed, in particular in turbulence modeling in *Large Eddy Simulations* (LES) by Germano (1991). The purpose of a subgrid model is to model the effect of unresolvable scales on resolvable scales corresponding to closure in turbulence modeling. The basic idea in dynamic subgrid modeling

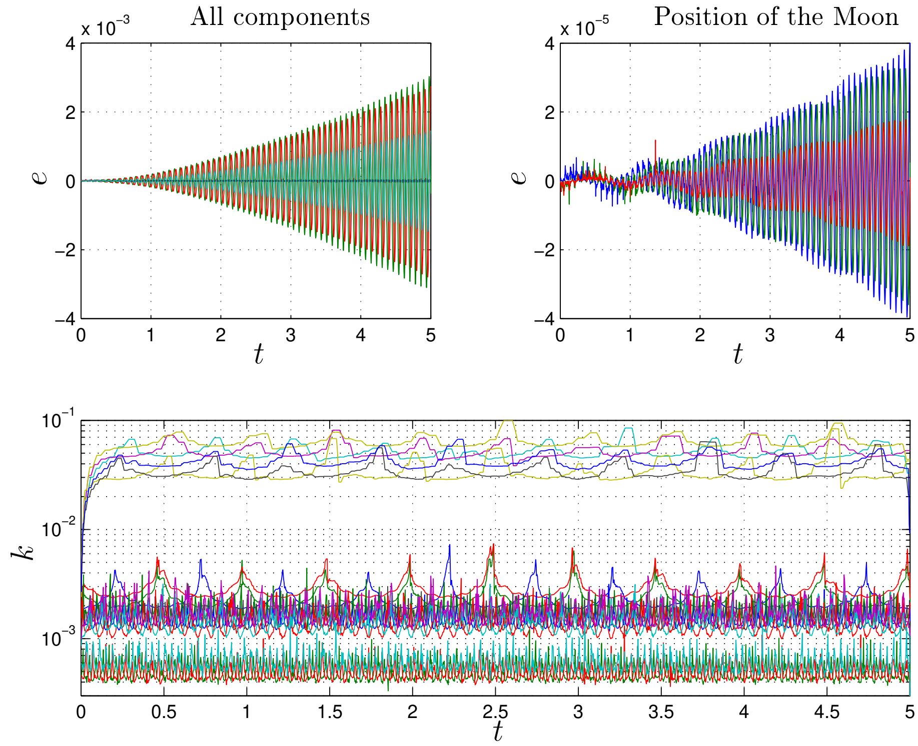

FIGURE 6. The growth of the error over 5 years for the Earth-Moon-Sun system computed with the mcG(2) method, together with the multi-adaptive time-steps.

is to fit a particular subgrid model based on computed solutions on different resolvable scales, and then extrapolate the model to subgrid scales. In order for such a process based on scale extrapolation to work, it is necessary that the underlying problem has some 'scale regularity', so that the experience gained by fitting the model on a coarse scale with a fine scale solution as reference may be extrapolated to the finer scale. It is conceivable that many problems involving a range of scales from large to small, such as fluid flow at larger Reynolds numbers and flow in heterogeneous porous media, in fact do have such a regularity, once the larger scales related to the geometry of the particular problem have been resolved.

In [5] we initiated a study of dynamic subgrid modeling in a stationary convection-diffusion problem with fractal coefficients, where a subgrid model in the form of a corrective force is extrapolated from coarser grids with the finest computational grid as reference. We showed that the quality of the solution on mesh size h with extrapolated subgrid model, is comparable to that of a solution without subgrid model on a scale $h/4$ or smaller. We here present extensions to time-dependent convection-diffusion-reaction systems with 'fractal' solutions, where the fractality again originates from data (see [5, 6, 7]). The relevance of studying fractal models is motivated by the abundant number of experimental observations of fractality in turbulent flows. Extensions to Navier-Stokes equations with the goal of connecting with DLES Dynamic Large Eddy Simulation is under way.

In the setting of a general problem of the form $A(u) = 0$ of Section 2 with exact solution u, the problem of subgrid modeling can be formulated as follows: For a given function v, let v^h represent a local average on a scale h which represents the finest computational scale. We seek an equation for the average u^h of the exact solution u and by averaging the equation $A(u) = 0$ we obtain $A(u^h) + F_h(u) = 0$, where $F_h(u) \equiv (A(u))^h - A(u^h)$ has to be modeled in terms of u^h to get a modified equation $\hat{A}(u^h) = 0$. In dynamic subgrid modelling we seek to model $F_h(u)$ by extrapolation from computing $F_H(U_h)$ where $H > h$ and U_h is a computed solution on the scale h.

4.1. **Systems of convection-diffusion-reaction equations.** We consider a model problem of the form: Find $u : \Omega \times [0, T] \to R^n$ such that

$$\dot{u} + Lu = \dot{u} - \epsilon\Delta u + \beta \cdot \nabla u = f(u), \quad \forall (x,t) \in \Omega \times (0,T), \tag{4.1}$$

$$u = u_D, \quad \forall (x,t) \in \Gamma_D \times (0,T), \quad \frac{\partial u}{\partial n} = u_N, \quad \forall (x,t) \in \Gamma_N \times (0,T), \tag{4.2}$$

$$u(x,0) = u_0(x), \quad \forall x \in \Omega, \tag{4.3}$$

where $f : R^n \to R^n$ is smooth, $\Omega \subset R^d$ and $\partial\Omega = \Gamma_D \bigcup \Gamma_N$. Typically we will assume that ϵ is small and that the solution u to (4.1)–(4.3) contains a range of scales, from very small scales to large scales, induced by either the initial condition $u_0(x)$ or the differential operator L through β. Assuming we want to find an approximation of u on the scale h, representing the finest spatial computational scale, we define for each fixed t the spatial *running average* u^h of u on the scale h by

$$u^h(x,t) = \frac{1}{h^d} \int_{x_1 - h/2}^{x_1 + h/2} \ldots \int_{x_d - h/2}^{x_d + h/2} u(y,t)\; dy_1 \ldots dy_d, \tag{4.4}$$

where we note that this operator commutes with space and time differentiation. Applying this operator to (4.1)–(4.3) we find that the running average u^h satisfies the following equation (modulo boundary effects)

$$\dot{u}^h + L_h u^h = \dot{u}^h + \beta^h \cdot \nabla u^h - \epsilon \Delta u^h = f(u^h) + F_h(u), \quad u^h(x,0) = u_0^h(x), \tag{4.5}$$

where L_h is a simplified operator on the scale h resulting from approximating β by β^h and the correction term $F_h(u) = (f(u))^h - f(u^h) + L_h u^h - (Lu)^h$ contains the influence of the unresolved on u^h. We consider a computational problem without subgrid model of the form

$$\dot{u}_h + L_h u_h = f(u_h), \quad u_h(x,0) = u_0^h(x), \tag{4.6}$$

and a corresponding problem with subgrid model of the form

$$\dot{\tilde{u}}_h + L_h \tilde{u}_h = f(\tilde{u}_h) + \tilde{F}_h(\tilde{u}_h), \quad \tilde{u}_h(x,0) = u_0^h(x), \tag{4.7}$$

where $\tilde{F}_h(\tilde{u}_h)$ should approximate $F_h(u)$. In this note we will consider a subgrid model of the form $\tilde{F}_h(\tilde{u}_h) = g(F_h(\tilde{u}_h), F_{2h}(\tilde{u}_h), F_{4h}(\tilde{u}_h))$ based on averaging on the coarser scales $2h$

and $4h$, where the function g is derived based on a scale regularity assumption on $F_h(u)$. This type of model corresponds to a *scale similarity model* in LES.

4.2. **Analysis of $F_h(u)$ using the Haar MRA.** In the rest of this note we let $\Omega = [0,1]^2$ and for each $h = 2^{-i}$, with $i = 0, 1, ...$, we define a corresponding regular quadratic mesh τ^h with elements corresponding to sub-domains $\Omega_{i,k}$ with side length h. We denote the space of piecewise constant functions on τ^h by V_i, and the closure of the union of the V_j's is equal to $L_2(\Omega)$. The chain of closed subspaces $V_0 \subset V_1 \subset ... \subset V_j \subset ...$ is denoted a *Haar Multi-resolution Analysis* (MRA) of $L_2(\Omega)$.

Each V_j is spanned by the dilates and integer translates of one *scale function* $\Phi \in V_0$, that is, $V_j = \text{span}\{\Phi_{j,k}(x) = 2^j \Phi(2^j x - k)\}$. The functions $\Phi_{j,k}$ form an L_2-orthonormal basis in V_j, and we denote the orthogonal complement of V_j in V_{j+1} by W_j, which is generated by another orthonormal basis (the *wavelets*) $\Psi_{j,k}(x) = 2^j \Psi(2^j x - k)$, where $\Psi \in W_0$ is called the *mother wavelet*. $W_j = W_j^1 \oplus W_j^2 \oplus W_j^3$, where the W_j^ν's represent differences in the horizontal, vertical and diagonal directions respectively. The space $L_2(\Omega)$ can now be represented as the direct sum $L_2(\Omega) = V_0 \oplus W_0^1 \oplus ... \oplus W_0^3 \oplus ... \oplus W_j^1 \oplus ... \oplus W_j^3 \oplus ...$, and each $f \in L_2(\Omega)$ has a unique decomposition $f = f_\Phi \Phi + \sum_{j,k} f_{j,k}^1 \Psi_{j,k}^1 + ... + f_{j,k}^3 \Psi_{j,k}^3 = f_\Phi + \sum_j f_j^1 + ... + f_j^3$, where the f_j^ν's represent the contributions on the different scales 2^{-j}. For the one dimensional Haar MRA in $L_2([0,1])$, the scale function is defined by $\varphi(x) = 1$ for $x \in [0,1]$ and 0 else, and the mother wavelet is defined by $\psi(x) = 1$ for $x \in (0, 1/2)$, -1 for $x \in (1/2, 1)$ and 0 else. In two dimensions the scale function and the wavelets are tensor products of the one dimensional scale function and wavelets. For the two dimensional Haar MRA in $L_2(\Omega)$ we have the scale function $\Phi(x_1, x_2) = \varphi(x_1)\varphi(x_2)$ and the wavelets $\Psi^1(x_1, x_2) = \varphi(x_1)\psi(x_2)$, $\Psi^2(x_1, x_2) = \psi(x_1)\varphi(x_2)$, $\Psi^3(x_1, x_2) = \psi(x_1)\psi(x_2)$. For $f \in L_2(\Omega)$, we define $[f]^h = f_\varphi + \sum_{j<i} f_{j,k}^1 \Psi_{j,k}^1 + f_{j,k}^2 \Psi_{j,k}^2 + f_{j,k}^3 \Psi_{j,k}^3$. The linear mapping $L_2 \ni f \rightarrow [f]^h \in V_i$ can then be identified with the L_2-projection of f onto V_i, and we note that $[f]^h = \bar{f}^h$, where $\bar{f}^h$ is the piecewise constant function on τ^h that equals f^h in the midpoints of each element in τ^h. If we let $\bar{F}_h(u)$ denote the piecewise constant function on τ^h that equals $F_h(u)$ in the midpoints of the elements of τ^h, we have

$$\bar{F}_h(u) = [f(u)]^h - f([u]^h) - ([\beta \cdot \nabla u]^h - [\beta]^h \cdot [\nabla u]^h),$$

which for second order reaction terms $f(u)$ leads us to model covariances of the form

$$E_h(v, w) = [vw]^h - [v]^h [w]^h, \tag{4.8}$$

for given functions v and w. The following observation from [6] shows that $E_h(v, w)$ equals the sum of the mean over the elements of τ^h of the Haar coefficients, for scales finer than and equal to h: if $x \in \Omega_{i,k}$ then

$$E_h(v, w)(x) = 2^{2i} \sum_{j \geq i} (v_{j,l}^1 w_{j,l}^1 + v_{j,l}^2 w_{j,l}^2 + v_{j,l}^3 w_{j,l}^3), \quad \forall l : \Omega_{j,l} \subset \Omega_{i,k}. \tag{4.9}$$

4.3. **Scale extrapolation using self-similarity.** We base our subgrid model on an Ansatz of the form for each separate covariance

$$E_h(v, w)(x) \approx C(x) h^{\mu(x)}, \quad x \in \Omega, \tag{4.10}$$

with the coefficients $C(x)$ and $\mu(x)$ to be extrapolated. The Ansatz can be motivated from (4.9) assuming some regularity of the Haar coefficients corresponding to fractality. The Ansatz leads to the following extrapolation formula: $\tilde{F}_h(\tilde{u}_h) = \sum \tilde{E}_h(\tilde{v}_h, \tilde{w}_h)$, where

$$\tilde{E}_h(\tilde{v}_h, \tilde{w}_h) = (1 - (\frac{E_{4h}(\tilde{v}_h, \tilde{w}_h) - [E_{2h}(\tilde{v}_h, \tilde{w}_h)]^{4h}}{[E_{2h}(\tilde{v}_h, \tilde{w}_h)]^{4h} - [E_h(\tilde{v}_h, \tilde{w}_h)]^{4h}})^{-n}) \frac{[E_{2h}(\tilde{v}_h, \tilde{w}_h)]^{4h} - [E_h(\tilde{v}_h, \tilde{w}_h)]^{4h}}{\frac{E_{4h}(\tilde{v}_h, \tilde{w}_h) - [E_{2h}(\tilde{v}_h, \tilde{w}_h)]^{4h}}{[E_{2h}(\tilde{v}_h, \tilde{w}_h)]^{4h} - [E_h(\tilde{v}_h, \tilde{w}_h)]^{4h}} - 1},$$

and $2^{-(n+1)}$ is the finest scale present in the exact solution.

4.4. **Applications.** In all examples we will have $h = 2^{-5}$. We construct two dimensional fractal data as sums of local tensor products of the one dimensional fractal *Weierstrass function* $W_{\gamma,\delta}(x) = \gamma \sum_{j=0}^{N} 2^{-j\delta} \sin(2^j \cdot 2\pi x)$, where we let $\gamma = \delta = 0.1$ in all examples.

4.4.1. *Volterra-Lotka (VL).* We consider a reaction dominated problem of the form

$$\begin{aligned} &\dot{u_1} - \epsilon \Delta u_1 = u_1(1 - u_2), \quad \dot{u_2} - \epsilon \Delta u_2 + \beta \cdot \nabla u_2 = u_2(u_1 - 1), \\ &\frac{\partial u}{\partial n}|_{\partial\Omega} = 0, \quad u(x,0) = (W^{2D}_{\gamma,\delta}(x), 1), \end{aligned}$$

where $\epsilon = 10^{-6}$, which corresponds to the classical Volterra-Lotka system with small diffusion and convection in one component. We have $F_h(u) = (-(u_1u_2)^h + u_1^h u_2^h, (u_1u_2)^h - u_1^h u_2^h)$, and with subgrid model we have that $\tilde{F}_h(\tilde{u}_h) = (-\tilde{E}_h(\tilde{u}_1, \tilde{u}_2), \tilde{E}_h(\tilde{u}_1, \tilde{u}_2))$, with $n = 4$ (reference scale minus computational scale). For these problems we use a central difference-Crank-Nicolson scheme for the midpoints of the elements, where we regard these midpoint values to represent a piecewise constant approximation over the elements, and the reference scale is 2^{-9} in the computation of the error. The solutions are oscillating and both u_1 and u_2 are fractal for $t > 0$, even though $u_2(x, 0) = 1$. We want to approximate u^h, and the errors $\|u^h - U\|$ are plotted in Fig.1-2, where U is the error without model ('*'), without model but computed on the finer scale $h/2$ and then projected onto the scale h ('o'), with the subgrid model ('+'), and with a simplified model $\tilde{F}_h(\tilde{u}_h) = F_{2h}(\tilde{u}_h, \tilde{u}_h)$ ('triangles'). We first let $\beta = 0$ and compute to $T = 2$, then we let β be a rotational mixing of order h: $\beta = h\ (\sin(\pi x_1)\cos(\pi x_2), -\cos(\pi x_1)\sin(\pi x_2))$ and here we only compute to $T = 1$, since after $T = 1$ the subgrid scales are dominated by the convective streaks due to β. We study the error for each component individually, and for $\beta = 0$ we find that the solution with the subgrid model is the best for both components, even though the modeling errors are smaller in u_2 since $u_2(x, 0)$ is constant. For $\beta \neq 0$ the solution with the subgrid model is best for u_1 but, because of the convection, subgrid scales in u_2 do not develop and the solutions with subgrid models does not differ significantly from the solution without subgrid model. The solution on $h/2$ is better since the numerical error is then reduced.

4.4.2. *Fractal convection (FC).* We also consider a convection dominated problem of the form

$$\dot{u} + \beta \cdot \nabla u - \epsilon \Delta u = 1, \quad u|_{x_1=0,x_2=0} = 0, \quad \frac{\partial u}{\partial n}|_{x_1=1,x_2=1} = 0, \quad u(x,0) = 0, \tag{4.11}$$

for $\beta = (W^{2D}_{\gamma,\delta}, W^{2D}_{\gamma,\delta})$ and $\epsilon = 10^{-3}$, which we solve by a Streamline Diffusion cG(1)cG(1) method [1] with bilinear elements. In the computation of the error the reference scale is 2^{-8}. The solution is in this case relatively smooth since the fractal β is only acting on the derivatives of the solution. We have $F_h(u) = \beta^h \cdot (\nabla u)^h - (\beta \cdot \nabla u)^h$, and the error in the solution with the subgrid model is smaller than in the solution without subgrid model.

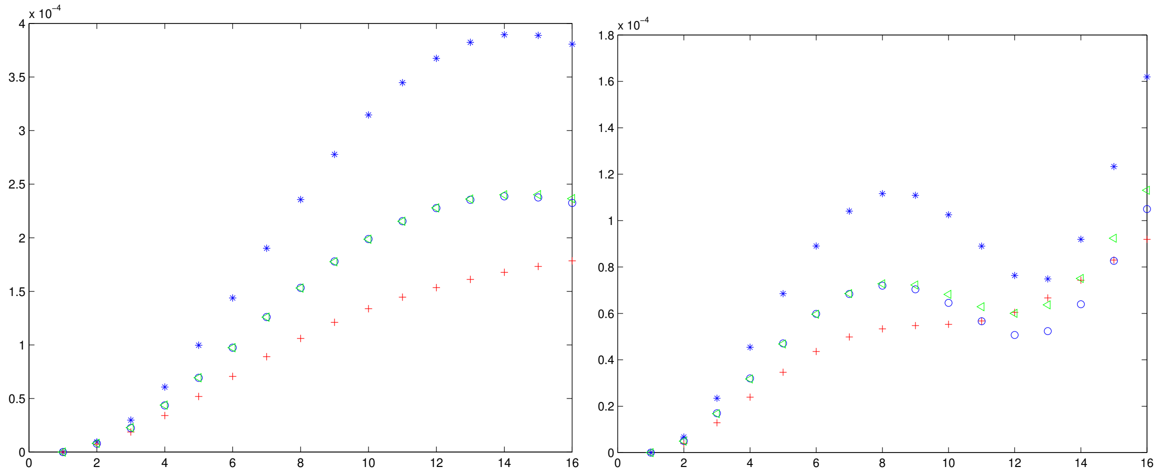

Figure 1: L_2-error in u_1(left) and u_2 (right), for VL with $\beta = 0$.

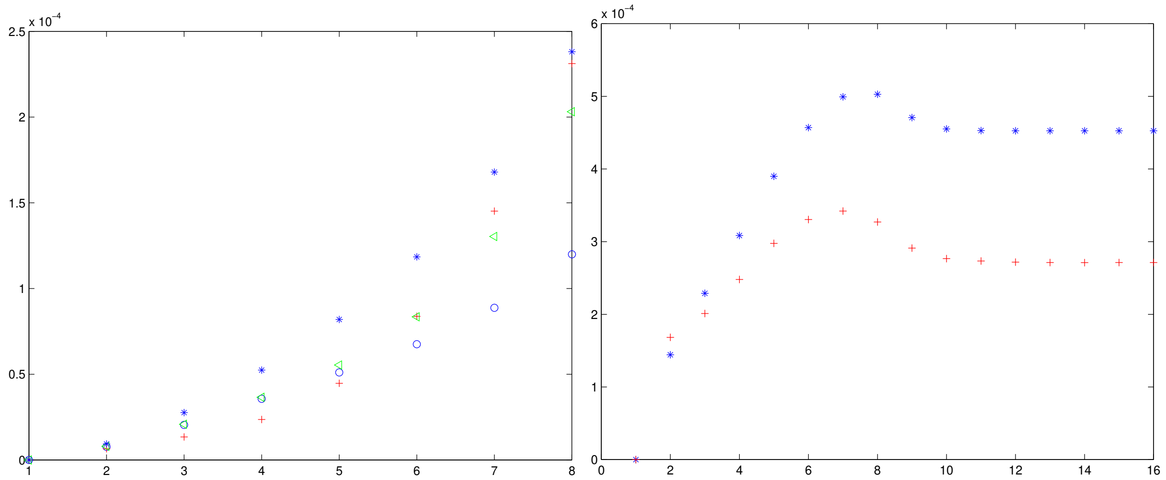

Figure 2: L_2-error in u_1 for VL with $\beta \neq 0$ (left), and u for FC (right).

References

[1] K. Eriksson, D. Estep, P. Hansbo and C. Johnson. (1995) Introduction to Adaptive Methods for Differential Equations. *Acta Numerica*, 105–158.

[2] K. Eriksson, D. Estep, P. Hansbo and C. Johnson. (1996) *Computational Differential Equations*. Studentlitteratur, Lund, Sweden.

[3] K. Eriksson, D. Estep and C. Johnson. (2001) *Applied Mathematics: Body and Soul*, Springer.

[4] K. Eriksson, D. Estep, P. Hansbo, C. Johnson. *Advanced Computational Differential Equations*, to appear 2002.

[5] J. Hoffman, C. Johnson and S. Bertoluzza. (1999) Dynamic Subgrid Modeling I. *Preprint. Chalmers Finite Element Center*, 1999, to appear in Comp. Meth. Appl. Mech. Engrng.

[6] J. Hoffman. (2000) Dynamic Subgrid Modeling II. *Preprint. Chalmers Finite Element Center.*

[7] J. Hoffman. (2001) Dynamic Subgrid Modeling for Convection-Diffusion Equations with Fractal Coefficients. *Multiscale and Multiresolution Methods*, Springer Lecture Series in Engineering, Springer Verlag.

[8] D. Estep and C. Johnson. (1998) The pointwise computability of the Lorenz system. M^3AS **8**, 1277–1306.

[9] M.G. Larson. (2000) Error Growth and A Posteriori Error Estimates for Conservative Galerkin Approximations of Periodic Orbits in Hamiltonian Systems M^3AS **10**, 31–46.

[10] A. Logg. (1998) A Multi-Adaptive ODE-Solver. *Preprint Chalmers Finite Element Center.*

[11] A. Logg. (2001) Multi-Adaptive Galerkin Methods for ODEs I. Submitted to SIAM J. Sci. Comput.

[12] A. Logg. (2001) Multi-Adaptive Galerkin Methods for ODEs II: Implementation & Applications. Submitted to SIAM J. Sci. Comput.

[13] R. Becker and R. Rannacher. (2001) An optimal control approach to error control and mesh adaption in finite element methods. *Acta Numerica.*

Chalmers Finite Element Center Preprints

2000–01 *Adaptive Finite Element Methods for the Unsteady Maxwell's Equations*
Johan Hoffman

2000–02 *A Multi-Adaptive ODE-Solver*
Anders Logg

2000–03 *Multi-Adaptive Error Control for ODEs*
Anders Logg

2000–04 *Dynamic Computational Subgrid Modeling* (Licentiate Thesis)
Johan Hoffman

2000–05 *Least-Squares Finite Element Methods for Electromagnetic Applications* (Licentiate Thesis)
Rickard Bergström

2000–06 *Discontinuous Galerkin Methods for Incompressible and Nearly Incompressible Elasticity by Nitsche's Method*
Peter Hansbo and Mats G. Larson

2000–07 *A Discountinuous Galerkin Method for the Plate Equation*
Peter Hansbo and Mats G. Larson

2000–08 *Conservation Properties for the Continuous and Discontinuous Galerkin Methods*
Mats G. Larson and A. Jonas Niklasson

2000–09 *Discontinuous Galerkin and the Crouzeix-Raviart element: Application to elasticity*
Peter Hansbo and Mats G. Larson

2000–10 *Pointwise A Posteriori Error Analysis for an Adaptive Penalty Finite Element Method for the Obstacle Problem*
Donald A. French, Stig Larson and Ricardo H. Nochetto

2000–11 *Global and Localised A Posteriori Error Analysis in the Maximum Norm for Finite Element Approximations of a Convection-Diffusion Problem*
Mats Boman

2000–12 *A Posteriori Error Analysis in the Maximum Norm for a Penalty Finite Element Method for the Time-Dependent Obstacle Problem*
Mats Boman

2000–13 *A Posteriori Error Analysis in the Maximum Norm for Finite Element Approximations of a Time-Dependent Convection-Diffusion Problem*
Mats Boman

2001–01 *A Simple Nonconforming Bilinear Element for the Elasticity Problem*
Peter Hansbo and Mats G. Larson

2001–02 *The $\mathcal{LL}^*$ Finite Element Method and Multigrid for the Magnetostatic Problem*
Rickard Bergström, Mats G. Larson, and Klas Samuelsson

2001–03 *The Fokker-Planck Operator as an Asymptotic Limit in Anisotropic Media*
Mohammad Asadzadeh

2001–04 *A Posteriori Error Estimation of Functionals in Elliptic Problems: Experiments*
Mats G. Larson and A. Jonas Niklasson

2001–05 *A Note on Energy Conservation for Hamiltonian Systems Using Continuous Time Finite Elements*
Peter Hansbo

2001–06 *Stationary Level Set Method for Modelling Sharp Interfaces in Groundwater Flow*
Nahidh Sharif and Nils-Erik Wiberg

2001–07 *Integration methods for the calculation of the magnetostatic field due to coils*
Marzia Fontana

2001–08 *Adaptive finite element computation of 3D magnetostatic problems in potential formulation*
Marzia Fontana

2001–09 *Multi-Adaptive Galerkin Methods for ODEs I: Theory & Algorithms*
Anders Logg

2001–10 *Multi-Adaptive Galerkin Methods for ODEs II: Applications*
Anders Logg

2001–11 *Energy norm a posteriori error estimation for discontinuous Galerkin methods*
Roland Becker, Peter Hansbo, and Mats G. Larson

2001–12 *Analysis of a family of discontinuous Galerkin methods for elliptic problems: the one dimensional case*
Mats G. Larson and A. Jonas Niklasson

2001–13 *Analysis of a nonsymmetric discontinuous Galerkin method for elliptic problems: stability and energy error estimates*
Mats G. Larson and A. Jonas Niklasson

2001–14 *A hybrid method for the wave equation*
Larisa Beilina, Klas Samuelsson, Krister Åhlander

2001–15 *A Finite Element Method for Domain Decomposition with Non-Matching Grids*
Roland Becker, Peter Hansbo and Rolf Stenberg

2001–16 *Application of stable FEM-FDTD hybrid to scattering problems*
Thomas Rylander and Anders Bondeson

2001–17 *Eddy current computations using adaptive grids and edge elements*
Y. Q. Liu, A. Bondeson, R. Bergström, C. Johnson, M. G. Larson, and K. Samuelsson

2001–18 *Adaptive finite element methods for incompressible fluid flow*
J. Hoffman, C. Johnson

2001–19 *Dynamic subgrid modeling for time dependent convection–diffusion–reaction equations with fractal solutions*
J. Hoffman

2001–20 *Topics in adaptive computational methods for differential equations*
Claes Johnson, Johan Hoffman and Anders Logg

These preprints can be obtained from

`www.phi.chalmers.se/preprints`