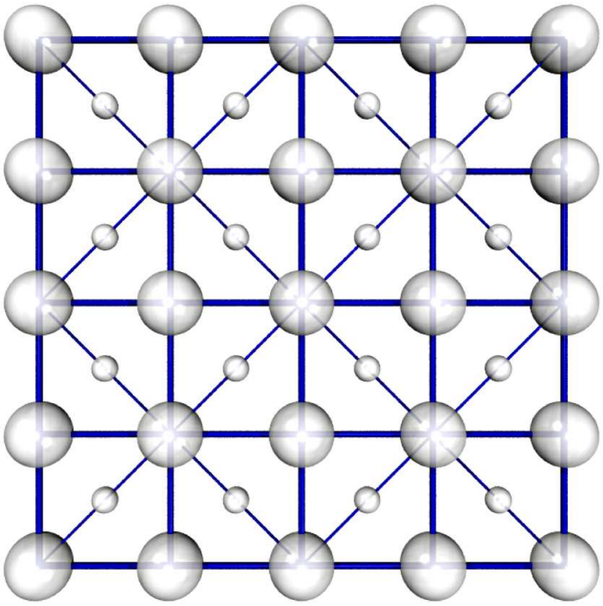
CHALMERS

FINITE ELEMENT CENTER



PREPRINT 2004–15

Computational modeling of dynamical systems

Johan Jansson, Claes Johnson, and Anders Logg

Chalmers Finite Element Center
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg Sweden 2004

CHALMERS FINITE ELEMENT CENTER

Preprint 2004–15

Computational modeling of dynamical systems

Johan Jansson, Claes Johnson, and Anders Logg

CHALMERS

Chalmers Finite Element Center
Chalmers University of Technology
SE–412 96 Göteborg Sweden
Göteborg, April 2004

Computational modeling of dynamical systems
Johan Jansson, Claes Johnson, and Anders Logg
NO 2004–15
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 2004

COMPUTATIONAL MODELING OF DYNAMICAL SYSTEMS

JOHAN JANSSON, CLAES JOHNSON, AND ANDERS LOGG

Abstract. In this short note, we discuss the basic approach to computational modeling of dynamical systems. If a dynamical system contains multiple time scales, ranging from very fast to slow, computational solution of the dynamical system can be very costly. By resolving the fast time scales in a short time simulation, a model for the effect of the small time scale variation on large time scales can be determined, making solution possible on a long time interval. This process of computational modeling can be completely automated. Two examples are presented, including a simple model problem oscillating at a time scale of 10^{-9} computed over the time interval $[0, 100]$, and a lattice consisting of large and small point masses.

1. Introduction

We consider a dynamical system of the form

$$\begin{aligned} \dot{u}(t) &= f(u(t), t), \quad t \in (0, T], \\ u(0) &= u_0, \end{aligned} \tag{1.1}$$

where $u : [0, T] \to \mathbb{R}^N$ is the solution to be computed, $u_0 \in \mathbb{R}^N$ a given initial value, $T > 0$ a given final time, and $f : \mathbb{R}^N \times (0, T] \to \mathbb{R}^N$ a given function that is Lipschitz-continuous in u and bounded. We consider a situation where the exact solution u varies on different time scales, ranging from very fast to slow. Typical examples include meteorological models for weather prediction, with fast time scales on the range of seconds and slow time scales on the range of years, protein folding represented by a molecular dynamics model of the form (1.1), with fast time scales on the range of femtoseconds and slow time scales on the range of microseconds, or turbulent flow with a wide range of time scales.

To make computation feasible in a situation where computational resolution of the fast time scales would be prohibitive because of the small time steps required, the given model (1.1) containing the fast time scales needs to be replaced with a *reduced model* for the variation of the solution u of (1.1) on resolvable time scales. As discussed below, the key step is to correctly model the effect of the variation at the fast time scales on the variation on slow time scales.

Date: April 22, 2004.

Key words and phrases. Modeling, dynamical system, reduced model, automation.

Johan Jansson, *email*: johanjan@math.chalmers.se. Claes Johnson, *email*: claes@math.chalmers.se. Anders Logg, *email*: logg@math.chalmers.se. Department of Computational Mathematics, Chalmers University of Technology, SE–412 96 Göteborg, Sweden.

The problem of model reduction is very general and various approaches have been taken, see e.g. [8] and [6]. We present below a new approach to model reduction, based on resolving the fast time scales in a short time simulation and determining a model for the effect of the small time scale variation on large time scales. This process of computational modeling can be completely *automated* and the validity of the reduced model can be evaluated a posteriori.

2. A simple model problem

We consider a simple example illustrating the basic aspects: Find $u = (u_1, u_2) : [0, T] \rightarrow \mathbb{R}^2$, such that

$$\begin{aligned} \ddot{u}_1 + u_1 - u_2^2/2 &= 0 \quad \text{on } (0, T], \\ \ddot{u}_2 + \kappa u_2 &= 0 \quad \text{on } (0, T], \\ u(0) = (0, 1) \quad \dot{u}(0) &= (0, 0), \end{aligned} \tag{2.1}$$

which models a moving unit point mass M_1 connected through a soft spring to another unit point mass M_2, with M_2 moving along a line perpendicular to the line of motion of M_1, see Figure 1. The second point mass M_2 is connected to a fixed support through a very stiff spring with spring constant $\kappa = 10^{18}$ and oscillates rapidly on a time scale of size $1/\sqrt{\kappa} = 10^{-9}$. The oscillation of M_2 creates a force $\sim u_2^2$ on M_1 proportional to the elongation of the spring connecting M_2 to M_1 (neglecting terms of order u_2^4).

The short time scale of size 10^{-9} requires time steps of size $\sim 10^{-10}$ for full resolution. With $T = 100$, this means a total of $\sim 10^{12}$ time steps for solution of (2.1). However, by replacing (2.1) with a reduced model where the fast time scale has been removed, it is possible to compute the (averaged) solution of (2.1) with time steps of size ~ 0.1 and consequently only a total of 10^3 time steps.

3. Taking averages to obtain the reduced model

Having realized that point-wise resolution of the fast time scales of the exact solution u of (1.1) may sometimes be computationally very expensive or even impossible, we seek instead to compute a time average $\bar{u}$ of u, defined by

$$\bar{u}(t) = \frac{1}{\tau} \int_{-\tau/2}^{\tau/2} u(t + s)\, ds, \quad t \in [\tau/2, T - \tau/2], \tag{3.1}$$

where $\tau > 0$ is the size of the average. The average $\bar{u}$ can be extended to $[0, T]$ in various ways. We consider here a constant extension, i.e., we let $\bar{u}(t) = \bar{u}(\tau/2)$ for $t \in [0, \tau/2)$, and let $\bar{u}(t) = \bar{u}(T - \tau/2)$ for $t \in (T - \tau/2, T]$.

We now seek a dynamical system satisfied by the average $\bar{u}$ by taking the average of (1.1). We obtain

$$\dot{\bar{u}}(t) = \bar{\dot{u}}(t) = \overline{f(u, \cdot)}(t) = f(\bar{u}(t), t) + (\overline{f(u, \cdot)}(t) - f(\bar{u}(t), t)),$$

or

$$\dot{\bar{u}}(t) = f(\bar{u}(t), t) + \bar{g}(u, t), \tag{3.2}$$

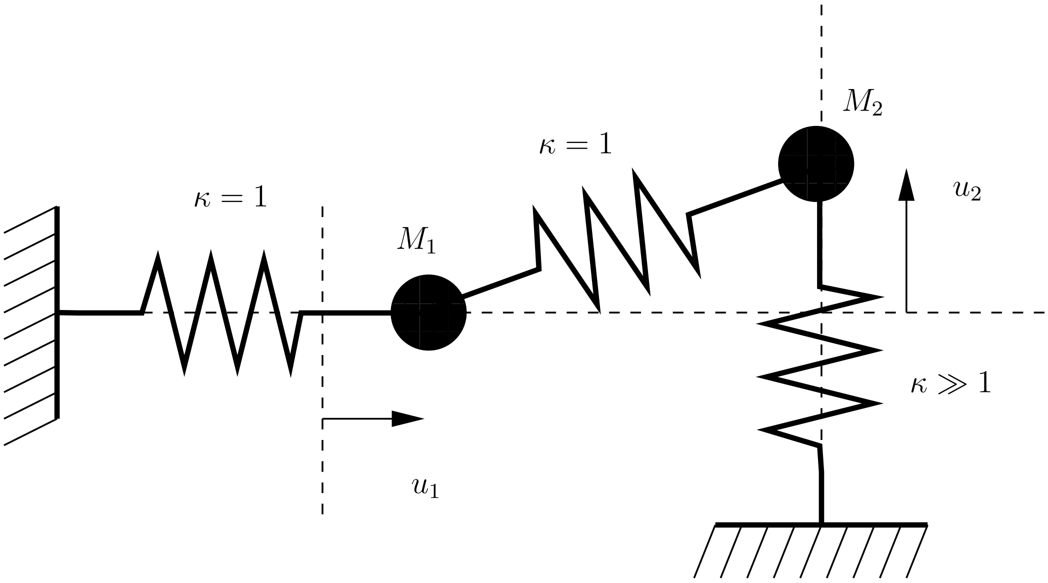


FIGURE 1. A simple mechanical system with large time scale ~ 1 and small time scale $\sim 1/\sqrt{\kappa}$.

where the *variance* $\bar{g}(u,t) = \overline{f(u,\cdot)}(t) - f(\bar{u}(t),t)$ accounts for the effect of small scales on time scales larger than τ. (Note that we may extend (3.2) to $(0,T]$ by defining $\bar{g}(u,t) = -f(\bar{u}(t),t)$ on $(0,\tau/2] \cup (T-\tau/2,T]$.)

We now seek to model the variance $\bar{g}(u,t)$ in the form $\bar{g}(u,t) \approx \tilde{g}(\bar{u}(t),t)$ and replace (3.2) and thus (1.1) by

$$\begin{aligned} \dot{\tilde{u}}(t) &= f(\tilde{u}(t),t) + \tilde{g}(\tilde{u}(t),t), \quad t \in (0,T], \\ \tilde{u}(0) &= \bar{u}_0, \end{aligned} \tag{3.3}$$

where $\bar{u}_0 = \bar{u}(0) = \bar{u}(\tau/2)$. We refer to this system as the *reduced model* with *subgrid model* $\tilde{g}$ corresponding to (1.1).

To summarize, if the solution u of the full dynamical system (1.1) is computationally unresolvable, we aim at computing the average $\bar{u}$ of u. However, since the variance $\bar{g}$ in the averaged dynamical system (3.2) is unknown, we need to solve the reduced model (3.3) for $\tilde{u} \approx \bar{u}$ with an approximate subgrid model $\tilde{g} \approx \bar{g}$. Solving the reduced model (3.3) using e.g. a Galerkin finite element method, we obtain an approximate solution $U \approx \tilde{u} \approx \bar{u}$. Note that we may not expect U to be close to u point-wise in time, while we hope that U is close to $\bar{u}$ point-wise.

4. Modeling the variance

There are two basic approaches to the modeling of the variance $\bar{g}(u,t)$ in the form $\tilde{g}(\tilde{u}(t),t)$; (i) scale-extrapolation or (ii) local resolution. In (i), a sequence of solutions is computed with increasingly fine resolution, but without resolving the fastest time scales.

A model for the effects of the fast unresolvable scales is then determined by extrapolation from the sequence of computed solutions, see e.g. [3]. In (ii), the approach followed below, the solution u is computed accurately over a short time period, resolving the fastest time scales. The reduced model is then obtained by computing the variance

$$\bar{g}(u,t) = \overline{f(u,\cdot)}(t) - f(\bar{u}(t),t) \tag{4.1}$$

and then determining $\tilde{g}$ for the remainder of the time interval such that $\tilde{g}(\tilde{u}(t),t) \approx \bar{g}(u,t)$.

For the simple model problem (2.1), which we can write in the form (1.1) by introducing the two new variables $u_3 = \dot{u}_1$ and $u_4 = \dot{u}_2$ with

$$f(u,\cdot) = (u_3, u_4, -u_1 + u_2^2/2, -\kappa u_2),$$

we note that $\bar{u}_2 \approx 0$ (for $\sqrt{\kappa}\tau$ large) while $\overline{u_2^2} \approx 1/2$. By the linearity of f_1, f_2, and f_4, the (approximate) reduced model takes the form

$$\begin{aligned} \ddot{\tilde{u}}_1 + \tilde{u}_1 - 1/4 &= 0 \quad \text{on } (0,T], \\ \ddot{\tilde{u}}_2 + \kappa\tilde{u}_2 &= 0 \quad \text{on } (0,T], \\ \tilde{u}(0) = (0,0), \quad \dot{\tilde{u}}(0) &= (0,0), \end{aligned} \tag{4.2}$$

with solution $\tilde{u}(t) = (\frac{1}{4}(1-\cos t), 0)$.

In general, the reduced model is constructed with subgrid model $\tilde{g}$ varying on resolvable time scales. In the simplest case, it is enough to model $\tilde{g}$ with a constant and repeatedly checking the validity of the model by comparing the reduced model (3.3) with the full model (1.1) in a short time simulation. Another possibility is to use a piecewise polynomial representation for the subgrid model $\tilde{g}$.

5. Solving the reduced system

Although the presence of small scales has been decreased in the reduced system (3.3), the small scale variation may still be present. This is not evident in the reduced system (4.2) for the simple model problem (2.1), where we made the approximation $\tilde{u}_2(0) = 0$. In practice, however, we compute $\tilde{u}_2(0) = \frac{1}{\tau}\int_0^\tau u_2(t)\,dt = \frac{1}{\tau}\int_0^\tau \cos(\sqrt{\kappa}t)\,dt \sim 1/(\sqrt{\kappa}\tau)$ and so $\tilde{u}_2$ oscillates at the fast time scale $1/\sqrt{\kappa}$ with amplitude $1/(\sqrt{\kappa}\tau)$.

To remove these oscillations, the reduced system needs to be stabilized by introducing damping of high frequencies. Following the general approach presented in [5], a least squares stabilization is added in the Galerkin formulation of the reduced system (3.3) in the form of a modified test function. As a result, damping is introduced for high frequencies without affecting low frequencies.

Alternatively, components such as u_2 in (4.2) may be *inactivated*, corresponding to a subgrid model of the form $\tilde{g}_2(\tilde{u},\cdot) = -f_2(\tilde{u},\cdot)$. We take this simple approach for the example problems presented below.

6. Error analysis

The validity of a proposed subgrid model may be checked a posteriori. To analyze the modeling error introduced by approximating the variance $\bar{g}$ with the subgrid model $\tilde{g}$, we introduce the *dual problem*

$$
\begin{aligned}
-\dot{\phi}(t) &= J(\bar{u}, U, t)^\top \phi(t), \quad t \in [0, T), \\
\phi(T) &= \psi,
\end{aligned}
\tag{6.1}
$$

where J denotes the Jacobian of the right-hand side of the dynamical system (1.1) evaluated at a mean value of the average $\bar{u}$ and the computed numerical (finite element) solution $U \approx \tilde{u}$ of the reduced system (3.3),

$$
J(\bar{u}, U, t) = \int_0^1 \frac{\partial f}{\partial u}(s\bar{u}(t) + (1-s)U(t), t)\, ds,
\tag{6.2}
$$

and where ψ is initial data for the backward dual problem.

To estimate the error $\bar{e} = U - \bar{u}$ at final time, we note that $\bar{e}(0) = 0$ and $\dot{\phi} + J(\bar{u}, U, \cdot)^\top \phi = 0$, and write

$$
\begin{aligned}
(\bar{e}(T), \psi) &= (\bar{e}(T), \psi) - \int_0^T (\dot{\phi} + J(\bar{u}, U, \cdot)^\top \phi, \bar{e})\, dt \\
&= \int_0^T (\phi, \dot{\bar{e}} - J\bar{e})\, dt = \int_0^T (\phi, \dot{U} - \dot{\bar{u}} - f(U, \cdot) + f(\bar{u}, \cdot))\, dt \\
&= \int_0^T (\phi, \dot{U} - f(U, \cdot) - \tilde{g}(U, \cdot))\, dt + \int_0^T (\phi, \tilde{g}(U, \cdot) - \bar{g}(u, \cdot))\, dt \\
&= \int_0^T (\phi, \tilde{R}(U, \cdot))\, dt + \int_0^T (\phi, \tilde{g}(U, \cdot) - \bar{g}(u, \cdot))\, dt.
\end{aligned}
$$

The first term, $\int_0^T (\phi, \tilde{R}(U, \cdot))\, dt$, in this *error representation* corresponds to the *discretization error* $U - \tilde{u}$ for the numerical solution of (3.3). If a Galerkin finite element method is used (see [1, 2]), the *Galerkin orthogonality* expressing the orthogonality of the residual $\tilde{R}(U, \cdot) = \dot{U} - f(U, \cdot) - \tilde{g}(U, \cdot)$ to a space of test functions can be used to subtract a test space interpolant $\pi\phi$ of the dual solution ϕ. In the simplest case of the cG(1) method for a partition of the interval $(0, T]$ into M subintervals $I_j = (t_{j-1}, t_j]$, each of length $k_j = t_j - t_{j-1}$, we subtract a piecewise constant interpolant to obtain

$$
\begin{aligned}
\int_0^T (\phi, \tilde{R}(U, \cdot))\, dt = \int_0^T (\phi - \pi\phi, \tilde{R}(U, \cdot))\, dt &\leq \sum_{j=1}^M k_j \max_{I_j} \|\tilde{R}(U, \cdot)\|_{l_2} \int_{I_j} \|\dot{\phi}\|_{l_2}\, dt \\
&\leq S^{[1]}(T) \max_{[0,T]} \|k\tilde{R}(U, \cdot)\|_{l_2},
\end{aligned}
$$

where the *stability factor* $S^{[1]}(T) = \int_0^T \|\dot{\phi}\|_{l_2}\, dt$ measures the sensitivity to discretization errors for the given output quantity $(\bar{e}(T), \psi)$.

The second term, $\int_0^T (\phi, \tilde{g}(U,\cdot) - \bar{g}(u,\cdot))\, dt$, in the error representation corresponds to the *modeling error* $\tilde{u} - \bar{u}$. The sensitivity to modeling errors is measured by the stability factor $S^{[0]}(T) = \int_0^T \|\phi\|_{l_2}\, dt$. We notice in particular that if the stability factor $S^{[0]}(T)$ is of moderate size, a reduced model of the form (3.3) for $\tilde{u} \approx \bar{u}$ may be constructed.

We thus obtain the error estimate

$$|(\bar{e}(T), \psi)| \leq S^{[1]}(T) \max_{[0,T]} \|k\tilde{R}(U,\cdot)\|_{l_2} + S^{[0]}(T) \max_{[0,T]} \|\tilde{g}(U,\cdot) - \bar{g}(u,\cdot)\|_{l_2}, \tag{6.3}$$

including both discretization and modeling errors. The initial data ψ for the dual problem (6.1) is chosen to reflect the desired output quantity, e.g. $\psi = (1, 0, \ldots, 0)$ to measure the error in the first component of U.

To estimate the modeling error, we need to estimate the quantity $\tilde{g} - \bar{g}$. This estimate is obtained by repeatedly solving the full dynamical system (1.1) at a number of control points and comparing the subgrid model $\tilde{g}$ with the computed variance $\bar{g}$. As initial data for the full system at a control point, we take the computed solution $U \approx \bar{u}$ at the control point and add a perturbation of appropriate size, with the size of the perturbation chosen to reflect the initial oscillation at the fastest time scale.

7. Numerical results

We present numerical results for two model problems, including the simple model problem (2.1), computed with DOLFIN version 0.4.10 [4]. With the option *automatic modeling* set, DOLFIN automatically creates the reduced model (3.3) for a given dynamical system of the form (1.1) by resolving the full system in a short time simulation and then determining a constant subgrid model $\bar{g}$. Components with constant average, such as u_2 in (2.1), are automatically marked as inactive and are kept constant throughout the simulation. The automatic modeling implemented in DOLFIN is rudimentary and many improvements are possible, but it represents a first attempt at the automation of modeling, following the directions for the *automation of computational mathematical modeling* presented in [7].

7.1. **The simple model problem.** The solution for the two components of the simple model problem (2.1) is shown in Figure 2 for $\kappa = 10^{18}$ and $\tau = 10^{-7}$. The value of the subgrid model $\bar{g}_1$ is automatically determined to $0.2495 \approx 1/4$.

7.2. **A lattice with internal vibrations.** The second example is a lattice consisting of a set of p^2 large and $(p-1)^2$ small point masses connected by springs of equal stiffness $\kappa = 1$, as shown in Figure 3 and Figure 4. Each large point mass is of size $M = 100$ and each small point mass is of size $m = 10^{-12}$, giving a large time scale of size ~ 10 and a small time scale of size $\sim 10^{-6}$.

The fast oscillations of the small point masses make the initially stationary structure of large point masses contract. Without resolving the fast time scales and ignoring the subgrid model, the distance D between the lower left large point mass at $x = (0,0)$ and the upper right large point mass at $x = (1,1)$ remains constant, $D = \sqrt{2}$. In Figure 6, we show the computed solution with $\tau = 10^{-4}$, which manages to correctly capture the

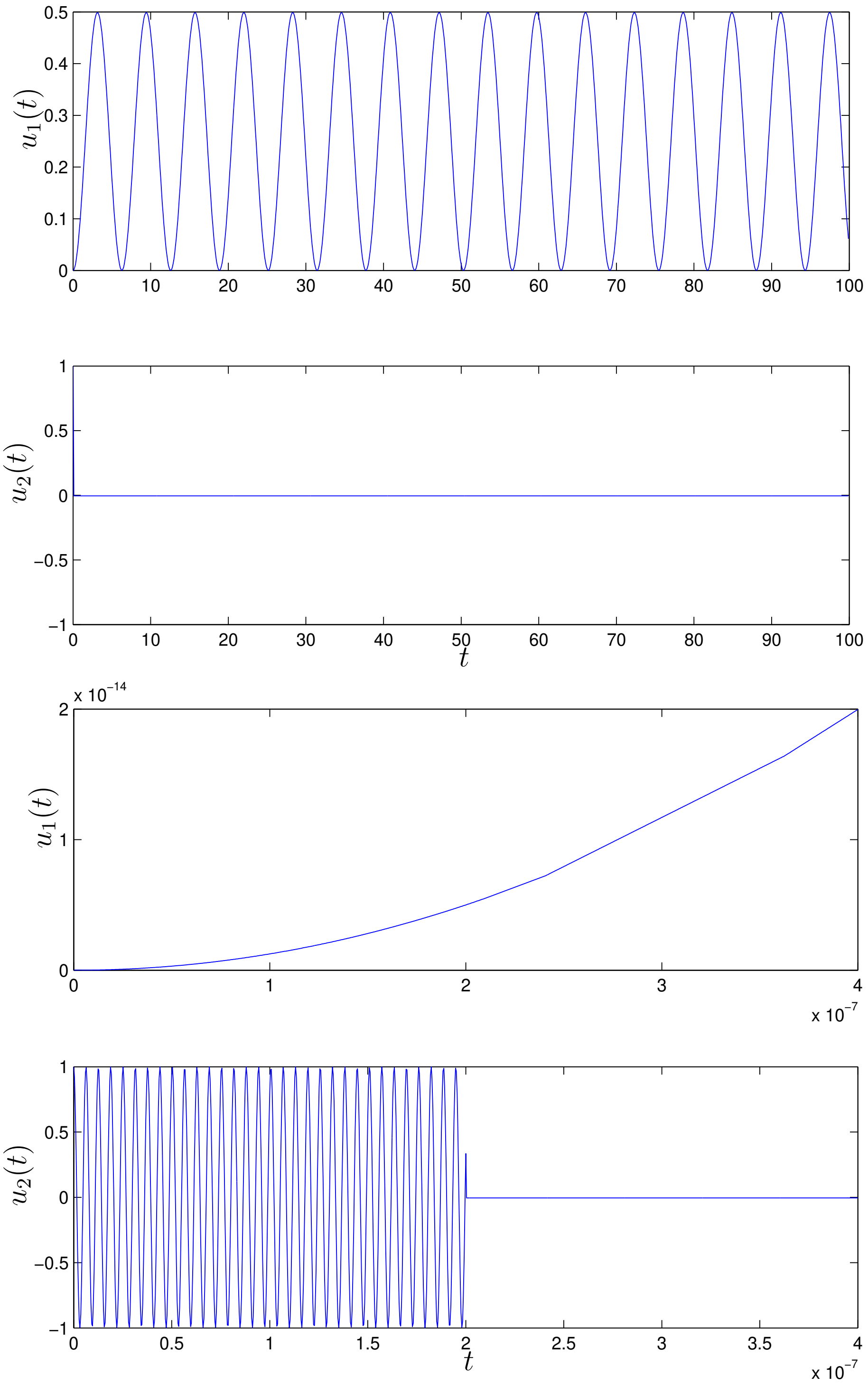


FIGURE 2. The solution of the simple model problem (2.1) on $[0, 100]$ (above) and on $[0, 4 \cdot 10^{-7}]$ (below). The automatic modeling is activated at time $t = 2\tau = 2 \cdot 10^{-7}$.

oscillation in the diameter D of the lattice as a consequence of the internal vibrations at time scale 10^{-6}.

With a constant subgrid model $\bar{g}$ as in the example, the reduced model stays accurate until the configuration of the lattice has changed sufficiently. When the change becomes too large, the reduced model can no longer give an accurate representation of the full system, as shown in Figure 5. At this point, the reduced model needs to be reconstructed in a new short time simulation.

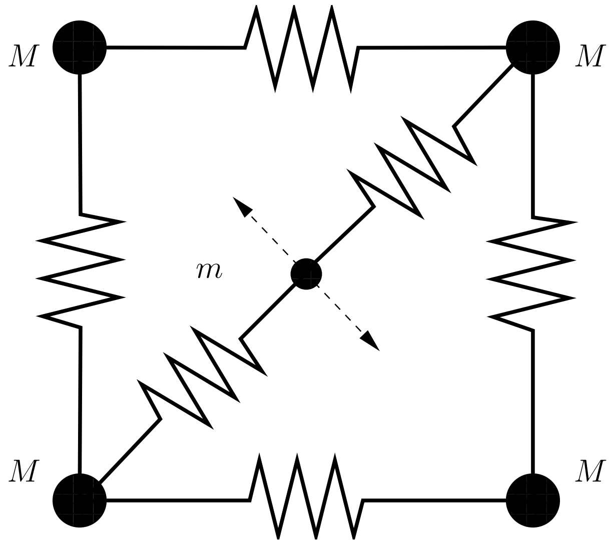


FIGURE 3. Detail of the lattice. The arrows indicate the direction of vibration perpendicular to the springs connecting the small mass to the large masses.

REFERENCES

[1] K. ERIKSSON, D. ESTEP, P. HANSBO, AND C. JOHNSON, *Introduction to adaptive methods for differential equations*, Acta Numerica, (1995), pp. 105–158.
[2] ——, *Computational Differential Equations*, Cambridge University Press, 1996.
[3] J. HOFFMAN, *Computational Modeling of Complex Flows*, PhD thesis, Chalmers University of Technology, 2002.
[4] J. HOFFMAN AND A. LOGG ET AL., *DOLFIN*, http://www.phi.chalmers.se/dolfin/.
[5] J. HOFFMAN AND C. JOHNSON, *Computability and adaptivity in CFD*, to appear in Encyclopedia of Computational Mechanics, (2004).
[6] H.-O. KREISS, *Problems with different time scales*, Acta Numerica, 1 (1991).
[7] A. LOGG, *Automation of Computational Mathematical Modeling*, PhD thesis, Chalmers University of Technology, 2004.
[8] A. RUHE AND D. SKOOGH, *Rational Krylov algorithms for eigenvalue computation and model reduction*, in Applied Parallel Computing — Large Scale Scientific and Industrial Problems, B. Kågström, J. Dongarra, E. Elmroth, and J. Waśniewski, eds., Lecture Notes in Computer Science, No. 1541, 1998, pp. 491–502.

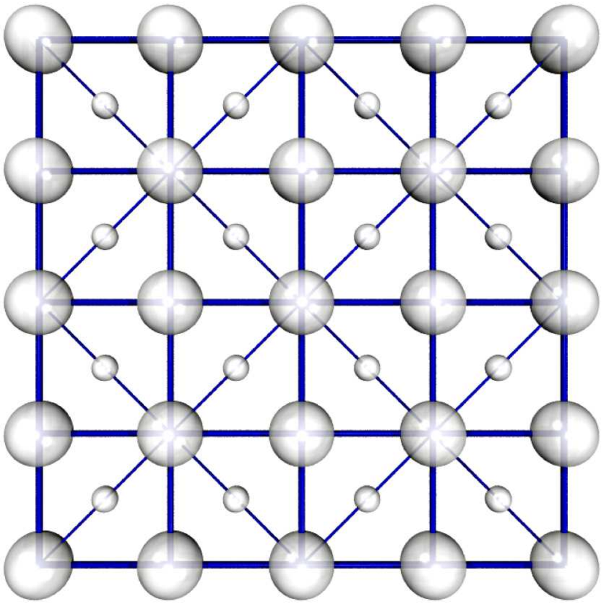


FIGURE 4. Lattice consisting of p^2 large masses and $(p-1)^2$ small masses.

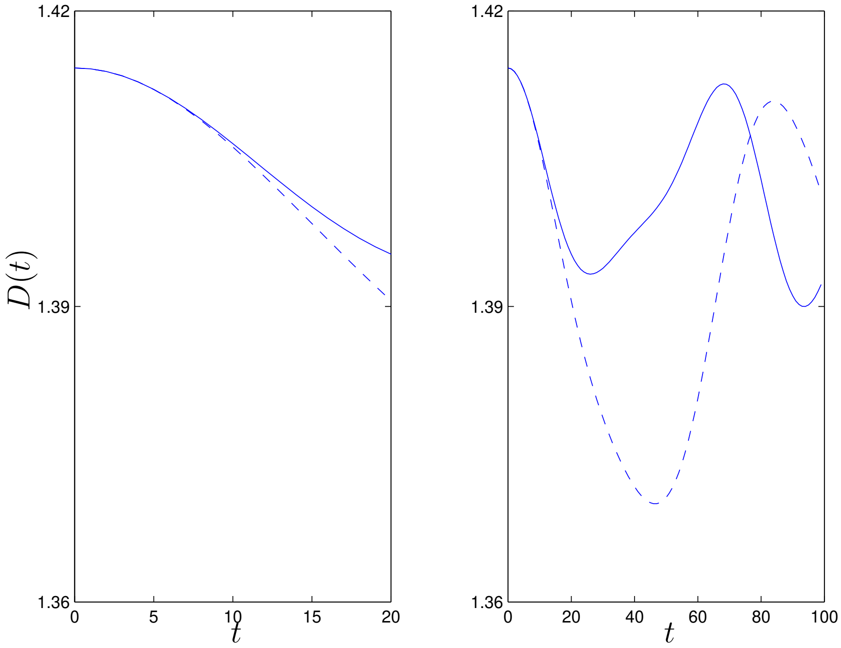


FIGURE 5. The diameter D of the lattice as function of time on $[0, 20]$ (left) and on $[0, 100]$ (right) for $m = 10^{-4}$ and $\tau = 1$. The solid line represents the diameter for the solution of the reduced system (3.3) and the dashed line represents the solution of the full system (1.1).

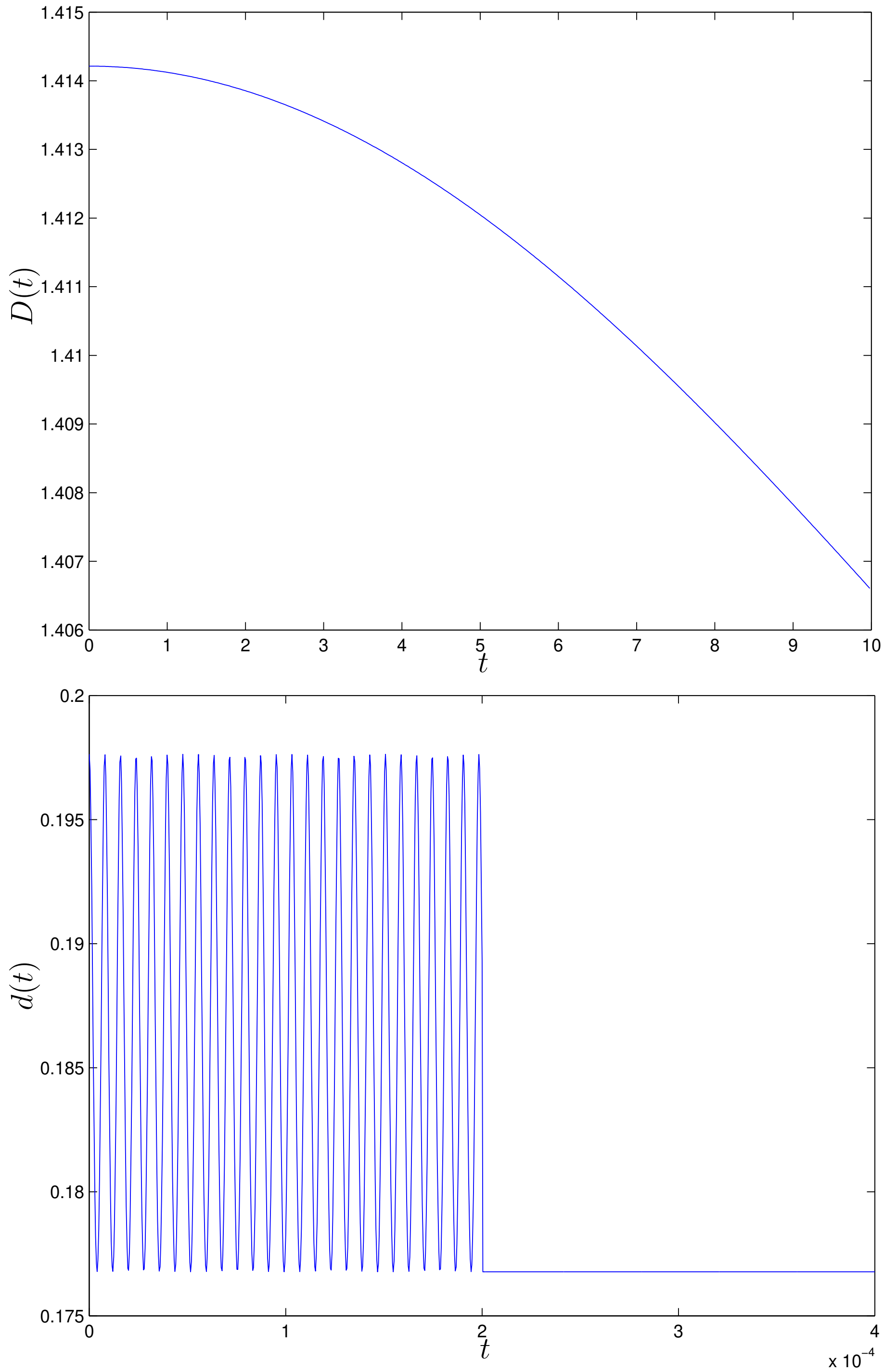


FIGURE 6. Distance D between the lower left large mass and the upper right large mass (above) and the distance d between the lower left large mass and the lower left small mass (below) as function of time on $[0, 10]$ and on $[0, 4 \cdot 10^{-4}]$, respectively.

Chalmers Finite Element Center Preprints

2003–01 *A hybrid method for elastic waves*
Larisa Beilina

2003–02 *Application of the local nonobtuse tetrahedral refinement techniques near Fichera-like corners*
L. Beilina, S. Korotov and M. Křížek

2003–03 *Nitsche's method for coupling non-matching meshes in fluid-structure vibration problems*
Peter Hansbo and Joakim Hermansson

2003–04 *Crouzeix–Raviart and Raviart–Thomas elements for acoustic fluid–structure interaction*
Joakim Hermansson

2003–05 *Smoothing properties and approximation of time derivatives in multistep backward difference methods for linear parabolic equations*
Yubin Yan

2003–06 *Postprocessing the finite element method for semilinear parabolic problems*
Yubin Yan

2003–07 *The finite element method for a linear stochastic parabolic partial differential equation driven by additive noise*
Yubin Yan

2003–08 *A finite element method for a nonlinear stochastic parabolic equation*
Yubin Yan

2003–09 *A finite element method for the simulation of strong and weak discontinuities in elasticity*
Anita Hansbo and Peter Hansbo

2003–10 *Generalized Green's functions and the effective domain of influence*
Donald Estep, Michael Holst, and Mats G. Larson

2003–11 *Adaptive finite element/difference method for inverse elastic scattering waves*
Larisa Beilina

2003–12 *A Lagrange multiplier method for the finite element solution of elliptic domain decomposition problems using non-matching meshes*
Peter Hansbo, Carlo Lovadina, Ilaria Perugia, and Giancarlo Sangalli

2003–13 *A reduced P^1–discontinuous Galerkin method*
R. Becker, E. Burman, P. Hansbo, and M. G. Larson

2003–14 *Nitsche's method combined with space–time finite elements for ALE fluid–structure interaction problems*
Peter Hansbo, Joakim Hermansson, and Thomas Svedberg

2003–15 *Stabilized Crouzeix–Raviart element for the Darcy-Stokes problem*
Erik Burman and Peter Hansbo

2003–16 *Edge stabilization for the generalized Stokes problem: a continuous interior penalty method*
Erik Burman and Peter Hansbo

2003–17 *A conservative flux for the continuous Galerkin method based on discontinuous enrichment*
Mats G. Larson and A. Jonas Niklasson

2003–18 *CAD–to–CAE integration through automated model simplification and adaptive modelling*
K.Y. Lee, M.A. Price, C.G. Armstrong, M.G. Larson, and K. Samuelsson

2003–19 *Multi-adaptive time integration*
Anders Logg

2003–20 *Adaptive computational methods for parabolic problems*
Kenneth Eriksson, Claes Johnson, and Anders Logg

2003–21 *The FEniCS project*
T. Dupont, J. Hoffman, C. Johnson, R. Kirby, M. Larson, A. Logg, and R. Scott

2003–22 *Adaptive finite element methods for LES: Computation of the mean drag coefficient in a turbulent flow around a surface mounted cube using adaptive mesh refinement*
Johan Hoffman

2003–23 *Adaptive DNS/LES: a new agenda in CFD*
Johan Hoffman and Claes Johnson

2003–24 *Multiscale convergence and reiterated homogenization of parabolic problem*
Anders Holmbom, Nils Svanstedt, and Niklas Wellander

2003–25 *On the relationship between some weak compactnesses with different numbers of scales*
Anders Holmbom, Jeanette Silfver, Nils Svanstedt, and Niklas Wellander

2003–26 *A posteriori error estimation in computational inverse scattering*
Larisa Beilina and Claes Johnson

2004–01 *Computability and adaptivity in CFD*
Johan Hoffman och Claes Johnson

2004–02 *Interpolation estimates for piecewise smooth functions in one dimension*
Anders Logg

2004–03 *Estimates of derivatives and jumps across element boundaries for multi-adaptive Galerkin solutions of ODEs*
Anders Logg

2004–04 *Multi-adaptive Galerkin methods for ODEs III: Existence and stability*
Anders Logg

2004–05 *Multi-adaptive Galerkin methods for ODEs IV: A priori error estimates*
Anders Logg

2004–06 *A stabilized non-conforming finite element method for incompressible flow*
Erik Burman and Peter Hansbo

2004–07 *On the uniqueness of weak solutions of Navier-Stokes equations: Remarks on a Clay Institute prize problem*
Johan Hoffman and Claes Johnson

2004–08 *A new approach to computational turbulence modeling*
Johan Hoffman and Claes Johnson

2004–09 *A posteriori error analysis of the boundary penalty method*
Kenneth Eriksson, Mats G. Larson, and Axel Målqvist

2004–10 *A posteriori error analysis of stabilized finite element approximations of the helmholtz equation on unstructured grids*
Mats G. Larson and Axel Målqvist

2004–11 *Adaptive variational multiscale methods based on a posteriori error estimation*
Mats G. Larson and Axel Målqvist

2004–12 *Multi-adaptive Galerkin methods for ODEs V: Stiff problems*
Johan Jansson and Anders Logg

2004–13 *Algorithms for multi-adaptive time-stepping*
Johan Jansson and Anders Logg

2004–14 *Simulation of mechanical systems with individual time steps*
Johan Jansson and Anders Logg

2004–15 *Computational modeling of dynamical systems*
Johan Jansson, Claes Johnson, and Anders Logg

These preprints can be obtained from

www.phi.chalmers.se/preprints