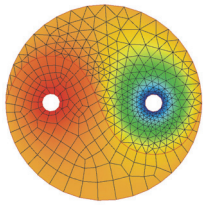
Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE

Informatyka w Technologii Materiałów

Vol. 6, 2006, No. 3-4



DIMENSIONAL REDUCTION FOR A LATTICE-LIKE MASS-SPRING POLYMER MODEL USING *hp*-ADAPTIVITY

CHETAN JHURANI, LESZEK DEMKOWICZ

Institute for Computational Engineering and Sciences
The University of Texas at Austin, Austin, Texas 78712
Corresponding Author: chetan@ices.utexas.edu (C. Jhurani)

Abstract

We present a method for dimensional reduction of the solution of the static equilibrium of a lattice-like network of point masses and harmonic springs. The dimensional reduction is achieved by energy-driven or goal-oriented *hp*-adaptivity. A discrete version of the fully-automatic *hp*-adaptive algorithm is used to constrain the positions of individual masses to conform to a piecewise polynomial. The lattice is divided into elements that consist of adjacent masses and springs. Element refinement is done by reducing the number of masses in an element (*h*-refinement) or increasing the polynomial degree (*p*-refinement). The necessity of a refinement and its optimal type depends on a local error estimate. The elements chosen for further refinement are those that give maximum reduction in error per number of invested degrees of freedom. The method generalizes the well-known Quasicontinuum method for molecular simulations (Tadmor, Ortiz, and Phillips 1996).

The presented numerical results confirm the exponential reduction in errors measured in the energy norm or in the quantity of interest. The method captures the variation in material constants by choosing large elements in regions of smooth variation of material data resulting in a significant dimensional reduction. The method fits into the "finite element" variational framework for boundary value problems. This enables utilization of an existing hp code for PDE problems with minimal changes.

Key words: hp-adaptivity, goal-oriented adaptivity, molecular statics, mass-spring lattice, SFIL

1. INTRODUCTION

Many problems in molecular statics can be modeled by representing molecules as point masses and the bonds between them as elastic springs. This approach is used, for example, in (Burns et. al. 2004) to solve for the equilibrium coordinates of individual monomers of a polymerized etch-barrier layer formed in the Step and Flash Imprint Lithography (SFIL) process. The molecular connectivities are obtained using a Monte Carlo simulation of the polymerization. The resulting cross-linked polymer network is prestrained and not in equilibrium. We wish to solve for the equilibrium configuration subject to some boundary constraints.

The number of degrees of freedom (DOFs) required to represent the equilibrium displacement values is proportional to the number of molecules. This results in a million DOFs for a typically small simulation of SFIL.

We present an approach for the dimensional reduction of this problem. The main idea of the algorithm is a generalization of the well-known Quasicontinuum method and it is related to the concept of mechanics with constraints developed by (Woźniak and Kleiber 1991). A molecular lattice representing the polymer is covered with an hp mesh fitting the structure of the lattice. Molecules within a single element are constrained then to follow the deformations prescribed by the corresponding element shape

 ISSN 1641-8581

functions. With the dimension of the Finite Element space smaller than the number of molecules by orders of magnitude, we achieve a dramatic reduction in the size of the linear (or nonlinear) system to solve.

Using this basic idea, we utilize the automatic mesh refinement strategy of hp-adaptivity developed in (Demkowicz, Rachowicz, and Devloo 2002). Starting from a coarse mesh, the hp-adaptivity algorithm finds the optimal refinements of the mesh to reduce the error in a suitable energy norm. In case of goal-oriented adaptivity, the refinements try to reduce an upper bound of the error estimate in the goal.

In this paper, we describe the transformation of the original equilibrium equations to a variational form suitable for the hp-algorithm. This is followed by a description of the goal-oriented hp-adaptivity. Finally, convergence results are presented for a 1-D and a linearized 2-D model with harmonic springs.

2. A 1-D MODEL IN VARIATIONAL FORM

The algorithm has been implemented first in 1-D. This section describes the 1-D mass-spring model and discusses the variational form of the problem.

Figure 1 shows a one-dimensional spring lattice of $N + 1$ masses with each mass connected to the nearest neighbor only. In an initial state the location of each of the masses, $\{X_i\}_{i=1}^{N+1}$, is specified. The equilibrium state is defined by displacements $\{u_i\}_{i=1}^{N+1}$ of the masses from the given initial state. Thus, $u_i := x_i - X_i$, $1 \le i \le N+1$. The spring constant k_i and the unstretched length l_i may vary with the spring.

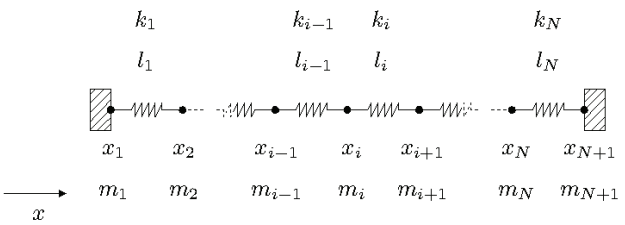


Figure 1. A one-dimensional spring network in static equilibrium. Both end-points are fixed by Dirichlet boundary conditions.

The initial state may be a "pre-stressed" state; the springs may have an initial non-zero stress. This is quantified by defining the pre-strain Δl_i in the i^{th} spring. $\Delta l_i := x_i - X_i - l_i$.

Define $\mathbf{u}$ to be a vector in $\mathbf{R}^{N+1}$ corresponding to $\{u_i\}_{i=1}^{N+1}$. We are interested in an approximate solution of the equilibrium state $\mathbf{u}$ or only in a specific quantity of interest (or goal) expressed in terms of a linear functional $\mathcal{G}(\mathbf{u})$ of the solution $\mathbf{u}$.

$$\mathcal{G}(\mathbf{u}) := \mathbf{g}^T\mathbf{u} = \sum_{i=1}^{N+1} g_i u_i \qquad \text{for a fixed } \mathbf{g} \in \mathbf{R}^{N+1}$$

When we have Dirichlet boundary conditions on both ends, the total potential energy functional $J : \mathbf{R}^{N+1} \to \mathbf{R}$ is defined as:

$$J(\mathbf{u}) := \sum_{i=1}^{N+1} \frac{1}{2} k_i \left(u_{i+1} - u_i - \Delta l_i\right)^2$$

J is zero when each spring has zero strain. The equilibrium state is the one that minimizes J while satisfying the boundary conditions. $\bar{x}_1$ and $\bar{x}_1^{N+1}$ are the final coordinates of the left and right end-points respectively.

Minimizing J gives the following variational form.

$$\begin{cases} \text{Find } \mathbf{u} \in \mathbf{R}^{N+1} \text{ such that} \\ u_1 = \bar{x} - X_1,\ u_{N+1} = \bar{x}_{N+1} - X_{N+1} \\ \sum_{i=1}^{N} k_i (u_{i+1} - u_i)(v_{i+1} - v_i) = \sum_{i=1}^{N} k_i (-\Delta l_i)(v_{i+1} - v_i) \quad \forall \mathbf{v} \in \mathrm{v} \end{cases}$$

where: $\mathcal{V} := \left\{\mathbf{v} \in \mathbf{R}^{N+1} : v_1 = v_{N+1} = 0\right\}$. This defines a symmetric and bilinear form $\mathcal{B} : \mathbf{R}^{N+1} \to \mathbf{R}$ a linear form $\mathcal{L} : \mathbf{R}^{N+1} \to \mathbf{R}$ and the semi-norm $\|\cdot\|_{\mathcal{U}}$.

$$\mathcal{B}(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{N} k_i \left(u_{i+1} - u_i\right)\left(v_{i+1} - v_i\right) \qquad (1)$$

$$\mathcal{L}(\mathbf{u}) = \sum_{i=1}^{N} k_i \left(-\Delta l_i\right)\left(v_{i+1} - v_i\right) \qquad (2)$$

$$\|\mathbf{u}\|_{\mathcal{U}} = \mathcal{B}\sqrt{\mathbf{u}, \mathbf{u}} \qquad (3)$$

3. *hp*-ADAPTIVITY APPROACH FOR THE 1-D MODEL

This section describes the *hp*-adaptivity approach for the 1-D model. The analysis carries over to higher dimensions without significant changes.

In the general case, when k_i and l_i are allowed to vary arbitrarily for each spring, the solution $\mathbf{u}$ requires $N+1$ DOFs. If they are "smooth" functions of the index i, then we may be able to provide an approximation of $\mathbf{u}$ using a fewer number of DOFs.

One such way is to use piecewise continuous polynomials specified by fewer than $N + 1$ constants and constrain the discrete values $\{u_i\}_{i=1}^{N+1}$ to conform to this curve. To give an extreme but simple example of why this approach may work, consider a network in which the first $N/2$ springs are identical and the rest are different but are identical to each other as well. A simple calculation will show that we require only three constants to specify all the components of $\{x_i\}_{i=1}^{N+1}$ and they lie on two piecewise linear curves with a jump in the slope at the central mass.

If the variations in k_i and l_i are completely random then this approach may fail to capture the behavior of solution with less d.o.f. because the solution will also be devoid of any smoothness. It may work, however, when there is an underlying smooth (possibly piecewise) variation in the material properties. With this motivation, we define a subspace $\mathcal{H}$ of $\mathbf{R}^{N+1}$ in which we look for an approximation of **u**.

Define "element" to be a non-empty set of adjacent springs. The whole network, for instance, can then be considered to be one element with N springs. Partition the whole network into M elements, $M \le N$. Let h_j be the number of springs in the j^{th} element and p_j be the polynomial order of approximation over the j^{th} element. Thus:

$$\sum_{j=1}^{M} h_j = N \qquad h_j \ge 1$$

As an example, consider the two adjacent elements in figure 2.

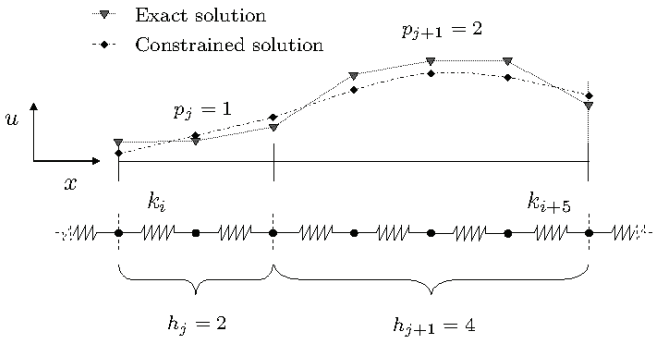


Figure 2. A piecewise polynomial is used to approximate the exact solution on two adjacent elements.

Let $0 \le p_j \le h_j$, $1 \le j \le M$, be M numbers and $\mathbf{p} := \{p_j\}_{j=1}^{M}$. Define a space of piecewise polynomials $\mathcal{P}^{\mathbf{p}} : [0, N] \to \mathbf{R}$

$$\mathcal{P}^{\mathbf{p}} = \left\{ w \in \mathcal{C}(0, N) : w\big|_j \in \mathcal{P}^{p_j} \right\}$$

where $\mathcal{C}$ denotes the space of continuous functions, $w\big|_j$ is the restriction of w to j^{th} element and $\mathcal{P}^{p_j}$ is the space of polynomials of degree p on the j^{th} element. Define $\mathcal{R} : \mathcal{P}^{\mathbf{p}} \to \mathbf{R}^{N+1}$:

$$(\mathcal{R}w)_i = w(i-1) \qquad \forall w \in \mathcal{P}^{\mathbf{p}},\ 1 \le i \le N+1$$

Let $\mathcal{H} := Range(\mathcal{R})$. $\mathcal{R}$ is an isomorphism between $\mathcal{P}$ and $\mathcal{H}$. The $N + 1$ components of an approximation of **u** in $\mathcal{H}$ identify a unique piecewise polynomial in $\mathcal{P}$ and any $w \in \mathcal{P}$ identifies a unique vector in $\mathcal{H}$. Moreover, this mapping is linear. $\mathcal{H}$ is the subspace in which we look for approximations of **u**. A simple observation shows that:

$$\dim(\mathcal{H}) = \dim(\mathcal{P}^{\mathbf{p}}) = \left(1 + \sum_{j=1}^{M} p_j\right) \le \left(1 + \sum_{j=1}^{M} h_j\right) =$$

$$N + 1 = \dim(\mathbf{R}^{N+1})$$

Hence we have a reduction in the dimension.

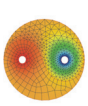
In figure 2 we use $p_j = 1$ and $p_{j+1} = 2$ to approximate **u** in the first and second element respectively. In general, the exact solution will require $p_j = 2$ and $p_{j+1} = 4$.

4. THE ELEMENT STIFFNESS AND LOAD MATRICES

Given the definition of the subspace $\mathcal{H}$ in previous section, we now define the variational formulation on this subspace using master element shape functions.

Define s_j, $1 \le j \le M$, to be the index of the left end-point of the j^{th} element. Thus $s_j = \sum_{m=1}^{j-1} h_j$. As special cases, we define s_1 to be 0 and s_{M+1} to be N, the left end-point of a hypothetical element beyond the last element on right. Figure 3 shows this graphically.

Using the definition of element boundary indices, we can define the affine mapping between the master element $\hat{K}$ and the physical element K. $\hat{K} = [0, 1] \ni x \to x_K(x) = i \in [s_j, s_{j+1}]$ where $x_K(x) = h_j x + s_j$ and $x_K^{-1}(i) = (i - s_j)/h_j$. Let $\hat{\varphi}_l$ denote the l^{th} shape function defined on $\hat{K}$ and let φ_l be the corresponding shape function on $\hat{K}$. $\varphi_l(i) = \hat{\varphi}_l\left[x_K^{-1}(i)\right]$ for i



$\in K$. Given a j, $1 \le j \le M$, and i an integer in (s_j, s_{j+1}), any $w \in \mathcal{H}$ can be written as:

$$w_{i+1} = \sum_{l=1}^{p_j+1} W_{j,l} \hat{\varphi}_l \left(\frac{i - s_j}{h_j} \right)$$

where $\left\{ \left\{ W_{j,l} \right\}_{l=1}^{p_j+1} \right\}_{j=1}^{M}$ are the degrees of freedom corresponding to w that constrain w to be in $\mathcal{H}$. It can be observed that we have more $W_{j,l}$'s than $dim(\mathcal{H})$. For every index i that falls on an inter-element boundary, w_i can be constructed from the left element or the right element. For example, to keep w_i single-valued for Peano shape functions we need $W_{j,2} \equiv W_{j+1,1}$. This reduces the number of independent $W_{j,l}$'s back to dim($\mathcal{H}$).

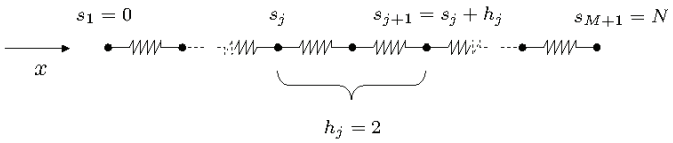


Figure 3. Definition of the element boundary indices.

These are the d.o.f.s ($W_{j,l}$) that we are interested in. We don't use such an expression for w directly while solving the system. This discussion was to describe how the solution is constrained and draw parallels to the FEM approach.

We can now write down the variational form for the particular case of Dirichlet boundary conditions on both the end-points.

$$\begin{cases} \text{Find } \mathbf{u} \in \mathcal{H} \text{ such that} \\ u_1 = \bar{x}_1 - X_1, u_{N+1} = \bar{x}_{N+1} - X_{N+1} \\ \mathcal{B}(\mathbf{u}, \mathbf{v}) = \mathcal{L}(\mathbf{v}) \quad \forall \mathbf{v} \in \mathcal{H}_{LR} \end{cases}$$

where $\mathcal{H}_{LR} := \{ v \in \mathcal{H} : v_1 = v_{N+1} = 0 \}$. $\mathcal{B}$ and $\mathcal{L}$ were defined in section 2, equations 1 and 2. For $1 \le l_1 \le p_{j+1}$ and $1 \le l_2 \le p_{j+1}$ we can write the (l_1, l_2) coefficient of the element stiffness matrix of j element as

$$\sum_{i=1+s_j}^{s_j+1} k_i \left[\hat{\varphi}_{l_1} \left(\frac{i - s_j}{h_j} \right) - \hat{\varphi}_{l_1} \left(\frac{i-1-s_j}{h_j} \right) \right] \left[\hat{\varphi}_{l_2} \left(\frac{i - s_j}{h_j} \right) - \hat{\varphi}_{l_2} \left(\frac{i-1-s_j}{h_j} \right) \right]$$

For $1 \le l \le p_j+1$, the l^{th} coefficient of the element load vector of j^{th} element is:

$$\sum_{i=1+s_j}^{s_j+1} k_i (-\Delta l_i) \left[\hat{\varphi}_l \left(\frac{i - s_j}{h_j} \right) - \hat{\varphi}_l \left(\frac{i-1-s_j}{h_j} \right) \right]$$

Using the boundary condition in the first and last element and assembling these matrices in the usual finite element way, we obtain a system of equations for $\left\{ \left\{ U_{j,l} \right\}_{l=1}^{p_j+1} \right\}_{j=1}^{M}$ the d.o.f.s for **u**.

5. A LINEARIZED 2-D MODEL

In higher space dimensions we have lattices with the basic unit being a cell. For a rectangular lattice, a cell consists of four point masses at the corners that are connected by four shared edge springs and 2 diagonal springs. A lattice is composed of 2-D arrays of cells.

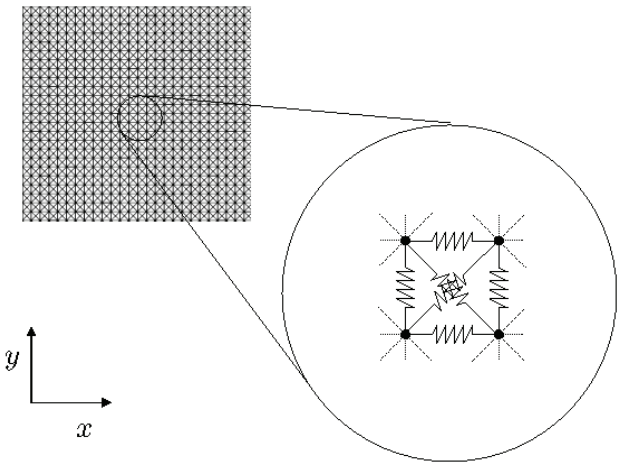


Figure 4. A 2-D rectangular lattice with a zoomed-in cell.

One big change when compared to 1-D is that the equilibrium equations are no longer linear. This is because of the square-root terms in the Euclidean distance between the particles. Assuming small rotations of bonds, we linearize the equilibrium equations and solve the resulting system. In this contribution, we focus on explaining the dimensional reduction methodology rather than the solution of a more general class of nonlinear problems. The solution of the linearized equations and the solution(s) of exact non-linear equations live in the same finite-dimensional space and thus there is no loss of generality here.

We now derive the variational form. Let $\mathcal{A}$ be the set of all point-masses of the lattice. We use the lowercase Greek alphabets $(\alpha, \beta, \eta, \zeta)$ to identify elements of $\mathcal{A}$. Let $k_{\alpha\beta}$ be the spring constant and $l_{\alpha\beta}$ be the unstretched length of the spring joining α and β. For particle α, $\mathbf{X}_\alpha$ is the initial position vector, x_α

is the equilibrium position vector, and $\mathbf{u}_\alpha$ is the displacement. See figure 5.

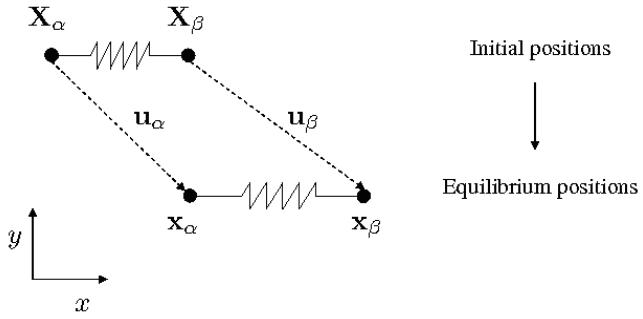


Figure 5. Various vector quantities for the masses in a 2-D lattice.

To simplify the expressions ahead, we also define the following symbols. $\mathbf{X}_{\alpha\beta} = \mathbf{X}_\alpha - \mathbf{X}_\beta$, $\mathbf{x}_{\alpha\beta} = \mathbf{x}_\alpha - \mathbf{x}_\beta$, $\mathbf{u}_{\alpha\beta} = \mathbf{u}_\alpha - \mathbf{u}_\beta$, and $\Delta l_{\alpha\beta} = \|X_{\alpha\beta}\| - l_{\alpha\beta}$. $\Delta l_{\alpha\beta}$ is the pre-strain in spring $k_{\alpha\beta}$. $\mathbf{F}_{\alpha\beta}^{EX}$ is the exact force on due to the spring $k_{\alpha\beta}$.

$$F_{\alpha\beta}^{EX} = -k_{\alpha\beta}\left(\left\|\mathbf{x}_{\alpha\beta}\right\| - l_{\alpha\beta}\right)\frac{\mathbf{x}_{\alpha\beta}}{\left\|\mathbf{x}_{\alpha\beta}\right\|}$$

We assume small deformations so that $\frac{\mathbf{X}_{\alpha\beta}}{\left\|\mathbf{X}_{\alpha\beta}\right\|}\cdot\frac{\mathbf{x}_{\alpha\beta}}{\left\|\mathbf{x}_{\alpha\beta}\right\|} \approx 1$. Thus, $\left\|\mathbf{x}_{\alpha\beta}\right\| \approx \frac{\mathbf{X}_{\alpha\beta}\cdot\mathbf{x}_{\alpha\beta}}{\left\|\mathbf{X}_{\alpha\beta}\right\|}$. The assumption also implies that the forces act along the initial spring directions $\frac{\mathbf{X}_{\alpha\beta}}{\left\|\mathbf{X}_{\alpha\beta}\right\|}$ instead of the final (equilibrium) direction $\frac{\mathbf{x}_{\alpha\beta}}{\left\|\mathbf{x}_{\alpha\beta}\right\|}$. Using these approximations, we can linearize the exact force. The linearized force $\mathbf{F}_{\alpha\beta}$ is:

$$F_{\alpha\beta} = -k_{\alpha\beta}\left(\frac{\mathbf{X}_{\alpha\beta}\cdot\mathbf{x}_{\alpha\beta}}{\left\|\mathbf{X}_{\alpha\beta}\right\|} - l_{\alpha\beta}\right)\frac{\mathbf{X}_{\alpha\beta}}{\left\|\mathbf{X}_{\alpha\beta}\right\|}$$

If $F_{\alpha\beta}^{EXT}$ is a specified external force on α (a Neumann boundary condition), the linearized variational problem is:

$$\begin{cases} \text{Find } \left(\{\mathbf{u}_\eta\}_{\eta\in\mathcal{A}}\right) \text{such that it satisfies the Dirichlet boundary conditions and} \\ \mathcal{B}\left(\{\mathbf{u}_\eta\}_{\eta\in\mathcal{A}}, \{\mathbf{u}_\zeta\}_{\zeta\in\mathcal{A}}\right) := \sum_{\substack{\alpha,\beta\in\mathcal{A}\\ \alpha\rightsquigarrow\beta}} \frac{1}{2}\frac{k_{\alpha\beta}}{\left\|\mathbf{X}_{\alpha\beta}\right\|^2}\left(\mathbf{X}_{\alpha\beta}\cdot\mathbf{v}_{\alpha\beta}\right) \\ = \\ \mathcal{L}\left(\{\mathrm{v}_\zeta\}_{\zeta\in\mathcal{A}}\right) := \sum_{\alpha\in\mathcal{A}} \mathbf{F}_\alpha^{EXT}\cdot\mathbf{v}_\alpha \sum_{\substack{\alpha,\beta\in\mathcal{A}\\ \alpha\rightsquigarrow\beta}} \frac{1}{2}\frac{k_{\alpha\beta}\Delta l_{\alpha\beta}}{\left\|\mathbf{X}_{\alpha\beta}\right\|^2}\left(\mathbf{X}_{\alpha\beta}\cdot\mathbf{v}_{\alpha\beta}\right) \\ \forall \text{ admissible } \{\mathrm{v}_\zeta\}_{\zeta\in\mathcal{A}} \end{cases}$$

$\alpha \rightsquigarrow \beta$ means particle α is connected to particle β (by the spring $(k_{\alpha\beta}, l_{\alpha\beta})$).

After deriving the "exact" variational form we can use tensor product polynomial spaces to approximate the solution on individual elements of the lattice. This is shown in figure 6. The computation of element stiffness and load matrices done in a way fully analogous to the 1-D computations.

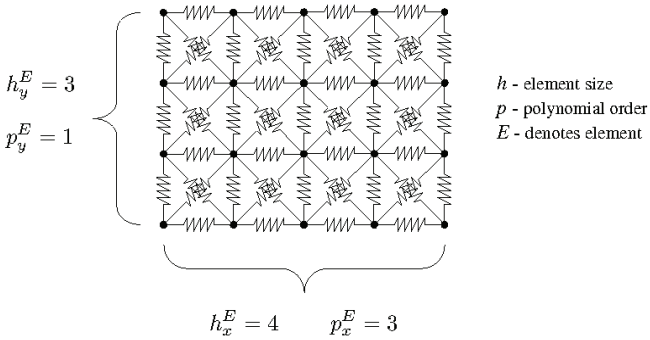


Figure 6. A sample element E of 12 cells with actual DOFs 20 per component that reduce to 8 when approximated by a polynomial.

6. GOAL-ORIENTED HP-ADAPTIVITY

This section describes goal-oriented hp-adaptivity in terms of an abstract variational formulation. Instead of refining the mesh to minimize the error in energy norm, we want the refinements to minimize the error in a specified quantity of interest (the goal). We shall use the notation of the 1-D lattice, with the algorithm being identical for the 2-D case.

The goal here is a linear and continuous functional of $\mathbf{u}$ − the displacement vectors. In the most general case, for a one-dimensional system, the goal $\mathcal{G}(\mathbf{u})$ is characterized by a fixed vector $\mathbf{g} \in \mathbf{R}^{N+1}$.

$$\mathcal{G}(\mathbf{u}) = \mathbf{g}^T\mathbf{u}$$

The superscript *EX* denotes the exact solution and the superscript *hp* denotes the approximate solution $\mathbf{e} := \mathbf{u}^{EX} - \mathbf{u}^{hp} \in \mathbf{R}_{LR}^{N+1}$ is the error. A Dirichlet lift is $\mathbf{u}^D \in \mathbf{R}^{N+1}$ such that $\bar{u}_1^D = \bar{x}_1 - X_1$ and $\bar{u}_{N+1}^D = \bar{x}_{N+1} - X_{N+1}$. $\mathcal{H}_{LR} := \{\mathbf{v} \in \mathrm{H} : v_1 = v_{N+1} = 0\}$ is the space of test functions for the approximation.

The variational formulation for the primal problem is:

$$\begin{cases} \text{Find } \mathbf{u}^{hp} \in \mathbf{u}^D + \mathcal{H}_{LR}^{N+1} \text{ such that} \\ \mathcal{B}(\mathbf{u}^{hp}, \mathbf{v}^{hp}) = \mathcal{L}(\mathbf{v}^{hp}) \quad \forall\, \mathbf{v}^{hp} \in \mathcal{H}_{LR} \end{cases}$$

The Galerkin approximation is

$$\begin{cases}\text{Find } \mathbf{u}^{hp} \in \mathbf{u}^D + \mathcal{H}_{LR} \text{ such that} \\ \mathcal{B}(\mathbf{u}^{hp}, \mathbf{v}^{hp}) = \mathcal{L}(\mathbf{v}^{hp}) \quad \forall \mathbf{v}^{hp} \in \mathcal{H}_{LR}\end{cases}$$

Define $\mathbf{r} := \mathbf{R}_{LR}^{N+1} \rightarrow \mathbf{R}$ as the residual.

$$\mathbf{r}(\mathbf{v}) = \mathcal{B}(\mathbf{e}, \mathbf{v}) \qquad \forall\, \mathbf{v} \in \mathbf{R}_{LR}^{N+1}$$

Galerkin orthogonality implies

$$\mathcal{B}\left(\mathbf{e}, \mathbf{v}^{hp}\right) = 0 \quad \forall \quad \mathbf{v}^{hp} \in \mathcal{H}_{LR}.$$

Thus $\mathbf{r}\left(\mathbf{v}^{hp}\right) = 0 \quad \forall \quad \mathbf{v}^{hp} \in \mathcal{H}_{LR}$.

Define w : $(\mathbf{R}_{LR}^{N+1})' \rightarrow \mathbf{R}$.

$$\mathbf{w}(\mathbf{r}) = \mathcal{G}(\mathbf{e})$$

$\mathbf{w}$ is called the *influence function*. By reflexivity of $\mathbf{R}_{LR}^{N+1}$ we can write this as $\mathbf{r}(\mathbf{w}) = \mathcal{G}(\mathbf{e})$

Thus $\mathcal{B}(\mathbf{e},\mathbf{w}) = \mathcal{G}(\mathbf{e}) \;\; \forall\, \mathbf{e} \in \mathbf{R}_{LR}^{N+1}$. This gives the variational formulation for the exact error in the goal (the dual problem). We have assumed that $\mathcal{B}$ is symmetric.

$$\begin{cases}\text{Find } \mathbf{w}^{EX} \in \mathbf{R}_{LR}^{N+1} \text{ such that} \\ \mathcal{B}(\mathbf{w}^{EX}, \mathbf{e}) = \mathcal{G}(\mathbf{e}) \quad \forall \mathbf{e} \in \mathbf{R}_{LR}^{N+1}\end{cases}$$

This suggests the Galerkin approximation

$$\begin{cases}\text{Find } \mathbf{w}^{hp} \in \mathcal{H}_{LR} \text{ such that} \\ \mathcal{B}\left(\mathbf{w}^{hp}, \mathbf{e}^{hp}\right) = \mathcal{G}\left(\mathbf{e}^{hp}\right) \quad \forall\, \mathbf{e}^{hp} \in \mathcal{H}_{LR}\end{cases}$$

We show how the error in the approximation of goal depends upon the approximation errors in the primal and dual problems.

$\mathcal{G}\left(\mathbf{u}^{EX}\right) - \mathcal{G}\left(\mathbf{u}^{hp}\right)$ error in the goal

$= \mathcal{G}\left(\mathbf{u}^{EX} - \mathbf{u}^{hp}\right)$ linearity of $\mathcal{G}$

$= \mathcal{B}\left(\mathbf{u}^{EX} - \mathbf{u}^{hp}, \mathbf{w}^{EX}\right)$ dual problem

$= \mathcal{B}\left(\mathbf{u}^{EX} - \mathbf{u}^{hp}, \mathbf{w}^{EX}\right) - \mathcal{B}\left(\mathbf{u}^{EX} - \mathbf{u}^{hp}, \mathbf{w}^{hp}\right)$ Galerkin orthogonality

$= \mathcal{B}\left(\mathbf{u}^{EX} - \mathbf{u}^{hp}, \mathbf{w}^{EX} - \mathbf{w}^{hp}\right)$ bilinearity of $\mathcal{B}$

This provides just the global error and not errors in each element individually. We need estimates of error in the primal and dual problem on each element for the refinement strategy to work. The details for this are shown next.

We work with two meshes – a coarse one and an *hp*-refined one – with the coarse one being a subspace of the refined one. The superscript *c* refers to the functions on the coarse mesh and *f* refers to those on the fine mesh. Π^c is the projection based interpolation operator from $\mathbf{R}^{N+1}$ to $\mathcal{H}^c$ described in full detail in (Demkowicz 2006). Repeating the steps above we get a representation for the relative error in goal.

$\mathcal{G}\left(\mathbf{u}^f\right) - \mathcal{G}\left(\mathbf{u}^c\right)$ The error estimate

$= \mathcal{G}\left(\mathbf{u}^f - \mathbf{u}^c\right)$ linearity $\mathcal{G}$

$= \mathcal{B}\left(\mathbf{u}^f - \mathbf{u}^c, \mathbf{w}^f\right)$ dual problem

$= \mathcal{B}\left(\mathbf{u}^f - \mathbf{u}^c, \mathbf{w}^f\right) - \mathcal{B}\left(\mathbf{u}^f - \mathbf{u}^c, \Pi^c \mathbf{w}^f\right)$ Galerkin orthogonality

$= \mathcal{B}\left(\mathbf{u}^f - \mathbf{u}^c, \mathbf{w}^f - \Pi^c \mathbf{w}^f\right)$ bilinearity of $\mathcal{B}$

We now break up this expression into estimates over elements by neglecting a higher order term, using the triangle inequality, and Cauchy-Schwarz inequality:

$\left|\mathcal{B}\left(\mathbf{u}^f - \mathbf{u}^c, \mathbf{w}^f - \Pi^c \mathbf{w}^f\right)\right|$ the error estimate

$\approx \left|\mathcal{B}\left(\mathbf{u}^f - \Pi^c \mathbf{u}^f, \mathbf{w}^f - \Pi^c \mathbf{w}^f\right) - \mathcal{B}\left(\Pi^c \mathbf{u}^f - \mathbf{u}^c, \mathbf{w}^f - \Pi^c \mathbf{w}^f\right)\right|$ bilinearity of $\mathcal{B}$

$\approx \left|\mathcal{B}\left(\mathbf{u}^f - \Pi^c \mathbf{u}^f, \mathbf{w}^f - \Pi^c \mathbf{w}^f\right) - \cancelto{0}{\mathcal{B}\left(\Pi^c \mathbf{u}^f - \mathbf{u}^c, \mathbf{w}^f - \Pi^c \mathbf{w}^f\right)}\right|$ ignore higher order term

$= \sum_{j=1}^{M} \left|\mathcal{B}\left(\mathbf{u}^f - \Pi^c \mathbf{u}^f, \mathbf{w}^f - \Pi^c \mathbf{w}^f\right)\right|$ Sum over elements

$\leq \sum_{j=1}^{M} \left\|\mathbf{u}^f - \Pi^c \mathbf{u}^f\right\|_{\mathcal{U}_j} \left\|\mathbf{w}^f - \Pi^c \mathbf{w}^f\right\|_{\mathcal{U}_j}$

$\|\cdot\|_{\mathcal{U}_j}$ is the energy norm over j^{th} element $\mathcal{B}_j$ gives the contribution to the bilinear form of j^{th} element. This estimate is computable element-wise for local refinements and drives the goal-oriented mesh adaptivity. Contrast this with the estimate used in energy-driven *hp*-adaptivity.

$$\left\|\mathbf{u}^f - \Pi^c \mathbf{u}^f\right\|_{\mathcal{U}} = \sqrt{\sum_{j=1}^{M} \left\|\mathbf{u}^f - \Pi^c \mathbf{u}^f\right\|_{\mathcal{U}_j}}$$

7. RESULTS FOR THE 1-D MODEL PROBLEM

We present some representative results of using *hp*-adaptivity to solve for the equilibrium displacements in one-dimensional linear spring networks.

The "lattice" consists of 1024 springs with identical equilibrium lengths but varying stiffnesses.

Both ends are fixed so that the lattice expands to reach equilibrium. The stiffness varies smoothly in each of the 4 patches (see figure 7). It also shows the final distribution of polynomial orders and the sizes of all the mesh elements. We see that the adaptivity algorithm refines elements around regions of fast variation or large jumps in material data and uses larger elements when the variation is smooth. Figure 8 shows that with just 20 DOFs the solution is visually identical to the exact 1025 DOF solution.

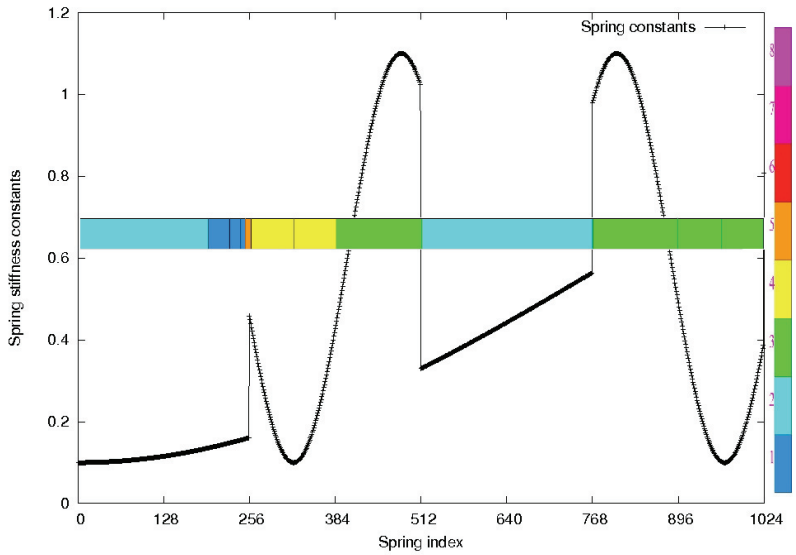


Figure 7. A piecewise smooth spring stiffness data shows that larger elements are selected in regions of slow variation. The color-coded polynomial scale is overlaid

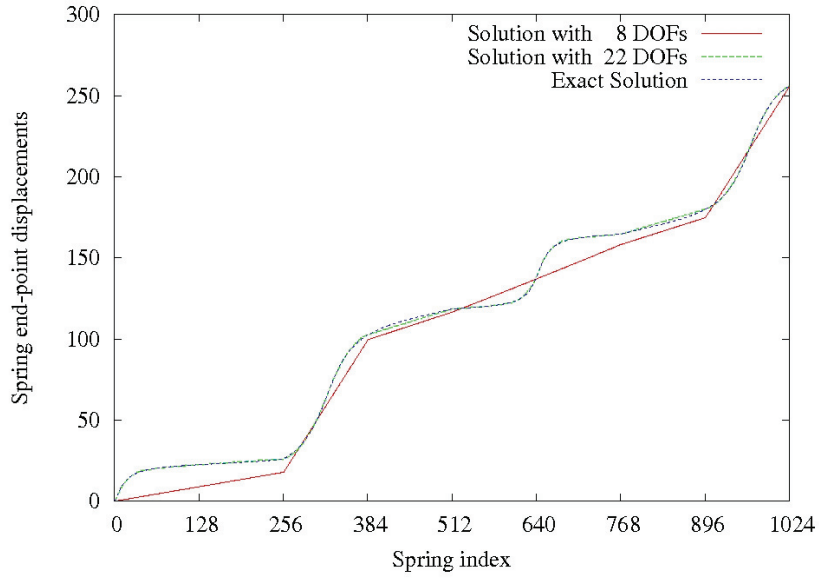


Figure 8. Convergence of solution for a 1025 DOF spring system with data shown in figure 7.

In figure 9 we see less than 2% error by using less than one-tenth of the actual DOFs of the solution. The error estimate is computed simply by evaluating the energy norm of the difference between the coarse and the fine grid solutions.

In the case of goal-oriented adaptivity, the goal is defined as the relative displacement of two masses at one-third and two-thirds of the total distance from the left end-point. For our sample case with N = 1024, this is given by (up to a sign factor).

$$\mathcal{G}(\mathbf{u}) = u_{341} - u_{682}$$

Figure 10 shows the minimized upper bound of the error in the goal as well as the actual error. The exact error does not decrease monotonically but it remains consistently below the estimate.

8. RESULTS FOR THE 2-D MODEL PROBLEM

For a 2-D representative problem, we chose a square lattice consisting of 64 cells in each direction. See figure 11(a). The bottom was fixed and a unit force in x direction was applied at the top right corner.

All the spring constants were chosen to be 1. The equilibrium length of the edge springs was 1 but that of the diagonal springs was 1.25 instead of $\sqrt{2}$. This led to a prestrain in all the cells. The equilibrium solution for this configuration is shown in figure 11(b).

We start the hp-adaptivity from the coarsest mesh with one bilinear element. Figure 12(a) shows the final refined mesh. The mesh refinement strategy automatically picks up higher gradients towards the top-right corner and fully resolves them. Figure 14(a) shows the convergence of error in the energy norm. We reach 1% error with almost two orders of magnitude fewer DOFs compared to the original discrete system.

For the goal-oriented algorithm, the goal is defined to be the x displacement of the top-left corner. In this case, the algorithm tries to refine near the top-left corner as well. Just like the energy-driven algorithm, we reach 1% error with a hundred times fewer DOFs.

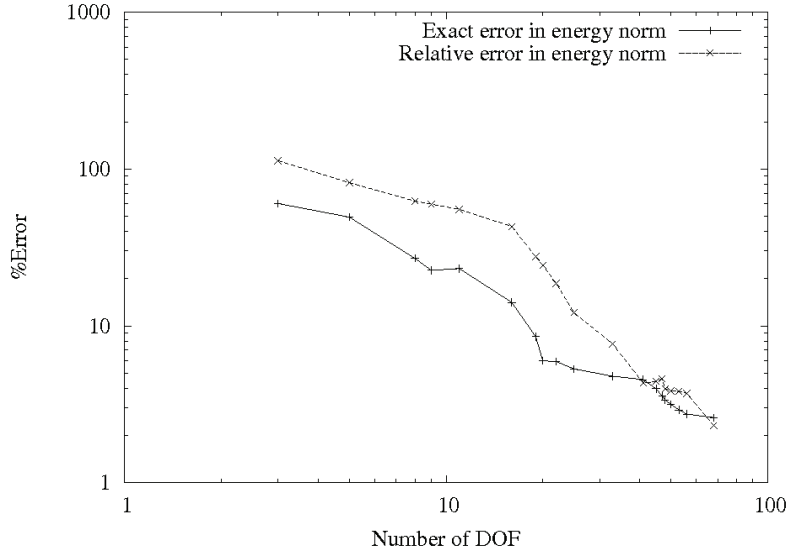


Figure 9. Convergence history for exact and estimated errors in energy norm (equation 5) for energy driven hp-adaptivity for data shown in figure 7.

Figure 10. Convergence history for the exact error and upper bound of the error in goal (equation (4)) for goal-driven hp-adaptivity for data shown in figure 7. The goal is given in equation(6).

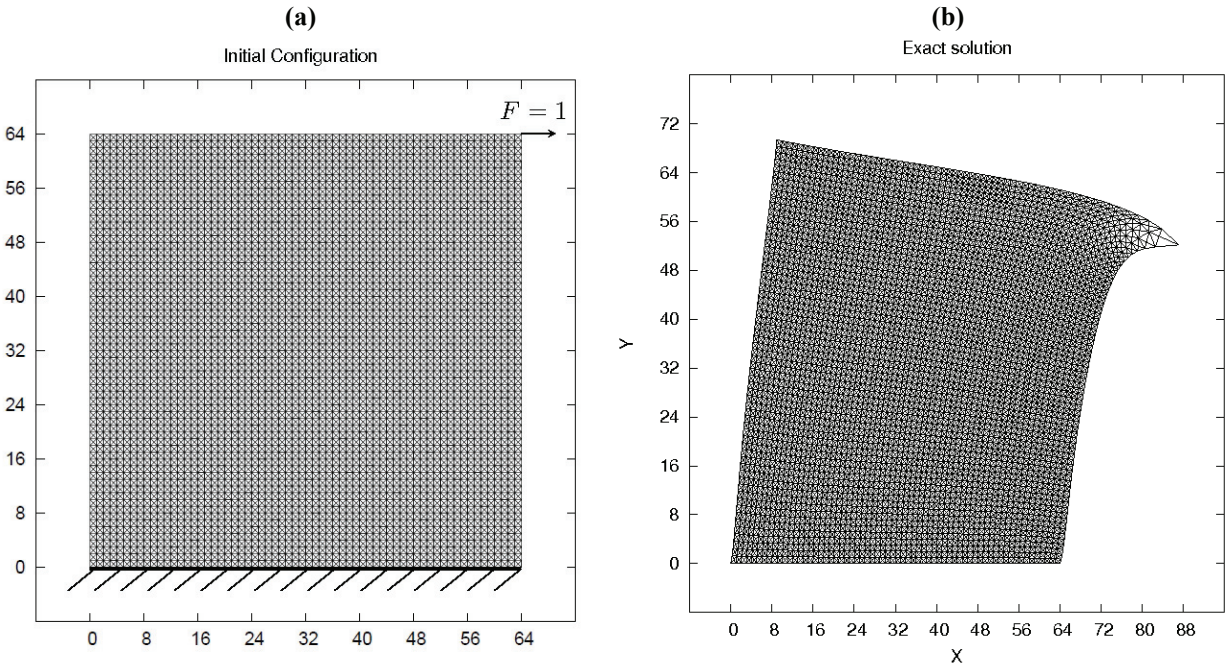


Figure 11. The initial configuration of the lattice and the exact solution of the equilibrium are shown.

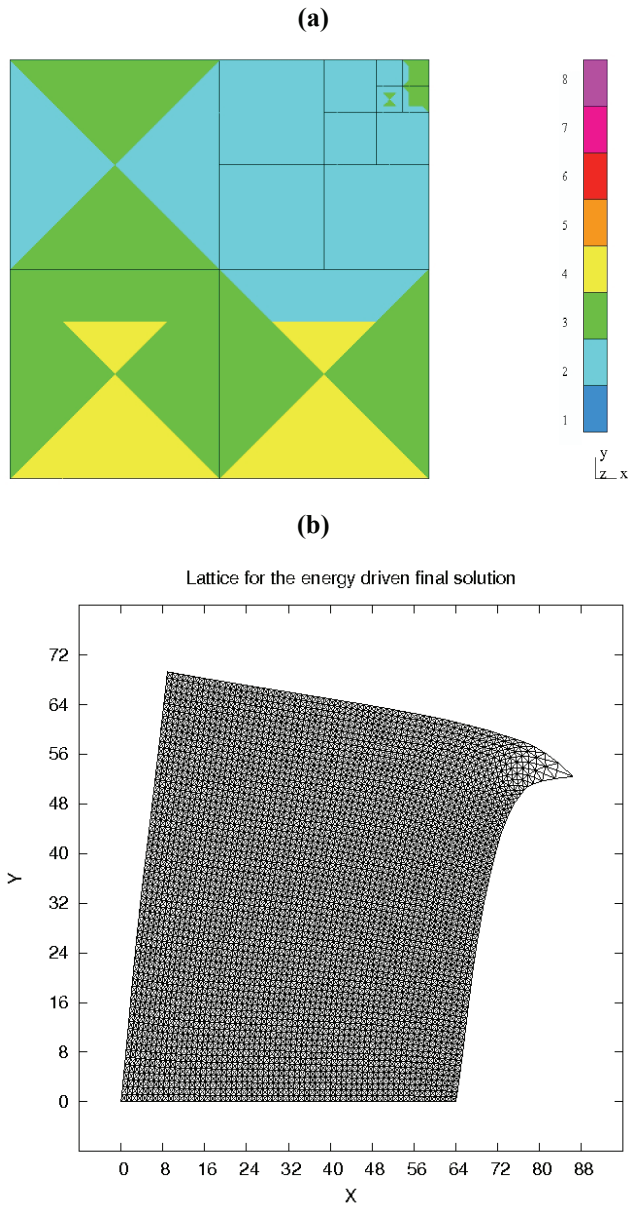


Figure 12. Final mesh and solution for the energy-driven hp-adaptive solution of the 2-D lattice.

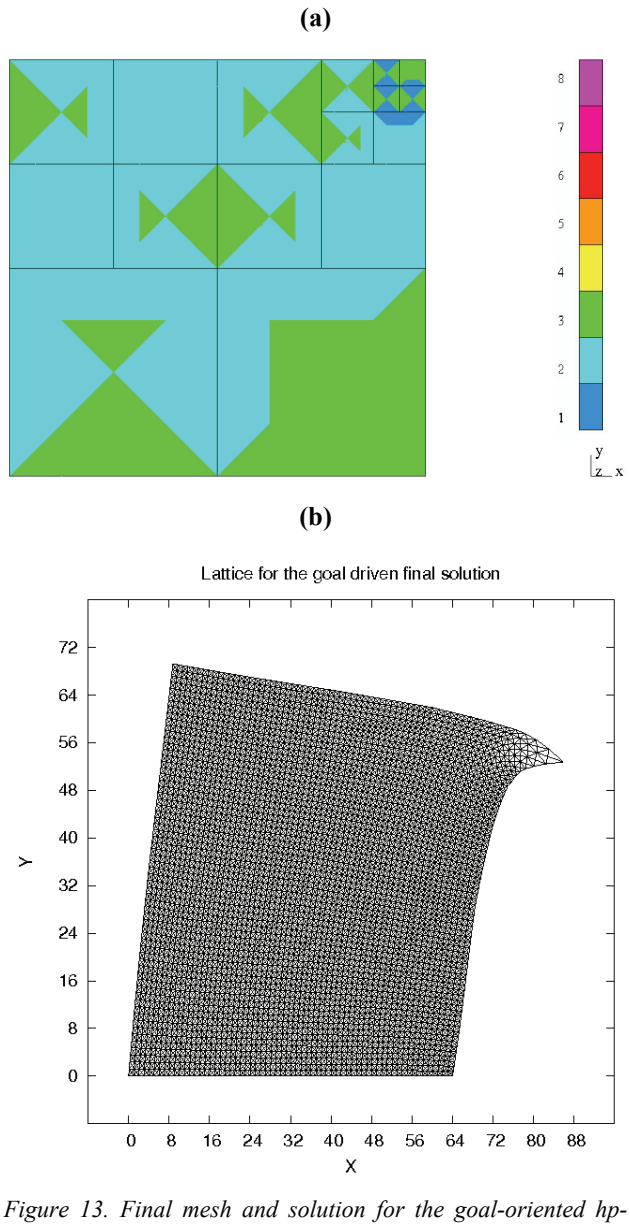
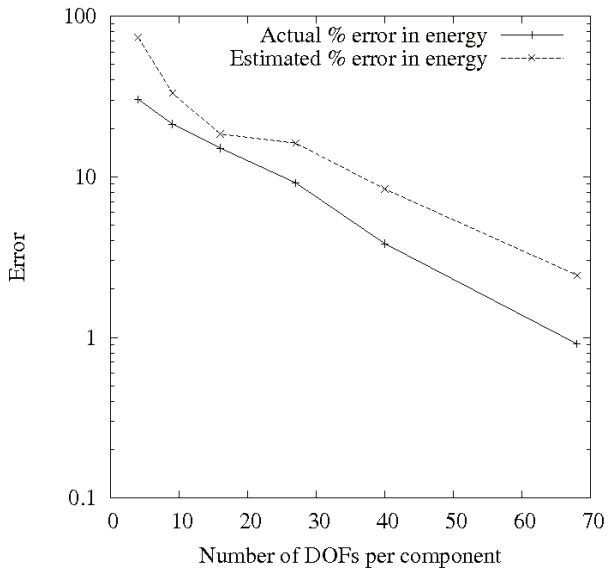


Figure 13. Final mesh and solution for the goal-oriented hp-adaptive solution of the 2-D lattice. The goal is the x displacement of the top-left corner.



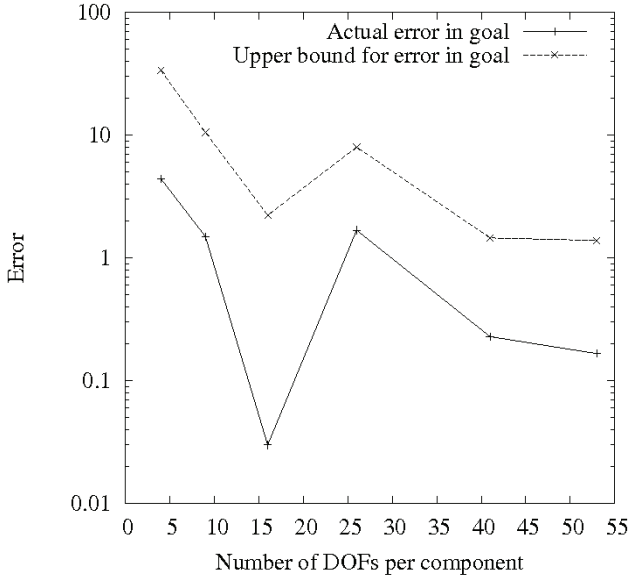


Figure 14. Convergence history for energy-driven and goal-oriented hp-adaptivity.

ISSN 1641-8581

In both the energy-driven and the goal-oriented examples, the mesh refinements were terminated at a moment when the refinements reached the resolution of the original model locally. In both cases this happened in the element at the top-right corner.

9. CONCLUSIONS AND FUTURE DIRECTIONS

We have shown how the general *hp*-adaptivity can be used for a dimensional reduction of a 1-D and 2-D spring lattices. In all presented test cases very high accuracy is achieved with very few degrees of freedom. In the presented cases the number of degrees of freedom required for 1% error were almost 100 times fewer than for the exact solution. Moreover, the algorithm has been implemented with limited changes in a general-purpose code that supports *hp*-adaptivity (Demkowicz 2006).

The presented implementations in 1-D and 2-D have been used to assess the feasibility of using constraints in networks that have a piecewise smooth variation of spring coefficients. The algorithm is fully general enough and can be used for 3-D spring networks.

Acknowledgments. This material is based upon work supported by the Department of Energy, USA under Grant No. DE-FG02-05ER25701. The authors would also like to thank Dr J. Tinsley Oden, Dr Jon Bass, Dr Serge Prudhomme, and Paul T. Bauman for many helpful discussions.

REFERENCES

Burns, R. L., Johnson, S. C., Schmid, G. M., Kim, E. K., Dickey, M. D., Meiring, J., Burns, S. D., Stacey, N. A., Willson, C. G., Convey, D., Wei, Y., Fejes, P., Gehoski, K. A., Mancini, D. P., Nordquist, K. J., Dauksher, W. J., Resnick, D. J., 2004, Mesoscale modeling for SFIL simulating polymerization kinetics and densification, in: *Emerging Lithographic Technologies VIII*, ed., Mackay R.S., Proc. SPIE, Vol. 5374, 348-360.

Demkowicz, L., 2006, *Computing with hp-adaptive finite elements*, Boca Raton, Chapman & Hall/CRC Applied Mathematics & Nonlinear Science.

Demkowicz, L., Rachowicz, W., Devloo, P., 2002, A fully automatic *hp*-adaptivity, *J. of Scientific Computing*, 17, 127–155.

Tadmor, E. B., Ortiz, M., Phillips, R., 1996, Quasicontinuum analysis of defects in solids, *Philosophical Magazine A*, 73, 1529–1563.

Woźniak, C., Kleiber, M., 1991, *Nonlinear Mechanics of Structures*, Kluwer Academic Publishers.

ZASTOSOWANIE *hp* ADAPTACJI DO REDUKCJI ROZMIARU PROBLEMU MODELOWANIA POLIMERU ZA POMOCĄ SIATKI CZĄSTEK POŁĄCZONYCH SPRĘŻYNAMI

Streszczenie

Artykuł przedstawia metodę redukcji wymiaru rozwiązania problemu statki cząsteczkowej dla sieci punktów materialnych połączonych sprężynami harmonicznymi. Redukcja rozmiaru uzyskana jest poprzez zastosowanie *hp* adaptacji kierowanej z pomocą normy energetycznej lub funkcjonału celu.

Dyskretna wersja algorytmu automatycznej *hp* adaptacji użyta została w celu aproksymacji położeń poszczególnych cząstek poprzez wielomian kawałkami ciągły. Siatka obliczeniowa podzielona jest na elementy składające się z sąsiadujących cząstek i sprężyn. Elementy adaptowane są poprzez redukcje ilości cząstek w elemencie (*h* adaptacja) lub poprzez zwiększanie wielomianowego stopnia aproksymacji (*p* adaptacja). Decyzje o konieczności adaptacji oraz ich rodzaju zależą od lokalnych oszacowań błędów. Do dalszych adaptacji wybierane są te elementy, dla których uzyskamy maksymalna redukcje błędu względem ilości dodanych stopni swobody. Metoda ta uogólnia dobrze znana metodę quasi-ciągłych symulacji molekularnych (Tadmor i in., 1996).

Przedstawiona metoda numeryczna osiąga eksponencjalną zbieżność błędu numerycznego mierzonego w normie energetycznej lub w normie zdefiniowanej przez funkcjonał celu. Metoda ta automatycznie konstruuje duże elementy w obszarach o małej zmienności stałych materiałowych, i dzięki temu prowadzi do dużej redukcji rozmiaru problemu obliczeniowego.

Metodę zaliczyć można do klasy metod elementów skończonych rozwiązujących sformułowanie wariacyjne dla problemu brzegowego. Dzięki temu możliwe było zastosowanie istniejących kodów *hp* adaptacji dla równań różniczkowych cząstkowych, z minimalną wymaganą ilością zmian w kodzie.

Submitted: December 12, 2006
Submitted in a revised form: January 4, 2007
Accepted: January 9, 2007