ECE 695
Numerical Simulations
Lecture 12: Thermomechanical FEM

Prof. Peter Bermel

February 6, 2017

Outline

- Static Equilibrium
- Dynamic Equilibrium
- Thermal transport mechanisms
- Thermal transport modeling in MATLAB

Static Equilibrium

- Newton's Law for a 1D wire :

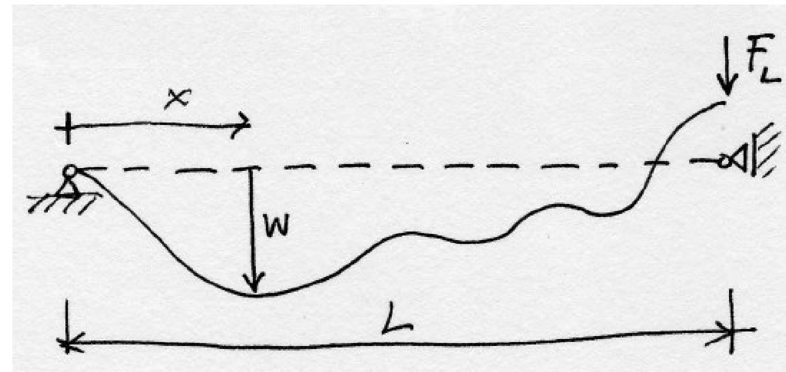
$$P\frac{\partial^2 w}{\partial x^2} + q = \mu\ddot{w}$$

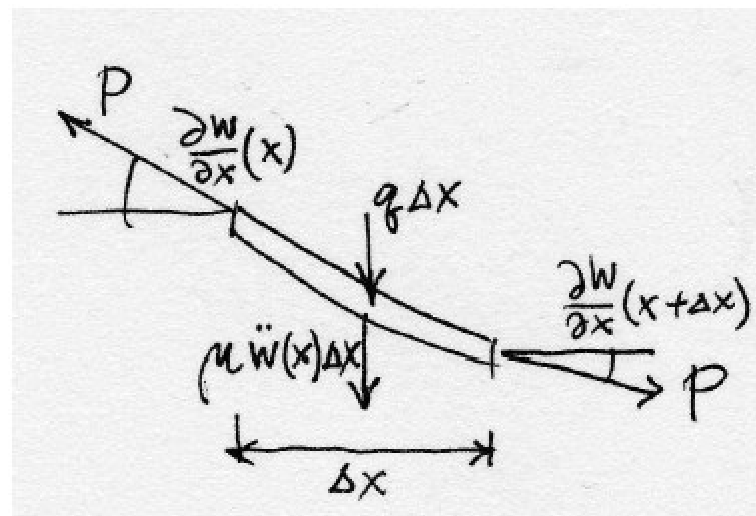
- In static equilibrium, forces balance exactly:

$$P\frac{\partial^2 w}{\partial x^2} + q = 0$$

- Define residual error in solution such that:

$$r_B = P\frac{\partial^2 w}{\partial x^2} + q$$





Galerkin Method

- In general, we want to apply Galerkin method with trial functions η_j:

$$\int_0^L dx\, \eta_j(x) r_B(x,t) = 0$$

- Analogy: stuff balloon into box, with each trial function a single ‘finger’.
- Substituting: $\int_0^L dx\, \eta_j(x) \left[P \frac{\partial^2 w}{\partial x^2} + q \right] = 0$

Galerkin Method

- Integrating by parts:

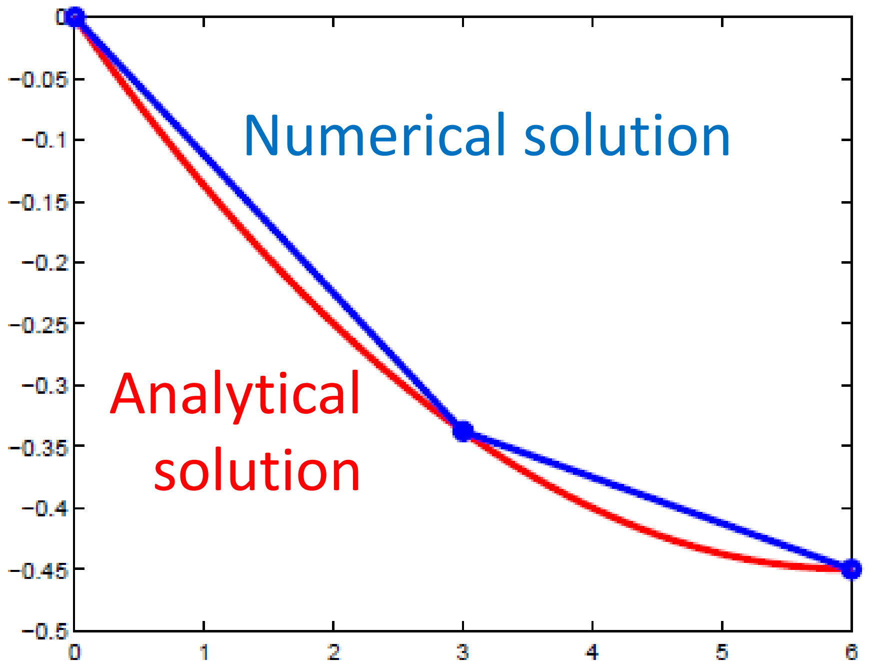
$$0 = \left[\eta_j P \frac{\partial w}{\partial x}\right]_0^L + \int_0^L dx \, \left[\eta_j q - \frac{\partial \eta_j}{\partial x} P \frac{\partial w}{\partial x}\right]$$

- Letting boundary terms vanish and substituting linear basis ($w = \sum_i N_i w_i$) yields:

$$0 = \int_0^L dx \, \eta_j q - \sum_{i=1}^{N} \left(\int_0^L dx \frac{\partial \eta_j}{\partial x} P \frac{\partial N_i}{\partial x} \right) w_i$$

$$0 = \boldsymbol{b} - \boldsymbol{K}\boldsymbol{w}$$

Static Equilibrium



Numerical solution matches analytical solution closely at key

Dynamic Equilibrium

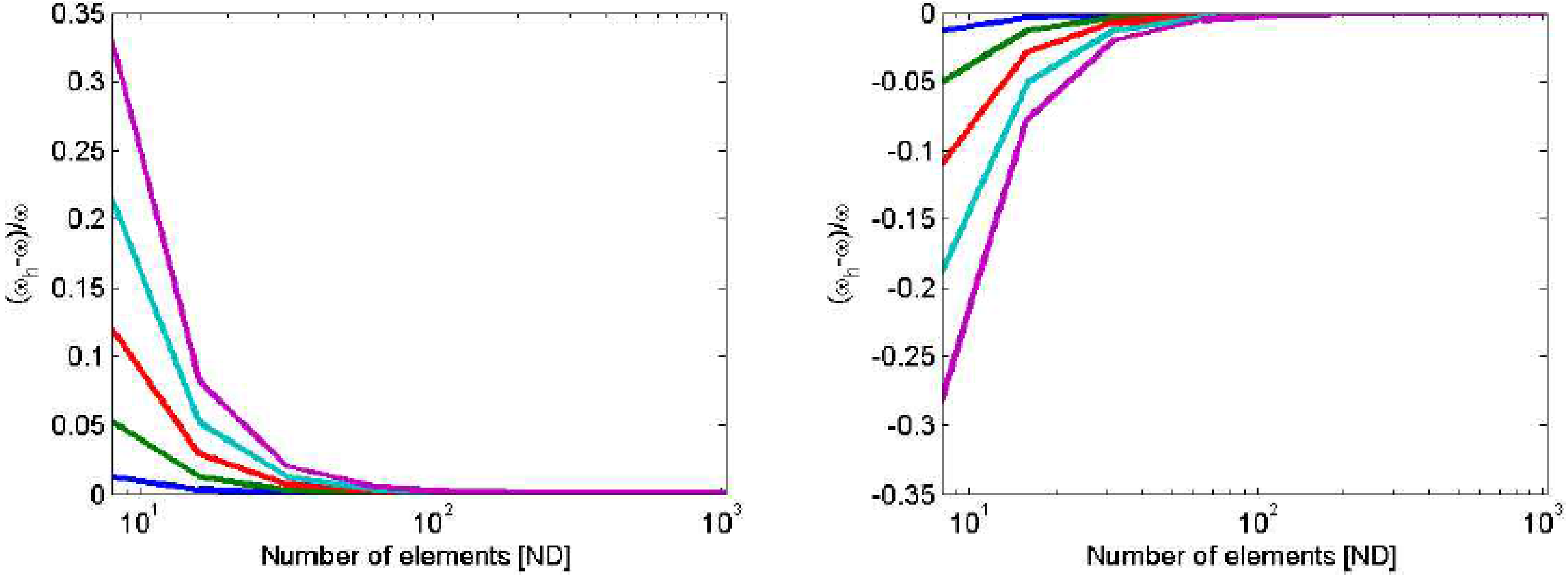
- Restoring time dependence will require tracking another second derivative term.
- In analogy with previous procedure, we can create another matrix, which yields:

$$0 = \boldsymbol{b} - \boldsymbol{K}\boldsymbol{w} - \boldsymbol{M}\ddot{\boldsymbol{w}}$$

- In absence of restoring force ($q = 0$), we have harmonic solutions ($\boldsymbol{w} = \boldsymbol{\phi} e^{i\omega t}$), so that:

$$\boldsymbol{K}\boldsymbol{\phi} - \omega^2 \boldsymbol{M}\boldsymbol{\phi} = 0$$

Dynamic Equilibrium



Convergence for first 5 frequencies as a function of the number of elements

Thermal Transport Mechanisms

- Convection: heat transfer by surface contact with gas or fluid molecules
- Conduction: volumetric heat transfer by propagation of phonons
- Radiative thermal transfer: emission of thermal photons from source to receiver

Thermal Transport: Convection

- Heat transfer by gas or fluid molecules
- Transfer rate per unit area given by:

$$Q = h(T_1 - T_2)$$

- Heat transfer constant *h* determined by many factors, including material choice, microstructures, fluid flow environment, etc.

Thermal Transport: Conduction

- Volumetric heat transfer through phonon transfer
- Change in heat energy per unit time is:

$$\frac{dU}{dt} = \frac{\partial}{\partial t}\int_V c_V T\, dV = -\oint \boldsymbol{q} \cdot d\boldsymbol{S} + \int_V Q\, dV$$

- Applying the divergence theorem thus yields:

$$\int_V \left[c_V \frac{\partial T}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{q} - Q \right] dV = 0$$

Thermal Transport: Conduction

- Local conservation of energy yields:

$$\frac{\partial u}{\partial t} = Q - \boldsymbol{\nabla} \cdot \boldsymbol{q}$$

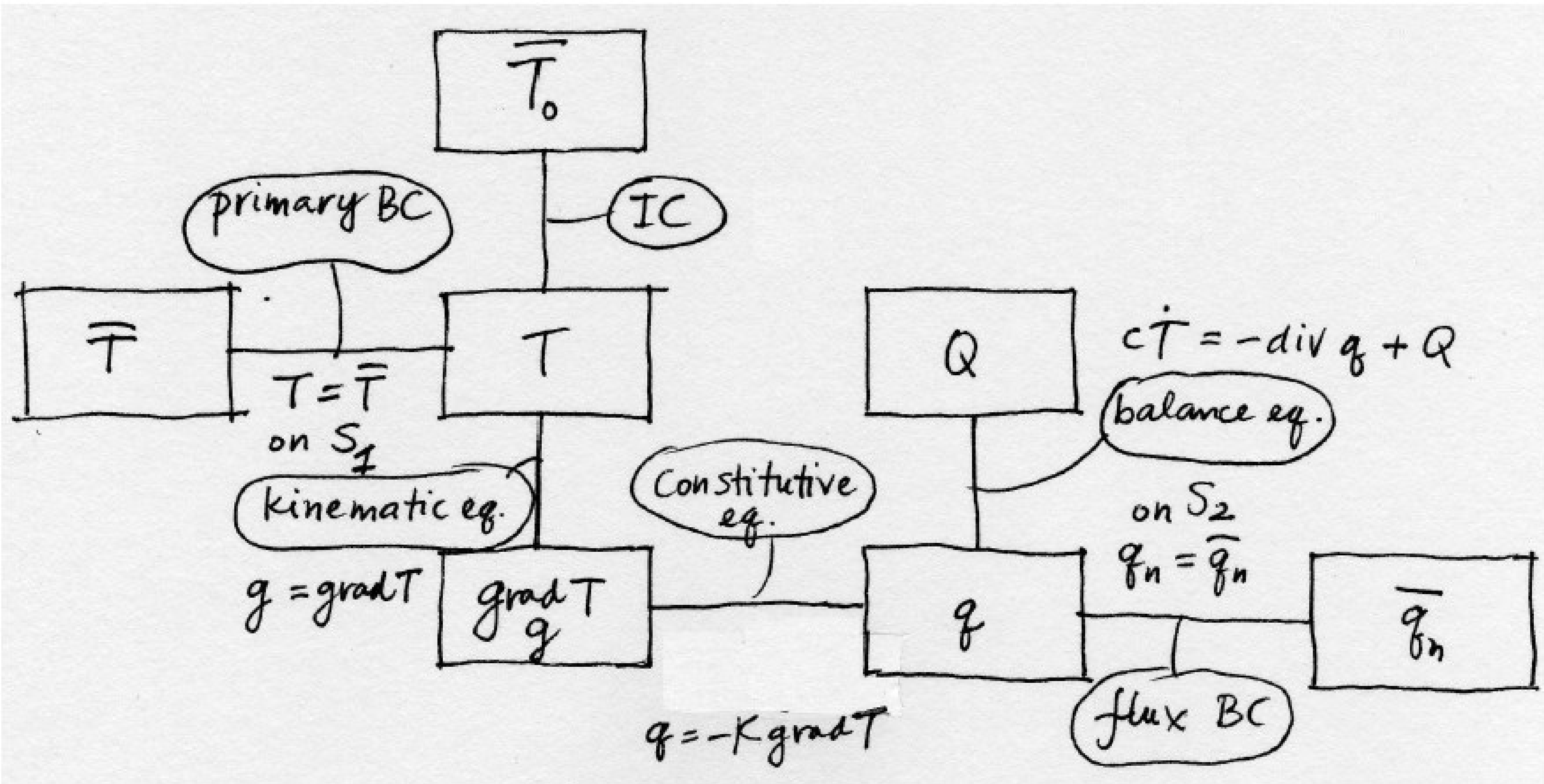
- Heat transfer rate quantified by Fourier's law:

$$\boldsymbol{q} = -k\boldsymbol{\nabla}T$$

- Combining with previous result yields thermal diffusion equation:

$$\frac{\partial u}{\partial t} - Q = k\nabla^2 T = \frac{k}{\rho c_v}\nabla^2 u \equiv \alpha\nabla^2 u$$

Thermal Transport: Conduction



Thermal Transport: Conduction

- Can formulate residual as:

$$r_B = c_V \frac{\partial T}{\partial t} - \boldsymbol{\nabla} \cdot k \boldsymbol{\nabla} T - Q$$

- Writing down Galerkin method:

$$\int_V \eta r_B \, dV = 0$$

Integrating by parts and dropping boundary yields:

$$\int_V \left[\eta c_V \frac{\partial T}{\partial t} + \nabla \eta \cdot k \nabla T - \eta Q \right] dV = 0$$

Thermal Transport: Conduction

- Now assume $\eta = N_i$ and $T = \sum_i N_i T_i$:

$$\sum_i \left[\left(\oint N_j c_V N_i \, dS \right) \frac{\partial T_i}{\partial t} + \left(\oint \nabla N_j \cdot k \nabla N_i \, dS \right) T_i - \oint N_j Q \, dS \right] = 0$$

- This can be simplified as:

$$\sum_i \left[C_{ji} \frac{\partial T_i}{\partial t} + K_{ji} T_i - L_j \right] = 0$$

Thermal Transport: Radiative Transfer

- Heat transfer via photon emission
- For a blackbody, total emission follows Stefan-Boltmann law:

$$P = \sigma T^4$$

- Net thermal transfer between two infinite surfaces becomes:

$$Q = \sigma(T_1^4 - T_2^4)$$

Thermal Transport: Radiative Transfer

- Emission for real materials depends on emissivity
- In thermal equilibrium, Kirchoff's law states emissivity=absorptivity at each wavelength
- Emission spectrum is given by:

$$\frac{dQ}{d\lambda} = \frac{2\pi hc^2\,\epsilon(\lambda)}{\lambda^5[e^{hc/\lambda kT} - 1]}$$

- Blackbody result recovered by setting $\epsilon(\lambda) = 1$ and integrating

Thermal Transport: Modeling

- Convection amounts to a boundary condition in most problems
- Will thus be first combined with conduction
- Strategy:
  - Create FEM grid for thermal conduction
  - Impose BC's from convection
  - (Optionally) include radiative transfer from disconnected bodies

Thermal Transport FEM

- Employ Galerkin method to reduce to linear algebra problem as before (see Petr Krysl's step-by-step introduction, Chapter 6):

$$C\dot{T} + (K + H)T = \sum_i L_i$$

- Where:

$$C_{ji} = \int_{S_c} N_j c_V N_i \, \mathrm{d}S \qquad L_{Q,j} = \int_{S_c} N_j Q \, \mathrm{d}S$$

$$K_{ij} = \int_{S_c} (\mathrm{grad} N_j) \, \boldsymbol{\kappa} (\mathrm{grad} N_i)^T \, \mathrm{d}S \qquad L_{q2,j} = -\int_{C_{c,2}} N_j \, \overline{q}_n \, \mathrm{d}C$$

$$H_{ji} = \int_{C_{c,3}} N_j \, h N_i \, \mathrm{d}C \qquad L_{q3,j} = \int_{C_{c,3}} N_j \, h T_a \, \mathrm{d}C$$

FAESOR: MATLAB FEM Toolbox

- Full rationale / background provided here:

http://hogwarts.ucsd.edu/~pkrysl/sofea/book-print-112005.pdf

- MATLAB code download here:

http://hogwarts.ucsd.edu/~pkrysl/faesor/faesor_publish.html

- Links also on course webpage

FAESOR: MATLAB FEM Toolbox

- 1D Meshing routine:

```
for j= 1:n+1
    fens=[fens fenode(struct ('id',j,'xyz',[x]));];
    x = x+(L/n);
end
gcells = [];
for j= 1:n
    gcells = [gcells gcell_l2(struct('id',j,'conn',[j j+1]))];
end
```

- Construct finite element block:

```
feb = feblock_defor_taut_wire(struct ('mater',mater_defor,...
    'gcells',gcells,...
    'integration_rule',simpson_1_3_rule,...
    'P',P));
geom = field(struct ('name',['geom'], 'dim', 1, 'fens',fens));
w    = 0*clone(geom,'w');
```

FAESOR: MATLAB FEM Toolbox

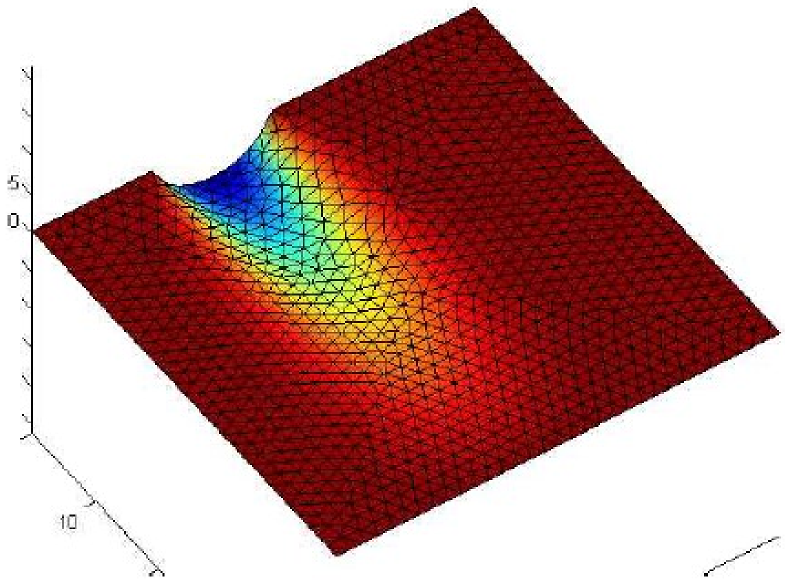
- Apply boundary conditions:

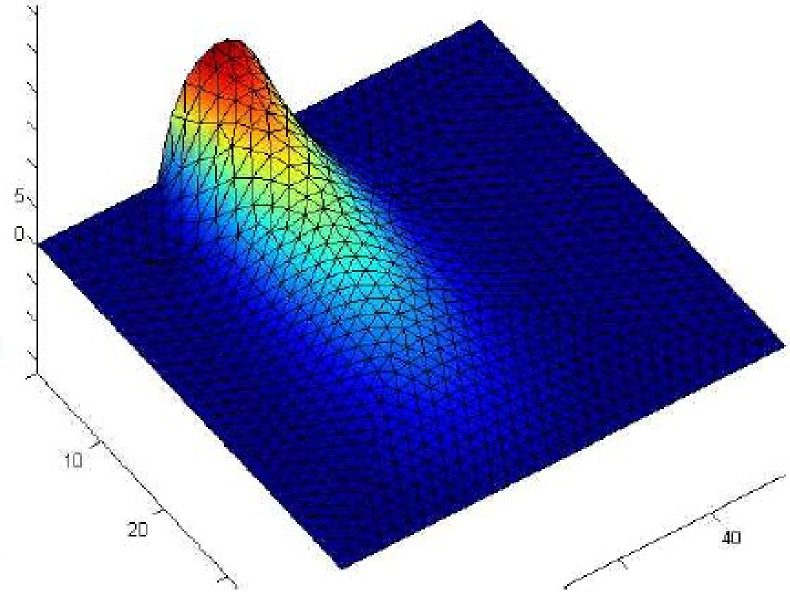
```
fenids=[1]; prescribed=[1]; component=[1]; val=0;
w   = set_ebc(w, fenids, prescribed, component, val);
w   = apply_ebc (w);
w   = numbereqns (w);
```

- Assemble and solve equations:

```
K = start (dense_sysmat, get(w, 'neqns'));
K = assemble (K, stiffness(feb, geom, w));
bl = body_load(struct ('magn',inline(num2str(q))));
F = start (sysvec, get(w, 'neqns'));
F = assemble (F, body_loads(feb, geom, w, bl));
w = scatter_sysvec(w, get(K,'mat')\get(F,'vec'));
```

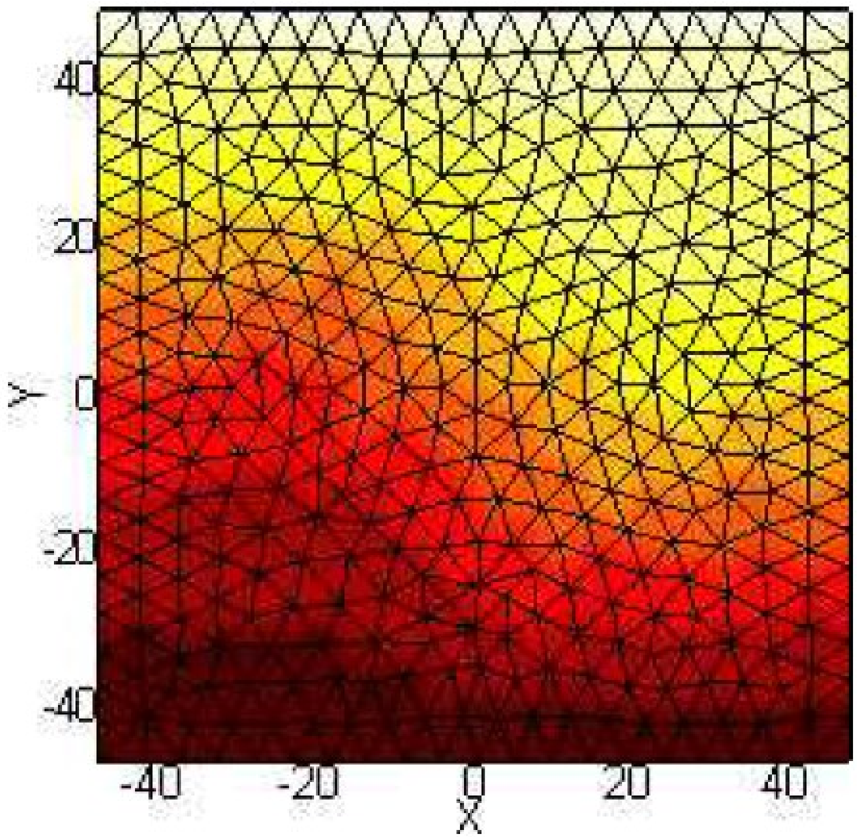
Results





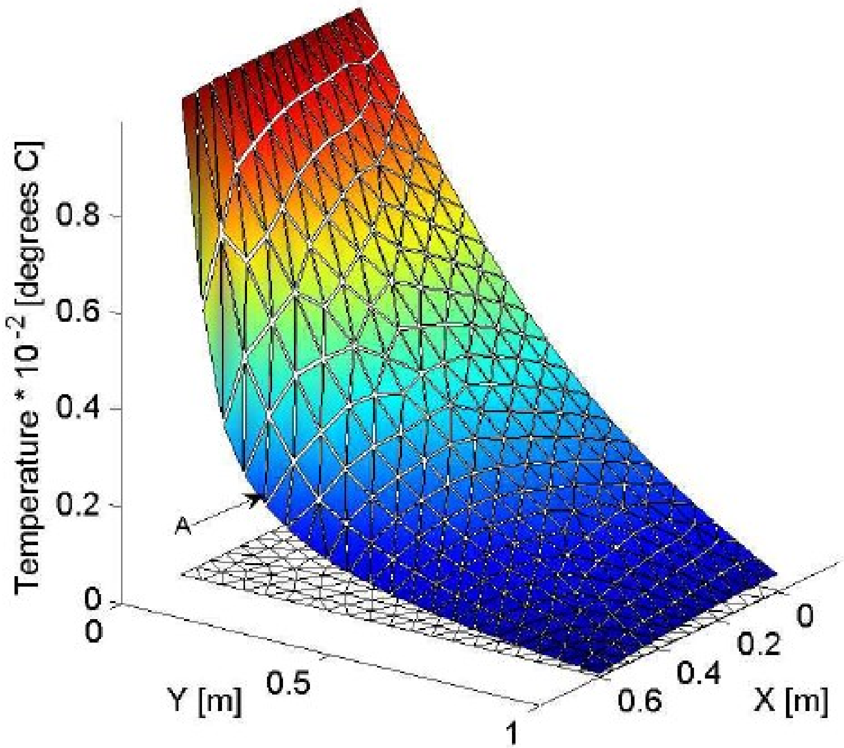
Steady-state solution for a thermally insulating medium, with a variable temperature placed along one surface

Results



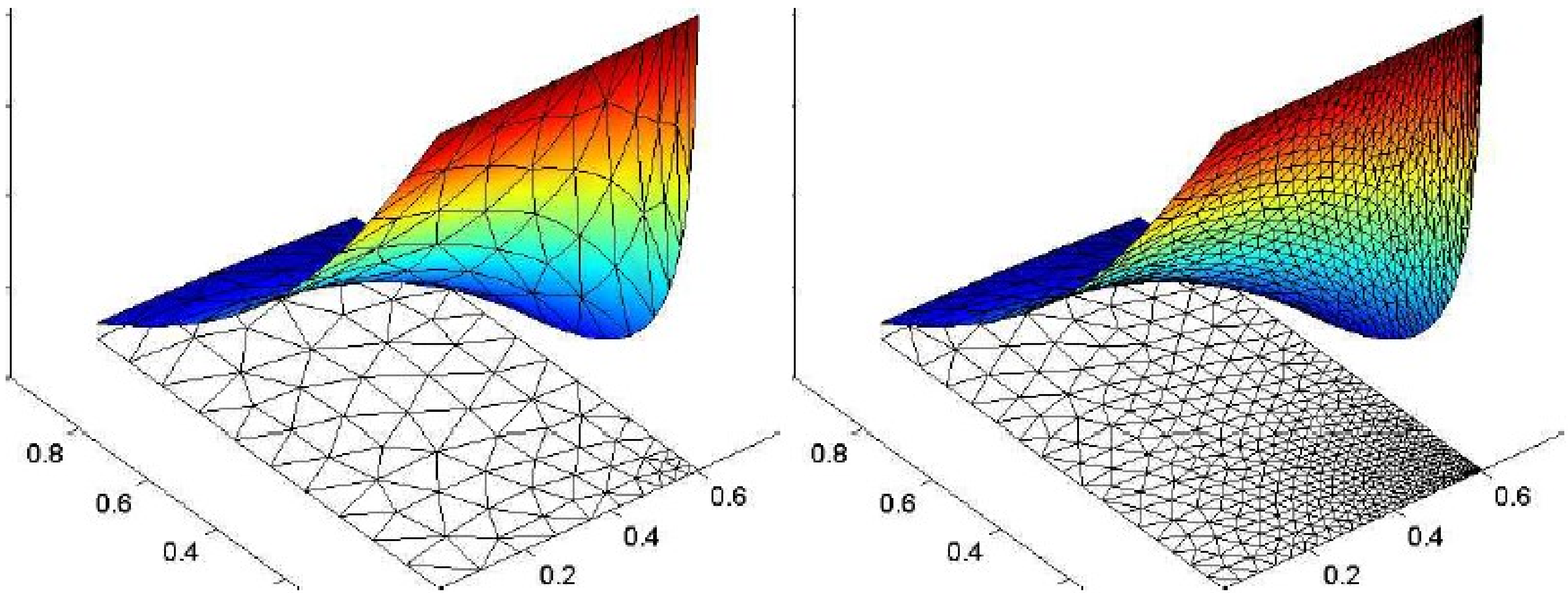
Steady-state solution for heat diffusion in an inhomogeneous medium

Thermal Conduction: Results



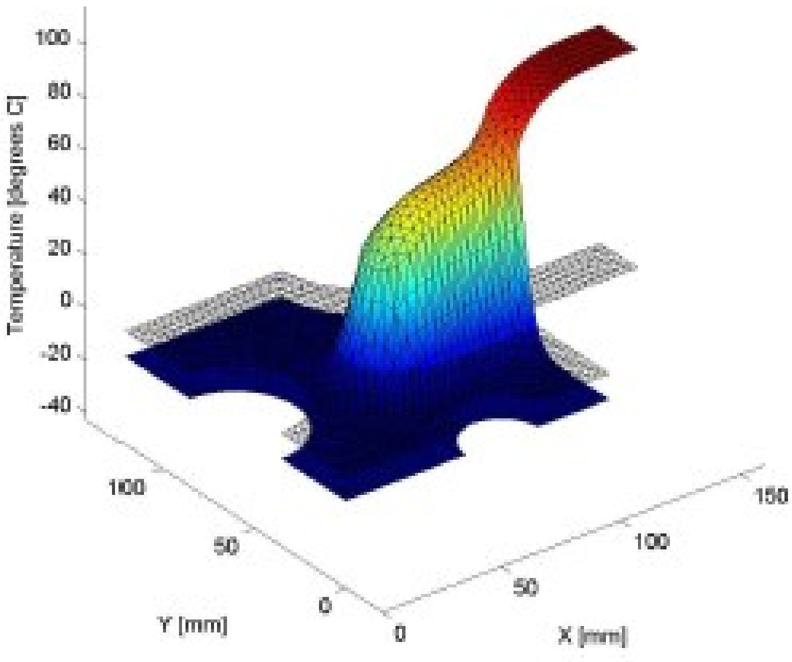
Steady-state thermal gradient between two adjoining walls with different temperatures

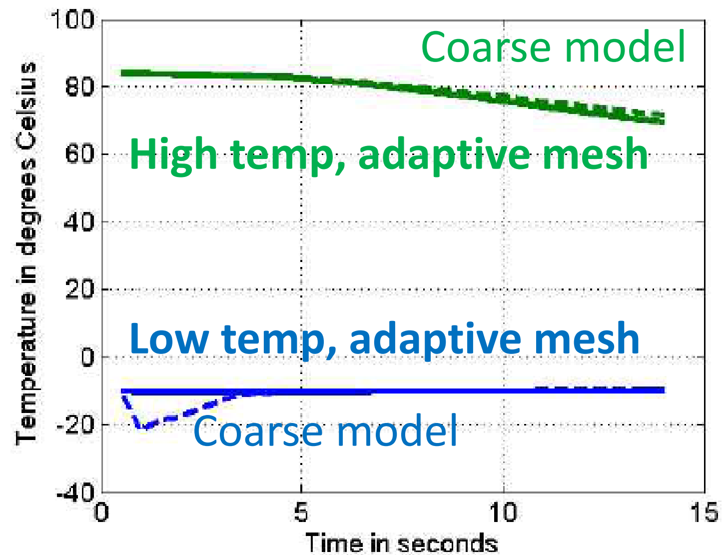
Thermal Conduction: Results



Example of adaptive mesh on T4 NAFEMS benchmark

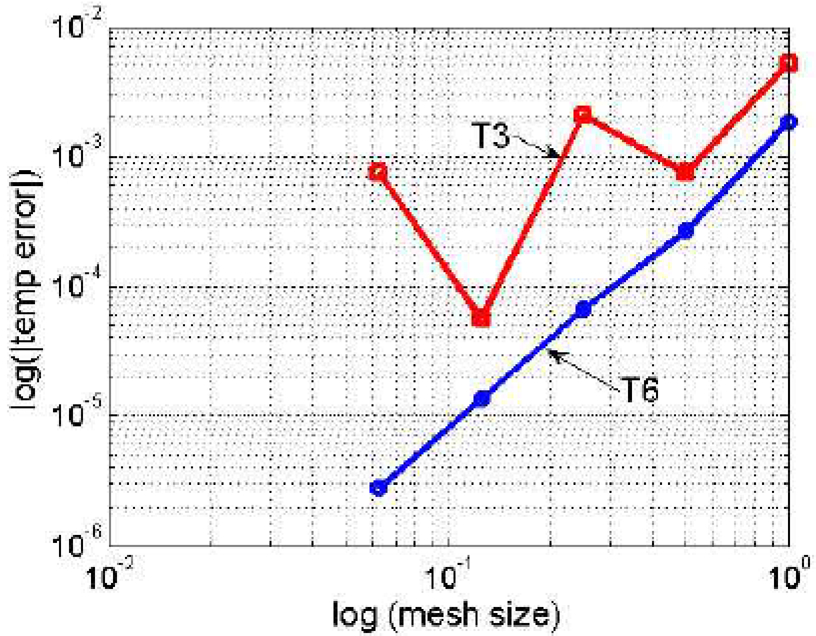
Thermal Conduction: Results





Transient cooling of a shrink-fitted assembly

Error Evaluation



- Error for linear elements T3 higher overall than quadratic elements T6
- Both decrease almost quadratically with mesh size
- T3 elements faster to compute

Next Class

- We will cover electronic transport