Introduction to Finite Element Modelling in Geosciences:

Code Verification

Day 4

Patrick Sanan

(See Chapter 9 in the Course Notes)

Verification and Validation

Verification :

- **Ensure that your code is solving the equations correctly**
- **Does the code produce approximate solutions that converge at the correct rate?**
- **A useful way to gain confidence in your code's correctness**

Validation :

- **ensure that your code produces physically-meaningful results**

(not covered here)

Convergence

- **If we can construct an exact solution to the continuous problem, we can use it to verify our code**
- **The FEM is well-known to converge, if implemented correctly!**
- **Convergence is expressed in terms of the grid spacing and an error norm**

$$||u - u^h||$$

- **If an analytical expression is available, one can compute the solution at different spacings and examine how quickly it converges**

$$||u - u^h|| \leq Ch^k$$

- **The rate of convergence can be computed by fitting a straight line to a log-log plot of the grid spacing versus the error**

But what if we don't have an exact solution to compare to?

The Method of Manufactured Solutions (MMS)

A very general approach

- **Choose your solution**
- **Plug it into your equations**
- **Extract the required boundary and forcing terms**
- **Now you have an exact solution to work with!**

Choosing your solution: make sure that it's complex enough! Don't just use polynomials. Using exponential or sinusoidal functions is often effective.

MMS: Example

Say our equation is the Poisson (steady-state heat) equation on the unit square. This example uses a constant conductivity, for simplicity, but the method works for any coefficients.

$$\nabla \cdot (\kappa \nabla T) = f(x, y)$$

Choose our solution (not just a polynomial!) :

$$T_{\mathrm{MMS}} \doteq \sin(x)\cos(2y)$$

Plug in:

$$f(x, y) = \kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = -5 \sin(x)\cos(2y)$$

Make sure boundary conditions are compatible, for example using Dirichlet BCs:

$$T(x, 0) = T_{\mathrm{MMS}}(x, 0),\ T(x, 1) = T_{\mathrm{MMS}}(x, 1),$$
$$T(0, y) = T_{\mathrm{MMS}}(0, y),\ T(1, y) = T_{\mathrm{MMS}}(1, y)$$

Voilà!

MMS: Example (continued)

Use our usual MATLAB code to solve the problem

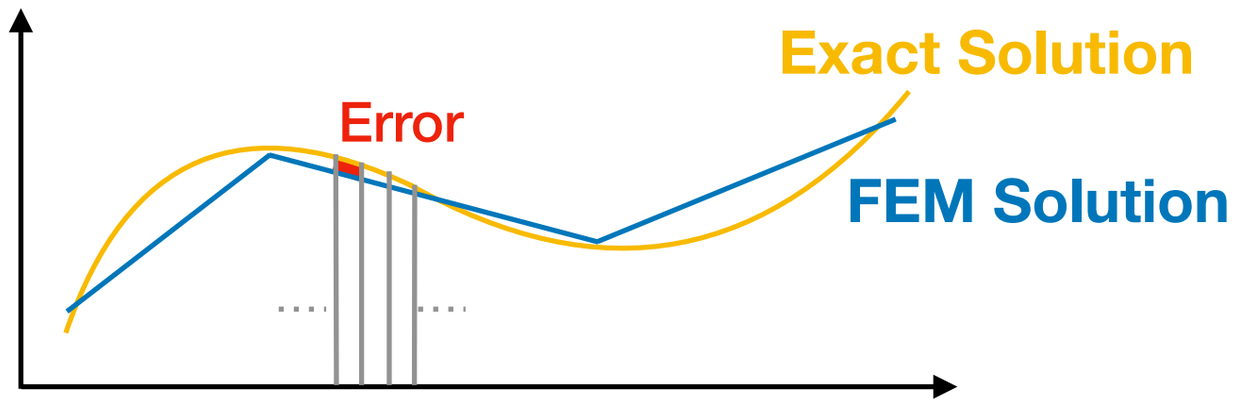
FEM_2D_TempDiff_MMS_PDS.m

```
% FEM code for 2D temperature diffusion
% Marcel Frehner, ETH Zurich, 2015
% Modified by Patrick Sanan, 2018

clear
close all
clc

% Manufactured Solution
MMS_val = @(x,y) sin(x).*cos(2*y);
MMS_f   = @(x,y,kappa) -kappa.*sin(x).*cos(2*y) - 4.*kappa.*sin(x).*cos(2*y); %% d/dx(k dT/dx) + d/dy(k dT/dy)

% GEOMETRICAL PARAMETERS
    Lx  =   10;
    Ly  =   10;

% PHYSICAL PARAMETERS
    kappa   =   1;

res_array = 2.^(3:7);
E1_array    =   NaN(1,length(res_array));
E2_array    =   NaN(1,length(res_array));
dxdy        =   NaN(1,length(res_array));

for ires = 1:length(res_array)
disp(['Resolution iteration step ' num2str(ires) ' of ' num2str(length(res_array))])

% NUMERICAL PARAMETERS
    nelx        =   res_array(ires);
    nely        =   res_array(ires);
    n_per_el    =   4;
    pts_per_el  =   4;

    % calculated from above
        el_tot  =   nelx*nely;
        nx      =   nelx+1;
        ny      =   nely+1;
        n_tot   =   nx*ny;

% NUMERICAL GRID
    dx          =   Lx/nelx;
    dy          =   Ly/nely;
    xcoord      =   0:dx:Lx;
    ycoord      =   0:dy:Ly;
    [x2d,y2d]   =   ndgrid(xcoord,ycoord);
    dxdy(ires)  =   dx*dy;

%     x2d         =   x2d   +   dx/2*(rand(nx,ny)-.5);
%     y2d         =   y2d +   dy/2*(rand(nx,ny)-.5);

    GCOORD      =   [x2d(:)';y2d(:)'];

% LOCAL-TO-GLOBAL MAPPING
    NoN     =   reshape(1:n_tot,nx,ny);
```

Compute the L2 Error

$$\left(\int_{\Omega} (T^h(x,y) - T(x,y))^2 d\vec{x} \right)^{1/2}$$

Important: to compute this properly, one must use a higher-order integration rule of each element. For example, use the trapezoid rule on a sub-grid.



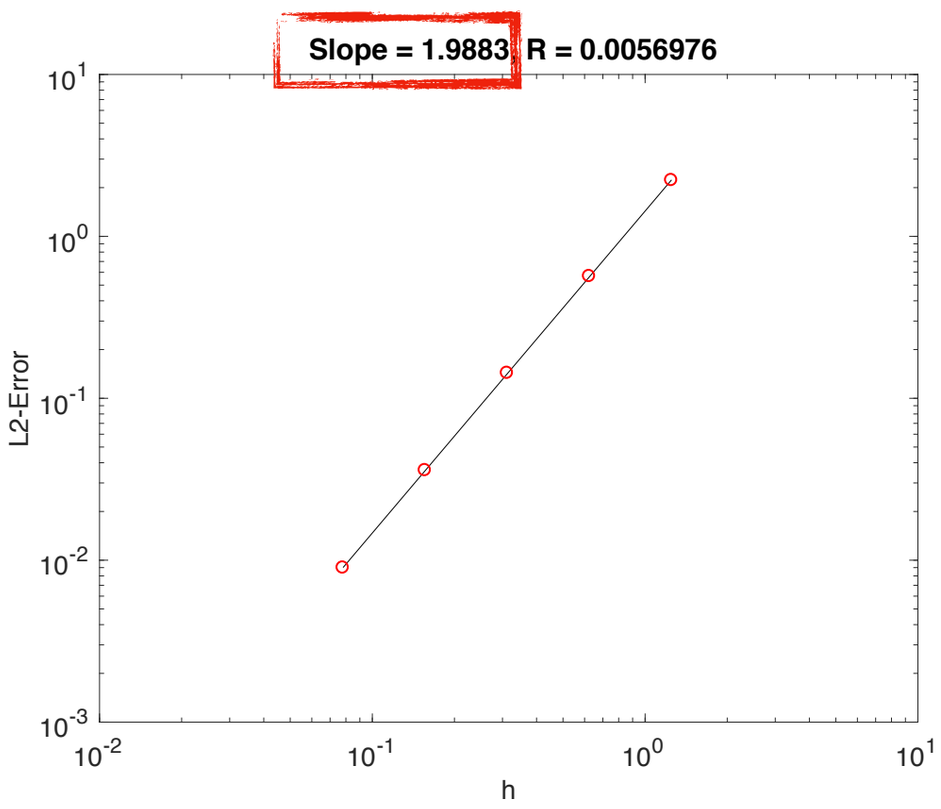
We cheat in the next slide, using a low-order rule and an incorrect normalization, and still get the expected convergence rate. This is not always the case!

MMS: Example (continued)

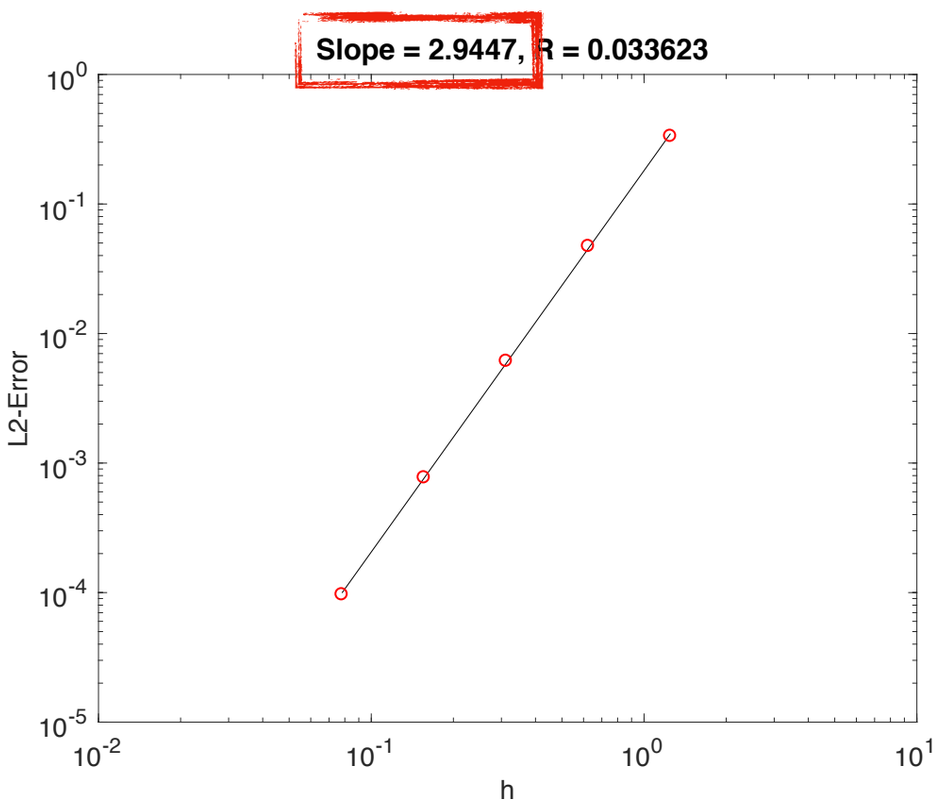
The expected convergence rate is given in the course notes

$$\left[\int_\Omega (T - T^h)^2 \, dV\right]^{1/2} \leq C_4 h^{p+1}$$

We run our code and plot the L2 error norms



Linear Elements $p = 1$



Quadratic Elements $p = 2$