`mcode.sty` Demo

Manuel Diaz, `manuel.ade@gmail.com`

June 24, 2013

NOTE

Partial Code,

```
  detJacobian=Le(e)/2;
  invJacobian=1/detJacobian;
  ngp = 4; % Prepare for integration with 4 gauss points
  [w,xi]=gauss1d(ngp);
  xc=0.5*(nodeCoordinates(elementDof(1))+...
    nodeCoordinates(elementDof(end)));
  Coefficient=zeros(ngp,ngp);
    for ip=1:ngp
        x_global = xc + detJacobian*xi(ip);
        [shape,naturalDerivatives]=shapeFunctionL2(xi(ip));
```

Full Code,

```
% Compute Normed Errors,
L2 = zeros(1,numberElements); en = zeros(1,numberElements);
for e=1:numberElements;
  % elementDof: element degrees of freedom (Dof)
  elementDof=elementNodes(e,:);
  detJacobian=Le(e)/2;
  invJacobian=1/detJacobian;
  ngp = 4; % Prepare for integration with 4 gauss points
  [w,xi]=gauss1d(ngp);
  xc=0.5*(nodeCoordinates(elementDof(1))+...
    nodeCoordinates(elementDof(end)));
  Coefficient=zeros(ngp,ngp);
    for ip=1:ngp
        x_global = xc + detJacobian*xi(ip);
        [shape,naturalDerivatives]=shapeFunctionL2(xi(ip));
```

```
        N=shape;
        B=naturalDerivatives*invJacobian;
        temp1 = detJacobian*w(ip)*(u_exact(x_global) - ...
            N*displacements(elementNodes(e,:)))^2;
        L2(e)=L2(e)+temp1;
        temp2 = detJacobian*w(ip)*(du_exact(x_global) - ...
            B*displacements(elementNodes(e,:)))^2;
        en(e)=en(e)+temp2;
    end
end

% Data for analysis,
h(i) = Le(1); % because of uniform mesh assumption
L2_linear(i) = sqrt(sum(L2));
en_linear(i) = sqrt(sum(en));
```

WOW!!

Manuel (manuel.ade@gmail.com)