Master of Science on Computational Science

Institute of Computational Science

Dr. Drosos Kourounis

31 Oct 2016, LAB5

Slow FE Assembly using MATLAB

```
function [M,K,b] = assembleDiscreteOperators(mesh)
  N    = mesh.N;
  Ne   = mesh.N_e;
  M    = sparse(N,N); K = sparse(N,N);
  b    = zeros(N,1);
  for e=1:N_e
    Me = makeMe(e, mesh);
    Ke = makeKe(e, mesh);
    fp = makebe(e, mesh);
    I = mesh.Elements(e, :);
    M(I, I) = M(I, I) + Me;
    K(I, I) = K(I, I) + Ke;
    b(I)    = b(I) + Me*fp;
  end % e loop
end
```

FE Assembly using MATLAB

```matlab
function [A] = assembleFast(mesh)
  ele_nodes= mesh.N_e*mesh.N_v^2;
  K = zeros(1, ele_nodes); iK = zeros(1, ele_nodes);
  jK = zeros(1, ele_nodes); i = 1:mesh.N_v;
  % ig = [1..nv, 1..nv, ... 1..nv] nv times
  ig        = repmat(i,1,mesh.N_v);
  % jg = [1 1 ... 1, 2 2 ... 2, ... , nv nv ... nv]
  jg        = repmat(1:mesh.N_v, mesh.N_v,1);
  jg        = jg(:)'; b = zeros(mesh.nodes,1);
  k         = 1:mesh.N_v^2;
  for e = 1:mesh.N_e
    Me    = makeMe(mesh.dx, mesh.dy); me = Me(:);
    Ke    = makeKe(mesh.dx, mesh.dy); ke = Ke(:);
    I     = mesh.Elements(e,:); iK(k) = I(ig); jK(k) = I(jg);
    x_e   = mesh.Points(I,:); f_e   = makeSource(x_e);
    b(I)  = b(I) + Me*f_e;
    K(k)  = ke + me; k   = k + mesh.N_v^2;
  end
  A = sparse(iK, jK, K);
end
```

Fast FE Assembly using MATLAB

```
function [As,b] = assembleFast(mesh)
  Me    = makeMe(mesh);
  Ke    = makeKe(mesh);
  me    = Me(:);
  ke    = Ke(:);
  M     = repmat(me, mesh.N_e, 1);
  K     = repmat(ke, mesh.N_e, 1);
  i=1:mesh.N_v;

  % ig = [1..N_v, 1..N_v, ... 1..N_v] N_v times
  ig=repmat(i, 1, mesh.N_v);

  % jg = [1 1 ... 1, 2 2 ... 2, ... , N_v N_v ... N_v]
  jg=repmat(1:mesh.N_v, mesh.N_v, 1); jg=jg(:)';

  iA    = mesh.Elements(:,ig)';
  jA    = mesh.Elements(:,jg)';
  A     = K + M;
  As    = sparse(iA(:), jA(:), A, mesh.N, mesh.N);
```

Fast FE Assembly of rhs using MATLAB

```
  % this is just a vector form of the source function
  f = makeSource(mesh.Points);
  I = mesh.Elements';
  F = f(I);
  b = Me*F;

  I = I(:);
  J = ones(mesh.N_v*mesh.N_e, 1);

  B = sparse(I, J, b(:), mesh.N, 1);

  b = full(B);

end
```