2D FEM Solution of the Helmholtz Equation (TMz)

^^

Description:

2D solution of the Helmholtz wave equation for the TMz polarization. A rectangular domain is assumed. The side walls are zero Neumann boundaries. The bottom wall is a Dirichlet boundary. The top wall is a Neumann boundary. This version assumes a homogeneous media. The domain is discretized into N triangular elements. First order Nodal elements are assumed. The results are plotted and the error is estimated against the exact solution.

^^

$f := \frac{300 \cdot 10^6}{16}$ $c0 := 2.998 \cdot 10^8$ $\lambda := \frac{c0}{f}$ $\lambda = 15.989$

$VDBC_0 := 1$ $k0 := \frac{2 \cdot \pi}{\lambda}$

Read in the coordinates of the nodes from an external file:

rNode := nodeList.txt

triNode := elemList.txt

numNodes := rows(rNode)

numNodes = 36

$rN := rNode^T$

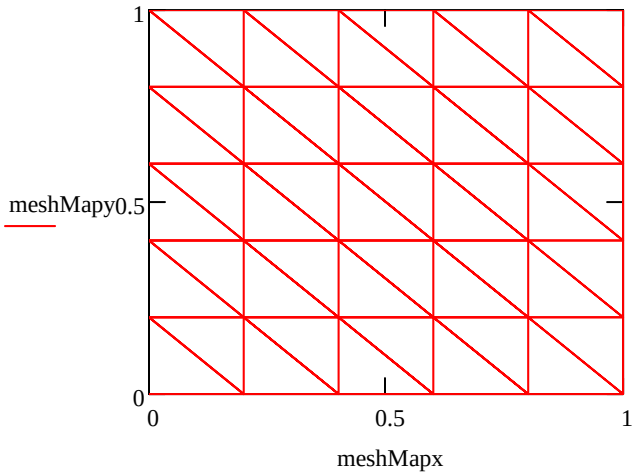
numTris := rows(triNode)

numTris = 50

Graph the Mesh:

```
meshMapx := | index ← 0
            | for i ∈ 0 .. numTris − 1
            |   for j ∈ 0 .. 2
            |     node ← triNode[i,j] − 1
            |     tmp[index] ← rNode[node,0]
            |     index ← index + 1
            | tmp
```

```
meshMapy := | index ← 0
            | for i ∈ 0 .. numTris − 1
            |   for j ∈ 0 .. 2
            |     node ← triNode[i,j] − 1
            |     tmp[index] ← rNode[node,1]
            |     index ← index + 1
            | tmp
```



Read in the boundary nodes:

The node # (1st column) and the Dirichlet boundary ID (2nd Column) are read in.
Note that a pre-processor has sorted all nodes so that nodes on Dirichlet boundaries are at the end of the list.

boundaryNodes := boundaryNodeList.txt

numBoundaryNodes := rows(boundaryNodes)

numBoundaryNodes = 12

DirchletboundaryNodes :=
- index ← 0
- for i ∈ 0 .. numBoundaryNodes − 1
  - $tmp_{index,0} \leftarrow boundaryNodes_{i,0}$ if $boundaryNodes_{i,1} = 1$
  - $tmp_{index,1} \leftarrow boundaryNodes_{i,1}$ if $boundaryNodes_{i,1} = 1$
  - index ← index + 1 if $boundaryNodes_{i,1} = 1$
- tmp

numDBoundaryNodes := rows(DirchletboundaryNodes) numDBoundaryNodes = 6

numIntNodes := numNodes − numDBoundaryNodes numIntNodes = 30

nodeBc :=
- for i ∈ 0 .. numNodes − 1
  - $tmp_i \leftarrow 0$
- for i ∈ 0 .. numDBoundaryNodes − 1
  - node ← $DirchletboundaryNodes_{i,0} - 1$
  - bc ← $DirchletboundaryNodes_{i,1}$
  - $tmp_{node} \leftarrow bc$
- tmp

NeumannNodes :=
index ← 0
for i ∈ 0 .. numBoundaryNodes − 1
  tmp_index ← boundaryNodes_{i, 0} if boundaryNodes_{i, 1} = 2
  index ← index + 1 if boundaryNodes_{i, 1} = 2
tmp

Compute the unitary vectors for the triangles:

ii := 0 .. numTris

aSub1Vectors :=
index ← 0
for i ∈ 0 .. numTris − 1
  node0 ← triNode_{i, 0} − 1
  node1 ← triNode_{i, 1} − 1
  asub1 ← rN^⟨node1⟩ − rN^⟨node0⟩
  for j ∈ 0 .. 2
    tmp_{j, i} ← asub1_j
tmp

aSub2Vectors :=
index ← 0
for i ∈ 0 .. numTris − 1
  node0 ← triNode_{i, 0} − 1
  node2 ← triNode_{i, 2} − 1
  asub2 ← rN^⟨node2⟩ − rN^⟨node0⟩
  for j ∈ 0 .. 2
    tmp_{j, i} ← asub2_j
tmp

rootG2d :=
index ← 0
for i ∈ 0 .. numTris − 1
  a1 ← aSub1Vectors^⟨i⟩
  a2 ← aSub2Vectors^⟨i⟩
  rg2d_i ← |a1 × a2|
rg2d

Compute the reciprocal unitary vectors for the triangles:

$$zhat := \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

aSup1Vectors :=
index ← 0
for i ∈ 0 .. numTris − 1
  asub2 ← aSub2Vectors^⟨i⟩
  $asup1 \leftarrow \frac{asub2 \times zhat}{rootG2d_i}$
  for j ∈ 0 .. 2
    tmp_{j, i} ← asup1_j
tmp

aSup2Vectors :=
index ← 0
for i ∈ 0 .. numTris − 1
  asub1 ← aSub1Vectors^⟨i⟩
  $asup2 \leftarrow \frac{zhat \times asub1}{rootG2d_i}$
  for j ∈ 0 .. 2
    tmp_{j, i} ← asup2_j
tmp

Compute the Global K-matrix:

$$Te := \frac{k0^2}{12}\cdot\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix}$$

$K :=$ for $j \in 0 .. numNodes - 1$

for $i \in 0 .. numNodes - 1$

$tmp_{i,j} \leftarrow 0$

for $i \in 0 .. numTris - 1$

for $j \in 0 .. 2$

$node_j \leftarrow triNode_{i,j} - 1$

$asup1 \leftarrow aSup1Vectors^{\langle i \rangle}$

$asup2 \leftarrow aSup2Vectors^{\langle i \rangle}$

$asup0 \leftarrow -asup1 - asup2$

for $j \in 0 .. 2$

$asup_{j,0} \leftarrow asup0_j$

$asup_{j,1} \leftarrow asup1_j$

$asup_{j,2} \leftarrow asup2_j$

for $q \in 0 .. 2$

$aq \leftarrow asup^{\langle q \rangle}$

for $s \in 0 .. 2$

$as \leftarrow asup^{\langle s \rangle}$

$Ke \leftarrow \left(\frac{aq \cdot as}{2} - Te_{q,s}\right)\cdot rootG2d_i$

$tmp_{(node_q),node_s} \leftarrow tmp_{(node_q),node_s} + Ke$ if $node_s < numIntNodes$

$tmp_{(node_q),node_q} \leftarrow 1$ otherwise

$blankspace \leftarrow 0$

tmp if $node_q < numIntNodes$

$K =$

	0	1	2	3	4
0	1.998	-1.001	0	0	0
1	-1.001	3.997	-1.001	0	0
2	0	-1.001	3.997	-1.001	0
3	0	0	-1.001	3.997	-1.001
4	0	0	0	-1.001	3.997
5	0	0	0	0	-1.001
6	-0.5	-5.147·10-4	0	0	0
7	0	-1.001	-5.147·10-4	0	0
8	0	0	-1.001	-5.147·10-4	0
9	0	0	0	-1.001	-5.147·10-4
10	0	0	0	0	-1.001
11	0	0	0	0	0
12	0	0	0	0	0

Compute the Right-Hand-Side:

$b :=$ for $j \in 0 .. numNodes - 1$

$rhs_j \leftarrow 0$

for $i \in 0 .. numTris - 1$

for $j \in 0 .. 2$

$node_j \leftarrow triNode_{i,j} - 1$

$bc_j \leftarrow nodeBc_{node_j} - 1$

$asup1 \leftarrow aSup1Vectors^{\langle i \rangle}$

$asup2 \leftarrow aSup2Vectors^{\langle i \rangle}$

$asup0 \leftarrow -asup1 - asup2$

for $j \in 0 .. 2$

$asup_{j,0} \leftarrow asup0_j$

$asup_{j,1} \leftarrow asup1_j$

$asup_{j,2} \leftarrow asup2_j$

for $q \in 0 .. 2$

$aq \leftarrow asup^{\langle q \rangle}$

for $s \in 0 .. 2$ if $node_q < numIntNodes$

$as \leftarrow asup^{\langle s \rangle}$

$Ke \leftarrow \left(\frac{aq \cdot as}{2} - Te_{q,s} \right) \cdot rootG2d_i$

$rhs_{node_q} \leftarrow rhs_{node_q} - Ke \cdot VDBC_{bc_s}$ if $node_s \geq numIntNodes$

$rhs_{node_q} \leftarrow VDBC_{bc_q}$ otherwise

$blankspace \leftarrow 0$

rhs

Compute the Solution:

$c := K^{-1} \cdot b$

$b =$

	0
0	0.501
1	1.001
2	1.001
3	1.001
4	1.001
5	0.5
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0

Compute the Error Relative to the Exact Solution:

$ymax := 1$

$$Exact := \left| \begin{array}{l} \text{for } i \in 0 .. numNodes - 1 \\ \quad \left| \begin{array}{l} y \leftarrow rN_{1,i} \\ tmp_i \leftarrow \dfrac{\cos[k0 \cdot (y - ymax)]}{\cos(k0 \cdot ymax)} \end{array} \right. \\ tmp \end{array} \right.$$

$Exact =$

	0
0	1.029457778195
1	1.029457778195
2	1.029457778195
3	1.029457778195
4	1.029457778195
5	1.029457778195
6	1.052560140149
7	1.052560140149
8	1.052560140149
9	1.052560140149
10	1.052560140149
11	1.052560140149
12	1.069164462115
13	1.069164462115
14	1.069164462115
15	1.069164462115

$c =$

	0
0	1.029472098139
1	1.02947085586
2	1.029453278535
3	1.029430599746
4	1.029413000984
5	1.029411719318
6	1.052621499798
7	1.052605800296
8	1.052559510568
9	1.052504111988
10	1.052457796435
11	1.052442029686
12	1.069327700942
13	1.069276320367
14	1.069180113109
15	1.069074146415

$Error := Exact - c$

$$meanError := \frac{1}{numIntNodes} \cdot \sum_{i=0}^{numIntNodes-1} \left| Error_i \right|$$

$meanError = 1.968 \times 10^{-4}$