

Numerical software 2

ANGENER

V. Dolejší

Charles University Prague, Faculty of Mathematics and Physics

Lecture 2

It modifies the input mesh \mathcal{T}_h and creates $\mathcal{T}_h^{\text{NEW}}$

Two versions

msekcce.karlin.mff.cuni.cz/~dolejsi/angen/angen3.0.htm

msekcce.karlin.mff.cuni.cz/~dolejsi/angen/angen3.1.htm

Downloading

```
gunzip angener3.1.tar.gz
```

```
tar xf angener3.1.tar
```

Instalation

```
make
```

Running (after preparation of data files)

```
./side
```

Files (1)

<code>main.f</code>	source file
<code>solver.f</code>	source file
<code>rest.f</code>	source file
<code>moving.f</code>	source file
<code>swapping.f</code>	source file
<code>insert.f</code>	source file
<code>remove.f</code>	source file
<code>rembou.f</code>	source file
<code>work.inc</code>	included file of array declaration
<code>makefile</code>	makefile used for translation

- `work.inc` declare the amount of allocated memory
- **Error in function MEMO: 10200 values miss in blank common**
⇒ not enough of memory
- increase dimension of array `work` in `work.inc`
- `rm -f main.o & make`

Files (2)

<code>paramet</code>	sample of input file
<code>profiles.01</code>	sample of input file – $\Omega = [0, 1] \times [0, 1]$
<code>triang.01</code>	sample of input file – $\Omega = [0, 1] \times [0, 1]$
<code>profiles.gam</code>	sample of input file – GAMM channel
<code>triang.gam</code>	sample of input file – GAMM channel
<code>profiles.2na</code>	sample of input file – double NACA profile
<code>triang.2na</code>	sample of input file – double NACA profile
<code>profiles.dca</code>	sample of input file – periodic DCA08 profile
<code>triang.dca</code>	sample of input file – periodic DCA08 profile
<code>spline.f</code>	first source code for spline interpolation
<code>library.f</code>	library subroutine for spline code
<code>readme</code>	first instruction
<code>manual.ps</code>	description of ANGENER code

Basic description (1)

Modes of ANGENER

- uniform mesh
- adapted mesh

Uniform mesh

Input files:

- `paramet` - contains the parameters for AMA algorithm,
- `profiles` - contains the description of curved parts of $\partial\Omega$,
- `triang` - input triangulation \mathcal{T}_h ,

Output files:

- `triangx` - new triangulation $\mathcal{T}_h^{\text{NEW}}$,
- `mesh` - figure of $\mathcal{T}_h^{\text{NEW}}$ for direct use in gnuplot.

Adapted mesh

Input files:

- `paramet` - contains the parameters for AMA algorithm,
- `profiles` - contains the description of curved parts of $\partial\Omega$,
- `triang` - input triangulation \mathcal{T}_h ,
- `results` - results u_h computed on \mathcal{T}_h .

Output files:

- `triangx` - new triangulation $\mathcal{T}_h^{\text{NEW}}$,
- `resultsx` - results u_h interpolated on $\mathcal{T}_h^{\text{NEW}}$,
- `mesh` - figure of $\mathcal{T}_h^{\text{NEW}}$ for direct use in gnuplot.

Important variables and arrays in ANGENER

$nelem$		# of triangles of \mathcal{T}_h
$npoin$		# of nodes (vertices) of \mathcal{T}_h
$nbelm$		# of boundary segments of \mathcal{T}_h
nbc		# of boundary components
nbp		# of non-polygonal parts of $\partial\Omega$
$ndim$		# of components of solution (=1 for scalar equation)
$x(i)$	$i = 1, \dots, npoin$	x-coordinates of nodes of \mathcal{T}_h
$y(i)$	$i = 1, \dots, npoin$	y-coordinates of nodes of \mathcal{T}_h
$lnd(i, j)$	$i = 1, \dots, nelem, j = 1, 2, 3$	nodes of triangles
$lbn(i, j)$	$i = 1, \dots, nbelm, j = 1, 2$	nodes of boundary edge
$ibc(i)$	$i = 1, \dots, nbelm$	type of BC for boundary edges

<i>ityp</i>	type of construction, $0 \leq ityp \leq ndim$
<i>ndim</i>	number of component of seeking solution
<i>ifv</i>	type of mesh association
<i>pos</i>	positivity
<i>numel</i>	prescribed number of elements for uniform triangulation
<i>eps1</i>	ε_1
<i>p</i>	p

- $ityp = 0$: a uniform triangulation is constructed
- $ityp > 0$: an adapted triangulation is constructed using $ityp$ -th component of u (it has $ndim$ components)
- $ifv = 0$: a cell-vertex scheme (continuous P_1)
- $ifv = 1$: a cell-centered scheme (discontinuous P_0)
- $pos > 0$: control the shape regularity

File paramet – example

```
0      ityp ( 0 - uniform trian, >0 - component of solution)
1      ndim
0      ifv ( 0 -cell vertex, 1- cell centered)
0.02   pos (=positivity)
500    numel (= c)
1.E+10 epsilon1
1.     p
```

File profile

a dense list of nodes defining nonpolygonal boundary:

2

16085

1. 0.249999985

0.999875009 0.249982268

0.999750018 0.249964535

⋮

0.999750018 0.250035465

0.999875009 0.250017732

1. 0.25

801

0.000 1.25

-0.00490878057 1.24999034

-0.00981743075 1.2499615

⋮

-0.0147257112 -1.24991322

-0.00981721189 -1.2499615

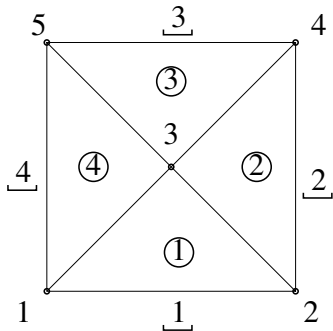
-0.00490856264 -1.24999034

0.000 -1.25

Files triang and triangx: unit square

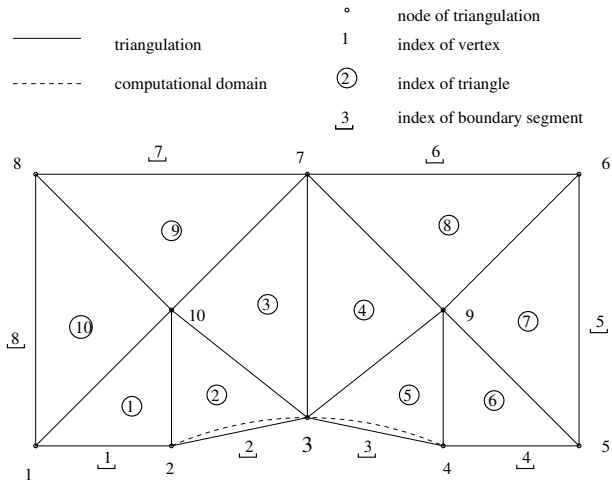
```
5 4 4      npoin nelel nbelm
0.0 0.0 0 0  periodic boundary
0.0 0.0      x(1) y(1)
1.0 0.0
0.5 0.5
1.0 1.0
0.0 1.0      x(5) y(5)
1 2 3        lnd(1,1) lnd(1,2) lnd(1,3)
2 4 3
4 5 3
5 1 3        lnd(4,1) lnd(4,2) lnd(4,3)
1 2 1        lbn(1,1) lbn(1,2) lbn(1,3)
2 4 2
4 5 3
5 1 4        lbn(4,1) lbn(4,2) lbn(4,3)
```

- node of triangulation
- 1 index of vertex
- ② index of triangle
- 3 index of boundary segm



Files triang and triangx: GAMM channel

```
10 10 8 4
0. 0. 0 0
-1.0 0.0
-0.5 0.0
0.0 0.1
0.5 0.0
1.0 0.0
1.0 1.0
0.0 1.0
-1.0 1.0
0.5 0.5
-0.5 0.5
1 2 10
2 3 10
3 7 10
3 9 7
3 4 9
4 5 9
5 6 9
6 7 9
7 8 10
8 1 10
1 2 3
2 3 3
3 4 3
4 5 3
5 6 2
6 7 4
7 8 4
8 1 1
```



ifv = 0 : cell-vertex, piecewise linear

$w(1, 1)$	$w(1, 2)$...	$w(1, ndim)$
$w(2, 1)$	$w(2, 2)$...	$w(1, ndim)$
$w(3, 1)$	$w(3, 2)$...	$w(3, ndim)$
\vdots	\vdots		\vdots
$w(npoin, 1)$	$w(npoin, 2)$...	$w(npoin, ndim)$

ifv = 1 : cell-centered, piecewise constant

$w(1, 1)$	$w(1, 2)$...	$w(1, ndim)$
$w(2, 1)$	$w(2, 2)$...	$w(1, ndim)$
$w(3, 1)$	$w(3, 2)$...	$w(3, ndim)$
\vdots	\vdots		\vdots
$w(nelem, 1)$	$w(nelem, 2)$...	$w(nelem, ndim)$

Example of performing the code

Unit square

```
cp triang.01 triang
cp profiles.01 profiles
./side
gnuplot> p 'mesh' w l
```

GAMM channel

```
cp triang.gam triang
cp profiles.gam profiles
./side
gnuplot> p 'mesh' w l
```

Solve numerically the following PDE:

$$\begin{aligned} -\Delta u &= 90x_1^8(1 - x_2^{20}) + 380x_2^{18}(1 - x_1^{10}), & \text{in } \Omega = (0, 1)^2, & (1) \\ u &= u_D & \text{on } \partial\Omega, \end{aligned}$$

where u_D is the exact solution given by

$$u(x_1, x_2) = (1 - x_1^{10})(1 - x_2^{20}), \quad (x_1, x_2) \in \Omega.$$

Instructions

- 1 Solve problem (1) by a suitable numerical method and by an arbitrary code based on your choice. You can use freely available software or you can write a simple own code.
- 2 Carry out several adaptation cycles using ANGENER.
- 3 Use a suitable visualization of the results, namely the adapted grids.

Combination of codes

- FEM code from Tutorials 11 (sparse) `femP1`
- ANGENER

Comments

- the same structure of files `triang`
- (1) is already implemented in `femP1`
- output of `femP1` in the ANGENER's format `results` is already available
- output of ANGENER: `resultsx` can be used for `femP1` (not necessary)
- careful setting of `paramet` file: *ityp*, *ifv*

Solve numerically the following PDE:

$$\begin{aligned} -\Delta u &= 0 & \text{in } \Omega &:= (-1, 1)^2 \setminus [0, 1]^2, \\ u &= u_D & \text{on } \partial\Omega, \end{aligned} \quad (2)$$

where u_D is the exact solution given by

$$u(r, \phi) = r^{2/3} \sin(2\phi/3), \quad (r, \phi) \in \Omega \text{ are polar coordinates.}$$

Instructions

- 1 Using ANGENER, generate a sequence of quasi-uniform grids of Ω and set experimental order of convergence (EOC).
- 2 Using the code ANGENER in combination with FEM generate a sequence of adaptively refined grids and set EOC.
- 3 Use a suitable visualization of the adapted grids and the corresponding solutions.