Anisotropic mesh adaptation – main tasks

Main task 3

Solve numerically the following PDE:

$$
\begin{aligned}
-\Delta u &= 90x_1^8(1-x_2^{20}) + 380x_2^{18}(1-x_1^{10}), \quad \text{in } \Omega = (0,1)^2, \\
u &= u_D \quad \text{on } \partial\Omega,
\end{aligned}
\tag{0.1}
$$

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 (0.1) by a suitable numerical method and by an arbitrary code based on your choice. You can used freely available software or you can write a simple own code. Hint: use the code from tutorials: link `http://msekce.karlin.mff.cuni.cz/~dolejsi/Vyuka/NS_source/FEM/index.html`
2. Carry out several adaptation cycles using the code ANGENER, see link `http://msekce.karlin.mff.cuni.cz/~dolejsi/angen/angen3.1.htm`
3. Use a suitable visualization of the adapted grids and the corresponding solutions.

Main task 4 We consider the L-shape computational domain $\Omega := (-1,1)\times(-1,1)\setminus[0,1]^2$. Solve numerically the following PDE:

$$\begin{aligned} -\Delta u &= 0, \qquad \text{in } \Omega, \\ u &= u_D \qquad \text{on } \partial\Omega, \end{aligned} \tag{0.2}$$

where u_D is the exact solution given by

$$u(r,\phi) = r^{2/3}\sin(2\phi/3)$$

with (r,ϕ) being the polar coordinates. Owing to the re-entrant corner, this problem features a singularity at the origin such that $u \in H^{5/3-\epsilon}(\Omega)$, $\epsilon > 0$. The presence of the singularity does not allow for faster convergence than $O(h^{2/3})$ on uniformly refined grids. Instructions:

1. Using the code ANGENER, generate a sequence of quasi-uniform grids of Ω and set the experimental order of convergence α in terms

$$\|u - u_h\| \approx ch^{\alpha}. \tag{0.3}$$

   For quasi-uniform grids we have $h \approx C/\sqrt{\#\mathscr{T}_h}$.

2. Using the code ANGENER in combination with FEM from Main task 3 generate a sequence of adaptively refined grids and set the experimental order of convergence α in terms

$$\|u - u_h\| \approx c\left(\frac{1}{\sqrt{\#\mathscr{T}_h}}\right)^{\alpha}. \tag{0.4}$$

3. Use a suitable visualization of the adapted grids and the corresponding solutions.

Setting of the working directory:

> `mkdir MainTask3`   //working directory

> `cd MainTask3/`

> `ln -s ../angener3.1/side .`   //link of ANGENER code to the current directory

> `cp ../angener3.1/profiles.01 profiles`   // copy of 'profiles' file

> `cp ../angener3.1/paramet .`   // copy of 'paramet' file, HAS TO BE MODIFIED

> `ln -s ../FEM-code/FEM-code3/femP1 .`   //link of FEM code to the current directory

> `cp ../FEM-code/FEM-code3/triang.square250  triang`   //copy of the initial mesh

Check list before the computation:

- file `paramet` has to be modified, (e.g., $ityp = 1$, $numel = 5$, $eps1 = 1E + 20$, $p = 5$)
- check the FEM code, if it solves the correct problem, i.e., file `sol.f90`, `function set_DBC` and `function set_RHS`
- if you modify FEM code, do not forgot to translate it, i.e., `make`

Adaptive computational cycle

> `./femP1 B S`   // FEM uses 'triang' and produce 'results' (and 'sol.gnu' for gnuplot)

> `./side`   // ANGENER uses 'triang' & 'results' and produce 'triangx' (and 'mesh' for gnuplot)

> `cp triangx triang`

- repeat previous 3 steps

Script for automatic computation, file 'compute.sh', running by > `./compute.sh 5`

```
#!/bin/csh

if( $#argv != 1) then
  echo "compute.sh <max_adapt_level>"
  exit
endif

  cp ../FEM-code/FEM-code3/triang.square250  triang
  rm -f res_out
@ max_lev = $argv[1]
@ i = 0

while($i <= $max_lev)
     ./femP1 B S
     cp sol.gn sol.lev_$i.gn
     if($i < $max_lev) then
       ./side
       cp triangx triang
     endif
@ i++
end

 echo " "
 echo " END of computation"
 echo " "

 more res_out
```

More sophisticated script for automatic computation, file 'compute.sh', running by > `./compute.sh 5`

```
#!/bin/csh

if( $#argv != 4) then
  echo "compute.sh <max_adapt_level> <numel <eps1> <p>"
  exit
endif

  cd ../FEM-code/FEM-code3/
  make
  cd ../../MainTask3
  cp ../FEM-code/FEM-code3/triang.square250  triang
  rm -f res_out

@ max_lev = $argv[1]
set numel = $argv[2]
set eps1 = $argv[3]
set pp = $argv[4]

cat << EOF > paramet
   1           ityp ( 0 - uniform trian,  >0 - component of solution)
   1           ndim
   0           ifv ( 0 -cell vertex, 1- cell centered)
 0.02         pos   (=positivity)
   $numel          numel (= c)
  $eps1       epsilon1
   $pp          p
EOF

@ i = 0
while($i <= $max_lev)
  ./femP1 B S > smazFE
  cp sol.gn sol.lev_$i.gn
   if($i < $max_lev) then
     ./side  > smaz
     cp triangx triang
   endif
```

```
@ i++
 end

 echo " "
 echo " END of computation"
 echo " "
 more res_out
```