Institute for Applied Mathematics
Heidelberg University
Prof. Dr. Thomas Richter

Problem Set 2 – Gascoigne Workshop
Summer Term 2013

Changing the Problem

The template for this problem set is given in: `cp -r /srv/share/gascoigne/2013/ps2 .`

First, you need to configure the example using `cmake ../src` in `ps2/bin`. After that, you have to compile the program with the command `make`.

In this problem set, we learn about changing the boundary conditions and the right-hand side of the problem. The problem `ps2` solves the Poisson equation

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \partial\Omega, \end{aligned}$$

with $f = 1$ and Dirichlet data $g = 0$ on the whole boundary of the domain.

Problem 2.1: (Changing the right-hand side)

The right-hand side in Gascoigne is specified by a class of type `DomainRightHandSide` as defined in the file `domainrighthandside.h` (see Chapter *"The right hand side"* in the tutorial)

Open the file `problem.h` in your directory `../ps2/src` and change the `class MyRhs`, which is derived from the class `DomainRightHandSide`, such that the following right-hand side is used:

$$f(x, y) = \begin{cases} 1 & \text{if } |x - 0.5| + |y - 0.5| \leq 0.25, \\ 0 & \text{else.} \end{cases}$$

For the absolute values use the C++ function `fabs(...)`.

Do not expect a large change in the resulting solution. Visualize the solution with VISUSIMPLE *using* `Actor->Scalarbar` *to get an idea of the absolute values of the solution and compare it to the solution of the first problem set ($f = 1$).*

Problem 2.2: (Changing the Dirichlet data)

Dirichlet boundary values can be set in the class `DirichletData` (see Chapter *"Dirichlet boundary data"*).

Modify the program, such that the Dirichlet data is given by the function:

$$u|_{\partial\Omega} = g, \quad g(x, y) = x^2 + y^2.$$

Compare the solution to the former solutions!

Problem 2.3: (Mixed boundary data)

If on a part of the boundary no Dirichlet data is indicated in the parameter file, Gascoigne uses the *natural boundary condition.* This condition depends on the type of differential equation. For the Poisson equation the natural boundary condition is the homogeneous Neumann condition

$$\partial_n u = 0.$$

This is seen by integration by parts:

$$(\nabla u, \nabla \phi)_\Omega = (-\Delta u, \phi)_\Omega + \langle \partial_n u, \phi \rangle_{\partial\Omega}. \tag{1}$$

The appropriate test space for the Poisson problem with Dirichlet conditions is $H_0^1(\Omega; \Gamma^D)$, which means

$$\phi = 0 \quad \text{on } \Gamma^D.$$

If we do not prescribe Dirichlet data on another part of the boundary $\Gamma^N \subset \partial\Omega$, we also do not restrict the test space there. Hence, the boundary term in the *weak formulation* in equation (1) changes to

$$0 = \langle \partial_n u, \phi \rangle_{\partial\Omega} = \langle \partial_n u, \underbrace{\phi}_{=0} \rangle_{\Gamma^D} + \langle \partial_n u, \phi \rangle_{\Gamma^N} = \langle \partial_n u, \phi \rangle_{\Gamma^N}.$$

That means

$$\partial_n u = 0 \text{ on } \Gamma^N.$$

Chapter *”Boundary Data“* describes how to choose different boundary conditions properly.

Our computational domain is a square given in the file `square.inp` in `../ps2/src`. You can visualize the domain with Visusimple by the command

```
visusimple square.inp &
```

The following code describes the boundary in `square.inp`:

```
0 4 line 0 1
1 2 line 1 2
2 4 line 2 3
3 2 line 3 0
```

Here the values 2 and 4 indicate ”colors“ corresponding to the different parts of the boundary (compare Chapter *”Definition of coarse meshes“*).

Change the parameter file, such that Dirichlet data is given only on the left and right parts of the boundary. These parts are indicated by the color value 2. On the top and bottom part, marked by the color value 4, we want to prescribe homogeneous Neumann data.

What happens if you use Neumann boundary values on all parts of the boundary?
For no Dirichlet values at the whole boundary, the `//Block BoundaryManager` has to be modified to

```
//Block BoundaryManager

dirichlet 0
```

Problem 2.4: (optional)

If we use Neumann data on all parts of the boundary, the solution is not unique: if u is a solution of $-\Delta u = f$, then for every $c \in \mathbb{R}$ another solution is given by $u + c$ since $-\Delta c = 0$. In this case, we must pose additional constraints on the solution

$$\int_\Omega u \; dx = 0.$$

This can be done by adding the following parameter to `//Block Solver` in the parameter file:

```
  pfilter 1 0
```

Solve **Problem 2.3** with Neumann conditions on the whole boundary with the above changes. Compare the results.