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

Problem Set 3 – Gascoigne Workshop
Summer Term 2013

Changing the equation

Get the third template as usual by typing: `cp -r /srv/share/gascoigne/2013/ps3 .`
Configure the code with `cmake` and compile it as in the previous problem sets.

This problem set deals with defining PDE's in GASCOIGNE (see Chapter *”Definition of the partial differential equations“*). We treat PDEs in GASCOIGNE PDE's as if they were nonlinear. Hence, every problem is solved with a Newton's method. Suppose, the problem is given in the weak formulation

$$a(u)(\phi) = (f, \phi) \quad \forall \phi \in V,$$

with a semilinear-form $a(\cdot)(\cdot)$ and a Hilbert space V. Then, every step of the Newton-iteration requires solving of the linear problem:

$$a'(u^k)(w^k, \phi) = (f, \phi) - a(u^k)(\phi) \quad \forall \phi,$$

where u^k is the last approximation, w^k the update, and $u^{k+1} = u^k + w^k$ the new solution. The Jacobian $a'(\cdot)(\cdot,\cdot)$ is defined as matrix of the directional derivatives:

$$a'(u)(w, \phi) := \frac{d}{ds} a(u + sw)(\phi)\Big|_{s=0},$$

For a linear problem (e.g., Poisson), we have

$$a'(u)(w, \phi) = a(w)(\phi).$$

In GASCOIGNE, we specify the semilinear-form $a(u)(\phi)$ in the method

```
void Form(VectorIterator& b, const FemFunction& U,
          const TestFunction& N) const;
```

and the matrix $a'(u)(w, \phi)$ in

```
void Matrix(EntryMatrix& A, const FemFunction& U,
          const TestFunction& M, const TestFunction& N) const;
```

Here, `U` is the current approximation u, `N` the test-function ϕ, and `M` is the trial-function, which corresponds to the update w. The result of the computation is saved in `b` and `A`, respectively.

Problem 3.1:

The class `MyEquation` in `../src/myequation.cc` and `../src/myequation.h` is an implementation of Laplace's equation in weak formulation:

$$a(u)(\phi) = (\nabla u, \nabla \phi).$$

Modify the bilinear form $a(\cdot)(\cdot)$ according to

$$a(u)(\phi) = (\nabla u, \nabla \phi) + (u, \phi),$$

and solve the modified problem. You only need to change `Form` and `Matrix`.

Problem 3.2:

Solve the nonlinear PDE

$$(\nabla u, \nabla \phi) + (u^2, \phi) = (f, \phi) \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega.$$

How does the Newton method converge?

Problem 3.3:

Solve the nonlinear PDE:

$$(\nabla u, \nabla \phi) + (\alpha u^\beta, \phi) = (f, \phi) \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega.$$

The parameters $\alpha, \beta \in \mathbb{R}$ must be read from the parameter file. The read procedure is to be implemented in the constructor of the class `MyEquation`. Here, the name of the parameter file is given as a parameter. Solve the equation for different values of α and β. Look at the output of the solvers to check if the problem is really solved!