Numerical Algorithms

Winter term 2019/20
Prof. Dr. Carsten Burstedde
Christopher Kacwin

UNIVERSITÄT BONN

Sheet 7

Submission on **Tuesday, 3.12.19 in class**.

Exercise 1. (Residual error)

Let $\Omega \subset \mathbb{R}^2$ be an open and bounded domain. We consider the problem:
Find $u \in H_0^1(\Omega)$ with

$$a(u,v) = F(v) \quad \forall v \in H_0^1(\Omega)\,,$$

where $a(u,v) = \int_\Omega \nabla u \nabla v \,\mathrm{d}x$ and $F(v) = \int_\Omega f v \,\mathrm{d}x$, $f \in L^2(\Omega)$. Furthermore, let $\mathcal{T}_h$ be a quasi-uniform triangulation of Ω and $\mathbb{V}_h \subset \mathbb{V}$ a conform finite element space. For the solution $u \in \mathbb{V}$ to the original problem and the approximation $u_h \in \mathbb{V}_h$ satisfying $a(u_h, v_h) = F(v_h)$ for all $v_h \in \mathbb{V}_h$, show that

$$\forall v \in \mathbb{V}\colon \quad a(u-u_h, v) = \sum_{T\in\mathcal{T}_h} \left(\int_T r_h v \,\mathrm{d}x + \frac{1}{2} \sum_{e \in \partial T \setminus \partial\Omega} \int_e R_h v \,\mathrm{d}S \right). \tag{7.1}$$

Here, $r_h = \Delta u_h + f$ denotes the residual on the corresponding element and $R_h = J(\nabla u_h \cdot n_e)$ denotes the jump of the normal derivative of u_h on the corresponding edge.

(5 points)

Exercise 2. (Trace theorem)

Consider an open and bounded domain $\Omega \subset \mathbb{R}^2$ with smooth boundary and the following version of the Trace theorem:

$$\|u\|_{0,\partial\Omega}^2 \le C_1 \|u\|_{0,\Omega}^2 + C_2 \|\nabla u\|_{0,\Omega}^2 \quad \forall u \in H^1(\Omega)\,. \tag{7.2}$$

Use a scaling argument to deduce the correct dependence of C_1, C_2 on $\operatorname{diam}(\Omega)$.

(5 points)

Programming Exercise 1. (Solving a PDE over a curved domain, part II)

In this programming exercise, we will solve a linear system arising from discretizing the Laplace equation over the unit circle. We will run a conjugate gradient solver that only requires matrix-vector products. Please refer to Sheet 4 for earlier definitions.

1. Write a routine that applies the $N \times N$ mass matrix to a given vector, storing the result in another. Ensure that the run time is linear in E and it uses only 1D Vandermonde matrices determined by a tensor-product Gauß quadrature of reasonable order. The scatter/gather matrices must be appplied based on your element-to-node table. You may use explicit geometry terms from your analytic definition.

2. Write a routine that applies the $N \times N$ stiffness matrix by the same principles.

3. Write a routine that applies the sum of the non-boundary diagonal block of the stiffness matrix and the boundary block of the unit diagonal matrix to a given vector. Do this by pre-/postprocessing the input/output vectors and calling the above full stiffness matrix function. To determine which entries are on the boundary, you may create a lookup list if you have not done so already.

4. Invent a smooth homogeneous solution u and compute the right hand side $f = -\Delta u$ on paper. Apply a conjugate gradient solver to a homogeneous Dirichlet problem with the right hand side f evaluated at your node points. Plot the result u_h and calculate the L_2 norm of the difference against the exact solution. How does the error norm scale with $h \approx 1/\sqrt{E}$? How does it depend on the size of the inner square?

(20 points)

The programming exercise has to be presented during the exercise class on Wednesday, December 11, 2019 in the computer room 2.038. You may bring your own laptop to class or prepare a solution that can be executed on the INS computers.