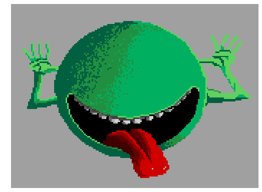
Course 401-3663-00L: Numerical Methods for Partial Differential Equations

Examination, Summer 2011

Prof. Ralf Hiptmair



Dont't panic !
Good luck !

Duration of examination: 180 minutes

Problem 1. **($L^2(\Omega)$-orthogonal projection (89 points))**

Let $\Omega \subset \mathbb{R}^2$ be a bounded polygon equipped with a triangular mesh $\mathcal{M}$. The $L^2(\Omega)$-orthogonal projection $\mathsf{P}_N f \in \mathcal{S}_1^0(\mathcal{M})$ of a function $f \in L^2(\Omega)$ is defined as the solution of the variational problem

$$\mathsf{P}_N f \in \mathcal{S}_1^0(\mathcal{M}): \quad \int_\Omega (\mathsf{P}_N f)(\boldsymbol{x}) v_N(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_\Omega f(\boldsymbol{x}) v_N(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \quad \forall v_N \in \mathcal{S}_1^0(\mathcal{M})\,. \tag{1}$$

(1a) **([I]** 5 points) Show that (1) has a unique solution.

(1b) **([I]** 5 points) Show that for any $f \in L^2(\Omega)$

$$\|f - \mathsf{P}_N f\|_{L^2(\Omega)} = \inf_{v_N \in \mathcal{S}_1^0(\mathcal{M})} \|f - v_N\|_{L^2(\Omega)}\,. \tag{2}$$

(1c) **([I]** 7 points) Assuming $f \in H^2(\Omega)$, give a meaningful, that is, reasonably sharp, bound for $\|f - \mathsf{P}_N f\|_{L^2(\Omega)}$ in terms of the meshwidth $h_\mathcal{M}$ and $|f|_{H^2(\Omega)}$.

Hint: Use (2).

(1d) **([I]** 7 points) Calculate the exact number of non-zero entries of the Galerkin matrix in terms of numbers of cells, edges, and vertices of the mesh, if the standard nodal basis ("tent function basis") of $\mathcal{S}_1^0(\mathcal{M})$ is used.

(1e) **([I]** 12 points) In a practical implementation of the finite element method the integrals in (1) are evaluated by means of local quadrature formulas. One option is the midpoint rule

$$\int_\Omega g(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \sum_{K \in \mathcal{M}} |K| g(\boldsymbol{m}_K)\,, \tag{3}$$

where $\boldsymbol{m}_K$ is the center of gravity of the triangle K, defined as $\boldsymbol{m}_K := \frac{1}{3}(\boldsymbol{a}_K^1 + \boldsymbol{a}_K^2 + \boldsymbol{a}_K^3)$, if $\boldsymbol{a}_K^1, \boldsymbol{a}_K^2, \boldsymbol{a}_K^3$ are the vertices of K.

Compute the element Galerkin matrix and element right hand side vector corresponding to (1), if the quadrature formula (3) is used together with the standard nodal basis ("tent function basis") of $\mathcal{S}_1^0(\mathcal{M})$.

(1f) (5 points) Show by means of an example that the Galerkin matrix computed in sub-problem (1e) may be singular.

Hint: You may study a "mesh" consisting of a single triangle.

(1g) (**[I]** 10 points) Another option is vertex based quadrature

$$\int_{\Omega} g(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} = \sum_{K \in \mathcal{M}} |K| \frac{1}{3} (g(\boldsymbol{a}_K^1) + g(\boldsymbol{a}_K^1) + g(\boldsymbol{a}_K^1)) \;. \tag{4}$$

Write down the linear system of equations arising from (1), the use of nodal basis functions, and the quadrature formula (4).

Hint: The matrix and vector entries can be expressed in terms of sums of cell volumes.

(1h) (**[I]** 15 points) The file `l2PrjLFE.m` contains the LehrFEM implementation of a function

```
[Pf,l2err] = l2PrjLFE(mesh,f)
```

that takes a mesh data structure `mesh` and a handle `f` to a function $f : \Omega \mapsto \mathbb{R}$ and returns the basis coefficient vector of $\mathsf{P}_N f$ in `Pf` and an approximation of $\|\mathsf{P}_N f - f\|_{L^2(\Omega)}$ in `l2err`. It relies on the quadrature rule (4) for the evaluation of the right hand side vector.

Use this function to perform a qualitative and quantitative study of the convergence of $\|\mathsf{P}_N f - f\|_{L^2(\Omega)}$ for $f(\boldsymbol{x}) := \exp(\|\boldsymbol{x}\|)$, $\Omega =]0,1[^2$ and a sequence of meshes obtained by five regular refinements of an intial mesh provided in the file `sqrmesh0.mat`. To this end extend the MATLAB template `cvgl2PrjLFE.m`.

Hint: Regular refinement of a triangular mesh in LehrFEM is achieved by means of the `refine_REG` function. You may use a reference implementation of `l2PrjLFE in l2PrjLFE.p`.

(1i) (10 points) Now we replace $\mathcal{S}_1^0(\mathcal{M})$ in (1) by $\mathcal{S}_{1,0}^0(\mathcal{M}) = \mathcal{S}_1^0(\mathcal{M}) \cap H_0^1(\Omega)$, which yields a modified discrete variational problem. Copy `l2PrjLFE.m` to `l2PrjzLFE.m` and implement in it a MATLAB function

```
[Pf,l2err] = l2PrjzLFE(mesh,f)
```

that solves the modified problem. The return values correspond to those of `l2PrjLFE`.

Hint: The supplied LehrFEM function `get_Bd_DOF(mesh)` can be used to tell whether a vertex of the mesh is located on the boundary $\partial\Omega$.

(1j) (10 points) Answer the questions of sub-problem (1h) for the modified function `l2PrjzLFE`.

Problem 2. (Least-squares Galerkin discretization (54 points))

On a bounded polygon $\Omega \subset \mathbb{R}^2$ we consider the stationary linear advection problem

$$\begin{aligned} \mathbf{v}(\boldsymbol{x}) \cdot \mathbf{grad}\, u &= f \quad \text{in } \Omega \,, \\ u &= g \quad \text{on } \Gamma_{\text{in}} := \{\boldsymbol{x} \in \partial\Omega : \; \mathbf{v}(\boldsymbol{x}) \cdot \boldsymbol{n} < 0\} \,, \end{aligned} \tag{5}$$

where $\mathbf{v} : \overline{\Omega} \mapsto \mathbb{R}^2$ is a given continuous velocity field, $f \in C^0(\overline{\Omega})$ a source term, and $g \in C^0(\overline{\Gamma}_{\text{in}})$ boundary values for the unknown u on the inflow boundary Γ_{in}.

The so-called *least squares variational formulation* of (5) boils down to a linear variational problem

$$u \in V: \quad \mathsf{a}(u,w) = \ell(w) \quad \forall w \in V \;, \tag{6}$$

with

$$\mathsf{a}(u,w) := (\boldsymbol{v} \cdot \mathbf{grad}\, u, \boldsymbol{v} \cdot \mathbf{grad}\, w)_{L^2} \quad , \quad \ell(w) := (\boldsymbol{v} \cdot \mathbf{grad}\, w, f)_{L^2} \;. \tag{7}$$

(2a) (**[I]** 5 points) Specify an appropriate function space V for the least squares variational formulation.

Hint: The Dirichlet boundary conditions in (5) should be treated as *essential boundary conditions*.

(2b) (**[I]** 10 points) The least squares variational formulation (6) is equivalent to a minimization problem for a functional J of the form

$$J(u) := \|T(u,f)\|^2_{L^2(\Omega)} \;, \tag{8}$$

where T is an expression involving the functions u and f. What is $T(u,f)$ in concrete terms.

(2c) (**[I]** 7 points) Consider the linear 2nd-order scalar elliptic boundary value problem

$$\begin{aligned} -\operatorname{div}(\mathbf{A}(\boldsymbol{x})\,\mathbf{grad}\, u) &= f \quad \text{in } \Omega \;, \\ u &= g \quad \text{on } \Gamma_{\text{in}} \;, \\ (\mathbf{A}(\boldsymbol{x})\,\mathbf{grad}\, u)\cdot \boldsymbol{n} &= 0 \quad \text{on } \partial\Omega \setminus \Gamma_{\text{in}} \;, \end{aligned} \tag{9}$$

where $\mathbf{A}: \overline{\Omega} \mapsto \mathbb{R}^{2,2}$ is a continuous matrix-valued function with $\mathbf{A}(\boldsymbol{x}) = \mathbf{A}(\boldsymbol{x})^T$ for all $\boldsymbol{x} \in \Omega$. Which choice of $\mathbf{A}$ makes the bilinear forms of the *standard* (i.e. not least squares) variational formulation of (9) and the variational problem (6) agree?

(2d) (**[I]** 7 points) The directory `EllBVP_LehrFEM` contains the complete LehrFEM implementation of a finite element solver for the boundary value problem (9); an approximate solution is computed by means of a piecewise linear Lagrangian finite element Galerkin discretization employing triangular meshes and local vertex based quadrature. The main routine is

```
u = solveellbvp(mesh,A_hd,f_hd,g_hd) ,
```

where `mesh` passes a LehrFEM mesh data structure complete with edge information and element flags, and `A_hd`, `f_hd`, `g_hd` are MATLAB function handles of type `@(x,varargin)` that provide the functions $\mathbf{A}$, f, and g. The inflow boundary is detected using the `markFlags` method, which gives inflow boundary edges an edge flag of `-1`, other boundary edges `-2` and interior edges `0`. The values of the finite element solution at the vertices are returned in the column vector `u`. The driver routine `solvebvp_main` demonstrates the use of this routine.

Copy the file `solveellbvp.m` to `solveadvbvp.m` and modify it so that it implements a LehrFEM routine

```
u = solveadvbvp(mesh,v_hd,g_hd) ,
```

that solves (5) in the case $f \equiv 0$ by means of the least squares Galerkin approach based on the variational formulation (6) and piecewise linear Lagrangian finite elements. The argument `v_hd` provides a function handle of type `@(x,varargin)` to the velocity field. This function should return a column vector $\in \mathbb{R}^2$. The `g_hd`-argument is a function handle of type `@(x,varargin)` and passes the real valued function g.

(2e) (**[I]** 15 points) Implement a MATLAB function

```
lsqphi = lsqrhs(mesh,v_hd,f_hd)
```

that computes the right hand side vector for the variational problem (6), when piecewise linear Lagrangian finite elements are employed for its Galerkin discretization.

As in (2d) the argument `mesh` contains a LehrFEM mesh data structure complete with edges and boundary information. The function handles `v_hd` and `f_hd` of type `@(x)` give the velocity field $\mathbf{v}$ and source term f, see (2d). Vertex based quadrature (trapezoidal rule) is to be used for local computations.

(2f) (10 points) Assume $g = 0$. Write a MATLAB function

```
u = solveadvlsq(mesh,v,f)
```

that computes the coefficient vector `u` of the least squares solution of (5) obtained by a linear Lagrangian finite element Galerkin solution of the related least squares variational problem (6). The arguments have the same meaning as in (2e).

Hint. You may copy large parts of your implementation of `solveadvbvp` from (2d). Also use `lsqrhs`, of which a reference implementation named `lsqrhsRef` is available in the file `lsqrhsRef.p`.

Problem 3. (Debugging finite elements (45 points))

Three different LehrFEM routines

`[A,phi] = assembleQFEX(mesh,f_hd)`, $\quad \mathtt{X} \in \{1, 2, 3\}$

purport to provide the Galerkin matrix and right hand side vector for the finite element discretization of the variational problem

$$u \in H^1(\Omega): \quad \mathsf{a}(u, v) := \int_\Omega \mathbf{grad}\, u \cdot \mathbf{grad}\, v \,\mathrm{d}\boldsymbol{x} = \ell(v) := \int_\Omega f(\boldsymbol{x}) v(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \quad \forall v \in H^1(\Omega) \tag{10}$$

using *quadratic* Lagrangian finite elements (space $\mathcal{S}_2^0(\mathcal{M})$) on a triagular mesh $\mathcal{M}$ of some polygon $\Omega \subset \mathbb{R}^2$. The argument `mesh` is is supposed to pass a LehrFEM mesh data structure complete with edge information and element flags, whereas `f_hd` contains a handle to the source function f of type `@(x,varargin)`.

The routines return the Galerkin matrix and right hand side vector for (10) w.r.t $\mathcal{S}_2^0(\mathcal{M})$ based on standard global shape functions of $\mathcal{S}_2^0(\mathcal{M})$, which are associated with interpolation nodes in the vertices and midpoints of edges. The following ordering of global shape functions is used: first we number the basis functions belonging to vertices based on the vertex array in the `mesh` data structure. Second, the basis functions associated with edges are ordered according to the numbering of the edges in the `mesh.Edges` field.

(3a) (**[I]** 10 points) Write a MATLAB function

```
Iu = interpolateQFE(mesh,u)
```

that accepts a LehrFEM mesh data structure `mesh` and a handle `u` to a real valued function u and returns the basis coeffcients of the nodal interpolant $\mathsf{I}_2 u \in \mathcal{S}_2^0(\mathcal{M})$.

(3b) (**[I]** 10 points) Determine a sharp bound $T(h_\mathcal{M})$ in the estimate

$$|\mathsf{a}(u,u) - \mathsf{a}(\mathsf{I}_2 u, \mathsf{I}_2 u)| \leq C T(h_\mathcal{M}) \;, \tag{11}$$

where $u : \overline{\Omega} \mapsto \mathbb{R}$ is supposed to be smooth and the unknown constant $C > 0$ may depend only on Ω and the shape regularity measure of $\mathcal{M}$.

Use the following result:

Theorem. *Let $\Omega \subset \mathbb{R}^d$, $d = 1,2,3$, be a bounded polygonal/polyhedral domain equipped with a simplicial mesh $\mathcal{M}$. Then the following interpolation error estimate holds for the nodal interpolation operator I_2 onto $\mathcal{S}_2^0(\mathcal{M})$*

$$\|u - \mathsf{I}_2 u\|_{H^1(\Omega)} \leq C h^{\min\{3,k\}-1} |u|_{H^k(\Omega)} \quad \forall u \in H^k(\Omega) \;, \quad k = 2,3 \;,$$

with a constant $C > 0$ depending only on k and the shape regularity measure $\rho_\mathcal{M}$.

(3c) (5 points) Write a MATLAB function

```
enu = test_assembleQFE(mesh,assfn)
```

that computes $\mathsf{a}(\mathsf{I}_2 u, \mathsf{I}_2 u)$ for $u(\boldsymbol{x}) = \exp(\|\boldsymbol{x}\|^2)$ and the domain triangulated by the mesh described by the LehrFEM mesh data structure `mesh`. The argument `assfn` passes a handle to an assembly routine for quadratic Lagrangian finite element with the calling syntax of `assembleQFEX` introduced above.

Hint. Use the function `interpolateQFE` developed in (3b). A reference implementation of this function is supplied as `interpolateQFERef` in the file `interpolateQFERef.p`.

(3d) (10 points) The file `squaremesh.mat` contains the LehrFEM mesh data structures for five increasingly refined triangular meshes of $\Omega =]0,1[^2$ in the variables `mesh1`, $\ldots$, `mesh5`. For each of the assembly routines `assembleQFEX(mesh,f)`, `X` $\in \{1,2,3\}$ plot $|\mathsf{a}(u,u) - \mathsf{a}(\mathsf{I}_2 u, \mathsf{I}_2 u)|$ for these meshes and the function $u(\boldsymbol{x}) = \exp(\|\boldsymbol{x}\|^2)$ from (3c) against the meshwidth $h_\mathcal{M}$ in a suitable scale.

Hint. Use the function `test_assembleQFE` implemented in (3c), for which a reference implementation is available in `test_assembleQFE.p`. The mesh width $h_\mathcal{M}$ of a mesh stored in the LehrFEM data structure `mesh` can be computed by calling `get_MeshWidth(mesh)`.

Hint. You may use

$$|u|^2_{H^1(\Omega)} = 23.7608$$

for $\Omega =]0,1[^2$.

(3e) (10 points) Which implementations of the assembly routine are wrong, which are correct? Explain your answer.

References

[NPDE] Lecture slides for course ”Numerical Methods for Partial Differential Equations”, Subversion Revision