MA5233 Computational Mathematics

Homework Sheet 7

Simon Etter, 2019/2020

Final deadline: 1 November 2019, 7pm

Quadratic finite elements

We have seen in class that linear finite elements on an equispaced mesh $(x_k = \frac{k}{n+1})_{k=0}^{n+1}$ leads to an approximate solution u_n which satisfies the error estimates

$$\|u - u_n\|_{H^1([0,1])} = \mathcal{O}\big(n^{-1}\big) \qquad \text{and} \qquad \|u - u_n\|_{L^2([0,1])} = \mathcal{O}\big(n^{-2}\big).$$

These convergence rates can be improved if we apply Galerkin's method to the subspace of piecewise quadratic rather than piecewise linear functions. Using the Céa and Aubin-Nitsche lemmas as in the linear case, one can show that this quadratic finite element method satisfies the error estimates

$$\|u - u_n\|_{H^1([0,1])} = \mathcal{O}\big(n^{-2}\big) \qquad \text{and} \qquad \|u - u_n\|_{L^2([0,1])} = \mathcal{O}\big(n^{-3}\big).$$

This homework sheet will demonstrate the implementation of this method.

A basis $(\phi_k)_{k=1}^{2n+1}$ for the space of continuous and piecewise quadratic functions on an equispaced mesh $(x_k = \frac{k}{n+1})_{k=0}^{n+1}$ is given by

$$\phi_{2k}(x) := \phi_{\text{vertex}}\Big((n+1)\,\big(x - \tfrac{k}{n+1}\big)\Big), \qquad \phi_{2k+1}(x) := \phi_{\text{edge}}\Big((n+1)\,\big(x - \tfrac{k}{n+1}\big)\Big)$$

where

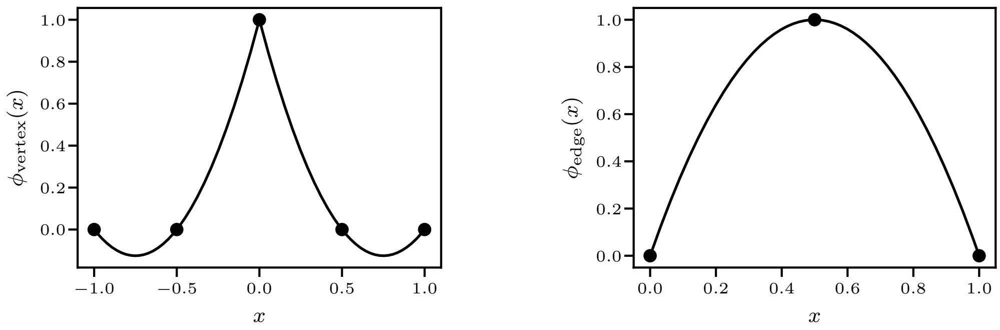
$$\phi_{\text{vertex}}(x) := \begin{cases} 2\,(x+1)\,(x+\frac{1}{2}) & \text{if } -1 \le x \le 0, \\ 2\,(x-1)\,(x-\frac{1}{2}) & \text{if } 0 \le x \le 1, \\ 0 & \text{otherwise} \end{cases}$$

and

$$\phi_{\text{edge}}(x) := \begin{cases} 4\,x\,(1-x) & \text{if } 0 \le x \le 1, \\ 0 & \text{otherwise,} \end{cases}$$

see also Figure 1. We observe that $\phi_k\big(\frac{\ell}{2\,(n+1)}\big) = \delta_{k\ell}$, i.e. $\phi_k(x)$ with k even is one in exactly one mesh point and zero in all other mesh points and the midpoints, while $\phi_k(x)$ with k odd is one in precisely one midpoint and zero in all mesh points and all other midpoints. Once the coefficients $c \in \mathbb{R}^{2n+1}$ in

$$u_n(x) = \sum_{k=1}^{2n+1} c_k\,\phi_k(x)$$



(a) $\phi_{\text{vertex}}(x)$ (b) $\phi_{\text{edge}}(x)$

Figure 1: Plots of $\phi_{\text{vertex}}(x)$ and $\phi_{\text{edge}}(x)$. Note that $\phi_{\text{vertex}}(x)$ is plotted on $[-1,1]$ while $\phi_{\text{edge}}(x)$ is plotted on $[-1,1]$.

have been computed, the Galerkin solution $u_n(x)$ can thus be evaluated by quadratic interpolation of the data points

$$\left(\tfrac{k}{n+1}, c_{2k}\right), \quad \left(\tfrac{k+1/2}{n+1}, c_{2k+1}\right), \quad \left(\tfrac{k+1}{n+1}, c_{2k+2}\right)$$

where k is such that the evaluation point x falls into the interval $[\frac{k}{n+1}, \frac{k+1}{n+1}]$ and where we assume that $c_0 = c_{2n+3} = 0$.

1. Complete the functions `querp()` and `d_querp()` which evaluate the quadratic interpolant and its derivative. You can test your implementation with the functions `test_querp()` and `test_d_querp()`.

2. Complete the function `galerkin_matrix()` which assembles the matrix

$$\left(A_{k\ell} := \int_0^1 \phi_k'(x)\, \phi_\ell'(x)\, dx \right)_{k,\ell=1}^{2n+1}.$$

Hints.

- The following integrals may help you:

$$\int_{-\infty}^{\infty} \Big(\phi_{\text{vertex}}'(x)\Big)^2 dx = \frac{14}{3} \qquad \int_{-\infty}^{\infty} \phi_{\text{vertex}}'(x)\, \phi_{\text{edge}}'(x)\, dx = -\frac{8}{3}$$

$$\int_{-\infty}^{\infty} \Big(\phi_{\text{edge}}'(x)\Big)^2 dx = \frac{16}{3} \qquad \int_{-\infty}^{\infty} \phi_{\text{vertex}}'(x)\, \phi_{\text{vertex}}'(x-1)\, dx = \frac{1}{3}$$

- A has five nonzero diagonals, i.e. $A_{k\ell} \neq 0$ for $|k-\ell| \leq 2$. Such a matrix can be conveniently assembled using the function `spdiagm()` in the `SparseArrays` package. Type `?spdiagm` in the REPL to learn more.

- The diagonal $d_k := A_{kk}$ has an alternating pattern, i.e. $d_k = d_{k+2}$ but $d_k \neq d_{k+1}$. Such a vector can be conveniently assembled as follows.

```
julia> d = zeros(5)
       d[1:2:end] .= 1
       d[2:2:end] .= 2
       d
5-element Array{Float64,1}:
 1.0
 2.0
 1.0
 2.0
 1.0
```

3. Complete the function `right_hand_side()` which assembles the vector

$$\Big(b_k := \int_0^1 f(x)\, \phi_k(x)\, dx \Big)_{k=1}^{2n+1}.$$

   using the composite Simpson rule on the mesh $(x_k = \frac{k}{n+1})_{k=0}^{n+1}$.

   Hints.

   - Simpon's rule is given by

   $$\int_a^b f(x)\, dx \approx \frac{b-a}{6} \Big(f(a) + 4\, f\big(\tfrac{b+a}{2}\big) + f(b) \Big).$$

   - The code for `right_hand_side()` can be greatly simplified by exploiting the property $\phi_k\big(\frac{\ell}{2\,(n+1)}\big) = \delta_{k\ell}$.
   - The last hint from Task 2 also applies here.

4. (unmarked) Once you have completed the above tasks, have a look at the functions `plot_solution()` and `convergence()`. Does their output match your expectations?