Special topics for C6.4 Finite Element Methods for PDEs

P. E. Farrell

February 25, 2017

1 Introduction

MSc students on the Mathematical Modelling and Scientific Computing course must complete a special topic if they wish to receive credit for this course.

The aim of the following special topics is for you to demonstrate your understanding of and competence with finite element discretisations of boundary value problems. An auxiliary aim is to have fun!

2 General advice

In all cases you should submit code (written in MATLAB or Python) for the solution of the problem at hand. This code should be structured into a generic module that solves a class of problems, and application-specific code that uses the generic code to solve a particular problem. For example, if one were to write a solver on a uniform mesh for

$$-u''(x) = f(x), \quad u(a) = 0 = u(b), a < b,$$

the generic code should take in as input a representation of $f(x)$, a and b, and a mesh parameter N; the application code should specify particular values for these parameters to exercise it.

The best way of verifying the correctness of numerical solvers is to test them on problems where the answer is known and check their order of convergence against our theoretical expectations. For example, if I wished to test the code above, I would choose a suitable $u(x)$, a and b (e.g. $u(x) = \sin(\pi x$,

$a = 0$, $b = 1$), and generate the $f(x)$ by plugging in the solution into the differential equation. This facilitates a convergence analysis: solve the problem for different refinements, and verify that the error scales as expected. In our running example, I would check that $\|u - u_h\|_{H^1(0,1)}$ halved each time I doubled the number of elements in the mesh.

In all cases use the standard MATLAB or Python facilities for sparse matrices, solving linear systems, and computing eigenvalues. I care only about your discretisation and assembly; leave the linear algebra and mesh generation (in two or three dimensions) to others. However, do not use any finite element toolboxes supplied by others: you can solve any of the problems below in twenty lines of FEniCS, but that doesn't mean you understand the finite element method!

The code should be clearly written and heavily commented; the comments are as important as the code itself. It should be accompanied by a report, between ten to twenty pages in length, explaining the problem, discretisation, known theoretical results, code, and examples used to exercise the code and verify its correctness.

3 Possible projects

Without further ado, here is a list of suggestions.

1. Write a finite element solver for the biharmonic equation

$$u''''(x) = f(x), \quad u(a) = u'(a) = 0 = u'(b) = u(b), a < b,$$

   using Hermite or Argyris finite elements. Verify your convergence against a known solution.

   This is probably the most straightforward option suggested (but can gain just as many marks as other projects).

   References:

   (a) P. G. Ciarlet (1978). *The Finite Element Method for Elliptic Problems*. Reprinted by SIAM in 2002. North-Holland

   (b) S. C. Brenner and L. R. Scott (2008). *The Mathematical Theory of Finite Element Methods*. Third edition. Vol. 15. Texts in Applied Mathematics. Springer-Verlag New York

2. Write a finite element solver for the Laplace equation

$$-\nabla^2 u = f \text{ in } \Omega,$$
$$u = 0 \text{ on } \partial\Omega,$$

for an arbitrary bounded polygonal domain $\Omega \subset \mathbb{R}^2$. You should use the simplest discretisation, piecewise linear finite elements. This will require some facility with meshing; I have provided a few references on freely available meshing software and quadrature below. Verify your solver against a known solution on a nonrectangular domain. You may find Lemma (3.1.10) of Brenner & Scott useful for constructing a suitable known solution.

References:

(a) J. R. Shewchuk (1996). "Triangle: engineering a 2D quality mesh generator and Delaunay triangulator". In: *Applied Computational Geometry: Towards Geometric Engineering*. Springer, pp. 203–222. DOI: 10.1007/BFb0014474

(b) P.-O. Persson and G. Strang (2004). "A simple mesh generator in MATLAB". in: *SIAM Review* 46.2, pp. 329–345. DOI: 10.1137/S0036144503429121

(c) R. Cools (2003). "An encyclopaedia of cubature formulas". In: *Journal of Complexity* 19.3, pp. 445–453. DOI: 10.1016/S0885-064X(03)00011-6

3. Write a finite element solver to compute the first ten (lowest) eigenvalues λ and corresponding eigenfunctions $u(x) \neq 0$ of the problem

$$-u''(x) + c(x)u(x) = \lambda u(x), \quad u(a) = 0 = u(b), a < b, c(x) \geq 0.$$

Verify your solver on the particular problem with $c(x) = 0$, $a = 0$, $b = \pi$. The eigenvalues are the squares of the integers: 1, 4, 9, ..., and the eigenvectors are of the form $u(x) = \sin(kx)$ for $k = 1, 2, 3, \ldots$.

References:

(a) D. Boffi (2010). "Finite element approximation of eigenvalue problems". In: *Acta Numerica* 19, pp. 1–120. DOI: 10.1017/S0962492910000012

4. Write a finite element solver for the Carrier equation

$$\varepsilon u''(x) + 2(1 - x^2)u(x) + u^2(x) = 1, \quad u(-1) = 0 = u(1),$$

for $\varepsilon = 0.01$. The equation is nonlinear due to the $u^2(x)$ term, and will therefore require the application of a Newton–Kantorovich iteration. For $\varepsilon = 0.01$, this equation supports eight distinct solutions; how many can you find?

References:

(a) C. M. Bender and S. A. Orszag (1999). *Advanced Mathematical Methods for Scientists and Engineers I: Asymptotic Methods and Perturbation Theory*. Springer. DOI: 10.1007/978-1-4757-3069-2, equation (9.7.7)

(b) S. J. Chapman and P. E. Farrell (2016). "Analysis of Carrier's problem." In: *SIAM Journal on Applied Mathematics*. arXiv:1609.08842 [math.CA]

5. Given an obstacle function $g(x) \in H^1(0,1)$ with $g(0), g(1) \leq 0$, define the closed convex set

$$K = \{v \in H_0^1(0,1) : v(x) \geq g(x) \text{ almost everywhere}\}.$$

Write a finite element solver for the minimisation problem

$$u = \operatorname*{argmin}_{v \in K} \int_0^1 (v'(x))^2 \, \mathrm{d}x.$$

The (necessary and sufficient) optimality condition for this system is the variational inequality

$$\text{find } u \in K \text{ such that } a(u, v - u) \geq 0 \text{ for all } v \in K,$$

where

$$a(u, v) = \int_0^1 u'(x)v'(x) \, \mathrm{d}x.$$

References:

(a) P. G. Ciarlet (1978). *The Finite Element Method for Elliptic Problems*. Reprinted by SIAM in 2002. North-Holland

(b) M. Ulbrich (2011). *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces*. Vol. 11. MOS-SIAM Series on Optimization. SIAM. ISBN: 978-1-61197-068-5. DOI: 10.1137/1.9781611970692

6. A topic of your choice.

   The theme here is that I want to go exactly one step beyond the Laplace equation in one dimension. That might mean posing the problem in higher dimensions (#1), considering a problem that requires elements beyond Lagrange (#2), an eigenvalue problem (#3), a nonlinear problem (#4), or a constrained convex minimisation problem (#5).

   Other ideas might be to consider an equation on an unbounded interval, or to consider a stochastic boundary value problem, or to consider a transient equation (a PDE involving space and time), or to consider the bifurcation analysis of an equation as a parameter is varied, or to implement an *a posteriori* error indicator and adaptive discretisation. I am open to discussion and keen for you to solve a problem you are enthused by!