`script_CQ_timestepping.m`

Using CQ for FEM time stepping

Prepared by: MEH and AH
Last modified: May 20, 2016

The goal of this script is to show how to use Convolution Quadrature for time stepping with finite elements. The model equation will be the wave equation in a bounded domain $\Omega \subset \mathbb{R}^3$. We will discretize in space with standard $\mathbb{P}_k$ FEM and use trapezoidal rule CQ for time stepping. The equations are

$$\begin{aligned} c^{-2}\ddot{u} &= \mathrm{div}\,(\kappa\nabla u) + f(x,y,z,y) && \text{in } \Omega\times[0,T],\\ \gamma u &= u_D && \text{in } \Omega\times[0,T],\\ \kappa\nabla u\cdot\mathbf{n} &= g_N && \text{in } \Omega\times[0,T], \end{aligned}$$

where u_D will be Dirichlet data, u_N will be the Neumann data, and $\mathbf{n}$ is the unit outward pointing normal vector. The script begins by assigning which elements on the boundary will be Dirichlet faces. We then generate a refinement of the unit cube for the geometry. We generate a plane wave transmitting the signal

$$\sin(2t)\chi(t)$$

where $\chi(t)$ is a smooth cutoff function in time so that the wave has compact support in time. The wave by default is moving in the direction $\mathbf{d} = (1,1,1)/\sqrt{3}$. We compute the spatial and temporal derivatives up to second order, which we will use later to generate a load vector. As a thorough test, we take the wave speed $c(x,y,z) = 1$ and the symmetric and uniformly positive definite matrix of material parameters to be

$$\kappa(x,y,z) = \left(\begin{array}{ccc} 1+0.5(x^2+y^2+z^2) & 0.25+0.5(x^2+y^2+z^2) & 2+(x^2+y^2+z^2)\\ 0.25+0.5(x^2+y^2+z^2) & 3+0.5(x^2+y^2+z^2) & 0.5+0.5(x^2+y^2+z^2)\\ 2+(x^2+y^2+z^2) & 0.5+0.5(x^2+y^2+z^2) & 6+0.5+0.5(x^2+y^2+z^2) \end{array}\right).$$

We also compute all of the spatial derivatives of first order of each element of the matrix to use in computing the load vector later on. With the values of $c(x,y,z)$ and $\kappa(x,y,z)$ set, we generate the FEM mass and stiffness matrices and denote them M_h and S_h, respectively. Note that although the wave equation we are interested in has the term $c^{-2}\ddot{u}$ on the left-hand-side, as written we are generating a mass matrix corresponding to the bilinear form

$$\Big(c(x,y,z)u^h(x,y,z,t), w^h(x,y,z)\Big)_\Omega.$$

If $c(x,y,z)$ is not taken to be identically one, we must be mindful of the solution we are using and how we are generating the mass matrix. We now generate the function handles `qx`, `qy`, `qz` depending on space and time, which correspond to the elements of the vector

$\kappa(x,y,z)\nabla u(x,y,z,t)$. The last piece of derived data we generate is a right-hand-side $f(x,y,z,t) := \ddot{u}(x,y,z,t) - \mathrm{div}(\kappa\nabla u(x,y,z,t))$ so that the given plane wave is a solution to the wave equation with the given wave speed and material properties. With all of these pieces, we loop over all times to generate the Dirichlet and Neumann boundary conditions, as well as the load vector at all times.

The key point we need now is the function handle `FEMsolve(s,v,rhs)`. This handle takes a complex frequency s, a FEM vector v and a right hand side (corresponding to the load vector and Neumann boundary conditions) and solves for the free (non-Dirichlet) degrees of freedom and places them in v. With all of these pieces in hand, we can pass the data to the CQ routine (which we have hard-coded into the script for simplicity). After the CQ routine to discretize the temporal component of the PDE, we compute the $L^2(\Omega)$ and $H^1(\Omega)$ errors, and either print them or store them, depending on the mode the script is being run in.

There are two different modes to run this script. If a variable `external` HAS NOT been defined, the script requests the user to input: the polynomial degree, the refinement level of the geometry, the final time, and the number of time steps. If the variable `external` has been defined, these data need to have been declared in advance. In this case, the errors are appended as an additional row to a matrix of errors.