`script_CQ_timestepping_elasticity.m`

Using CQ for FEM time stepping for the elastic wave equation

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 when discretizing the elastic wave equation The model will be the elastic 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

$$\begin{aligned} \rho\ddot{\mathbf{u}} &= \operatorname{div}\sigma(\mathbf{u}) + \mathbf{f}(x,y,z,t) && \text{in } \Omega \times [0,T], \\ \gamma\mathbf{u} &= \mathbf{u}_D && \text{on } \Gamma \times [0,T], \\ \sigma(\mathbf{u})\mathbf{n} &= \mathbf{u}_N && \text{on } \Gamma \times [0,T], \end{aligned}$$

where $\mathbf{u}_D$ will be Dirichlet (displacement) data, $\mathbf{u}_N$ will be the prescribed normal stress, $\mathbf{n}$ is the unit outward pointing normal vector, and $\sigma(\mathbf{u}) := \mu\varepsilon(\mathbf{u}) + \lambda(\nabla \cdot \mathbf{u})\mathbf{I}$. 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 pressure 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 $\mathbf{d} = (1,1,1)/\sqrt{3}$ with wave speed $c = 1$ and amplitude $a = 1$. 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 variable coefficients

$$\begin{aligned} \lambda(x,y,z) &= 5 + (x^2 + y^2 + z^2) \\ \mu(x,y,z) &= 2 + (x^2 + y^2 + z^2) \\ \rho(x,y,z) &= 3 + (x^2 + y^2 + z^2). \end{aligned}$$

We then compute the spatial derivatives of $\lambda(x,y,z)$ and $\mu(x,y,z)$ to use to compute a load vector so that the solution is the prescribed pressure wave. A final computation is the divergence of the three components of the solution, the values of the stress tensor σ, and the components of the load vector so that the given wave is a solution to the PDE. We generate the mass and stiffness matrices, and then loop over all times to generate discretizations of the load vector, normal stresses, and displacement boundary conditions.

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.