Problem Set 8

SEM solution to 1D wave equation

Use the Spectral-Element method (SEM) to solve the 1D wave equation to find the displacement $s(x,t)$ for $x \in [0, L = 100]$ such that (strong form)

$$\rho\, \partial_t^2 s = \partial_x(\mu\, \partial_x s)$$

where ρ is the medium density and μ is the shear modulus, with the following initial & boundary conditions:

$$(a)\ \ Dirichlet \qquad \begin{cases} s(x,0) &= f(x) \\ s(L,t) &= 0 \\ s(0,t) &= 0 \end{cases}$$

and

$$(b)\ \ Neumann \qquad \begin{cases} s(x,0) &= f(x) \\ \partial_x s(L,t) &= 0 \\ \partial_x s(0,t) &= 0 \end{cases}$$

Problem:

Follow these steps to solve the problems (a) and (b):

- write the weak form of the wave equation for the test function $w(x)$
- discretize the mesh: $\Omega = [0, L] = \bigcup_e \Omega_e$
- on the elemental level, calculate the mass and stiffness matrices
- impose the boundary conditions for (a) and (b)
- consider the initial condition with $f(x) = \exp[-(x - 50)^2 * 0.1]$ and media properties $\rho = 1$ and $\mu = 1$

Plot several time steps.

Time scheme:

You will be using the Newmark algorithm seen in class to march in time:

- Predictor:

$$\begin{aligned} d_{n+1} &= d_n + \Delta t v_n + \frac{1}{2}\Delta t^2 a_n \\ v_{n+1} &= v_n + \frac{1}{2}\Delta t a_n \\ a_{n+1} &= 0 \quad \text{(initialization at the beginning of each time step)} \end{aligned}$$

- Solve:

$$\begin{aligned} F_{n+1} &= K\, d_{n+1} \\ \Delta a &= M^{-1}\, F_{n+1} \end{aligned}$$

- Corrector:

$$\begin{aligned} a_{n+1} &= a_{n+1} + \Delta a \\ v_{n+1} &= v_{n+1} + \frac{1}{2} a_{n+1} \\ d_{n+1} &= d_{n+1} \end{aligned}$$

Material:

In the directory codes/ you will find the following files:

`Makefile`: used for the compilation. By default it uses gfortran.

directory `obj/`, `bin/`: where you will find the ouputs of the compilation.

`constants.h`: a numbers of static constants, like the number of spectral elements (NSPEC=11), number of GLL points (NGLL=7), number of global points (`NGLOB = (NGLL-1)*NSPEC +1`), and two variables used to evaluate the GLL points & the weights.

`gll_library.f90`: library to compute the GLL points and the weights.

`lagrange_poly.f90`: library to compute the Lagrange interpolants based upon the GLL points as well as the first derivatives of these polynomials at any point $\xi \in [-1, 1]$.

`define_derivative_matrix.f90`: store the derivatives of the Lagrange polynomials as a NGLL$\times$NGLL matrix.

`wave.f90`: the main program.

After compilation you will find the executable *xwave* in the directory `bin/`.

You will see that the full structure of the code is given in the file `wave.f90`. You are asked to input your code in lieu of "`>TODO: put your code here`".