

INF5620

Project 1

Sebastian Gjertsen

October 10, 2015

In this project we use the linear, two-dimensional wave equation:

$$\frac{\partial^2 u}{\partial t^2} + b \frac{\partial u}{\partial t} = \frac{\partial}{\partial x}(q(x, y)u(x, y, t)) + \frac{\partial}{\partial y}(q(x, y)u(x, y, t))$$

Our general scheme for internal points is:

$$u_{i,j}^{n+1} = \frac{\frac{dt*b}{2} - 1}{\frac{dt*b}{2} + 1} u_{i,j}^{n-1} + \frac{2}{1 + K} u_{i,j}^n +$$
$$0.5 \left(\frac{\frac{dt^2}{dx^2}}{\frac{dt*b}{2} + 1} (q_{i+1,j} + q_{i,j})(u_{i+1,j}^n - u_i^n) - (q_{i,j} + q_{i-1,j})(u_{i,j}^n - u_{i+1,j}^n) \right) +$$
$$0.5 \left(\frac{\frac{dt^2}{dy^2}}{\frac{dt*b}{2} + 1} (q_{i,j+1} + q_j)(u_{i,(j+1)}^n - u_{i,j}^n) - (q_{i,j} + q_{i,j-1})(u_{i,j}^n - u_{i,j+1}^n) \right) + f_{i,j}^n$$

I have used ghost so i only need to update the ghost point at each edge with the line in from the edge.

3

3.4

Here I have inserted the exact solution into the pde, to get $\omega = \sqrt{q * (k_x^2 + k_y^2)}$

In the gif standing wave, I have plotted the numerical and exact solution on top of eachother, and as we can see there is no difference so this tells me that the solver is accurate.

See the following command :

```
[fenics@ubuntu64-box] ../INF5620/Wave-project > python Project_1.py
Error is 1.8090169943749476733074744 with dt: 0.100000, dx:1.0000, dy:1.0000
Error is 1.9707532449271689678482744 with dt: 0.050000, dx:0.5000, dy:0.5000
Error is 1.0629860660020418183790980 with dt: 0.025000, dx:0.2500, dy:0.2500
Error is 0.2728955717293958249669572 with dt: 0.012500, dx:0.1250, dy:0.1250
Error is 0.0681798355948715212315747 with dt: 0.006250, dx:0.0625, dy:0.0625
Error is 0.0170306051544297926136906 with dt: 0.003125, dx:0.0312, dy:0.0312
```

Error is 0.0042569491001733239521343 with dt: 0.001563, dx:0.0156 , dy:0.0156
The convergence rate: ..
[-0.12354118892171646, 0.89062446436077825,
1.9617017963021701, 2.0009319584616945,
2.0012154209330246, 2.000237954508191]
As we can see the convergence tends toward 2 which again assures me that the so

4

In this section I have modelled different ocean floors and look at how it effects the wave. I have used more or less the same wave for every run. The first one is a simulation of a beach, found in *beach₁.gif* The red is the seafloor and the blue is the wave. We see that it acts the way we would expect a wave to hit a beach. It increases in amplitude as it hits the beach and gets pulled down a bit as it moves out. The next simulation was of a wide ridge in the middle of the domain, *ridge₁.gif*. We see that the wave gets pulled down a bit from ocean floor, as expected. Now we see in *ridge₂.gif*, that when we increase the height of the ridge and increase the amplitude we get some errors in the solution. Its not too bad, but it indicates at what capacity our solver starts making errors. In the *peak_smooth.gif* with a little dampning, the effects look natural and we get a pull from the geometry on the wave when it moves over. With the *peak_crash.gif* we see a that if we choose a peak thats too high the solver crashes. In the *rough_mesh.gif*, and *super_rough_mesh.gif* we see that the solver delivers a good solution even with a coarse mesh.