Homework 2

Computational Math

Author:
Mikola Lysenko

March 11, 2010

1

a I checked the exactness condition for the given u using sympy and a Python script:

```
from sympy import *;

def uexact(x):
return exp(sin(x)) * exp(cos(x));

def q1(x):
return sin(x) * cos(x);

def q2(x):
return -2 * cos(x)**4 + cos(x)**3 + (8 + 3 * sin(x)) * cos(x)**2 - (1 + sin(x)) * cos(x);

u = uexact(x);
ux = diff(u, x);
uxx = diff(ux, x);
uxxx = diff(uxx, x);
print trigsimp(simplify(uxxx + q1(x) * uxx + q2(x) * u(x) - 3 * exp(sin(x)) * exp(cos(x)))) == 0;
```

By inspection, it should be clear that if uexact is an exact solution for the ode, then it ought to print out True; which is exactly what happens, and so the exactness condition is satisified.

b Based on class discussion, pick

$$u(x) = \sum_{j=0}^{N} v_j \frac{\sin(\pi(x - x_j)/h)}{2\pi/h \tan((x - x_j)/2)}$$

c

2 Solve using chebfft.m + RK-4 integrator

3

a Assume that $\sigma(x) > 0$ for all $x \in [0, 1]$, (otherwise the solution could be potentially imaginary or undetermined, and so it would be impossible to directly apply Sturm-Liouville theory). From the boundary condition, we know that $\varphi(1) = 0$. Combined with the asymptotic solution for the eigenmodes, we get the following equation for λ:

$$0 = \sigma(1)^{-\frac{1}{4}} \sin\left(\sqrt{\lambda} \int_0^1 \sqrt{\sigma(\xi)} d\xi\right)$$

Which is true if and only if $\sin(...) = 0$, and so it must be that $... = 0, \pi, 2\pi, ...k\pi$ for all $k \in \mathbb{Z}$. Thus:

$$\sqrt{\lambda} \int_0^1 \sqrt{\sigma(\xi)} d\xi = k\pi$$

And so

$$\lambda = \left(\frac{k\pi}{\int_0^1 \sqrt{\sigma(\xi)} d\xi} \right)^2$$

b Once again, use chebfft.m + Rk-4 integrator. Can recycle code from prog 2

4

a Just run the MATLAB script, add the necessary lines

b Should be self-explanatory

5 The forward Euler method is stable for values of $\Delta t \approx |\lambda|^{-1}$ where λ is the largest eigenvalue of the system. In this case, since we are dealing with a second-order dispersive system, the largest eigenvalue should be in $O(N^2)$ where N is the number of samples. So to ensure stability, the timestep should be in $\Delta t \in O(N^{-2})$.

For second part, just use formulas from text, plug into matlab