HOMEWORK 5: DFT and spectral methods
for Mathmatical Models, Analysis and Simulation, Fall 2012
Report due Mon Dec 3, 2012.
Maximum score 6.0 pts.

Read Strang's book, Sections 4.1-4.3 and 5.5, as well as the handout by Canuto et al.

1. Selected theoretical problems (Score: 1.0)

a) Let $u(x,t) = \sum_{n=1}^{\infty} a_n(t)\sin(nx)$ on the interval $x \in [0,\pi]$.
If $\frac{\partial u}{\partial x} = \sum_{n=1}^{\infty} a_n^{(1)}(t)\sin(nx)$, then use the previous result to show that

$$a_n^{(1)}(t) = \frac{4}{\pi} \sum_{\substack{\text{m=1} \\ \text{m+n odd}}}^{\infty} \frac{nm}{n^2 - m^2} a_m(t)$$

b) Given the equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = d\frac{\partial^3 u}{\partial x^3},$$

defined on the interval $x \in [0,L]$ with periodic boundary conditions $u(0,t) = u(L,t)$, where c and d are constants. Let

$$u^N(x,t) = \sum_{k=-N/2}^{N/2} \hat{u}_k(t) e^{2\pi ikx/L}$$

be a spectral expansion of u.

Derive the Galerkin equations for the expansion coefficients $\hat{u}_k(t)$.

2. DFT and frequency content of functions (Score 0.2)

Consider a $1-$periodic function f. Introduce the grid $x_j = j/N,\ j = 0,1,2,\dots,N-1$, and denote $f_j = f(x_j)$. Define the DFT coefficients $\hat{f}_k$ by

$$\hat{f}_k = \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-2\pi ikx_j/L}, \quad k = 0,\dots,N-1.$$

Note the different $1/N$ scaling as compared to Matlabs `fft`.

For a few different functions $f(x)$, use Matlabs FFT and examine the real and imaginary part of the DFT coefficients.

i) $f(x) = 1$, $f(x) = \sin(2\pi x)$ and $f(x) = \cos(2\pi x)$.

ii) $f(x) = \sin(4\pi x) + \cos(6\pi x)$

iii) $f(x) = \sin^3(2\pi x)$. What trigonometric formula can you deduce from the computed DFT coefficients?

iv) $f(x) = \sin^2(2\pi x)\cos^2(2\pi x)$. What trigonometric formula can you deduce from the computed DFT coefficients?

3. Spectral interpolation and differentiation (Score 1.8)

Consider a $L-$periodic function f. Introduce the grid $x_j = jL/N,\ j = 0,1,2,\ldots,N-1$, and denote $f_j = f(x_j)$. Define the DFT coefficients $\hat{f}_k$ by

$$\hat{f}_k = \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-2\pi i k x_j/L}, \quad k = 0,\ldots,N-1 \tag{1}$$

Define the spectral interpolant Πf by

$$\Pi f(x) = \sum_{k=0}^{N-1} \hat{f}_k e^{2\pi i k x/L}, \quad 0 \le x < L.$$

Hence, the interpolant is defined for all x in the interval, and coincides with the discrete values f_j at the points x_j, i.e.

$$\Pi f(x_j) = f_j, \quad j = 0,\ldots,N-1.$$

The definition of the DFT coefficients above is in accordance with Strang. Note the different $1/N$ scaling a compared to Matlabs `fft`.

Alternatively, define the DFT coefficients $\hat{f}_k$ for a different range of k,

$$\hat{f}_k = \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-2\pi i k x_j/L}, \quad k = -N/2,\ldots,N/2-1, \tag{2}$$

and define the spectral interpolant Πf by

$$\Pi f(x) = \sum_{k=-N/2}^{N/2-1} \hat{f}_k e^{2\pi i k x/L}, \quad 0 \le x < L.$$

To achieve this set of DFT coefficients, look up what `fftshift` does.

a) Consider

$$f(x) = e^{-M(x/L-0.3)^2} \quad (*)$$

on $[0,L]$ with periodic extension $f(x) = f(x+L)$ for all x. Take $N = 2^m$ with $m = 8$. Compute the DFT coefficients. To study the decay of the Fourier coefficients, plot $|\hat{f}_k|$ vs k for $k = 0,\ldots,N/2-1$ with log scale on the y axis. (Use `semilogy`). Do this for different values of M, say $M = 10, 100, 200$ and 400 in the same plot. Explain the result!

b) For $M = 100$ in (*), compute the DFT coefficients $\hat{f}_k$ for $N = 2^4$. Then evaluate Πf on a much denser grid, say with 400 points, and plot the data and the interpolant vs. x. Note: Evaluate the interpolant and then take the real part, before you plot.

Do this both with the Fourier coefficients as defined in (1) and in (2), and evaluate the respective interpolant. Plot them both. There is a huge difference in quality of the interpolant. Comment on this result.

c) From now on, we will use the centered formula (2). Compute an approximation to $f'(x_k)$ and $f''(x_k)$ by computing with `fft`, whereafter differentiation is carried out - this is simply now a product to be computed for each k. Then use `ifft` to transform back. Make sure to use `fftshift` as needed. Denote this approximation to the derivatives at x_j, $j = 0, \ldots, N-1$ by Df_j for the first derivative and $D^2 f_j$ for the second derivative.

Using the discrete 2-norm,

$$\|g\|_2 = \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} |g_j|^2}, \tag{3}$$

we measure the error in the derivatives computed using the FFT.

First, for the approximation of the first derivatives, plot the error in computing the first derivative versus N, i.e $\|Df - f'\|_2$. (Use e.g. $N = 2^m$, with $m = 4, 5, 6, 7, 8$). Do this in a log-log plot, for $M = 100$, 200, 400 and 800.

For an infinitely smooth function, one would expect to see a flatter error curve for small values of N, where the function is under resolved. Then one should enter the "exponential range", where errors decay exponentially. Then at some point, the errors flatten out due to round off errors.

In this case, you get different results for different values of M. Explain what you see!

Redo the same for the second derivatives, now plotting $\|D^2 f - f''\|_2$ vs N for different values of M. Compare to the results for the first derivative. Comment.

Note: What is meant by exponentially? In a log-log plot, plot both e^{-qN} vs N for some q and then N^{-q} vs N. What do the slopes look like?

4. Spectral methods for differential equations (Score: 3.0)

Consider the model equation

$$u_t + A(u^2)_x + Bu_x = \varepsilon_2 u_{xx} + \varepsilon_3 u_{xxx}, \quad 0 \le x < L, \quad t \ge 0, \quad u(x,0) = f(x) \tag{**}$$

with $\varepsilon_2 \ge 0$, f and u L-periodic, and $A, B, \varepsilon_1, \varepsilon_2$ constant.

a) See Strang Sections 6.4 and 6.6 and give the values of $A, B, \varepsilon_2, \varepsilon_3$ which give the heat, convection-diffusion, Schrödinger, Burger's, and Korteweg-deVries equations.

Introduce the expansion:

$$u^N(x) = \sum_{k=-N/2}^{N/2-1} \hat{u}_k(t) e^{2\pi ikx/L}.$$

Using a collocation approach, and introducing a vector notation;
$U(t) = (u^N(x_0,t), u^N(x_1,t), \ldots, u^N(x_{N-1},t))$, where $x_j = jL/N$,
the discrete problem can be given as

$$\frac{\partial U}{\partial t} + AD_N(U \cdot U) + BD_N U = \varepsilon_2 D_N^2 U + \varepsilon_3 D_N^3 U$$

where the $\cdot$ means pointwise multiplication and D_N is the matrix that represents the Fourier collocation differentiation. See the lecture notes.

Also, expand in a centered Fourier series and write down the ODEs for the Fourier coefficients, and consider a pseudo-spectral treatment of the nonlinear term. (This is the pseudo-spectral Galerkin method).

Comment on the difference between the collocation method and the pseudo-spectral Galerkin method.

b) Your task is to implement a high-order method (fourth order in time, exponential order in space) for numerical solution of the initial-boundary value problem for this family of models.

Do this using the collocation method above. If you prefer, you can rewrite the non-linear term using $(u^2)_x = 2uu_x$, which yields a slightly different discretization. Use the classical fourth order Runge-Kutta scheme for the time-stepping. The differentations should be implemented using `fft`, `ifft` and `fftshift`.

c) **The heat equation**
Consider the heat equation. Take $L = 1$ and $\varepsilon_2 = 1$. Choose the initial data to be $f(x) = \sin(2\pi x)$, such that the exact solution is known. Solve until time $T = 0.1$ and $T = 1.0$ for $N = 2^m$ with $m = 3,4,5,6,7$.

i) What is the time-step limit Δt_{max} for stability? The RK4 stability region on the real line is approx. $[-2.8, 0]$. Answer this question on the form $\Delta t_{max} \leq C/N^\alpha$ with C and α given. You can find a theoretical estimate by considering the ODEs for the Fourier coefficients. For the values of N given above, check experimentally what limit you get and compare to the theoretical estimate.

ii) For each m, i.e. each value of N, choose a sufficiently small Δt that the error is dominated by the spatial error. Use the discrete 2-norm as defined in (3) and measure the error compared to the exact solution. (Sufficient to do this at $t = 0.1$). Plot the errors versus N in a log-log diagram. Do the errors decay exponentially?

iii) Explain how you in this case can find the exact solutions to the ODEs for the Fourier coefficients that you wrote down in a) by using an integrating factor. Give the solutions.

d) **Burger's equation**
Again, let $L = 1$ and take the initial condition $f(x) = \sin(2\pi x)$. If we set $\varepsilon_2 = 0$, the solution gets sharper and sharper, and develops discontinuities after finite time, even if f is smooth. For $\varepsilon_2 > 0$, the solution is smooth at all times. In the experiments below, use $N = 2^m$ with $m = 4, 5, 6, 7$.

i) First, run with $\varepsilon_2 = 0$. You will observe oscillations around the front when it becomes steep enough. At what time does the solution break?

ii) Now, let $\varepsilon_2 = 0.02$. Run until $T = 0.2$. When an exact solution is not known, we can use consecutive refinements to estimate the errors. Hence, compute $\|u^N - u^{2N}\|_2$ for the solutions that you have, and plot versus N. Comment on the results. (Note: remember, there are both spatial and temporal errors. How did you pick your Δt:s?)

ii) Now, let us turn to the pseudo-spectral Galerkin approach. Implement a time-stepping for the Fourier coefficients using an integrating factor technique, combined with a forward Euler method. (See lecture notes). Certainly, forward Euler is low order in time, and could be replaced. But let us focus on the time-step restrictions for now. Would you expect a different scaling of Δt_{max} with N? With $\varepsilon_2 = 0.05$, do experiments and compare the time step limits of your two implementations for different N. Explain.