Homework #2, due April 7

Objectives: (1) Get familiar with basics of continuous and discrete Fourier Transform. (2) Explore ill-conditioned matrices. (3) Explore tridiagonal solver.

(1) On page 4, equation (3) of the lecture note, the expression $\hat{f}_k$, $\hat{f}^e$ and $\hat{f}^o$ might be a little confusing. A better notation should contain both number of grids N (missing in all 3 of them) and the wave number k (missing in $\hat{f}^e$ and $\hat{f}^o$). Write a proper expression for all 3 of them. Then explain why 5 appears in the recursion formula $J_N = 2J_{N/2} + 5N$.

(2) Let $\hat{f}_k$ denote the properly normalized discrete Fourier transform of a *periodic* discrete function f_j. Prove the following identities.

(a) (conformality)

$$\sum_k \hat{f}_k \bar{\hat{g}}_k = \sum_j f_j \bar{g}_j$$

Therefore $\|f\|_2 = \|\hat{f}\|_2$ (Plancheral's identity). (Hint: basis of l_2)

(b) Show that $\widehat{fg} = \hat{f} * \hat{g}$.

(c) Show that $\check{\hat{f}} = f$ and $\hat{\check{g}} = g$

Remark: The corresponding statements are also true for DCT, DST and higher dimensions.

(3) Let $\hat{f}(\xi)$, $\xi \in R$ denote the properly normalized continuous Fourier transform of a function $f(x)$, $x \in R$. and $G(x)$ the Gaussian distribution. Show that the Fourier transform of a Gaussian distortion is still a Gaussian distribution, but with inverse the variance. That is, $\widehat{G(\frac{x}{\epsilon})}(\xi) = G(\epsilon\xi)$. This example is to demonstrate the following statement, which is more or less true: 'The thicker f is, the thinner $\hat{f}$ and vice versa'.

(4) Explore the efficiency in FFT.

Find the number of operations, $c(n)$, for `fft` in Matlab for vectors of length $n = 1, 2, \cdots$ (Use the `flops` command). Plot, on a log-log scale, $n^2/c(n)$ and $(n \log n)/c(n)$. For which n does `fft` behave like a "slow" Fourier transformation? For which values of n besides powers of 2 does `fft` behavior like "fast transformation"?

(5) (Optional) The free space Poisson/Helmholtz equation.

Consider the following whole space Helmholtz equation in 1D:

$$\begin{aligned} &-u'' + u = \exp(-x^2/2), \quad x \in R \\ &u(\pm\infty) = 0 \end{aligned} \tag{1}$$

The analytic expression of the solution should be possible in terms of the error function $\mathrm{erf}(x)$. Several numerical approaches are available.

(a) Set homogeneous Dirichlet or Neumann boundary data at $|x|$ sufficiently large. Then proceed with FFT.

(b) Direct numerical quadrature using fast convolution with FFT. In general, this approach will not work in R^2 or R^3.

(c) Direct numerical quadrature to get the Dirichlet or Neumann boundary data at $|x|$ sufficiently large (you don't need fast convolution in 1D case), then proceed with FFT.

(d) Impose the 'Dirichlet to Neumann' boundary condition at $|x|$ sufficiently large,

$$u'(M) = -u(M), \qquad u'(-M) = u(-M)$$

then solve the resulting tridiagonal matrix problem using standard package or Gauss elimination. (Remark: This will not work in a 2D square or a 3D cube, but a 2D disk and a 3D ball instead.)

Try to solve the ODE numerically with one or more approaches from (b), (c), (d) and compare it with (a).

Remark: If you want more accuracy for the discretization part, the ODE $v'' = g$ can be done more efficiently using the 4th order compact scheme:

$$D^2 v = g + \frac{h^2}{12} D^2 g + O(h^4)$$

which involves only 3 neighboring stencils.

(6) (Optional) Explore an ill conditioned matrix.

For this assignment you will need software that estimates the condition number of a matrix. This can be MATLAB, or a standard scientific subroutine package (e.g. NAG) or software from netlib. If you don't have access to such software, you will have to skip part **(a)**. If you don't find a routine that estimates condition number but you can find one that computes the singular values or the singular value decomposition, the condition number is $\sigma_{\max}/\sigma_{\min}$.

For a fixed n, define A to be the $n \times n$ matrix with all entries zero except the $a_{ii} = 1$ and $a_{i,i+1} = 2$ for all i. That is, A has 1 on the diagonal and 2 immediately above the diagonal and all the rest zero. The determinant of this matrix is 1 and its norm is about 3 (depending on exactly which norm is used).

(a) Find the (estimated) condition number of A as a function of n for $n = 2, 3, \ldots$ until the condition number becomes (in the computer) infinite.

(b) Let $x \in R^n$ be the vector with n components whose components are given by $x_j = 1$ if j is even and $x_j = 1/3$ if j is odd. The value $1/3$ is not stored exactly in the computer in floating point. Compute $b = Ax$ by simple matrix multiplication. Now compute y by solving the equations $Ay = b$ where b has just been computed. Write a program to do this in MATLAB, FORTRAN, C, or C++ using back substitution on an upper triangular matrix that is bidiagonal (see Kahaner, Moler, and Nash). Compute $\|x - y\|$ for some norm and plot this as a function of n so long as possible. Compute $|Ay - b|$ for the y you have and check its dependence on n. Comment on the contrast between this and the dependence of $\|x - y\|$ on n.

(c) Is the error $\|x - y\|$ of the order predicted by the size of roundoff error (in single or double precision) and the condition number of A? Turn in a table or semi-log plot showing the error $y - x$, the residue $Ay - b$, and $|y - x|/\text{cond}$.

(d) How would you invert a matrix with condition number around 10^{20}? This probably can not be done in matlab. If you do not have access to C or Fortran, skip this one. For Intel Fortran Compile, this can be done with a -r16 compiler option. Verify that the errors are as predicted.

(7) (Optional) Follow up of homework # 1 (5)

Suppose, or pretend, that you do not have the regularity information of the integrand and you want to distribute the grid in such a way that the graph $(x, f(x))$ is cut into pieces by $(x(y_j), f(x(y_j)))$ of roughly equal arclength by the mapping $x = x(y)$ with $y \in [0,1]$ equally spaced.

(a) Show that the choice of the weight $w(x) = \sqrt{1 + (\frac{Df}{Dx})^2}$ in the following nonlinear equation will do the job.

$$D\,(w(x)Dx) = 0 \tag{2}$$

Here D is the difference quotient $(Du)_{i+1/2} = (u_{i+1} - u_i)/h$, $(Dv)_i = (v_{i+1/2} - v_{i-1/2})/h$. In $w(x)$,

$$(\frac{Df}{Dx})_{i+1/2} = \frac{f(x(y_{i+1})) - f(x(y_i))}{x(y_{i+1}) - x(y_i)}$$

and $y_j = j/N$.

(b) Since the function $x(y)$ is unknown, the equation (2) has to be solved iteratively

$$D\left(w(x^{(k)})Dx^{(k+1)}\right) = 0 \tag{3}$$

This tridiagonal matrix problem can be solved efficiently using Gauss elimination.

(c) Evaluate the error in each iteration $e^{(k)} = \text{Simpson}(f, x^{(k)}) - \pi/4$ and see how it works. Be sure to redistribute the grid lengths to be pairwisely equal.

(d) Try this method with other integrands that has a different type of singularity.