Assignment # 7

Due 3 April, 2013

Please read Lectures 8, 10, 11, 12, and 13 in the textbook *Numerical Linear Algebra* by Trefethen and Bau. Do these exercises:

Exercise 10.2 in Lecture 10.

Exercise 11.3 in Lecture 11.

Exercise 12.3 in Lecture 12.

P14. Recall the very basic equation of QR factorization, namely (7.6) which can be written

$$q_j = \frac{1}{r_{jj}} a_j - \frac{r_{1j}}{r_{jj}} q_1 - \cdots - \frac{r_{j-1,j}}{r_{jj}} q_{j-1}$$

Recall also that the stages of classical Gram-Schmidt (= Algorithm 7.1) are

$$A = \left[\begin{array}{c|c|c|c|c} a_1 & a_2 & a_3 & \dots & a_n \end{array} \right] \to \left[\begin{array}{c|c|c|c|c} q_1 & a_2 & a_3 & \dots & a_n \end{array} \right] \to \left[\begin{array}{c|c|c|c|c} q_1 & q_2 & a_3 & \dots & a_n \end{array} \right]$$

$$\to \quad \dots \quad \to \left[\begin{array}{c|c|c|c|c} q_1 & q_2 & q_3 & \dots & q_n \end{array} \right] = \hat{Q}$$

With these hints, and comparing page 61 which describes the corresponding upper-triangular matrices for *modified* Gram-Schmidt, carefully describe the upper-triangular matrices $\bar{R}_1, \bar{R}_2, \dots, \bar{R}_n$ so that classical Gram-Schmidt is the triangular orthogonalization procedure

$$A\bar{R}_1\bar{R}_2 \dots \bar{R}_n = \hat{Q}.$$

(*Pleas write $\bar{R}_j$ in terms of r_{ij}. You do not need to explain how r_{ij} is computed.*) Specifically give $\bar{R}_4$ in the case $A \in \mathbb{C}^{5\times 5}$. Compare to R_4 for the modified Gram-Schmidt process.

P15. *It is time to "reveal", if you don't already know it, the most important single source of large matrix problems. The kind of differential equation problem here illustrates a general principle that if a linear system is very large then: (1) it came from a linear system which is actually infinite and (2) the user wanted it to be infinite but in putting it on the computer it became finite and as big as the computer could handle.*

Consider the temperature $u(x, t)$ of a uniform solid rod of length L. If the thermal conductivity k, linear density ρ, and heat capacity c are all constant then the temperature might evolve in time by the equation

$$\rho c\, u_t = k\, u_{xx} + f(x),$$

where $f(x)$ represents a time-independent (but not constant) source of heat. Here the subscripts denote partial derivatives: $u_t = \partial u/\partial t$, etc. Let's suppose we hold the ends of the rod at a common fixed temperature $u = 0$. Then there is a steady heat distribution $U(x)$ which the time-dependent function $u(x,t)$ will approach,

$$U(x) = \lim_{t\to\infty} u(x,t).$$

This limit is approached regardless of the initial distribution of temperature. The steady state $U(x)$ solves the differential equation "boundary value problem"

$$0 = k\,U'' + f(x), \qquad U(0) = 0, \quad U(L) = 0. \tag{1}$$

(a) One may approximately solve problem (1) by the following "finite difference" technique. Choose N and an equally-spaced grid of points $x_0, x_1, \dots, x_N$ given by $x_j = j\Delta x$ where $\Delta x = L/N$. Denote the approximations to the actual unknown values $U(x_j)$ by U_j. The list $U_1, \dots, U_{N-1}$ gives the unknowns of the problem because $U_0 = U(x_0) = U(0) = 0$ and similarly $U_N = U(L) = 0$. One replaces the differential equation (1) by the approximation[1]

$$0 = k\frac{U_{j+1} - 2U_j + U_{j-1}}{\Delta x^2} + f(x_j) \qquad \text{for} \qquad j = 1, 2, \dots, N-1. \tag{2}$$

Incorporating the already noted boundary values $U_0 = 0$ and $U_N = 0$, write equations (2) as a linear system $Av = b$ in the $N = 5$ case. (*Of course, in this linear system* $v = [U_1\,U_2\,U_3\,U_4]^* \in \mathbb{C}^4$. *Don't worry here that* $f(x)$ *is not specified.*)

(b) Fix $L = 3$, $k = 1$, and $f(x) = \exp(-20(x-2)^2)$ for this and the next part.
Write a MATLAB function

```
function [A,b] = assemble(N)
```

which produces $A \in \mathbb{C}^{m\times m}$ and $b \in \mathbb{C}^m$ where $m = N - 1$. In the $N = 5, 10, 20, 50, 500$ cases, using `assemble` at the command line or in a script, solve the linear systems using backslash ("\") and display the approximate solutions as functions of $x \in [0,3]$. (*Show them all on a single plot with* x *versus* U *axes.*) Is this plot evidence of good approximation to the exact solution $U(x)$? Estimate the maximum of the exact solution $U(x)$ on $[0,3]$.

(c) Compute A from `assemble(100)`; *do not* waste paper to show me the entries of A. Use MATLAB to find the five eigenvectors of A which have eigenvalues closest to zero (i.e. "lowest frequency"). Normalize these vectors so their maximum entries have absolute value one. Plot the resulting vectors as functions of $x \in [0,3]$. Guess formulas for these functions. (*Hint: These are the numerical versions of the Fourier modes one would find by hand for* (1) *if one expanded* $U(x)$ *in a Fourier series which respected the boundary values. Your guessed functions* $V(x)$ *should have the property that* V'' *is a multiple of* V.)

[1] If you have never seen this before, a key fact is to notice that $U_{j+1} - 2U_j + U_{j-1} = (U_{j+1} - U_j) - (U_j - U_{j-1})$, a difference of differences, and thus that the fraction in (2) is an approximation of $d^2U/dx^2 = U''$.