# MATH 128B: WORKSHEET 1

This worksheet is intended to give you a chance to think about and talk about (I) some basic Matlab tools and (II) some ideas in approximation theory. It will not be collected or graded.

Part I: Make Friends with Matlab
You don't need to everything described below- just try these commands if you don't know them.
*If you know the name of a command you want to use but are not sure how, type 'help [name of command here]' in the Matlab console and an explanation and examples will appear. If you don't know the name of the command for what you want to do, type a description into Google.*

1. (Making arrays and matrices)
   - Make a grid of $50$ evenly spaced points on [1,2] by typing 'x = linspace(1,2,50)' . You can also use 'x = 1:1/50:2'.
   - Make a 2D grid with '[X,Y] = meshgrid(x)'. Look at the matrices X and Y. Define a function 'z = sin(pi*X).*cos(3*pi*Y)'. Now type 'surf(X,Y,z); colorbar' to see your function. Notice that we use '.*' instead of '*' in between the matrices because we are multiplying them element-by-element, not matrix multiplying them!.
   - Now try making some matrices with these commands: 'zeros', 'ones', 'eye' (identity), 'rand', and 'delsq' (from your HW!). If you have a big matrix A that is mostly zeros, use 'A = sparse(A)' to avoid storing all those zeros needlessly. Figure out how to use the 'diag' and 'spdiags' commands to make a matrix B which is $1000 \times 1000$ matrix with -2 on the main diagonal and 1 on the diagonals directly above and below. Type 'whos B' to see how many bytes variable B needs in each case.
2. (What's taking so long?)
   - Find the iterdemo.m script from course website and copy it. Set n = 500. Add the line 'tic' to the beginning and 'toc' to the end. How long did the script take to run? Change the Jacobi solver so that instead of using '\' to form the inverse matrix at the beginning, you apply the inverse within the loop using the 'inv' command. How long does it take to run now?
   - Now type 'Profile On', run the script again, type 'Profile Off,' and then finally, 'Profile Viewer.' The profiler lets you break down how much time your code spends on each task.
3. (Plotting)
   - Try plotting a quadratic function using the array of points you used before. Type 'plot(x,x.^2)' (what happens if you forget the '.' before the multiplication operator? Now make a plot with logrithmic axes by using loglog (also check out its cousins semilogx and semilogy).
   - Make a matrix A using the commands above. Use 'spy(A)' to make a plot showing where the nonzero entries are. Make sure to try this one on delsq! Now make another matrix B. Type 'subfig(2,1,1) spy(A)' then 'subfig(2,1,2) spy(B)'. Wow! Two plots in one figure.

Part II: Approximation Methods

1. (Normal Equations as closest point) Minimize the 2-norm of the residual

$$\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}$$

with respect to $x$ to obtain the normal equations

$$\mathbf{A}^T\mathbf{A}x = \mathbf{A}^T\mathbf{y}$$

Show that the approximant $\mathbf{A}\mathbf{x}$ is orthogonal to the residual.

2. (Trig Identities) Use the complex exponential definition of sine and cosine to prove the product formulas.

$$\cos(u)\cos(v) = \frac{1}{2}\left(\cos(u-v) + \cos(u+v)\right), \quad \sin(u)\sin(v) = \frac{1}{2}\left(\cos(u-v) - \cos(u+v)\right),$$

and now show that

$$\boxed{\cos^2(x) = \frac{1}{2}\left(1 + \cos(2x)\right), \quad \sin^2 x = \frac{1}{2}\left(1 - \cos(2x)\right).}$$

Remember this formula. It is incredibly handy.[1]

3. (Polynomial Long Division) Find the remainder when $x^2 - 3x + 1$ is divided into $4x^3 - 2x + 5$.

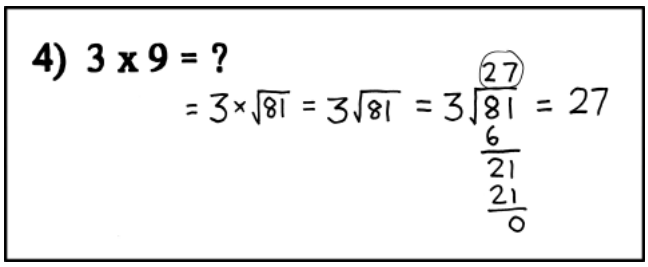


**Figure 1:** source: www.xkcd.com

4. (Chebyshev Polynomials) List your favorite property of Chebeyshev polynomials. Explain. For bonus points, explain in Haiku.

Prepared by ANNA LIEB. Comments and questions: **lieb@math.berkeley.edu**

[1] Handy mnemonic from my high school calc teacher: the one for $\sin(x)$ has a negative sign and sin is negative