

ACM 106a, Problem Set 1

Due: Thursday, October 18, 2007

Theory

Stoer & Bulirsch, exercise 1.4 (p. 33)

MATLAB warm-up

This warm-up is designed to get you comfortable with the basic features of MATLAB: specifically, writing and calling functions, and manipulating arrays (vectors and matrices). Be sure to use the online documentation for MATLAB, linked from the class web site, for reference.

Remark: Whenever possible, you should avoid using for loops to iterate over the entries of an array individually; try to see if you can use MATLAB's array manipulation syntax instead. Doing things this way is usually much, much faster than the same code done with a for loop, since these routines are highly optimized.

What to hand in: a printout of your code for problems 1–4 (`dotproduct.m`, `blproduct.m`, `makeblproduct.m`, and `hw1test.m`), and the output from the `hw1test` script from problem 4.

1. Write a function `dotproduct` that takes two column vectors u and v as input, and outputs their dot product $u \cdot v$.
2. Write a function `blproduct` that takes a square matrix M and two column vectors u and v , and outputs the bilinear product $u^T M v$. (Here u^T denotes the transpose of u , which is a row vector.) You should verify that this gives the same result as `dotproduct(u, v)` when $M = I$ is the identity matrix.
3. Write a function `makeblproduct` that takes a square matrix M and outputs a *new function*, representing M as a bilinear product. This new function should then take two column vectors u and v and output $u^T M v$. You should verify that the following code

```
f = makeblproduct(M);  
f(u, v)
```

produces the same result as

```
blproduct(M, u, v)
```

and in particular, that when $M = I$, the function f is the same as the `dotproduct` function.

Remark: Why on earth would we ever want to write a function that outputs another function? Many routines in MATLAB (nonlinear solvers, differential equation solvers, etc.) require you to input the handle to a function with particular specifications. It's often useful to write a function to create these types of functions, which can then be passed to the solver.

4. Write a script called `hw1test.m` that does the following. First, assign the variables

$$u = \begin{pmatrix} 4 \\ 3 \\ 1 \end{pmatrix}, \quad v = \begin{pmatrix} 5 \\ 6 \\ 2 \end{pmatrix}, \quad M = \begin{pmatrix} 8 & 6 & 9 \\ 4 & 5 & 1 \\ 7 & 3 & 2 \end{pmatrix}.$$

Next, output the values of $u \cdot v$ and $u^T M v$ using the functions you programmed above. You should have *two* (hopefully identical) outputs for $u^T M v$: one using the method from problem 2, the other using the method from problem 3. (The output of your test script will be used to check the correctness of the functions you wrote, so be sure to double-check this part!)