HW 1

Due: Monday, Sep 10.

Remember that you may (and should!) talk about the problems amongst yourselves, or discuss them with me, providing attribution for any good ideas you might get – but your final write-up should be your own.

1: A problem of performance. In addition to dense matrices, MATLAB supports a sparse matrix data structure, in which only the nonzero elements of the matrix are stored. For a variety of square matrix size n and sparsity levels s (where s is the fraction of the entries that are nonzero), compare the speed of dense matrix-vector multiply and sparse matrix-vector multiply. You can use `As = sparse(A)` to make a sparse version of a dense matrix A. What do you observe about the relative performance of these options?

Note: Your performance will vary depending on your machine and your version of MATLAB, of course. You may also do the equivalent experiment in Python.

2: Identity plus low-rank. Suppose $u, v \in \mathbb{R}^m$ and $A = I + uv^T$. Then

1. If A is nonsingular, then $A^{-1} = I + \alpha uv^T$. What is α?
2. Under what conditions is A singular? And if it is singular, what is the null space of A?
3. Assuming u and v are given, give a MATLAB code fragment that computes $x = A^{-1}b$ in $O(n)$ time.

3: Norm! Show the following norm identities for $x, y \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m\times n}$:

1. $\|x\|_2^2 \leq \|x\|_1 \|x\|_\infty$
2. $\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1$
3. $\|xy^T\|_2 = \|x\|_2 \|y\|_2$
4. $\|A\|_\infty \leq \sqrt{n}\|A\|_2$
5. $\|A\|_2 \leq \sqrt{m}\|A\|_\infty$

4: Orthogonality and perturbations. Suppose $Q : \mathbb{R} \rightarrow \mathbb{R}^{n \times n}$ is a differentiable matrix-valued function and that $Q(t)$ is orthogonal for all t. Show that $S(t) = Q(t)^T \dot{Q}(t)$ is a skew-symmetric matrix, i.e. $S(t) = -S(t)^T$.

5: Error in a classic recurrence. The following routine estimates π by recursively computing the semiperimeter of a sequence of 2^{k+1}-gons embedded in the unit circle:

```
N = 4;
L(1) = sqrt(2);
s(1) = N*L(1)/2;
for k = 1:30
  N = N*2;
  L(k+1) = sqrt( 2*(1-sqrt(1-L(k)^2/4)) );
  s(k+1) = N*L(k+1)/2;
end
```

Plot the error $|s_k - \pi|$ against k. Explain why the algorithm behaves as it does, and describe a reformulation of the algorithm that does not suffer from this problem.