Practice Final

This practice final is meant to give you an idea what to expect on the actual final. I expect to follow roughly this format. Minor errors in MATLAB syntax and function names will not be penalized.

1: (2 points each) Let $u, v \in \mathbb{R}^n$. True or false:

1. $|\operatorname{fl}(u^T v) - u^T v| \leq n\epsilon_{\mathrm{mach}}|u|^T|v| + O(\epsilon_{\mathrm{mach}}^2)$
2. $\|u\|_\infty \leq \|u\|_2 \leq \sqrt{n}\|u\|_\infty$
3. For any real symmetric $A \in \mathbb{R}^{n\times n}$ there is an associated inner product $\langle x, y\rangle_A = x^T Ay$.
4. For any square A, sensitivity of the eigenvalues of A is bounded by $\kappa_2(A) = \|A\|\|A^{-1}\|$.
5. The QR iteration is a backward stable method to compute eigenvalues.
6. If A is a square matrix, every eigenvalue is within $\|A\|_1$ of some diagonal entry of A.

2: (6 points each) One-liners in MATLAB:

1. Suppose A is a real symmetric matrix and `[L,U,P] = lu(A)` returns $P = I$. Compute the number of positive eigenvalues of A.
2. Suppose $A = QR$ is computed using Householder reflections, and you are given R. Compute $\det(A)$ (do not use `det` or refer explicitly to A or Q)
3. Suppose $A = QR$ is given. Solve $Ax = b$ in $O(n^2)$ time.

3: (10 points) Fill in the following routine

```
function Sdot = differentiate_svd(A,E)
%
% Return Sdot = dS/dt(0) where S(t) is the singular value
% matrix for A+tE.  Assume the singular values of A are distinct.
```

4: (20 points) Without using the Matlab backslash, write a routine that solves $Ax = b$ where A is a symmetric positive definite tridiagonal.

```
function x = solve_tridiag(alpha, beta, b)
%
% Solve Ax = b where A = diag(alpha) + diag(beta,-1) + diag(beta,1)
```

5: (20 points) Fill in the following routine

```
function A = generate_test(lambda, kappa)
%
% Produce a randomly-generated test matrix A = V Lambda V^{-1}
% where Lambda = diag(lambda) and cond(V) = kappa.
```

6: (20 points) Fill in the following routine. Your routine should not use `schur`, `eig`, or any other Matlab functions beyond basic arithmetic and matrix operations.

```
function [Q,T] = schur2(A)
%
% Compute a Schur factorization of the 2-by-2 matrix A.
```