Midterm
(due: 2020-03-13)

You may (and should) consult any resources you wish except for people (outside the course staff). Provide attribution for any good ideas you might get. Your final write-up should be your own.

**True/false** For each of the following statements, either give a brief argument that it is true or provide an example to show it is false.

1 pt The eigenvalues of a real symmetric positive definite matrix are all positive.

1 pt Every square nonsingular matrix $A$ can be decomposed as $A = LU$ where $L$ is unit lower triangular and $U$ is upper triangular.

1 pt The eigenvalues of a 2-by-2 matrix are differentiable functions of the matrix entries.

1 pt For every square matrix, power iteration eventually converges to a dominant eigenvector.

1 pt If $A \in \mathbb{R}^{n \times n}$ has condition number $\kappa_2(A) = 1$, then $A = \alpha Q$ for some real $\alpha \neq 0$ and orthogonal matrix $Q \in \mathbb{R}^{n \times n}$.

1 pt If $A \in \mathbb{R}^{m \times n}$ has full column rank, then $A^\top A = I$.

**Speedy and stable** Consider each of the following:

2 pts Rewrite the following code for better speed and numerical stability assuming $PA = LU$ is already computed.

```
   dx = -inv(A)*dA*inv(A)*b
```

2 pts For $x > 1$, the equation $x = \cosh(y)$ can be solved as

$$ y = -\ln \left( x - \sqrt{x^2 - 1} \right). $$

What happens when $x = 10^8$? How can we fix it?

2 pts Explain the output of the following code fragment

1 x = 2;
2 for k = 1:100, x = sqrt(x); end
3 for k = 1:100, x = x^2; end
4 disp(x);

Polynomial fitting Consider fitting a polynomial of degree at most $d$ to $n$ data points, i.e.

$$\minimize \sum_{i=1}^{n} (p(x_i) - y_i)^2$$

2 pts Complete the following code to find $p(x) = \sum_{j=0}^{d} c_j x^j$ that solves the minimization problem by a standard linear least squares formulation:

```matlab
function [c] = poly_least_squares(x, y, d)
    % You may write equivalent code in Julia or Python.
end
```

2 pts Modify the code to solve the problem

$$\minimize \sum_{i=1}^{n} (p(x_i) - y_i)^2 + \mu \|c\|^2$$

In MATLAB, your routine should have the interface

```matlab
function [c] = poly_least_squares_tik(x, y, d, mu)
    % You may write equivalent code in Julia or Python.
end
```

2 pts Modify the code to solve the problem

$$\minimize \sum_{i=1}^{n} (p(x_i) - y_i)^2 + \mu \int_{-1}^{1} p'(x)^2 \, dx$$

In MATLAB, your routine should have the interface

```matlab
function [c] = poly_least_squares_smooth(x, y, d, mu)
    % You may write equivalent code in Julia or Python.
end
```
Sensitive pseudoinverse Consider the problem

\[ x(s) = (A + sE)^\dagger b \]

where \( A \in \mathbb{R}^{m \times n} \) has full column rank. Suppose the economy QR factorization \( A = QR \) is given.

2 pts Compute \( x(0) \). Your code should take at most \( O(mn) \) time. In MATLAB, this means filling in the second line in this fragment:

1. \([Q,R] = qr(A,0); \ % \text{This is } O(mn^2)\]
2. \( x0 = \ldots; \ % \text{This should be } O(mn)\)

4 pts Compute \( x'(0) \) (worth two points). Your code should take at most \( O(mn) \) time (worth two points). You can sanity check your code by a finite difference test:

1. \( h = 1e-6; \)
2. \( xp = (A+h*E)\backslash b; \)
3. \( xm = (A-h*E)\backslash b; \)
4. \( dx_fd = (xp-xm)/2/h; \)
5. \( dx = \% \text{your computation} \)
6. \( \text{relerr} = \text{norm(dx_fd-dx)/norm(dx)} \)

Conditioning

2 pts Argue that if \( A \in \mathbb{R}^{n \times n} \) is symmetric and positive definite and \( A_{11} \in \mathbb{R}^{k \times k} \) is a leading principal submatrix, then \( \kappa_2(A_{11}) \leq \kappa(A) \), where \( \kappa \) denotes the usual 2-norm condition number for solving linear systems.

Hint: You may use the interlace theorem, which tells us that if \( \mu_1 \leq \mu_2 \leq \ldots \leq \mu_k \) are the eigenvalues of \( A_{11} \) and \( \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \) are the eigenvalues of \( A \), then

\[ \lambda_j \leq \mu_j \leq \lambda_{n-k+j}. \]

2 pts Show by example that the hypothesis that \( A \) is positive definite is necessary in the above statement.

2 pts Show that if \( A \in \mathbb{R}^{m \times n} = [A_1 \ A_2] \), then \( \kappa(A_1) \leq \kappa(A) \), where \( \kappa \) here denotes the usual condition number for least squares.