Final

You may use any written material you want (textbook, notes, review notes, assignments and solutions, etc), but not a friend nor a computing device. All ten questions are worth 6 points total, split evenly among subproblems. Not all questions are the same difficulty, but I have tried to arrange the questions in order of increasing difficulty.

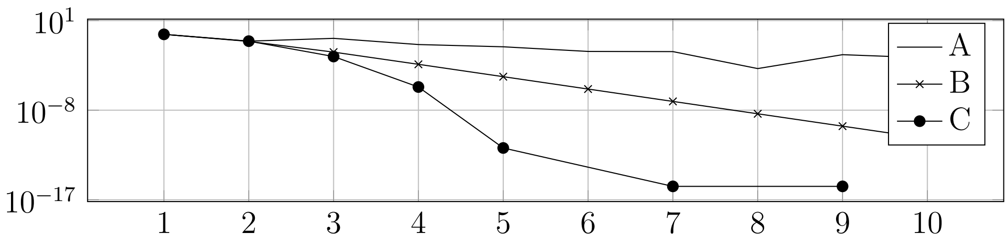
Please write your NetID at the start of each page.

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

1: Which is which The following three graphs show the convergence of three methods of computing $\sqrt{2}$

1. Bisection on $x^2 - 2 = 0$ with initial interval $[0, 2]$
2. $x_{k+1} = x_k/2 + 1/x_k$ with $x_0 = 1$
3. $x_{k+1} = x_k - (x_k^2 - 2)/3$ with $x_0 = 1$

Label which is which



Answer: (2 points each)

1. Bisection is A
2. Newton is C
3. Fixed point is B

2: Rewrite for efficiency Rewrite the following MATLAB code to achieve an equivalent computation within the given efficiency constraints.

```
[L,U,P] = lu(A); % Ignore this cost

% Rewrite for O(n) time
B = u*v';
G = B'*B;
x = G*b;

% Rewrite for O(n^2) time
y = A\b;
yhat = A\b - A\E*A*b;

% Rewrite for O(n^2) time
Ainv = inv(A);
z = Ainv(1,1);
```

Answer: Rubric: 2 points each (1: basic idea, 1: correct performance)

```
x = u*((v'*v)*(v'*b));

y = U\(L\(P*b));
yhat = y - U\(L\(P*(E*b)));

v = U\(L\P(:,1));
z = v(1);
```

3: Extending QR Suppose $A \in \mathbb{R}^{m \times n}$, $m > n$ has full column rank and we have pre-computed the economy QR decomposition

1. Write one line of MATLAB to efficiently minimize $\|Ax - b\|_2^2$ using the QR decomposition.
2. Let $v = r/\|r\|$ where $r = b - Ax$ is the least squares residual (assume r nonzero). Argue that v is orthogonal to the columns of Q, and write b as a linear combination of v and the columns of Q.
3. Write the economy QR decomposition of $\tilde{A} = \begin{bmatrix} A & b \end{bmatrix}$ in terms of the QR factors of A and the solution and residual to the least squares problem of minimizing $\|Ax - b\|_2^2$.

Answer:

1. (2 points: 1 correctness, 1 performance)

```
x = R\(Q'*b);
```

2. (1 point each) $0 = A^T r = R^T Q^T v \|r\|$ and R is nonsingular, so $Q^T v = 0$. And
$$b = r + Ax = \|r\|v + Q(Rx).$$

3. (2 points) Put together the QR decomposition with the previous step
$$\tilde{A} = \begin{bmatrix} Q & v \end{bmatrix} \begin{bmatrix} R & Rx \\ 0 & \|r\| \end{bmatrix}$$

4: Rayleigh redux Consider the Rayleigh quotient

$$\rho_A(v) = \frac{v^T A v}{v^T v}.$$

1. If A is symmetric, show $\rho_A(v)$ is a weighted average of the eigenvalues.
2. Show $\rho_A(v)$ is the solution to the least squares problem

$$\text{minimize}_\mu \|Av - v\mu\|_2^2.$$

Note: This holds whether or not A is symmetric.

Answer:

1. (1 point for $A = Q\Lambda Q^T$ + 2 for rest.) Write $A = Q\Lambda Q^T$ and let $w = Q^T v/\|v\|$. Then

$$\rho_A(v) = w^T \Lambda w = \sum_{j=1}^{n} w_j^2 \lambda_j$$

where $\sum_j w_j^2 = \|w\|^2 = 1$.

2. (3 points) The normal equations for the problem are

$$x^T x \mu = x^T A x,$$

which gives $\mu = x^T Ax/x^T x = \rho_A(x)$.

5: Schur thing Suppose $A = UTU^*$ is a given complex Schur form.

1. What is the relationship between the eigenvalues of A and of $(A - zI)^{-1}$?
2. Give one line of MATLAB to compute $f(z) = \mathrm{tr}((A - zI)^{-1})$ for arbitrary $z \in \mathbb{C}$ in $O(n)$ time.

Answer:

1. (3 points) By the spectral mapping theorem, the eigenvalues of $(A - zI)^{-1}$ are $1/(\lambda - z)$ where λ is an eigenvalue of A.
2. (2 points for idea, 1 for performance)

```
fz = sum(1./(diag(T)-z));
```

6: Nearly Newton Consider the approximate iteration

$$x_{k+1} = x_k - \hat{J}(x_k)^{-1} f(x_k).$$

where $\hat{J}(x_k) = J(x_k) + E(x_k)$ is an approximation to the Jacobian $J(x_k)$, and suppose x_* satisfies $f(x_*) = 0$.

1. Write an iteration for the error $e_k = x_k - x_*$.
2. Suppose that in a neighborhood of x_* we have
$$\kappa(\hat{J}(x)) \frac{\|E(x)\|}{\|\hat{J}(x)\|} < \gamma.$$
Argue that the iteration converges at least linearly with rate γ for starting points near enough to x_*.
Hint: Note that $\hat{J}^{-1} J = I - \hat{J}^{-1} E$.
3. A friend suggests an approximate Gauss-Newton iteration for the least squares problem of minimizing $\|f(x)\|^2$:
$$x_{k+1} = x_k - \hat{J}(x_k)^{\dagger} f(x_k).$$
Explain why this is a bad idea.

Answer:

1. (2 points) The error iteration is
$$e_{k+1} = e_k - \hat{J}_k^{-1}(J_k e_k) + O(\|e_k\|^2) = \hat{J}_k^{-1} E_k e_k + O(\|e_k\|^2)$$
2. (2 points) Taking norms gives
$$\begin{aligned} \|e_{k+1}\| &\leq \|\hat{J}_k^{-1}\| \|E_k\| \|e_k\| + O(\|e_k\|^2) \\ &= \kappa(\hat{J}_k) \frac{\|E_k\|}{\|\hat{J}_k\|} \|e_k\| + O(\|e_k\|^2) \\ &\leq \gamma \|e_k\| + O(\|e_k\|^2) \end{aligned}$$
3. (2 points) The iteration has the wrong stationary point equation: $\hat{J}(x)^T f(x) = 0$ rather than $J(x)^T f(x) = 0$.

7: Simply stationary Suppose A is symmetric and positive definite and consider the energy function

$$\phi(x) = \frac{1}{2}x^T Ax - x^T b.$$

1. Write a MATLAB code for a fixed number of steps of the steepest descent iteration
$$x_{k+1} = x_k - \eta \nabla \phi(x_k)$$
for fixed η.
2. Write an error iteration. What range of η gives convergence?
3. Find the η that minimizes the spectral radius of the iteration matrix.

Answer:

1. (2 points) The iteration is
```
for k = 1:maxiter
  x = x − eta∗(b−A∗x);
end
```
2. (1 for iteration, 1 for analysis) The error iteration is
$$e_{k+1} = (I - \eta A)e_k$$
and the eigenvalues of $I - \eta A$ are $1 - \eta\lambda$ for eigenvalues λ of A. For $0 < \eta < 2/\lambda_{\max}(A)$, all the eigenvalues of $I - \eta A$ lie between -1 and 1.
3. (2 points) At $\eta = 2/(\lambda_{\max}(A) + \lambda_{\min}(A))$, we have
$$\rho = 1 - \eta\lambda_{\min} = -(1 - \eta\lambda_{\max})$$
The spectral radius is $(\lambda_{\max} - \lambda_{\min})/(\lambda_{\max} + \lambda_{\min})$.

8: Simple solver Consider the iteration

$$Ax_{k+1} = b - \psi(x_k)$$

where ψ satisfies

$$\|\psi(x - y)\| \leq \gamma \|x - y\|.$$

This condition is called a *Lipschitz* condition on ψ.

1. Let $d_k = x_{k+1} - x_k$ be the size of step k, and show that

$$\|d_{k+1}\| \leq \gamma \|A^{-1}\| \|d_k\|.$$

   This computation is very much like the one we do when setting up an error iteration.

2. Assuming $\gamma \|A^{-1}\| < 1$, argue that for any initial guess, for all k,

$$\|x_k - x_0\| \leq \frac{\|d_0\|}{1 - \gamma \|A^{-1}\|}.$$

Answer: Rubric: 3 points each.

1. Subtract two successive iteration steps to get

$$A(x_{k+1} - x_k) = -(\psi(x_k) - \psi(x_k - 1)).$$

   Multiply by A^{-1} and use the Lipschitz condition to obtain

$$\|d_{k+1}\| \leq \gamma \|A^{-1}\| \|d_k\|$$

2. Note that

$$\|x_k - x_0\| = \|\sum_{j=0}^{k-1} d_j\| \leq \sum_{j=0}^{k-1} \|d_j\| \leq \sum_{j=0}^{k-1} (\gamma \|A^{-1}\|)^j \|d_0\| < \frac{\|d_0\|}{1 - \gamma \|A^{-1}\|}.$$

9: Prony problem Consider the problem of finding α and β to minimize $\sum_{j=1}^{n} (\alpha \exp(\beta x_j) - y_j)^2$ where y_j is given data. An approximation to the problem is to minimize $\sum_{j=1}^{n} \left(\log \hat{\alpha} + \hat{\beta} x_j - \log(y_j) \right)^2$.

1. Write a MATLAB code to solve the latter least squares problem.
2. Write MATLAB for a Gauss-Newton iteration for the nonlinear problem. Just take a fixed number of steps – no need to worry about globalization nor convergence testing.

Answer:

```
% Solve initial linear least squares problem
n = length(x);
z = [ones(n,1), x]\log(y);
alpha = exp(z(1));
beta = z(2);

% Gauss-Newton iteration
for k = 1:maxiter
  r = alpha*exp(beta*x)-y;
  J = [exp(beta*x), alpha*(x.*exp(beta*x)];
  p = -J\r;
  alpha = alpha + p(1);
  beta = beta + p(2);
end
```

10: Bordered solves Suppose $A \in \mathbb{R}^{n \times n}$ is positive definite and $g : \mathbb{R}^n \to \mathbb{R}$ is nonlinear, and consider the constrained optimization problem

$$\text{minimize } \phi(x) \text{ s.t. } g(x) = 0, \quad \phi(x) \equiv \frac{1}{2}x^T Ax - x^T b.$$

1. Form the Lagrangian and its gradient and Hessian.
2. Write MATLAB code to take `maxiter` modified Newton steps for the problem (drop the term involving the Hessian of g). Assume an initial guess is given, and don't worry about line search, termination criteria, etc. Assume a function `[g,gradg] = evalg(x)` that returns the function value and gradient of g in $O(n^2)$ time. Your code should take $O(n^3)$ to set up, then $O(n^2)$ per step.

Answer: The Lagrangian, its gradient, and its Hessian are

$$\begin{aligned} L &= \frac{1}{2}x^T Ax - x^T b + \lambda g(x) \\ \nabla L &= \begin{bmatrix} Ax - b + \lambda \nabla g(x) \\ g(x) \end{bmatrix} \\ H_L &= \begin{bmatrix} A + H_g & \nabla g \\ \nabla g^T & 0 \end{bmatrix} \end{aligned}$$

Rubric: 1 point each

Though the code can be written in several ways, the key is to use block Gaussian elimination and to avoid re-factoring A

```
R = chol(A);
for k = 1:maxiter
  [g,dg] = evalg(b−A∗x−lambda∗g);
  r = b−A∗x−lambda∗dg;
  u = R\(R'\dg);
  dlambda = (u'∗r)/(u'∗dg);
  dx = R\(R'\(r−dlambda∗dg);
  lambda = lambda + dlambda;
  x = x + dx;
end
```

Rubric: 1 point for right idea, 1 point for correctness, 1 point for performance

Extra work space.