Errors and Complexity

1. What is the formula for relative error?

2. What is the formula for absolute error?

3. Which is usually the better way to calculate error?

4. If you have k accurate significant digits, what is the tightest bound on your relative error?

5. Given a maximum relative error, what is the largest (or smallest) value you could have?

6. How do you compute relative and absolute error for vectors?

7. What is truncation error?

8. What is rounding error?

9. What does it mean for an algorithm to take O(n^3) time?

10. Can you give an example of an operation that takes O(n^2) time?

11. What does it mean for error to follow O(h^4)?

12. If you know an operation is O(n^4), can you predict its time/error on unseen inputs from inputs that you already have?

13. If you are given runtime or error data for an operation, can you find the best x such that the operation is O(n^x)?

Floating point numbers

1. What are the differences between floating point and fixed point representation?

2. Given a real number, what is the rounding error involved in storing it as a machine number? What is the relative error?

3. Explain the different parts of a floating-point number: sign, significand, and exponent.

4. How is the exponent of a machine number actually stored?

5. What is machine epsilon?

6. What is underflow (UFL)?

7. What is overflow?

8. Given a toy floating-point system, determine machine epsilon and UFL for that system.

9. How can we bound the relative error of representing a real number as a normalized machine number?

10. How do you store zero as a machine number?

11. What is cancellation? Why is it a problem?

12. What types of operations lead to catastrophic cancellation?

13. Given two different equations for evaluating the same value, can you identify which is more accurate for certain x and why?

Taylor series

1. What is the general form of a Taylor series?

2. How do you use a Taylor series to approximate a function at a given point?

3. How can you approximate the derivative of a function using Taylor series?

4. How can you approximate the integral of a function using Taylor series?

5. Given a function and a center, can you write out the n-th degree Taylor polynomial?

6. For an n-th degree Taylor polynomial, what is the bound on the error of your approximation as a function of distance from the center?

7. Can you determine how many terms are required for a Taylor series approximation to have less than some given error?

Vector and matrix norms

1. What is a vector norm? (What properties must hold for a function to be a vector norm?)

2. What is the definition of an induced matrix norm? What do they measure?

3. What properties do induced matrix norms satisfy?

4. For an induced matrix norm, given $\|x\|$ and $\|Ax\|$ for a few vectors, can you determine a lower bound on $\|A\|$?

5. For a given vector, compute the 1, 2 and ∞ norm of the vector.

6. For a given matrix, compute the 1, 2 and ∞ norm of the matrix.

Sparse matrices

1. What does it mean for a matrix to be sparse?

2. Given a sparse matrix, put the matrix in CSR format.

3. Given a sparse matrix, put the matrix in COO format.

4. For a given matrix, how many bytes total are required to store it in CSR format?

5. For a given matrix, how many bytes total are required to store it in COO format?

Linear System of Equations

1. Given a factorization PA=LU, how would you solve the system Ax=b?

2. Recognize and understand Python code implementing forward substitution, back substitution, and LU factorization. Write pseudo-codes here.

3. What happens if we try to solve a system Ax=b with a singular matrix A?

4. Why do we use pivoting when solving linear systems?

5. How do we choose a pivot element?

6. What is the cost of matrix-matrix multiplication?

7. What is the cost of computing an LU or LUP factorization?

8. What is the cost of forward or back substitution?

9. What is the cost of solving Ax=b for a general matrix?

10. What is the cost of solving Ax=b for a triangular matrix?

11. What is the cost of solving Ax=bi with the same matrix A and several right-hand side vectors bi?

12. Given a process that takes time O(n^k), what happens to the runtime if we double the input size (i.e. double n)? What if we triple the input size?

Conditioning

1. What is the definition of a condition number?

2. What is the condition number of solving Ax=b?

3. Calculate the p-norm condition number of a matrix for a given p.

4. Do you want a small condition number or a large condition number?

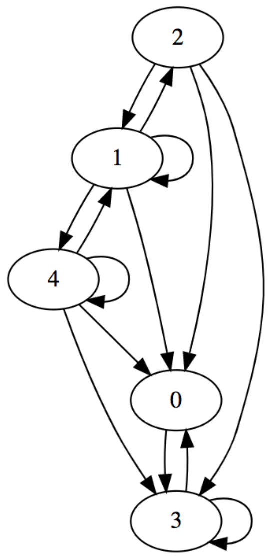
5. If you have p accurate digits in a A and b, how many accurate digits do you have in the solution of Ax=b if the condition number of A is 10^k?

6. When solving a linear system Ax=b, does a small residual guarantee an accurate result?

7. Consider solving a linear system Ax=b. When does Gaussian elimination with partial pivoting produce a small residual?

Graphs

1. Given a given graph (weighted or unweighted), determine the adjacency matrix



2. What is a transition matrix? Given a graph representing the transitions or a description of the problem, determine the transition matrix. (Write the transition matrix for the graph above)

3. Given an initial state, how can you use a transition matrix to determine the state (probability vector) after 1 timestep? After 2?

4. What properties must be true for a transition matrix?

Eigenvalues

1. What is the definition of an eigenvalue/eigenvector pair?

2. If v is an eigenvector of A, what can we say about cv for any nonzero scalar c?

3. What is the relationship between the eigenvalues of A and the eigenvalues of

    a. cA for some scalar c,
    b. $(A-\sigma I)$ for some scalar σ
    c. $A-1$?

4. What is the relationship between the eigenvectors of A and the eigenvectors of

    d. cA for some scalar c,
    e. $(A-\sigma I)$ for some scalar σ
    f. $A-1$?

5. Be able to run a few steps of normalized power method.

6. When can power method (or normalized power method) fail?

7. How can we approximate an eigenvalue of A given an approximate eigenvector?

8. What happens to the result of power method if the initial guess does not have any components of the dominant eigenvector? Does this depend on whether we are using finite or infinite precision?

9. How can we find eigenvalues of a matrix other than the dominant eigenvalue?

10. Complete the table:

	(Normalized) Power Iteration	Inverse Iteration	Shifted inverse iteration	Rayleigh Quotient iteration
Converges to which eigenvalue?				
Cost				
Convergence rate				

Singular value decomposition

1. For a matrix A with SVD decomposition $A=U\Sigma V^T$, what are the columns of U and how can we find them? What are the columns of V and how can we find them? What are the entries of Σ and how can we find them?

2. What are the shapes of U, Σ, and V in the full SVD of an $m\times n$ matrix?

3. What are the shapes of U, Σ, and V in the reduced SVD of an $m\times n$ matrix?

4. What is the cost of computing the SVD?

5. Given an already computed SVD of a matrix A, what is the cost of using the SVD to solve a linear system $Ax=b$? How would you use the SVD to solve this system?

6. Given an already computed SVD of a matrix A, what is the cost of using the SVD to solve a least squares problem $Ax\cong b$? How do you solve a least squares problem using the SVD?

7. How do you use the SVD to compute a low-rank approximation of a matrix?

8. Given the SVD of a matrix A, what is the SVD of A+ (the pseudoinverse of A)?

9. Given the SVD of a matrix A, what is the 2-norm of the matrix? What is the 2-norm condition number of the matrix?

Linear Least-squares

1. What does the least-squares solution minimize?

2. For a given model and given data points, can you form the system $Ax \cong b$ for a least squares problem?

3. What are the differences between least squares data fitting and interpolation?

4. Given the SVD of a matrix A, how can we use the SVD to compute the least squares solution? Consider also the case when A is rank deficient. Be able to do this for a small problem.

5. Why would you use the SVD instead of normal equations to find the solution to $Ax \cong b$?

6. Which costs less: solving a least squares problem via the normal equations or solving a least squares problem using the SVD?

Interpolation

1. How many points do you need to uniquely determine an nth-degree polynomial?

2. An nth-degree polynomial can interpolate how many points?

3. How do you create a Vandermonde matrix for interpolation with the monomial basis functions?

4. How do you create a generalized Vandermonde matrix for interpolation with other basis functions?

5. What do you know about the columns of the Vendermonde matrix?

6. What does the right-hand-side vector of an interpolation system look like?

7. How does the error of interpolation vary with degree of the interpolating polynomial? Meaning, how accurate is an interpolant in approximating a function f(x) over an interval?

Nonlinear Equations

1. For a given nonlinear equation (1D), you should be able to run a couple steps of:
    a. Bisection method

    b. Secant method

    c. Newton's method

2. How many function evaluations are required per iteration for bisection?

3. What is the convergence rate of bisection method? Will it always converge?

4. Using the bisection method, given a specific initial interval [a,b] and a given tolerance tol, how many iterations would be required for the approximate root to be accurate to the given tolerance? Remember that the error for the bisection method is defined as the length of the interval.

5. How many function evaluations are required per iteration for 1D Newton's method for root-finding? Which functions must be evaluated?

6. What is the convergence rate of Newton's method for solving 1D nonlinear equations?

7. How many function evaluations are required per iteration for secant method?

8. What is the convergence rate of secant method? Will it always converge?

9. For a given vector-valued function f(x), what is the Jacobian.

10. For a simple system of nonlinear equations, you should be able to run one step of N-dimensional Newton's method.

11. What is the convergence rate of Newton's method for root-finding in N dimensions? Will it always converge?

12. What operations are required per iteration for Newton's method in N dimensions?

Optimization

1. What are the necessary and sufficient conditions for a point to be a local minimum in one dimension?

2. What are the necessary and sufficient conditions for a point to be a local minimum in N dimensions?

3. How do you classify extrema as minima, maxima, or saddle points (answer for 1D and ND)?

4. Run one iteration of golden section search.

5. Calculate the gradient of a function (function has many inputs, one output).

6. Calculate the Hessian of a function.

7. Find the search direction in steepest/gradient descent.

8. Why must you perform a line search each step of gradient descent?

9. Run one step of Newton's method in one dimension.

10. Run one step of Newton's method in N dimensions.

11. What operations do you need to perform each iteration of golden section search?

12. What operations do you need to perform each iteration of Newton's method in one dimension?

13. What operations do you need to perform each iteration of Newton's method in N dimensions?