# Linear systems

* , where is unknown, , and is given:
* Normal matrices:
  + Symmetric matrices:
    - The eigenvalues are always real
    - The eigenvectors corresponding to distinct eigenvalues are always orthogonal
  + Orthogonal matrices: where denotes the identity matrix
    - Product of orthogonal matrix is also orthogonal
* Inverse of a matrix:
* Norm: the function is a norm if it is
  + Point separating:
  + Sub-additive:
  + Homogeneous:
* Examples:
  + L1 norm:
  + L2 norm:
  + Lp norm:
  + Infinity norm:
  + Unit sphere: for any vector space
* *Induced* norms:
  + All induced norms satisfy: and are sub-multiplicative
  + orthogonal matrix
* Other matrix norms and operators:
  + L1 norm:
  + L2 norm:
  + Infinity norm:
  + Trace:
    - Notice that:
      * Notice that:
  + Frobenius norm:
    - Notice that
  + Condition number for the norm :
    - For a symmetric (or more generally normal) and inversible matrix, the condition number for the L2 norm is the ratio between the largest absolute value of its eigenvalues and the smallest value of its eigenvalues
* Triangular systems: , where is unknown, , and is given:
* Backwards substitution method (for upper triangular systems)
  + Algorithm:
* Triangularisable matrices:
  + *Invertible operations*: let be an invertible matrix:
* Gaussian Elimination
  + Use sequence of invertible operations such that:
  + being an upper triangular matrix. Then solve:
    - , , where for and
    - , where is the *k*th unit coordinate vector and
      * (invertible)
      * (compositions are lower triangular)
  + Choosing a pivot:
    - Default: choose as the pivot
    - Partial pivot: on column , it is chosen the element below the diagonal with the largest absolute value:
    - Total pivot: choose the largest element below or to the right of the diagonal:
  + Gaussian elimination gives a triangular decomposition
    - * LU decomposition: Let be an invertible matrix such that the sub-matrix is invertible for . Then the LU decomposition exists: . If is enforced, then de decomposition is unique
  + Each system can be solved with two triangular solves:
* Gauss Jordan method for inversion: Gaussian elimination to compute the inverse of A
* Cholesky decomposition
  + is positive definite if
    - M is positive definite if and only if it is symmetric, and all of its eigenvalues are positive
    - For positive definite matrices, it is possible to compute an LU decomposition with
  + symmetric positive definite matrix. There exists a lower triangular matrix such that
  + Algorithm:

# Eigenvalues and singular values

* is an eigenvector with associated eigenvalue of if
  + Consequently is not invertible:
    - The eigenvalues of A are distinct if and only if has n solutions
  + Theorem (Abel-Ruffini): there is no exact algebraic formula for the roots of a polynomial with degree 5 or more
* Let , and : is eigenvector and an eigenvalue of if and
  + is referred to as an eigenpair of
  + is the spectrum of if contains all the eigenvalues of A, that is:
  + is invertible if
  + If then
  + If is an orthogonal matrix then every is such that
  + Similarity transform: is similar to if there exists such a invertible such that . is diagonalizable for diagonal matrix
    - If are similar matrices then
    - Sufficient but not necessary conditions for to be diagonalizable
      * is a normal matrix
      * The eigenvalues of are distinct
* Spectral theorem for symmetric matrices: symmetric matrices are diagonalizable. That is, let , with . Then there exists an orthogonal matrix and such that
* Theorem (singular value decomposition): let . There exists orthogonal matrices and such that , where and
* Jacobi Method
  + Iteratively minimize off-diagonal elements
    1. Find largest off diagonal element
    2. Replace by a zero using similarity transformations: use the Givens/Jacobi Transform for this
  + Offset:
  + Givens/Jacobi Transform for and . Carefully choosing and applying Jacobi similar transform eliminates (and because of symmetry)
    - Thus:
    - Algorithm:
  + Algorithm:
  + Lemma
    - is an orthogonal matrix
      * is an orthogonal matrix
    - Let and let be an orthogonal matrix. Consequently
  + Theorem:

# Linear Programming

* Fundamental theorem of linear programming: Let then either
  + (Bounded feasible) and there exists a vertex of such that
  + (Infeasible region)
  + (Unbounded feasible region) There exists such that for all and
* Problem notation
  + variables and constraints
  + The linear objective function
  + The inequality constraints in standard form
    - Transform to standard form
      * Replace by and
      * Replace by
      * Replace by
      * Replace by
  + The positivity constraints
  + denotes the value of the ith variable
  + a feasible solution if it satisfies the inequality and positivity constraints
* Dictionary notation
  + The slack variables
  + The initial dictionary
  + Valid dictionary if of the variables can be expressed as function of the remaining variables
  + The variables on the left-hand side are the basic variables
  + The variables on the right-hand side are the non-basic variables
  + After row elimination operations, a new basis is obtained
    - Basic variable set and non-basic set , with and
    - Current objective value
    - For each basis set I there is a corresponding dictionary
    - where are coefficients resulting from the row operations
      * For this to be a feasible dictionary, it is required that
      * Basic solution: and
* Simplex method
  + Algorithm
  + How to choose who enters the basis?
    - Mad hatter rule: randomly
    - Dantzig’s 1st rule:
    - Dantzig’s 2st rule: choose that maximizes the increase in z:
    - Bland’s rule: choose the smallest indices and :
      * Degeneracy
        + If any of the basic variables are zero, then the solution is degenerate
        + Theorem: if Bland’s rule is used on all degenerate dictionaries, then the simplex algorithm will not cycle
* Finding an initial feasible dictionary
  + Set up an auxiliary problem
    - Pivot on most infeasible variable in the basis with the most negative value; continue with simplex algorithm until a feasible basis without is found; then, remove column with and replace with (eliminating base variables from z)
* Upper bounds to check progress
  + The LP in standard form
    - total quantity of the resource
    - amount of resource consumed by producing one unit of product
    - profit brought by one unit of product j
    - manufactured quantity of product
  + Find upper bound: find so that for all . For an upper bound as tight as possible: suppose so that and . Consequently:
    - minimum price where it makes sense to sell resource directly rather than manufacturing products
  + Lemma (weak duality): If is a feasible point for () and is a feasible point for () then
    - If () has an unbounded solution, then there exist no feasible point for ()
    - If () has an unbounded solution, then there exist no feasible point for ()
    - if and are primal and dual feasible, respectively, and , then and are the primal and dual optimal points, respectively
  + Theorem (strong duality: checking for optimality): if () or () is feasible, then . Moreover, if is the cost vector of the optimal dictionary of the primal problem () then is the optimal dual solution for
    - Thus the distance to optimal is given by
* Theorem (complementary slackness): a feasible solution is optimal if there exists , an optimal solution for the dual problem, such that
* A small variation of leads to a variation of equal to , where is an optimal solution of ().

# Nonlinear programming

* Minimize a nonlinear differentiable function :
  + This problem is often impossible: first check there exists a minimum; develop iterative methods so that
    - Template method: where is a step size and is search direction. Satisfy the descent condition
* Definition of local minimum: The point is a local minimum of if there exists such that
* Definition of global minimum: The point is a global minimum of if
* Multivariate calculus
  + For a differentiable function , the gradient evaluated at is
    - Note that is a column-vector
  + For any vector valued function define the Jacobian matrix by
    - 1st order Taylor where is a real valued such that
      * Definition of limit: given any constant there exists such that
  + Hessian matrix: if , the Hessian matrix is
    - if then
      * 2nd order Taylor where is a real valued such that
      * Product rule: the vector valued version of the product rule
        + For any function and matrix ,
        + For any two vector valued functions and ,
* How to choose ?
  + Lemma (steepest descent): for the local change of around is . Let . We have subject to
    - Corollary (descent condition): if then there exists such that
  + Theorem (necessary optimality conditions): If is a local minimum of then and (i.e.  is positive definite)
  + Theorem (sufficient optimality conditions): If is such that and (i.e.  is positive semi-definite) then is a local minimum
  + Quadratic functions: , symmetric positive definite
    - Convex: Only has one global minimum which must be the global minimum
  + Convex functions:
  + Theorem: if is a convex function, then every local minimum of is also a global minimum
  + Theorem: if is twice continuously differentiable, then the following three statements are equivalent:
* How to choose ?
  + For quadratic functions, chosen a fixed step size of , then GD converges
  + For non-quadratic functions:
    - (zig-zagging convergence)
    - Backtracking line search algorithm:
    - Gradient descent algorithm:
* Newton’s method
  + Minimizes local quadratic approximation (2nd Taylor)
  + Theorem: let be a -strongly convex function: . If the Hessian is also Lipschitz then the Newton’s method converges according to . In particular, if , then for ,
* Constrained nonlinear optimization
  + Let , for and . Consider the constrained optimization problem:
    - Feasible point : satisfies all constraints
    - Feasible set X: all the feasible points:
    - Abbreviated form:
  + Theorem (existence): if the feasible set is bounded and non-empty, then there exists a solution to
  + Definition: is coercive if
  + Theorem: If is non-empty and is coercive, then there exists a solution to
  + Definition: we say that is an admissible direction at if there exists a differentiable curve such that and . We denote by the set of admissible directions at
    - Lemma: Let be a curve, be continuously differentiable. Then the first order Taylor expansion of the composition around can be written as where
  + Theorem (necessary condition for admissible direction): let be the indexes of saturated inequalities. If is an admissible direction, then:
    - For every we have that
    - For every we have that
    - Let be the set of directions, denominated the cone of feasible directions, that satisfy the above two conditions. Thus
  + Definition: constraint qualifications hold at if for every there exists a sequence such that
  + Theorem (necessary conditions): let be a local minimum. If the constraint qualifications hold at then for every , . Every direction in the feasible cone is not a descent direction
  + Theorem (Lagrange’s condition): let be a local minimum and suppose that the constraint qualifications hold at . It follows that the gradient of the objective is a linear combination of the gradients of constraints at that is, there exists for such that
  + Theorem (Karush, Kuhn and Tuckers condition): let be a local minimum and suppose that the constraint qualifications hold at . It follows that there exists and for and such that
  + Theorem (separating hyperplane theorem): let be two disjoint convex sets. Then there exists a hyperplane defined by and such that
  + Theorem (separating a cone and a point): consider a given vector and the cone . Then either or there exists a vector such that
  + Theorem (sufficient conditions): let and for be convex functions. Let be linear for . Suppose the constraint qualifications hold at and the KKT conditions are verified. Then is a local minimum
* Lagrangian formulation
  + The KKT conditions are often described with the help of an auxiliary function called the Lagrangian function , where and
  + Theorem: let . The KKT conditions hold if: