Simplex method — summary

**Problem**: optimize a linear objective, subject to linear constraints

1. **Step 1**: Convert to standard form:
   - variables on right-hand side, positive constant on left
   - slack variables for \( \leq \) constraints
   - surplus variables for \( \geq \) constraints
   - \( x = x^- - x^+ \) with \( x^-, x^+ \geq 0 \) if \( x \) unrestricted
   - in standard form, all variables \( \geq 0 \), all constraints equalities

2. **Step 2**: Add artificial variables:
   - one for each constraint without a slack variable

3. **Step 3**: Create an objective constraint:
   - add new variable \( z \), and add new constraint \( z - \) objective = 0

4. **Step 4**: Form the initial tableau:
   - first column to identify basic variables
   - last column for constants on right-hand sides of constraints
   - in between, one column for each variable (beginning with \( z \))
   - first row for labels
   - remaining rows for constraints (beginning with objective — but see Step 5 below)

5. **Step 5**: Identify the initial objective function (\( z = ? \))
   - if all constraints were \( \leq \), so slack variable in each constraint, use objective function from original problem
   - if there are artificial variables, and \( M \)-method is being used, objective function is original objective plus
     - large positive multiple of each artificial variable (if minimization problem)
     - large negative multiple of each artificial variable (if maximization problem)
   - if there are artificial variables, and two-phase method is being used, objective function is sum of artificial variables, and this should be minimized (whether or not original problem was minimization)
6. **Step 6**: Identify initial basic variables:
   - slack variables together with artificial variables
   - looking at constraint rows only in columns of these initial basic variables, should see permutation of columns of identity matrix
   - label each constraint row by the basic variable occurring once in that row

7. **Step 7**: Modify the $z$-row
   - if entry in $z$-row in column of basic variable is not zero, add appropriate multiple of the row in which that basic variable appears, so that entry becomes 0
   - at end of process, objective is expressed entirely in terms of non-basic variables
   - if initial basic variables consist of all slack variables, this step not necessary

8. **Step 8**: Identify an entering basic variable and pivot column:
   - maximization problem — most negative coefficient in $z$-row
   - minimization problem — most positive coefficient in $z$-row
   - break ties by choosing left-most column
   - column of entering variable is pivot column
   - if no entering variable, STOP — optimum reached
     - current basic feasible solution is optimal
     - optimal objective value is last entry (solution column) of objective row

9. **Step 9**: Identify a departing basic variable and pivot row:
   - for each non-basic variable, take ratio of entry in solution column and entry in pivot column
   - non-basic variable with smallest non-negative ratio is departing variable, and corresponding row is pivot row
   - break ties by choosing top-most column

10. **Step 10**: Pivot on pivot entry:
    - pivot entry is intersection of pivot row and pivot column
    - scale pivot row so pivot element is one
    - add multiples of pivot row to other rows (including objective row) so rest of pivot column is zero
    - change label of pivot row to that of entering variable
11. **Step 11**: Iterate:

- repeat steps 8 through 10 until optimal is reached
- if using $M$-method or all-slack starting solution, problem is completely done; if using two-phase method, go onto step 12

12. **Step 12**: Phase 2 of two-phase method:

- as long as phase 1 of two-phase method returns minimum of zero, continue to phase 2
- create a new initial tableau
  - objective row given by original objective of problem
  - constraint rows given by constraint rows of final tableau of phase 1, with artificial columns removed
  - initial basic variables given by basic variables at end of phase 1
- go back to step 7
Four quirky situations

1. Degeneracy:
   - Geometric idea
     - can happen when more constraints meet at a point then would be normal (three or more two variable constraints, four or more three variable constraints, etc.)
     - typically one of these constraints is redundant (its removal doesn’t change the feasible space)
     - at the degenerate point, one (or more) of the basic variables is zero, so in fact the same point corresponds to numerous basic feasible solutions
   - Simplex manifestation
     - occurs whenever there is a tie for departing variable
     - at next iteration, entering variable will be constrained to enter at value zero
     - simplex algorithm will move to a new basic feasible solution, but it’s geometrically the same point, and the objective doesn’t change
   - Implications
     - typically, slows down simplex algorithm
     - in worst case, can lead to cycling — algorithm loops, staying at the same (suboptimal) point forever
     - this is so wildly unlikely (and difficult to deal with) that no commercial implementation of simplex algorithm bothers to deal with it
2. **Alternative optima:**

- **Geometric idea**
  - happens when one of the constraints which is satisfied tightly at the optimum is parallel to the objective
  - there is more than one optimal basic feasible solution, and infinitely many optimum solutions that are not basic
- **Simplex manifestation**
  - when optimality is reached, one (or more) of the non-basic variables has coefficient zero in objective
  - each one can enter into the set of basic variables, without changing the objective value
- **Implications**
  - gives a variety of choices for optimum, some of which may be more desirable than others
  - doesn’t affect the running of the algorithm

3. **Unbounded solution:**

- **Geometric idea**
  - happens when the constraints do not trap a finite region in space, but allow at least one variable to go to infinity inside feasible region
  - the objective value can be made as large (or small, if a minimization problem) as one wishes
- **Simplex manifestation**
  - when ratio test is being used to determine constraints on entering variable, all ratios are either negative or infinity
  - the current entering variable is the one that can be made as large as desired
- **Implications**
  - suggests that the original problem may have been poorly posed, or fed into solver incorrectly
  - simplex algorithm should be coded to stop when finding an unbounded variable
4. **Unfeasible problem:**

- **Geometric idea**
  - happens when the constraints are inconsistent
  - there is no feasible point that satisfies all the constraints
  - cannot occur when all constraints are \( \leq \), because all-zero solution (all-slack solution) is feasible in this case

- **Simplex manifestation**
  - occurs only when \( M \)-method or two-phase method are being used
    * \( M \)-method: no matter how large \( M \) is, one of the artificial variables is always basic in optimum solution
    * two-phase method: phase 1 ends by discovering that minimum of sum of artificial variables is positive

- **Implications**
  - suggests that the original problem may have been poorly posed, or fed into solver incorrectly