Numerical Optimization

Todd Munson

Mathematics and Computer Science Division
Argonne National Laboratory

January, 2014

Part I

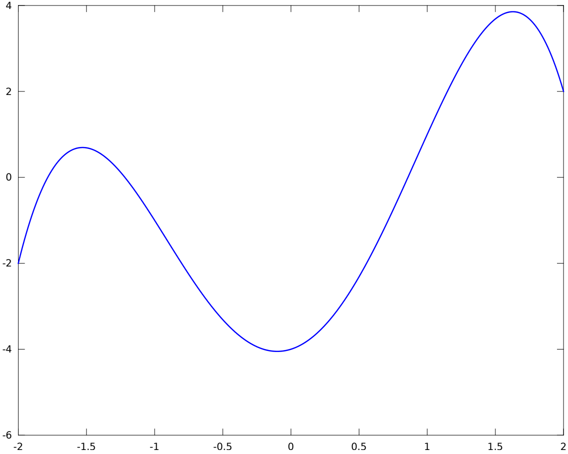
Introduction

Overview of Optimization

- One-dimensional unconstrained optimization
  - Characterization of critical points
  - Basic algorithms
- Nonlinear systems of equations
- Multi-dimensional unconstrained optimization
  - Critical points and their types
  - Computation of local minimizers
- Multi-dimensional constrained optimization
  - Critical points and Lagrange multipliers
  - Second-order sufficiency conditions
  - Globally-convergent algorithms
- Complementarity constraints
  - Stationarity concepts
  - Constraint qualifications
  - Numerical methods

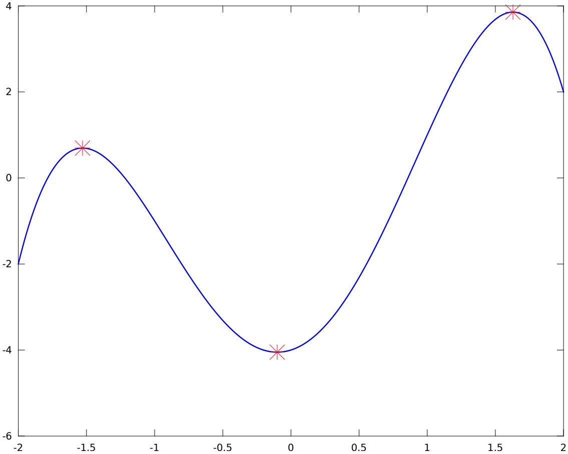
One-dimensional Unconstrained Optimization

$$\min_x f(x) = -x^4 + 5x^2 + x - 4$$



Critical Points

- Stationarity: $\nabla f(x) = -4x^3 + 10x + 1 = 0$
- Local maximizer: $\nabla^2 f(x) = -12x^2 + 10 < 0$
- Local minimizer: $\nabla^2 f(x) = -12x^2 + 10 > 0$



Locally Convergent Newton Method

- All good algorithms are variants of Newton's method
- Compute stationary points: $F(x) = \nabla f(x) = 0$
  - Form Taylor series approximation around x^k

$$F(x) \approx \nabla F(x^k)(x - x^k) + F(x^k)$$

  - Solve for x and iterate

$$x^{k+1} = x^k - \frac{F(x^k)}{\nabla F(x^k)}$$

- Several possible outcomes
  - Convergence: $\lim_{k\to\infty} x^k = x^*$
  - Divergence: $\lim_{k\to\infty} \|(f(x^k), x^k)\| \to \infty$
  - Sequence cycles
    - Multiple convergent subsequences (limit points)
    - Limit points are not solutions

Illustration of Newton's Method

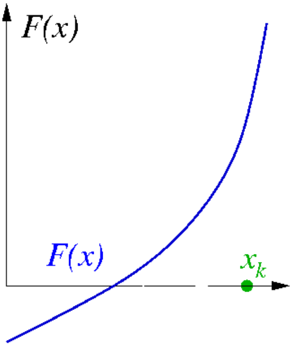


Illustration of Newton's Method

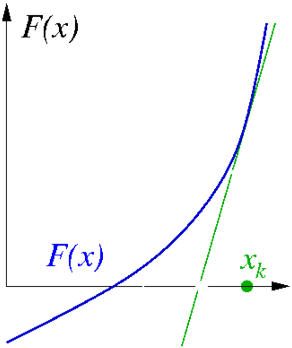


Illustration of Newton's Method

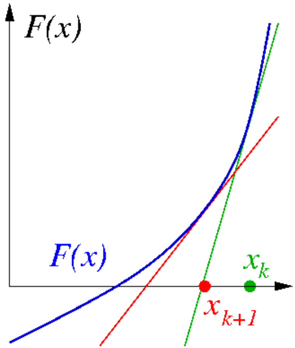


Illustration of Newton's Method

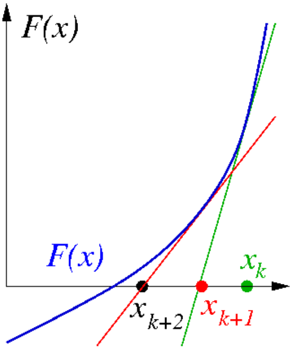


Illustration of Divergence

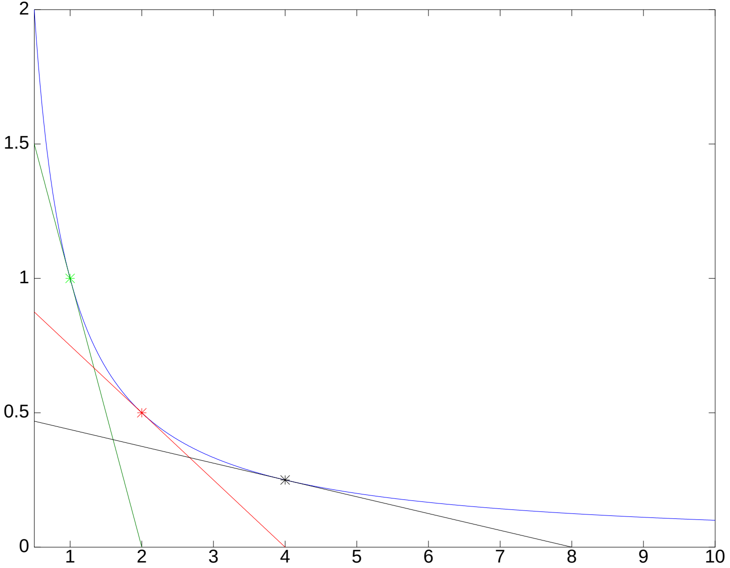
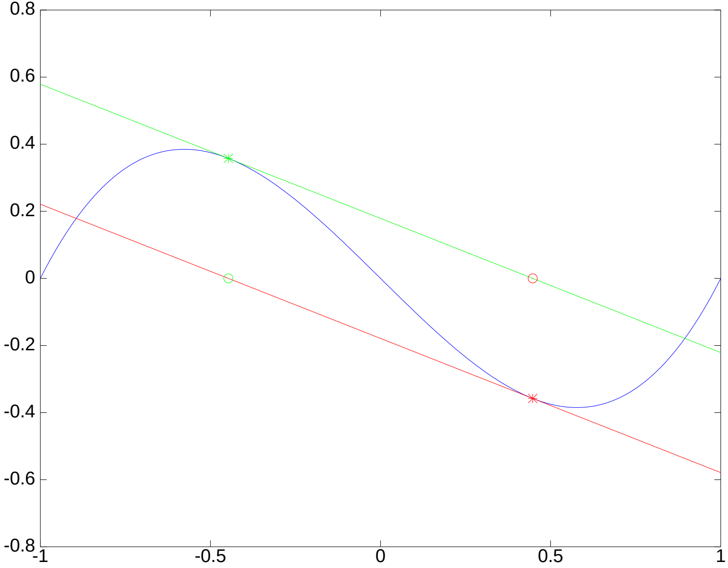


Illustration of Cycling



Globally Convergent Newton Method

- Use Newton method to compute a direction

$$s^k = -\frac{F(x^k)}{\nabla F(x^k)}$$

- Check direction for descent using objective function

$$\nabla f(x^k) s^k < 0$$

- Determine an appropriate stepsize
  - Line search to minimize the objective function

$$f(x^k + ts^k) \leq f(x^k) + \sigma t \nabla f(x^k) s^k$$

  - Trust region around the approximation
- Iterate until convergence
- Two possible outcomes
  - Convergence: $\lim_{k \to \infty} x^k \to x^*$
  - Divergence: $\lim_{k \to \infty} \|(f(x^k), x^k)\| \to \infty$

Newton Method for Square Systems of Equations

- Given $F : \Re^n \to \Re^n$, compute x such that

$$F(x) = 0$$

- First-order Taylor series approximation

$$\nabla F(x^k)(x - x^k) + F(x^k) = 0$$

- Solve linear system of equations

$$x^{k+1} = x^k - \nabla F(x^k)^{-1} F(x^k)$$

  - Direct method – compute factorization
  - Iterative method – use Krylov subspace
- Method has local (fast) convergence under suitable conditions
  - If x^k is near a solution, method converges to a solution x^*
  - The distance to the solution decreases quickly; ideally,

$$\|x^{k+1} - x^*\| \le c\|x^k - x^*\|^2$$

Globally Convergent Newton Method

- Solve linear system of equations

$$\nabla F(x^k) s_k = -F(x^k)$$

- Determine step length by minimizing merit function

$$t_k \in \arg \min_{t \in (0,1]} \|F(x^k + t s_k)\|_2^2$$

- Update iterate

$$x^{k+1} = x^k + t_k s_k$$

Globalized Newton Method with Proximal Perturbation

- Solve linear system of equations

$$(\nabla F(x^k) + \lambda_k I)s_k = -F(x^k)$$

- Check step and possibly use steepest descent direction
- Determine step length

$$t_k \in \arg\min_{t\in(0,1]} \|F(x^k + ts_k)\|_2^2$$

- Update iterate

$$x^{k+1} = x^k + t_k s_k$$

- Update perturbation

Nonsquare Nonlinear Systems of Equations

- Given $F : \Re^n \to \Re^m$, compute x such that

$$F(x) = 0$$

- System is underdetermined if $m < n$
  - More variables than constraints
  - Solution typically not unique
  - Need to select one solution

$$\min_x \; \|x\|_2 \text{ subject to } F(x) = 0$$

- System is overdetermined if $m > n$
  - More constraints than variables
  - Solution typically does not exist
  - Need to select approximate solution

$$\min_x \; \|F(x)\|_2$$

- System is square if $m = n$
  - Jacobian has full rank then solution is unique
  - If Jacobian is rank deficient then
    - Underdetermined when compatible
    - Overdetermined when incompatible

Part II

Unconstrained Optimization

Model Formulation

- Classify m people into two groups using v variables
  - $c \in \{0, 1\}^m$ is the known classification
  - $d \in \Re^{m \times v}$ are the observations
  - $\beta \in \Re^{v+1}$ defines the separator
  - logit distribution function
- Maximum likelihood problem

$$\max_{\beta} \quad \sum_{i=1}^{m} c_i \log(f(\beta, d_{i,\cdot})) + (1 - c_i) \log(1 - f(\beta, d_{i,\cdot}))$$

where

$$f(\beta, x) = \frac{\exp\left(\beta_0 + \sum_{j=1}^{v} \beta_j x_j\right)}{1 + \exp\left(\beta_0 + \sum_{j=1}^{v} \beta_j x_j\right)}$$

Model Formulation

- Classify m people into two groups using v variables
  - $c \in \{0,1\}^m$ is the known classification
  - $d \in \Re^{m \times v}$ are the observations
  - $\beta \in \Re^{v+1}$ defines the separator
  - logit distribution function
- Maximum likelihood problem

$$\min_{\beta} \quad -\left(\sum_{i=1}^{m} c_i \log(f(\beta, d_{i,\cdot})) + (1-c_i)\log(1 - f(\beta, d_{i,\cdot})) \right)$$

where

$$f(\beta, x) = \frac{\exp\left(\beta_0 + \sum_{j=1}^{v} \beta_j x_j\right)}{1 + \exp\left(\beta_0 + \sum_{j=1}^{v} \beta_j x_j\right)}$$

Basic Theory

$$\min_x f(x)$$

- Convex functions – local minimizers are global minimizers
- Nonconvex functions
  - Stationarity: $\nabla f(x) = 0$
  - Local minimizer: $\nabla^2 f(x)$ is positive definite (min eig positive)
  - Local maximizer: $\nabla^2 f(x)$ is negative definite (max eig negative)

Solution Techniques

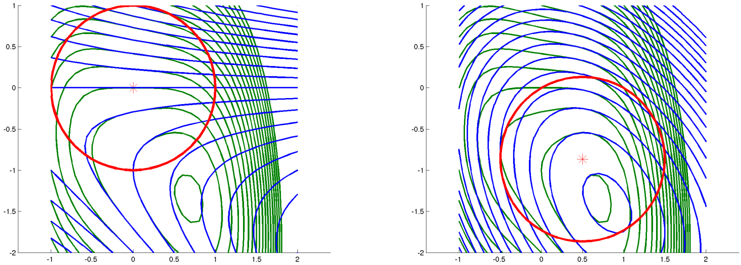
$$\min_x f(x)$$

Main ingredients of solution approaches:

- Local method: given x_k (solution guess) compute a step s^k
  - Gradient Descent
  - Quasi-Newton Approximation
  - Sequential Quadratic Programming
- Globalization strategy: converge from any starting point
  - Trust region
  - Line search

Trust-Region Method

$$
\begin{array}{ll}
\min\limits_{s} & f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T H(x_k) s \\
\text{subject to} & \|s\|_2 \leq \Delta_k
\end{array}
$$



Trust-Region Method

1. Initialize trust-region radius
2. Compute a new iterate
   2.1 Solve trust-region subproblem

$$
\begin{array}{ll}
\min_s & f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T H(x_k) s \\
\text{subject to} & \|s\|_2 \leq \Delta_k
\end{array}
$$

Trust-Region Method

1. Initialize trust-region radius
2. Compute a new iterate

   2.1 Solve trust-region subproblem

   $$\begin{array}{ll} \min_s & f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T H(x_k) s \\ \text{subject to} & \|s\|_2 \leq \Delta_k \end{array}$$

   2.2 Accept or reject iterate

   2.3 Update trust-region radius

   - Increase if actual reduction more than predicted
   - Decrease if actual reduction less than predicted

3. Check convergence

Solving a Convex Quadratic Program

- Assume the quadratic program is strictly convex

$$\min_s \quad \frac{1}{2} s^T H s + c^T s$$

  - H is symmetric and positive definite
  - H^{-1} exists
- Stationary points are necessary and sufficient

$$Hs = -c$$

  - Cholesky factorization
    - Compute (sparse) lower triangular matrix with $H = LL^T$
    - Solve $s = L^{-T}(L^{-1}c)$ exploiting lower triangular property
  - Conjugate gradient method
    - Iteratively compute a set of H conjugate directions
    - Analytically minimize quadratic along the directions
    - Objective function decreases monotonically
    - Guaranteed convergence in n steps

Solving a Nonconvex Quadratic Program

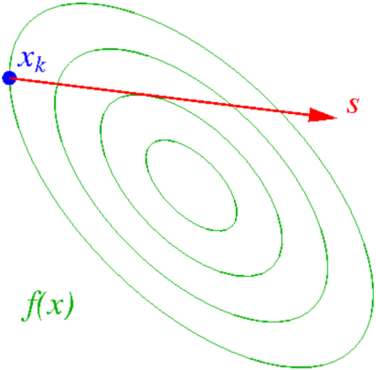
- No assumptions on the quadratic program

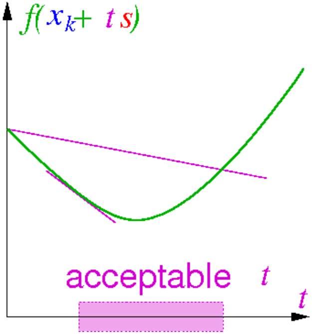
$$\begin{array}{ll} \min\limits_{s} & \frac{1}{2} s^T H s + c^T s \\ \text{subject to} & \|s\|_2 \leq \Delta_k \end{array}$$

  - Trust region bounds objective function
  - No unbounded solutions
- Can detect inertia with a LDL^T factorization and use direct method
- Global solutions computed with Moré-Sorensen method
  - Requires repeated factorization of a matrix
  - Can be expensive to calculate
  - Little benefit
- Conjugate gradient method with a trust region
  - Iteratively compute a set of H conjugate directions
  - Analytically minimize quadratic along the directions
  - Stop when trust region boundary is encountered
  - Objective function decreases monotonically

Line-Search Method

$$\min_{s} \quad f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T (H(x_k) + \lambda_k I) s$$





Line-Search Method

1. Initialize perturbation to zero
2. Solve perturbed quadratic model

$$\min_{s} \quad f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T (H(x_k) + \lambda_k I) s$$

Line-Search Method

1. Initialize perturbation to zero
2. Solve perturbed quadratic model

$$\min_{s} \quad f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T (H(x_k) + \lambda_k I) s$$

3. Find new iterate
   3.1 Search along Newton direction
   3.2 Search along gradient-based direction

Line-Search Method

1. Initialize perturbation to zero
2. Solve perturbed quadratic model

$$\min_{s} \quad f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T (H(x_k) + \lambda_k I) s$$

3. Find new iterate
   3.1 Search along Newton direction
   3.2 Search along gradient-based direction
4. Update perturbation
   - Decrease perturbation if the following hold
     - Iterative method succeeds
     - Search along Newton direction succeeds
   - Otherwise increase perturbation
5. Check convergence

Solving the Subproblem

- Use LDL^T to determine inertia and update perturbation
- Apply conjugate gradient method and stop on unbounded directions

Solving the Subproblem

- Use LDL^T to determine inertia and update perturbation
- Apply conjugate gradient method and stop on unbounded directions
- Conjugate gradient method with trust region
  - Initialize radius
  - Update radius

Performing the Line Search

- Backtracking Armijo line search
  - Find t to satisfy sufficient decrease condition

$$f(x_k + ts) \leq f(x_k) + \sigma t \nabla f(x_k)^T s$$

  - Try $t = 1, \beta, \beta^2, \ldots$ for $0 < \beta < 1$
- More-Thuente line search
  - Find t to satisfy strong Wolfe conditions

$$\begin{aligned} f(x_k + ts) &\leq f(x_k) + \sigma t \nabla f(x_k)^T s \\ |\nabla f(x_k + ts)^T s| &\leq \delta |\nabla f(x_k)^T s| \end{aligned}$$

  - Construct cubic interpolant
  - Compute t to minimize interpolant
  - Refine interpolant

Updating the Perturbation

1. If increasing and $\lambda_k = 0$
$$\lambda_{k+1} = \mathrm{Proj}_{[\ell_0, u_0]} \left(\alpha_0 \|\nabla f(x^k)\| \right)$$
2. If increasing and $\lambda_k > 0$
$$\lambda_{k+1} = \mathrm{Proj}_{[\ell_i, u_i]} \left(\max \left(\alpha_i \|\nabla f(x^k)\|, \beta_i \lambda_k \right) \right)$$
3. If decreasing
$$\lambda_{k+1} = \min \left(\alpha_d \|\nabla f(x^k)\|, \beta_d \lambda_k \right)$$
4. If $\lambda_{k+1} < \ell_d$, then $\lambda_{k+1} = 0$

Trust-Region Line-Search Method

1. Initialize trust-region radius
2. Compute a new iterate
   2.1 Solve trust-region subproblem

$$\begin{array}{ll} \min_s & f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T H(x_k) s \\ \text{subject to} & \|s\| \leq \Delta_k \end{array}$$

   2.2 Search along direction
   2.3 Update trust-region radius
3. Check convergence

Iterative Methods

- Conjugate gradient method
  - Stop if negative curvature encountered
  - Stop if residual norm is small

Iterative Methods

- Conjugate gradient method
  - Stop if negative curvature encountered
  - Stop if residual norm is small
- Conjugate gradient method with trust region
  - Nash
    - Follow direction to boundary if first iteration
    - Stop at base of direction otherwise
  - Steihaug-Toint
    - Follow direction to boundary
  - Generalized Lanczos
    - Compute tridiagonal approximation
    - Find global solution to approximate problem on boundary
    - Initialize perturbation with approximate minimum eigenvalue

Preconditioners to Improve Performance

- Modify system of equations solved

$$AHs = -Ac$$

  - A is symmetric positive definite
  - A^{-1} can be easily applied to vector
  - AH is well conditioned or has clustered eigenvalues
  - Corresponds to changing to an elliptic trust region

$$\begin{array}{ll} \min_s & \frac{1}{2} s^T H s + c^T s \\ \text{subject to} & \|s\|_A \leq \Delta_k \end{array}$$

- Preconditioners are problem specific
- Many possibly preconditioners
  - No preconditioner – $A = I$
  - Diagonal of Hessian – $A = |\text{diag}(H(x_k))|$
  - Diagonal of perturbed Hessian – $A = |\text{diag}(H(x_k) + \lambda_k I)|$
  - Quasi-newton approximation to Hessian matrix
  - Incomplete Cholesky factorization of Hessian
  - Block Jacobi with Cholesky factorization of blocks

Termination

- Typical convergence criteria
  - Absolute residual $\|\nabla f(x_k)\| < \tau_a$
  - Relative residual $\frac{\|\nabla f(x_k)\|}{\|\nabla f(x_0)\|} < \tau_r$
  - Unbounded objective $f(x_k) < \kappa$
  - Slow progress $|f(x_k) - f(x_{k-1})| < \epsilon$
  - Iteration limit
  - Time limit
- Check the solver status

Convergence Issues

- Quadratic convergence – best outcome
- Linear convergence
  - Far from a solution – $\|\nabla f(x_k)\|$ is large
  - Hessian is incorrect – disrupts quadratic convergence
  - Hessian is rank deficient – $\|\nabla f(x_k)\|$ is small
  - Limits of finite precision arithmetic
    1. $\|\nabla f(x_k)\|$ converges quadratically to small number
    2. $\|\nabla f(x_k)\|$ hovers around that number with no progress
- Domain violations such as $\frac{1}{x}$ when $x = 0$
  - Make implicit constraints explicit
- Nonglobal solution
  - Apply a multistart heuristic
  - Use global optimization solver

Some Available Software

- TRON – Newton method with trust-region
- LBFGS – Limited-memory quasi-Newton method with line search
- TAO – Toolkit for Advanced Optimization
  - NLS – Newton line-search method
  - NTR – Newton trust-region method
  - NTL – Newton line-search/trust-region method
  - LMVM – Limited-memory quasi-Newton method
  - CG – Nonlinear conjugate gradient methods

Optimization without Derivatives

- Methods of last resort
- Limited to small problems (up to 100 variables)
- Derivatives exist but are not computed

Optimization without Derivatives

- Methods of last resort
- Limited to small problems (up to 100 variables)
- Derivatives exist but are not computed
- Nelder-Mead
  - Initialize a simplex
  - Update simplex based on function values
    - Increase size of the simplex
    - Reduce size of the simplex
    - Reflection of the simplex
  - Convergent methods can be constructed
  - Matlab and numerical recipes versions do not have guarantees

Optimization without Derivatives

- Methods of last resort
- Limited to small problems (up to 100 variables)
- Derivatives exist but are not computed
- Nelder-Mead
  - Initialize a simplex
  - Update simplex based on function values
    - Increase size of the simplex
    - Reduce size of the simplex
    - Reflection of the simplex
  - Convergent methods can be constructed
  - Matlab and numerical recipes versions do not have guarantees
- Pattern search methods
  - Evaluate objective along coordinate directions
  - If no improvement shrink the length
  - Show convergence

Optimization without Derivatives

- Methods of last resort
- Limited to small problems (up to 100 variables)
- Derivatives exist but are not computed
- Nelder-Mead
  - Initialize a simplex
  - Update simplex based on function values
    - Increase size of the simplex
    - Reduce size of the simplex
    - Reflection of the simplex
  - Convergent methods can be constructed
  - Matlab and numerical recipes versions do not have guarantees
- Pattern search methods
  - Evaluate objective along coordinate directions
  - If no improvement shrink the length
  - Show convergence
- Model based methods
  - Sample the objective at various points
  - Construct a model based on the samples
  - Compute optimal solution for the model
  - Update the set of sample points
  - Show convergence

Part III

Constrained Optimization

Social Planning Model

- Economy with n agents and m commodities
  - $e \in \Re^{n \times m}$ are the endowments
  - $\alpha \in \Re^{n \times m}$ and $\beta \in \Re^{n \times m}$ are the utility parameters
  - $\lambda \in \Re^{n}$ are the social weights
- Social planning problem

$$
\begin{array}{lll}
\max\limits_{x \geq 0} & \displaystyle\sum_{i=1}^{n} \lambda_i \left(\sum_{k=1}^{m} \frac{\alpha_{i,k} (1 + x_{i,k})^{1-\beta_{i,k}}}{1 - \beta_{i,k}} \right) & \\
\text{subject to} & \displaystyle\sum_{i=1}^{n} x_{i,k} \leq \sum_{i=1}^{n} e_{i,k} & \forall k = 1, \ldots, m
\end{array}
$$

Life-Cycle Saving Model

- Maximize discounted utility
  - $u(\cdot)$ is the utility function
  - R is the retirement age
  - T is the terminal age
  - w is the wage
  - β is the discount factor
  - r is the interest rate
- Optimization problem

$$
\begin{array}{lll}
\max\limits_{s,c} & \sum\limits_{t=0}^{T} \beta^t u(c_t) & \\
\text{subject to} & s_{t+1} = (1+r)s_t + w - c_t & t = 0, \ldots, R-1 \\
 & s_{t+1} = (1+r)s_t - c_t & t = R, \ldots, T \\
 & s_0 = s_{T+1} = 0 &
\end{array}
$$

Theory Revisited

- Strict descent direction d

$$\nabla f(x)^T d < 0$$

- Stationarity conditions (first-order conditions)
  - No feasible, strict descent directions
  - For all feasible directions d

$$\nabla f(x)^T d \geq 0$$

  - Unconstrained case, $d \in \Re^n$ and

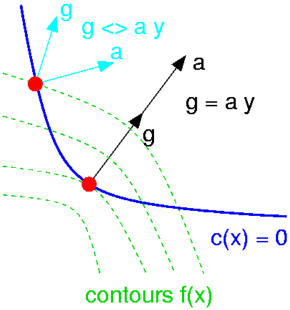
$$\nabla f(x) = 0$$

  - Constrained cases
    - Characterize superset of feasible directions
    - Requires constraint qualification

Convergence Criteria

$$\begin{array}{ll} \min\limits_{x} & f(x) \\ \text{subject to} & c(x) \geq 0 \end{array}$$

- Feasible and no strict descent directions
  - Constraint qualification – LICQ, MFCQ
  - Linearized active constraints characterize directions
  - Objective gradient is a linear combination of constraint gradients



Optimality Conditions

- If x^* is a local minimizer and a constraint qualification holds, then there exist multipliers $\lambda^* \geq 0$ such that

$$\nabla f(x^*) - \nabla c_{\mathcal{A}}(x^*)^T \lambda^*_{\mathcal{A}} = 0$$

  - Lagrangian function $\mathcal{L}(x, \lambda) := f(x) - \lambda^T c(x)$
- Optimality conditions can be written as

$$\begin{aligned} &\nabla f(x) - \nabla c(x)^T \lambda = 0 \\ &0 \leq \lambda \quad \perp \quad c(x) \geq 0 \end{aligned}$$

  - Complementarity problem

Solving Constrained Optimization Problems

Main ingredients of solution approaches:

- Local method: given x_k (solution guess) find a step s.
  - Sequential Quadratic Programming (SQP)
  - Sequential Linear/Quadratic Programming (SLQP)
  - Interior-Point Method (IPM)
- Globalization strategy: converge from any starting point.
  - Trust region
  - Line search
- Acceptance criteria: filter or penalty function.

Sequential Linear Programming

1. Initialize trust-region radius
2. Compute a new iterate

Sequential Linear Programming

1. Initialize trust-region radius
2. Compute a new iterate
    2.1 Solve linear program

$$\begin{array}{ll} \min\limits_{s} & f(x_k) + s^T \nabla f(x_k) \\ \text{subject to} & c(x_k) + \nabla c(x_k)^T s \geq 0 \\ & \|s\| \leq \Delta_k \end{array}$$

Sequential Linear Programming

1. Initialize trust-region radius
2. Compute a new iterate
   2.1 Solve linear program

$$\begin{array}{ll} \displaystyle\min_{s} & f(x_k) + s^T \nabla f(x_k) \\ \text{subject to} & c(x_k) + \nabla c(x_k)^T s \geq 0 \\ & \|s\| \leq \Delta_k \end{array}$$

   2.2 Accept or reject iterate
   2.3 Update trust-region radius
3. Check convergence

Sequential Quadratic Programming

1. Initialize trust-region radius
2. Compute a new iterate

Sequential Quadratic Programming

1. Initialize trust-region radius
2. Compute a new iterate
   2.1 Solve quadratic program

$$\begin{array}{ll} \min\limits_{s} & f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T W(x_k) s \\ \text{subject to} & c(x_k) + \nabla c(x_k)^T s \geq 0 \\ & \|s\| \leq \Delta_k \end{array}$$

Sequential Quadratic Programming

1. Initialize trust-region radius
2. Compute a new iterate
    2.1 Solve quadratic program

$$\begin{array}{ll} \min\limits_{s} & f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T W(x_k) s \\ \text{subject to} & c(x_k) + \nabla c(x_k)^T s \geq 0 \\ & \|s\| \leq \Delta_k \end{array}$$

    2.2 Accept or reject iterate
    2.3 Update trust-region radius
3. Check convergence

Sequential Linear Quadratic Programming

1. Initialize trust-region radius
2. Compute a new iterate

Sequential Linear Quadratic Programming

1. Initialize trust-region radius
2. Compute a new iterate
    2.1 Solve linear program to predict active set

$$
\begin{array}{ll}
\min\limits_{d} & f(x_k) + d^T \nabla f(x_k) \\
\text{subject to} & c(x_k) + \nabla c(x_k)^T d \geq 0 \\
 & \|d\| \leq \Delta_k
\end{array}
$$

Sequential Linear Quadratic Programming

1. Initialize trust-region radius
2. Compute a new iterate
   2.1 Solve linear program to predict active set

$$\begin{array}{ll} \min\limits_{d} & f(x_k) + d^T \nabla f(x_k) \\ \text{subject to} & c(x_k) + \nabla c(x_k)^T d \geq 0 \\ & \|d\| \leq \Delta_k \end{array}$$

   2.2 Solve equality constrained quadratic program

$$\begin{array}{ll} \min\limits_{s} & f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T W(x_k) s \\ \text{subject to} & c_{\mathcal{A}}(x_k) + \nabla c_{\mathcal{A}}(x_k)^T s = 0 \end{array}$$

   2.3 Accept or reject iterate
   2.4 Update trust-region radius
3. Check convergence

Acceptance Criteria

- Decrease objective function value: $f(x_k + s) \leq f(x_k)$
- Decrease constraint violation: $\|c_-(x_k + s)\| \leq \|c_-(x_k)\|$

Acceptance Criteria

- Decrease objective function value: $f(x_k + s) \leq f(x_k)$
- Decrease constraint violation: $\|c_-(x_k + s)\| \leq \|c_-(x_k)\|$
- Four possibilities
  1. step can decrease both $f(x)$ and $\|c_-(x)\|$ GOOD
  2. step can decrease $f(x)$ and increase $\|c_-(x)\|$???
  3. step can increase $f(x)$ and decrease $\|c_-(x)\|$???
  4. step can increase both $f(x)$ and $\|c_-(x)\|$ BAD

Acceptance Criteria

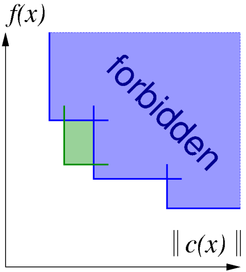
- Decrease objective function value: $f(x_k + s) \leq f(x_k)$
- Decrease constraint violation: $\|c_-(x_k + s)\| \leq \|c_-(x_k)\|$
- Four possibilities
  1. step can decrease both $f(x)$ and $\|c_-(x)\|$ GOOD
  2. step can decrease $f(x)$ and increase $\|c_-(x)\|$???
  3. step can increase $f(x)$ and decrease $\|c_-(x)\|$???
  4. step can increase both $f(x)$ and $\|c_-(x)\|$ BAD
- Filter uses concept from multi-objective optimization

(h_{k+1}, f_{k+1}) dominates (h_ℓ, f_ℓ) iff $h_{k+1} \leq h_\ell$ and $f_{k+1} \leq f_\ell$

Filter Framework

Filter $\mathcal{F}$: list of non-dominated pairs (h_ℓ, f_ℓ)

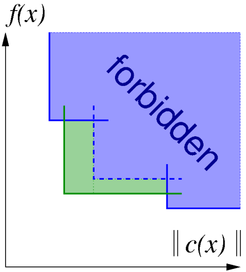
- new x_{k+1} is acceptable to filter $\mathcal{F}$ iff for all $\ell \in \mathcal{F}$
  1. $h_{k+1} \leq h_\ell$ or
  2. $f_{k+1} \leq f_\ell$



Filter Framework

Filter $\mathcal{F}$: list of non-dominated pairs (h_ℓ, f_ℓ)

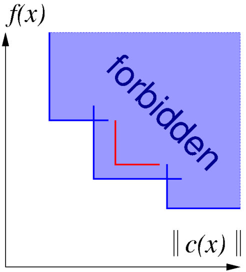
- new x_{k+1} is acceptable to filter $\mathcal{F}$ iff for all $\ell \in \mathcal{F}$
  1. $h_{k+1} \leq h_\ell$ or
  2. $f_{k+1} \leq f_\ell$
- remove redundant filter entries



Filter Framework

Filter $\mathcal{F}$: list of non-dominated pairs (h_ℓ, f_ℓ)

- new x_{k+1} is acceptable to filter $\mathcal{F}$ iff for all $\ell \in \mathcal{F}$
  1. $h_{k+1} \le h_\ell$ or
  2. $f_{k+1} \le f_\ell$
- remove redundant filter entries
- new x_{k+1} is rejected if for some $\ell \in \mathcal{F}$
  1. $h_{k+1} > h_\ell$ and
  2. $f_{k+1} > f_\ell$



Termination

- Feasible and complementary $\|\min(c(x_k), \lambda_k)\| \leq \tau_f$
- Optimal $\|\nabla_x \mathcal{L}(x_k, \lambda_k)\| \leq \tau_o$
- Other possible conditions
  - Slow progress
  - Iteration limit
  - Time limit
- Multipliers and reduced costs

Convergence Issues

- Quadratic convergence – best outcome
- Globally infeasible – linear constraints infeasible
- Locally infeasible – nonlinear constraints locally infeasible
- Unbounded objective – hard to detect
- Unbounded multipliers – constraint qualification not satisfied
- Linear convergence rate
  - Far from a solution – $\|\nabla f(x_k)\|$ is large
  - Hessian is incorrect – disrupts quadratic convergence
  - Hessian is rank deficient – $\|\nabla f(x_k)\|$ is small
  - Limits of finite precision arithmetic
- Domain violations such as $\frac{1}{x}$ when $x = 0$
  - Make implicit constraints explicit
- Nonglobal solutions
  - Apply a multistart heuristic
  - Use global optimization solver

Some Available Software

- filterSQP
  - trust-region SQP; robust QP solver
  - filter to promote global convergence
- SNOPT
  - line-search SQP; null-space CG option
  - ℓ_1 exact penalty function
- SLIQUE – part of KNITRO
  - SLP-EQP
  - trust-region with ℓ_1 penalty
  - use with `knitro_options = "algorithm=3";`

Interior-Point Method

- Reformulate optimization problem with slacks

$$
\begin{array}{ll}
\min\limits_{x} & f(x) \\
\text{subject to} & c(x) = 0 \\
& x \geq 0
\end{array}
$$

- Construct perturbed optimality conditions

$$
F_\tau(x, y, z) = \begin{bmatrix} \nabla f(x) - \nabla c(x)^T \lambda - \mu \\ c(x) \\ X\mu - \tau e \end{bmatrix}
$$

- Central path $\{x(\tau), \lambda(\tau), \mu(\tau) \mid \tau > 0\}$
- Apply Newton's method for sequence $\tau \searrow 0$

Interior-Point Method

1. Compute a new iterate
   1.1 Solve linear system of equations

$$\left[\begin{array}{ccc} W_k & -\nabla c(x_k)^T & -I \\ \nabla c(x_k) & 0 & 0 \\ \mu_k & 0 & X_k \end{array}\right] \left(\begin{array}{c} s_x \\ s_\lambda \\ s_\mu \end{array}\right) = -F_\tau(x_k, \lambda_k, \mu_k)$$

   1.2 Accept or reject iterate
   1.3 Update parameters
2. Check convergence

Convergence Issues

- Quadratic convergence – best outcome
- Globally infeasible – linear constraints infeasible
- Locally infeasible – nonlinear constraints locally infeasible
- Dual infeasible – dual problem is locally infeasible
- Unbounded objective – hard to detect
- Unbounded multipliers – constraint qualification not satisfied
- Duality gap
- Domain violations such as $\frac{1}{x}$ when $x = 0$
  - Make implicit constraints explicit
- Nonglobal solutions
  - Apply a multistart heuristic
  - Use global optimization solver

Termination

- Feasible and complementary $\| \min(c(x_k), \lambda_k) \| \leq \tau_f$
- Optimal $\|\nabla_x \mathcal{L}(x_k, \lambda_k)\| \leq \tau_o$
- Other possible conditions
  - Slow progress
  - Iteration limit
  - Time limit
- Multipliers and reduced costs

Some Available Software

- IPOPT – open source in COIN-OR
  - line-search filter algorithm
- KNITRO
  - trust-region Newton to solve barrier problem
  - ℓ_1 penalty barrier function
  - Newton system: direct solves or null-space CG
- LOQO
  - line-search method
  - Newton system: modified Cholesky factorization

Part IV

Optimal Control

Optimal Technology

Optimize energy production schedule and transition between old and new reduced-carbon technology to meet carbon targets

- Maximize social welfare
- Constraints
  - Limit total greenhouse gas emissions
  - Low-carbon technology less costly as it becomes widespread
- Assumptions on emission rates, economic growth, and energy costs

Model Formulation

- Finite time: $t \in [0, T]$
- Instantaneous energy output: $q^o(t)$ and $q^n(t)$
- Cumulative energy output: $x^o(t)$ and $x^n(t)$

$$x^n(t) = \int_0^t q^n(\tau) d\tau$$

- Discounted greenhouse gases emissions

$$\int_0^T e^{-at} \left(b_o q^o(t) + b_n q^n(t) \right) dt \leq z_T$$

- Consumer surplus $S(Q(t), t)$ derived from utility
- Production costs
  - c_o per unit cost of old technology
  - $c_n(x^n(t))$ per unit cost of new technology (learning by doing)

Continuous-Time Model

$$\max_{\{q^o, q^n, x^n, z\}(t)} \quad \int_0^T e^{-rt} \left[S(q^o(t) + q^n(t), t) - c_o q^o(t) - c_n(x^n(t)) q^n(t) \right] dt$$

$$\text{subject to} \quad \dot{x}^n(t) = q^n(t) \quad x(0) = x_0 = 0$$

$$\dot{z}(t) = e^{-at} \left(b_o q^o(t) + b_n q^n(t) \right) \quad z(0) = z_0 = 0$$

$$z(T) \leq z_T$$

$$q^o(t) \geq 0, \quad q^n(t) \geq 0.$$

Optimal Technology Penetration

Discretization:

- $t \in [0, T]$ replaced by $N + 1$ equally spaced points $t_i = ih$
- $h := T/N$ time integration step-length
- approximate $q_i^n \simeq q^n(t_i)$ etc.

Replace differential equation

$$\dot{x}(t) = q^n(t)$$

by

$$x_{i+1} = x_i + hq_i^n$$

Optimal Technology Penetration

Discretization:

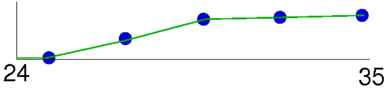
- $t \in [0, T]$ replaced by $N + 1$ equally spaced points $t_i = ih$
- $h := T/N$ time integration step-length
- approximate $q_i^n \simeq q^n(t_i)$ etc.

Replace differential equation

$$\dot{x}(t) = q^n(t)$$

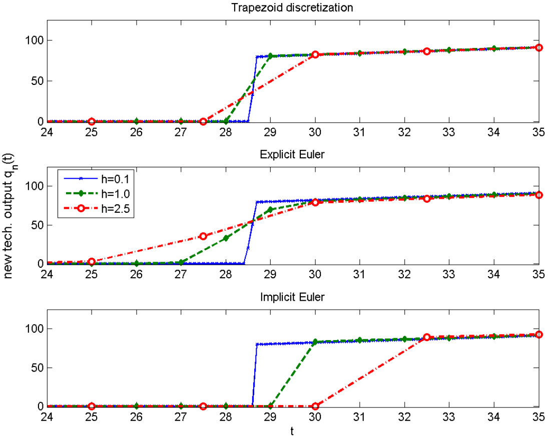
by

$$x_{i+1} = x_i + hq_i^n$$



Output of new technology between $t = 24$ and $t = 35$

Solution with Varying h



Output for different discretization schemes and step-sizes

Optimal Technology Penetration

Add adjustment cost to model building of capacity:

Capital and Investment:

- $K^j(t)$ amount of capital in technology j at t.
- $I^j(t)$ investment to increase $K^j(t)$.
- initial capital level as $\bar{K}_0^j$:

Notation:

- $Q(t) = q^o(t) + q^n(t)$
- $C(t) = C^o(q^o(t), K^o(t)) + C^n(q^n(t), K^n(t))$
- $I(t) = I^o(t) + I^n(t)$
- $K(t) = K^o(t) + K^n(t)$

Optimal Technology Penetration

$$\underset{\{q^j, K^j, I^j, x, z\}(t)}{\text{maximize}} \quad \left\{ \int_0^T e^{-rt} \left[\tilde{S}(Q(t), t) - C(t) - K(t) \right] dt + e^{-rT} K(T) \right\}$$

$$\text{subject to} \quad \dot{x}(t) = q^n(t), \quad x(0) = x_0 = 0$$

$$\dot{K}^j(t) = -\delta K^j(t) + I^j(t), \quad K^j(0) = \bar{K}_0^j, \quad j \in \{o, n\}$$

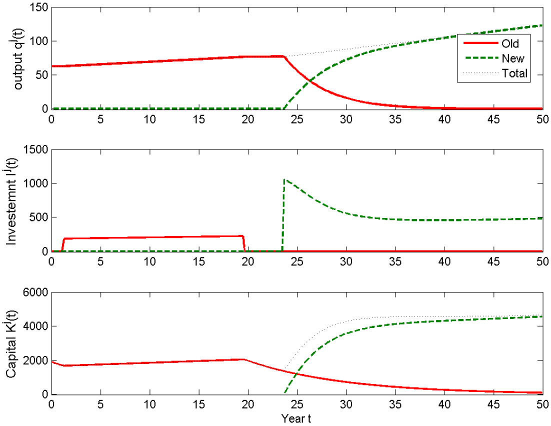
$$\dot{z}(t) = e^{-at}[b_o q^o(t) + b_n q^n(t)], \quad z(0) = z_0 = 0$$

$$z(T) \leq z_T$$

$$q^j(t) \geq 0, \ j \in \{o, n\}$$

$$I^j(t) \geq 0, \ j \in \{o, n\}$$

Optimal Technology Penetration



Optimal output, investment, and capital for 50% CO2 reduction.

Pitfalls of Discretizations [Hager, 2000]

Optimal Control Problem

$$\text{minimize } \frac{1}{2}\int_0^1 u^2(t) + 2y^2(t)dt$$

subject to

$$\begin{aligned} \dot{y}(t) &= \frac{1}{2}y(t) + u(t), \; t \in [0,1], \\ y(0) &= 1. \end{aligned}$$

$$\begin{aligned} \Rightarrow y^*(t) &= \frac{2e^{3t} + e^3}{e^{3t/2}(2+e^3)}, \\ u^*(t) &= \frac{2(e^{3t} - e^3)}{e^{3t/2}(2+e^3)}. \end{aligned}$$

Pitfalls of Discretizations [Hager, 2000]

Optimal Control Problem

$$\text{minimize } \frac{1}{2}\int_0^1 u^2(t) + 2y^2(t)dt$$

subject to

$$\begin{aligned} \dot{y}(t) &= \frac{1}{2}y(t) + u(t),\ t \in [0,1], \\ y(0) &= 1. \end{aligned}$$

$$\begin{aligned} \Rightarrow y^*(t) &= \frac{2e^{3t} + e^3}{e^{3t/2}(2+e^3)}, \\ u^*(t) &= \frac{2(e^{3t} - e^3)}{e^{3t/2}(2+e^3)}. \end{aligned}$$

Discretize with 2nd order RK

$$\text{minimize } \frac{h}{2}\sum_{k=0}^{K-1} u^2_{k+1/2} + 2y^2_{k+1/2}$$

subject to $(k = 0, \ldots, K)$:

$$\begin{aligned} y_{k+1/2} &= y_k + \frac{h}{2}(\frac{1}{2}y_k + u_k), \\ y_{k+1} &= y_k + h(\frac{1}{2}y_{k+1/2} + u_{k+1/2}), \end{aligned}$$

Pitfalls of Discretizations [Hager, 2000]

Optimal Control Problem

$$\text{minimize } \frac{1}{2}\int_0^1 u^2(t) + 2y^2(t)dt$$

subject to

$$\begin{aligned} \dot{y}(t) &= \frac{1}{2}y(t) + u(t),\ t \in [0,1], \\ y(0) &= 1. \end{aligned}$$

$$\Rightarrow y^*(t) = \frac{2e^{3t} + e^3}{e^{3t/2}(2+e^3)},$$

$$u^*(t) = \frac{2(e^{3t} - e^3)}{e^{3t/2}(2+e^3)}.$$

Discretize with 2nd order RK

$$\text{minimize } \frac{h}{2}\sum_{k=0}^{K-1} u_{k+1/2}^2 + 2y_{k+1/2}^2$$

subject to $(k = 0, \ldots, K)$:

$$\begin{aligned} y_{k+1/2} &= y_k + \frac{h}{2}(\frac{1}{2}y_k + u_k), \\ y_{k+1} &= y_k + h(\frac{1}{2}y_{k+1/2} + u_{k+1/2}), \end{aligned}$$

Discrete solution $(k = 0, \ldots, K)$:

$$y_k = 1, \quad y_{k+1/2} = 0,$$

$$u_k = -\frac{4+h}{2h}, \quad u_{k+1/2} = 0,$$

DOES NOT CONVERGE!

Tips to Solve Continuous-Time Problems

- Use discretize-then-optimize with different schemes
- Refine discretization: $h = 1$ discretization is nonsense
- Check implied discretization of adjoints

Tips to Solve Continuous-Time Problems

- Use discretize-then-optimize with different schemes
- Refine discretization: $h = 1$ discretization is nonsense
- Check implied discretization of adjoints

Alternative: Optimize-Then-Discretize

- Consistent adjoint/dual discretization
- Discretized gradients can be wrong!
- Harder for inequality constraints

Part V

Complementarity Constraints

Nash Games

- Non-cooperative game played by n individuals
  - Each player selects a strategy to optimize their objective
  - Strategies for the other players are fixed
- Equilibrium reached when no improvement is possible

Nash Games

- Non-cooperative game played by n individuals
  - Each player selects a strategy to optimize their objective
  - Strategies for the other players are fixed
- Equilibrium reached when no improvement is possible
- Characterization of two player equilibrium (x^*, y^*)

$$
\begin{aligned}
x^* &\in \left\{ \begin{array}{ll} \displaystyle\operatorname*{arg\,min}_{x \geq 0} & f_1(x, y^*) \\ \text{subject to} & c_1(x) \leq 0 \end{array} \right. \\
y^* &\in \left\{ \begin{array}{ll} \displaystyle\operatorname*{arg\,min}_{y \geq 0} & f_2(x^*, y) \\ \text{subject to} & c_2(y) \leq 0 \end{array} \right.
\end{aligned}
$$

Nash Games

- Non-cooperative game played by n individuals
  - Each player selects a strategy to optimize their objective
  - Strategies for the other players are fixed
- Equilibrium reached when no improvement is possible
- Characterization of two player equilibrium (x^*, y^*)

$$
\begin{aligned}
x^* &\in \left\{ \begin{array}{ll} \operatorname*{arg\,min}_{x \geq 0} & f_1(x, y^*) \\ \text{subject to} & c_1(x) \leq 0 \end{array} \right. \\
y^* &\in \left\{ \begin{array}{ll} \operatorname*{arg\,min}_{y \geq 0} & f_2(x^*, y) \\ \text{subject to} & c_2(y) \leq 0 \end{array} \right.
\end{aligned}
$$

- Many applications in economics
  - Bimatrix games
  - Cournot duopoly models
  - General equilibrium models
  - Arrow-Debreau models

Complementarity Formulation

- Assume each optimization problem is convex
  - $f_1(\cdot, y)$ is convex for each y
  - $f_2(x, \cdot)$ is convex for each x
  - $c_1(\cdot)$ and $c_2(\cdot)$ satisfy constraint qualification
- Then the first-order conditions are necessary and sufficient

$$\begin{array}{ll} \min\limits_{x \geq 0} & f_1(x, y^*) \\ \text{subject to} & c_1(x) \leq 0 \end{array} \quad \Leftrightarrow \quad \begin{array}{lcl} 0 \leq x & \perp & \nabla_x f_1(x, y^*) + \lambda_1^T \nabla_x c_1(x) \geq 0 \\ 0 \leq \lambda_1 & \perp & -c_1(x) \geq 0 \end{array}$$

Complementarity Formulation

- Assume each optimization problem is convex
  - $f_1(\cdot, y)$ is convex for each y
  - $f_2(x, \cdot)$ is convex for each x
  - $c_1(\cdot)$ and $c_2(\cdot)$ satisfy constraint qualification
- Then the first-order conditions are necessary and sufficient

$$\begin{array}{ll} \min\limits_{y \geq 0} & f_2(x^*, y) \\ \text{subject to} & c_2(y) \leq 0 \end{array} \quad \Leftrightarrow \quad \begin{array}{lcl} 0 \leq y & \perp & \nabla_y f_2(x^*, y) + \lambda_2^T \nabla_y c_2(y) \geq 0 \\ 0 \leq \lambda_2 & \perp & -c_2(y) \geq 0 \end{array}$$

Complementarity Formulation

- Assume each optimization problem is convex
  - $f_1(\cdot, y)$ is convex for each y
  - $f_2(x, \cdot)$ is convex for each x
  - $c_1(\cdot)$ and $c_2(\cdot)$ satisfy constraint qualification
- Then the first-order conditions are necessary and sufficient

$$
\begin{array}{lcl}
0 \leq x & \perp & \nabla_x f_1(x, y) + \lambda_1^T \nabla_x c_1(x) \geq 0 \\
0 \leq y & \perp & \nabla_y f_2(x, y) + \lambda_2^T \nabla_y c_2(y) \geq 0 \\
0 \leq \lambda_1 & \perp & -c_1(y) \geq 0 \\
0 \leq \lambda_2 & \perp & -c_2(y) \geq 0
\end{array}
$$

- Nonlinear complementarity problem
  - Square system – number of variables and constraints the same
  - Each solution is an equilibrium for the Nash game

Model Formulation

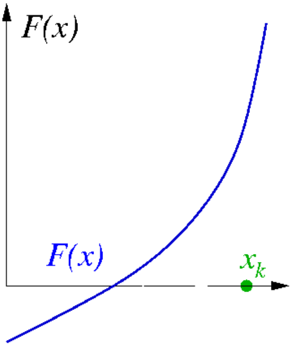
- Economy with n agents and m commodities
  - $e \in \Re^{n \times m}$ are the endowments
  - $\alpha \in \Re^{n \times m}$ and $\beta \in \Re^{n \times m}$ are the utility parameters
  - $p \in \Re^{m}$ are the commodity prices
- Agent i maximizes utility with budget constraint

$$\begin{array}{ll} \max\limits_{x_{i,*} \geq 0} & \sum\limits_{k=1}^{m} \dfrac{\alpha_{i,k}(1 + x_{i,k})^{1-\beta_{i,k}}}{1 - \beta_{i,k}} \\ \text{subject to} & \sum\limits_{k=1}^{m} p_k \left(x_{i,k} - e_{i,k}\right) \leq 0 \end{array}$$

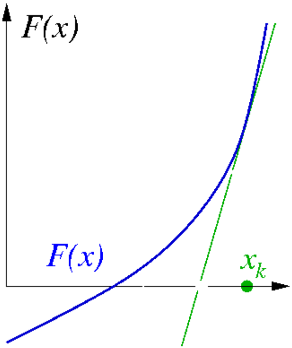
- Market k sets price for the commodity

$$0 \leq p_k \quad \perp \quad \sum_{i=1}^{n} (e_{i,k} - x_{i,k}) \geq 0$$

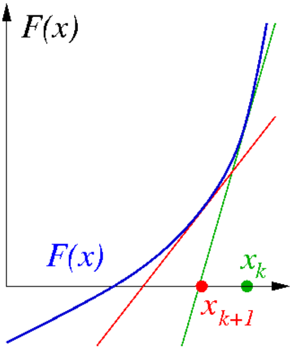
Newton Method for Nonlinear Equations



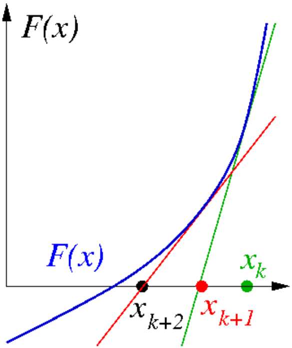
Newton Method for Nonlinear Equations



Newton Method for Nonlinear Equations



Newton Method for Nonlinear Equations



Methods for Complementarity Problems

- Sequential linearization methods (PATH)
    1. Solve the linear complementarity problem

$$0 \leq x \quad \perp \quad F(x_k) + \nabla F(x_k)(x - x_k) \geq 0$$

    2. Perform a line search along merit function
    3. Repeat until convergence

Methods for Complementarity Problems

- Sequential linearization methods (PATH)
  1. Solve the linear complementarity problem

$$0 \le x \quad \perp \quad F(x_k) + \nabla F(x_k)(x - x_k) \ge 0$$

  2. Perform a line search along merit function
  3. Repeat until convergence
- Semismooth reformulation methods (SEMI)
  - Solve linear system of equations to obtain direction
  - Globalize with a trust region or line search
  - Less robust in general
- Interior-point methods

Semismooth Reformulation

- Define Fischer-Burmeister function

$$\phi(a, b) := a + b - \sqrt{a^2 + b^2}$$

  - $\phi(a, b) = 0$ iff $a \geq 0$, $b \geq 0$, and $ab = 0$
- Define the system

$$[\Phi(x)]_i = \phi(x_i, F_i(x))$$

  - x^* solves complementarity problem iff $\Phi(x^*) = 0$
- Nonsmooth system of equations

Semismooth Algorithm

1. Calculate $H^k \in \partial_B \Phi(x^k)$ and solve the following system for d^k:

   $$H^k d^k = -\Phi(x^k)$$

   If this system either has no solution, or

   $$\nabla \Psi(x^k)^T d^k \leq -p_1 \|d^k\|^{p_2}$$

   is not satisfied, let $d^k = -\nabla \Psi(x^k)$.

Semismooth Algorithm

1. Calculate $H^k \in \partial_B \Phi(x^k)$ and solve the following system for d^k:

$$H^k d^k = -\Phi(x^k)$$

   If this system either has no solution, or

$$\nabla \Psi(x^k)^T d^k \leq -p_1 \|d^k\|^{p_2}$$

   is not satisfied, let $d^k = -\nabla \Psi(x^k)$.

2. Compute smallest nonnegative integer i^k such that

$$\Psi(x^k + \beta^{i^k} d^k) \leq \Psi(x^k) + \sigma \beta^{i^k} \nabla \Psi(x^k) d^k$$

3. Set $x^{k+1} = x^k + \beta^{i^k} d^k$, $k = k + 1$, and go to 1.

Convergence Issues

- Quadratic convergence – best outcome
- Linear convergence
  - Far from a solution – $r(x_k)$ is large
  - Jacobian is incorrect – disrupts quadratic convergence
  - Jacobian is rank deficient – $\|\nabla r(x_k)\|$ is small
  - Converge to local minimizer – guarantees rank deficiency
  - Limits of finite precision arithmetic
    1. $r(x_k)$ converges quadratically to small number
    2. $r(x_k)$ hovers around that number with no progress
- Domain violations such as $\frac{1}{x}$ when $x = 0$

Some Available Software

- PATH – sequential linearization method
- MILES – sequential linearization method
- SEMI – semismooth linesearch method
- TAO – Toolkit for Advanced Optimization
  - SSLS – full-space semismooth linesearch methods
  - ASLS – active-set semismooth linesearch methods
  - RSCS – reduced-space method

Definition

- Leader-follower game
  - Dominant player (leader) selects a strategy y^*
  - Then followers respond by playing a Nash game

$$x_i^* \in \begin{cases} \underset{x_i \geq 0}{\arg\min} & f_i(x, y) \\ \text{subject to} & c_i(x_i) \leq 0 \end{cases}$$

- Leader solves optimization problem with equilibrium constraints

$$\begin{array}{ll} \underset{y \geq 0, x, \lambda}{\min} & g(x, y) \\ \text{subject to} & h(y) \leq 0 \\ & 0 \leq x_i \quad \perp \quad \nabla_{x_i} f_i(x, y) + \lambda_i^T \nabla_{x_i} c_i(x_i) \geq 0 \\ & 0 \leq \lambda_i \quad \perp \quad -c_i(x_i) \geq 0 \end{array}$$

- Many applications in economics
  - Optimal taxation
  - Tolling problems

Model Formulation

- Economy with n agents and m commodities
  - $e \in \Re^{n \times m}$ are the endowments
  - $\alpha \in \Re^{n \times m}$ and $\beta \in \Re^{n \times m}$ are the utility parameters
  - $p \in \Re^{m}$ are the commodity prices
- Agent i maximizes utility with budget constraint

$$\begin{array}{ll} \max\limits_{x_{i,*} \geq 0} & \sum\limits_{k=1}^{m} \dfrac{\alpha_{i,k}(1 + x_{i,k})^{1-\beta_{i,k}}}{1 - \beta_{i,k}} \\ \text{subject to} & \sum\limits_{k=1}^{m} p_k \left(x_{i,k} - e_{i,k} \right) \leq 0 \end{array}$$

- Market k sets price for the commodity

$$0 \leq p_k \quad \perp \quad \sum_{i=1}^{n} (e_{i,k} - x_{i,k}) \geq 0$$

Nonlinear Programming Formulation

$$
\begin{array}{ll}
\min\limits_{x,y,\lambda,s,t\geq 0} & g(x,y) \\
\text{subject to} & h(y) \leq 0 \\
& s_i = \nabla_{x_i} f_i(x,y) + \lambda_i^T \nabla_{x_i} c_i(x_i) \\
& t_i = -c_i(x_i) \\
& \sum\limits_i \left(s_i^T x_i + \lambda_i t_i \right) \leq 0
\end{array}
$$

- Constraint qualification fails
  - Lagrange multiplier set unbounded
  - Constraint gradients linearly dependent
  - Central path does not exist
- Able to prove convergence results for some methods
- Reformulation very successful and versatile in practice

Penalization Approach

$$\begin{array}{ll}
\displaystyle\min_{x,y,\lambda,s,t\geq 0} & g(x,y) + \pi \sum_i \left(s_i^T x_i + \lambda_i t_i\right) \\
\text{subject to} & h(y) \leq 0 \\
 & s_i = \nabla_{x_i} f_i(x,y) + \lambda_i^T \nabla_{x_i} c_i(x_i) \\
 & t_i = -c_i(x_i)
\end{array}$$

- Optimization problem satisfies constraint qualification
- Need to increase π

Relaxation Approach

$$
\begin{array}{ll}
\min\limits_{x,y,\lambda,s,t \geq 0} & g(x,y) \\
\text{subject to} & h(y) \leq 0 \\
& s_i = \nabla_{x_i} f_i(x,y) + \lambda_i^T \nabla_{x_i} c_i(x_i) \\
& t_i = -c_i(x_i) \\
& \sum\limits_i \left(s_i^T x_i + \lambda_i t_i \right) \leq \tau
\end{array}
$$

- Need to decrease τ

Limitations

- Multipliers may not exist
- Solvers can have a hard time computing solutions
  - Try different algorithms
  - Compute feasible starting point
- Stationary points may have descent directions
  - Checking for descent is an exponential problem
  - Strong stationary points found in certain cases
- Many stationary points – global optimization

Limitations

- Multipliers may not exist
- Solvers can have a hard time computing solutions
  - Try different algorithms
  - Compute feasible starting point
- Stationary points may have descent directions
  - Checking for descent is an exponential problem
  - Strong stationary points found in certain cases
- Many stationary points – global optimization
- Formulation of follower problem
  - Multiple solutions to Nash game
  - Nonconvex objective or constraints
  - Existence of multipliers

Part VI

Mixed Integer and Global Optimization

Global Optimization

I need to find the GLOBAL minimum!

- use any NLP solver (often work well!)
- use the multi-start trick from previous slides
- global optimization based on branch-and-reduce: `BARON`
  - constructs global underestimators
  - refines region by branching
  - tightens bounds by solving LPs
  - solve problems with 100s of variables
- "voodoo" solvers: genetic algorithm & simulated annealing
  no convergence theory ... usually worse than deterministic

Derivative-Free Optimization

My model does not have derivatives!

- Change your model ... good models have derivatives!
- pattern-search methods for min $f(x)$
  - evaluate $f(x)$ at stencil $x_k + \Delta M$
  - move to new best point
  - extend to NLP; some convergence theory h
  - matlab: `NOMADm.m`; parallel `APPSPACK`
- solvers based on building interpolating quadratic models
  - DFO project on `www.coin-or.org`
  - Mike Powell's `NEWUOA` quadratic model
- "voodoo" solvers: genetic algorithm & simulated annealing
  no convergence theory ... usually worse than deterministic

Optimization with Integer Variables

Mixed-Integer Nonlinear Program (MINLP)

- modeling discrete choices $\Rightarrow$ $0-1$ variables
- modeling integer decisions $\Rightarrow$ integer variables
  e.g. number of different stocks in portfolio (8-10)
  not number of beers sold at Goose Island (millions)

MINLP solvers:

- branch (separate $z_i = 0$ and $z_i = 1$) and cut
- solve millions of NLP relaxations: `MINLPBB`, `SBB`
- outer approximation: iterate MILP and NLP solvers
  `BONMIN (COIN-OR)` & `FilMINT` on NEOS

Portfolio Management

- N: Universe of asset to purchase
- x_i: Amount of asset i to hold
- B: Budget

$$\text{minimize} \;\; u(x) \quad \text{subject to} \; \sum_{i \in N} x_i = B, \quad x \geq 0$$

Portfolio Management

- N: Universe of asset to purchase
- x_i: Amount of asset i to hold
- B: Budget

$$\text{minimize } u(x) \quad \text{subject to } \sum_{i \in N} x_i = B, \quad x \geq 0$$

- Markowitz: $u(x) \stackrel{\text{def}}{=} -\alpha^T x + \lambda x^T Q x$
  - α: maximize expected returns
  - Q: variance-covariance matrix of expected returns
  - λ: minimize risk; aversion parameter

More Realistic Models

- $b \in \mathbb{R}^{|N|}$ of “benchmark” holdings
- Benchmark Tracking: $u(x) \stackrel{\text{def}}{=} (x - b)^T Q(x - b)$
  - Constraint on $\mathbb{E}$[Return]: $\alpha^T x \geq r$

More Realistic Models

- $b \in \mathbb{R}^{|N|}$ of "benchmark" holdings
- Benchmark Tracking: $u(x) \stackrel{\text{def}}{=} (x-b)^T Q(x-b)$
  - Constraint on $\mathbb{E}$[Return]: $\alpha^T x \geq r$
- Limit Names: $|i \in N \ : \ x_i > 0| \leq K$
  - Use binary indicator variables to model the implication $x_i > 0 \Rightarrow y_i = 1$
  - Implication modeled with variable upper bounds:

$$x_i \leq By_i \qquad \forall i \in N$$

  - $\sum_{i \in N} y_i \leq K$

Optimization Conclusions

Optimization is General Modeling Paradigm

- linear, nonlinear, equations, inequalities
- integer variables, equilibrium, control

AMPL (GAMS) Modeling and Programming Languages

- express optimization problems
- use automatic differentiation
- easy access to state-of-the-art solvers

Optimization Software

- open-source: COIN-OR, IPOPT, SOPLEX, & ASTROS (soon)
- current solver limitations on laptop:
  - 1,000,000 variables/constraints for LPs
  - 100,000 variables/constraints for NLPs/NCPs
  - 100 variables/constraints for global optimization
  - 500,000,000 variable LP on BlueGene/P