Step Computation for Nonlinear Optimization

GIAN Short Course on Optimization: Applications, Algorithms, and Computation

Sven Leyffer

Argonne National Laboratory

September 12-24, 2016

Outline

Methods for Nonlinear Optimization: Introduction

Nonlinear Program (NLP) of the form

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

where

- objective $f : \mathbb{R}^n \to \mathbb{R}$ twice continuously differentiable
- constraints $c : \mathbb{R}^n \to \mathbb{R}^m$ twice continuously differentiable
- y multipliers of $c(x) = 0$
- $z \geq 0$ multipliers of $x \geq 0$.

... can reformulate more general NLPs easily

General Framework for NLP Solvers

Cannot solve NLP directly or explicitly:

$$\underset{x}{\text{minimize}} \quad f(x) \quad \text{subject to } c(x) = 0, \quad x \geq 0$$

$\Rightarrow$ approximate NLP & generate sequence $\{x^{(k)}\}$

- Each approximate problem can be solved inexactly.
- Refine approximation, if no progress made

Framework for Nonlinear Optimization Methods

Given $(x^{(0)}, y^{(0)}, z^{(0)}) \in \mathbb{R}^{n+m+n}$, set $k = 0$
while $x^{(k)}$ *is not optimal* **do**
  repeat
    Approx. solve/refine an approximation of NLP around $x^{(k)}$.
  until *Improved solution* $x^{(k+1)}$ *is found*;
  Check whether $x^{(k+1)}$ is optimal; set $k = k + 1$.
end

General Framework for NLP Solvers

Basic Components of NLP Methods

1. convergence test checks for optimal solutions or detects failure
2. approximate subproblem computes improved new iterate
3. globalization strategy ensures convergence from remote $x^{(0)}$
4. globalization mechanism truncates steps by local model
   ... enforce globalization strategy, refining local model

Categorize Algorithms by Components ... get interplay right!

Notation

- Iterates $x^{(k)}, k = 1, 2, \ldots$
- Function values, $f^{(k)} = f(x^{(k)})$ and $c^{(k)} = c(x^{(k)})$
- Gradients $g^{(k)} = \nabla f(x^{(k)})$ and Jacobian $A^{(k)} = \nabla c(x^{(k)})$
- Hessian of the Lagrangian is $H^{(k)}$.

Outline

Convergence Test for Feasible Limits

Check optimality & provide local improvement model

Approximate Convergence Test, from KKT Conditions

$$\|c^{(k)}\| \leq \epsilon, \ \|g^{(k)} - A^{(k)} y^{(k)} - z^{(k)}\| \leq \epsilon, \text{ and } \|\min(x^{(k)}, z^{(k)})\| \leq \epsilon$$

where $\epsilon > 0$ is tolerance

Note: $\min(x^{(k)}, z^{(k)}) = 0 \quad \Leftrightarrow \quad x^{(k)} \geq 0, \ z^{(k)} \geq 0 \ x_i^{(k)} z_i^{(k)} = 0$

If constraints do not satisfy MFCQ then may only get

$$\|A^{(k)} y^{(k)} + z^{(k)}\| \leq \epsilon, \qquad \text{approx. Fritz-John Point}$$

... equivalent to setting "objective multiplier" $y_0 = 0$

Convergence Test for Feasible Limits

Unless NLP is convex, cannot guarantee to find feasible point!

$$\underset{x}{\text{minimize}} \quad f(x) \quad \text{subject to } c(x) = 0, \quad x \geq 0$$

Proving that NLP is infeasible is hard

Only realistic to consider local feasibility problem

$$\underset{x}{\text{minimize}} \; \|c(x)\| \quad \text{subject to } x \geq 0,$$

formulate as smooth problem, e.g. ℓ_1 norm, $\|v\|_1 = \sum |v_i|$

$$\Leftrightarrow \quad \begin{array}{ll} \underset{x}{\text{minimize}} & \displaystyle\sum_{i=1}^{m} s_i^+ + s_i^- \\ \text{subject to} & s^+ - s^- = c(x), \; x \geq 0, \; s^+, s^- \geq 0 \end{array}$$

Approximate Optimality for Infeasible NLPs

Local feasibility problem

$$\underset{x}{\text{minimize}} \ \|c(x)\| \quad \text{subject to } x \geq 0,$$

Approximate Infeasible First-Order Stationary Point

$$\|A^{(k)}y^{(k)} - z^{(k)}\| \leq \epsilon \ \text{ and } \ \|\min(x^{(k)}, z^{(k)})\| \leq \epsilon,$$

where $y^{(k)}$ multipliers/weights corresponding to feasibility norm

Example: For ℓ_1 norm (verify using KKT conditions!)

- If $[c^{(k)}]_i < 0$ then $[y^{(k)}]_i = -1$
- If $[c^{(k)}]_i > 0$ then $[y^{(k)}]_i = 1$
- $-1 \leq [y^{(k)}]_i \leq 1$ otherwise.

By-product of solving the local model ... related to dual norm ℓ_∞

Outline

SQP for Equality Constraints

Sequential Quadratic Programming (SQP) for Equality Constraints

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f(x) \\ \text{subject to} & c(x) = 0. \end{array}$$

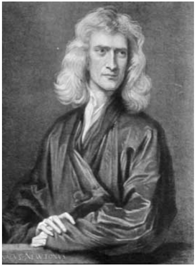
SQP is Newton's method applied to KKT conditions

Let Lagrangian, $\mathcal{L}(x,y) = f(x) - y^T c(x)$, apply Newton's method

$$\begin{pmatrix} \nabla_x \mathcal{L}(x,y) \\ \nabla_y \mathcal{L}(x,y) \end{pmatrix} = 0 \quad \Leftrightarrow \quad \begin{pmatrix} g(x) - A(x)y \\ -c(x) \end{pmatrix} = 0.$$

Let's quickly recall Newton's method ...

Newton's Method for Nonlinear Equations



Solve $F(x) = 0$:
Get approx. $x^{(k+1)}$ of solution of $F(x) = 0$
by solving linear model about $x^{(k)}$:

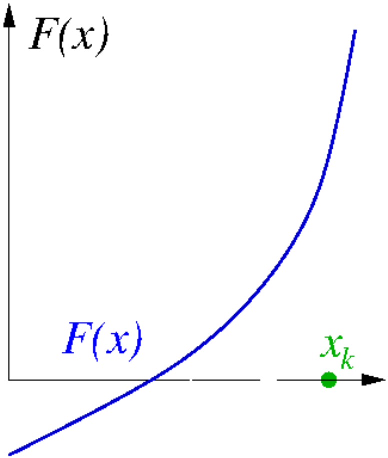
$$F(x^{(k)}) + \nabla F(x^{(k)})^T (x - x^{(k)}) = 0$$

for $k = 0, 1, \ldots$

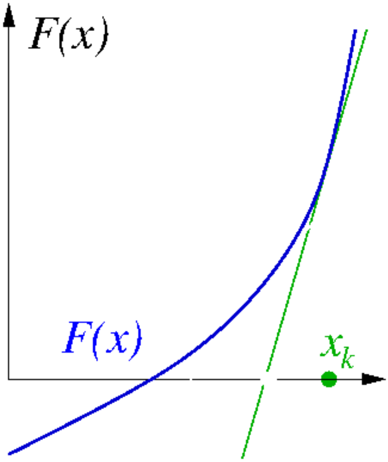
Theorem (Newton's Method)

If $F \in \mathcal{C}^2$, and $\nabla F(x^*)$ nonsingular,
then Newton converges quadratically near x^*.

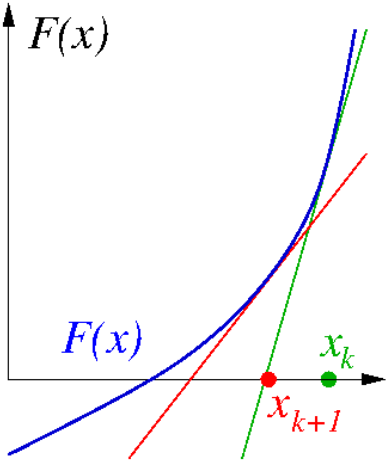
Newton's Method for Nonlinear Equations



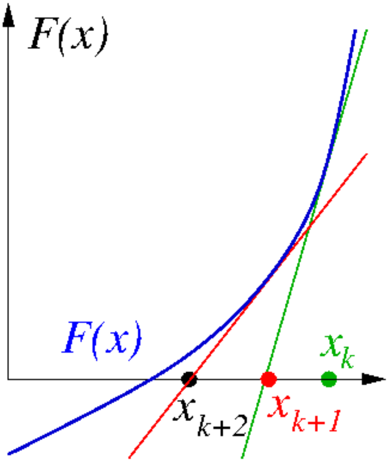
Newton's Method for Nonlinear Equations



Newton's Method for Nonlinear Equations



Newton's Method for Nonlinear Equations



Newton's Method Applied to KKT Systems

Apply Newton's around $(x^{(k)}, y^{(k)})$ to KKT system ...

$$\begin{pmatrix} \nabla_x \mathcal{L}(x,y) \\ \nabla_y \mathcal{L}(x,y) \end{pmatrix} = 0 \quad \Leftrightarrow \quad \begin{pmatrix} g(x) - A(x)y \\ -c(x) \end{pmatrix} = 0.$$

... gives linearized systems

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}^{(k)} & \nabla^2_{xy}\mathcal{L}^{(k)} \\ \nabla^2_{yx}\mathcal{L}^{(k)} & \nabla^2_{yy}\mathcal{L}^{(k)} \end{bmatrix} \begin{pmatrix} d_x \\ d_y \end{pmatrix} = -\begin{pmatrix} g^{(k)} - A^{(k)}y^{(k)} \\ c^{(k)} \end{pmatrix}.$$

Lagrangian linear in $y \Rightarrow \nabla^2_{yy}\mathcal{L}^{(k)} = 0$ and $\nabla^2_{xy}\mathcal{L}^{(k)} = -A^{(k)}$:

$$\begin{bmatrix} H^{(k)} & -A^{(k)} \\ -A^{(k)^T} & 0 \end{bmatrix} \begin{pmatrix} d_x \\ d_y \end{pmatrix} = -\begin{pmatrix} g^{(k)} - A^{(k)}y^{(k)} \\ -c^{(k)} \end{pmatrix},$$

where $H^{(k)} = \nabla^2_{xx}\mathcal{L}^{(k)}$.

Newton's Method Applied to KKT Systems

Newton system

$$\begin{bmatrix} H^{(k)} & -A^{(k)} \\ -A^{(k)^T} & 0 \end{bmatrix} \begin{pmatrix} d_x \\ d_y \end{pmatrix} = - \begin{pmatrix} g^{(k)} - A^{(k)} y^{(k)} \\ -c^{(k)} \end{pmatrix},$$

If we take full steps,

$$x^{(k+1)} = x^{(k)} + d_x \quad \text{and} \quad y^{(k+1)} = y^{(k)} + d_y,$$

then system equivalent to KKT matrix

$$\begin{bmatrix} H^{(k)} & -A^{(k)} \\ -A^{(k)^T} & 0 \end{bmatrix} \begin{pmatrix} d_x \\ y^{(k+1)} \end{pmatrix} = \begin{pmatrix} -g^{(k)} \\ c^{(k)} \end{pmatrix}.$$

... are the KKT conditions of quadratic program:

$$\begin{aligned} \underset{d}{\text{minimize}} \quad & q^{(k)}(d) = \tfrac{1}{2} d^T H^{(k)} d + g^{(k)^T} d + f^{(k)} \\ \text{subject to} \quad & c^{(k)} + A^{(k)^T} d = 0. \end{aligned}$$

SQP for Equality Constraints

Equality-Constrained NLP

$$\underset{x}{\text{minimize}} \; f(x) \quad \text{subject to } c(x) = 0$$

Sequential Quadratic Programming Method

Given $(x^{(0)}, y^{(0)})$, set $k = 0$

repeat

Solve QP subproblem around $x^{(k)}$

$$\begin{aligned} \underset{d}{\text{minimize}} \quad & q^{(k)}(d) = \tfrac{1}{2} d^T H^{(k)} d + {g^{(k)}}^T d + f^{(k)} \\ \text{subject to} \quad & c^{(k)} + {A^{(k)}}^T d = 0. \end{aligned}$$

... let solution be $(d_x, y^{(k+1)})$

Set $x^{(k+1)} = x^{(k)} + d_x$ and $k = k + 1$

until $(x^{(k)}, y^{(k)})$ *optimal*;

SQP for Equality Constraints

Convergence of Newton's Method

- Does not converge from an arbitrary starting point.
- Quasi-Newton approximations of Hessian, $H^{(k)}$ with

$$\gamma^{(k)} = \nabla\mathcal{L}(x^{(k+1)}, y^{(k+1)}) - \nabla\mathcal{L}(x^{(k)}, y^{(k+1)})$$

Theorem (Quadratic Convergence of SQP)

Let x^* be second-order sufficient, and assume that the KKT matrix

$$\begin{bmatrix} H^* & -A^* \\ -{A^*}^T & 0 \end{bmatrix} \quad \textit{is nonsingular}$$

If $x^{(0)}$ sufficiently close to x^*, SQP converges quadratically to x^*.

Outline

SQP for General Constraints

Sequential Quadratic Programming (SQP) for Equality Constraints

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

SQP Methods

Date back to 1970's [Han, 1977, Powell, 1978]

- Minimize quadratic model, $m^{(k)}(d)$
- Subject to linearization of constraints

around $x^{(k)}$ for step $d := x - x^{(k)}$

SQP for General Constraints

Sequential Quadratic Programming (SQP) for Equality Constraints

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x) = 0 \\ & x \geq 0 \end{aligned}$$

QP Subproblem

$$\begin{aligned} \underset{d}{\text{minimize}} \quad & m^{(k)}(d) := g^{(k)^T} d + \frac{1}{2} d^T H^{(k)} d \\ \text{subject to} \quad & c^{(k)} + A^{(k)^T} d = 0 \\ & x^{(k)} + d \geq 0, \end{aligned}$$

New iterate $x^{(k+1)} = x^{(k)} + d$, and $y^{(k+1)}$ multipliers

where

- $H^{(k)} \simeq \nabla^2 \mathcal{L}(x^{(k)}, y^{(k)})$ approx Hessian of Lagrangian
- $y^{(k)}$ multiplier estimate at iteration k

Discussion of SQP

QP Subproblem

$$\begin{aligned} \underset{d}{\text{minimize}} \quad & m^{(k)}(d) := g^{(k)^T} d + \tfrac{1}{2} d^T H^{(k)} d \\ \text{subject to } & c^{(k)} + A^{(k)^T} d = 0 \\ & x^{(k)} + d \geq 0, \end{aligned}$$

New iterate $x^{(k+1)} = x^{(k)} + d$, and $y^{(k+1)}$ multipliers

- If $H^{(k)}$ not positive definite on null-space equations
  $\Rightarrow$ QP subproblem is nonconvex ... SQP is OK with local min
- Solution QP subproblem can be computationally expensive
  - Factorization of $[A^{(k)} : V]$ is OK
  - Factorization of dense reduced Hessian, $Z^T H^{(k)} Z$, slow

$\Rightarrow$ look for cheaper alternatives for large-scale NLP

Outline

Sequential Linear Programming (SLP)

General nonlinear program (NLP):

$$\underset{x}{\text{minimize}} \;\; f(x) \quad \text{subject to } c(x) = 0 \quad x \geq 0$$

LPs can be solved more efficiently than QPs

- Consider sequential linear algorithm
- Need a trust-region, because can be LP unbounded

LP Trust-Region Subproblem

$$\begin{aligned} &\underset{d}{\text{minimize}} \;\; m^{(k)}(d) = g^{(k)^T} d \\ &\text{subject to } c^{(k)} + A^{(k)^T} d = 0, \\ &\qquad\qquad x^{(k)} + d \geq 0, \quad \text{and} \quad \|d\|_\infty \leq \Delta_k, \end{aligned}$$

where $\Delta_k > 0$ trust-region radius; need $\Delta_k \to 0$ to converge

Generalizes steepest descent method from Part II

SLQP Methods

General nonlinear program (NLP):

$$\underset{x}{\text{minimize}} \ f(x) \quad \text{subject to } c(x) = 0 \quad x \geq 0$$

Sequential Linear/Quadratic Programming (SLQP) methods

- Combine SLP (fast subproblem) & SQP (fast convergence)
- Use LP Trust-Region Subproblem

$$\begin{aligned} &\underset{d}{\text{minimize}} \ \ m^{(k)}(d) = g^{(k)^T} d \\ &\text{subject to } c^{(k)} + A^{(k)^T} d = 0, \\ &\qquad\qquad\quad x^{(k)} + d \geq 0, \quad \text{and} \quad \|d\|_\infty \leq \Delta_k, \end{aligned}$$

  to find active set estimate

$$\mathcal{A}^{(k)} := \left\{ i : [x^{(k)}]_i + \hat{d}_i = 0 \right\}$$

- Given active-set estimate, do one equality SQP step

SLQP Methods

Given estimate of active set,

$$\mathcal{A}^{(k)} := \left\{ i : [x^{(k)}]_i + \hat{d}_i = 0 \right\}$$

... excludes trust-region bounds!

Construct equality-constrained QP (EQP):

$$\begin{aligned} \underset{d}{\text{minimize}} \quad & q^{(k)}(d) = g^{(k)^T} d + \tfrac{1}{2} d^T H^{(k)} d \\ \text{subject to } & c^{(k)} + A^{(k)^T} d = 0, \\ & [x^{(k)}]_i + d_i = 0, \ \forall i \in \mathcal{A}^{(k)}, \end{aligned}$$

... "generalizes" projected-gradient method

SLQP Methods

Equality-constrained QP (EQP):

$$\begin{aligned} \underset{d}{\text{minimize}} \quad & q^{(k)}(d) = g^{(k)^T} d + \tfrac{1}{2} d^T H^{(k)} d \\ \text{subject to } & c^{(k)} + A^{(k)^T} d = 0, \\ & [x^{(k)}]_i + d_i = 0, \ \forall i \in \mathcal{A}^{(k)}, \end{aligned}$$

If $H^{(k)}$ second-order sufficient ... pos.-def. on null-space
Then solution of EQP equivalent to KKT system of EQP:

$$\begin{bmatrix} H^{(k)} & -A^{(k)} & -I^{(k)} \\ A^{(k)^T} & & \\ I^{(k)^T} & & \end{bmatrix} \begin{pmatrix} x \\ y \\ z_{\mathcal{A}} \end{pmatrix} = \begin{pmatrix} -g^{(k)} + H^{(k)} x^{(k)} \\ -c^{(k)} \\ 0 \end{pmatrix},$$

where

- $I^{(k)} = [e_i]_{i \in \mathcal{A}^{(k)}}$ are normals of active inequalities
- $z_{\mathcal{A}}$ multipliers active inequalities, $x_{\mathcal{A}} = 0$
- Can ensure that $[A^{(k)} : I^{(k)}]$ has full rank
- MA57 detects inertia ... perturb $H^{(k)} + \mu I$ if not correct

SQQP Methods

LP prediction of active set may be poor ...
... alternative is to use QP to predict active set

Wait a Minute ... This Sounds Crazy!

Why would you solve two QPs, i.e. the same problem twice?

SQQP Methods

LP prediction of active set may be poor ...
... alternative is to use QP to predict active set

Wait a Minute ... This Sounds Crazy!

Why would you solve two QPs, i.e. the same problem twice?

- First QP model is positive-definite,
  e.g. using limited-memory quasi-Newton updates
  - Ensures descend direction without line-search
  - Easier to solve ... no indefinite Hessian
- Second equality QP uses exact Hessian of Lagrangian
  - Ensures fast asymptotics
  - Uses "correct" Hessian information

Theory of SQP, SLP, SLQP, SQQP Methods

Second-order convergence under reasonable assumptions

- $H^{(k)}$ exact Hessian of Lagrangian
- Jacobian of active constraints has full rank
- A constraint qualification holds
- Limit x^* is second-order sufficient

e.g. [Boggs and Tolle, 1995].

Under additional assumption of strict complementarity
can show identification of optimal active set in finite iterations

Summary and Teaching Points

Sequential Quadratic Programming (SQP) et al.

- Family of active-set methods
- Motivated by Newton's method $\Rightarrow$ fast local convergence
- Modern implementations exploit fast LP solvers
  1. SLQP: alternate between active-set indentifying LP and EQP
  2. SQQP: alternatev between convex QP and EQP
- All methods require mechanism to enforce convergence from remote starting points

Boggs, P. and Tolle, J. (1995).
Sequential quadratic programming.
Acta Numerica, 4:1–51.

Han, S. (1977).
A globally convergent method for nonlinear programming.
Journal of Optimization Theory and Applications, 22(3):297–309.

Powell, M. (1978).
A fast algorithm for nonlinearly constrained optimization calculations.
In Watson, G., editor, *Numerical Analysis, 1977*, pages 144–157.
Springer–Verlag, Berlin.