Overview of continuous optimization advances and applications to circuit tuning

Andrew R. Conn and Chandu Visweswariah

T.J. Watson Research Center, IBM

Two fundamental approaches

Line search :

- Find a ‘good’ descent direction
- Asymptotically Newton: $d = -B^{-1}g$; $B \simeq \nabla^2 f(x)$
- $x^{i+1} = x^i + \alpha d$: 1 dimensional ‘minimization’

Trust Region :

- Trust a model [2^{nd}-order Taylor's expansion]

  $m(x+d) = f(x) + d^T \nabla f(x) + \frac{1}{2} d^T B d$

  Minimize (approximately) within a region

  $\|d\| \leq \Delta$: n-dimensional 'minimization'

- Trust region management:

$$\rho = \frac{f(x+d) - f(x)}{m(x+d) - m(x)}$$

  - $\rho > 0.75$ $\quad x \leftarrow x^{i+1} \quad \Delta \leftarrow 2\Delta$
  - $\rho < 0.15$ $\quad x \leftarrow x^{i} \quad \Delta \leftarrow \frac{1}{2}\Delta$
  - Otherwise $\quad x \leftarrow x^{i+1} \quad \Delta \leftarrow \Delta$

- Good match between theory and practice — simple

Fundamental result:

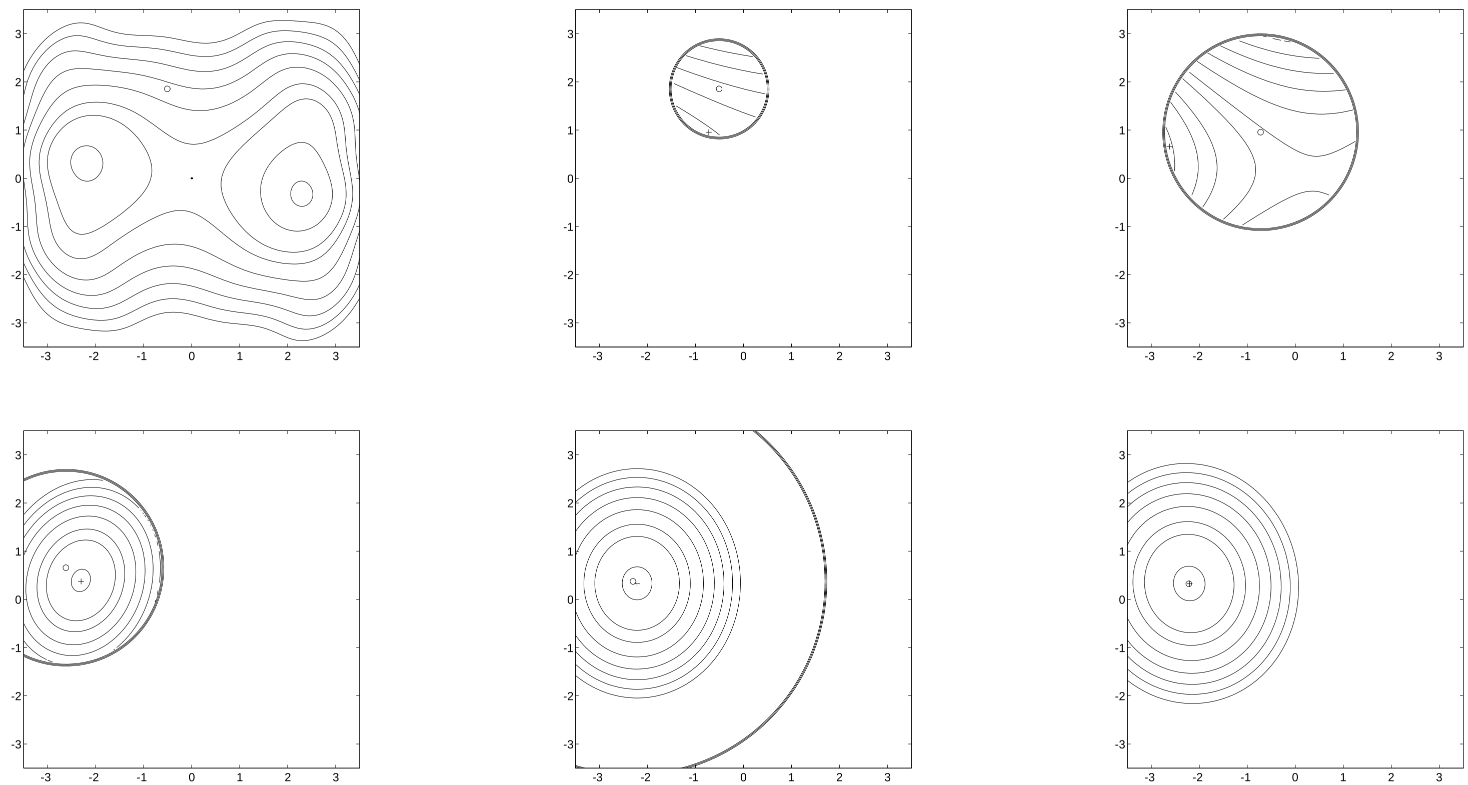
$$m_k(x_k) - m_k(x_k^{\mathrm{C}}) \geq \tfrac{1}{2}\|g_k\| \min\left[\frac{\|g_k\|}{\beta_k}, \Delta_k\right],$$

where

$$g_k = \nabla_x m_k(x_k)$$

$$\beta_k = 1 + \max_{x \in \mathcal{B}_k} \|\nabla_{xx} m_k(x)\|$$

Trust-region method illustrated



Illustrating six iterations of a trust-region algorithm

Methods aim to satisfy stationarity conditions

- $\nabla f(x) = 0$ First-order conditions
  [Solve linear system if quadratic model]

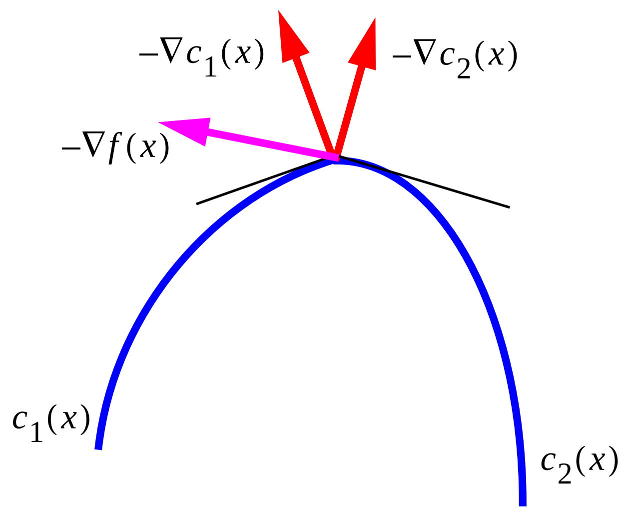
- $\nabla^2 f(x) > 0$ Second-order conditions

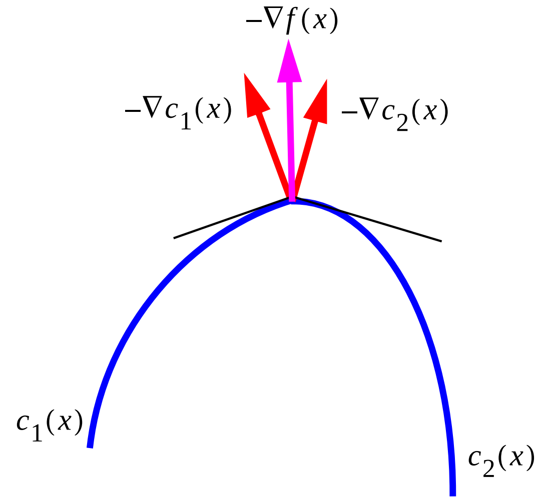
Constraints — restricted versions of the same conditions

- $Z^T \nabla f(x) = 0$ First-order conditions

- $Z^T \nabla^2 f(x) Z > 0$ Second-order conditions

Z is a basis for the null space of the activities





Constrained optimality conditions

More precisely

Dual Feasibility : $\nabla f(x) = \sum_i \lambda_i \nabla c_i(x), \quad \lambda_i \geq 0$

Primal Feasibility : $c_i(x) \geq 0$

Complimentary Slackness : $c^T \lambda = 0$

Two fundamental approaches

- Change to unconstrained : penalty methods

- Deal with constraints explicitly
  - maintain feasibility
  - active-set methods

Penalty methods

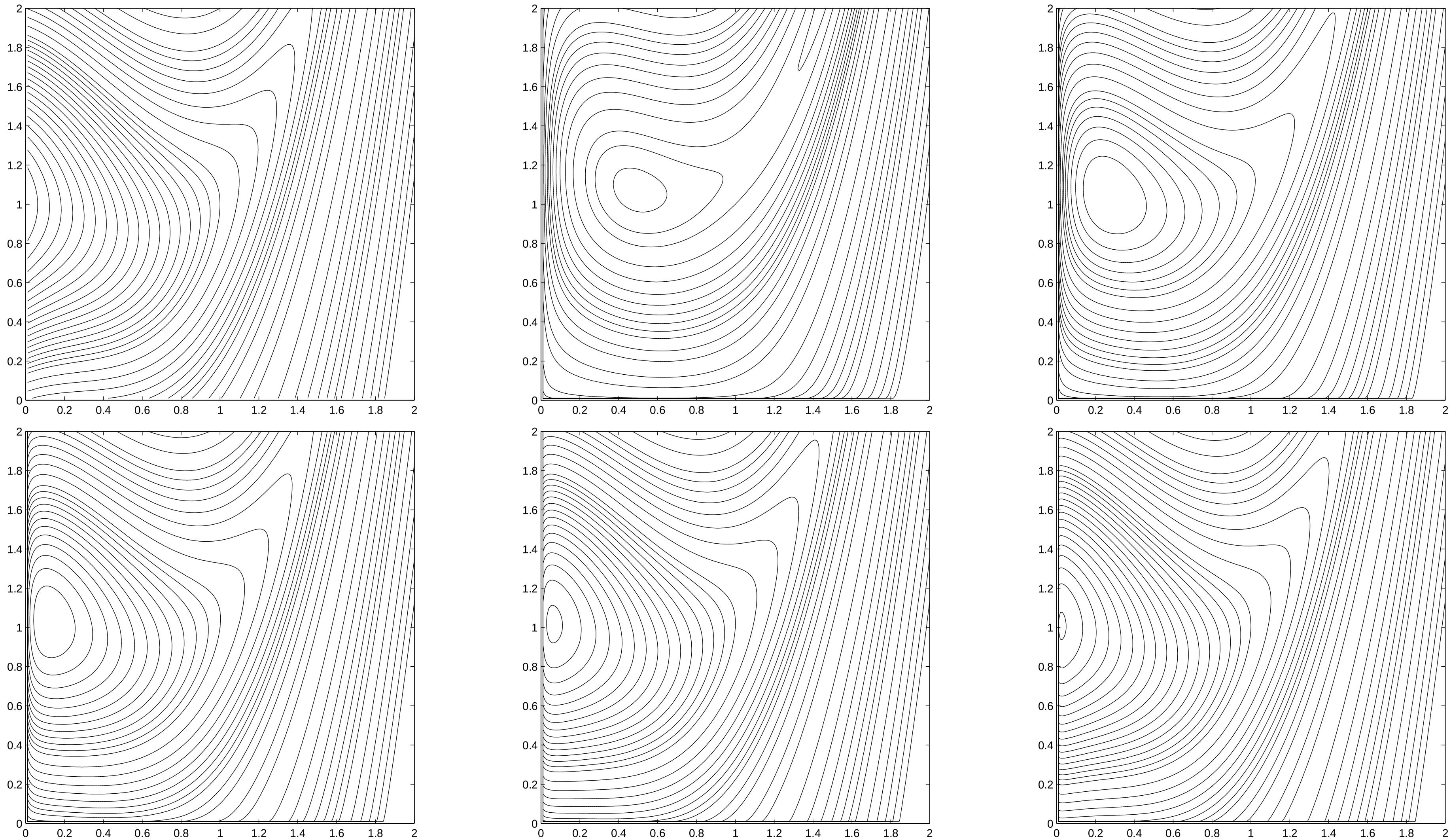
Exterior eg quadratic penalty

- $f(x) + \frac{1}{2\mu} \sum_i c_i^2(x)$ [or $-min\{c_i(x), 0\}^2$]
- $\liminf_{\mu \to 0+} x(\mu) = x^*$

Interior eg log barrier

- $f(x) - \mu \sum_i log c_i(x)$
- $\liminf_{\mu \to 0+} x(\mu) = x^*$

Log barrier illustrated



Important compromise

Augmented Lagrangian

- $f(x) - \sum_i \lambda_i c_i(x) + \frac{1}{2\mu} \sum_i c_i^2(x)$

- Equivalent to the shifted penalty $f(x) + \frac{1}{2\mu} [c_i(x) + \mu\lambda_i]^2$

- Now have a choice of updating λ or μ

- $\liminf_{\mu \to 0+}$ no longer essential

- Update $\lambda_i^+ = \lambda_i + \frac{c_i(x)}{\mu}$

  Since $\nabla f(x) - \sum_i \lambda_i \nabla c_i(x) + \frac{1}{\mu} \sum_i c_i(x) \nabla c_i(x) = 0$

Active-set methods

Guess the correct active set intelligently

- Try to keep activities tight by choosing, d, for (near) active constraints such that $d^T \nabla c_i(x) = 0$
- Whenever 1^{st}-order conditions satisfied, i.e. $Z^T \nabla f(x) = 0$ drop some activity (using sign of the multipliers) that shouldn't be active
- Ultimately satisfy the correct optimality conditions

We use an active-set method in LANCELOT for the simple bounds

Successive quadratic programming

Use second-order models

$$\begin{array}{ll} \text{min} & f(x) + d^T \nabla f(x) + \tfrac{1}{2} d^T \nabla^2 f(x) d \\ \text{s.t.} & c_i(x) + d^T \nabla c_i(x) + \tfrac{1}{2} d^T \nabla^2 c_i(x) d \geq 0 \end{array}$$

At optimality

$$\nabla f(x) + \nabla^2 f(x) d = \sum \lambda_i \left[\nabla c_i(x) + \nabla^2 c_i(x) d \right]$$

$$\lambda_i c_i(x) = 0 \quad c_i(x) \geq 0 \quad \text{and} \quad \lambda_i \geq 0$$

Suppose $\widehat{\lambda}_i$ is a good estimate for λ_i then

$$\nabla f(x) + [\nabla^2 f(x) - \sum \widehat{\lambda}_i \nabla^2 c_i(x)] d \simeq \sum \lambda_i \nabla c_i(x)$$

Now observe ......

Optimality conditions for

$$\begin{array}{ll} \text{min} & f(x) + d^T \nabla f(x) + \frac{1}{2} d^T B d \\ \text{s.t.} & c_i(x) + d^T \nabla c_i(x) \geq 0 \end{array}$$

are

$$\nabla f(x) + [\nabla^2 f(x) - \sum \hat{\lambda}_i \nabla^2 c_i(x)] d \simeq \sum \lambda_i \nabla c_i(x)$$

where

$$B = [\nabla^2 f(x) - \sum \hat{\lambda}_i \nabla^2 c_i(x)]$$

Hence B is an approximation to the Hessian of the Lagrangian

THIS IS THE BASIS FOR SQP

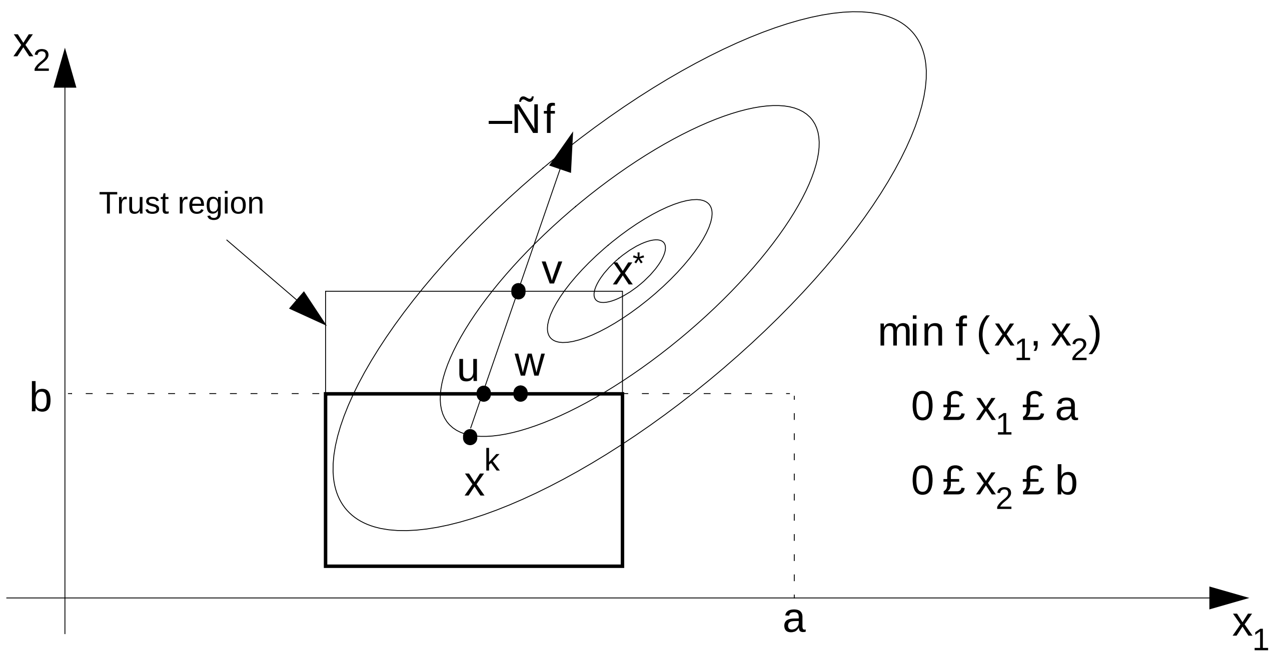
LANCELOT

Augmented Lagrangian + Trust Region +
Group Partial Separable

$$f(x) = \sum_{i=1}^{n_g} g_i(\alpha_i(x)); \quad \alpha_i(x) = \sum_{j \in \mathcal{J}_i} w_{i,j} f_j(x^{[j]}) + a_i^T x - b_i$$

ANOOXAPE

- Augmented Lagrangian for general constraints
- Active set method for linear (in)equalities
- Log barrier for slacks, surpluses and simple bounds



Typical LANCELOT iteration

Three important statements

- Nonlinear optimization has progressed enormously in the past thirty years — good applications must use the latest advances

- Unfortunately algorithms too complex to be just used as a black box

- Whenever possible it is essential to use derivatives

If you are looking for software

- http://www-fp.mcs.anl.gov/otc/Guide/SoftwareGuide
- LANCELOT
- MINOS
- SNOPT
- LOQO