Arc Search Algorithms

Nick Henderson and Walter Murray

Stanford University
Institute for Computational and Mathematical Engineering

November 10, 2011

Unconstrained Optimization

$$\underset{x \in D}{\text{minimize}} \quad F(x)$$

where $F(x)$ is a twice continuously differentiable real-valued function on an open convex set $D \subseteq \mathbf{R}^n$.

We are mainly concerned with methods that use second derivatives. Every iteration of the algorithm has access to:

- The gradient: $g_k = \nabla F(x_k) \in \mathbf{R}^n$
- The Hessian: $H_k = \nabla^2 F(x_k) \in \mathbf{R}^{n \times n}$

Newton's method

The gold standard in optimization is Newton's method:

$$x_{k+1} = x_k - H_k^{-1} g_k$$

- has quadratic convergence if $H^* \succ 0$ and x_k is close enough to x^*
- given arbitrary starting point, x_0, may diverge or converge to maximizer
- not defined if H_k is singular

Descent Methods

Descent methods are designed to mimic Newton's method when it works, but enforce convergence to satisfactory points. These methods satisfy the descent property:

$$F(x_{k+1}) < F(x_k) \text{ for } k = 0, 1, 2, \ldots$$

List of methods:

- The line search method
- The trust-region method
- The method of gradients
- The arc search method

Line search

- Update: $x_{k+1} = x_k + \alpha_k p_k$
  - $p_k \in \mathbf{R}^n$ is the search direction
  - $\alpha_k \in \mathbf{R}$ is the step length
- Subproblem: $H_k p_k = -g_k$
- Step length, α_k, chosen by a univariate search routine over $F(x_k + \alpha_k p_k)$.
- Comments:
  - If H_k is not positive definite, it must be replaced by a sufficiently positive definite approximation $\bar{H}_k$.
  - The process to determine $\bar{H}_k$ will also give a direction of negative curvature, d_k, such that $d_k^T H_k d_k < 0$.

Trust Region

- Update: $x_{k+1} = x_k + s_k$
- Subproblem, s_k solves:

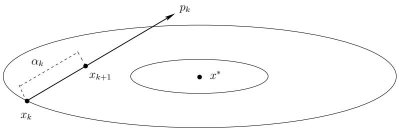
$$\begin{array}{ll} \text{minimize} & s^T H_k s + g_k^T s \\ \text{subject to} & s^T s \leq \Delta^2 \end{array}$$

- Trust region size, Δ, is decreased if $F(x_k + s_k) \geq F(x_k)$
- s_k is obtained by solving:

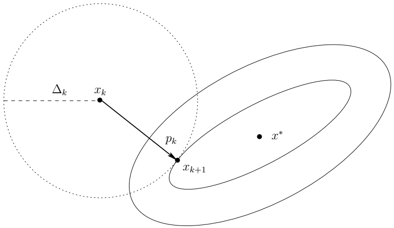
$$(H_k + \pi I)s = -g_k$$

- May require several solves to get correct π

Direction and Distance



Linesearch Method



Trust-Region Method

Ok, what's the problem?

Line search:

- must come up with special methods to deal with indefinite case, not clear what's best
- not clear how to best use directions of negative curvature

Trust-region:

- may need to solve linear system multiple times
- scaling of the trust region is an issue
- no longer convergent to second order optimality conditions when presented with constraints

Arc search

Arc search methods search along an arc, Γ_k, constructed at each iteration. A univariate search procedure selects an appropriate step length α_k, which gives the update:

$$x_{k+1} = x_k + \Gamma_k(\alpha_k)$$

Issues:

- Convergence theory
- Construction of arcs
- Selection of step length

Arc search: convergence theory

Convergence theory for line search algorithms can be extended to arc search. First, we form a univariate search function

$$\phi_k(\alpha) = F(x_k + \Gamma_k(\alpha)) \tag{1}$$

and look at the first and second derivatives with $\alpha = 0$

$$\phi_k'(0) = g_k^T \Gamma_k'(0) \tag{2}$$

$$\phi_k''(0) = \Gamma_k'(0)^T H_k \Gamma_k'(0) + g_k^T \Gamma_k''(0). \tag{3}$$

Arc Search: convergence theory

The step parameter α is selected to satisfy a **descent condition**

$$\phi_k(\alpha_k) \le \phi_k(0) + \mu \left(\phi_k'(0)\alpha_k + \tfrac{1}{2}\min\{\phi_k''(0), 0\}\alpha_k^2\right) \tag{4}$$

and a **curvature condition**

$$|\phi_k'(\alpha_k)| \le \eta|\phi_k'(0) + \min\{\phi_k''(0), 0\}\alpha_k|. \tag{5}$$

Theorem

If $\phi_k \in C^2$ *and bounded below with* $\phi_k'(0) < 0$ *and* $0 < \mu \le \eta < 1$*, then* α_k *is well defined.*

Arc Search: convergence theory

Theorem

If the sequence $\{x_k\}_{k=0}^{\infty}$ *is generated as specified above, then*

$$\lim_{k\to\infty} \phi_k'(0) = 0 \textit{ and} \tag{6}$$

$$\lim_{k\to\infty} \phi_k''(0) \geq 0. \tag{7}$$

Convergence to optimality

Convergence to points statisfying the first and second order necessary conditions is guaranteed if

$$\lim_{k\to\infty} \phi'(0) = 0 \implies \lim_{k\to\infty} g_k = 0$$

and

$$\liminf_{k\to\infty} \phi_k''(0) \geq 0 \implies \liminf_{k\to\infty} \lambda_{\min}(H_k) \geq 0.$$

Arc properties sufficient for convergence

We obtain properties of the arc sufficient for convergence by substituting in for ϕ. The sufficient descent condition is:

$$\lim_{k\to\infty} g_k^T \Gamma'_k(0) = 0 \implies \lim_{k\to\infty} g_k = 0$$

The sufficient curvature condition is:

$$\liminf_{k\to\infty} \Gamma'_k(0)^T H_k \Gamma'_k(0) + g_k^T \Gamma''_k(0) \geq 0 \implies \liminf_{k\to\infty} \lambda_{\min}(H_k) \geq 0$$

It remains to construct arcs statisfying these conditions.

Examples arcs

Several arcs have been studied. In the following table, s_k is a descent direction and d_k is a direction of negative curvature.

algorithm	$\Gamma_k(\alpha)$
gradient descent	$-\alpha g_k$
Moré & Sorensen	$\alpha^2 s_k + \alpha d_k$
Goldfarb	$\alpha s_k + \alpha^2 d_k$
Forsgren & Murray	$\alpha(s_k + d_k)$

We can prove convergence to second order points with these arcs. However, it is not clear how to scale d_k and we still have to modify H_k to get s_k.

Courant's Address in 1941

On May 3rd, 1941, Richard Courant gave an address to the American Mathematical Society in which he proposed three variational methods for numerically solving PDEs arising in problems of equilibrium and vibrations:

- The Rayleigh-Ritz method
- The method of finite differences
- The method of gradients

The idea behind the method of gradients is very old; Courant himself quotes a work of Jacques Hadamard published in 1908.

Variational Problems

- Problems of equilibrium and vibrations lead to
  - PDEs for an unknown function $u(x_1, x_2)$ in B, a bounded subset of $\mathbf{R}^2$.
  - Equivalent variational problems for the kinetic and potential energies of the system.
- Each of these PDEs has a functional E such that a sufficiently smooth minimizer of E satisfies the PDE.
- Example: the problem of determining the equilibrium of a membrane with given boundary deflections $\bar{u}(s)$ can be formulated as:

$$\Delta u = 0,\ u = \bar{u} \text{ on } \delta B$$

or

$$\underset{u}{\text{minimize}}\ E(u) = \int_B (u_{x_1}^2 + u_{x_2}^2)\, dx_1 dx_2$$

The Method of Gradients

Let u^* be a function defined in $B \subseteq \mathbf{R}^2$ and having prescribed boundary values such that u^* is the solution of a variational problem:

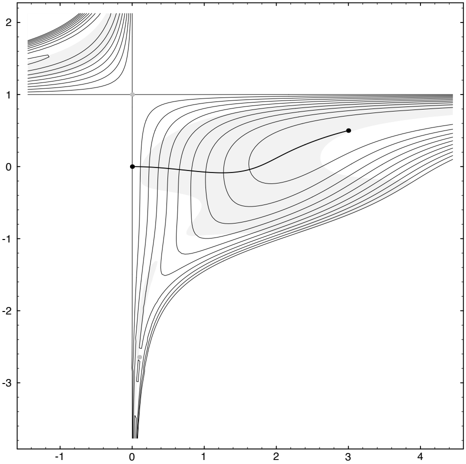
$$\underset{u}{\text{minimize}}\ E(u) = \int_B G(x_1, x_2, u, u_{x_1}, u_{x_2}) dx_1 dx_2$$

- Interpret u^* as $\lim_{t \to \infty} u(x_1, x_2, t)$.
- For $t = 0$, the value of u is chosen arbitrarily.
- For $t > 0$, the values of u are chosen in such as way that the expression $E(u)$, considered as a function of t, decreases as rapidly as possible toward its minimal value $E(u^*)$.

The Method of Gradients

- While historically much of the motivation for the method of gradients and the mathematical analysis behind it came from the desire to solve PDEs, the function it seeks to minimize need not come from a PDE.
- Its convergence theory requires knowledge of the objective function at a continuum of points.
- It has not found general acceptance because of the difficulties inherent in solving a system of nonlinear ODEs.

Example Gradient Flow



Gradient Flows

- The method of gradients starts with an initial point x_0 and seeks to find a minimizer of F by following the curve y defined by a system of n ODEs:

$$
\begin{aligned}
y'(t) &= -\nabla F(y(t)) \\
y(0) &= x_0
\end{aligned}
$$

- The solution is called an integral curve of $-\nabla F$ and is simply the curve that each instant proceeds in the direction of steepest descent of F.
- If this curve contains no stationary points then y has the desirable property that F is always decreasing along it.

Behrman's Method: Discrete & Linear Approximation

Solve a sequence of ODEs defined at iterates x_k:

$$\begin{aligned} y_k'(t) &= -\nabla F(y_k(t)) \\ y_k(0) &= x_k \end{aligned}$$

Replace y_k with a curve $x_k + w_k(t)$ by approximating the vector field $-\nabla F$ in a neighborhood of x_k with a vector field w_k. In particular, consider the linear vector field:

$$v_k(x) = -\nabla F(x_k) - \nabla^2 F(x_k)(x - x_k)$$

The curve w_k solves the following problem:

$$\begin{aligned} & w_k'(t) = v_k(x_k + w_k(t)) = -\nabla F(x_k) - \nabla^2 F(x_k) w_k(t) \\ & w_k(t) = 0 \end{aligned}$$

Solution of the ODE

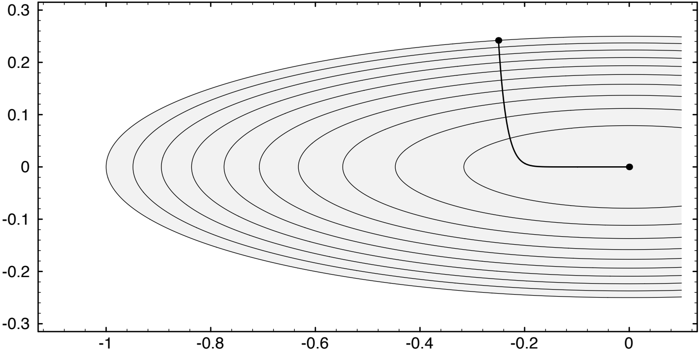
Given eigendecomposition $H_k = U\Lambda U^T$, the solution to the ODE is

$$\begin{aligned} w_k(t) &= -Uq(t,\Lambda)U^T g_k \\ q(t,\lambda) &= \begin{cases} -\frac{1}{\lambda}(e^{-\lambda t}-1) & \text{for } \lambda \neq 0 \\ t & \text{for } \lambda = 0. \end{cases} \end{aligned}$$

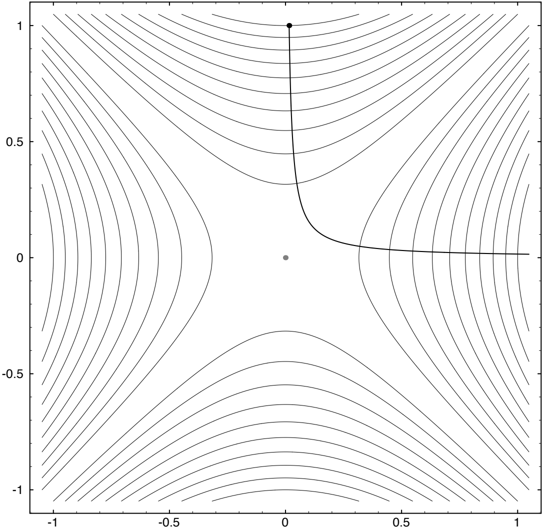
Properties:

- $w_k'(0) = -g_k$
- if $H_k \succ 0$, $\lim_{t\to\infty} w_k(t) = -H_k^{-1} g_k$
- if $H_k \not\succeq 0$, $w_k(t)$ diverges from saddle point as long as $\bar{U}^T g_k \neq 0$, where $\bar{U}$ are the columns of U corresponding to negative eigenvalues

Example: Positive Definite Hessian



Example: Indefinite Hessian



Problems with this arc

- Computing $H_k = U \Lambda U^T$ is expensive
- Function $q(t, \lambda)$ is annoying, must send the step parameter to ∞ to get Newton step
- Can't prove convergence to second order point because $\bar{U}^T g_k = 0$ is possible

Fix 1: reparameterize

Construct a function $t(s)$ by setting $s = q(t, \lambda_{\min})$ and solving for t. Here $\lambda_{\min}$ is the minimum eigenvalue of H_k. Now plug $t(s)$ into $q(t, \lambda)$. The result is

$$q(s,\lambda) = \begin{cases} ((s\lambda_{min} + 1)^{\lambda/\lambda_{\min}} - 1)/\lambda & \text{for } \lambda \neq 0 \\ \log(s\lambda_{\min} + 1)/\lambda_{\min} & \text{for } \lambda = 0. \end{cases}$$

Now if $\lambda_{\min} > 0$ then $s = 1/\lambda_{\min}$ gives the Newton step.
This works but should still give you a headache.

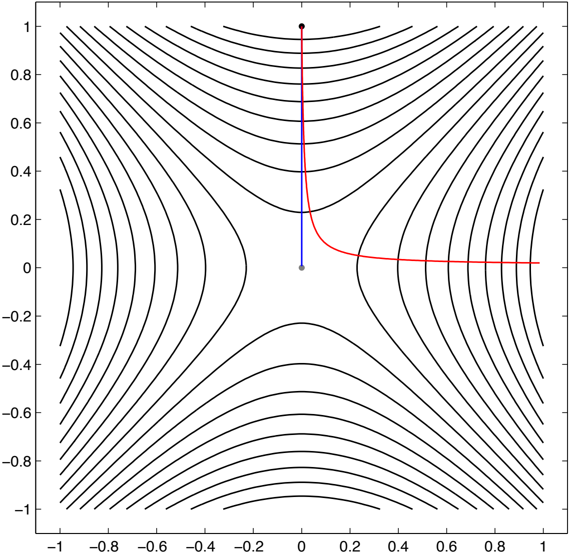
Fix 2: perturb

If g_k is entirely in a subspace spanned by the eigenvectors of H_k corresponding to positive eigenvalues, the arc defined by this ODE will not take advantage of negative curvature. This is analagous to the "hard case" in trust-region methods. In this case, the problem is solved by perturbing the inhomogenous term by a direction of sufficient negative curvature,

$$w_k(t) = -Uq(t, \Lambda)U^T(g_k + d_k).$$

We choose d_k such that $g_k^T d_k < 0$ and $\lim_{k \to \infty} d_k^T H_k d_k = 0$ implies $\liminf_{k \to \infty} \lambda_{\min}(H_k) \geq 0$.

Example: Perturbed curve



Fix 3: Subspace Method

The gradient flow algorithm described above requires an eigendecomposition of the Hessian at each iteration. This has a computation time of $O(n^3)$ and a storage requirement of $O(n^2)$.
Save time and space by constructing search arcs $\Gamma_k(t)$ in 2 or 3 dimensional subspaces.
In particular, define the subspace to be spanned by the columns of

$$S_k = [g_k \; p_k \; d_k]$$

where p_k is a (modified) Newton direction and d_k is a direction of negative curvature. If there is no negative curvature, use

$$S_k = [g_k \; p_k].$$

This was studied by Del Gatto in his PhD work.

Implementing the “projection”

1. Obtain an orthonormal basis with the QR factorization:

$$QR = S_k$$

2. Project the gradient and Hessian:

$$\bar{g} = Q^T g_k, \quad \bar{H} = Q^T H_k Q$$

3. Solve the 2D or 3D ODE:

$$w_k'(t) = -\bar{g} - \bar{H} w_k(t), \quad w_k(0) = 0$$

4. Update the iterate:

$$x_{k+1} = x_k + Q w_k(t)$$

Properties of the subspace search arc

- $Qw(0)$ is the steepest descent direction at x_k.
- If $\nabla^2 F(x_k)$ is positive definite, $Qw(t)$ converges to the Newton step.
- If $\nabla^2 F(x_k)$ is not positive definite, then $Qw(t)$ diverges along the eigenvector corresponding to the smallest eigenvalue.

Problems with Linear Equality Constraints

$$\begin{array}{ll} \underset{x}{\text{minimize}} & F(x) \\ \text{subject to} & Ax = b \end{array}$$

Can be solved with linesearch in the nullspace of A:

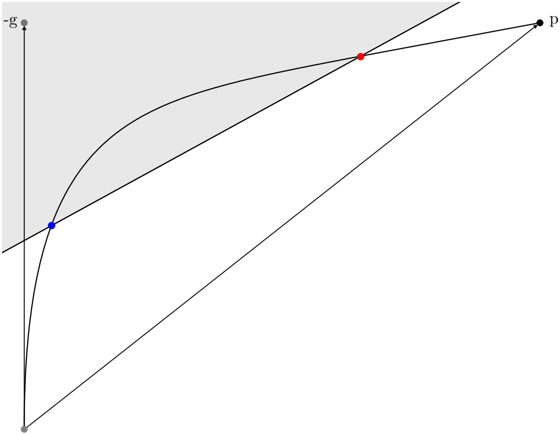
- Start with a feasible point x_0, such that $Ax_0 = b$.
- $x_{k+1} = x_k + \alpha_k p$ is feasible if $p \in \mathbf{null}(A)$.
- Say $p = Zp_z$, with $AZ = 0$. The columns of Z span $\mathbf{null}(A)$.
- Compute p_z by solving $Z^T H_k Z p_z = -Z^T g_k$.
- Terminate when $Zg = 0$ and $Z^T H Z$ positive semidefinite.

Problems with Linear Inequality Constraints

$$\begin{array}{ll} \underset{x}{\text{minimize}} & F(x) \\ \text{subject to} & Ax \geq b \end{array}$$

- Active set methods solve this problem by finding a set of constraints (rows of A) that give equality at the solution, $\overset{*}{A}\overset{*}{x} = \overset{*}{b}$.
- During the procedure, the algorithm must be able to:
  - add a constraint when one is hit by the search procedure.
  - delete a constraint when doing so allows a decrease in the objective function.

Intersecting a Linear Constraint



In this example: $-a^T g < 0$ and $p^T a < 0$

The non-ascent pair is $(-g, p)$ and the Hessian is positive definite.

Solve for the intersection

- It's easy for line search, just solve n univariate linear equations.
- For Behrman's method, we'd have to solve

$$\sum_{i=1}^{m} a_i \exp(b_i t) + ct + d = 0.$$

- For Del Gatto's subspace reduction (in 2D) it looks like

$$a_1 e^{b_1 t} + a_2 e^{b_2 t} + c = 0.$$

- Interesting equations, but I don't know of a good way to solve them.

Constraint intersection table for 2D ODE

Non-ascent pair: (s, d)
Linear constraint: $a^T x \geq b$

$s^T a$	$d^T a$	$H \succ 0$	$H \not\succ 0$
$+$	$+$	0	0
-	-	2	1^*
$+$	-	1	1^*
-	$+$	2°	2°
0	$+$	0	0
0	-	1	1^*
$+$	0	0	0
-	0	2°	1

Table: First two columns show the sign of the inner products. Third and fourth columns give the number of intersections possible with marked exceptions.

$*$ indicates intersection must occur.

$\circ$ indicates single intersection only occurs at point of tangency.

The trust-region arc

The trust-region subproblem is:

$$\begin{array}{ll} \text{minimize} & w^T H_k w + g_k^T w \\ \text{subject to} & w^T w \leq \Delta^2. \end{array}$$

If we have the right Lagrange multiplier we can get the solution by solving

$$(H_k + \pi I)w = -g_k.$$

We can also think of the solution as an arc parameterized by Δ or π.

The trust-region arc

Given the eigen-decomposition $H_k = U \Lambda U^T$, the solution to the subproblem is

$$\begin{aligned} w(\pi) &= -U q(\pi, \Lambda) U^T g_k \\ q(\pi, \lambda) &= \frac{1}{\lambda + \pi}. \end{aligned}$$

This should look familiar. This q is much better. But we still need to reparameterize with

$$q(s, \lambda) = \frac{s}{s(\lambda - \lambda_{min}) + 1}.$$

Trust-region arc properties

$$
\begin{aligned}
w(s) &= -Uq(s,\Lambda)U^T g_k \\
q(s,\lambda) &= \frac{s}{s(\lambda-\lambda_{min})+1}
\end{aligned}
$$

- $w(0) = 0$, just like the ODE
- $w'(0) = -g_k$, initially steepest descent
- If $H_k \succ 0$, then $s = 1/\lambda_{\min}$ gives the Newton step.
- If $H_k \not\succeq 0$, then $w(s)$ diverges away from saddle point.
- $d/ds\|w(s)\| > 0$ for all $s \geq 0$
- Can apply same perturbation to get second order convergence
- Can reduce to a 2D or 3D subspace and maintain properties

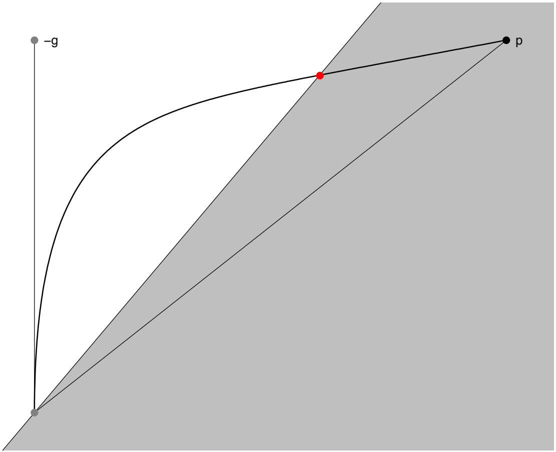
Trust-region arc-constraint intersection

For a trust region arc in a p-dimensional subspace, the arc-constraint intersection reduces to finding the largest root of a degree p polynomial. The math is easy. The inner product between the arc and constraint gives

$$\psi(\lambda) = \sum_{i=1}^{n} \frac{\alpha_i}{\lambda_i + \lambda} + \beta = 0.$$

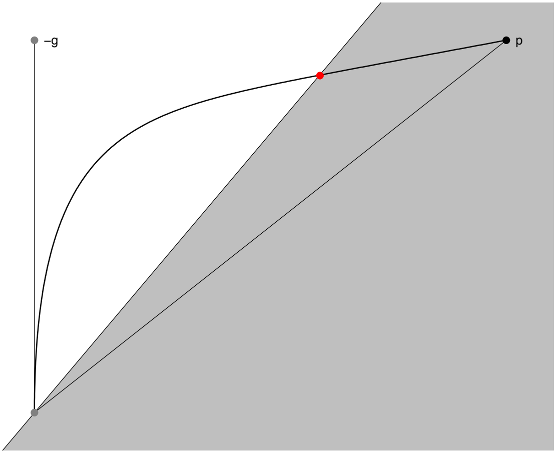
Simply multiply through by $\prod_{i=1}^{p}(\lambda_i + \lambda)$ and do some arithmetic.

Deleting a constraint: the arc may come back



This complicates the convergence theory.

Deleting a constraint: the arc may come back



This complicates the convergence theory. But, it's been worked out.

Concluding Remarks

- Arcs allow greater freedom when thinking about optimization algorithms.
- Trust-region arcs and ODE arcs are similar in spirit to the method of gradients, but only require small eigen-decompositions at each iteration.
- Trust-region arcs can be used in active-set algorithms, because it is easy to compute the intersection with a linear constraint.
- A general convergence theory (not tied to a particular arc) has been worked out.
- Methods can be applied to problems with many variables.