Geometry Optimization

C. David Sherrill

School of Chemistry and Biochemistry

Georgia Institute of Technology

Optimization

Many problems in computational chemistry (and scientific computing in general!) are optimization problems: i.e., *finding the "stationary points" where a multidimensional function has vanishing gradients*.

- The energy as a function of nuclear coordinates. Minima, transition states may be of interest.
- Error functions depending on parameters (e.g., fitting molecular mechanics methods).
- Variational electronic structure methods like Hartree-Fock, MCSCF, CI. Minima are ground states, saddle points are excited states. (Usually done as matrix diagonalization).

Optimization Methods

- Dumb method: step one variable until the function is a minimum in this dimension, then go on to other variables, repeat until no change. This won't normally work for complicated functions/many variables.
- Most methods assume an analytic (i.e., directly calculated, not numerical) gradient is available.
- Some methods assume the Hessian (second derivative) or an approximation of it is also available.

Steepest Descent

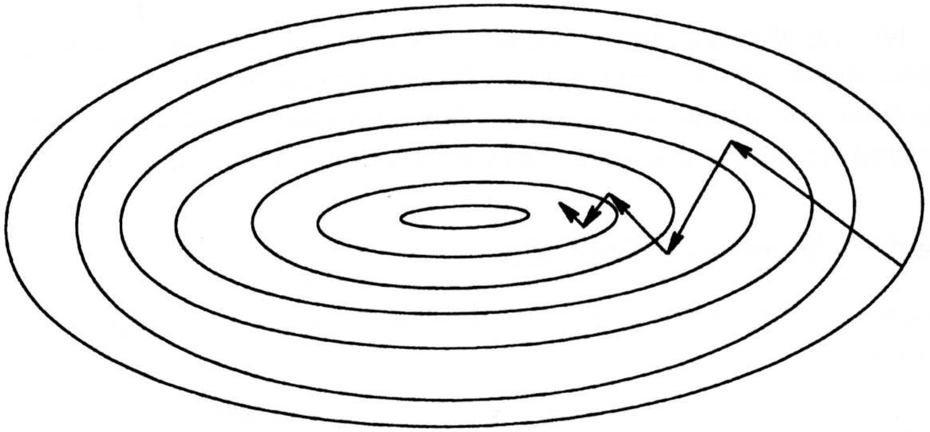

Figure 14.1 Steepest descent minimization

- Step in gradient direction, do a “line minimization” until the minimum in that direction is reached; repeat.
- Simple approach guaranteed to find minimum but may be slow; suffers from zig-zags.

Conjugate Gradient Methods

- Improvement on Steepest Descent; attempts not to spoil previous progress
- Do line minimizations in a direction which is a combination of the current gradient and the previous one

  $$\mathbf{d}_i = -\mathbf{g}_i + \beta_i \mathbf{d}_{i-1}$$

  The different conjugate-gradient methods provide different ways to choose β; they involve dot products of current and previous gradients, e.g., Polak-Ribiere:

  $$\beta_i^{PR} = \frac{\mathbf{g}_i^{\dagger}(\mathbf{g}_i - \mathbf{g}_{i-1})}{\mathbf{g}_{i-1}^{\dagger}\mathbf{g}_{i-1}}$$

Newton-Raphson Methods

Expand the energy function to second order around the current geometry $\mathbf{x}_0$:

$$E(\mathbf{x}) \approx E(\mathbf{x}_0) + \mathbf{g}^\dagger(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^\dagger \mathbf{H}(\mathbf{x} - \mathbf{x}_0)$$

If we find the stationary point ($dE/d\mathbf{x}$) of this approximate energy expression, we obtain

$$(\mathbf{x} - \mathbf{x}_0) = -\mathbf{H}^{-1}\mathbf{g}$$

This makes intuitive sense...think about a 1D example like Harmonic Oscillator. Need larger step if (i) gradient is larger, or (ii) force constant is smaller. Exact step if quadratic PES and exact gradient, Hessian.

Newton-Raphson with Diagonal Hessian

Suppose the Hessian matrix **H** (second derivative of electronic energy with respect to nuclear coordinates) is diagonalized to give normal modes and eigenvalues

$$\mathbf{H} u_i = \epsilon_i \mathbf{u}_i . \tag{1}$$

The Newton-Raphson step in terms of the normal modes becomes

$$(\mathbf{x} - \mathbf{x}_0) = \sum_i \frac{-F_i}{\epsilon_i} \mathbf{u}_i , \tag{2}$$

where $F_i = \mathbf{g}^\dagger \mathbf{u}_i$, i.e., the component of the gradient in each normal mode direction.

Minimize/Maximize

The Newton-Raphson step

$$(\mathbf{x} - \mathbf{x}_0) = \sum_i \frac{-F_i}{\epsilon_i}\mathbf{u}_i, \tag{3}$$

minimizes along directions with a positive Hessian eigenvalue ϵ_i and maximizes along directions with a negative eigenvalue. Finds minimum if all ϵ_i are positive ($\mathbf{H}$ positive definite), otherwise a saddle point if some ϵ_i are negative. *Big problems if the Hessian has the wrong "local structure."* Can sometimes solve by *level shifting* to *make* positive definite:

$$(\mathbf{x} - \mathbf{x}_0) = \sum_i \frac{-F_i}{\epsilon_i - \lambda}\mathbf{u}_i. \tag{4}$$

Obtaining the Hessian

- Can compute analytically if the program allows it, but can be slow (can take 10 times as long as energy alone); cost scales at least as $\mathcal{O}(N^3)$
- Can approximate using tables or lower level computations
- Can improve approximate Hessian using gradient information gained during optimization: Davidon-Fletcher-Powell (DFP), Broyden-Fletcher-Goldfarb-Shanno (BFGS), Powell.

Choice of Coordinates

- Z-matrix coordinates: old-style, inefficient
- Cartesian coordinates: even worse, un-natural, not like normal modes, more of them ($3N$ vs $3N-6$)
- Internal coordinates: simple and symmetry-adapted
- Natural internal coordinates/redundant internal coordinates (Peter Pulay): similar to symmetry-adapted internals, commonly used now

Multiple minima

- For larger molecules, more minima become possible
- Thermodynamic properties arise from Boltzmann distribution of all energetically accessible minima
- Can be hard to locate all relevant minima; search depends on initial guess geometries (and Hessians)
- Methods to help find additional minima: genetic algorithms, molecular dynamics, Monte Carlo methods, simulated annealing

Constrained Optimization

- Sometimes it's useful to fix a geometrical parameter and perform the optimization subject to this constraint
- Can rewrite function to be minimized as a Lagrange function which is the original function minus a Lagrange multiplier times a constraint function $g = 0$

$$L(x_1, x_2, \cdots, x_N, \lambda) = f(x_1, x_2, \cdots, x_N) - \lambda g(x_1, x_2, \cdots, x_N)$$

- Constrained optimization not always well supported by the quantum chemistry programs

Transition State Optimization

- Use an eigenvector following method: always go uphill in the direction of the lowest Hessian eigenvector
- Usually need a good guess for geometry *and* Hessian
- Also need a theoretical method that doesn't break down around the transition state region
- Many fancy algorithms exist but most never make it into standard program packages

Example Optimization

We can set up a simple Hartree–Fock optimization in Psi4 like this:

```
molecule h2o {
  O
  H 1 0.98
  H 1 0.98 2 105.0
}

set basis cc-pVDZ
optimize(’scf’)
```

Example Optimization

The program will perform a Hartree–Fock energy and gradient computation at each step. The gradient is a vector with $3N$ elements (forces on x, y, and z coordinates for each atom), and looks like this:

```
-Total Gradient:
 Atom           X            Y            Z
 ------   -----------   -----------  -----------
   1       -0.000000     0.000000    -0.038851
   2        0.000000    -0.029592     0.019425
   3       -0.000000     0.029592     0.019425
```

(The units are hartree/bohr). Note the symmetry in the gradient.

Example Optimization

The optimizer converts the gradient into internal coordinates, computes the force along each internal coordinate, and computes a desired stepsize along each coordinate meant to reach the minimum (the stepsize is scaled by the current estimate for the Hessian).

```
--- Internal Coordinate Step in ANG or DEG, aJ/ANG or AJ/DEG ---
 ---------------------------------------------------------------------------
   Coordinate                 Previous        Force       Change         New
   ----------                 --------       ------       ------      ------
    1 R(1,2)          =       0.980000    -0.290848    -0.036009    0.943991
    2 R(1,3)          =       0.980000    -0.290848    -0.036009    0.943991
    3 B(2,1,3)        =     105.000000    -0.000367    -1.674663  103.325337
 ---------------------------------------------------------------------------
```

Example Optimization

Each time a step is taken, the gradient is recomputed, the Hessian is updated with any new gradient information, and the process is iterated until convergence is achieved (within some tolerance).

```
Measures of convergence in internal coordinates in au.
-------------------------------------------------------------------------------------- ~
 Step    Total Energy       Delta E        MAX Force    RMS Force    MAX Disp     RMS Disp  ~
-------------------------------------------------------------------------------------- ~
   1    -76.024678625   -76.024678625     0.035302     0.028958     0.068041     0.058066  ~
   2    -76.026968520    -0.002289895     0.004412     0.003764     0.024060     0.014425  ~
   3    -76.027032430    -0.000063910     0.000314     0.000307     0.001501     0.000955  ~
   4    -76.027032781    -0.000000350     0.000035     0.000029     0.000059     0.000050  ~
-------------------------------------------------------------------------------------- ~
```