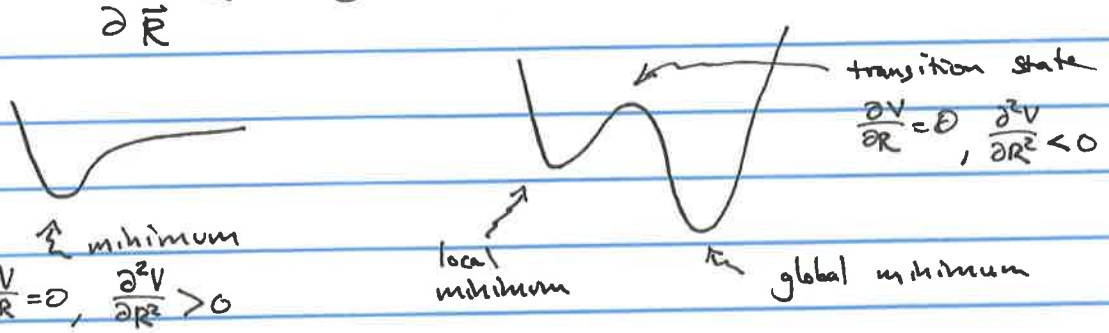
Now that we have an interatomic Potential Energy Surface (e.g. a force field), what do we do with it?

What do we want to know?

- conformation at local minima
- conformation at global minimum
- what structural changes occur as you go from structure 1 to structure 2.
- Saddle point geometries (transition states): the highest point on a minimum energy path connecting 2 local minima
- The reaction path itself
- Barrier heights, frequencies
- Relative energies, etc.

A smooth PES will have stationary points, that is

$$\frac{\partial V(\vec{R})}{\partial \vec{R}} = 0 \quad \text{for all coordinates } \vec{R}$$



In 1-D, finding these locations is a relatively easy problem. In higher dimensions, it is extremely hard.

In higher dimensions:

Minimum: A Stationary point $\left(\frac{\partial V}{\partial \vec{R}} = 0\right)$ with all positive Hessian Eigenvalues

$$\underline{\underline{\omega^2}} = \underline{\underline{U}}^T \cdot \underline{\underline{K}} \cdot \underline{\underline{U}}$$

$\underline{\underline{\omega^2}}$ ← frequencies (diagonal)

$\underline{\underline{K}}$ ← mass-weighted Hessian matrix

$$\underline{\underline{K}} = \begin{bmatrix} \frac{\partial^2 V}{\partial x_1^2}\frac{1}{m_1} & \frac{\partial^2 V}{\partial x_1 \partial y_1}\frac{1}{m_1} & \frac{\partial^2 V}{\partial x_1 \partial z_1}\frac{1}{m_1} & \frac{\partial^2 V}{\partial x_1 \partial x_2}\frac{1}{\sqrt{m_1 m_2}} \\ \frac{\partial^2 V}{\partial x_1 \partial y_1}\frac{1}{m_1} & \ddots & & \\ & & & \end{bmatrix}$$

← big $3N \times 3N$ matrix!

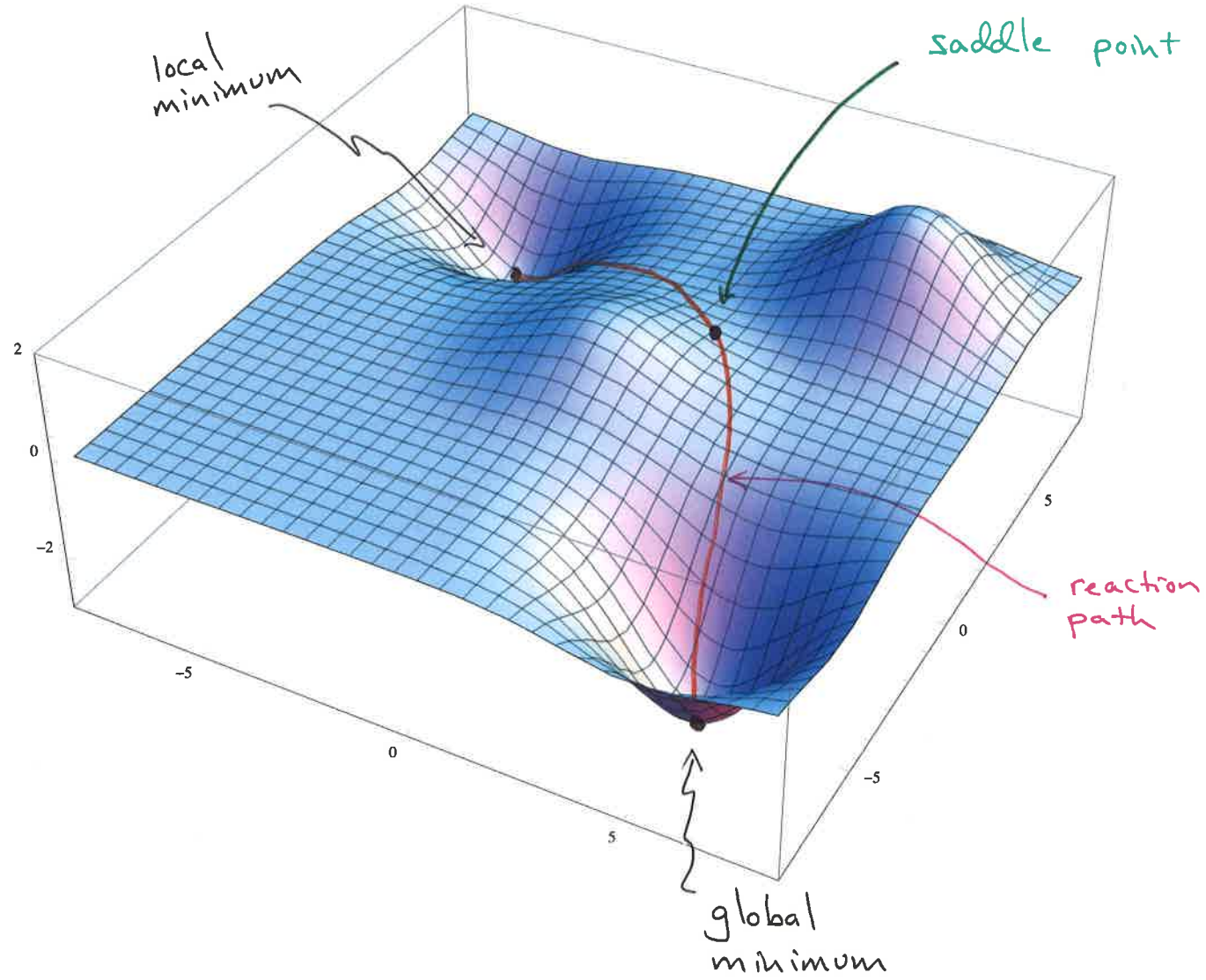
This is <u>hard</u> because:

- The PES is complicated for molecular systems
- It is highly non-linear (so guesses based on 1st derivatives are often incorrect)
- There are <u>many</u> degrees of freedom

  T4 lysozyme: 1328 atoms → 3984 coordinates

  Ethane: 8 atoms → 24 coordinates

Saddle: A Stationary point with 1 negative eigenvalue of the Hessian. ← an even harder problem!

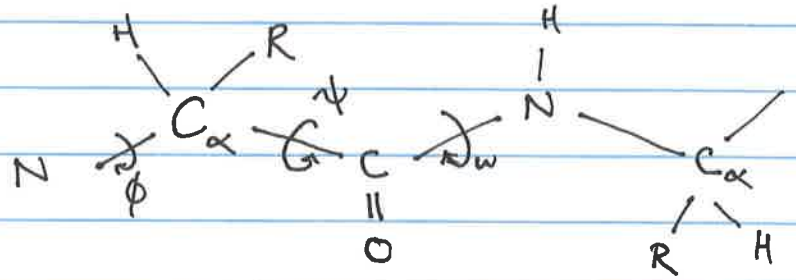

local minimum
saddle point
reaction path
global minimum
2
0
−2
−5
0
5
5
0
−5

Finding Minima & Saddle Points

A Difficult way: Grid-Based Searching

- Divide each coordinate up into a grid
- visit all grid points and calculate the energy
- tabulate the lowest energy you find.

Why is this a bad way? Consider proteins



Protein structure is governed by torsion angles (ϕ, ψ, ω)

Suppose we make a grid which divides up each torsion angles into 60° samples:

(0, 60, 120, 180, 240, & 300)

The total number of conformations we'd have to sample is then:

$$N_{conf} = \prod_{i=1}^{N_{tors}} \frac{R_i}{\delta_i}$$

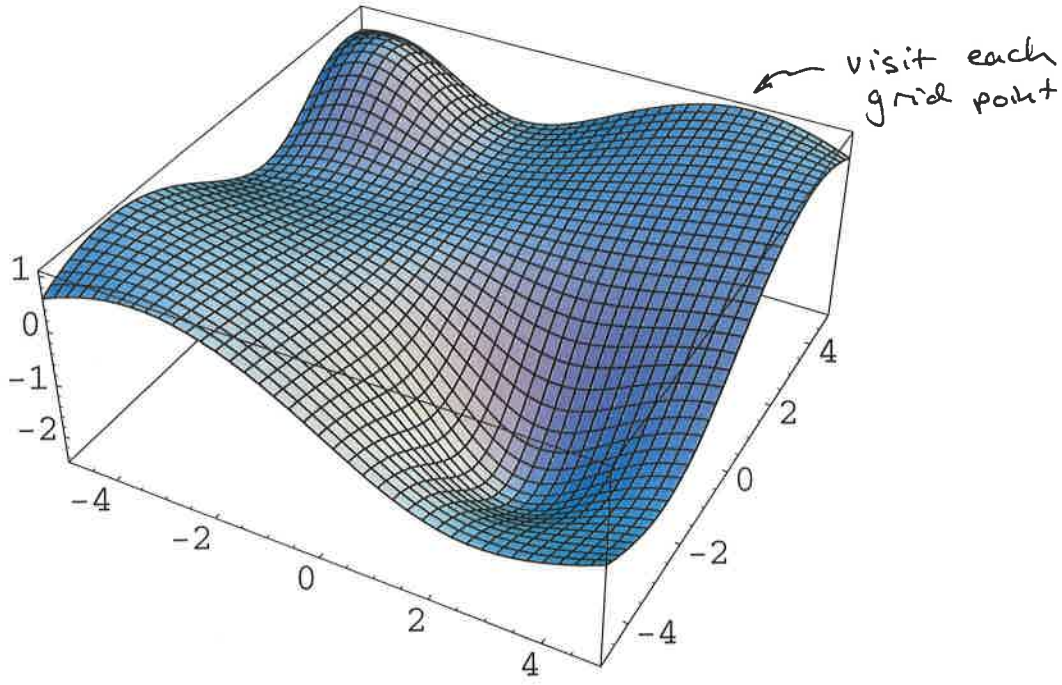
(N_{tors}: total # of torsions in the protein; R_i: Range of torsion i; δ_i: increment for torsion i)

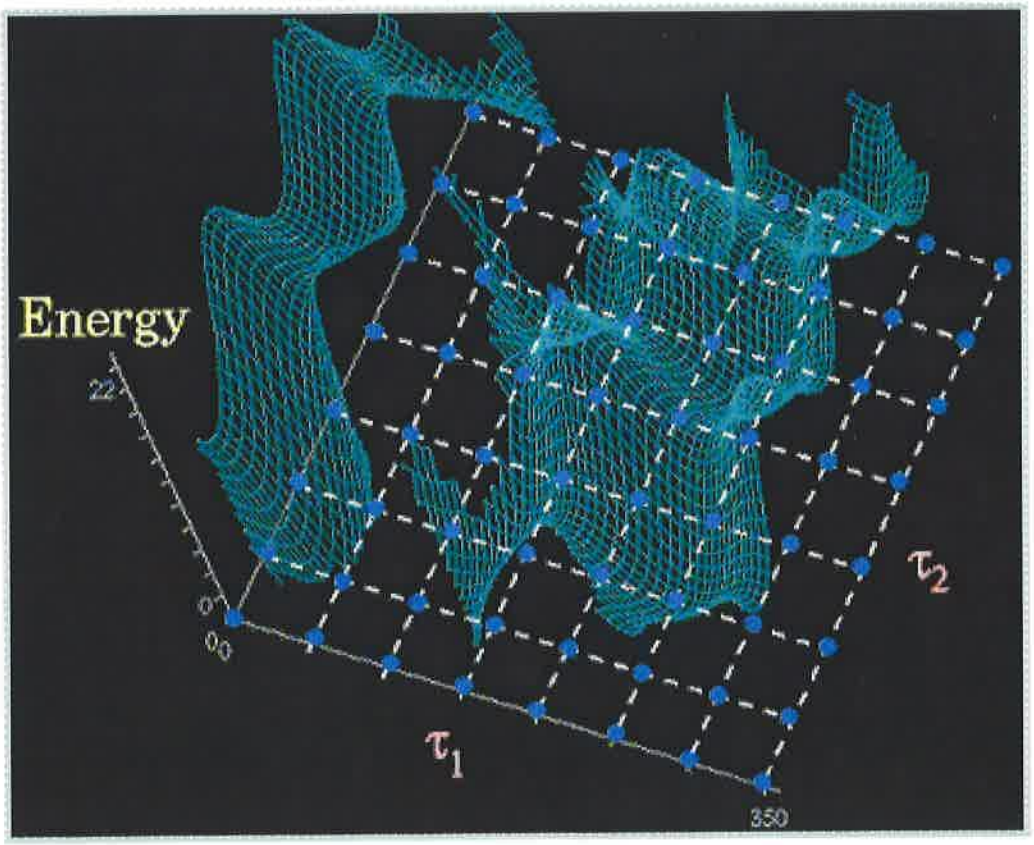
Let's be lazy and only visit the ϕ angles for each amino acid residue:

(N_{AA}: 1 ϕ per amino acid)

$$N_{conf} = \left(\frac{360}{60}\right)^{N_{AA}} = 6^{N_{AA}}$$

Conformational searches the difficult way





A typical (small) protein might have 100 amino acid residues:

$$N_{conf} = 6^{100} = 6.53 \times 10^{77}$$

If we were able to visit each conformation for 1 fs (10^{-15} s) ← this is very fast, either computationally or in real life

the total time required to visit all conformations

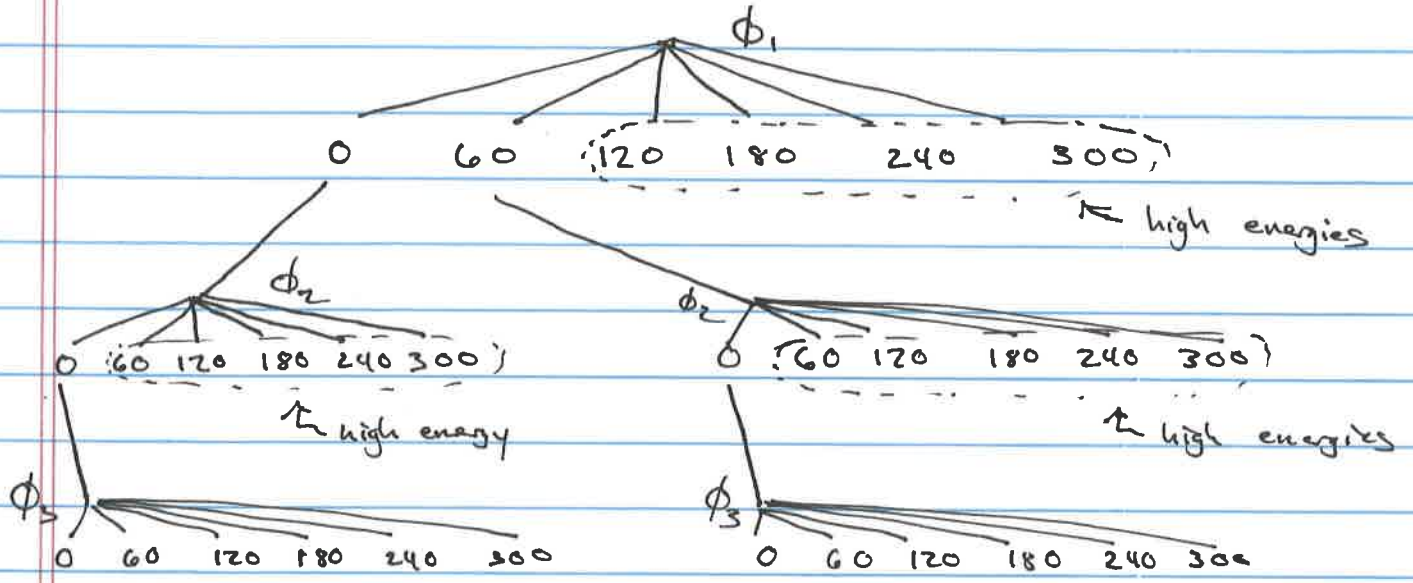
$$T = N_{conf} \cdot t_{conf} = 6.53 \times 10^{77} \times 10^{-15} \text{ s}$$
$$= 6.53 \times 10^{62} \text{ s}$$
$$= 2.07 \times 10^{55} \text{ years}$$

(as a comparison, the age of the universe is $\sim 1.5 \times 10^{10}$ years)

This is sometimes called Levinthal's paradox. Never mind the computational problem of finding the global minimum. How can a protein find the global minimum?

The Solution to Levinthal's paradox resides in knowing that not all conformations are going to lead to realistic or low-energy parts of the potential energy surface.

Consider the torsion angle tree:



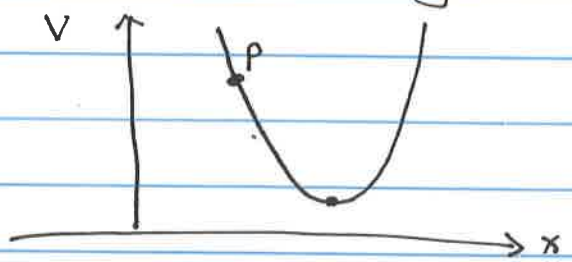
We can "prune" the tree when we hit a torsion angle that sends us into the high-energy portion of the landscape. Perhaps a value of 240° on torsion 1 makes the energy high due to local steric interactions. If we prune this branch of the tree, we've reduced the total # of conformations dramatically.

Real proteins probably find the global minimum using an "algorithm" similar to this one.

However, even once we get near the global minimum we have only explored the system in 60° increments the real minimum might have some angles at 52° or 34°. We need methods to refine our conformational searches once we have good candidates Refining the search down to a local minimum is called "minimization".

Minimization - Refining a conformation to find the **local** minimum:

With only 1 variable this is a relatively easy problem:



We can use a taylor approximation around our current guess:

$$V(x) = V(P) + V'(P)\cdot(x-P) + \frac{1}{2}V''(P)(x-P)^2$$

Here $V(x)$ is the quadratic approximation to the real potential using only information at our current location. Taking the 1st derivative and setting it to zero, we get an estimate for the location of the minimum:

$$x = P - \frac{V'(P)}{V''(P)}$$

Doing this over & over again is sometimes called Newton's method!

In more dimensions, this becomes more difficult

$$V(\vec{x}) = V(\vec{P}) + \vec{b}\cdot(\vec{x}-\vec{P}) + \frac{1}{2}(\vec{x}-\vec{P})^T\cdot\underline{\underline{K}}\cdot(\vec{x}-\vec{P})$$

$\vec{b}$: vector of 1st derivatives

$\underline{\underline{K}}$: matrix of 2nd derivs!

The major minimization methods all start here, but make **different** approximations about $\vec{b}$ & $\underline{\underline{K}}$ at each step.

1. Calculate $\vec{b}$ at each step, but don't keep track of $\vec{b}$ from previous steps
2. Calculate $\vec{b}$ at each step, and use previous $\vec{b}$ values to help make steps
3. Calculate $\vec{b}$ & $\underline{\underline{K}}$ (or an approximation to $\underline{\underline{K}}$) and use information from previous steps.

Let's talk about some minimization strategies:

1. Sequential Univariate:
   - minimize along 1 coordinate (e.g. torsion angle)
   - move on to another coordinate...
   - Changes in later coordinates may mess up the minimization in earlier coordinates!

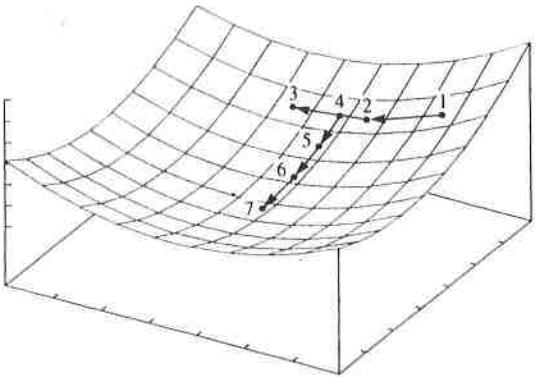
2. Line Minimization: Minimize along a line that is not necessarily a single coordinate
   - Then switch to a perpendicular line
   - Line searching is effectively 1-D: we can use Newton-like quadratic fits to speed up minimization.

3. Steepest Descent:

$$\vec{g}_k = \left.\frac{\partial V}{\partial \vec{x}}\right|_{\vec{x}=\vec{x}_k} = \text{force or gradient at point } \vec{x}_k$$

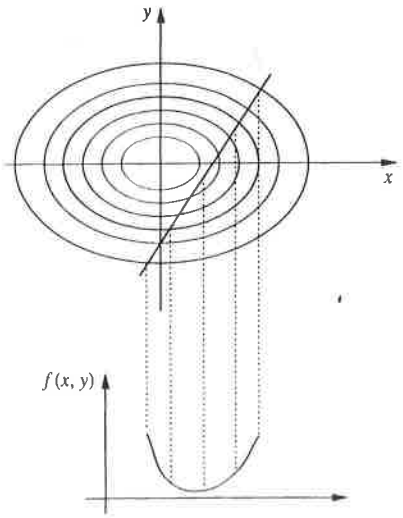
Direction of Step:

$$\vec{s}_k = \frac{\vec{g}_k}{|\vec{g}_k|}$$ ← unit vector pointing in same direction as force



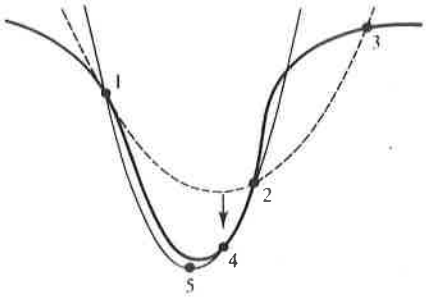
Sequential Univariate

- minimize along one degree of freedom
- switch to another degree of freedom



Line Minimization

- Minimize along a line, not necessarily one coordinate
- Switch to a perpendicular line



- use quadratic fits to get line minimizations done quickly.

Length of step: Arbitrary, but we usually use a line minimization to find the minimum along the gradient direction.

The next step is perpendicular to the previous step.

Like minimization without gradients can use only energies + dynamic step sizes:

$$\vec{x}_{k+1} = \vec{x}_k + \lambda_k \vec{s}_k$$

($\vec{s}_k$: direction; λ_k: step size)

If $V(\vec{x}_{k+1}) > V(\vec{x}_k)$ then $\lambda_{k+1} = \lambda_k * \frac{1}{2}$

If $V(\vec{x}_{k+1}) < V(\vec{x}_k)$ then $\lambda_{k+1} = \lambda_k * 1.2$

Refines step size so that we get to minimum more efficiently!

Steepest Descent is one of the most common minimization techniques. However, it is slow in long, narrow valleys.

Consider what we would do with Newton's method in many dimensions:

$$V(\vec{x}) = V(\vec{p}) + \vec{b}(\vec{x}-\vec{p}) + \frac{1}{2}(\vec{x}-\vec{p})^T \cdot \underline{\underline{K}} \cdot (\vec{x}-\vec{p})$$

$$V'(x) = \vec{b} + \underline{\underline{K}} \cdot (\vec{x}-\vec{p}) = 0$$

$$\underline{\underline{K}} \cdot (\vec{x} - \vec{P}) = -\vec{b}$$

$$(\vec{x} - \vec{P}) = -\underline{\underline{K}}^{-1} \cdot \vec{b}$$

$$\vec{x} = \vec{P} - \underline{\underline{K}}^{-1} \cdot \vec{b}$$

So, if we had the computer power and time we could get a reasonable guess at the minimum location using information at our current location

$$\vec{X}_{k+1} = \vec{X}_k - V''(\vec{X}_k)^{-1} \cdot V'(\vec{X}_k)$$

(V': vector of 1st derivatives!; V'': matrix inverse of 2nd derivative matrix)

The Conjugate gradient method tries to make steps that approximate this 2nd derivative method by using the 1st derivative at previous steps!

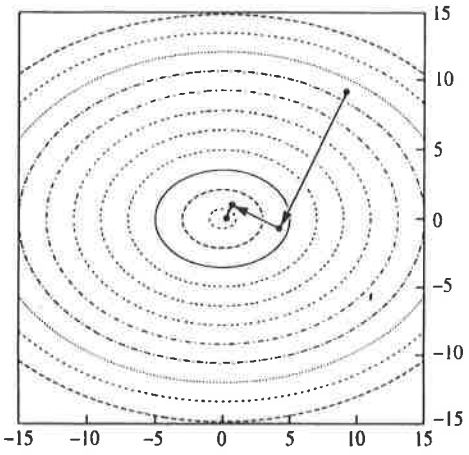
Direction of Step: $\vec{V}_k = -\vec{g}_k + \gamma_k \vec{V}_{k-1}$

($\vec{V}_{k-1}$: direction of previous step; $\vec{g}_k$: gradient at $\vec{X}_k$; γ_k: "weight" of previous step's direction)

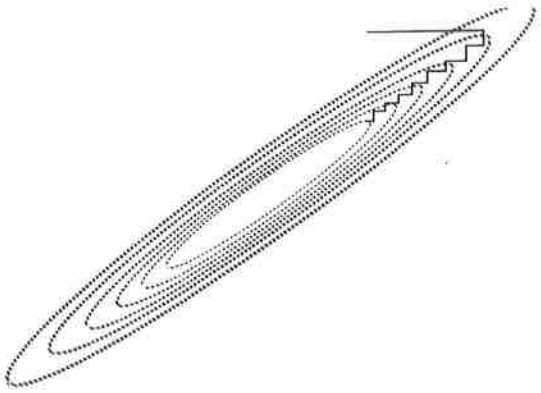
$$\gamma_k = \frac{\vec{g}_k \cdot \vec{g}_k}{\vec{g}_{k-1} \cdot \vec{g}_{k-1}} \leftarrow \text{"Fletcher-Reeves" form}$$

$$\gamma_k = \frac{(\vec{g}_k - \vec{g}_{k-1}) \cdot g_k}{\vec{g}_{k-1} \cdot \vec{g}_{k-1}} \leftarrow \text{"Polak-Ribiere" form}$$

Steepest Descent



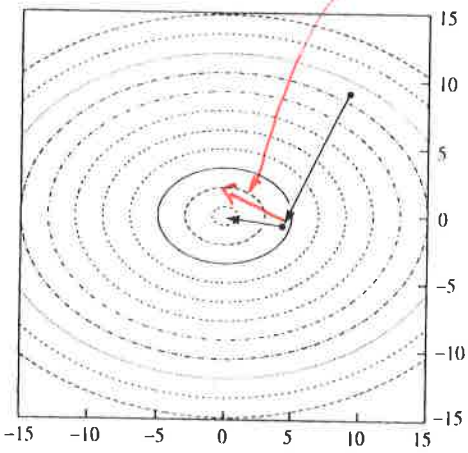
- move in a direction parallel to net force (perpendicular to contour lines) - that is: walk down hill.
- Step size is related to magnitude of the force



- Steepest descent requires many steps in long, narrow valleys

SD is ⊥ at each step!

Conjugate Gradient



- This method uses information about the direction of previous steps to modify the step we take
- It is very fast at refining estimates of local minima

What conjugate gradient accomplishes

- Much faster convergence to local minima, particularly on smooth potential energy surfaces.

Non-derivative methods ($\frac{\partial V}{\partial \vec{x}}$ is hard to compute)

Simplex

Allowed moves:

1) Reflect

reflect highest point through surface connecting the other points.

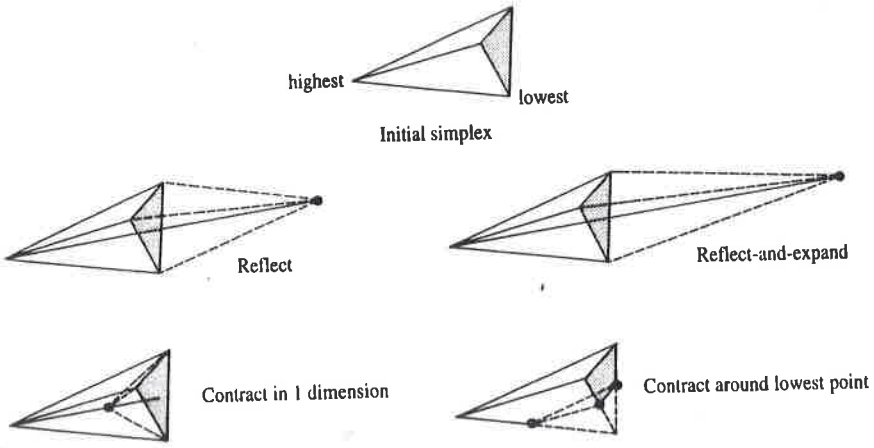
2) Reflect & Expand

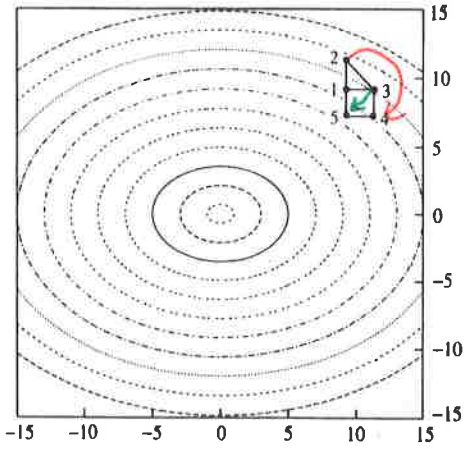
3) Contract in 1D:

4) Contract around lowest point:

Simplex requires lots of energy evaluations, but no gradients!

Simplex Method





Last time we talked about some common minimization techniques:

Non-Derivative: Using only potential energy at each step, $V(\vec{x}_k)$

- Sequential Univariate
- Line Minimization
- Simplex

Derivative Methods: Using Potential & First derivatives: $V(\vec{x}_k)$, & $\vec{b}_k = \left.\frac{\partial V}{\partial \vec{x}}\right|_{\vec{x}=\vec{x}_k}$

- Steepest Descent
- Conjugate Gradient

Today, we'll consider 2 techniques that try to use Newton's Original idea:

$$V(\vec{x}) \cong V(\vec{P}) + \vec{b}\cdot(\vec{x}-\vec{P}) + \frac{1}{2}(\vec{x}-\vec{P})^T \cdot \underline{K} \cdot (\vec{x}-\vec{P})$$

Harmonic approximation to potential around $\vec{x}=\vec{P}$

$\underline{K}$ matrix of 2nd derivatives

To pinpoint the minimum:

$$V'(\vec{x}) = \vec{b} + \underline{K}\cdot(\vec{x}-\vec{P}) = 0$$

$$\underline{K}\cdot(\vec{x}-\vec{P}) = -\vec{b}$$

$$\vec{x} = \vec{P} - \underline{K}^{-1}\cdot\vec{b}$$

So we could create an iterative method if we have the computer power to generate (and invert) $\underline{\underline{K}}$ at each step:

$$\vec{x}_{k+1} = \vec{x}_k - \underbrace{V''(\vec{x}_k)^{-1}}_{\Sigma} \cdot V'(\vec{x}_k)$$

$V'(\vec{x}_k)$: vector of 1st derivatives

Σ: inverse matrix of 2nd derivatives.

Σ This is called the "Newton-Raphson" method and it works well for small systems where we can compute $\underline{\underline{K}}$ easily.

BFGS is a method that approximates $\underline{\underline{K}}$ using multiple 1st derivative evaluations. Otherwise it is similar to Newton-Raphson

---

The choice of minimization method should be matched to the problem at hand:

- SD works best when gradients are large (far from minimum), but converges poorly close to the minimum
- CG generally finds the minimum in fewer steps than SD, but can create problems far from minimum.
- NR & BFGS are good for small numbers of variables, but can encounter serious problems when the starting point is far from the minimum.

Knowing when to stop : Convergence Criteria

$$\vec{g} = \frac{\partial V}{\partial \vec{x}}$$ ← close to a minimum, this should get close to 0

$$RMS = \sqrt{\frac{\vec{g}^T \cdot \vec{g}}{3N}}$$ ← units $\frac{kcal}{mol\ \text{Å}}$

When the RMS falls below a value that we set we're done!

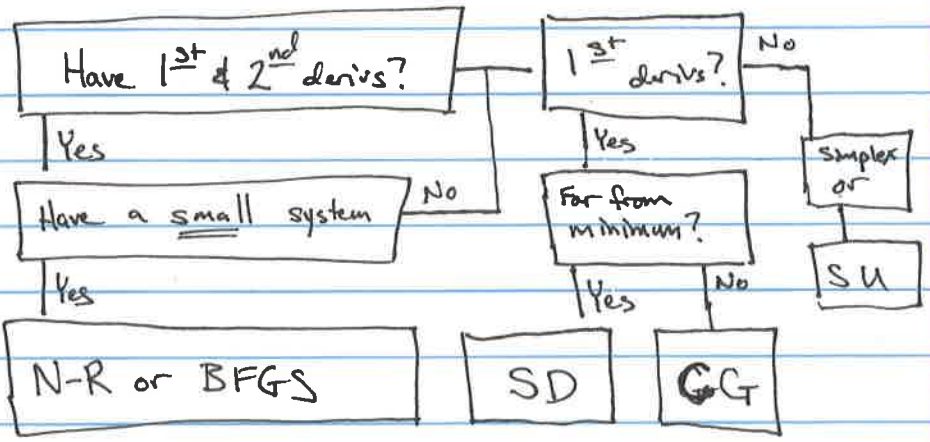
- OR when ΔE between points falls below fixed value
- OR when $\vec{x}_{k+1} - \vec{x}_k$ is sufficiently small

Convergence & Time :

	1 kcal/mol Å	< 0.1 $\frac{kcal}{mol\ \text{Å}}$
SD	67 s	1405 s
CG	149 s	257 s

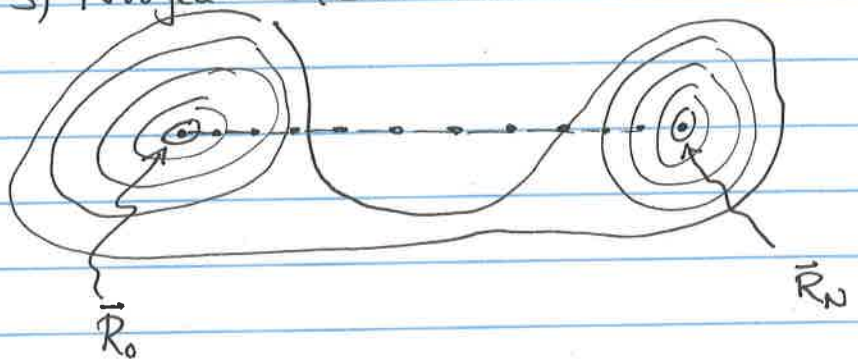
≷ more efficient at "fine-grained" minimization
fast for "rough" minimization

A flowchart:



Finding Saddle Points

1) Shallowest Ascent (numerically unstable)
2) Dragging Method (shallowest ascent + relaxation)
3) Nudged Elastic Band



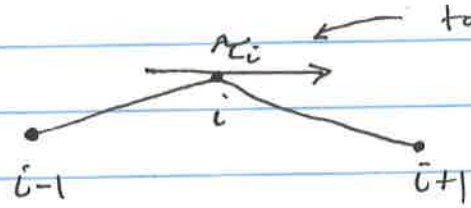
$$S(\vec{R}_0, \vec{R}_1 \cdots \vec{R}_N) = \sum_{i=1}^{N-1} V(\vec{R}_i) + \sum_{i=1}^{N} \frac{k}{2} (\vec{R}_i - \vec{R}_{i-1})^2$$

Σ harmonic springs between beads

We can minimize S with respect to all of the bead coordinates $\vec{R}_1, \vec{R}_2 \ldots \vec{R}_{N-1}$

S is the energy for an elastic band, and if we do this, the beads will slide down hill, and the path will cut some corners.

To fix this, the NEB method only allows the harmonic springs to act perpendicular to the direction of the real forces on these beads



tangent vector at i

$$\hat{\tau}_i = \frac{\vec{R}_{i+1} - \vec{R}_{i-1}}{|\vec{R}_{i+1} - \vec{R}_{i-1}|}$$

So the allowed force from the real potential must only include perpendicular contributions

$$\vec{g}_i = \frac{\partial V}{\partial \vec{R}_i} = \text{gradient at point } i$$

$$\vec{g}_i^{\perp} = \vec{g}_i - \underbrace{(\vec{\tau}_i \cdot \vec{g}_i)\vec{\tau}_i}_{\text{projection of force onto tangent}}$$

only the $\perp$ component of force to tangent

The elastic force is allowed to act only parallel to the tangent

$$f_i^{\parallel} = \underbrace{\left\{k\left[(\vec{R}_{i-1} - \vec{R}_i) - (\vec{R}_i - \vec{R}_{i-1})\right] \cdot \vec{\tau}_i\right\}}_{\text{project the elastic forces onto the tangent}} \vec{\tau}_i$$

Net force on bead:

$$\vec{f}_i = \vec{f}_i^{\parallel} + \vec{g}_i^{\perp}$$

Then we minimize using all of the $\vec{f}_i$ forces

The beads walk downhill in a direction perpendicular to the path, and the elastic forces keep the beads spaced evenly along the path.