Geometry optimization

Trygve Helgaker

Hylleraas Centre, Department of Chemistry, University of Oslo, Norway

European Summer School in Quantum Chemistry (ESQC) 2019
Torre Normanna, Sicily, Italy
September 8–21, 2019
Geometry optimization

- Good standard methods are available for minimization
  - Fletcher: “Practical Methods of Optimization” (2nd edn., 1987)
  - Gill, Murray, and Wright: “Practical Optimization” (1982)

- Methods for saddle points are much less developed
  - less intuitive and experimental information available for saddle points
  - many methods have been considered over the years but

  Localization of a saddle point is easy to make only in laboratories other than our own

  Havlas and Zahradník
Sections

1. Introduction
   - Smooth functions
   - Minima and saddle points
   - Strategies for optimization

2. Local region
   - Local region
   - Linear and quadratic models
   - Newton’s method
   - The quasi-Newton method
   - Convergence and stopping criteria

3. Global strategies for minimization
   - Global region
   - The trust-region method
   - The line-search method

4. Global strategies for saddle points
   - Saddle points
   - Levenberg–Marquardt trajectories
   - Gradient extremals
   - Image functions
Outline

1 Introduction
   - Smooth functions
   - Minima and saddle points
   - Strategies for optimization

2 Local region
   - Local region
   - Linear and quadratic models
   - Newton’s method
   - The quasi-Newton method
   - Convergence and stopping criteria

3 Global strategies for minimization
   - Global region
   - The trust-region method
   - The line-search method

4 Global strategies for saddle points
   - Saddle points
   - Levenberg–Marquardt trajectories
   - Gradient extremals
   - Image functions
Multivariate smooth functions

- Taylor expansion of a smooth function $f$ about the current point $x_c$:

$$
    f(x) = f_c + \tilde{s}g_c + \frac{1}{2}\tilde{s}H_c s + \cdots,
    \quad s = x - x_c
$$

- Multivariate function $f$ in $x$ with gradient $g_c$ and Hessian $H_c$ at $x_c$:

$$
    x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix},
    \quad g_c = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix},
    \quad H_c = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}
$$

- Diagonal Hessian representation:

$$
    f(x) = f_c + \sum_i \phi_i \sigma_i + \frac{1}{2} \sum_i \lambda_i \sigma_i^2 + \cdots
$$

- step in the diagonal representation $\sigma_i$
- gradient in the diagonal representation $\phi_i$
- diagonal Hessian with eigenvalues $\lambda_i$
- Hessian index: the number of negative Hessian eigenvalues
Stationary points

▶ A smooth function $f(x)$ has a **stationary point** at $x^*$ if the gradient vanishes:

$$g(x^*) = 0 \quad \text{(zero slope)}$$

▶ The function $f(x)$ has a **minimum** at $x^*$ (the minimizer) if the Hessian index is zero:

- small eigenvalue
- large eigenvalue
Strong and weak minima

- At a minimum, all Hessian eigenvalues are nonnegative
  - if, in addition, all eigenvalues are positive, we have a strong minimum
  - if one or more eigenvalues are zero, we have a weak minimum
Local and global minima

- A minimum $x_*$ is **global** if $f(x) \geq f(x_*)$ for all $x$
- A minimum $x_*$ that is not global is said to be **local**

Most practical methods do not discriminate between local and global minima
Saddle points

- A saddle point is a stationary point with one or more negative Hessian eigenvalues
  - a kth-order saddle point has Hessian index $k$

- The gradient and Hessian are both needed to characterize a stationary point

- Potential-energy surfaces:
  - minimum: stable molecular conformation
  - first-order saddle point: transition state

- Electronic-structure energy functions:
  - minimum: ground state
  - saddle point: excited state
Minima and saddle points
Strategies for optimization: global and local regions

- Any optimization is iterative, proceeding in steps or iterations
- At each step, a local model $m(x)$ is constructed of the surface $f(x)$
  - this model must be (locally) accurate, flexible, and easy to determine
- A search proceeds in two regions: the global region and the local region

Local region:
- the local model $m(x)$ represents $f(x)$ accurately around the optimizer $x^*$
- take a step to the optimizer of the model $m(x)$
- this region usually presents few problems

Global region:
- the local model $m(x)$ does not represent $f(x)$ accurately around the optimizer $x^*$
- the model cannot locate $x^*$ but must instead guide us in the right general direction
- relatively simple for minimizations, difficult in saddle-point searches
Section 2

Local region
Outline

1. Introduction
   - Smooth functions
   - Minima and saddle points
   - Strategies for optimization

2. Local region
   - Local region
   - Linear and quadratic models
   - Newton’s method
   - The quasi-Newton method
   - Convergence and stopping criteria

3. Global strategies for minimization
   - Global region
   - The trust-region method
   - The line-search method

4. Global strategies for saddle points
   - Saddle points
   - Levenberg–Marquardt trajectories
   - Gradient extremals
   - Image functions
In the local region, the local model extends to the optimizer $x_*$ of the true function.

We can then proceed in a simple manner:

1. Construct a local model $m_c(x)$ of $f(x)$ around the current point $x_c$

$$m_c(x_c) = f(x_c)$$
$$m_c(x_*) \approx f(x_*)$$

2. Determine the stationary point $x_+$ of the local model

$$\frac{d m_c(x)}{dx} \bigg|_{x=x_+} = 0$$

3. If $x_+ = x_*$ (to some preset threshold), terminate; otherwise, set $x_c = x_+$ and iterate again

The convergence of the optimization depends on the quality of the local model.

We shall build the local model by expansion around the current point

- the linear model
- the quadratic model
The *local linear or affine model* arises by truncation after the first-order term:

\[ m_A(x) = f(x_c) + \tilde{g}_c s \]

- The linear model is typically constructed from the *exact gradient*.
- The linear model is not very useful since
  - it is unbounded
  - it has no curvature information
  - it has no stationary points
- The linear model forms the basis for the *steepest-descent* method
  - it is often used in combination with line search (vide infra)
Second-order model

- In the second-order (SO) model, we truncate the expansion after second order:

\[ m_{SO}(x) = f(x_c) + \tilde{g}_cs + \frac{1}{2}\tilde{s}Hcs \]

- requires the exact gradient \( g_c \) and Hessian \( H_c \) at the current point
- The SO models contain full information about local slope and curvature

- Unlike the first-order (linear) model, the SO model has a stationary point
  - this point may or may not be close to the true stationary point
  - in the local region, the SO stationary point is close to the true stationary point
Newton’s method

- The SO model is given by
  \[ m_{SO}(x) = f(x_c) + \tilde{g}_c s + \frac{1}{2} \tilde{s} H_c s \]

- Differentiating the SO model and setting the result equal to zero, we obtain
  \[ \frac{dm_{SO}(s)}{ds} = 0 \quad \Rightarrow \quad g_c + H_c s = 0 \quad \Rightarrow \quad s = -H_c^{-1} g_c \]

- The new point \( x_+ \) and the current point \( x_c \) are related as
  \[ x_+ = x_c - H_c^{-1} g_c \quad \leftarrow \text{Newton step} \]

- When iterated, we obtain Newton’s method

- Note: the Newton step does not discriminate between minima and maxima
Convergence of Newton’s method

The relation between the new and old points is given by

\[ x_+ = x_c - H_c^{-1} g_c \quad \leftarrow \text{Newton step} \]

Subtracting the true optimizer \( x_* \), we obtain a relation between new and old errors

\[ e_+ = e_c - H_c^{-1} g_c, \quad e_+ = x_+ - x_*, \quad e_c = x_c - x_* \]

We next expand the gradient and inverted Hessian around the true optimizer \( x_* \):

\[ g_c = g_* + H_* e_c + \mathcal{O}(e_c^2) = H_* e_c + \mathcal{O}(e_c^2) \quad \text{(since } g_* = 0) \]

\[ H_c^{-1} = H_*^{-1} + \mathcal{O}(e_c) \]

Inserted in the error expression above, these expansions give

\[ e_+ = e_c - H_c^{-1} g_c = e_c - (H_*^{-1} + \mathcal{O}(e_c))(H_* e_c + \mathcal{O}(e_c^2)) = \mathcal{O}(e_c^2) \]

We conclude that Newton’s method converges quadratically

\[ e_+ = \mathcal{O}(e_c^2) \]

close to the optimizer, the number of correct digits doubles at each iteration
The quasi-Newton method

- If the exact Hessian is unavailable or expensive, use an approximate Hessian
  - this gives the more general quadratic model
    \[ m_Q(x) = f(x_c) + \tilde{g}_c s + \frac{1}{2} \tilde{s} B_c s, \quad B_c \approx H_c \]
  - the associated quasi-Newton step is given by
    \[ x_+ = x_c - B_c^{-1} g_c \]

- In the quasi-Newton method, the approximate Hessian is iteratively improved upon
  - at each iteration, the exact Hessian satisfies the relation
    \[ (g_+ - g_c) = H_+ (x_+ - x_c) + O((x_+ - x_c)^2) \]
  - by analogy, we require the new approximate Hessian to satisfy the relation
    \[ (g_+ - g_c) = B_+ (x_+ - x_c) \quad \leftarrow \text{the quasi-Newton condition} \]
  - the new Hessian is updated in a simple manner from \( B_c, g_+ - g_c \) and \( x_+ - x_c \)
    \[ B_+ = f(B_c, g_+ - g_c, x_+ - x_c) \]
  - several update schemes are available
Hessian updates

- Apart from the quasi-Newton condition, other conditions are often imposed
- Hereditary symmetry:

\[ B_c \text{ symmetric} \Rightarrow B_+ \text{ symmetric} \]

- Powell–symmetric–Broyden (PSB) update:

\[ B_+ = B_c + \frac{(\tilde{s}_c s_c) T_c \tilde{s}_c + (\tilde{s}_c s_c) s_c \tilde{T}_c - (\tilde{T}_c s_c) s_c \tilde{s}_c}{(\tilde{s}_c s_c)^2} \]

\[ T_c = (g_+ - g_c) - B_c s_c \]

- simple matrix and vector manipulations
- Hereditary positive definiteness:

\[ B_c \text{ positive definite} \Rightarrow B_+ \text{ positive definite} \]

- Broyden–Fletcher–Goldfarb–Shanno (BFGS) update:

\[ B_+ = B_c + \frac{y_c \tilde{y}_c}{\tilde{y}_c s_c} - \frac{B_c s_c \tilde{s}_c B_c}{\tilde{s}_c B_c s_c} \]

\[ y_c = g_+ - g_c \]

- Many other schemes exist
Convergence in local region

Consider a sequence $x_k$ that converges to $x^*$

$$\lim_{k \to \infty} x_k = x^* \quad \Leftarrow \text{convergent sequence}$$

$$e_k = x_k - x^* \quad \Leftarrow \text{error vector}$$

Linear, superlinear, and quadratic rates of convergence:

$$\lim_{k \to \infty} \frac{|e_{k+1}|}{|e_k|} = a \quad \Leftarrow \text{linear convergence} \quad \text{(steepest descent, gradient)}$$

$$\lim_{k \to \infty} \frac{|e_{k+1}|}{|e_k|} = 0 \quad \Leftarrow \text{superlinear convergence} \quad \text{(quasi-Newton, updated Hessian)}$$

$$\lim_{k \to \infty} \frac{|e_{k+1}|}{|e_k|^2} = a \quad \Leftarrow \text{quadratic convergence} \quad \text{(Newton, exact Hessian)}$$

The local region presents few problems for methods based on the quadratic model

- convergence to weak or near-weak minima will still be slow
- such minima require a quartic model for fast convergence

As our model improves, fewer but more expensive steps are needed for convergence
Stopping criteria

- An optimization is terminated when one or several convergence criteria are satisfied.
- Typically, the following criteria are used:
  - the gradient norm:
    \[ \| g_c \| \leq \varepsilon_g \]
  - the norm of the predicted second-order change in the function:
    \[ \frac{1}{2} | \tilde{g}_c H_c^{-1} g_c | \leq \varepsilon_f \]
  - the norm of the (quasi-)Newton step:
    \[ \| H_c^{-1} g_c \| \leq \varepsilon_s \]
- In addition, we should always check the structure of the Hessian (the Hessian index).
- Finally, inspect the solution and use common sense!
Section 3

Global strategies for minimization
Outline

1. Introduction
   - Smooth functions
   - Minima and saddle points
   - Strategies for optimization

2. Local region
   - Local region
   - Linear and quadratic models
   - Newton’s method
   - The quasi-Newton method
   - Convergence and stopping criteria

3. Global strategies for minimization
   - Global region
   - The trust-region method
   - The line-search method

4. Global strategies for saddle points
   - Saddle points
   - Levenberg–Marquardt trajectories
   - Gradient extremals
   - Image functions
Global region

- The local region is fairly simple — we now consider the more difficult global region.

Local region

- the local model $m(x)$ represents $f(x)$ accurately around the optimizer $x^*$
- take a step to the optimizer of the model $m(x)$
- the same method works for minima and saddle points

Global region

- the local model $m(x)$ does not represent $f(x)$ accurately around the optimizer $x^*$
- the model must guide us in the right general direction
- this is relatively simple in minimizations but difficult in saddle-point searches
Global strategies are needed when the local model represents $f(x)$ poorly around $x_*$.

- The Newton step above leads us away from the minimizer, increasing $f(x)$.
- The following is a useful global strategy for the minimization method:
  - the function $f(x)$ should be (sufficiently) reduced at each step.
- In addition, the method we seek should be globally convergent:
  - it should converge to some (possibly local) minimum from any starting point.
  - however, we cannot ensure that the minimum is global.
- There are two standard global strategies:
  - the trust-region method and the line-search method.
Global strategies for minimization

The trust-region method

In the trust-region method, we recognize that the second-order model is good only in some region around $x_c$: the trust region (TR)

The trust region cannot be specified in detail, we assume that it is a hypersphere

$$\sqrt{\tilde{s}s} \leq h \quad \leftarrow \text{trust radius } h$$

The trust radius is updated by a feedback mechanism

This gives us the restricted second-order (RSO) model

$$m_{SO}(x) = f(x_c) + \tilde{g}_c s + \frac{1}{2} \tilde{s}H_c s, \quad \tilde{s}s \leq h^2$$

At each iteration, we minimize $m_{SO}(x)$ subject to the constraint that $\tilde{s}s \leq h^2$
Stationary points in the trust region

- The trust region may or may not have a stationary point in the interior
- However, there are always two or more stationary points on the boundary
- Consider $f(x, y)$ below expanded about $(12, 8)$ in $s_x = x - 12$ and $s_y = y - 8$:
  \[ f(x, y) = 8(x - y)^2 + (x + y)^2 = 528 + [s_x, s_y] \begin{bmatrix} 104 \\ -24 \end{bmatrix} + \frac{1}{2} + [s_x, s_y] \begin{bmatrix} 18 \\ -14 \\ 18 \end{bmatrix} [s_x \ s_y] \]
- with trust radius $h = 10$, there are four stationary points on the boundary

- In the global region, we minimize $f$ globally on the boundary and go to point A
The level-shifted Newton step

To determine stationary points on the boundary, we use Lagrange’s method

\[ L(s, \mu) = m_{SO}(s) - \frac{1}{2} \mu(\tilde{s}s - h^2) \quad \leftarrow \text{Lagrangian} \]

The stationary points are now obtained by setting the gradient to zero

\[ \frac{dL}{ds} = g_c + H_c s - \mu s = 0 \quad \Rightarrow \quad s(\mu) = -(H_c - \mu I)^{-1} g_c \]

We obtain a level-shifted Newton step \( s(\mu) \) that depends on \( \mu \)

we select \( \mu \) such that the step is to the boundary

Note: we have always at least two stationary points on the boundary
The trust-region algorithm

1. Construct a restricted second-order model of the surface at $x_c$:

$$m_{RSO}(s) = f(x_c) + \tilde{g}_c s + \frac{1}{2} \tilde{s} H_c s, \quad \|s\| \leq h_c$$

2. Take the Newton step if $\|s(0)\| < h_c$ and if $H_c$ has correct structure

$$s(0) = -H_c^{-1} g_c$$

3. Otherwise, take the level-shifted Newton step to the minimum on the boundary

$$s(\mu) = -(H_c - \mu I)^{-1} g_c, \quad \mu < \min(0, \lambda_1), \quad \|s(\mu)\| = h_c$$

- The Levenberg–Marquardt trajectory: the step $s(\mu)$ as a function of $\mu$:
**Trust-radius update**

- The trust radius $h_c$ is updated by a feedback mechanism:

  $$R_c = \frac{\text{actual change}}{\text{predicted change}} = \frac{f_+ - f_c}{\hat{g}_c s + \frac{1}{2} \hat{s} H_c s} = 1 + \mathcal{O}(s^3)$$

- Important safety measure: step rejection
  - always reject the step if the function increases
  - calculate new step with reduced radius

- Typically implemented with the exact Hessian: an updated Hessian may not be accurate enough for an unbiased search in all directions
The line-search method

- If the Newton step must be rejected, it may still provide a direction for a line search.
- In the line-search method, such searches form the basis for the global optimization.

Line search

One-dimensional search in a descent direction until an acceptable reduction is obtained.

- A descent direction is a vector $z$ such that $\tilde{g}_c z < 0$.

Examples of descent directions:

- Steepest-descent step:
  \[ z = -g_c \text{ since } -\tilde{g}_c g_c < 0 \]
- Newton step with a pos. def. Hessian:
  \[ z = -B_c^{-1} g_c \text{ since } -\tilde{g}_c B_c^{-1} g_c < 0 \]
- BFGS step guarantees pos. def. Hessian.
- Newton step is usually better than the steepest-descent step.
Line searches

- **Exact line search:**
  - expensive, unnecessary

- **Inexact or partial line search:**
  - try Newton step first; then **backtrack** until an acceptable step is found

- Line searches are often used with updated Hessians: **quasi-Newton methods**
  - relatively stable and efficient

- Backtracking does not make full use of available information
  - the Hessian is used to generate the direction of the step but not its length
  - the coupling between direction and length is ignored

<table>
<thead>
<tr>
<th>trust-region method</th>
<th>line-search method</th>
</tr>
</thead>
<tbody>
<tr>
<td>first step size, next direction</td>
<td>first direction, next step size</td>
</tr>
<tr>
<td>handles indefinite Hessians naturally</td>
<td>handles indefinite Hessians poorly</td>
</tr>
<tr>
<td>less suited for updated Hessians</td>
<td>well suited for updated Hessians</td>
</tr>
<tr>
<td>“guaranteed” convergence</td>
<td>no guarantee of convergence</td>
</tr>
<tr>
<td>conservative</td>
<td>risky</td>
</tr>
</tbody>
</table>

Trygve Helgaker  (University of Oslo)
Coordinate systems for geometry optimizations

- A judicious choice of coordinates may improve convergence by reducing
  - quadratic couplings
  - higher-order (anharmonic) terms

Cartesian coordinates

- simple to set up and to automate
- provide universal and uniform quality
- yield strong couplings and large anharmonicities
- contain rotations and translations

Internal coordinates

- primitive internal coordinates: bond lengths, bond angles, dihedral angles
- physically well motivated: small couplings and anharmonicities
- nonredundant system difficult to set up
- solution: use redundant internal coordinates
- redundancies controlled by projections
The initial Hessian

- The efficiency of update methods depends on the quality of the initial Hessian
- The exact initial Hessian gives fewest iterations but is expensive
- A more efficient scheme may be to use a less accurate but cheaper initial Hessian
- A good approximate Hessian is easiest to set up in primitive internal coordinates
  - diagonal harmonic model Hessian

\[
B_{pp} = \begin{cases} 
0.45 \rho_{ij} & \text{bond length} \\
0.15 \rho_{ij}\rho_{jk} & \text{bond angle} \\
0.005 \rho_{ij}\rho_{jk}\rho_{kl} & \text{dihedral angle}
\end{cases}
\]

- here \(\rho_{ij}(r_{ij})\) is a decaying model function for each atom pair \(ij\)

\[
\rho_{ij}(r_{ij}) = \exp[\alpha_{ij}(R_{ij}^2 - r_{ij}^2)]
\]

\(\alpha_{ij}\) and \(R_{ij}\) tabulated for all atom pairs
Numerical comparisons

- Total number of iterations/timings for 30 representative molecules (Baker set)
- 1st-order quasi-Newton (BFGS) with different initial Hessians
- 2nd-order Newton method
- Optimizations in Cartesian and redundant internal coordinates

<table>
<thead>
<tr>
<th></th>
<th>Cart. diagonal</th>
<th>int. diagonal</th>
<th>exact</th>
</tr>
</thead>
<tbody>
<tr>
<td>quasi-Newton</td>
<td>1.0</td>
<td>0.4</td>
<td>1.0</td>
</tr>
<tr>
<td>Newton</td>
<td>hmh</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cartesian</td>
<td>iter.</td>
<td>768</td>
<td>318</td>
</tr>
<tr>
<td></td>
<td>time</td>
<td>2261</td>
<td>931</td>
</tr>
<tr>
<td></td>
<td>time</td>
<td>1475</td>
<td>781</td>
</tr>
<tr>
<td>internal</td>
<td>iter.</td>
<td>503</td>
<td>269</td>
</tr>
<tr>
<td></td>
<td>time</td>
<td>1475</td>
<td>781</td>
</tr>
<tr>
<td></td>
<td>time</td>
<td>1064</td>
<td>664</td>
</tr>
</tbody>
</table>

- The best method: the BFGS quasi-Newton method in redundant internal coordinates with initial harmonic model Hessian (hmh)
Section 4

Global strategies for saddle points
Outline

1 Introduction
- Smooth functions
- Minima and saddle points
- Strategies for optimization

2 Local region
- Local region
- Linear and quadratic models
- Newton’s method
- The quasi-Newton method
- Convergence and stopping criteria

3 Global strategies for minimization
- Global region
- The trust-region method
- The line-search method

4 Global strategies for saddle points
- Saddle points
- Levenberg–Marquardt trajectories
- Gradient extremals
- Image functions
Saddle points

- Saddle-point optimizations are more difficult than minimizations
  - less experimental and intuitive information available
  - less developed and stable
- There are a large number of methods in use
- The local region presents few problems provided a second-order model is used
  - the Newton step is always to the stationary point of the second-order model, be it a minimum, a maximum or a saddle point
- All difficulties with saddle-point optimizations are in the global region
  - it is hard to measure progress in saddle-point optimizations
Minima and saddle points
A simple approach is to explore other solutions to the restricted 2nd-order problem

\[ s(\mu) = - (H_c - \mu I)^{-1} g_c \]

Select walks to reduce or increase the function along the various modes

- note: the trajectories depend on the expansion point
- this approach has been used with some success
Gradient extremals

- Levenberg–Marquardt trajectories are dependent on the expansion point
- Are there well-defined lines connecting stationary points of a smooth function?
- Steepest-descent paths:
  - follow gradient down from the saddle point
  - not locally defined (not recognizable)
  - intrinsic reaction coordinate
- Gradient extremals:
  - connect stationary points
  - locally defined (recognizable) by the condition

\[ H(x)g(x) = \lambda(x)g(x) \]

- The gradient is an eigenvector of the Hessian at gradient extremals
From stationary points to gradient extremals

- Consider the gradient in the diagonal representation of the Hessian
  - at a stationary point, all elements are zero
  - at a gradient extremal, all elements except one are zero

\[
\phi(x_{sp}) = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad \rightarrow \quad \phi(x_{ge}) = \begin{bmatrix} 0 \\ \vdots \\ \phi(t) \end{bmatrix}
\]

- Gradient extremals are therefore points where we have relaxed just one of the conditions for a stationary point
  - \(3N - 6\) conditions specify a point
  - \(3N - 7\) conditions specify a line
- Only one nonzero gradient component in the eigenvector basis implies the condition
  \[H(x)g(x) = \lambda(x)g(x)\]
- It should be possible to follow gradient extremals between stationary points
Gradient extremals as optimum ascent paths

- A gradient extremal corresponds to an optimum ascent path

Optimize the gradient norm on a contour line \( f(x) = k \)

\[
\frac{d}{dx} [\ddot{g} g - 2\mu(f(x) - k)] = 0 \quad \Rightarrow \quad H(x)g(x) = \mu(x)g(x)
\]

- Some properties of gradient extremals:
  - locally defined, intersect at stationary points
  - not necessarily tangent to gradient, curves a lot and difficult to follow
Image functions

Imagine a function $\bar{f}(x)$ with the following properties:

<table>
<thead>
<tr>
<th>$f(x)$</th>
<th>$\bar{f}(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimum</td>
<td>saddle point</td>
</tr>
<tr>
<td>saddle point</td>
<td>minimum</td>
</tr>
</tbody>
</table>

the function $\bar{f}(x)$ is said to be the image function of $f(x)$.

We may locate a saddle point of $f(x)$ by minimizing $\bar{f}(x)$!

a trivial example:

In general, we cannot construct an image function—it may not even exist.

however, we know its gradient and Hessian.

this is sufficient for second-order optimizations.

Trygve Helgaker (University of Oslo)
Trust-region image minimization

- The gradient and Hessians of a function $f$ and its image $\bar{f}$ are related as

$$\phi(x) = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}, \quad \lambda(x) = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$\bar{\phi}(x) = \begin{bmatrix} -\phi_1 \\ \phi_2 \end{bmatrix}, \quad \bar{\lambda}(x) = \begin{bmatrix} -\lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

- To minimize $\bar{f}(x)$, we must use the trust-region method—line search is impossible

- The level-shifted Newton step for the image function is now given by

$$s(\mu) = - (\bar{H}_c - \mu 1)^{-1} \bar{g}_c = - \frac{\bar{\phi}_1}{\lambda_1 - \mu} \bar{v}_1 - \frac{\bar{\phi}_2}{\lambda_2 - \mu} \bar{v}_2$$

$$= - \frac{\phi_1}{\lambda_1 + \mu} \bar{v}_1 - \frac{\phi_2}{\lambda_2 - \mu} \bar{v}_2$$

- a simple sign change in the level-shift parameter $\mu$ for one mode
- the level-shifted Newton method maximizes this mode and minimize all others

- Trust-region image minimization is typically applied to the lowest Hessian mode
- robust but not selective