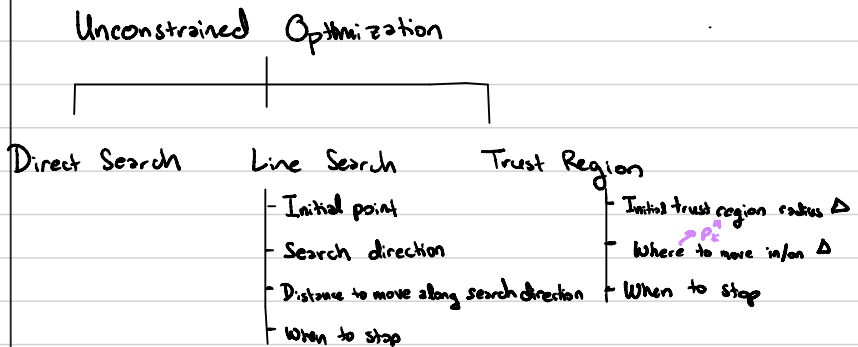
# TRUST REGION METHOD

## What is it?




- An iterative method to get the minimizer for an unconstrained optimization problem by guessing the shape of the objective function in a "neighborhood" design space (e.g. circle).

Trust Region Methods: "neighborhood"

vs.

Line Search Method: "search direction"

## Advantages?

- allows for indefinite ($\pm\lambda$) or semi-definite ($\pm\lambda, 0$) Hessian approx. (Line search does not!)

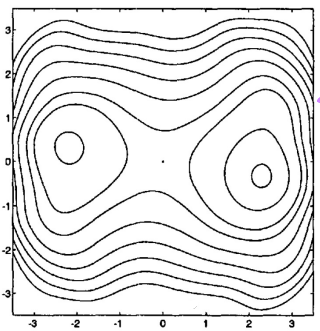
## Cons?

- trust region subproblem may need to be resolved several times in one iteration before obtaining an acceptable step
  ∴ total cost for one iteration may be expensive

## Main Points:

- Trust Region Method is iterative in that it builds better and better guesses (or iterates) $x_k$ ← search point of the solution of the problem.
- It produces a local solution

## Why care?

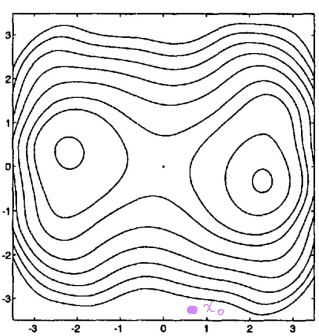


Say you have some objective function $f(x)$ that you want to minimize.

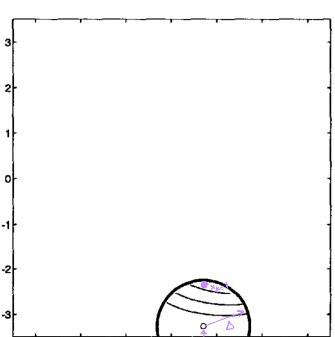
$$f(x_1, x_2) = -10x_1^2 + 10x_2^2 + 4\sin(x_1 x_2) - 2x_1 + x_1^4$$

To find the minimizer, you could plot the function (like above) and find the point $x_*$ where $f(x_*)$ is the lowest value.
But! This procedure is costly b/c you would need to compute the objective function at all possible points.

So what would be the trust region approach?




Assume at an initial guess of $x_0 = [0.7067, -3.2672]^T$, we only know the value of $f(x_0) = 97.6305$, its slope, and its curvature.




Based on this limited info, we build a model $m_k$ of the objective function and decide with some degree of arbitrariness on the region $\Delta$ containing $x_0$ in which we believe the model $m_k$ represents the objective function more or less adequately.

$m_k(x_k + p)$: model of the objective function

$\Delta$: trust region radius ← we "trust" the model to be a faithful representation of the objective function in this region.

The circle represents the trust region around $x_0$
Contour lines of this model are shown inside the circle.

Analyze above Plot

Examining the model within the trust region, we notice its value decreases as we move up $x_2$ (increase $x_2$ value).

We make a step $p_k$ towards a trial point $x_k$, where the decrease in the model (compared to its value at $x_0$) is significant.

∴ We have done all we could with the information at our disposal ⇒ compute the objective function @ new trial point $x_{k=1}$

@ new point $x_{k=1}$, we repeat steps:

1.) $x_{k=1}$ is our new best approx. minimizer

2.) Use slope and curvature of $f$ at $x_{k=1}$ to construct new updated model of the objective function.

$$m_k(x_k + p) = f(x_k) + g(x_k)^T + \frac{1}{2}p^T B_k p$$

where: $g(x_k) = \nabla f(x_k)$

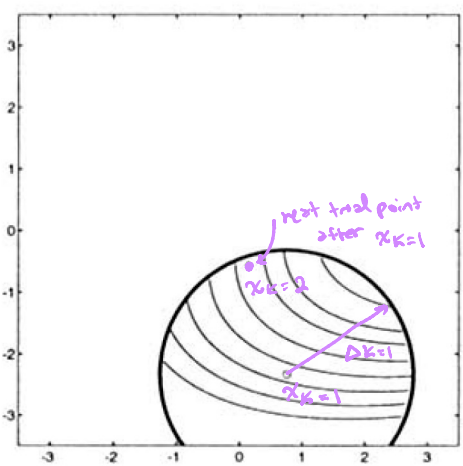
$B_k$: Hessian or approx of Hessian

$p$: step

$x_k$: current search point

3.) can change size of radius of the new trust region radius around the new point $x_{k=1}$.

e.g. increased $\Delta_{k=1}$

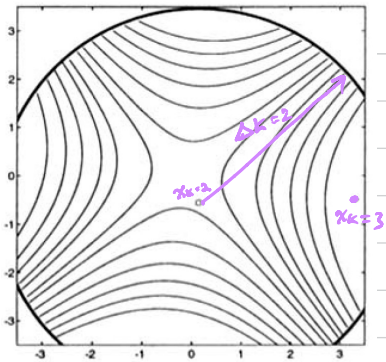



Analyze above Plot

Repeating analysis:

Take step $p_k$ towards a new trial point $x_{k=2}$

This new point $x_{k=2}$ is acceptable b/c reduces objective function.




Analyze above Plot

@ new point, we repeat the steps

1.) $x_{k=2}$ is our new point

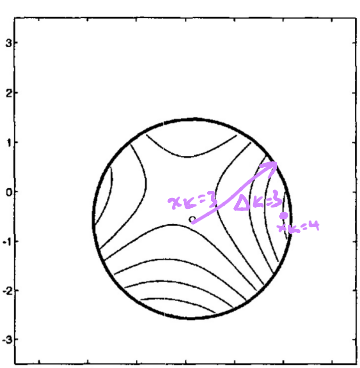
2.) use slope and curvature of f at $x_{k=2}$ to construct new updated model $m_k$ of the objective function.

3.) decide to increase $\Delta$

We now see that the model has a saddle point within the trust region: the model increases if we move northwards or southwards, the model decreases if we move eastwards or westwards.

∴ we move to the right limit of the trust region.

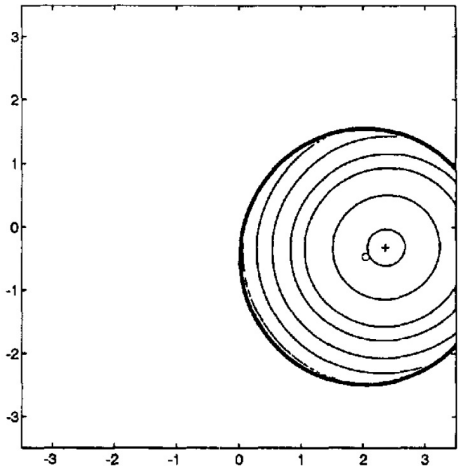
But! This move is too bold b/c as you can see in the 1st plot of the function, the model no longer represents objective function well this far from $x_{k=2}$




∴ Stay where we are

$x_{k=3} = x_{k=2}$ and reduce the trust region radius.

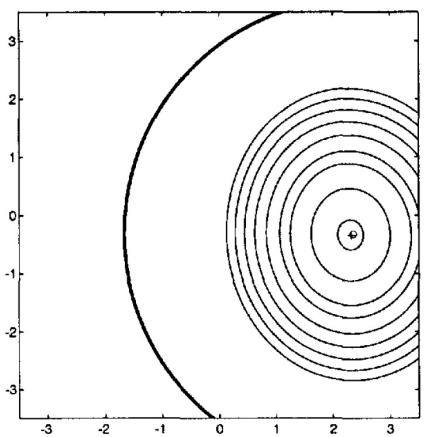
Pick a new trial point $x_{k=4}$




## keep repeating steps .....

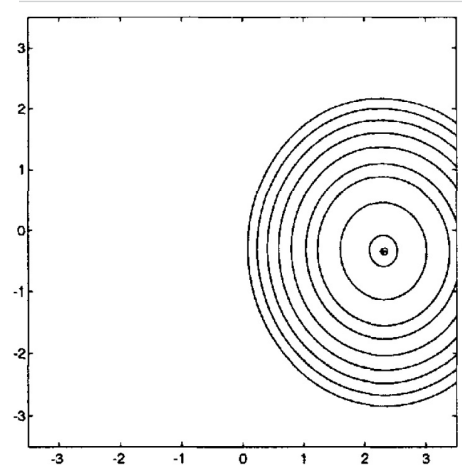
- new $x_k$
- build model $m_k$
- save $\Delta_k$

- Analyze




- new $x_k$
- build model $m_k$
- increase $\Delta_k$

- Analyze




- new $x_k$
  - ⇒ turns out to be the rightmost minimizer of this objective function.
- build model $m_k$
- increase $\Delta_k$
  - ⇒ increased to the point where it disappears from the region shown in the figure.

→ Used 8 evaluations of the objective function (together with its slope and curvature) to reach one of the two local minimizers!

# TRUST REGION METHOD

## Takeaways from the problem

→ Trust Region Method is iterative.

→ It produces a local solution

→ if the function has multiple local solutions, the found local solution is based on what we chose as our initial guess

→ • At each iteration $x_k$, we build a model that approximates the objective function in a region centered at $x_k$. This region is called the trust region radius.

• We then compute a step to a trial point within the trust region at which we obtain a sufficient model decrease.

• After computing the value of $f(x_k)$, we compute the achieved reduction in f to the predicted reduction in the model:

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)}$$

"actual improvement" / "expected improvement"

- This $\rho_k$ value changes the trust region radius:

if $\rho_k \geq \eta$ : • $x_{k+1} = x_k + p_k$ (new starting point)

o.w. : • $x_k = x_k$ (same starting point)

• Then, can set the new trust region radius:

if $\rho_k < 0.25$ : $\Delta_{k+1} = 0.25\Delta_k$ (shrink trust region $\Delta_{k+1}$)

else if $\rho_k > 0.75$ and $\|p_k\|_2 = \Delta_k$ :

set $\Delta_{k+1} = \min(2\Delta_k, \Delta_{max})$

(increase $\Delta$ b/c quadratic model is found to be reasonable ∴ ↑ $\Delta$ to allow larger steps)

# TRUST REGION METHOD

## How does it work?

Trust Region Method Algorithm:

**1.** Set $k = 1$, the initial trust region radius $\Delta_{k=1}$ and choose tolerance $\eta \in [0, 1/4)$ and $\Delta_{\max}$

**Note:** you can select the trust region radius - it will update each step. It is usually taken to be an ellipse such that $p^T D^2 p \leq \Delta^2$ where $D$ is a diagonal scaling (often taken from the diagonal of the approximate Hessian). The trust region radius $\Delta_k$ is modified dynamically each step - if the quadratic model is found to be a poor representation of the space the radius is resized in step 6.

**2.** Update the model function $m_k(x_k + p)$:

$$m_k(x_k + p) = f(x_k) + g(x_k)^T p + \frac{1}{2} p^T B_k p$$

Step 3 is the minimization sub-problem →

**3.** Compute an approximate solution to: $p_k^* = \arg\min_p m_k(x_k + p)$, such that $\|p\|_2 \leq \Delta_k$

**Note:** This is the **minimization subproblem** that can be solved using the Cauchy point algorithm (inexact solution), exact trust region algorithm, Steihaug-Toint algorithm, etc. This step will be shown in further detail in the next Section. Also, $\|.\|_2$ is "two-norm" and means the trust region is a circle. $\|p\|_2 \leq \Delta_k$ is the same as $p^T p \leq \Delta_k^2$.

- There are different algorithms to compute this:
  1.) Cauchy point algorithm - inexact
  2.) Exact trust region algorithm

**4.** Compute $f(x_k + p_k)$, and compute the ratio:

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)}$$

**Note:** The $\rho_k$ ratio is "actual improvement" over "inspected improvement".

**5.** Set the next point:
- If $\rho_k \geq \eta$, define $x_{k+1} = x_k + p_k$ (calculate new starting point because doesn't meet tolerance), otherwise define $x_{k+1} = x_k$ (keep the same starting point)

**6.** Set the new trust region radius:

- If $\rho_k < 0.25$, then set $\Delta_{k+1} = 0.25\Delta_k$ (shrink the trust region $\Delta_{k+1}$)

- Else If $\rho_k > 0.75$ and $\|p_k\|_2 == \Delta_k$, then set $\Delta_{k+1} = \min(2\Delta_k, \Delta_{\max})$ (increase the trust region because the quadratic model is found to be reasonable, so the radius is increased to allow larger, more efficient steps to be taken).

- Else, set $\Delta_{k+1} = \Delta_k$ (keep the same trust region).

**7.** Set $k = k + 1$, Go to step 2

## Main Points:

Trust region method algorithm has 7 steps!
Step 3 of the trust region method algorithm is called the minimization sub-problem.

# TRUST REGION METHOD

**Breakdown of Step 3**

"Minimization Sub-problem"

Remember Step 3 of the Trust Region Method:

Compute an approx. solution to $p_k^* = \arg\min_p m_k(x_k + p)$

s.t. $\|p\|_2 \leq \Delta_k$

∴ We need to find the solution of the constrained minimization problem:

$$\min_p g^T p + \frac{1}{2} p^T B p \quad \text{s.t. } \|p\|_2 \leq \Delta_k$$

There are several algorithms to do this.

The two main ones are:

1.) Cauchy point algorithm
- gives an estimate of $p^*$

2.) Exact trust region algorithm
- gives exact $p^*$

**Main Points:**

- Step 3 of the Trust Region method is: $p_k^* = \arg\min_p m_k(x+p)$
- 2 algorithms to solve Step 3: Cauchy Point algorithm and Exact trust region algorithm.

# Cauchy Point Algorithm:

Cauchy point is the minimum value of $m(x+p)$ subject to $\|p\|_2 \leq \Delta$ along the steepest descent direction: $p = -\alpha g$

- $\alpha = \gamma \frac{\Delta}{\|g\|_2}$
- $\gamma$: factor that governs whether the curvature is along $g$ or not

$$p = -\gamma \frac{\Delta}{\|g\|_2} g,$$

$$\gamma = \begin{cases} 1, & g^T B g \leq 0 \\ \min\left(1, \frac{\|g\|_2^3}{\Delta g^T B g}\right) & \text{otherwise} \end{cases}$$

## Constraints:

if $\gamma = 1$, then the trust region constraint bound is active

$\therefore$ steps to boundary / trust region radius

if $\gamma = \frac{\|g\|_2^3}{\Delta g^T B g}$, the step is to a minimizer within the radius.

# Exact trust region Algorithm:

Exact solution $p^*$ is characterized as follows:

1.) $(B + \lambda I) p^* = -g$

2.) $(\Delta - \|p^*\|_2)\lambda = 0$

3.) $\lambda \geq 0$

4) $(B + \lambda I)$ positive, semi-definite

We have to find $p^*$ and $\lambda$ that will satisfy these conditions.

## Complementary constraint:

Either:

- $\|p^*\|_2 \leq \Delta$ (inside trust region) and $\lambda = 0$

or

- $\|p^*\|_2 = \Delta$ (on trust region boundary) and $\lambda \neq 0$

## Cauchy Point Case 1:

B is positive definite and the unconstrained minimizer lies outside the trust region

$$p = -\gamma \frac{\Delta g}{\|g\|_2} g, \quad \gamma = 1 \text{ b/c } g^T B g \leq 0$$

$$\Rightarrow p = \frac{-\Delta}{\|g\|_2} g$$

## Cauchy Point Case 2:

B is positive definite and the unconstrained minimizer lies inside the trust region.

$$\gamma = \frac{\|g\|_2^3}{\Delta g^T B g} \text{ b/c } g^T B g \geq 0$$

$$\rightarrow p = -\gamma \frac{\Delta}{\|g\|_2} g,$$

$$\gamma = \min\left(1, \frac{\|g\|_2^3}{\Delta g^T B g}\right)$$

## Cauchy Point Case 3:

B is indefinite

$g^T B g > 0$ $\therefore$ inside b/c bounding constraint

## Exact Case 1:

B is positive definite and $\|p\| > \Delta$

Then: solve for $\lambda$ with $\lambda \geq 0$

For $\lambda_1$: $(B - \lambda_1 I) q_1 = 0$

to get $q_1$

For $\lambda_2$: $(B - \lambda_2 I) q_2 = 0$

to get $q_2$

Use: $\lambda_1, \lambda_2, q_1, q_2, g,$ and $\Delta$

$$\Rightarrow \|p(\lambda)\|_2^2 = \sum_{i=1}^{2} \frac{(q_i^T g)^2}{(\lambda_i + \lambda)^2} \leq \Delta^2$$ to get $\lambda$ ($\lambda \geq 0$)

Then: $(B + \lambda I) p = -g$

$\Rightarrow p$

## Exact Case 2:

B is positive definite and $\|p\| \leq \Delta$.

Then: $\lambda = 0$

$\Rightarrow p^* = p$

## Exact Case 3:

B is indefinite Hessian.

Then: Solve $\lambda$ with 1.) $\lambda \geq 0$ 2) $\lambda \geq (-\lambda_1)$

For $\lambda_1$: $(B - \lambda_1 I) q_1 = 0$

to get $q_1$

For $\lambda_2$: $(B - \lambda_2 I) q_2 = 0$

to get $q_2$

Use: $\lambda_1, \lambda_2, q_1, q_2, g, \Delta$

$$\Rightarrow \|p(\lambda)\|_2^2 = \sum_{i=1}^{2} \frac{(q_i^T g)^2}{(\lambda_i + \lambda)^2} \leq \Delta^2$$

to get $\lambda$ multiplier with constraints $\lambda \geq 0$ and $\lambda \geq (-\lambda_1)$

Then: plug $\lambda$ into $(B + \lambda I) p = -g$

$\Rightarrow p$