Unconstrained Optimization – Trust Region

WQD7011 Numerical Optimization

Recalling…

- What is the **generic terms for solving unconstrained optimization**?
- How does **line search** works?

Unconstrained Optimization - The Iterative function

- **Iterative**
- Solve a much **easier** **optimization problem** at every iteration
- **Line Search**
  - 1-dimensional sub-problem: only search either direction or step size at once
- **Trust Region**
  - n-dimensional sub-problem
  - In simpler objective function – a linear of quadratic model
  - Trusted in simple region – a ball of specified radius

Trust Region

- **Why simpler objective function?**
  - Initial way: n-dimensional unconstrained sub-problems
  - With TR: **n-dimensional constrained sub-problems**.
- In addition:
  - Sub-problem needs **NOT** to be solved to **high accuracy** – only need an approximate one.
  - **Model is proven to be effectives** in many cases.

Trust Region

- Instead of working on $f(x)$, which is more difficult to find the minimum, we work on another function m_k, which is easier.
- **What is m_k?**
  - m_k is just a **local approximation of f** (i.e., model inherited from f), it can represent f only in a region near x_k.
  - New m_k is generated for each iteration.

Line Search vs Trust Region *(methodology form)*

- **Line Search**
  a) Pick descent direction, p_k
  b) Pick step size, α_k, to reduce $f(x_k + \alpha p_k)$
  c) Then, $x_{k+1} = x_k + \alpha_k p_k$

Line Search vs Trust Region *(methodology form)*

- **Trust Region**
  - Pick step, p_k, to reduce the model of $f(x_k + p_k)$
  - Accept $x_{k+1} = x_k + p_k$ if the notable decrease gained by the model inherited by $f(x_k + p_k)$.
  - Otherwise set $x_{k+1} = x_k$ and improve the model.

Trust Region - Redefined

- A region around the **current iterate within**, which they trust the model to be an **adequate representation of the objective function**.
- Usually a **spherical** area of **radius $\boldsymbol{\Delta_k}$**.
- After every iteration, work on new model, m_{k+1}, and find a new radius Δ_{k+1}

Trust Region Sub-problem
The Generic Form

- Assuming m_k as a **quadratic model** which works in many cases:

$$m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p \qquad \text{subject to } \|p\| \leq \Delta_k \ (1)$$

$f(x_k)$

$g_k = \nabla f(x_k)$

B_k - Hessian or approximation

Trust Region Sub-Problem

- If we can use the quadratic equation to model the objective function f, then our optimization problem **has been reduced to a series of smaller and simpler sub-problems**.
- In each sub-problem, we need to determine the direction p such that $\|p\| \leq \Delta_k$.

Determination factor, $\boldsymbol{\rho_k}$

- How to determine whether this is a **good model (representation region with good step)**?
- Use **ratio of actual reduction and predictive reduction**.

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}$$

Determination factor, ρ_k

- Numerator is the actual reduction.
- Denominator is the predicted reduction (i.e., the reduction in f predicted by the model function)
- Since p_k is obtained by minimizing m_k over a region that includes $p = 0$, the predicted reduction will be always be non-negative.

Determination factor, ρ_k

- Thus, if ρ_k is **negative**: the new objective value $f(x_k + p_k)$ is greater than the current value $f(x_k)$, so the step **MUST BE REJECTED**.
- If ρ_k is **close to 1**, there is **GOOD** agreement between the m_k and the function f over this step
  - So it is safe to expand the trust region for the next iteration.
- If ρ_k is **positive BUT significantly smaller than 1**, we **do not alter the trust region**.
- If ρ_k is **close the zero or negative**, we **shrink the trust region** by reducing Δ_k at next iteration.

Trust Region Algorithm

Initialization: k = 0 and Δ_k = upper bound of radius of trust region

while not converge {

obtain p_k by solving trust region sub-problem (eq. (1))

evaluate ρ_k

if ρ_k is too small

consider a smaller radius Δ_k

else if ρ_k is large enough

consider to increase the radius Δ_k

else

consider current radius Δ_k

if ρ_k is larger then a threshold

accept this model and take this move

else

try again with a new model (smaller radius)

increase k by 1

}

Trust Region Algorithm

Algorithm 4.1 (Trust Region).

Given $\hat{\Delta} > 0$, $\Delta_0 \in (0, \hat{\Delta})$, and $\eta \in \left[0, \frac{1}{4}\right)$:

for $k = 0, 1, 2, \ldots$

  Obtain p_k by (approximately) solving (4.3);

  Evaluate ρ_k from (4.4);

  if $\rho_k < \frac{1}{4}$

    $\Delta_{k+1} = \frac{1}{4}\Delta_k$

  else

    if $\rho_k > \frac{3}{4}$ and $\|p_k\| = \Delta_k$

      $\Delta_{k+1} = \min(2\Delta_k, \hat{\Delta})$

    else

      $\Delta_{k+1} = \Delta_k$;

  if $\rho_k > \eta$

    $x_{k+1} = x_k + p_k$

  else

    $x_{k+1} = x_k$;

end (for).

Solving trust region sub-problem…

- **…is another optimization problem.**

$$m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p$$

$$\text{subject to } ||p|| \leq \Delta_k \ (1)$$

- Sometimes, **cheaper and approximate solution is preferred:**
  - Cauchy Point
  - Dogleg Method
  - Two-dimensional subspace minimization.

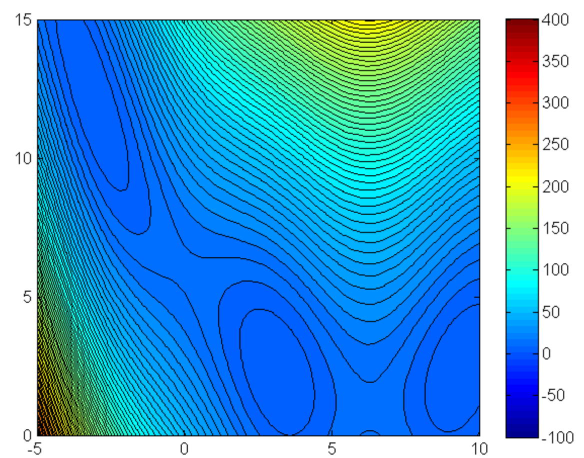
Example of Trust Region Method

Here we use the trust-region method to solve an unconstrained problem as an example. The trust-region subproblems are solved by calculate the Cauchy point.

$min\ f(x_1, x_2) = (x_2 - 0.129{x_1}^2 + 1.6x_1 - 6)^2 + 6.07cos(x_1) + 10$ (This is the Branin function which is widely used as a test function. It has 3 global optima.)

Starting point $x_1 = 6.00, x_2 = 14.00$ The iteration stops when the stopping criteria $||g_k|| <= 0.01$ is met.

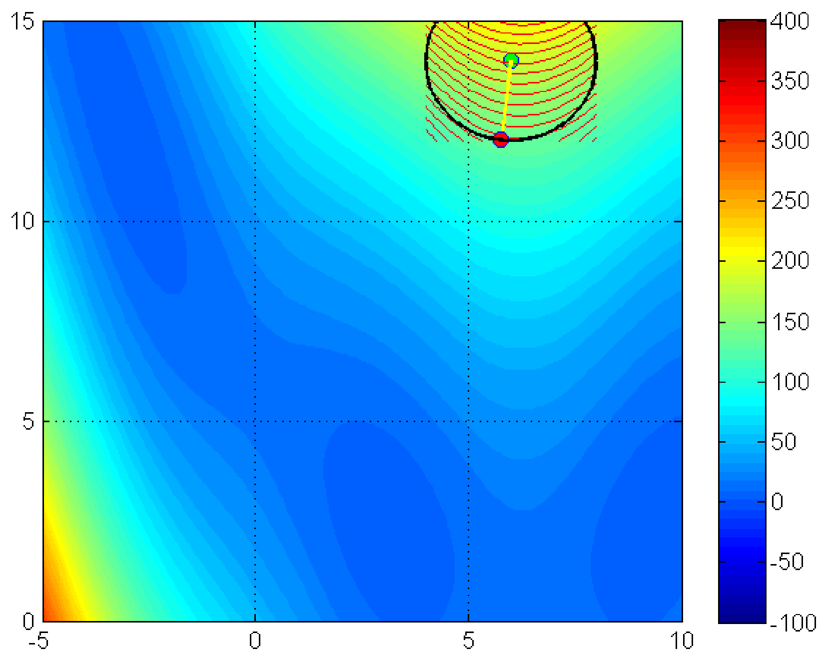
$$\Delta_0 = 2.0, \Delta_M = 5.0, t_1 = 0.25, t_2 = 2.0, \eta_1 = 0.2, \eta_2 = 0.25, \eta_3 = 0.75$$



Contour of a 'Branin' function.

Example of Trust Region Method

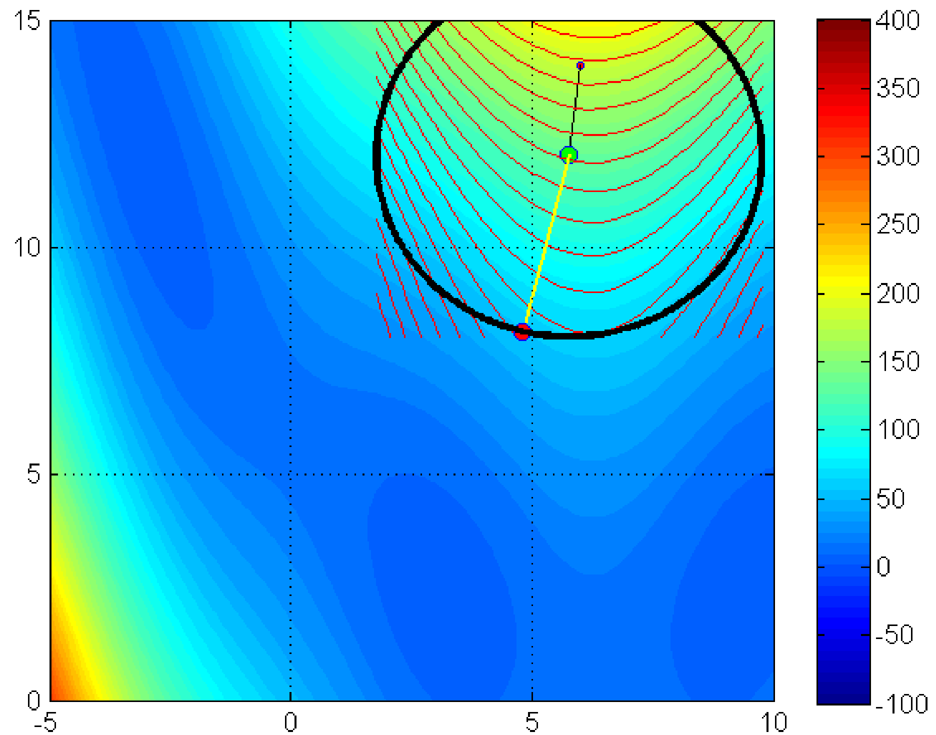
Iteration 1: The algorithm start from the initial point(marked as a green dot) $x_1 = 6.00, x_2 = 14.00$. The trust-region is defined as the area inside the circle centered at the starting point. The contour of the quadratic model can be visualized. After calculating the Cauchy point, ρ_k is evaluated and a full step was taken since the model gives a good prediction. Set $x_{k+1} = x_k + p_k(x_1 = 5.767, x_2 = 12.014)$ and $\Delta_k = min(2\Delta_k, \Delta_M)$ ($\rho_k > \eta_3$ and a full step was taken.) (The current best solution is denoted as the red dot.)



Graphical illustration of iteration 1 (The quadratic model's contours are marked as red lines).

Example of Trust Region Method

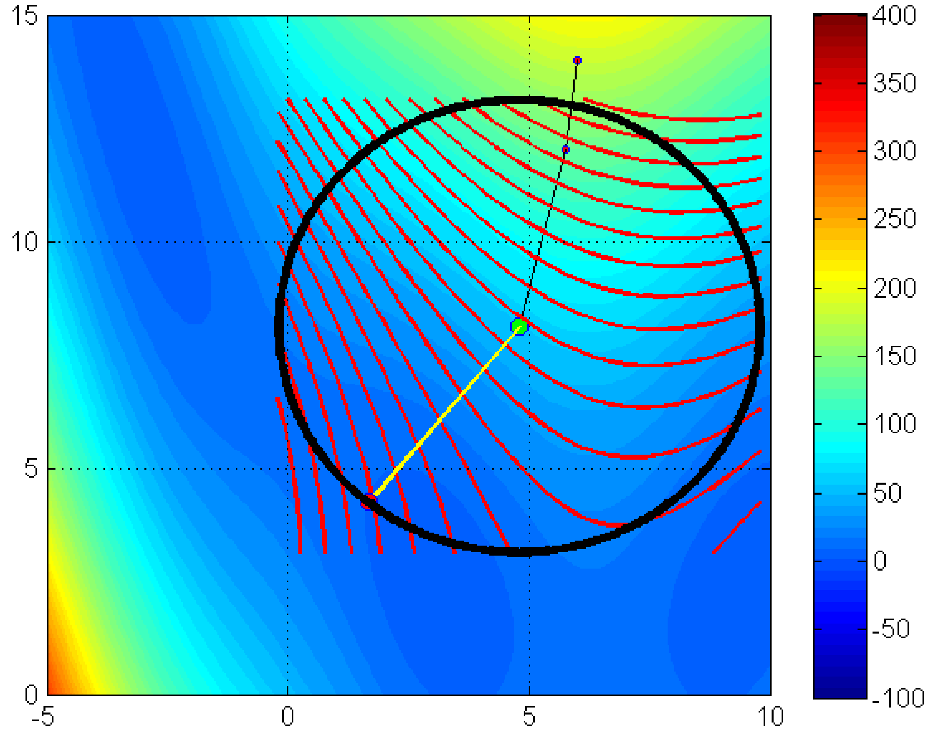
Iteration 2: Start with $x_1 = 5.767, x_2 = 12.014$ and an enlarged trust-region. The new iteration gives a more ambitious full step to the new point $(x_1 = 4.800, x_2 = 8.132)$. With $\rho_k = 0.980$, the model is "trusted" again to increase its size in the next iteration.



Graphical illustration of iteration2.

Example of Trust Region Method

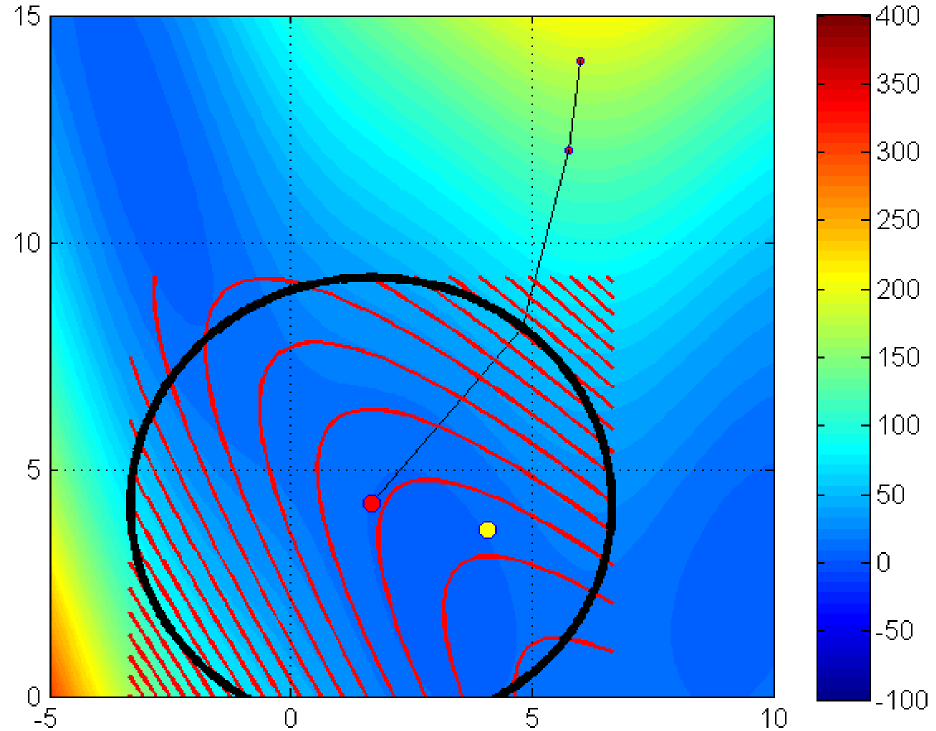
Iteration 3: Start with $x_1 = 4.800, x_2 = 8.132$ and an enlarged trust-region. The new iteration gives a satisfactory but not good enough to the new point $(x_1 = 1.668, x_2 = 4.235)$. With $\rho_k = 0.578$, which is not high enough to trigger a new increment for the trust-region's size. So the radius is maintained in the next iteration.



Graphical illustration of iteration3.

Example of Trust Region Method

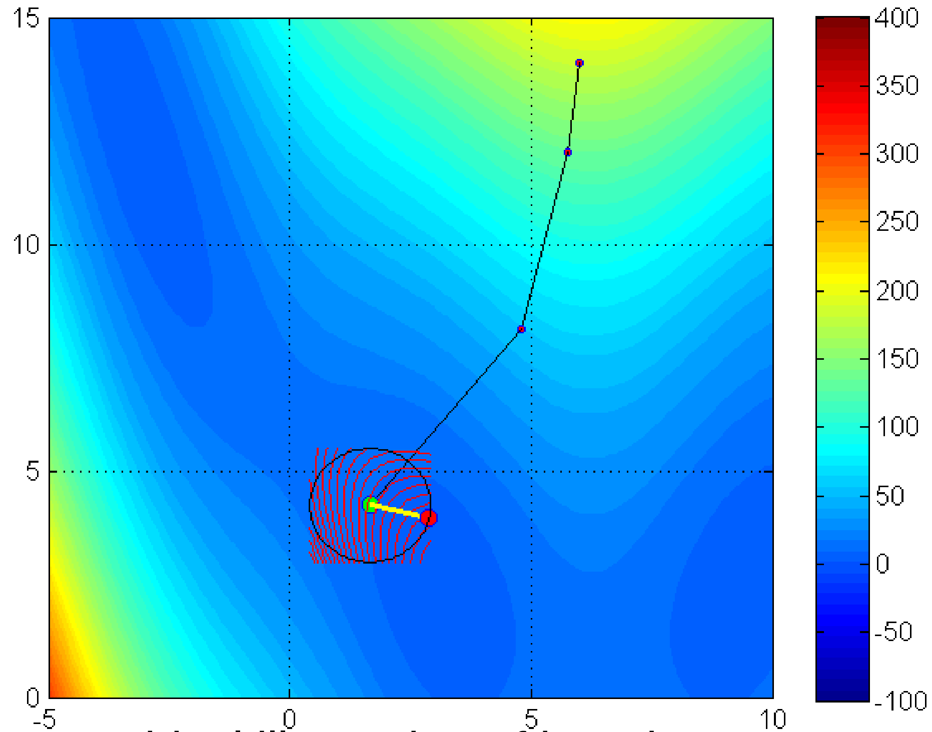
Iteration 4: Start with $x_1 = 1.668, x_2 = 4.235$ and the maximum-sized trust-region. The new iteration gives a poor prediction. With $\rho_k = -0.160$, which incurs the decrease in the trust-region's size to improve the model's validity. Current best solution is unchanged and the radius for the trust-region is diminished to 1/4 of the current iteration.



Graphical illustration of iteration4.

Example of Trust Region Method

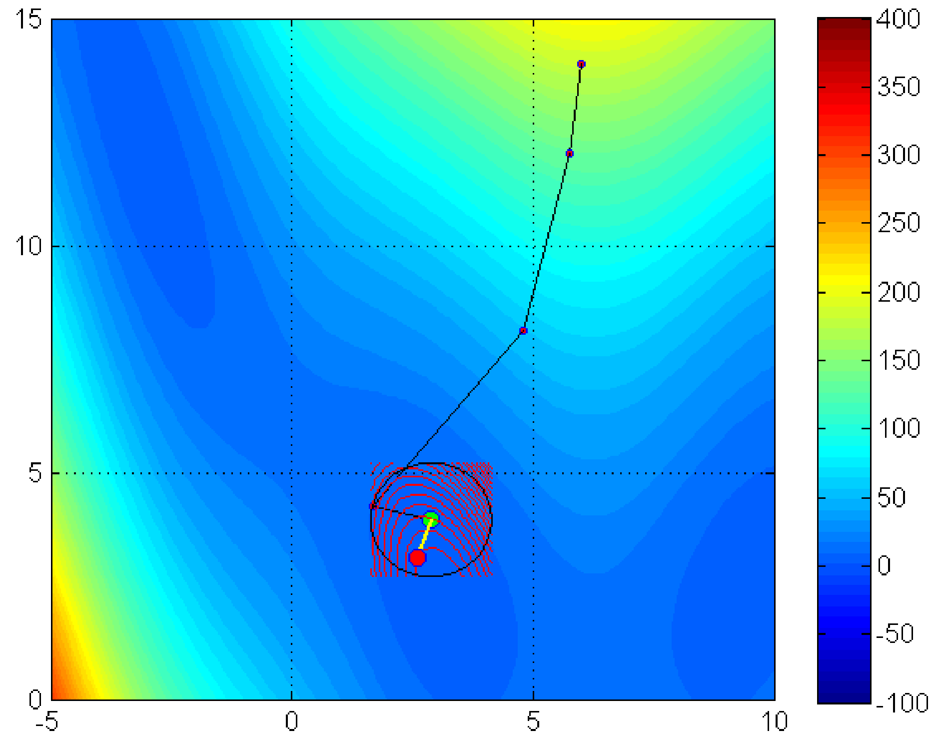
Iteration 5: Start with $x_1 = 1.668, x_2 = 4.235$ and a shrinked trust-region. The new iteration gives a satisfactory but not good enough to the new point $(x_1 = 2.887, x_2 = 3.956)$. With $\rho_k = 0.729$, which is not high enough to trigger a new increment for the trust-region's size. So the radius is maintained in the next iteration.



Graphical illustration of iteration5.

Example of Trust Region Method

Iteration 6: Start with $x_1 = 2.887, x_2 = 3.956$. The new iteration gives a satisfactory but not a full step to the new point $(x_1 = 2.594, x_2 = 3.109)$. With $\rho_k = 0.989$, which is high enough to trigger a new increment for the trust-region's size, however not a full step is taken thereby the radius is maintained in the next iteration.

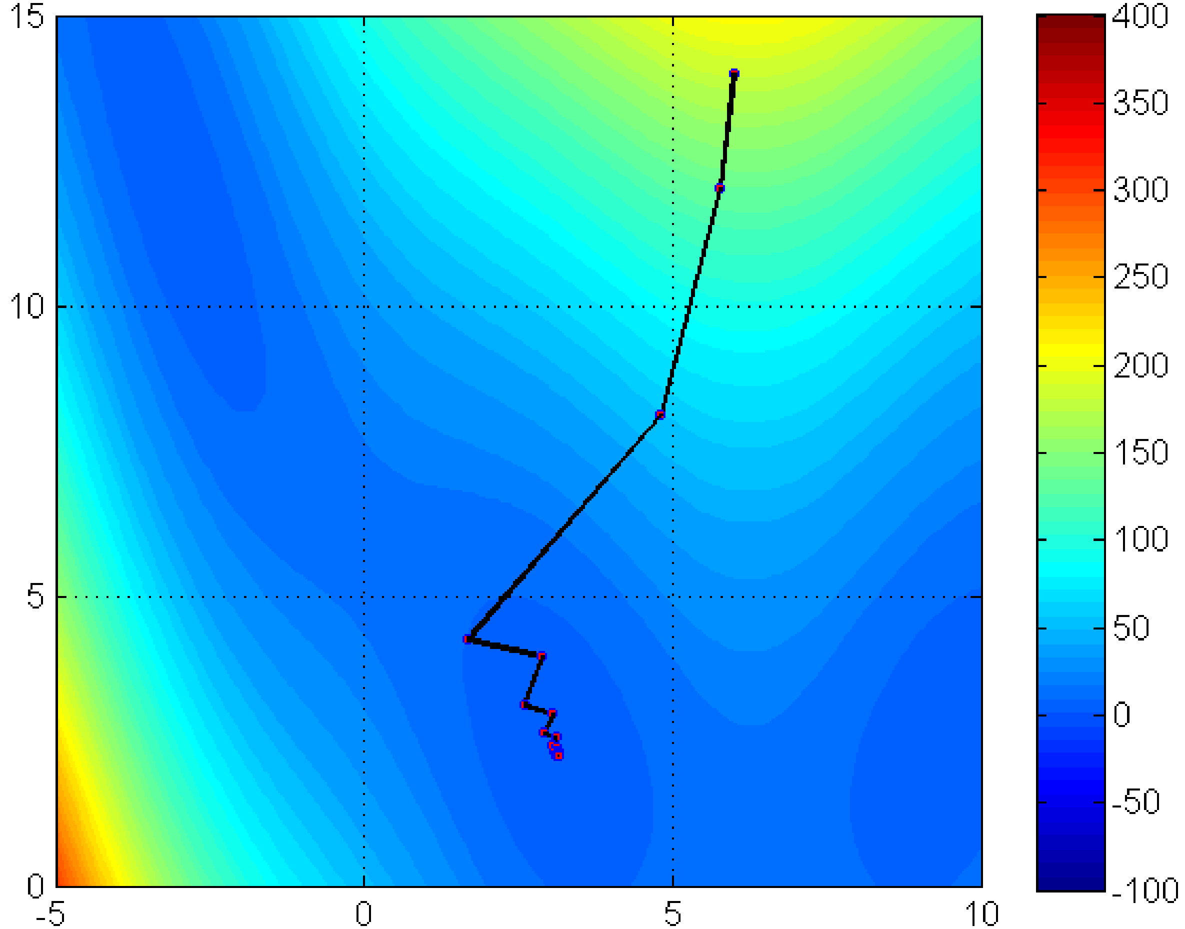


Graphical illustration of iteration6.

Example of Trust Region Method

Iteration	f(x)	x_1	x_2	ρ_k	Δ_k	$\|\|p_k\|\|$	$\|\|g_k\|\|$
1	183.686	6.000	14.000	0.999	2.00	2.000	26.090
2	135.192	5.767	12.014	0.980	4.00	4.000	22.570
3	57.318	4.800	8.132	0.578	5.00	5.000	17.550
4	9.708	1.668	4.235	-0.160	5.00	2.474	4.890
5	9.708	1.668	4.235	0.729	1.25	1.250	4.890
6	6.376	2.887	3.956	0.989	1.25	0.897	3.173
7	4.970	2.594	3.109	0.956	1.25	0.493	2.553
8	4.369	3.063	2.958	0.992	1.25	0.353	1.418
9	4.121	2.920	2.635	0.996	1.25	0.204	1.064
10	4.013	3.108	2.556	0.996	1.25	0.154	0.616
11	3.966	3.046	2.414	1.001	1.25	0.088	0.466
12	3.946	3.127	2.380	0.998	1.25	0.067	0.264
13	3.937	3.101	2.318	1.001	1.25	0.038	0.202
14	3.933	3.135	2.304	0.999	1.25	0.029	0.113
15	3.931	3.124	2.277	1.000	1.25	0.016	0.087
16	3.931	3.139	2.271	1.000	1.25	0.012	0.048
17	3.930	3.134	2.260	1.000	1.25	0.007	0.037
18	3.930	3.140	2.257	1.000	1.25	0.005	0.020
19	3.930	3.138	2.252	1.000	1.25	0.003	0.016

Example of Trust Region Method



Optimization trajectory of the example function(unconstrained).

Practical Exercise

- In this exercise, we will use different approach in calculating the step in Trust Region.
- We need the following .py files:
  a. main.py
  b. trust_region.py
  c. step_finders.py
  d. linalg_utils.py
- All these files can be downloaded via Spectrum (no typing needed today!)

Exercise 4

- Read and try to understand the code, then answer the following questions:
    1. Use one sentence to describe the purpose of each function in file (b), (c), and (d).
    2. Modify the code to display ρ for each iteration.
    3. What is(are) the convergence criteria(s) for the trust region method?
    4. Modify the code the use different step finder, and record the ρ.
    5. From the answers in Q4, compare the recorded results, and state your observation(s).
- Add on question: can you plot the graph to show the trajectory?

Discuss previous exercise… Exercise 2

1. Why the "+"?
2. Identify ONE technique / method (not in lecture) to perform step length calculation.
3. Identify ONE technique / method (not in lecture) to perform direction searching.
4. Explain the practical steps in Armijo function using words.
5. Explain the practical steps in Wolfe function using words.
6. What is your observation for the results generated using Armijo conditions and wolfe conditions? Discuss.

Discuss previous exercise… Exercise 3

1. Explain the general steps in step-size searching.
2. Explain *sufficient decrease condition*.
3. Differentiate between exact line search and inexact line search.
4. What is the main purpose of having multiple conditions (e.g., Armijo, curvature, etc.) in step size searching?
5. *The more conditions we add into the step size searching algorithm, the lesser possible candidates for step size searching.* Do you agree with this statement? Justify your answer.