AE6310: Optimization for the Design of Engineered Systems
Assignment 2: Trust Region Methods
Due: March 5th, 2020

Answer the following questions. Organize your work and be careful to properly answer all parts of each question. Points will be deducted for unorganized presentation of results.

1. (30 points) For each quadratic model and trust region radius below:
   (a) Plot the model function and the trust region radius
   (b) Find the Cauchy step
   (c) Find the exact trust region step and give the Lagrange multiplier for the exact solution. You may need to use a numerical methods or a symbolic algebra package to find the correct roots.
   (d) Indicate whether the exact step is on the trust region radius boundary

   - $\Delta = 1$ with
   $$g = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \end{bmatrix} \; B = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$$
   - $\Delta = 1/2$ with
   $$g = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \end{bmatrix} \; B = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$$
   - $\Delta = 1$ with
   $$g = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \; B = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}$$

2. (30 points) Create two trust region subroutines with the following prototypes:
   (a) `pC = cauchy_step(g, B, delta)` that returns the Cauchy step for a given quadratic model
   (b) `p = trust_region_step(g, B, delta)` that returns a solution to the exact trust region problem

   Note that the solution procedure can be specialized to two-dimensional problems where $g \in \mathbb{R}^2$ and $B \in \mathbb{R}^{2\times 2}$ is a symmetric matrix.

   The exact trust region step code should perform the following steps:

   - Compute the eigenvalues and eigenvectors of the matrix B
   - If the matrix is positive definite, compute the step to the model minimizer $p = -B^{-1}g$, and check if $||p||_2 \leq \Delta$ in which case $\lambda = 0$ and $p^* = p$.
   - Otherwise, the solution must have $\lambda > 0$. Solve for λ using the equation $||p(\lambda)||_2^2 = \Delta^2$.

   Given the eigenvalues λ_1 and λ_2, and eigenvectors q_1 and q_2, and the model gradient g, find a general expression for $||p(\lambda)||_2^2$. Use a root finding algorithm (such as numpy.roots or the matlab equivalent roots) to find the roots of the equation $||p(\lambda)||_2^2 - \Delta^2 = 0$.

   (a) Write out the polynomial coefficients for the equation $||p(\lambda)||_2^2 - \Delta^2 = 0$ for the $n = 2$ case using the eigenvalues and eigenvectors.
   (b) Verify that these routines give the same answers from Question 1
   (c) Create three of your own model functions which fall into the three categories that we discussed in class:
      - Minimizer is in the interior of the trust region
      - Positive definite model, but minimizer is constrained

- Indefinite/negative definite Hessian

Test your implementation against the problems you created. Plot the contours of the model, the trust region constraint boundary, the Cauchy point and the exact solution. Be sure to thoroughly test your code. Try difficult problems!

3. (40 points) Implement a trust region algorithm for unconstrained optimization. The algorithm is outlined below. Be sure to reuse your code to compute the solution of the trust region problem and Cauchy steps.

   (a) Apply the algorithm to minimize the functions

$$f(x_1, x_2) = -10x_1^2 + 10x_2^2 + 4\sin(x_1 x_2) - 2x_1 + x_1^4$$
$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

   (b) Compare the performance of the Cauchy point method and the exact trust region step method in terms of the number of function evaluations.

   (c) Comment and discuss on the performance from various starting points.

Set the initial trust region radius Δ_1 and choose $\eta \in [0, 1/4)$ and $\Delta_{\max}$
Evaluate $f(x_1)$ and $g_1 = \nabla f(x_1)$
Set $B_1 = I$
while $||g_k||_2 = ||\nabla f(x_k)||_2 > \epsilon$ **do**
  if Use Cauchy point step **then**
    $p_k =$ `cauchy_step(`g_k, B_k, Δ_k`)` ▷ Solve the trust region problem
  else
    $p_k =$ `trust_region_step(`g_k, B_k, Δ_k`)`
  end if
  Compute $f(x_k + p_k)$ and evaluate the ratio:

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

  Compute $\nabla f(x_k + p_k)$
  Set $y_k = \nabla f(x_k + p_k) - \nabla f(x_k)$ and $s_k = p_k$ and update B_k using the SR-1 formula:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{s_k^T (y_k - B_k s_k)}$$

  if $\rho_k \geq \eta$ **then** ▷ Set the new point
    Set $x_{k+1} = x_k + p_k$
  else
    Set $x_{k+1} = x_k$
  end if
  if $\rho_k < 0.25$ **then** ▷ Set the new trust region radius
    Set $\Delta_{k+1} = 0.25\Delta_k$
  else if $\rho_k > 0.75$ and $||p_k||_2 == \Delta_k$ **then**
    Set $\Delta_{k+1} = min(2\Delta_k, \Delta_{\max})$
  else
    Set $\Delta_{k+1} = \Delta_k$
  end if
  Set $k = k + 1$, Goto step 2
end while