Numerical Optimization

Shuai Lu 170742

Homework 5

Exercise 5.1 [3 points] Consider the constrained optimization problem

$$\begin{array}{l} \min_{x_1,x_2} x_1^2 + 2x_2^2 \\ s.t. \quad 3x_1 + 5x_2 = 7 \end{array}$$

Write down its Lagrange function. Formulate the first order necessary condition for the local minimum and solve the problem analytically. Check the second order sufficient condition at the solution. Compute the Hessian of the Lagrange function w.r.t. all the variables (x_1, x_2, ∇) at the solution and report if it is positive or negative (semi-) definite or indefinite.

Solution:

The Lagrange function $L(x, \lambda) = x^T Ax - \lambda \left(b^T x - c\right)$
where

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, b = \begin{bmatrix} 3 \\ 5 \end{bmatrix}, c = 7$$

So $L(x_1, x_2, \lambda) = x_1^2 + 2x_2^2 - \lambda(3x_1 + 5x_2 - 7)$.
From the primal feasibility condition:

$$\nabla_x L(x, \lambda) = 2Ax - \lambda b = 0 \rightarrow x_* = \frac{\lambda A^{-1} b}{2}$$

Substituting x in $b^T x - c = 0$ with x_*, we get $\lambda_* = \frac{2c}{b^T A^{-1} b} = \frac{28}{43}$.
So

$$x_* = \frac{\lambda_* A^{-1} b}{2} = \begin{bmatrix} \frac{42}{43} \\ \frac{35}{43} \end{bmatrix}$$

Since $\nabla_{xx} L(x, \lambda) = 2A = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$ is positive definite, the second order sufficient condition at x_* is satisfied.
The Hessian

$$\nabla_{x,\lambda} L(x, \lambda) = \begin{bmatrix} 2 & 0 & -3 \\ 0 & 4 & -5 \\ -3 & -5 & 0 \end{bmatrix}$$

Its eigenvalues are -4.4, 2.8, 7.9, so the Hessian matrix is indefinite.

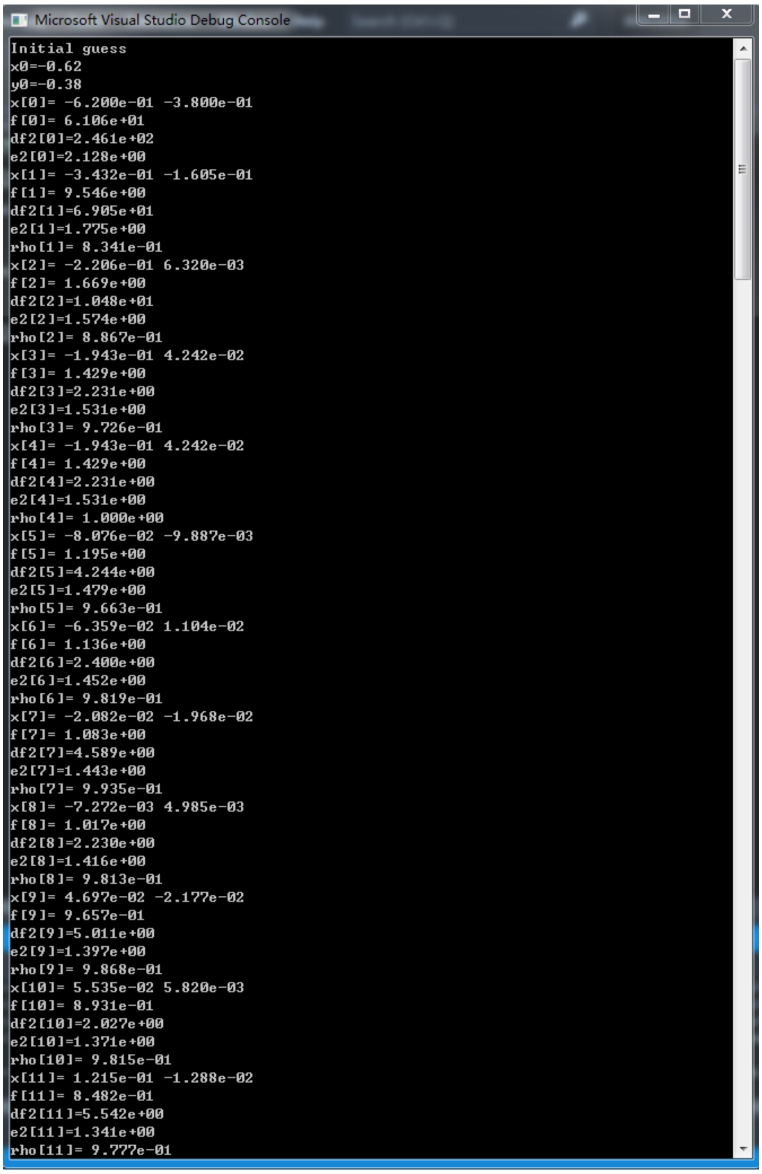
Exercise 5.2 [6 points] Implement the Cauchy point trust region method. Test it for the problem settings from Exercise 4.3 (a). Use the same termination criteria as in Ex. 4.3 (and 3.3(b)). In your tests, set the maximum radius of the trust region to 1, the initial radius to 0.5 and the threshold η to 0.2. After every iteration k as well as for the initial guess (k = 0), print $\mathbf{x}_k, f(\mathbf{x}_k), \|\nabla f(\mathbf{x}_k)\|_2$, the current radius of the trust region as well as $\|\mathbf{e}_k\|_2$ for $\mathbf{e}_k := \mathbf{x}_k - \mathbf{x}_*$. For k≥1, print also the reduction $\|\mathbf{e}_k\|_2 / \|\mathbf{e}_{k-1}\|_2$ of the error norm.

Solution:

The output of the program are shown in Figure 1 to Figure 5.

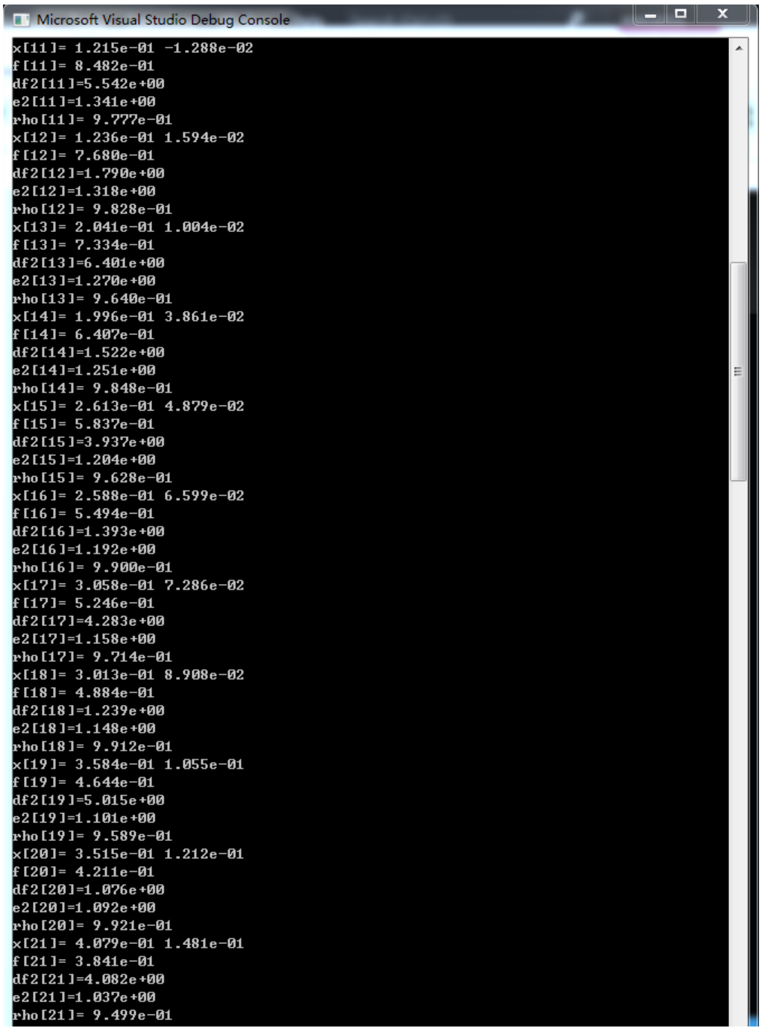
Exercise 5.3 [6 points] Extend your implementation of the Cauchy point method from Exercise 5.2 (with the same trust region settings and the same termination criteria) to the dogleg method. Test it for the same problem settings.

Solution:



```
Initial guess
x0=-0.62
y0=-0.38
x[0]= -6.200e-01 -3.800e-01
f[0]= 6.106e+01
df2[0]=2.461e+02
e2[0]=2.128e+00
x[1]= -3.432e-01 -1.605e-01
f[1]= 9.546e+00
df2[1]=6.905e+01
e2[1]=1.775e+00
rho[1]= 8.341e-01
x[2]= -2.206e-01 6.320e-03
f[2]= 1.669e+00
df2[2]=1.048e+01
e2[2]=1.574e+00
rho[2]= 8.867e-01
x[3]= -1.943e-01 4.242e-02
f[3]= 1.429e+00
df2[3]=2.231e+00
e2[3]=1.531e+00
rho[3]= 9.726e-01
x[4]= -1.943e-01 4.242e-02
f[4]= 1.429e+00
df2[4]=2.231e+00
e2[4]=1.531e+00
rho[4]= 1.000e+00
x[5]= -8.076e-02 -9.887e-03
f[5]= 1.195e+00
df2[5]=4.244e+00
e2[5]=1.479e+00
rho[5]= 9.663e-01
x[6]= -6.359e-02 1.104e-02
f[6]= 1.136e+00
df2[6]=2.400e+00
e2[6]=1.452e+00
rho[6]= 9.819e-01
x[7]= -2.082e-02 -1.968e-02
f[7]= 1.083e+00
df2[7]=4.589e+00
e2[7]=1.443e+00
rho[7]= 9.935e-01
x[8]= -7.272e-03 4.985e-03
f[8]= 1.017e+00
df2[8]=2.230e+00
e2[8]=1.416e+00
rho[8]= 9.813e-01
x[9]= 4.697e-02 -2.177e-02
f[9]= 9.657e-01
df2[9]=5.011e+00
e2[9]=1.397e+00
rho[9]= 9.868e-01
x[10]= 5.535e-02 5.820e-03
f[10]= 8.931e-01
df2[10]=2.027e+00
e2[10]=1.371e+00
rho[10]= 9.815e-01
x[11]= 1.215e-01 -1.288e-02
f[11]= 8.482e-01
df2[11]=5.542e+00
e2[11]=1.341e+00
rho[11]= 9.777e-01
```

Figure 1: Cauchy point method (Step 0 to 11)



```
Microsoft Visual Studio Debug Console
x[11]= 1.215e-01 -1.288e-02
f[11]= 8.482e-01
df2[11]=5.542e+00
e2[11]=1.341e+00
rho[11]= 9.777e-01
x[12]= 1.236e-01 1.594e-02
f[12]= 7.680e-01
df2[12]=1.790e+00
e2[12]=1.318e+00
rho[12]= 9.828e-01
x[13]= 2.041e-01 1.004e-02
f[13]= 7.334e-01
df2[13]=6.401e+00
e2[13]=1.270e+00
rho[13]= 9.640e-01
x[14]= 1.996e-01 3.861e-02
f[14]= 6.407e-01
df2[14]=1.522e+00
e2[14]=1.251e+00
rho[14]= 9.848e-01
x[15]= 2.613e-01 4.879e-02
f[15]= 5.837e-01
df2[15]=3.937e+00
e2[15]=1.204e+00
rho[15]= 9.628e-01
x[16]= 2.588e-01 6.599e-02
f[16]= 5.494e-01
df2[16]=1.393e+00
e2[16]=1.192e+00
rho[16]= 9.900e-01
x[17]= 3.058e-01 7.286e-02
f[17]= 5.246e-01
df2[17]=4.283e+00
e2[17]=1.158e+00
rho[17]= 9.714e-01
x[18]= 3.013e-01 8.908e-02
f[18]= 4.884e-01
df2[18]=1.239e+00
e2[18]=1.148e+00
rho[18]= 9.912e-01
x[19]= 3.584e-01 1.055e-01
f[19]= 4.644e-01
df2[19]=5.015e+00
e2[19]=1.101e+00
rho[19]= 9.589e-01
x[20]= 3.515e-01 1.212e-01
f[20]= 4.211e-01
df2[20]=1.076e+00
e2[20]=1.092e+00
rho[20]= 9.921e-01
x[21]= 4.079e-01 1.481e-01
f[21]= 3.841e-01
df2[21]=4.082e+00
e2[21]=1.037e+00
rho[21]= 9.499e-01
```

Figure 2: Cauchy point method (Step 11 to 21)

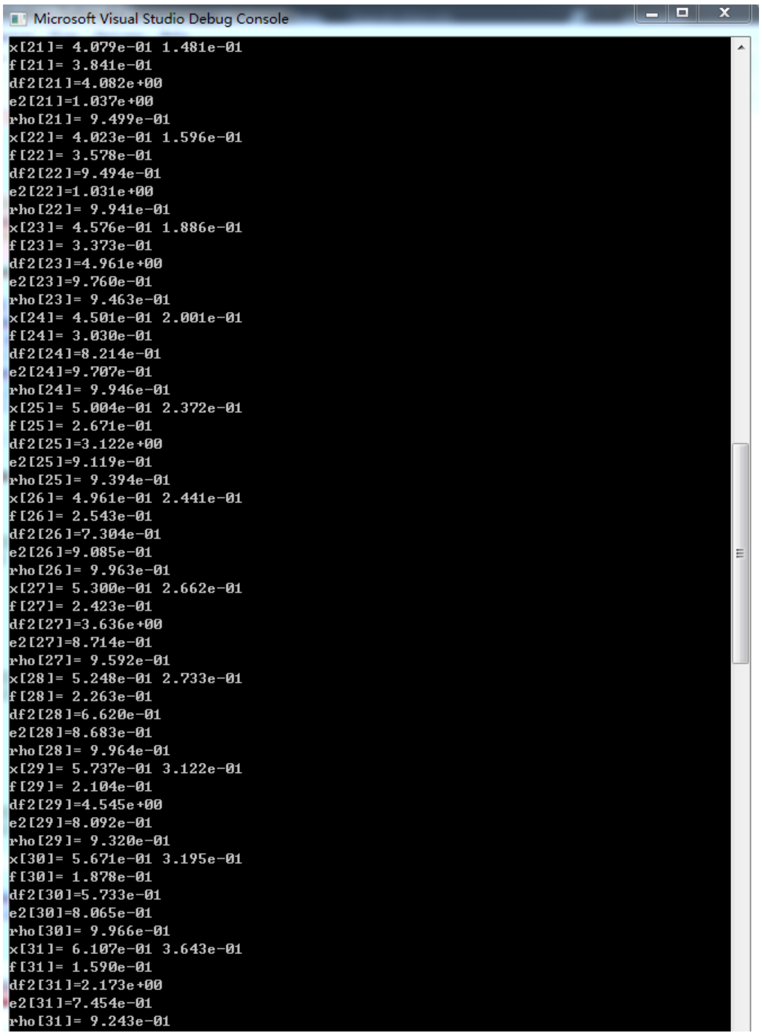


Figure 3: Cauchy point method (Step 21 to 31)

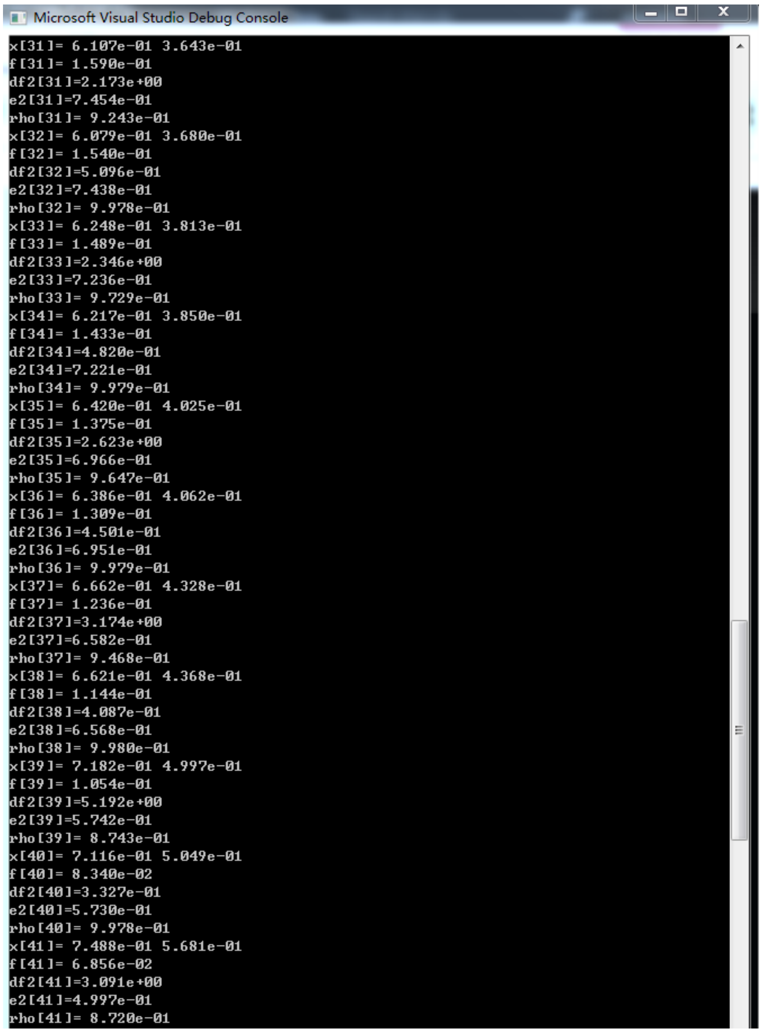


Figure 4: Cauchy point method (Step 31 to 41)

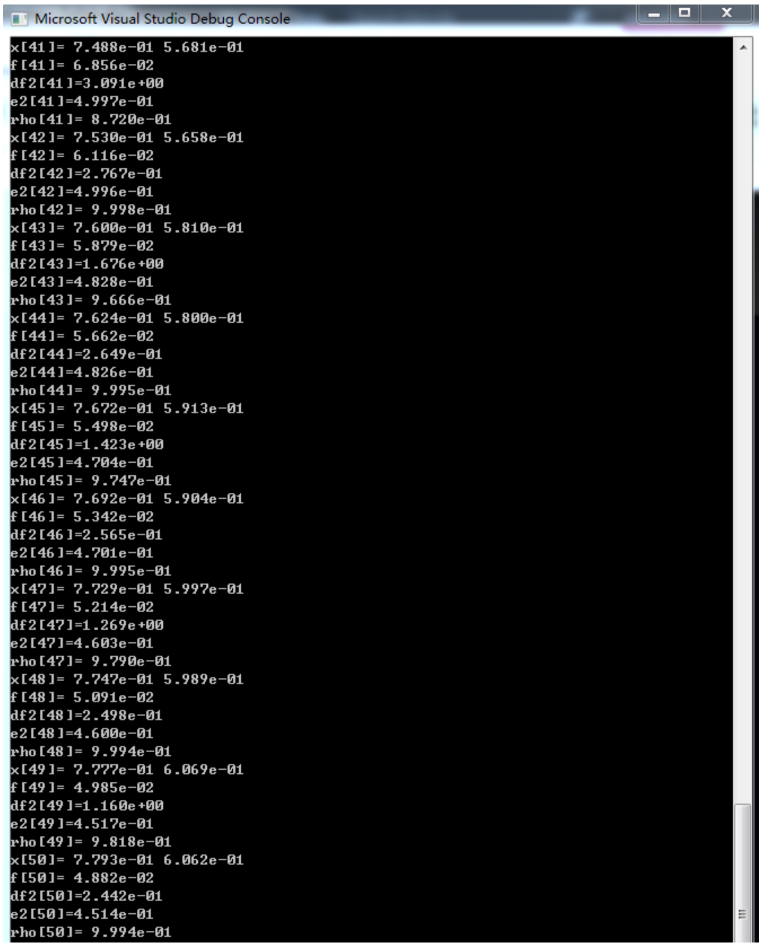
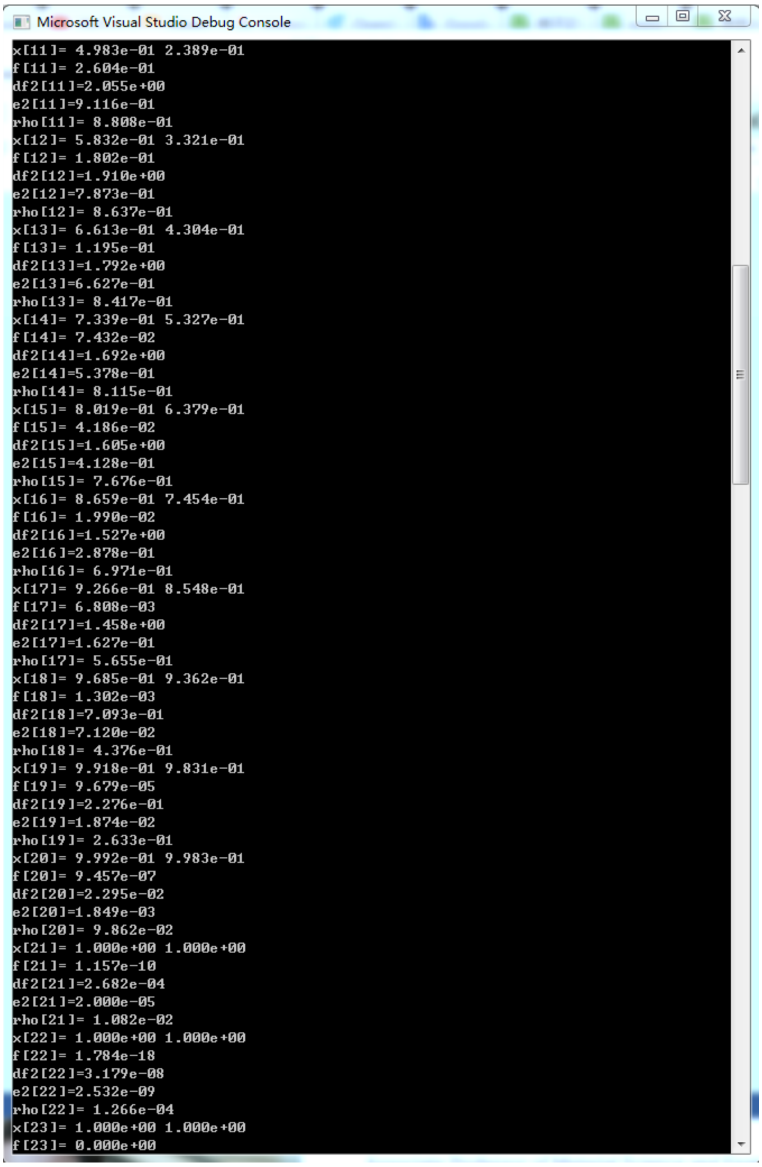


Figure 5: Cauchy point method (Step 41 to 50)

Microsoft Visual Studio Debug Console

```
Initial guess
x0=-0.62
y0=-0.38
x[0]= -6.200e-01 -3.800e-01
f[0]= 6.106e+01
df2[0]=2.461e+02
e2[0]=2.128e+00
x[1]= -4.862e-01 1.252e-01
f[1]= 3.446e+00
df2[1]=3.317e+01
e2[1]=1.725e+00
rho[1]= 8.104e-01
x[2]= -4.223e-01 1.742e-01
f[2]= 2.024e+00
df2[2]=3.628e+00
e2[2]=1.645e+00
rho[2]= 9.536e-01
x[3]= -4.223e-01 1.742e-01
f[3]= 2.024e+00
df2[3]=3.628e+00
e2[3]=1.645e+00
rho[3]= 1.000e+00
x[4]= -3.161e-01 1.001e-01
f[4]= 1.732e+00
df2[4]=2.613e+00
e2[4]=1.594e+00
rho[4]= 9.694e-01
x[5]= -1.929e-01 4.131e-02
f[5]= 1.425e+00
df2[5]=2.226e+00
e2[5]=1.530e+00
rho[5]= 9.599e-01
x[6]= -7.671e-02 -4.838e-03
f[6]= 1.171e+00
df2[6]=3.281e+00
e2[6]=1.473e+00
rho[6]= 9.623e-01
x[7]= 5.569e-02 -7.938e-03
f[7]= 9.039e-01
df2[7]=2.752e+00
e2[7]=1.381e+00
rho[7]= 9.378e-01
x[8]= 1.847e-01 2.177e-02
f[8]= 6.800e-01
df2[8]=2.569e+00
e2[8]=1.273e+00
rho[8]= 9.220e-01
x[9]= 3.014e-01 7.854e-02
f[9]= 5.032e-01
df2[9]=2.456e+00
e2[9]=1.156e+00
rho[9]= 9.080e-01
x[10]= 4.051e-01 1.532e-01
f[10]= 3.658e-01
df2[10]=2.246e+00
e2[10]=1.035e+00
rho[10]= 8.950e-01
x[11]= 4.983e-01 2.389e-01
f[11]= 2.604e-01
df2[11]=2.055e+00
e2[11]=9.116e-01
rho[11]= 8.808e-01
```

Figure 6: Dogleg method (Step 0 to 11)



```
Microsoft Visual Studio Debug Console
x[11]= 4.983e-01 2.389e-01
f[11]= 2.604e-01
df2[11]=2.055e+00
e2[11]=9.116e-01
rho[11]= 8.808e-01
x[12]= 5.832e-01 3.321e-01
f[12]= 1.802e-01
df2[12]=1.910e+00
e2[12]=7.873e-01
rho[12]= 8.637e-01
x[13]= 6.613e-01 4.304e-01
f[13]= 1.195e-01
df2[13]=1.792e+00
e2[13]=6.627e-01
rho[13]= 8.417e-01
x[14]= 7.339e-01 5.327e-01
f[14]= 7.432e-02
df2[14]=1.692e+00
e2[14]=5.378e-01
rho[14]= 8.115e-01
x[15]= 8.019e-01 6.379e-01
f[15]= 4.186e-02
df2[15]=1.605e+00
e2[15]=4.128e-01
rho[15]= 7.676e-01
x[16]= 8.659e-01 7.454e-01
f[16]= 1.990e-02
df2[16]=1.527e+00
e2[16]=2.878e-01
rho[16]= 6.971e-01
x[17]= 9.266e-01 8.548e-01
f[17]= 6.808e-03
df2[17]=1.458e+00
e2[17]=1.627e-01
rho[17]= 5.655e-01
x[18]= 9.685e-01 9.362e-01
f[18]= 1.302e-03
df2[18]=7.093e-01
e2[18]=7.120e-02
rho[18]= 4.376e-01
x[19]= 9.918e-01 9.831e-01
f[19]= 9.679e-05
df2[19]=2.276e-01
e2[19]=1.874e-02
rho[19]= 2.633e-01
x[20]= 9.992e-01 9.983e-01
f[20]= 9.457e-07
df2[20]=2.295e-02
e2[20]=1.849e-03
rho[20]= 9.862e-02
x[21]= 1.000e+00 1.000e+00
f[21]= 1.157e-10
df2[21]=2.682e-04
e2[21]=2.000e-05
rho[21]= 1.082e-02
x[22]= 1.000e+00 1.000e+00
f[22]= 1.784e-18
df2[22]=3.179e-08
e2[22]=2.532e-09
rho[22]= 1.266e-04
x[23]= 1.000e+00 1.000e+00
f[23]= 0.000e+00
```

Figure 7: Dogleg method (Step 11 to 22)

The output of the program are shown in Figure 6 and Figure 7.

Exercise 5.4 [1 points] Compare the convergence rates of the methods implemented in Exercises 5.2 and 5.3 with the convergence of the gradient and the Newton's methods. How can you characterize the convergence behaviour of these methods?

Solution:
Sorting by convergence rate from low to high: Cauchy point method, Gradient method, Newton's method, Dogleg method. Both Cauchy point method and Gradient method are linear convergence. Both Dogleg method and Newton's method are quadratic convergence. The convergence rate of Cauchy point method and Dogleg method are shown in Figure 8 and Figure 9.

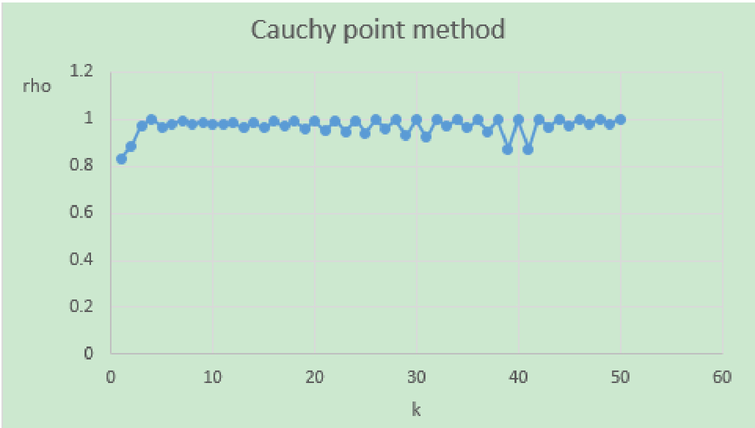


Figure 8: The convergence rate of Cauchy point method

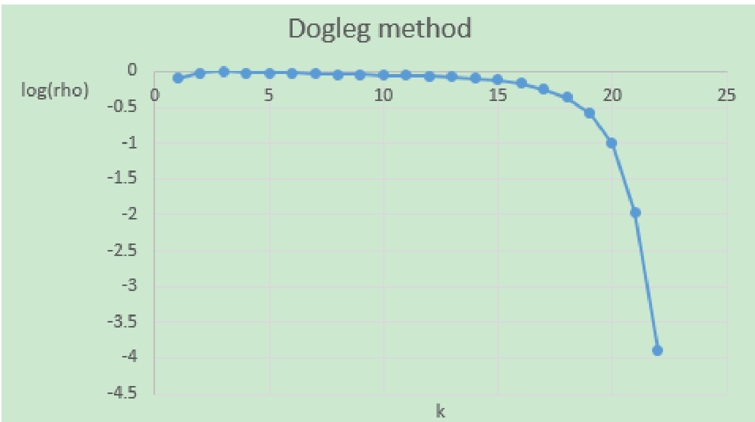


Figure 9: The convergence rate of Dogleg method