Numerical Optimization

Shuai Lu 170742

Homework 4

Exercise 4.1 [2 points] Let $f : R^n \to R$ be a smooth function, $\mathbf{B} \in R^{n\times n}$ a symmetric positive definite matrix and $\mathbf{x} \in R^n$ such a point that $\nabla f(\mathbf{x}) \neq 0$. Consider the direction $p = -B\nabla f(\mathbf{x})$. Prove that there is an interval $(0, S) \subset R$ such that for every $s \in (0, S)$, $f(\mathbf{x} + s\mathbf{p}) < f(\mathbf{x})$, i.e. that f decreases along p

Solution:

From Taylor expansion:

$$f(\mathbf{x}+s\mathbf{p}) - f(\mathbf{x}) = s\nabla^\top f(\mathbf{x})\,\mathbf{p} + o\left(s^2\right) = -s\nabla^\top f(\mathbf{x})\,\mathbf{B}\nabla f(\mathbf{x}) + o\left(s^2\right)$$

Since $\mathbf{B}$ is symmetric positive definite and $\nabla f(\mathbf{x}) \neq 0$, $-s\nabla^\top f(\mathbf{x})\,\mathbf{B}\nabla f(\mathbf{x}) < 0$. There is a S small enough, when $s \in (0, S)$, $-s\nabla^\top f(\mathbf{x})\,\mathbf{B}\nabla f(\mathbf{x}) + o\left(s^2\right) < 0$. So we found S, for every $s \in (0, S)$, $f(\mathbf{x} + s\mathbf{p}) < f(\mathbf{x})$.

Exercise 4.2 [2points] Prove that for any quadratic minimization problem $f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^\top\mathbf{B}\mathbf{x} + \mathbf{c}^\top\mathbf{x} + d \to \min$ with a symmetric positive definite matrix $\mathbf{B} \in R^{n\times n}, \mathbf{b} \in R^n, d \in R$, the Newton's method $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k, \quad \mathbf{p}_k = -\left[\nabla^2 f(\mathbf{x}_k)\right]^{-1}\nabla f(\mathbf{x}_k)$ (without linesearch) convergence in one iteration (i.e. for any x_0 , the iterates $x_1 = x_2 = ...$ are equal to the exact minimizer of the problem).
Solution:

Since $\nabla f(\mathbf{x}_k) = \mathbf{B}\mathbf{x}_k + \mathbf{c}$ and $\nabla^2 f(\mathbf{x}_k) = \mathbf{B}$, $\mathbf{p}_k = -\left[\nabla^2 f(\mathbf{x}_k)\right]^{-1}\nabla f(\mathbf{x}_k) = -\mathbf{B}^{-1}(\mathbf{B}\mathbf{x}_k + \mathbf{c})$. So we have $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k = \mathbf{x}_k - \mathbf{B}^{-1}(\mathbf{B}\mathbf{x}_k + \mathbf{c}) = -\mathbf{B}^{-1}\mathbf{c}$ for any $\mathbf{x_0}$. Since $\nabla f(\mathbf{x}_{k+1}) = \mathbf{B}\mathbf{x}_{k+1} + \mathbf{c} = -\mathbf{B}\mathbf{B}^{-1}\mathbf{c} + \mathbf{c} = 0$ and $\nabla^2 f(\mathbf{x}_{k+1}) = \mathbf{B}$ is positive definite, $\mathbf{x}_{k+1}$ is the minimizer of the problem.

Exercise 4.3 In your code from Exercise 3.3 (the gradient method with the backtracking linesearch), set $\mathbf{p}_k = -\left[\nabla^2 f(\mathbf{x}_k)\right]^{-1}\nabla f(\mathbf{x}_k)$, i.e. implement the Newton's method with the backtracking linesearch. Note that the computation of the direction should be performed by solving the linear system

$$\mathbf{B}_k\mathbf{p}_k = -\nabla f(\mathbf{x}_k), \quad \mathbf{B}_k := \nabla^2 f(\mathbf{x}_k)$$

rather than by the computation of the inverse of the Hessian. (You can apply the Gaussian elimination from Ex. 1.3(b).) Use the same parameters of the backtracking linesearch as in Ex. 3.3(a) and the same termination criterion as in Ex. 3.3(b). Use Ex. 4.2 to debug the code: For the problem from Ex.2.3, the method should converge in one iteration, and the line search should always choose the unit step.
(a) [5 points] Apply this method to the Rosenbrock function

$$f(x, y) = (1-x)^2 + 100\left(y - x^2\right)^2$$

(Also evaluate its minimizer and the minimal value analytically.) Use the analytical formulas for the gradient and the Hessian. Start with $\mathbf{x}_0 = (-0.62, -0.38)^\top$. Print the the same data as in Ex. 3.3(a).

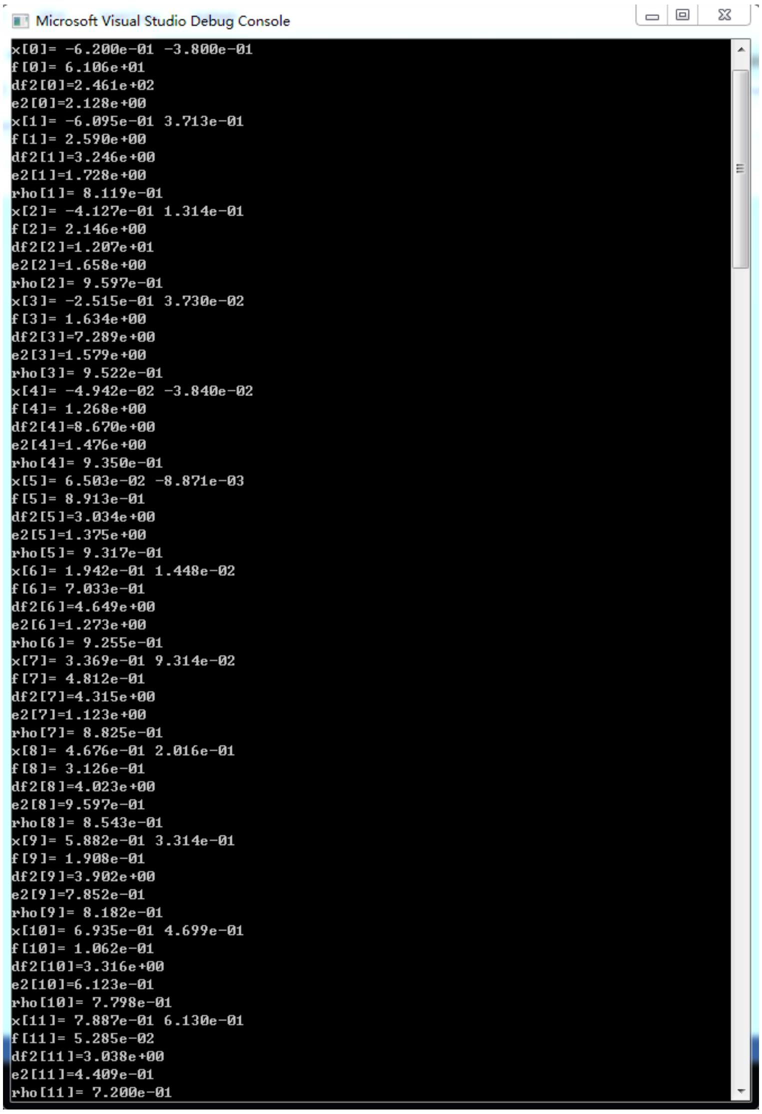
Solution:

The output of the program for the test function from Ex.2.3 ($f(x, y) := 5x^2 + 5y^2 - 6xy + 10x + 6y + 5$) is shown in Figure 1. The method converge in only one iteration.

The output of the program for the Rosenbrock function is shown in Figure 2 and 3. The method converge in 17 iterations.
(b) [3 points] Apply the gradient method with the same parametes as in Ex. 4.3(a) to the Rosenbrock function: Merely set $\mathbf{B}_k = \mathbf{I}$ (the identity matrix) in the code from Ex. 4.3(a). Compare the convergence rate of the methods

```
Microsoft Visual Studio Debug Console
Initial guess
x0=-0.62
y0=-0.38
x[0]= -6.200e-01 -3.800e-01
f[0]= -2.250e+00
df2[0]=8.486e+00
e2[0]=2.121e+00
x[1]= -2.125e+00 -1.875e+00
f[1]= -1.125e+01
df2[1]=0.000e+00
e2[1]=0.000e+00
rho[1]= 0.000e+00
```

Figure 1: The output of the test function $f(x, y) := 5x^2 + 5y^2 - 6xy + 10x + 6y + 5$



```
Microsoft Visual Studio Debug Console
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]= -6.095e-01 3.713e-01
f[1]= 2.590e+00
df2[1]=3.246e+00
e2[1]=1.728e+00
rho[1]= 8.119e-01
x[2]= -4.127e-01 1.314e-01
f[2]= 2.146e+00
df2[2]=1.207e+01
e2[2]=1.658e+00
rho[2]= 9.597e-01
x[3]= -2.515e-01 3.730e-02
f[3]= 1.634e+00
df2[3]=7.289e+00
e2[3]=1.579e+00
rho[3]= 9.522e-01
x[4]= -4.942e-02 -3.840e-02
f[4]= 1.268e+00
df2[4]=8.670e+00
e2[4]=1.476e+00
rho[4]= 9.350e-01
x[5]= 6.503e-02 -8.871e-03
f[5]= 8.913e-01
df2[5]=3.034e+00
e2[5]=1.375e+00
rho[5]= 9.317e-01
x[6]= 1.942e-01 1.448e-02
f[6]= 7.033e-01
df2[6]=4.649e+00
e2[6]=1.273e+00
rho[6]= 9.255e-01
x[7]= 3.369e-01 9.314e-02
f[7]= 4.812e-01
df2[7]=4.315e+00
e2[7]=1.123e+00
rho[7]= 8.825e-01
x[8]= 4.676e-01 2.016e-01
f[8]= 3.126e-01
df2[8]=4.023e+00
e2[8]=9.597e-01
rho[8]= 8.543e-01
x[9]= 5.882e-01 3.314e-01
f[9]= 1.908e-01
df2[9]=3.902e+00
e2[9]=7.852e-01
rho[9]= 8.182e-01
x[10]= 6.935e-01 4.699e-01
f[10]= 1.062e-01
df2[10]=3.316e+00
e2[10]=6.123e-01
rho[10]= 7.798e-01
x[11]= 7.887e-01 6.130e-01
f[11]= 5.285e-02
df2[11]=3.038e+00
e2[11]=4.409e-01
rho[11]= 7.200e-01
```

Figure 2: The output of the Rosenbrock function (Step 1 to 11)

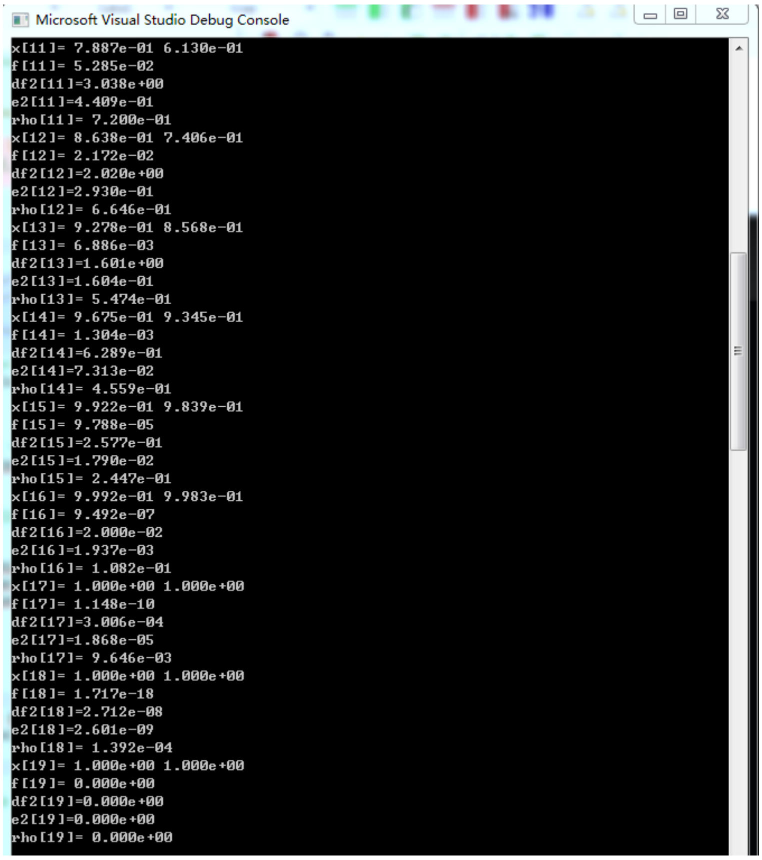


Figure 3: The output of the Rosenbrock function (Step 11 to 19)

from Ex. 4.3(a) and 4.3(b). What can you say about the convergence orders of these methods?

Solution:
The output of the program is shown in Figures 4 to 8. Compared with the converge rate of the previous method, the converge rate of this method is slower. It fails to converge to the exact solution in 50 iterations.

Exercise 4.4 [3 points] Explain in terms of the minimization of a quadratic objective function over R^d why the conjugate gradient method converges in at most d iterations and the gradient method does not.

Solution:
The direction set generated by the conjugate gradient method is a conjugate direction set $\{p_i\}$ and they are independent. $\forall x_0, \exists\{\sigma_i\}$ s.t. $x^* - x_0 = \sigma_0 p_0 + \sigma_1 p_1 + ... + \sigma_{d-1} p_{d-1}$. Since $\{p_i\}$ are conjugate w.r.t. A and x_k from conjugate gradient method is equal to $x_0 + \alpha_0 p_0 + ... + \alpha_{k-1} p_{k-1}$, $p_k^T A(x_k - x_0) = 0$. So

$$\sigma_k = \frac{p_k^T A(x^* - x_0)}{p_k^T A p_k} = \frac{p_k^T A(x^* - x_k)}{p_k^T A p_k} = \frac{p_k^T (b - Ax_k)}{p_k^T A p_k} = \frac{p_k^T r_k}{p_k^T A p_k}$$

which is the step size of conjugate gradient method at kth step. So $\forall x_0$ the conjugate gradient method product $x_k = x_0 + \sigma_0 p_0 + ... + \sigma_{k-1} p_{k-1}$ at kth step. When $k = d$, $x_d = x_0 + \sigma_0 p_0 + ... + \sigma_{d-1} p_{d-1} = x_*$. So it converges to the exact solution at most d step. The direction set of gradient method is not conjugate, so it can not promise to converge at most d step.

(b) [1 point] Why is the conjugate gradient method (in its pure form) not used as a direct solver for large sparse systems of linear equations with positive difinite matrices? (The conjugate gradient method is used as an iterative method, typically with preconditioning.)

Solution:
The original condition number of the Hessian matrix may be very large, so we usually choose a preconditioner $\mathbf{C}$ to make the condition number of $\mathbf{C}^{-\top}\mathbf{A}\mathbf{C}^{-1}$ smaller. The complexity of the conjugate gradient method is $O(n)$. When d is very large, the iteration steps of the conjugate gradient method for convergence will be very large as well.

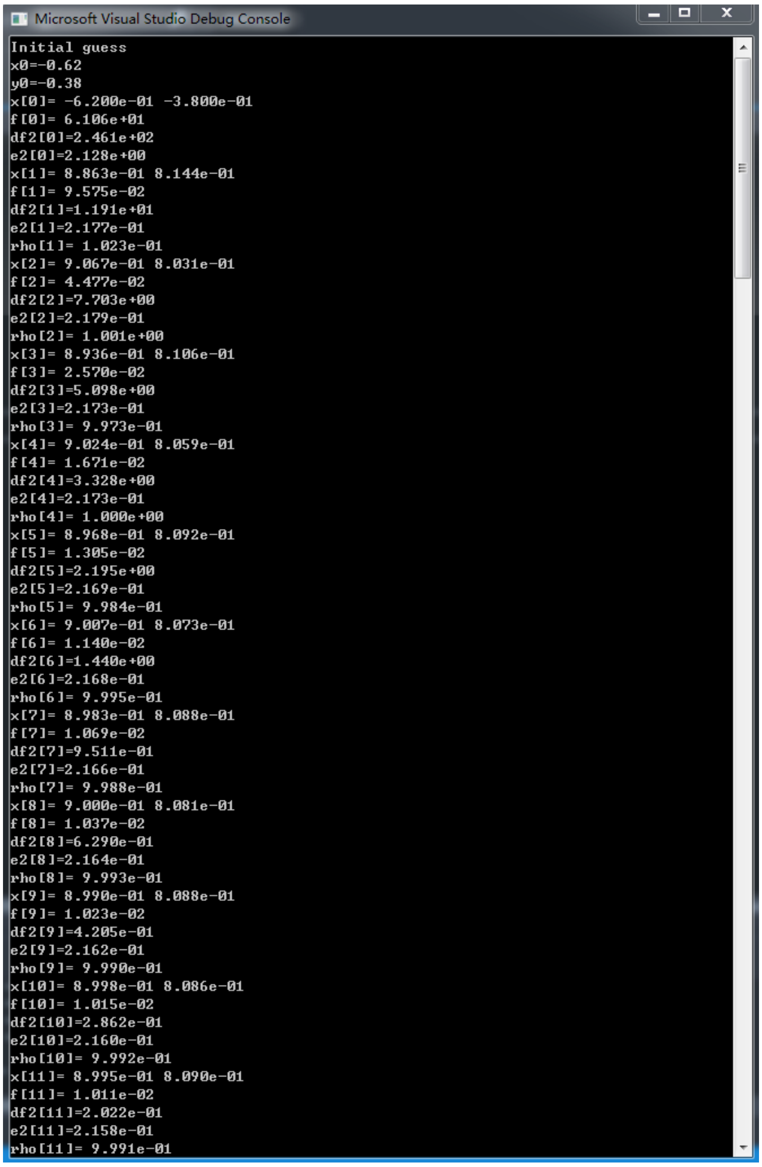
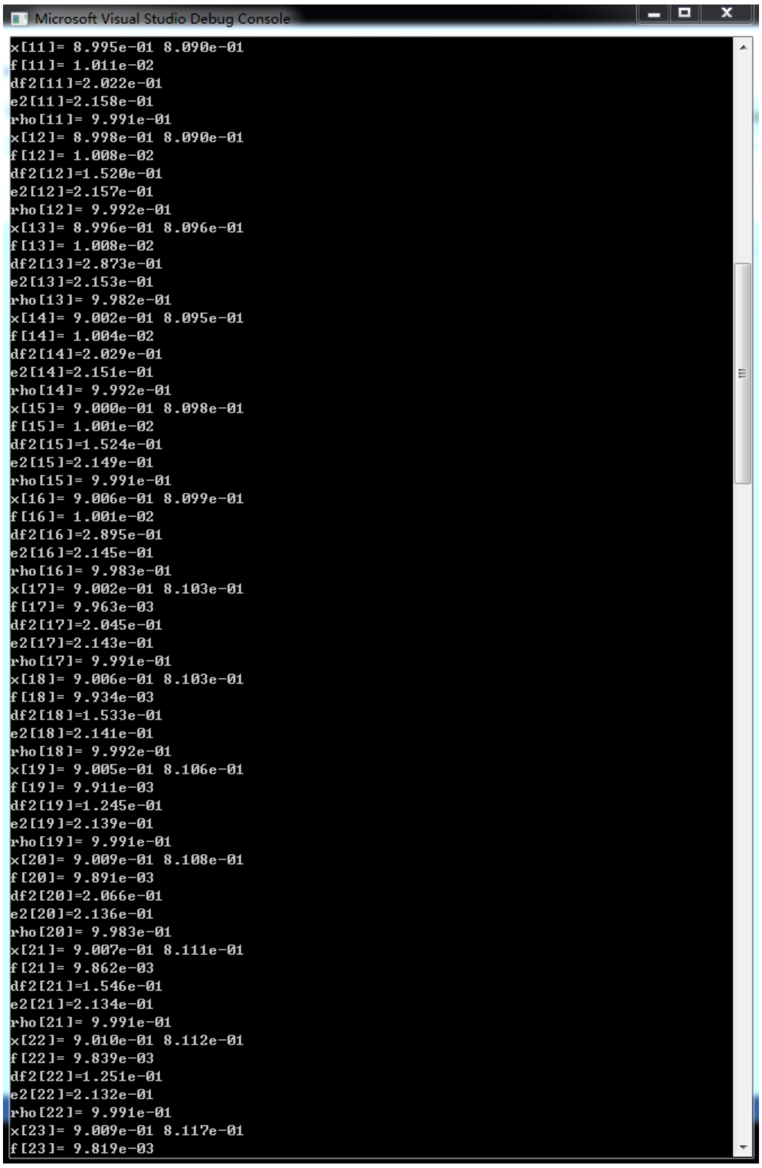
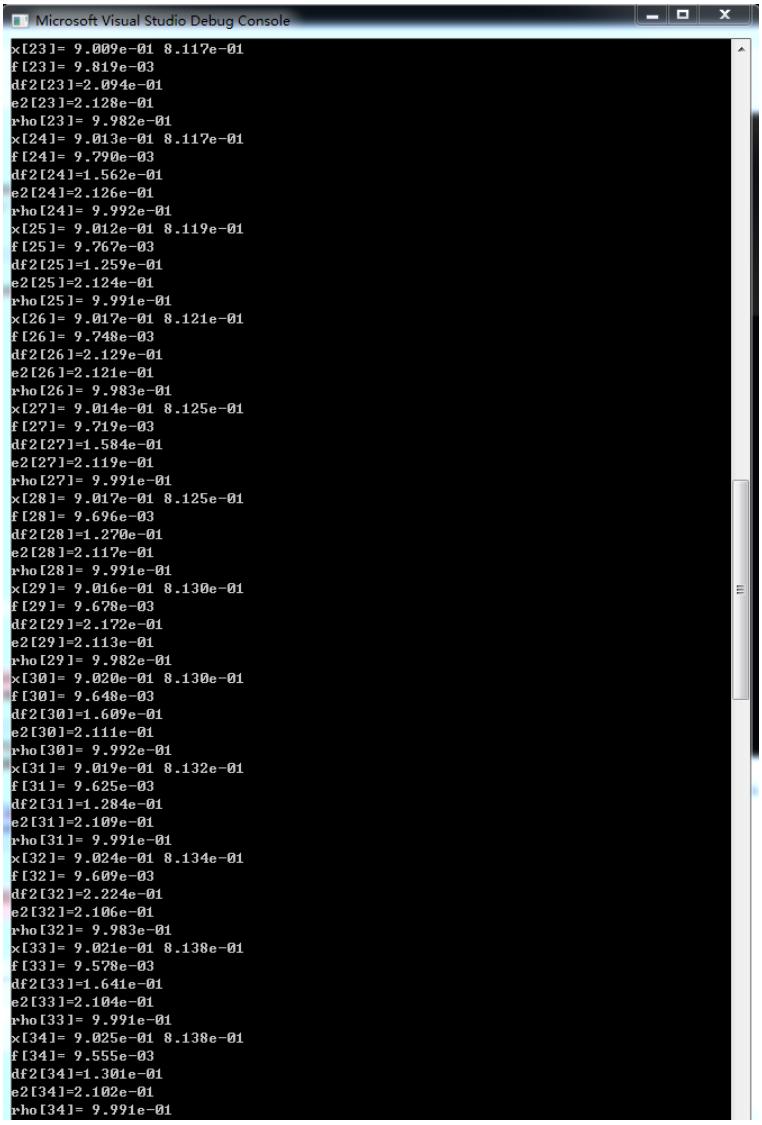


Figure 4: The output of the Rosenbrock function ($\mathbf{B}_k = \mathbf{I}$, Step 1 to 11)



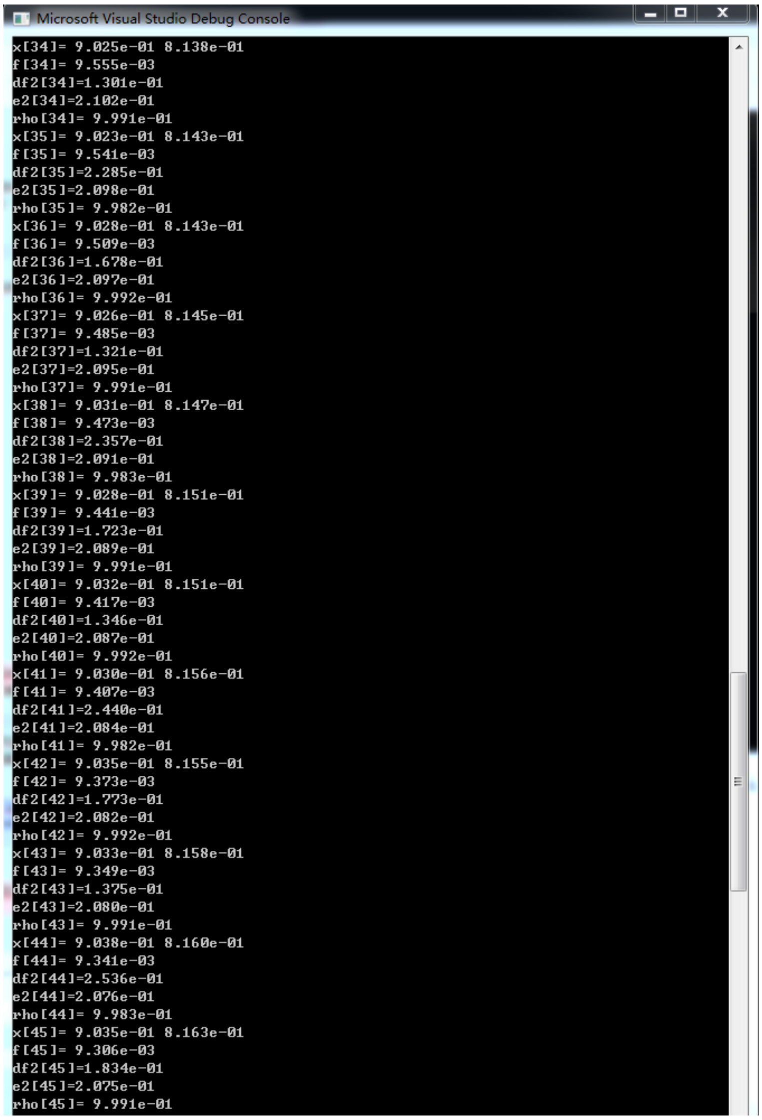
```
Microsoft Visual Studio Debug Console
x[11]= 8.995e-01 8.090e-01
f[11]= 1.011e-02
df2[11]=2.022e-01
e2[11]=2.158e-01
rho[11]= 9.991e-01
x[12]= 8.998e-01 8.090e-01
f[12]= 1.008e-02
df2[12]=1.520e-01
e2[12]=2.157e-01
rho[12]= 9.992e-01
x[13]= 8.996e-01 8.096e-01
f[13]= 1.008e-02
df2[13]=2.873e-01
e2[13]=2.153e-01
rho[13]= 9.982e-01
x[14]= 9.002e-01 8.095e-01
f[14]= 1.004e-02
df2[14]=2.029e-01
e2[14]=2.151e-01
rho[14]= 9.992e-01
x[15]= 9.000e-01 8.098e-01
f[15]= 1.001e-02
df2[15]=1.524e-01
e2[15]=2.149e-01
rho[15]= 9.991e-01
x[16]= 9.006e-01 8.099e-01
f[16]= 1.001e-02
df2[16]=2.895e-01
e2[16]=2.145e-01
rho[16]= 9.983e-01
x[17]= 9.002e-01 8.103e-01
f[17]= 9.963e-03
df2[17]=2.045e-01
e2[17]=2.143e-01
rho[17]= 9.991e-01
x[18]= 9.006e-01 8.103e-01
f[18]= 9.934e-03
df2[18]=1.533e-01
e2[18]=2.141e-01
rho[18]= 9.992e-01
x[19]= 9.005e-01 8.106e-01
f[19]= 9.911e-03
df2[19]=1.245e-01
e2[19]=2.139e-01
rho[19]= 9.991e-01
x[20]= 9.009e-01 8.108e-01
f[20]= 9.891e-03
df2[20]=2.066e-01
e2[20]=2.136e-01
rho[20]= 9.983e-01
x[21]= 9.007e-01 8.111e-01
f[21]= 9.862e-03
df2[21]=1.546e-01
e2[21]=2.134e-01
rho[21]= 9.991e-01
x[22]= 9.010e-01 8.112e-01
f[22]= 9.839e-03
df2[22]=1.251e-01
e2[22]=2.132e-01
rho[22]= 9.991e-01
x[23]= 9.009e-01 8.117e-01
f[23]= 9.819e-03
```

Figure 5: The output of the Rosenbrock function ($\mathbf{B}_k = \mathbf{I}$, Step 11 to 23)



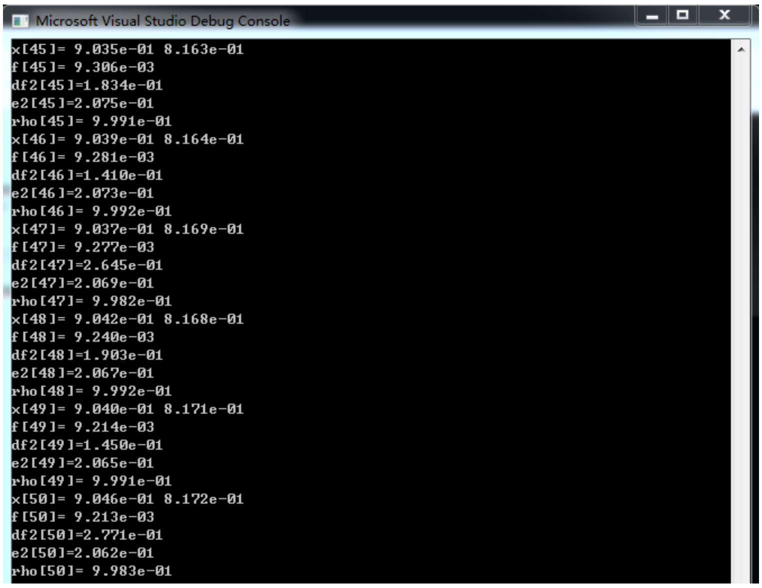
```
Microsoft Visual Studio Debug Console
x[23]= 9.009e-01 8.117e-01
f[23]= 9.819e-03
df2[23]=2.094e-01
e2[23]=2.128e-01
rho[23]= 9.982e-01
x[24]= 9.013e-01 8.117e-01
f[24]= 9.790e-03
df2[24]=1.562e-01
e2[24]=2.126e-01
rho[24]= 9.992e-01
x[25]= 9.012e-01 8.119e-01
f[25]= 9.767e-03
df2[25]=1.259e-01
e2[25]=2.124e-01
rho[25]= 9.991e-01
x[26]= 9.017e-01 8.121e-01
f[26]= 9.748e-03
df2[26]=2.129e-01
e2[26]=2.121e-01
rho[26]= 9.983e-01
x[27]= 9.014e-01 8.125e-01
f[27]= 9.719e-03
df2[27]=1.584e-01
e2[27]=2.119e-01
rho[27]= 9.991e-01
x[28]= 9.017e-01 8.125e-01
f[28]= 9.696e-03
df2[28]=1.270e-01
e2[28]=2.117e-01
rho[28]= 9.991e-01
x[29]= 9.016e-01 8.130e-01
f[29]= 9.678e-03
df2[29]=2.172e-01
e2[29]=2.113e-01
rho[29]= 9.982e-01
x[30]= 9.020e-01 8.130e-01
f[30]= 9.648e-03
df2[30]=1.609e-01
e2[30]=2.111e-01
rho[30]= 9.992e-01
x[31]= 9.019e-01 8.132e-01
f[31]= 9.625e-03
df2[31]=1.284e-01
e2[31]=2.109e-01
rho[31]= 9.991e-01
x[32]= 9.024e-01 8.134e-01
f[32]= 9.609e-03
df2[32]=2.224e-01
e2[32]=2.106e-01
rho[32]= 9.983e-01
x[33]= 9.021e-01 8.138e-01
f[33]= 9.578e-03
df2[33]=1.641e-01
e2[33]=2.104e-01
rho[33]= 9.991e-01
x[34]= 9.025e-01 8.138e-01
f[34]= 9.555e-03
df2[34]=1.301e-01
e2[34]=2.102e-01
rho[34]= 9.991e-01
```

Figure 6: The output of the Rosenbrock function ($\mathbf{B}_k = \mathbf{I}$, Step 23 to 34)



```
Microsoft Visual Studio Debug Console
x[34]= 9.025e-01 8.138e-01
f[34]= 9.555e-03
df2[34]=1.301e-01
e2[34]=2.102e-01
rho[34]= 9.991e-01
x[35]= 9.023e-01 8.143e-01
f[35]= 9.541e-03
df2[35]=2.285e-01
e2[35]=2.098e-01
rho[35]= 9.982e-01
x[36]= 9.028e-01 8.143e-01
f[36]= 9.509e-03
df2[36]=1.678e-01
e2[36]=2.097e-01
rho[36]= 9.992e-01
x[37]= 9.026e-01 8.145e-01
f[37]= 9.485e-03
df2[37]=1.321e-01
e2[37]=2.095e-01
rho[37]= 9.991e-01
x[38]= 9.031e-01 8.147e-01
f[38]= 9.473e-03
df2[38]=2.357e-01
e2[38]=2.091e-01
rho[38]= 9.983e-01
x[39]= 9.028e-01 8.151e-01
f[39]= 9.441e-03
df2[39]=1.723e-01
e2[39]=2.089e-01
rho[39]= 9.991e-01
x[40]= 9.032e-01 8.151e-01
f[40]= 9.417e-03
df2[40]=1.346e-01
e2[40]=2.087e-01
rho[40]= 9.992e-01
x[41]= 9.030e-01 8.156e-01
f[41]= 9.407e-03
df2[41]=2.440e-01
e2[41]=2.084e-01
rho[41]= 9.982e-01
x[42]= 9.035e-01 8.155e-01
f[42]= 9.373e-03
df2[42]=1.773e-01
e2[42]=2.082e-01
rho[42]= 9.992e-01
x[43]= 9.033e-01 8.158e-01
f[43]= 9.349e-03
df2[43]=1.375e-01
e2[43]=2.080e-01
rho[43]= 9.991e-01
x[44]= 9.038e-01 8.160e-01
f[44]= 9.341e-03
df2[44]=2.536e-01
e2[44]=2.076e-01
rho[44]= 9.983e-01
x[45]= 9.035e-01 8.163e-01
f[45]= 9.306e-03
df2[45]=1.834e-01
e2[45]=2.075e-01
rho[45]= 9.991e-01
```

Figure 7: The output of the Rosenbrock function ($\mathbf{B}_k = \mathbf{I}$, Step 34 to 45)



```
Microsoft Visual Studio Debug Console
x[45]= 9.035e-01 8.163e-01
f[45]= 9.306e-03
df2[45]=1.834e-01
e2[45]=2.075e-01
rho[45]= 9.991e-01
x[46]= 9.039e-01 8.164e-01
f[46]= 9.281e-03
df2[46]=1.410e-01
e2[46]=2.073e-01
rho[46]= 9.992e-01
x[47]= 9.037e-01 8.169e-01
f[47]= 9.277e-03
df2[47]=2.645e-01
e2[47]=2.069e-01
rho[47]= 9.982e-01
x[48]= 9.042e-01 8.168e-01
f[48]= 9.240e-03
df2[48]=1.903e-01
e2[48]=2.067e-01
rho[48]= 9.992e-01
x[49]= 9.040e-01 8.171e-01
f[49]= 9.214e-03
df2[49]=1.450e-01
e2[49]=2.065e-01
rho[49]= 9.991e-01
x[50]= 9.046e-01 8.172e-01
f[50]= 9.213e-03
df2[50]=2.771e-01
e2[50]=2.062e-01
rho[50]= 9.983e-01
```

Figure 8: The output of the Rosenbrock function ($\mathbf{B}_k = \mathbf{I}$, Step 45 to 50)