Otimização em Engenharia Mecânica
4ª Lista de Exercícios

por Allan Moreira de Carvalho

Prof. Cícero

Exercício 1. Verificar se os seguintes vetores servem como direções conjugadas para minimizar a função

$$f(\mathbf{x}) = 2x_1^2 + 16x_2^2 - 2x_1x_2 - x_1 - 6x_2 - 5$$

a) $\mathbf{s}_1^T = \{\ 15\ \ -1\ \}$ e $\mathbf{s}_2^T = \{\ 1\ \ 1\ \}$

b) $\mathbf{s}_1^T = \{\ -1\ \ 15\ \}$ e $\mathbf{s}_2^T = \{\ 1\ \ 1\ \}$

Solution. A função $f(\mathbf{x})$ escrita na forma matricial é

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{B}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} \\ &= \{\ -1\ \ -6\ \}\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \frac{1}{2}\{\ x_1\ \ x_2\ \}\begin{bmatrix} 4 & -2 \\ -2 & 32 \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} \end{aligned}$$

a) A direções $\mathbf{s}_1^T = \{\ 15\ \ -1\ \}$ e $\mathbf{s}_2^T = \{\ 1\ \ 1\ \}$ são conjugadas, uma vez que

$$\begin{aligned} \mathbf{s}_1^T\mathbf{A}\mathbf{s}_2 &= \{\ 15\ \ -1\ \}\begin{bmatrix} 4 & -2 \\ -2 & 32 \end{bmatrix}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \\ &= \{\ 62\ \ -62\ \}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \\ &= 0 \end{aligned}$$

b) A direções $\mathbf{s}_1^T = \{\ -1\ \ 15\ \}$ e $\mathbf{s}_2^T = \{\ 1\ \ 1\ \}$ não são conjugadas, uma vez que

$$\begin{aligned} \mathbf{s}_1^T\mathbf{A}\mathbf{s}_2 &= \{\ -1\ \ 15\ \}\begin{bmatrix} 4 & -2 \\ -2 & 32 \end{bmatrix}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \\ &= \{\ -34\ \ 482\ \}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \\ &= 448 \end{aligned}$$

Exercício 2. Demonstre que as direções de busca usadas no método de Fletcher-Reevs são **A**-conjugadas quando minimizam a seguinte função. Adote como ponte de partida $\mathbf{x}_1^T = \{\ 1\ \ 1\ \}$

$$f(\mathbf{x}) = x_1^2 + 4x_2^2$$

Solution. No método de Fletcher-Reevs a primeira direção de busca aponda na direção inversa do gradiente da função f (*Steepest Descent*)

$$\mathbf{s}_1 = -\nabla f(\mathbf{x}_1)$$

, que nesse caso é

$$\begin{aligned} \mathbf{s}_1 &= -\begin{Bmatrix} 2x_1 \\ 8x_2 \end{Bmatrix}\Bigg|_{\mathbf{x}_1} \\ &= -\begin{Bmatrix} 2 \\ 8 \end{Bmatrix} \end{aligned}$$

A segunda direção de busca é determinada como uma combinação linear

$$\mathbf{s}_2 = -\nabla f(\mathbf{x}_2) + \frac{|\nabla f_2|^2}{|\nabla f_1|^2}\mathbf{s}_1$$

com

$$
\begin{aligned}
\frac{\partial f(\mathbf{x}_1+\lambda_1\mathbf{s}_1)}{\partial\lambda_1} &= 0\\
\frac{\partial f\left(\left\{\begin{array}{c} 1-2\,\lambda_1\\ 1-8\,\lambda_1\end{array}\right\}\right)}{\partial\lambda_1} &= 0\\
520.0\,\lambda_1-68.0 &= 0\\
\lambda_1 &= 0.13076923076923078
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{x}_2 &= \mathbf{x}_1+\lambda_1\mathbf{s}_1\\
&= \left\{\begin{array}{c} 0.73846154\\ -0.04615385\end{array}\right\}\\
\nabla f_2 &= \left\{\begin{array}{c} 1.47692308\\ -0.36923077\end{array}\right\}\\
\mathbf{s}_2 = -\nabla f(\mathbf{x}_2)+\frac{|\nabla f_2|^2}{|\nabla f_1|^2}\mathbf{s}_1 &= -\left\{\begin{array}{c} 1.47692308\\ -0.36923077\end{array}\right\}-\frac{\left|\left\{\begin{array}{c} 1.47692308\\ -0.36923077\end{array}\right\}\right|^2}{\left|\left\{\begin{array}{c} 2\\ 8\end{array}\right\}\right|^2}\left\{\begin{array}{c} 2\\ 8\end{array}\right\}\\
&= \left\{\begin{array}{c} -1.54508876\\ 0.09656805\end{array}\right\}
\end{aligned}
$$

Da função objetivo, temos uma matrix $\mathbf{A}$, tal que

$$
\begin{aligned}
f(\mathbf{x}) &= x_1^2+4x_2^2\\
&= \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}\\
&= \frac{1}{2}\{\begin{array}{cc} x_1 & x_2\end{array}\}\left[\begin{array}{cc} 2 & 0\\ 0 & 8\end{array}\right]\left\{\begin{array}{c} x_1\\ x_2\end{array}\right\}
\end{aligned}
$$

, e portanto

$$
\begin{aligned}
\mathbf{s}_1^T\mathbf{A}\mathbf{s}_2 &= -\{\begin{array}{cc} 2 & 8\end{array}\}\left[\begin{array}{cc} 2 & 0\\ 0 & 8\end{array}\right]\left\{\begin{array}{c} -1.54508876\\ 0.09656805\end{array}\right\}\\
&= -\{\begin{array}{cc} 4 & 64\end{array}\}\left\{\begin{array}{c} -1.54508876\\ 0.09656805\end{array}\right\}\\
&= -1.5999e-7\sim 0
\end{aligned}
$$

Apesar do erro numérico, o produto $\mathbf{s}_1^T\mathbf{A}\mathbf{s}_2$ é praticamente zero e portando as direções de busca são **A**-conjugadas.

Exercício 3. Dado o seguinte problema de programação linear, onde as variáveis x e y são irrestritas em sina, pede-se:

Maximizar:

$$f(\mathbf{x}) = 19x+7y$$

Sujeito à:

$$
\begin{aligned}
7x+6y &\leqslant 42\\
5x+9y &\leqslant 45\\
x-y &\leqslant 4
\end{aligned}
$$

a) Resolver o problea usando o método Simplex;

b) Usar um pacote ou biblioteca de sotware apropriado para conferir a sua resposta;

c) Incluir o print da tela do computador, mostrando a solução obtida através do software.

Solution. Esse problema é análogo à:

Minimizar:

$$f(\mathbf{x}) = -19x-7y$$

, usando variáveis auxiliares, a, b e c , o prolma pode ser reescrito na forma canonica

$$\begin{aligned} 7x+6y+a &= 42 \\ 5x+9y+b &= 45 \\ x-y+c &= 4 \\ -19x-7y-f &= 0 \end{aligned}$$

Uma solução viável é que $f=0$, quando todas as variáveis não-básicas são zero

$$x=y=0$$

, e as variáveis básicas são

$$a=42 \qquad b=45 \qquad c=4$$

No entendo essa solução não é ótima, pois todos os coeficientes da função objetivo são negativos

$$c_1''=-19 \qquad c_2''=-7$$

, escolhe-se o menor deles

$$\begin{aligned} c_s'' &= \min(c_i''<0) \\ &= -19 \end{aligned}$$

A variável não-básica x, será básica na próxima iteração. o novo sistema é obtido selecionando o pivô

$$\frac{b_r''}{a_{\text{rs}}''} = \min_{a_{\text{is}}^{\star}>0}\left(\frac{b_i''}{a_{\text{is}}''}\right)$$

$$\frac{b_1''}{a_{11}'} = \frac{42}{7} = 6 \qquad \frac{b_2''}{a_{21}'} = \frac{45}{5} = 9 \qquad \frac{b_3''}{a_{31}'} = \frac{4}{1} = 4$$

, então a_{31}'' será o pivô, o novo sistema de equação é

$$\begin{aligned} \frac{13}{7}y+\frac{1}{7}a-c &= 2 \\ \frac{14}{5}y+\frac{1}{5}b-c &= 5 \\ x-y+c &= 4 \\ -\frac{26}{19}y-\frac{1}{19}f+c &= 4 \end{aligned}$$

Com o coeficiente de y é negativo, a solução desse sistema não é ótima. A além disso a variável y será não-básica na próxima iteração Os candidatos à pivô

$$\frac{b_1''}{a_{12}'} = \frac{14}{13} \qquad \frac{b_2''}{a_{22}'} = \frac{25}{14}$$

, fazendo o pivoteamoento com $\frac{14}{23}$

$$\begin{aligned} y+\frac{1}{13}a-\frac{7}{13}c &= \frac{14}{13} \\ y+\frac{1}{14}b-\frac{5}{14}c &= \frac{25}{14} \\ x-y+c &= 4 \\ -y-\frac{1}{26}f+\frac{19}{26}c &= \frac{4\cdot 19}{26} \end{aligned}$$

eliminando as váriaveis y por pivoteamento

$$\begin{aligned} y+\frac{1}{13}a-\frac{7}{13}c &= \frac{14}{13} \\ \frac{1}{14}b-\frac{1}{13}a+\left(\frac{7}{13}-\frac{5}{14}\right)c &= \frac{25}{14}-\frac{14}{13} \\ x+\frac{1}{13}a+\frac{6}{13}c &= \frac{4\cdot 13+14}{13} \\ -\frac{1}{26}f+\frac{5}{26}c+\frac{1}{13}a &= \frac{4\cdot 19+2\cdot 14}{26} \end{aligned}$$

Que possuiu como solutção básica $a=0$, $c=0$

$$\begin{aligned} f &= -(4\cdot 19+2\cdot 14) \\ &= -104 \end{aligned}$$

Nesse caso os coeficientes d função objetivo são possítivos e a solução é ótima

$$y=\frac{14}{13}=1,0769 \qquad x=4+\frac{14}{13}=5,0769$$

O resultado foi validado com uso da biblioteca `scipy`, o resultado pode ser visto na Figura .

```python
from scipy import optimize
optimize.linprog([-19,-7],
                 A_ub=[[7,6],
                       [5,9],
                       [1,-1]],
                 b_ub=[42,45,4],method='simplex')
```

Python

```
     con: array([], dtype=float64)
     fun: -103.99999999999999
 message: 'Optimization terminated successfully.'
     nit: 4
   slack: array([0.00000000e+00, 9.92307692e+00, 4.44089210e-16])
  status: 0
 success: True
       x: array([5.07692308, 1.07692308])
```

Figura 1. Saída do método `simplex` da biblioteca `scipy`.

Exercício 4. Minimizar a função $f(\mathbf{x})=100(x_2-x_1^2)^2+(1-x_1)^2$, a partir do ponto $\mathbf{x}_0^T=\{\ -1,2\ \ 1,0\ \}$, usando três iterações

a) Do método de Fletcher-Reeves;

b) Do método de Newton;

Usar a função apropriada do Matlab, Octava ou Scilab para conferir sua resposta. Incluir o print da tela do computador, mostrando a solução obtida através do software.

Solution.

a) Método Fletcher-Reeves. Nesse método a primeira direção de busca é definida no sentido oposto ao gradiente da função objetivo, calculado no ponto de partida $\mathbf{x}_1^T=\{\ -1,2\ \ 1,0\ \}$. Portanto

$$\mathbf{s}_1 = -\nabla f(\mathbf{x}_1)=-\left\{\begin{array}{c} -400.0\,x_1\,(-x_1^2+x_2)-2.0\,(1.0-x_1) \\ 200.0\,(-x_1^2+x_2) \end{array}\right\}=-\left\{\begin{array}{c} -215.6 \\ -88 \end{array}\right\}$$

O próximo ponto, é determinado avançando um passo λ_1 na direção de $\mathbf{s}_1$. Onde o passo ótimo, é obtido resolvendo-se o conjunto de equações determinados por

$$\begin{aligned}
\frac{\partial f(\mathbf{x}_1+\lambda_1\mathbf{s}_1)}{\partial\lambda_1} &= 0 \\
\frac{\partial f\left(\left\{\begin{array}{c} 215.6\,\lambda_1-1.2 \\ 88.0\,\lambda_1+1.0 \end{array}\right\}\right)}{\partial\lambda_1} &= 0 \\
216070275688.96\ (0.00378630116239446-4.0\ (\lambda_1\ -\ 0.005565862\backslash \\
70871985)\ 1.0)\ (0.00189315058119723\ \lambda_1-(\lambda_1-0.0055658627087\backslash \\
1985)\ 2.0+2.15130747863321e-5)\ 1.0-92966.72\ (0.010204081632\backslash \\
6531-\lambda_1)\ 1.0 &= 0 \\
\lambda_1 &= 0.012248965891443657
\end{aligned}$$

Com esse novo passo, temos

$$\begin{aligned}
\mathbf{x}_2 &= \mathbf{x}_1+\lambda_1\mathbf{s}_1 \\
&= \left\{\begin{array}{c} 1.44087705 \\ 2.077909 \end{array}\right\}
\end{aligned}$$

A nova direção de busca é determinada usando os gradientes da função objetivo, calculados em $\mathbf{x_1}$ e $\mathbf{x}_2$, e a direção $\mathbf{s}_1$ de modo que

$$\begin{aligned}
\mathbf{s}_2 &= -\nabla f_2+\frac{|\nabla f_2|^2}{|\nabla f_1|^2}\mathbf{s}_1 \\
&= -\left\{\begin{array}{c} -0.14549683 \\ 0.35646724 \end{array}\right\}+\frac{\left|\left\{\begin{array}{c} -0.14549683 \\ 0.35646724 \end{array}\right\}\right|^2}{\left|\left\{\begin{array}{c} -215.6 \\ -88 \end{array}\right\}\right|^2} \\
&= \left\{\begin{array}{c} 0.14608621 \\ -0.35622668 \end{array}\right\}
\end{aligned}$$

De maneira análoga ao passo iterativo anterior, o próximo ponto determinando usando o passo λ_2 que mais minima a a função objetivo calculada em $\mathbf{x}_2 + \lambda_2 \mathbf{s}_2$

$$
\begin{aligned}
\frac{\partial f(\mathbf{x}_2 + \lambda_2 \mathbf{s}_2)}{\partial \lambda_2} &= 0 \\
\frac{\partial f\left(\left\{\begin{array}{c} 0.146086205560835\,\lambda_2 + 1.44087704619525 \\ 2.07790899844704 - 0.35622667838942\,\lambda_2 \end{array}\right\}\right)}{\partial \lambda_2} &= 0 \\
-0.128812109595067\,(-0.331353620746538\,\lambda_2 - 1)\,1.0 + 431.7705805\backslash & \\
82719\,(-0.405200151721146\,(0.101387003107994\,\lambda_2 + 1)\,1.0 - 0.34287\backslash & \\
0336146243)\,(-0.171435168073121\,\lambda_2 - 0.999142245307173\,(0.101387\backslash & \\
003107994\,\lambda_2 + 1)\,2.0 + 1)\,1.0 &= 0 \\
\lambda_2 &= 0.001226618461930137
\end{aligned}
$$

E com isso temos o ponto

$$
\begin{aligned}
\mathbf{x}_3 &= \mathbf{x}_2 + \lambda_2 \mathbf{s}_2 \\
&= \left\{\begin{array}{c} 1.44105624 \\ 2.07747204 \end{array}\right\}
\end{aligned}
$$

Onde a função objetivo vale

$$
f(\mathbf{x}_3) = 0.19459932316258796
$$

O resultado obtido difere do gabarito proposto devido à aproximações para os passos ótimos λ_i. O método foi implementado em `python` e com 15 passos convergiu para um solução com baiximo erro relativo

$$
\begin{aligned}
\mathbf{x}_{15}^T &= \left\{ \begin{array}{cc} 1.00000629 & 1.00001262 \end{array} \right\} \\
f(\mathbf{x}_{15}) &= 3.97806538923264 17e-11
\end{aligned}
$$

Uma solução que minimizou mais a função objetivo do que o método `CG` (Conjugate Gradient ou Fletcher Reeves do biblioteca `scipy`), que levou 25 passos até a seguinte solução

$$
\begin{aligned}
\mathbf{x}_{25}^T &= \left\{ \begin{array}{cc} 0.99944622 & 0.99881446 \end{array} \right\} \\
f(\mathbf{x}_{25}) &= 9.194084286321785e-07
\end{aligned}
$$

, conforme mostrado no `printscreen` da Figura 2.

```python
from scipy.optimize import minimize
x0=[-1.2,1.0]
def fun(x):
    x1 = x[0]
    x2 = x[1]
    return 100.0*(x2-x1**2.0)**2.0+(1.0-x1)**2.0

minimize(fun, x0, method='CG', options={'maxiter':25})
```

✓ 0.1s Python

```
     fun: 9.194084286321785e-07
     jac: array([ 0.0301919 , -0.01565393])
 message: 'Maximum number of iterations has been exceeded.'
    nfev: 166
     nit: 25
    njev: 55
  status: 1
 success: False
       x: array([0.99944622, 0.99881446])
```

Figura 2. Saída do método `CG` (Conjugate Gradient-Fletcher Reeves).

b) Método de Newton. O método de Newton leva em conta a informação das derivadas segundas da função objetivo, na forma de uma matrix Hessiana. A partir do ponto initial $\mathbf{x}_1$, cada próximo ponto é determinado por

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{J}_i^{-1}\nabla f_i$$

No caso do problema proposto, temos

$$\begin{aligned}
\nabla f_1 &= \left\{ \begin{array}{c} -400.0\,x_1\,(-x_1^2+x_2) - 2.0\,(1.0 - x_1) \\ 200.0\,(-x_1^2+x_2) \end{array} \right\} \\
&= \left\{ \begin{array}{c} -215.6 \\ -88 \end{array} \right\} \\
\mathbf{J}_1 &= \left[\begin{array}{cc} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 x_2} \\ \dfrac{\partial^2 f}{\partial x_2 x_1} & \dfrac{\partial^2 f}{\partial x_1^2} \end{array} \right] \\
&= \left[\begin{array}{cc} 800.0\,x_1^2 - 400.0\,(-x_1^2 + x2) + 2.0 & -400.0\,x_1 \\ -400.0\,x1 & 200.00 \end{array} \right] \\
&= \left[\begin{array}{cc} 1330 & 480 \\ 480 & 200 \end{array} \right] \\
\mathbf{J}_1^{-1} &= \left[\begin{array}{cc} 0.00561798 & -0.01348315 \\ -0.01348315 & 0.03735955 \end{array} \right]
\end{aligned}$$

E portanto o próximo ponto é

$$\begin{aligned}
\mathbf{x}_2 &= \mathbf{x}_1 - \mathbf{J}_1^{-1}\nabla f_1 \\
&= \left\{ \begin{array}{c} -1.2 \\ 1.0 \end{array} \right\} - \left[\begin{array}{cc} 0.00561798 & -0.01348315 \\ -0.01348315 & 0.03735955 \end{array} \right] \left\{ \begin{array}{c} -215.6 \\ -88 \end{array} \right\} \\
&= \left\{ \begin{array}{c} -1.1752809 \\ 1.38067416 \end{array} \right\}
\end{aligned}$$

Na iteração seguinte

$$\begin{aligned}
\nabla f_2 &= \left\{ \begin{array}{c} -400.0\,x_1\,(-x_1^2+x_2) - 2.0\,(1.0 - x_1) \\ 200.0\,(-x_1^2+x_2) \end{array} \right\} \\
&= \left\{ \begin{array}{c} -4.63781641 \\ -0.12220679 \end{array} \right\} \\
\mathbf{J}_2 &= \left[\begin{array}{cc} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 x_2} \\ \dfrac{\partial^2 f}{\partial x_2 x_1} & \dfrac{\partial^2 f}{\partial x_1^2} \end{array} \right] \\
&= \left[\begin{array}{cc} 800.0\,x_1^2 - 400.0\,(-x_1^2 + x2) + 2.0 & -400.0\,x_1 \\ -400.0\,x1 & 200.00 \end{array} \right] \\
&= \left[\begin{array}{cc} 1107.2725666 & 470.11235955 \\ 470.11235955 & 200 \end{array} \right] \\
\mathbf{J}_2^{-1} &= \left[\begin{array}{cc} 0.44555068 & -1.04729441 \\ -1.04729441 & 2.46673023 \end{array} \right]
\end{aligned}$$

$$\begin{aligned}
\mathbf{x}_3 &= \mathbf{x}_2 - \mathbf{J}_2^{-1}\nabla f_2 \\
&= \left\{ \begin{array}{c} -1.1752809 \\ 1.38067416 \end{array} \right\} - \left[\begin{array}{cc} 0.44555068 & -1.04729441 \\ -1.04729441 & 2.46673023 \end{array} \right] \left\{ \begin{array}{c} -4.63781641 \\ -0.12220679 \end{array} \right\} \\
&= \left\{ \begin{array}{c} 0.76311487 \\ -3.17503385 \end{array} \right\}
\end{aligned}$$

Finalmente, para a terceira iteração

$$
\begin{aligned}
\nabla f_3 &= \left\{ \begin{array}{c} -400.0\,x_1\,(-x_1^2+x_2)-2.0\,(1.0-x_1) \\ 200.0\,(-x_1^2+x_2) \end{array} \right\} \\
&= \left\{ \begin{array}{c} -4.63781641 \\ -0.12220679 \end{array} \right\}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{J}_3 &= \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 x_2} \\ \dfrac{\partial^2 f}{\partial x_2 x_1} & \dfrac{\partial^2 f}{\partial x_1^2} \end{bmatrix} \\
&= \begin{bmatrix} 800.0\,x_1^2-400.0\,(-x_1^2+x2)+2.0 & -400.0\,x_1 \\ -400.0\,x1 & 200.00 \end{bmatrix} \\
&= \begin{bmatrix} 1970.82670983 & -305.24594847 \\ -305.24594847 & 200 \end{bmatrix}
\end{aligned}
$$

$$
\mathbf{J}_3^{-1} = \begin{bmatrix} 0.00066447 & 0.00101414 \\ 0.00101414 & 0.00654781 \end{bmatrix}
$$

$$
\begin{aligned}
\mathbf{x}_4 &= \mathbf{x}_3 - \mathbf{J}_3^{-1} \nabla f_3 \\
&= \left\{ \begin{array}{c} 0.76311487 \\ -3.17503385 \end{array} \right\} - \begin{bmatrix} 0.00066447 & 0.00101414 \\ 0.00101414 & 0.00654781 \end{bmatrix} \left\{ \begin{array}{c} -4.63781641 \\ -0.12220679 \end{array} \right\} \\
&= \left\{ \begin{array}{c} 0.76342968 \\ 0.58282478 \end{array} \right\}
\end{aligned}
$$

, nesse ponto a função objetivo vale

$$
f(\mathbf{x}_4) = 0.055965516833937 81
$$

O método foi implementado em `python` e após 6 iterações temos econtra-se o ponto ótimo

$$
\begin{aligned}
\mathbf{x}_6 &= \left\{ \begin{array}{c} 0.9999957 \\ 0.99999139 \end{array} \right\} \\
f(\mathbf{x}_6) &= 1.8527397585949878e-11
\end{aligned}
$$

Para validação, a biblioteca `scipy` foi utilizada, e novamente o método implementado pelo `scipy` foi menos efetivo, levando 83 iterações para obter o seguinte ponto de ótimo

$$
\begin{aligned}
\mathbf{x}_{83} &= \left\{ \begin{array}{c} 0.9999957 \\ 0.99999139 \end{array} \right\} \\
f(\mathbf{x}_{83}) &= 3.3076113268503843e-07
\end{aligned}
$$

, conforme pode ser visto no `printscreen` da Figura 3.

```python
import numpy as np
def dfun(x):
    x1=x[0]
    x2=x[1]
    dx1=-400*(x2-x1**2)*x1-2*(1-x1)
    dx2=200*(x2-x1**2)
    return np.array([dx1,dx2])

minimize(fun, x0, method='Newton-CG', options={'maxiter':83}, jac=dfun)
```
0.9s Python

```
     fun: 3.3076113268503843e-07
     jac: array([ 0.07002874, -0.03553699])
 message: 'Warning: Maximum number of iterations has been exceeded.'
    nfev: 103
    nhev: 0
     nit: 83
    njev: 306
  status: 1
 success: False
       x: array([0.99942534, 0.99884871])
```

Figura 3. Saída do método `Newton-CG` da biblioteca `scipy`.