EN.553.761: Nonlinear Optimization I

Homework Assignment #1

Exercise 1.1: Compute $\nabla f(x)$ and $\nabla^2 f(x)$ for the following functions $f : \mathbb{R}^n \to \mathbb{R}$.

Answer: In Multivariate Calculus, we know that

$$df = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} dx_i = \left(\frac{\partial f}{\partial \mathbf{x}}\right)^T d\mathbf{x} \tag{1}$$

Thus, once we write the df in the form,

$$df = (\nabla f(x))^T \, d\mathbf{x} \tag{2}$$

we can obtain $\nabla f(x)$.

(a) $f(x) = \frac{1}{2} x^T H x$, where $H \in \mathbb{R}^{n \times n}$ is a fixed matrix. What if H is symmetric?

$$\begin{aligned} df &= \tfrac{1}{2}\left[(d\mathbf{x})^T Hx + x^T H d\mathbf{x}\right] \\ &= \tfrac{1}{2}\left[(Hx)^T d\mathbf{x} + x^T H d\mathbf{x}\right] \qquad (u^T v = v^T u \text{ if both } u \text{ and } v \text{ are vectors}) \\ &= \tfrac{1}{2}\left[(Hx)^T + x^T H\right] d\mathbf{x} \end{aligned} \tag{3}$$

According to (2), the $\nabla f(x)$ can be obtained by taking the transpose of $\frac{1}{2}\left[(Hx)^T + x^T H\right]$

$$\begin{aligned} \nabla f(x) &= \tfrac{1}{2}\left[(Hx)^T + x^T H\right]^T \\ &= \tfrac{1}{2}\left(Hx + H^T x\right) \\ &= \tfrac{1}{2}\left(H + H^T\right) x \end{aligned} \tag{4}$$

Thus, $\nabla f(x) = \frac{1}{2}\left(H + H^T\right) x = g(x)$, which will be used to compute $\nabla^2 f(x)$ in the following:

$$\begin{aligned} g(x) &= \nabla f(x) \\ \nabla^2 f(x) &= \nabla g(x) \end{aligned}$$

$$\begin{aligned} dg &= \tfrac{1}{2}\left(H + H^T\right) d\mathbf{x} \\ \nabla g(x) &= \left[\tfrac{1}{2}\left(H + H^T\right)\right]^T \\ &= \tfrac{1}{2}\left(H^T + H\right) \end{aligned}$$

Thus, $\nabla^2 f(x) = \nabla g(x) = \frac{1}{2}\left(H^T + H\right)$.

If H is symmetric, then $H^T = H$, then $\nabla f(x) = Hx$ and $\nabla^2 f(x) = H$.

(b) $f(x) = b^T Ax - \frac{1}{2}x^T A^T Ax$, where $A \in \mathbb{R}^{m\times n}$ is a fixed matrix and $b \in \mathbb{R}^m$ is a fixed vector.

According to the distributive rule of differentiation, $\nabla f(x)$ can be computed by following equation:

$$\nabla f(x) = \nabla \left(b^T Ax\right) - \nabla \left(\tfrac{1}{2}x^T A^T Ax\right) \tag{5}$$

$\nabla \left(b^T Ax\right)$ can be computed by following:

$$\begin{aligned} d(b^T Ax) &= \left(b^T A\right) d\mathbf{x} \\ \nabla \left(b^T Ax\right) &= \left(b^T A\right)^T \\ &= A^T b \end{aligned} \tag{6}$$

For the reason that $A^T A \in \mathbb{R}^{n\times n}$, $\nabla \left(\frac{1}{2}x^T A^T Ax\right)$ can be computed based on Question 1 by the following:

$$\begin{aligned} \nabla \left(\tfrac{1}{2}x^T A^T Ax\right) &= \tfrac{1}{2}\left[(A^T A) + (A^T A)^T\right] x \\ &= A^T Ax \end{aligned} \tag{7}$$

From (6) and (7), we can compute $\nabla f(x)$:

$$\begin{aligned} \nabla f(x) &= \nabla \left(b^T Ax\right) - \nabla \left(\tfrac{1}{2}x^T A^T Ax\right) \\ &= A^T b - A^T Ax \end{aligned} \tag{8}$$

Thus, $\nabla f(x) = A^T b - A^T Ax = g(x)$, which will be used to compute $\nabla^2 f(x)$ in the following:

$$\begin{aligned} g(x) &= \nabla f(x) \\ \nabla^2 f(x) &= \nabla g(x) \end{aligned}$$

$$\begin{aligned} dg &= -A^T A d\mathbf{x} \\ \nabla g(x) &= -\left(A^T A\right)^T \\ &= -A^T A \end{aligned} \tag{9}$$

Because $A^T A$ is symmetric. Thus, $\nabla^2 f(x) = \nabla g(x) = -A^T A$.

(c) $f(x) = \|x\|_2 = \left(\sum_{i=1}^n x_i^2\right)^{1/2}$

$df(x)$ can be simplified by the following equation:

$$df(x) = \tfrac{1}{2}\|\mathbf{x}\|_2^{-1}\, d\|\mathbf{x}\|_2^2 \tag{10}$$

$d\|\mathbf{x}\|_2^2$ is computed by the following:

$$\begin{aligned} d\|\mathbf{x}\|_2^2 &= d\left(\mathbf{x}^T\mathbf{x}\right) \\ &= (d\mathbf{x})^T\,\mathbf{x} + \mathbf{x}^T d\mathbf{x} \\ &= \mathbf{x}^T d\mathbf{x} + \mathbf{x}^T d\mathbf{x} \qquad ((d\mathbf{x})^T \in \mathbb{R}^n) \\ &= 2\mathbf{x}^T d\mathbf{x} \end{aligned} \tag{11}$$

Thus,

$$\begin{aligned} df(x) &= \tfrac{1}{2}\|\mathbf{x}\|_2^{-1}\, 2\mathbf{x}^T d\mathbf{x} \\ &= \|\mathbf{x}\|_2^{-1}\, \mathbf{x}^T d\mathbf{x} \end{aligned} \tag{12}$$

According to (2), the $\nabla f(x)$ can be obtained by taking the transpose of $\|\mathbf{x}\|_2^{-1}\mathbf{x}^T$

$$\begin{aligned} \nabla f(x) &= \left(\|\mathbf{x}\|_2^{-1}\, \mathbf{x}^T\right)^T \\ &= \|\mathbf{x}\|_2^{-1}\, \mathbf{x} \end{aligned} \tag{13}$$

Thus, $\nabla f(x) = \|\mathbf{x}\|_2^{-1}\, \mathbf{x} = g(x)$, which will be used to compute $\nabla^2 f(x)$ in the following:

$$\begin{aligned} g(x) &= \nabla f(x) \\ \nabla^2 f(x) &= \nabla g(x) \end{aligned}$$

$$\begin{aligned} dg &= \mathbf{x} d\|\mathbf{x}\|_2^{-1} + \|\mathbf{x}\|_2^{-1} d\mathbf{x} \\ &= \mathbf{x}\left(-\|\mathbf{x}\|_2^{-2}\right) d\|\mathbf{x}\|_2 + \|\mathbf{x}\|_2^{-1} d\mathbf{x} \\ &= -\|\mathbf{x}\|_2^{-2}\, \mathbf{x}\, \|\mathbf{x}\|_2^{-1}\, \mathbf{x}^T d\mathbf{x} + \|\mathbf{x}\|_2^{-1} d\mathbf{x} \\ dg &= -\|\mathbf{x}\|_2^{-3}\, \mathbf{x}\,\mathbf{x}^T d\mathbf{x} + \|\mathbf{x}\|_2^{-1} d\mathbf{x} \\ &= \left(-\|\mathbf{x}\|_2^{-3}\, \mathbf{x}\,\mathbf{x}^T + \|\mathbf{x}\|_2^{-1}\mathbf{I}\right) d\mathbf{x} \\ \nabla g(x) &= \left(-\|\mathbf{x}\|_2^{-3}\, \mathbf{x}\,\mathbf{x}^T + \|\mathbf{x}\|_2^{-1}\mathbf{I}\right)^T \\ &= -\|\mathbf{x}\|_2^{-3}\, \mathbf{x}\,\mathbf{x}^T + \|\mathbf{x}\|_2^{-1}\mathbf{I} \end{aligned}$$

Thus, $\nabla^2 f(x) = \nabla g(x) = -\|\mathbf{x}\|_2^{-3}\, \mathbf{x}\,\mathbf{x}^T + \|\mathbf{x}\|_2^{-1}\mathbf{I}$.

(d) $f(x) = \|Ax - b\|_2$, where $A \in \mathbb{R}^{m\times n}$ is a fixed matrix and $b \in \mathbb{R}^m$ is a fixed vector.

$df(x)$ can be simplified by the following equation:

$$df(x) = \tfrac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^{-1}\, d\|A\mathbf{x} - \mathbf{b}\|_2^2 \tag{14}$$

$d\|A\mathbf{x} - \mathbf{b}\|_2^2$ is computed by the following:

$$\begin{aligned} d\|A\mathbf{x} - \mathbf{b}\|_2^2 &= d\left[\left(A\mathbf{x} - \mathbf{b}\right)^T\left(A\mathbf{x} - \mathbf{b}\right)\right] \\ &= d\left(\mathbf{x}^T A^T A\mathbf{x} - \mathbf{x}^T A^T\mathbf{b} - \mathbf{b}^T A\mathbf{x} + \mathbf{b}^T\mathbf{b}\right) \\ &= (2\mathbf{x}^T A^T A - 2\mathbf{b}^T A) d\mathbf{x} \qquad \text{(From previous result in (6) and (7))} \end{aligned} \tag{15}$$

Thus,

$$\begin{aligned} df(x) &= \tfrac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^{-1}\, (2\mathbf{x}^T A^T A - 2\mathbf{b}^T A) d\mathbf{x} \\ &= \|A\mathbf{x} - \mathbf{b}\|_2^{-1}\, (\mathbf{x}^T A^T A - \mathbf{b}^T A) d\mathbf{x} \end{aligned} \tag{16}$$

According to (2), the $\nabla f(x)$ can be obtained by taking the transpose of $\|A\mathbf{x}-\mathbf{b}\|_2^{-1}\,(\mathbf{x}^T A^T A - \mathbf{b}^T A)$

$$\begin{aligned} \nabla f(x) &= \left(\|A\mathbf{x} - \mathbf{b}\|_2^{-1}\, (\mathbf{x}^T A^T A - \mathbf{b}^T A)\right)^T \\ &= \|A\mathbf{x} - \mathbf{b}\|_2^{-1}\, (A^T A\mathbf{x} - A^T\mathbf{b}) \end{aligned} \tag{17}$$

Thus, $\nabla f(x) = \|A\mathbf{x} - \mathbf{b}\|_2^{-1} (A^T A\mathbf{x} - A^T\mathbf{b}) = g(x)$, which will be used to compute $\nabla^2 f(x)$ in the following:

$$g(x) = \nabla f(x)$$
$$\nabla^2 f(x) = \nabla g(x)$$

$$\begin{aligned}
dg &= (A^T A\mathbf{x} - A^T\mathbf{b})\, d\|A\mathbf{x} - \mathbf{b}\|_2^{-1} + \|A\mathbf{x} - \mathbf{b}\|_2^{-1}\, d(A^T A\mathbf{x} - A^T\mathbf{b}) \\
&= (A^T A\mathbf{x} - A^T\mathbf{b})\, (-\|A\mathbf{x} - \mathbf{b}\|_2^{-2})\, d\|A\mathbf{x} - \mathbf{b}\|_2 + \|A\mathbf{x} - \mathbf{b}\|_2^{-1} A^T A\, d\mathbf{x} \\
&= -\|A\mathbf{x} - \mathbf{b}\|_2^{-2}\, (A^T A\mathbf{x} - A^T\mathbf{b})\, \|A\mathbf{x} - \mathbf{b}\|_2^{-1}\, (\mathbf{x}^T A^T A - \mathbf{b}^T A) d\mathbf{x} + \|A\mathbf{x} - \mathbf{b}\|_2^{-1} A^T A\, d\mathbf{x} \\
dg &= -\|A\mathbf{x} - \mathbf{b}\|_2^{-3}\, (A^T A\mathbf{x} - A^T\mathbf{b})\, (\mathbf{x}^T A^T A - \mathbf{b}^T A) d\mathbf{x} + \|A\mathbf{x} - \mathbf{b}\|_2^{-1} A^T A\, d\mathbf{x} \\
&= [-\|A\mathbf{x} - \mathbf{b}\|_2^{-3}\, (A^T A\mathbf{x}\mathbf{x}^T A^T A - 2A^T\mathbf{b}\mathbf{x}^T A^T A + A^T\mathbf{b}\mathbf{b}^T A) + \|A\mathbf{x} - \mathbf{b}\|_2^{-1} A^T A]\, d\mathbf{x} \\
\nabla g(x) &= \left[-\|A\mathbf{x} - \mathbf{b}\|_2^{-3}\, (A^T A\mathbf{x}\mathbf{x}^T A^T A - 2A^T\mathbf{b}\mathbf{x}^T A^T A + A^T\mathbf{b}\mathbf{b}^T A) + \|A\mathbf{x} - \mathbf{b}\|_2^{-1} A^T A\right]^T \\
&= -\|A\mathbf{x} - \mathbf{b}\|_2^{-3}\, (A^T A\mathbf{x}\mathbf{x}^T A^T A - 2A^T A\mathbf{x}\mathbf{b}^T A + A^T\mathbf{b}\mathbf{b}^T A) + \|A\mathbf{x} - \mathbf{b}\|_2^{-1} A^T A
\end{aligned}$$

Thus,

$$\begin{aligned}
\nabla^2 f(x) = \nabla g(x) &= -\|A\mathbf{x} - \mathbf{b}\|_2^{-3}\, (A^T A\mathbf{x}\mathbf{x}^T A^T A - 2A^T A\mathbf{x}\mathbf{b}^T A + A^T\mathbf{b}\mathbf{b}^T A) + \|A\mathbf{x} - \mathbf{b}\|_2^{-1} A^T A \\
&= -\|A\mathbf{x} - \mathbf{b}\|_2^{-3}\, A^T (Ax - \mathbf{b})(Ax - \mathbf{b})^T A + \|A\mathbf{x} - \mathbf{b}\|_2^{-1} A^T A
\end{aligned}$$

Exercise 1.2: Let $\mathcal{S} \subseteq \mathbb{R}^n$, $f : \mathcal{S} \to \mathbb{R}$. Let $x \in \mathcal{S}$ and $s \in \mathbb{R}^n$ be such that $[x, x+s] \in \mathcal{S}$.

(a) By defining $\phi(\alpha) = f(x + \alpha s)$ and using the Fundamental Theorem of Calculus:

$$\phi(1) = \phi(0) + \int_0^1 \phi'(\alpha) d\alpha,$$

show that

$$|f(x+s) - f(x) - g(x)^T s| \le \tfrac{1}{2}\gamma^L \|s\|_2^2$$

whenever f has a Lipschitz continuous gradient with Lipschitz constant γ^L on $\mathcal{S}$.

Answer:
With Fix s and $\phi(\alpha) = f(x + \alpha s)$, $f(x+s)$ can be rewrote as following form:

$$\begin{aligned}
f(x+s) = \phi(1) &= \phi(0) + \int_0^1 \phi'(\alpha) d\alpha \\
&= \phi(0) + \phi'(0) + \int_0^1 \phi'(t) - \phi'(0)\, dt \qquad (18) \\
\phi(1) - \phi(0) - \phi'(0) &= \int_0^1 \phi'(t) - \phi'(0)\, dt
\end{aligned}$$

Then by taking the absolute value of both size, the equation can be rewrote as following:

$$|\phi(1) - \phi(0) - \phi'(0)| = |\int_0^1 \phi'(t) - \phi'(0)\, dt| \leqslant \int_0^1 |\phi'(t) - \phi'(0)|\, dt \qquad (19)$$

$$|\phi(1) - \phi(0) - \phi'(0)| \leqslant \int_0^1 |\phi'(t) - \phi'(0)|\, dt = \int_0^1 |\nabla f(x+ts)^T s - \nabla f(x)^T s|\, dt \qquad (20)$$

With Cauchy-Schwartz inequality and Lipschitz continuous property:

$$\int_0^1 |\nabla f(x+ts)^T s - \nabla f(x)^T s|\, dt \leqslant \int_0^1 \|s\|_2 \, \|\nabla f(x+ts)^T - \nabla f(x)^T\|_2 \, dt \leqslant \int_0^1 \|s\|_2 \gamma^L \|x+ts-x\|_2 \, dt \tag{21}$$

Thus, combining (20) and (21), the equation can be rewrote as following:

$$|\phi(1) - \phi(0) - \phi'(0)| \leqslant \gamma^L \|s\|_2^2 \int_0^1 t\, dt = \tfrac{1}{2}\gamma^L \|s\|_2^2 \tag{22}$$

$$|f(x+s) - f(x) - g(x)^T s| \leqslant \tfrac{1}{2}\gamma^L \|s\|_2^2 \tag{23}$$

(b) Justify the formula

$$\phi(1) = \phi(0) + \phi'(0) + \int_0^1 \int_0^\alpha \phi''(t)\; dt\; d\alpha.$$

Hence, show that

$$|f(x+s) - f(x) - g(x)^T s - \tfrac{1}{2} s^T H(x) s| \leq \frac{1}{6}\gamma^Q \|s\|_2^3,$$

whenever f has a Lipschitz continuous Hessian with Lipschitz constant γ^Q on $\mathcal{S}$.

Answer:

First, let's justify the formula.

$$RHS = \phi(0) + \phi'(0) + \int_0^1 \int_0^\alpha \phi''(t) dt d\alpha = \phi(0) + \phi'(0) + \int_0^1 \left[\phi'(t)|_0^\alpha \right] d\alpha \tag{24}$$

$$= \phi(0) + \phi'(0) + \int_0^1 \left(\phi'(\alpha) - \phi'(0) \right] d\alpha = \phi(0) + \phi'(0) + \phi(\alpha)|_0^1 - \phi'(0) \tag{25}$$

$$= \phi(0) + \phi'(0) + \phi(1) - \phi(0) - \phi'(0) = \phi(1) = LHS \tag{26}$$

Thus, this formula is justified.
With Fix s and $\phi(\alpha) = f(x + \alpha s)$, $f(x+s)$ can be rewrote as following form:

$$\begin{aligned} f(x+s) = \phi(1) = \phi(0) + \phi'(0) + \tfrac{1}{2}\phi''(0) + \int_0^1 \int_0^\alpha \phi''(t) - \phi''(0)\; dt\; d\alpha \\ \phi(1) - \phi(0) - \phi'(0) - \tfrac{1}{2}\phi''(0) = \int_0^1 \int_0^\alpha \phi''(t) - \phi''(0)\; dt\; d\alpha \end{aligned} \tag{27}$$

Because

$$\int_0^1 \int_0^\alpha -\phi''(0)\; dt\; d\alpha = -\tfrac{1}{2}\phi''(0)$$

Then by taking the absolute value of both size, the equation can be rewrote as following:

$$|\phi(1) - \phi(0) - \phi'(0) - \tfrac{1}{2}\phi''(0)| = |\int_0^1 \int_0^\alpha \phi''(t) - \phi''(0)\; dt\; d\alpha| \leqslant \int_0^1 \int_0^\alpha |\phi''(t) - \phi''(0)|\; dt\; d\alpha \tag{28}$$

$$|\phi(1) - \phi(0) - \phi'(0) - \tfrac{1}{2}\phi''(0)| \leqslant \int_0^1 \int_0^\alpha |\phi''(t) - \phi''(0)|\; dt\; d\alpha = \int_0^1 \int_0^\alpha |s^T H(x+ts)s - s^T H(x) s|\; dt\; d\alpha \tag{29}$$

With Cauchy-Schwartz inequality and Lipschitz continuous property:

$$\int_0^1 \int_0^\alpha |s^T H(x+ts)s - s^T H(x)s| \; dt \; d\alpha \leqslant \int_0^1 \int_0^\alpha \|s\|_2^2 \|H(x+ts) - H(x)\|_2 \; dt \; d\alpha \leqslant \int_0^1 \int_0^\alpha \|s\|_2^2 \gamma^Q \|x+ts-x\|_2 \; dt \; d\alpha \tag{30}$$

Thus, combining (29) and (30), the equation can be rewrote as following:

$$|\phi(1) - \phi(0) - \phi'(0) - \tfrac{1}{2}\phi''(0)| \leqslant \gamma^Q \|s\|_2^3 \int_0^1 \int_0^\alpha t \, dt = \gamma^Q \|s\|_2^3 \int_0^1 \tfrac{1}{2}\alpha^2 d\alpha = \frac{1}{6}\gamma^Q \|s\|_2^3 \tag{31}$$

$$|f(x+s) - f(x) - g(x)^T s - \tfrac{1}{2} s^T H(x) s| \leqslant \frac{1}{6}\gamma^Q \|s\|_2^3 \tag{32}$$

This is the output for Ex 1.3

This file calls “my_func” and “newton”
Functions, which are both attached below.

```
tol      = 1.0e-14  ; % VERY tight stopping tolerance
maxit    = 50       ; % maximum number of iterations allowed
x0       = [-10; 10 ]  ; % initial guess at a solution
printlevel = 1;

[x,F,J,iter,status] = newton('my_func',x0,maxit,printlevel,tol);

sprintf('The final x is')
disp(x)
sprintf('The final Function value is')
disp(F)
sprintf('The final Jacobian matrix is')
disp(J)
```

```
 iter       |F|            |J|
    1   3.005102e+02   1.326500e+02
    2   8.879244e+01   5.848417e+01
    3   2.608146e+01   2.577306e+01
    4   8.833433e+00   1.223421e+01
    5   2.734851e+00   5.770677e+00
    6   1.018182e+00   2.561776e+00
    7   3.670292e-01   1.210689e+00
    8   8.555974e-02   1.560064e+00
    9   5.522496e-03   1.585249e+00
   10   2.743997e-05   1.594009e+00
   11   7.882539e-10   1.594037e+00
   12   0.000000e+00   1.594037e+00

ans =

    'The final x is'

   -0.2366
   -0.5132


ans =

    'The final Function value is'

     0
     0


ans =

    'The final Jacobian matrix is'

    0.1679   -1.0000
   -1.0000   -1.0265
```

my_func Function

```
% referenced from matlab demo2
% http://www.ams.jhu.edu/~abasu9/AMS_553-761/demos_lecture03_661.m
function [F,J]=my_func(X)
    x1 = X(1);
    x2 = X(2);

    F = [x1^3 - x2 - 0.5;
         -x1 + x2^2 - 0.5];
    J = [3*x1^2, -1;
         -1, 2*x2];
end
```

```
Not enough input arguments.

Error in my_func (line 4)
    x1 = X(1);
```

Published with MATLAB® R2018b

Newton Function

```
% referenced from matlab demo2
% http://www.ams.jhu.edu/~abasu9/AMS_553-761/demos_lecture03_661.m

function [x,F,J,iter,status] = newton(Fun,x0,maxit,printlevel,tol)
% cast Fun from string to a Function
func = str2func(Fun);
%
iter     = 0        ;
[F0,J0]  = func(x0);
F        = F0       ;
J        = J0       ;
x        = x0       ;
if printlevel == 1
    flag = 1;

while norm(F) > tol*norm(F0) && iter < maxit

   iter   = iter + 1 ;
   s      = J\(-F) ;
   x      = x + s    ;
   [F,J]  = func(x)  ;

   if printlevel == 1
       if flag == 1
           fprintf('\n iter       |F|            |J| \n');
           flag = 0;
       end
       fprintf(' %4g  %13.6e  %13.6e \n', iter, norm(F), norm(J))
   end
end

if iter == maxit
    status = 1;
else
    status = 0;
end


end
```

Not enough input arguments.

Error in newton (line 6)
func = str2func(Fun);

Published with MATLAB® R2018b

EX 1.4 (a)

We can regard the λ as the $n+1$th variable (like x_{n+1}) in the iteration using Newton's method. So the $n+1$ nonlinear equations should be like below:

$$\begin{bmatrix} a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n - \lambda x_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n - \lambda x_2 \\ \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \ldots + a_{n,n}x_n - \lambda x_n \\ x_1^2 + x_2^2 + \ldots + x_n^2 - 1 \end{bmatrix} = 0$$

Then,

$$F = \begin{bmatrix} a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n - \lambda x_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n - \lambda x_2 \\ \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \ldots + a_{n,n}x_n - \lambda x_n \\ x_1^2 + x_2^2 + \ldots + x_n^2 - 1 \end{bmatrix}$$

$$J = \begin{bmatrix} a_{1,1} - \lambda & a_{1,2} & a_{1,3} & \ldots & a_{1,n} & -x_1 \\ a_{2,1} & a_{2,2} - \lambda & a_{2,3} & \ldots & a_{2,n} & -x_2 \\ & & & \vdots & & \\ a_{n,1} & a_{n,2} & a_{n,3} & \ldots & a_{n,n} - \lambda & -x_n \\ 2x_1 & 2x_2 & 2x_3 & \ldots & 2x_n & 0 \end{bmatrix}$$

Now we could solve the Newton's method equation:

$$x_{k+1} = x - J^{-1}F$$

Where $x_{k+1} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ \lambda \end{bmatrix}$

This is the output for Ex 1.4b

This file calls “Fun” and “newton” Functions,
which are both attached below.

```
global A
A = [4,2,1 ; 2,3,0 ; 1,0,1];
x0 = [1/5;-1/5;4/5];
lambda0 = 1;
x0 = [x0;lambda0];
tol      = 1.0e-14  ; % VERY tight stopping tolerance
maxit    = 50       ; % maximum number of iterations allowed
printlevel = 1;

[x,F,J,iter,status] = newton('Fun',x0,maxit,printlevel,tol);

sprintf('The final x is')
disp(x(1:3))
sprintf('The final lambda is')
disp(x(4))
sprintf('The final Function value is')
disp(F)
sprintf('The final Jacobian matrix is')
disp(J)
```

```
 iter        |F|            |J|
    1    3.789240e+03   1.193503e+02
    2    9.471708e+02   5.969302e+01
    3    2.366601e+02   2.988274e+01
    4    5.906209e+01   1.501784e+01
    5    1.691411e+01   8.181787e+00
    6    2.239788e+01   9.467221e+00
    7    1.171360e+01   7.114551e+00
    8    3.012441e+00   4.006215e+00
    9    5.654165e-01   3.874784e+00
   10    5.105603e-02   3.874622e+00
   11    6.200240e-04   3.874588e+00
   12    9.604808e-08   3.874587e+00
   13    2.455074e-15   3.874587e+00

ans =

    'The final x is'

    0.4318
   -0.7331
    0.5255


ans =

    'The final lambda is'

    1.8218
```

```
ans =

    'The final Function value is'

   1.0e-14 *

   -0.0111
   -0.0222
         0
    0.2442


ans =

    'The final Jacobian matrix is'

    2.1782    2.0000    1.0000   -0.4318
    2.0000    1.1782         0    0.7331
    1.0000         0   -0.8218   -0.5255
    0.8637   -1.4661    1.0510         0
```

Published with MATLAB® R2018b

Fun Function

```
function [F,J] = Fun(X)
    global A
    x = X(1:3);
    lambda = X(4);
    F = [A*x-lambda*x; sum(x.^2)-1];
    A_size = size(A);
    J = [A-lambda*eye(A_size(1)),-x; 2*x',0];

end
```

```
Not enough input arguments.

Error in Fun (line 3)
    x = X(1:3);
```

Published with MATLAB® R2018b

Newton Function

```
% referenced from matlab demo2
% http://www.ams.jhu.edu/~abasu9/AMS_553-761/demos_lecture03_661.m

function [x,F,J,iter,status] = newton(Fun,x0,maxit,printlevel,tol)
% cast Fun from string to a Function
func = str2func(Fun);
%
iter      = 0        ;
[F0,J0]   = func(x0);
F         = F0       ;
J         = J0       ;
x         = x0       ;
if printlevel == 1
    flag = 1;

while norm(F) > tol*norm(F0) && iter < maxit

   iter    = iter + 1 ;
   s       = J\(-F) ;
   x       = x + s    ;
   [F,J]   = func(x)  ;

   if printlevel == 1
       if flag == 1
           fprintf('\n iter       |F|            |J| \n');
           flag = 0;
       end
       fprintf(' %4g  %13.6e  %13.6e \n', iter, norm(F), norm(J))
   end
end

if iter == maxit
    status = 1;
else
    status = 0;
end


end
```

```
Not enough input arguments.

Error in newton (line 6)
func = str2func(Fun);
```

Published with MATLAB® R2018b