ASE-4046 Exercise 4 Solutions

Problem 1

Part a

Enter the data to be fitted:

```
x=[210.2   197     191.4   185.7   183.2   181.15]'; % column vector
v=[28.559  21.892  19.758  17.267  16.385  15.919]';
```

Create a function to compute the residuals $r_i = a_1 e^{a_2 x_i} - v_i$:

```
r=@(a) a(1)*exp(a(2)*x)-v;
```

Create a function to compute the Jacobian matrix; its two columns are $\frac{\partial r_i}{\partial a_1} = e^{a_2 x_i}$ and $\frac{\partial r_i}{\partial a_2} = x_i a_1 e^{a_2 x_i}$:

```
J=@(a) [ exp(a(2)*x) a(1)*x.*exp(a(2)*x) ];
```

In exercise 3.1 we made a least-squares fit to the log-transformed data:

$$\log(v_i) \approx \log(a_1 e^{a_2 x_i}) = \log(a_1) + a_2 x_i = b_2 + b_1 x_i$$

```
b=[x ones(size(x))]\log(v);
a0=[exp(b(2)) b(1)]'
```

```
a0 =

    0.3942
    0.0204
```

The Gauss-Newton correction is

```
da=-J(a0)\r(a0);
```

and the first iterand for the coefficients of the NLLS fit is

```
a=a0+da
```

```
a =

    0.4004
    0.0203
```

The next iterand is

```
a=a-J(a)\r(a)
```

```
a =

    0.4005
```

```
    0.0203
```

Part b

The solution with `lsqnonlin` is

```
a=lsqnonlin(r,a0)
```

```
Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>
a =

    0.4005
    0.0203
```

Part c

In exercise 3.1 the fitting problem was based on log-transformed data:

$$\log(v_i) \approx \log(a_1 e^{a_2 x_i}) = \log(a_1) + a_2 x_i$$

The minimisation problem's cost function was the sum of squared residuals:

$$\sum_i \left(\log(a_1) + a_2 x_i - \log(v_i)\right)^2$$

Problem 2

The residual is

$$r_i = y_i - \|x_{1:3} - s_i\| - x_4$$

where $x_{1:3} = u$ and $x_4 = v$. It is computed by the `trilat` function, see the end of this live script.

Enter the problem data: the landmark (i.e. satellite) locations s, pseudo-ranges y, and "initial guess" location u:

```
s=[ 7766188.44   -21960535.34   12522838.56
  -25922679.66    -6629461.28      31864.37
   -5743774.02   -25828319.92    1692757.72
   -2786005.69   -15900725.8    21302003.49 ];
y=[22228206.42;   24096139.11;  21729070.63; 21259581.09];
u0=[2.7950e+06; 1.2361e+06; 5.5797e+06];
```

Solve for the receiver location:

```
x0=[u0; 0];               % initial guess has clock error = 0
f=@(x) trilat(x,y,s);     % cost function
x=lsqnonlin(f,x0);        % solve for receiver location and clock error
```

```
Local minimum found.
```

```
Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>
```

```
u=x(1:3)                 % receiver location
```

```
u =

   1.0e+06
   -2.4307
   -4.7023
    3.5466
```

Display more digits in the floating point numbers:

```
num2str(u,9)
```

```
ans =
 -2430745.1
-4702345.11
 3546568.71
```

Alternative solution using the Jacobian formula. The residual and its Jacobian are computed using the `trilatg` function included at the end of this live script.

Solve for the receiver location:

```
fg=@(x) trilatg(x,y,s);
options=optimoptions('lsqnonlin','Jacobian','on','checkgradients',true);
xg=lsqnonlin(fg,x0,[ ],[ ],options);
```

```
____________________________________________________________
   CheckGradients Information
Objective function derivatives:
Maximum relative difference between supplied
and finite-difference derivatives = 1.13701e-07.

CheckGradients successfully passed.
____________________________________________________________
Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>
```

```
u=xg(1:3); num2str(u,9)
```

```
ans =
 -2430745.1
-4702345.11
 3546568.71
```

Local functions

The residual function for Problem 2 is obtained by modifying the `trilat` function from slide 12. A third-dimension term is added to the distance function h, and a clock error term is added to the residual `r`.

```
function r=trilat(x,y,s)
n=length(y);
r=zeros(n,1);
for i=1:n
    h=sqrt((x(1)-s(i,1)).^2+(x(2)-s(i,2)).^2+(x(3)-s(i,3)).^2); % modified
    r(i)=y(i)-h-x(4);   % modified
end
end % end of function trilat
```

A second version of the residual function is obtained by modifying the `trilatg` function from slide 14. The last column of the Jacobian is $\partial r_i / \partial x_4 = -1$.

```
function [r,J]=trilatg(x,y,s)
n=length(y);
m=length(x);
J=zeros(n,m);
r=zeros(n,1);
for i=1:n
    h=sqrt((x(1)-s(i,1)).^2+(x(2)-s(i,2)).^2+(x(3)-s(i,3)).^2); % modified
    J(i,:)=[-(x(1:3)'-s(i,:))/h -1];   % modified
    r(i)=y(i)-h-x(4);   % modified
end
end % end of function trilatg
```