Numerical Method: Labwork 2 Report

Nguyễn Gia Phong–BI9-184

Fall 2019

III Polynomial

1.c. At the time of writing, function `fzero` in Octave have not support the `Display` option just yet*. However, the implementation of this option is rather trivial, thus I made a quick patch (which is also attached at the bug report). Using this, one can easily display all the iterations as followed:

```
octave:1> fzero (@(x) x.^2 - 9, 0, optimset ('display', 'iter'))

Search for an interval around 0 containing a sign change:
Func-eval 1,  how = initial,  a = 0,  f(a) = -9,  b = 0,  f(b) = -9
Func-eval 2,  how = search,  a = 0,  f(a) = -9,  b = 0.099,  f(b) = -8.9902
Func-eval 3,  how = search,  a = 0,  f(a) = -9,  b = 0.1025,  f(b) = -8.98949
Func-eval 4,  how = search,  a = 0,  f(a) = -9,  b = 0.095,  f(b) = -8.99098
Func-eval 5,  how = search,  a = 0,  f(a) = -9,  b = 0.11,  f(b) = -8.9879
Func-eval 6,  how = search,  a = 0,  f(a) = -9,  b = 0.075,  f(b) = -8.99437
Func-eval 7,  how = search,  a = 0,  f(a) = -9,  b = 0.15,  f(b) = -8.9775
Func-eval 8,  how = search,  a = 0,  f(a) = -9,  b = 0,  f(b) = -9
Func-eval 9,  how = search,  a = 0,  f(a) = -9,  b = 0.35,  f(b) = -8.8775
Func-eval 10,  how = search,  a = 0,  f(a) = -9,  b = -0.4,  f(b) = -8.84
Func-eval 11,  how = search,  a = 0,  f(a) = -9,  b = 1.1,  f(b) = -7.79
Func-eval 12,  how = search,  a = 0,  f(a) = -9,  b = -4.9,  f(b) = 15.01

Search for a a zero in the interval [-4.9, 0]:
Func-eval 13,  how = initial,  x = 0,  f(x) = -9
Func-eval 14,  how = interpolation,  x = -1.83673,  f(x) = -5.62641  (NaN%)
Func-eval 15,  how = interpolation,  x = -3.36837,  f(x) = 2.3459  (141.7%)
Func-eval 16,  how = interpolation,  x = -3.19097,  f(x) = 1.1823  (-49.6%)
```

*Bug report: `https://savannah.gnu.org/bugs/?56954`

```
Func-eval 17,  how = interpolation,  x = -2.99725,  f(x) = -0.0164972  (-101.4%)
Func-eval 18,  how = interpolation,  x = -3.00258,  f(x) = 0.0154927  (193.9%)
Func-eval 19,  how = interpolation,  x = -3,  f(x) = 3.07975e-07  (-100.0%)
Func-eval 20,  how = interpolation,  x = -3,  f(x) = -7.10543e-15  (-100.0%)
Func-eval 21,  how = interpolation,  x = -3,  f(x) = 5.32907e-15  (169.7%)

Algorithm converged

ans = -3.0000
```

To answer the question in part b, (since I believe these parts are linked to each other), the current implementation of `fzero` search for the second bracket over quantitative chages below if `X0` if it is a single scalar, thus $[-4.9, 0]$ is gotten and the found solution is negative:

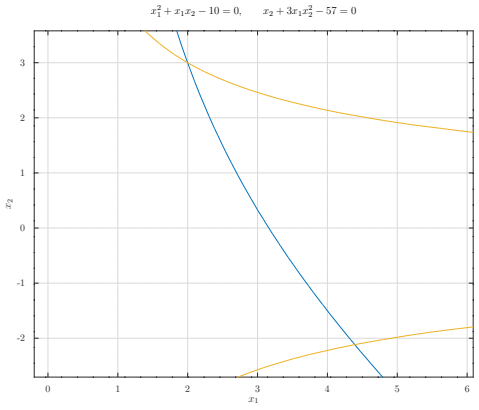
```
[-.01 +.025 -.05 +.10 -.25 +.50 -1 +2.5 -5 +10 -50 +100 -500 +1000]
```

IV Non-linear Systems

1.a. These statements were used to plot the given functions:

```
ezplot(@(x1, x2) x1 .^ 2 + x1 .* x2 - 10)
hold on
ezplot(@(x1, x2) x2 + 3 .* x1 .* x2 .^ 2 - 57)
```

As shown in the graphs (where $x_1^2 + x_1 x_2 = 10$ are the blue lines and $x_2 + 3x_1 x_2 = 57$ are the yellow ones), the solutions of (x_1, x_2) are quite close to $(2, 3)$ and $(4.5, -2)$.



I would also like to note that I am personally impressed how gnuplot (which is utilised by Octave) is able to export to TikZ graphics with ease.