Math.375
I - A beastiary of errors

Vageli Coutsias

Introduction

– 1 + 1 = 0 or “machine epsilon”?

– » eps = 2.220446049250313e-016

How does matlab produce its numbers?

- Where we learn about number formats, truncation errors and roundoff

Matlab real number formats

format long %(default for π)
pi = 3.14159265358979
format short
» pi = 3.1416
» format short e
» pi = 3.1416e+000
» format long e
» pi =3.141592653589793e+000

Floating-point numbers

$$x = \pm\left(\frac{d_1}{b} + \frac{d_2}{b^2} + \frac{d_3}{b^3} + \cdots + \frac{d_t}{b^t}\right) b^e$$

$$0 \le d_i \le b - 1, i = 1, \ldots, p; d_1 \ne 0$$

Base or radix
t Precision
[L,U] Exponent range

$$L \le e \le U$$

SYSTEM	Base	Precision	L(ow Exp)	U(pperExp)
IEEE SP	2	24	-126	127
IEEE DP	2	53	-1022	1023
Cray	2	48	-16383	16384
HP Calc	10	12	-499	499
IBM mainfr	16	6	-64	63

Floating point numbers

$$Underflow\_level := UFL = b^{L}$$

$$Overflow\_level := OFL = b^{U+1}(1 - b^{-t})$$

$$e := eps = \frac{1}{2} b^{1-t}$$

The machine precision is the smallest nuber ε such that:

$$fl(1+e) > 1$$

$IEEE\_sp\_e = 2^{-24} \approx 10^{-7}$ $IEEE\_dp\_e = 2^{-53} \approx 10^{-16}$

- Overflow does not only cause programs to crash!

Arianne V’s short maiden flight on 7/4/96 was due to a floating exception.

FLOAT → INTEGER

- During the conversion of a 64-bit floating-point number to a 16-bit signed integer
- Caused by the float being outside the range representable by such integers
- The programming philosophy employed did not guard against software errors-a fatal assumption!

Types of errors in numerical computation

- Roundoff errors
  Pi = 3.14159
  Pi = 3.1415926535897932384626
- Truncation errors
  Cos x = 1 – x^2/2
  Cos x = 1 – x^2/2 + x^4/4!

Errors usually accumulate randomly (random walk)

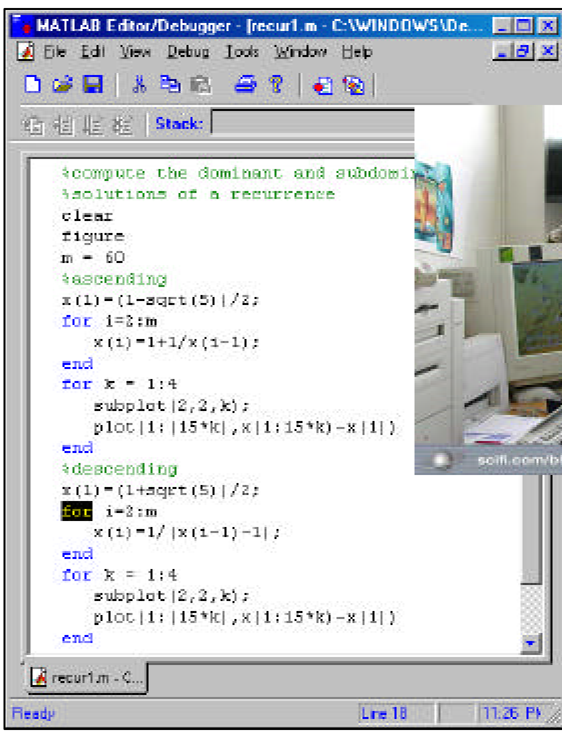
But they can also be systematic, and the reasons may be subtle!

Recurrence relations

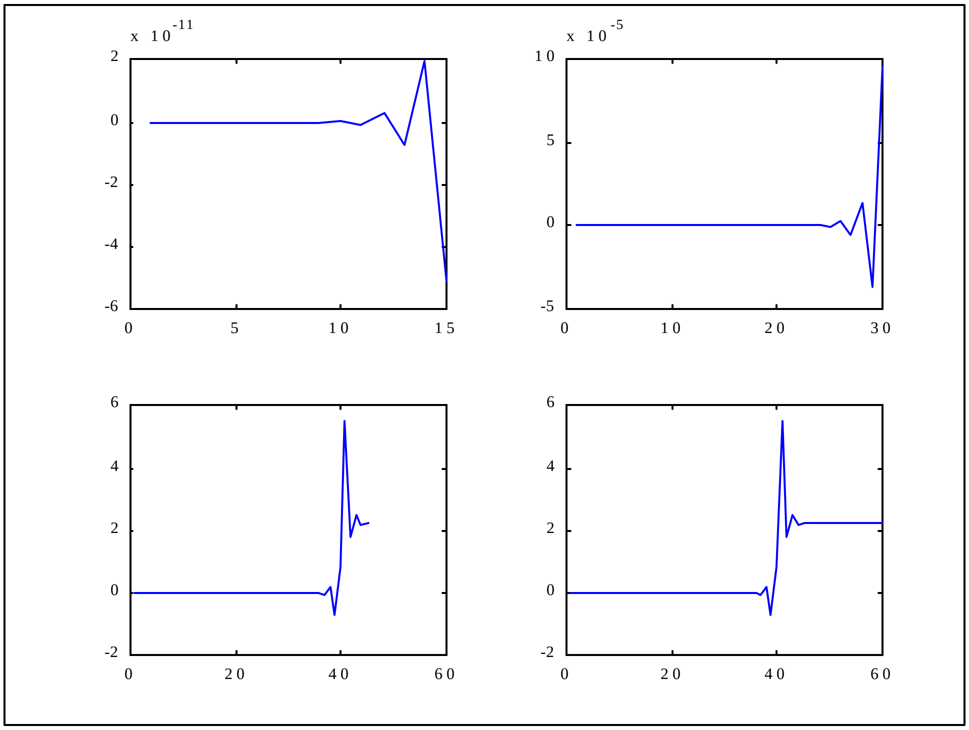
$$x^2 - x - 1 = 0 \Leftrightarrow x = 1 + \frac{1}{x} \qquad x_\pm = \frac{1 \pm \sqrt{5}}{2}$$

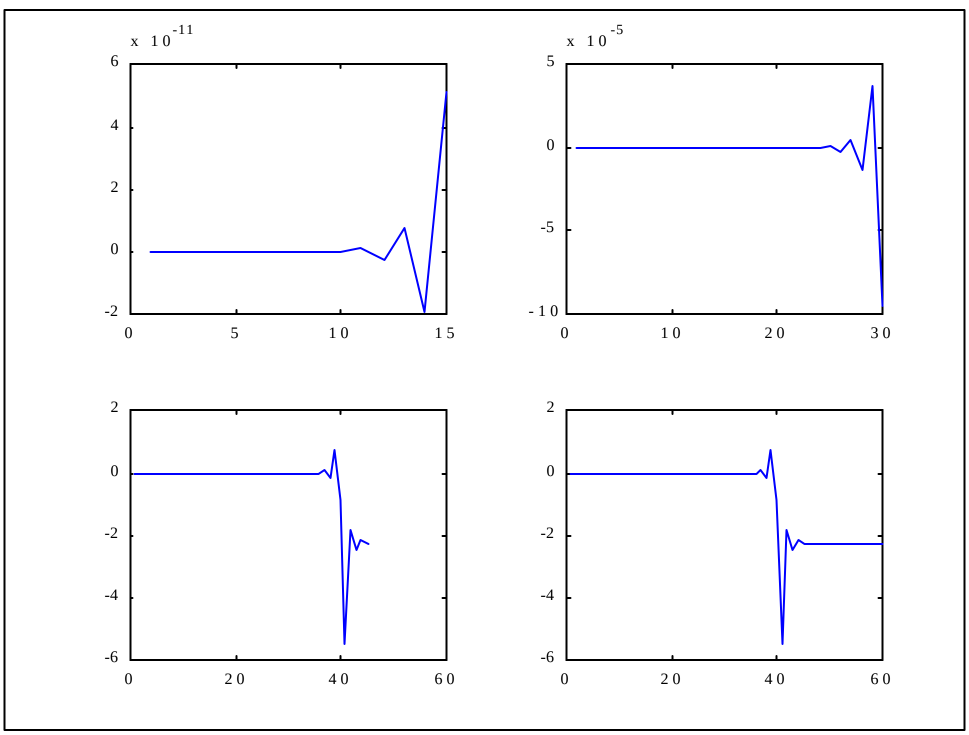
$$x_{n+1} = 1 + \frac{1}{x_n} \qquad x_n^+ \to x_+$$

$$x_{n+1} = \frac{1}{x_n - 1} \qquad x_n^- \to x_-$$



Instability is inescapable;
But one can learn to ride it!


x 10^-11
2
0
-2
-4
-6
0
5
10
15
x 10^-5
10
5
0
-5
0
10
20
30
6
4
2
0
-2
0
20
40
60
6
4
2
0
-2
0
20
40
60


x 10^-11
6
4
2
0
-2
0
5
10
15
x 10^-5
5
0
-5
-10
0
10
20
30
2
0
-2
-4
-6
0
20
40
60
2
0
-2
-4
-6
0
20
40
60

Due to limitations in computer arithmetic, need to practice "defensive coding".

- Mathematics → numerics + implementation
- The sequence of computations carried out for the solution of a mathematical problem can be so complex as to require a mathematical analysis of its own correctness.
- How can we determine efficiency?

T

Vocabulary

- Consistency – correctness - …
- Convergence (rate)
- Efficiency (operation counts)
- Numerical Stability (error amplification)
- Accuracy (relative vs. absolute error)
- Roundoff vs. Truncation

The first monster:

T(x) = e^x * log(1+e^-1)

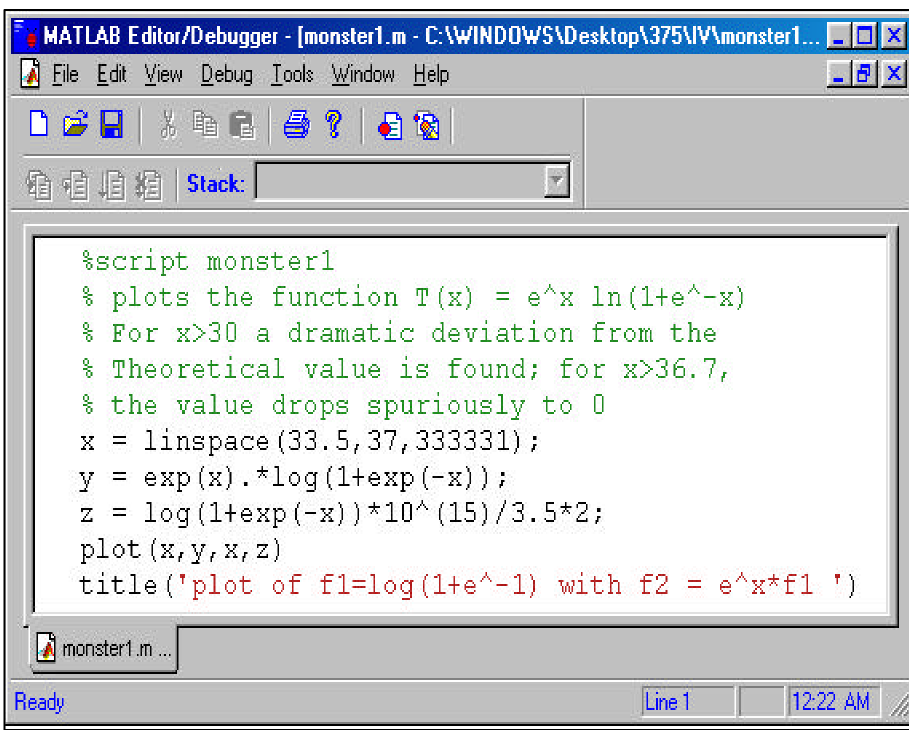
For x>30 a dramatic deviation from the theoretical value 1 is found.

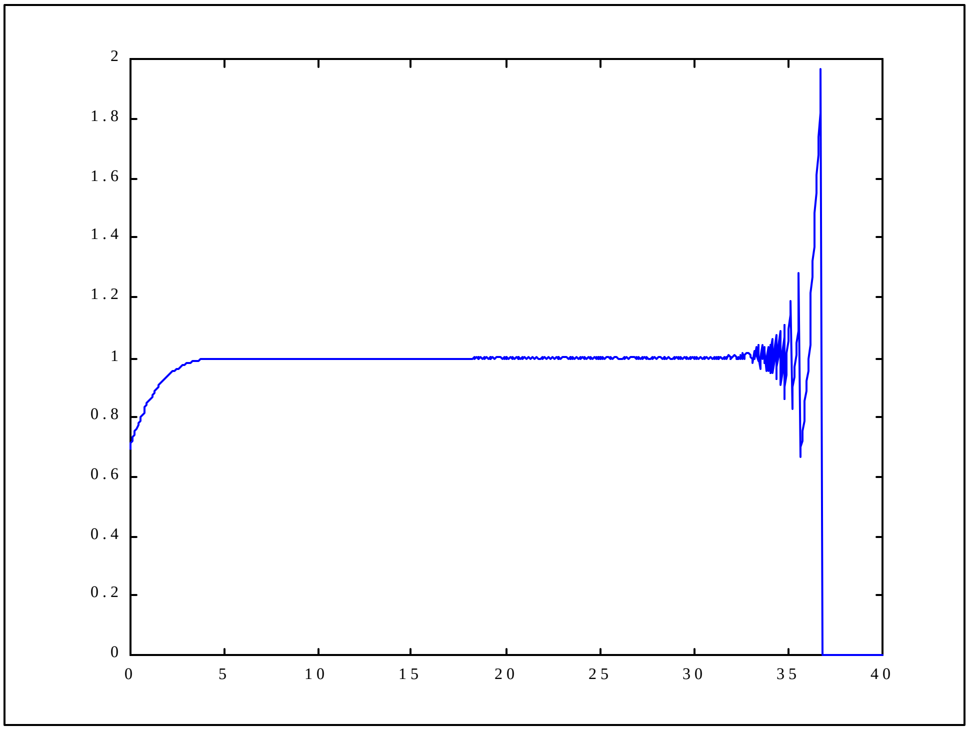
For x>36.7 the plotted value drops (incorrectly) to zero

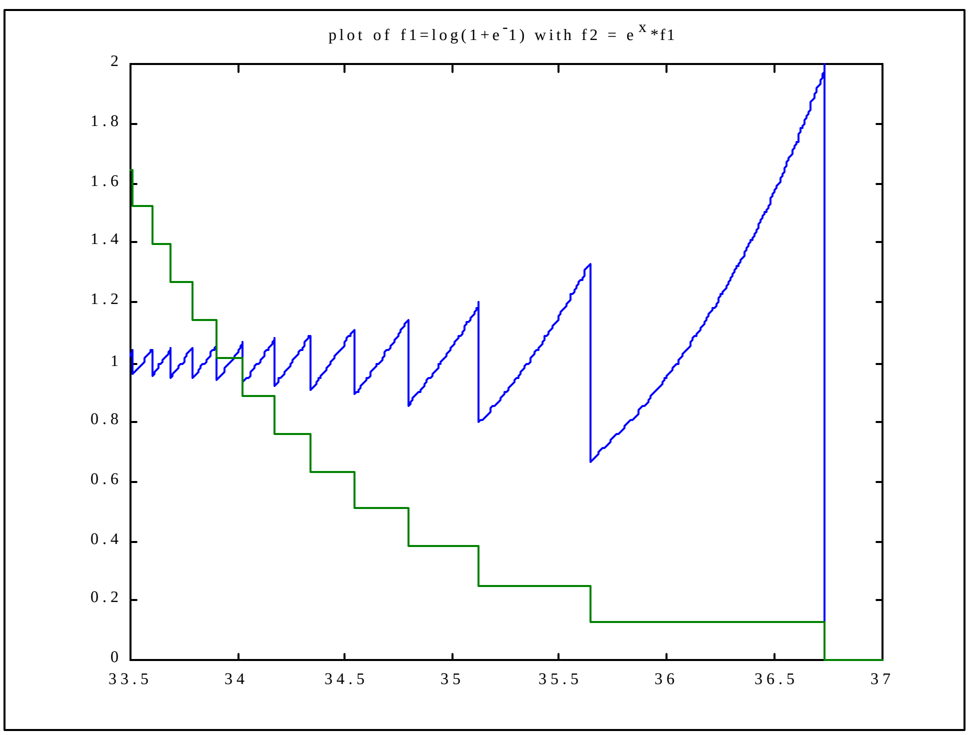
Here:

x = linspace(0,40,1000);

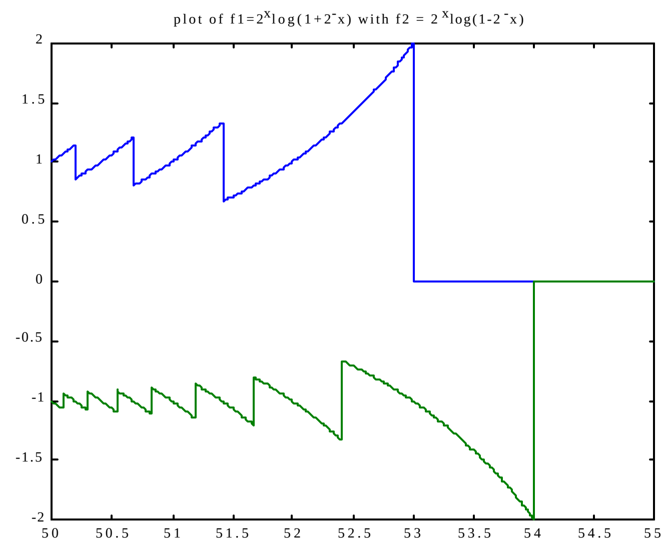
y = exp(x).*log(1+exp(-x));

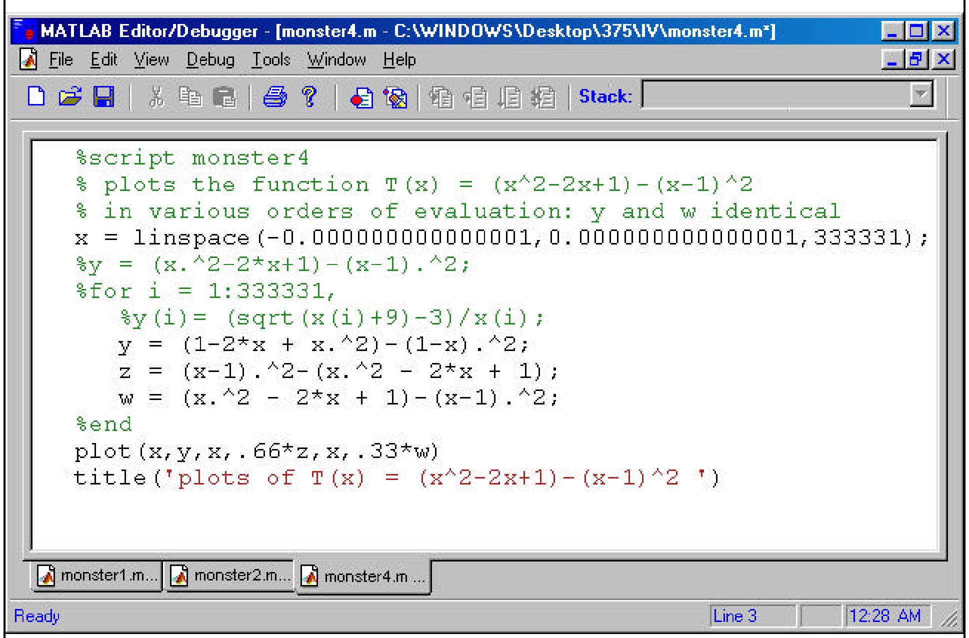



2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
5
10
15
20
25
30
35
40

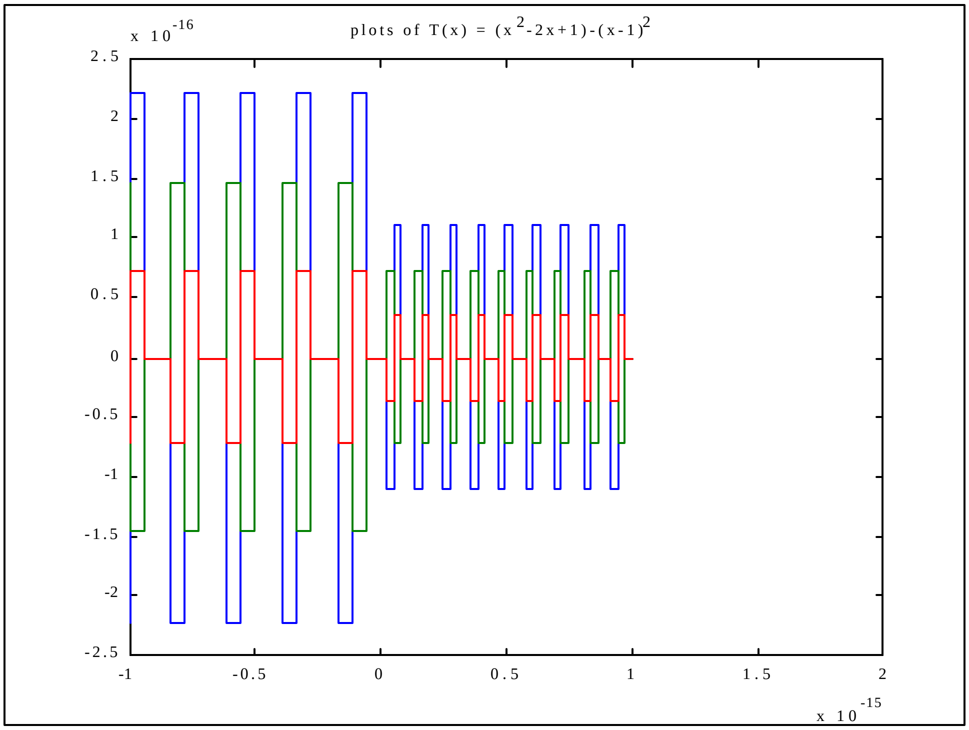

plot of f1=log(1+e-1) with f2 = e^x*f1
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
33.5
34
34.5
35
35.5
36
36.5
37

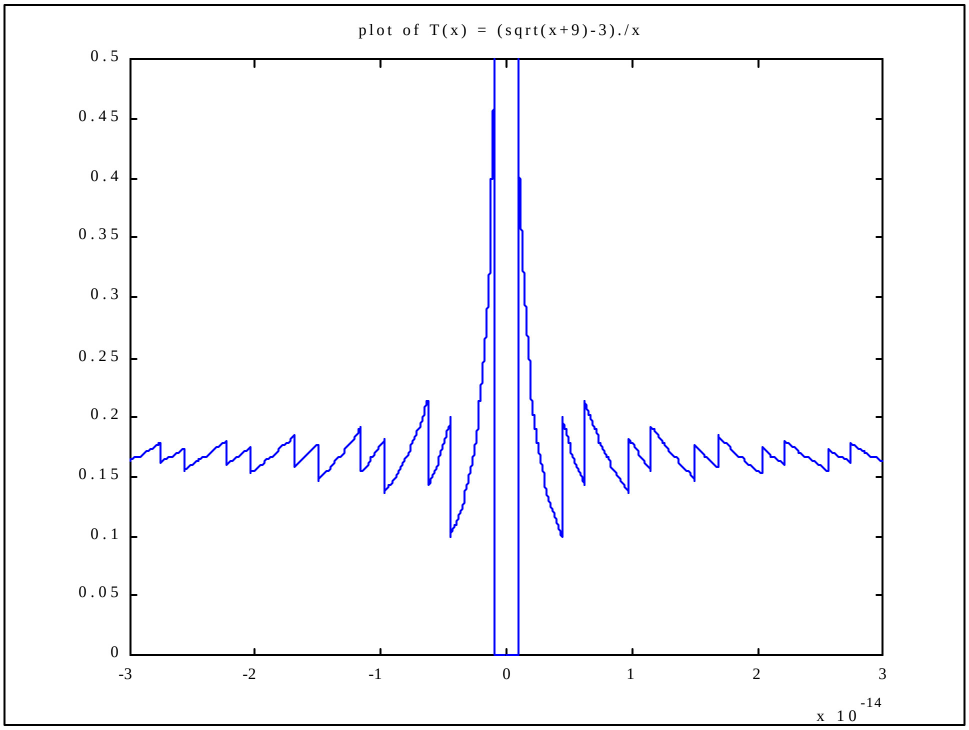
Pet Monster I $T(x) = 2^x \ln(1+2^{-x})$






```
%script monster4
% plots the function T(x) = (x^2-2x+1)-(x-1)^2
% in various orders of evaluation: y and w identical
x = linspace(-0.000000000000001,0.000000000000001,333331);
%y = (x.^2-2*x+1)-(x-1).^2;
%for i = 1:333331,
   %y(i)= (sqrt(x(i)+9)-3)/x(i);
   y = (1-2*x + x.^2)-(1-x).^2;
   z = (x-1).^2-(x.^2 - 2*x + 1);
   w = (x.^2 - 2*x + 1)-(x-1).^2;
%end
plot(x,y,x,.66*z,x,.33*w)
title('plots of T(x) = (x^2-2x+1)-(x-1)^2 ')
```


plots of T(x) = (x^2-2x+1)-(x-1)^2
x 10^-16
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-1
-0.5
0
0.5
1
1.5
2
x 10^-15


plot of T(x) = (sqrt(x+9)-3)./x
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-3
-2
-1
0
1
2
3
x 10^-14

Summary

- Roundoff and other errors
- Multiple plots
- Formats and floating point numbers

References

- C. Essex, M.Davidson, C. Schulzky, "Numerical Monsters", preprint.
- Higham & Higham, Matlab Guide, SIAM
- SIAM News, 29(8), 10/98 (Arianne V failure)
- B5 Trailer; http://www.scifi.com/b5rangers/

- As usual, find m-files at

aix: ~coutsias/375/IV