Math.375 Fall 2005
4. Errors in Numerical Computation

Vageli Coutsias

Types of errors in numerical computation

- Roundoff errors

  Pi = 3.14159

  Pi = 3.14159265358979323846 26

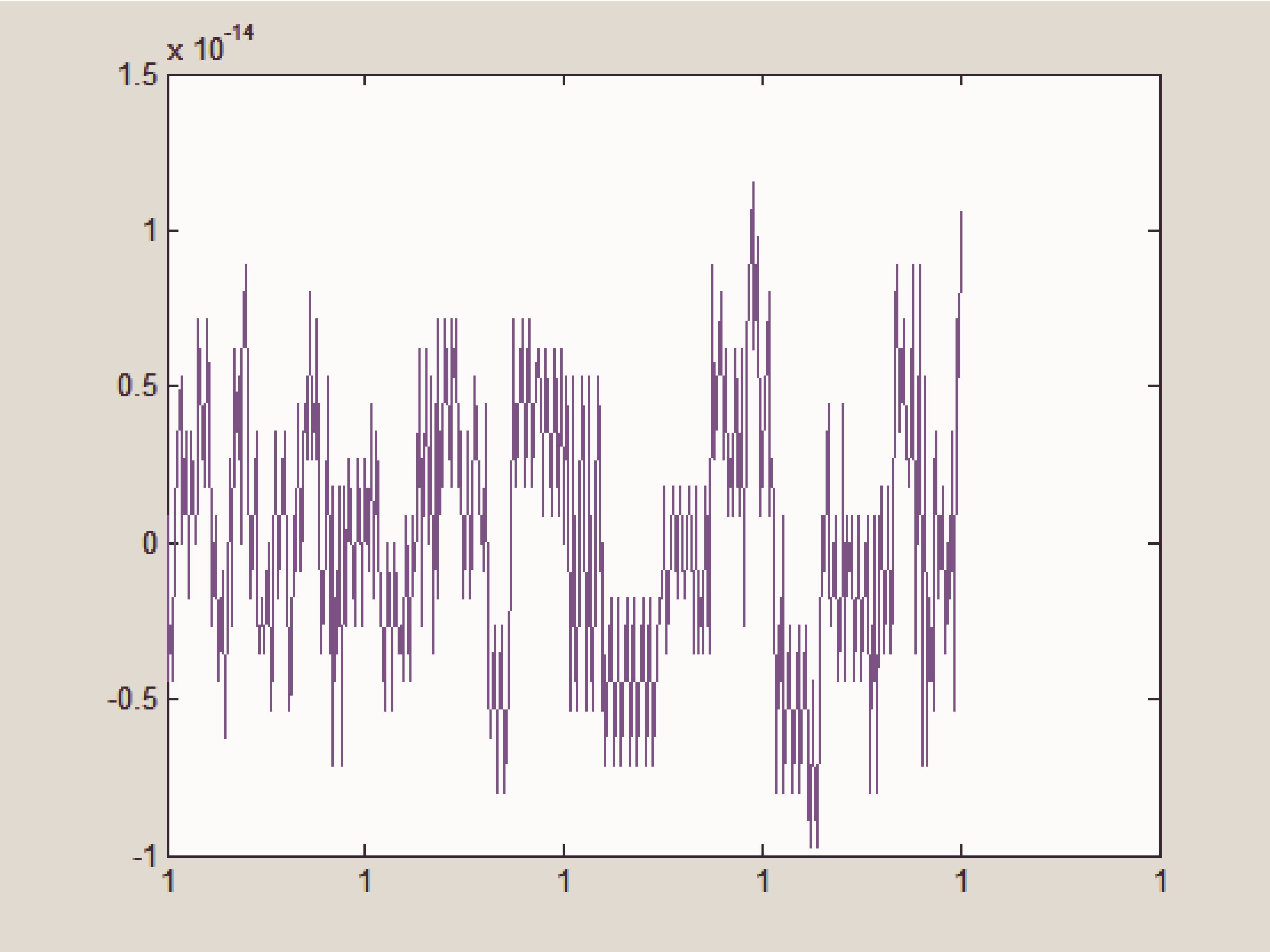
- Truncation errors

  $\cos x = 1 - x^2/2$

  $\cos x = 1 - x^2/2 + x^4/4!$

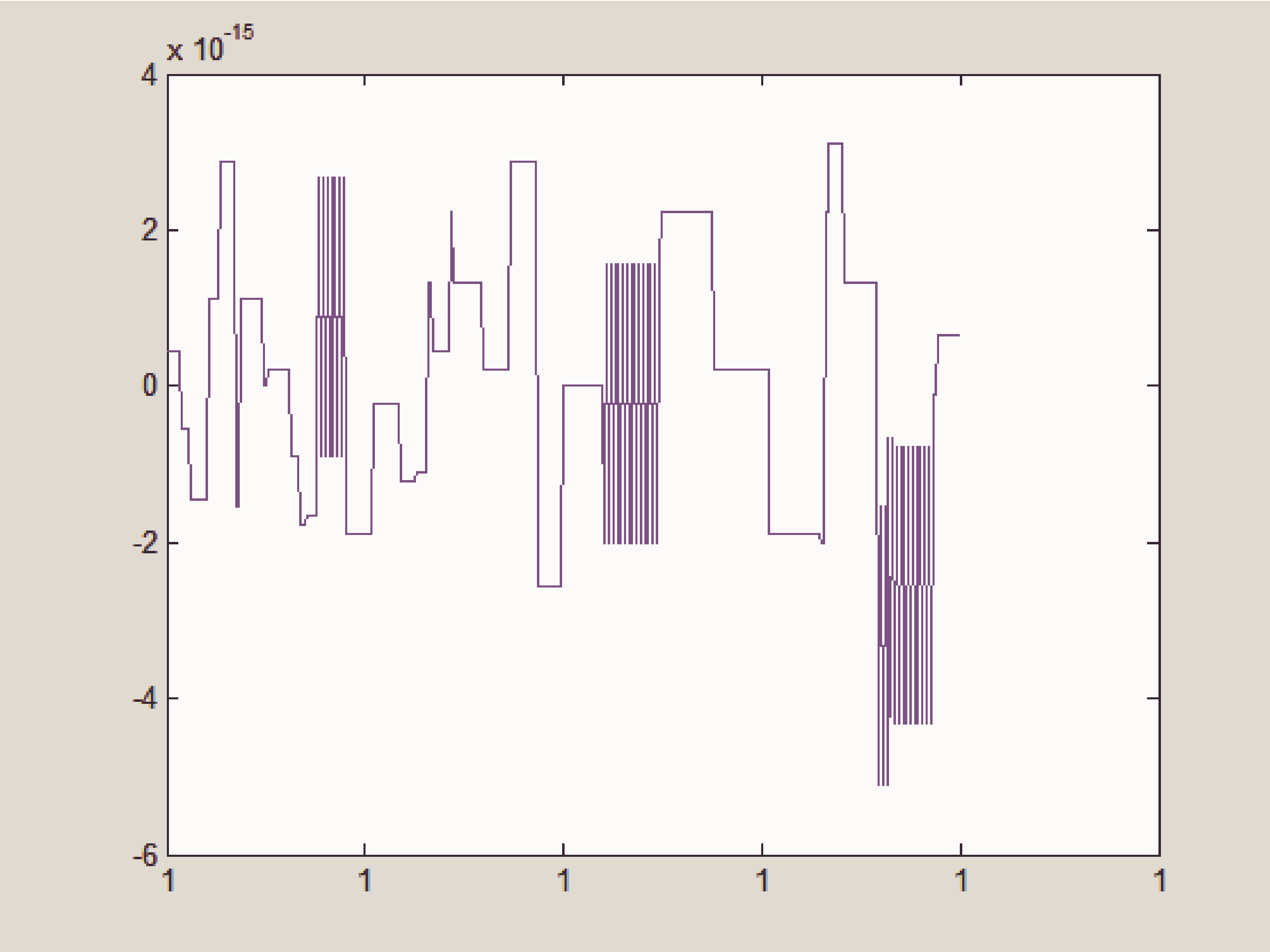
Errors usually accumulate randomly (random walk)

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

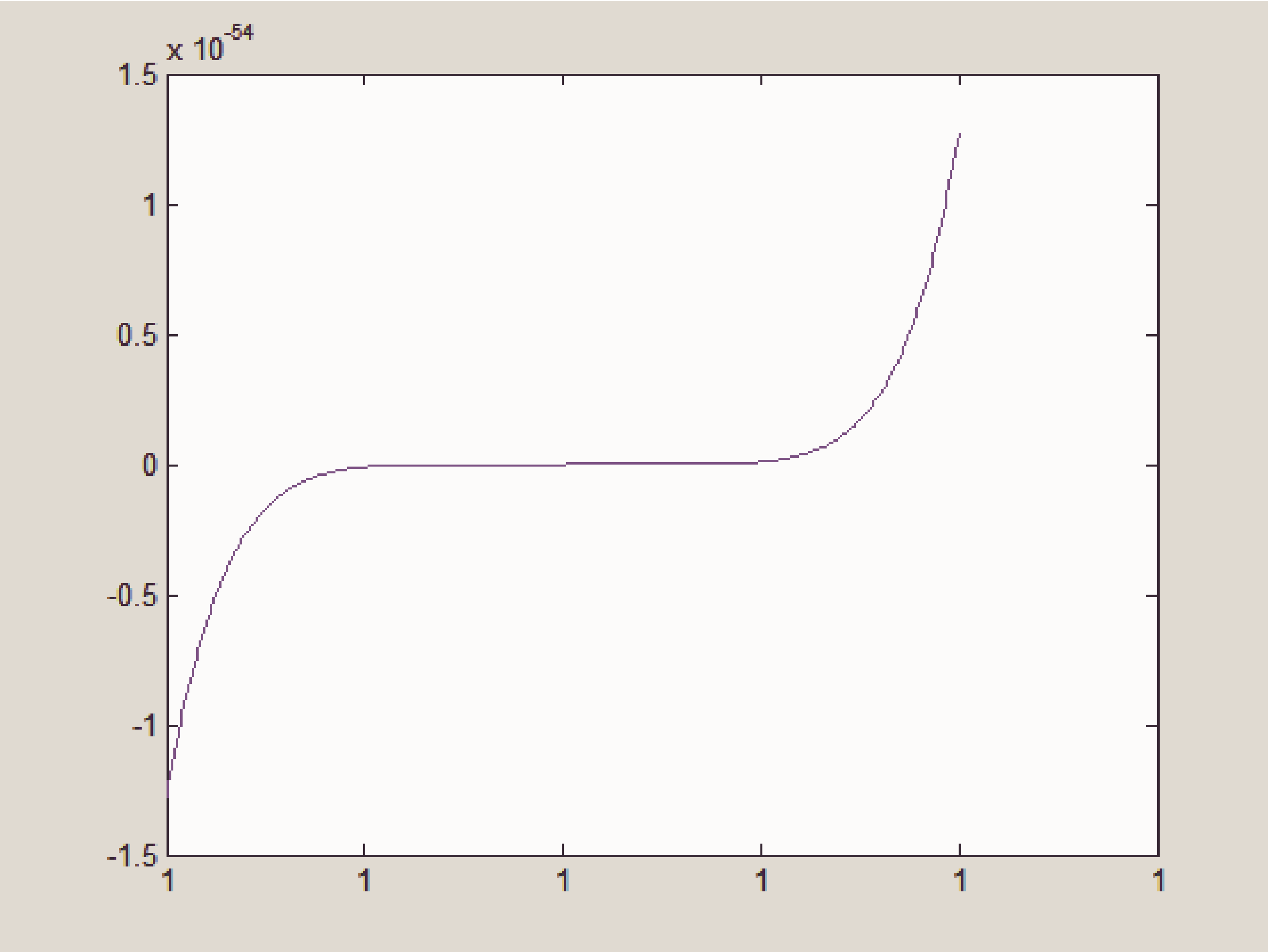



```
x = linspace(1-2*10^-8,1+2*10^-8,401);
f = x.^7-7*x.^6+21*x.^5-35*x.^4+35*x.^3-21*x.^2+7*x-1;
plot(x,f)
```

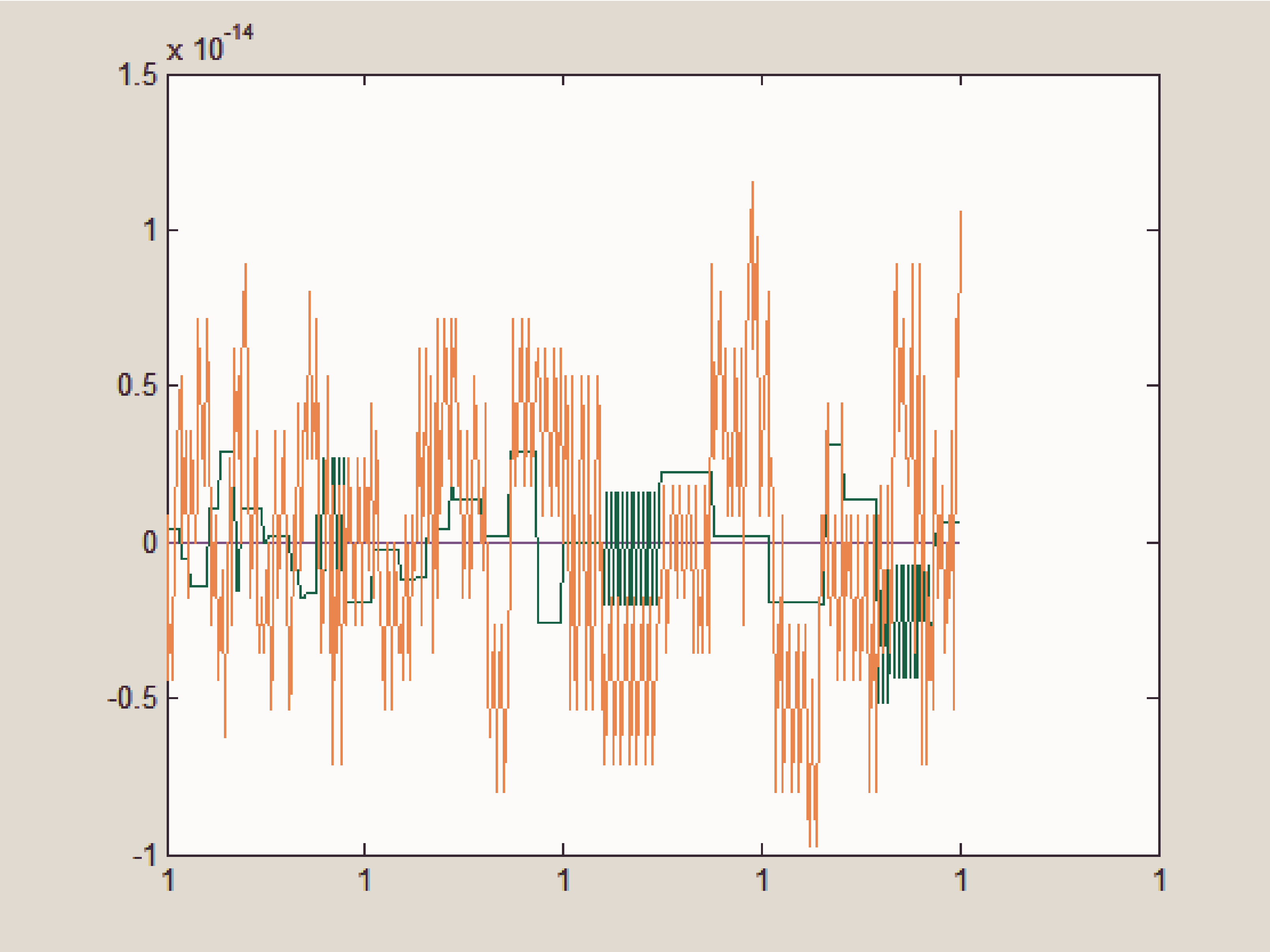
CANCELLATION ERRORS



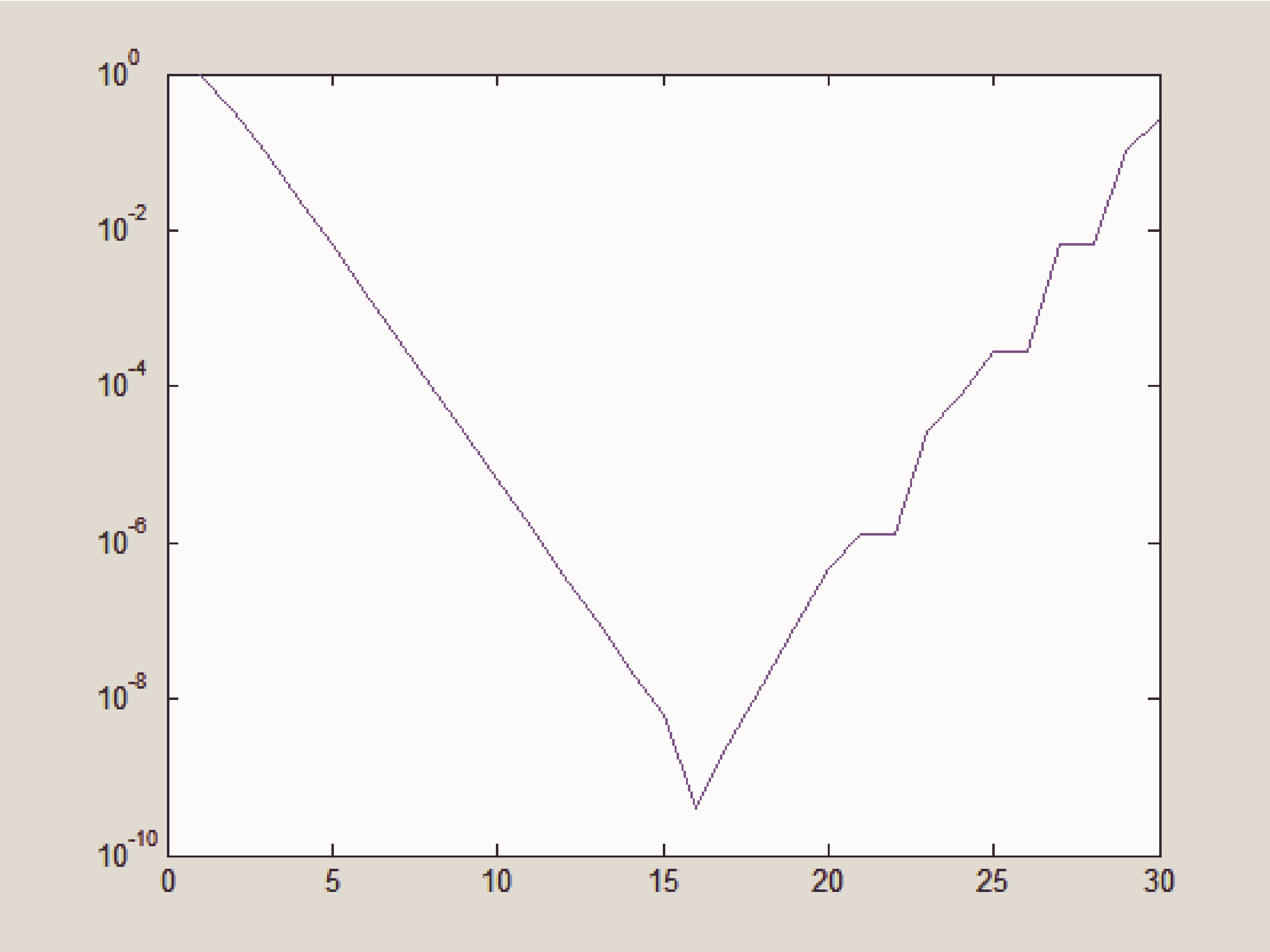
g = -1+x.*(7+x.*(-21+x.*(35+x.*(-35+x.*(21+x.*(-7+x))))));
plot(x,g)




```
h = (x-1).^7;
plot(x,h)
```



plot(x,h,x,g,x,f)




```
z(1) = 0; z(2) = 2;
for k = 2:29
    z(k+1) = 2^(k-1/2)*(1-(1-4^(1-k)*z(k)^2)^(1/2))^(1/2);
end
semilogy(1:30,abs(z-pi)/pi)
```

Vocabulary

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

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

Recurrence relations

$$x^2 - x - 1 = 0 \Leftrightarrow x = 1 + \frac{1}{x}$$

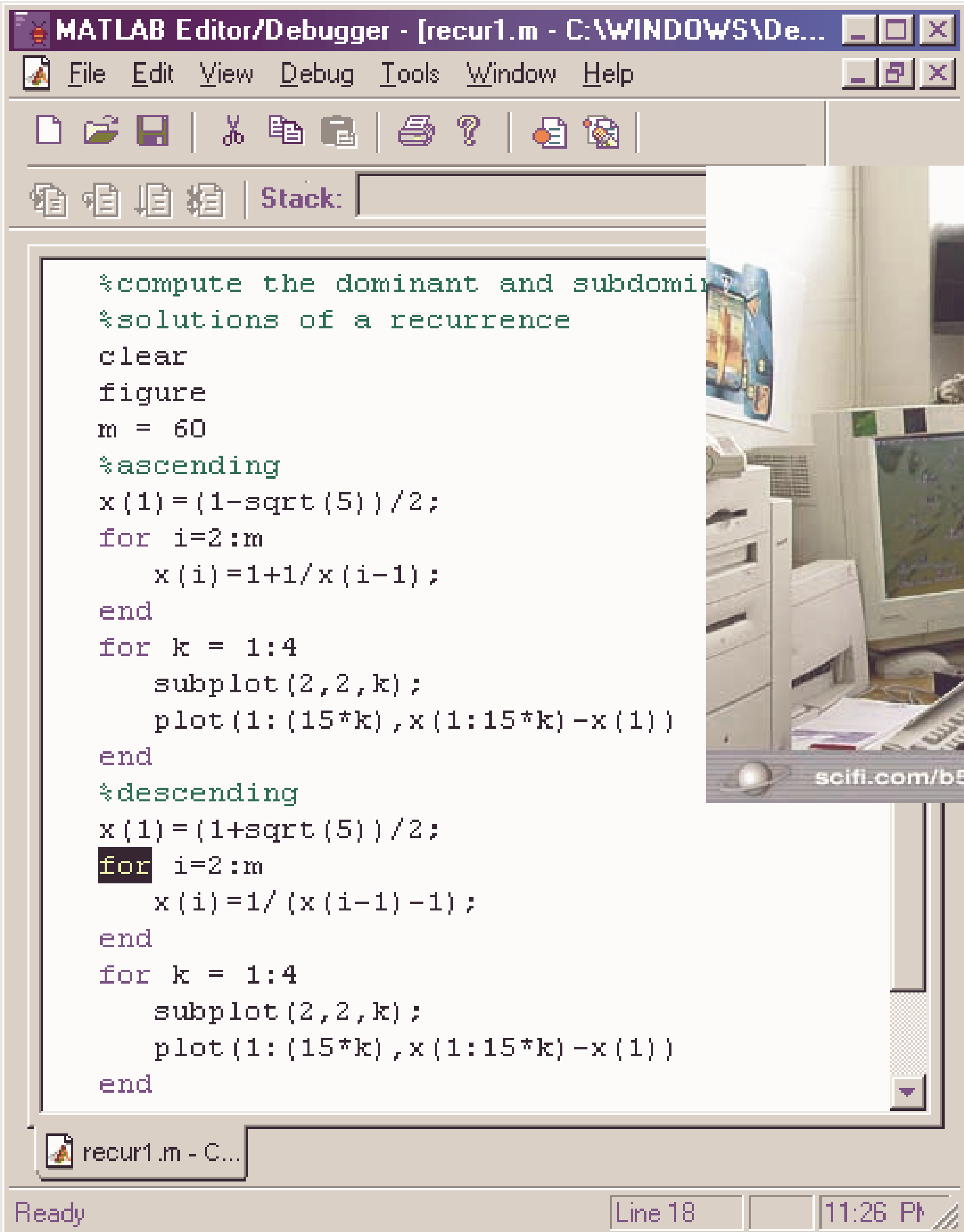
$$x_\pm = \frac{1 \pm \sqrt{5}}{2}$$

$$x_{n+1} = 1 + \frac{1}{x_n}$$

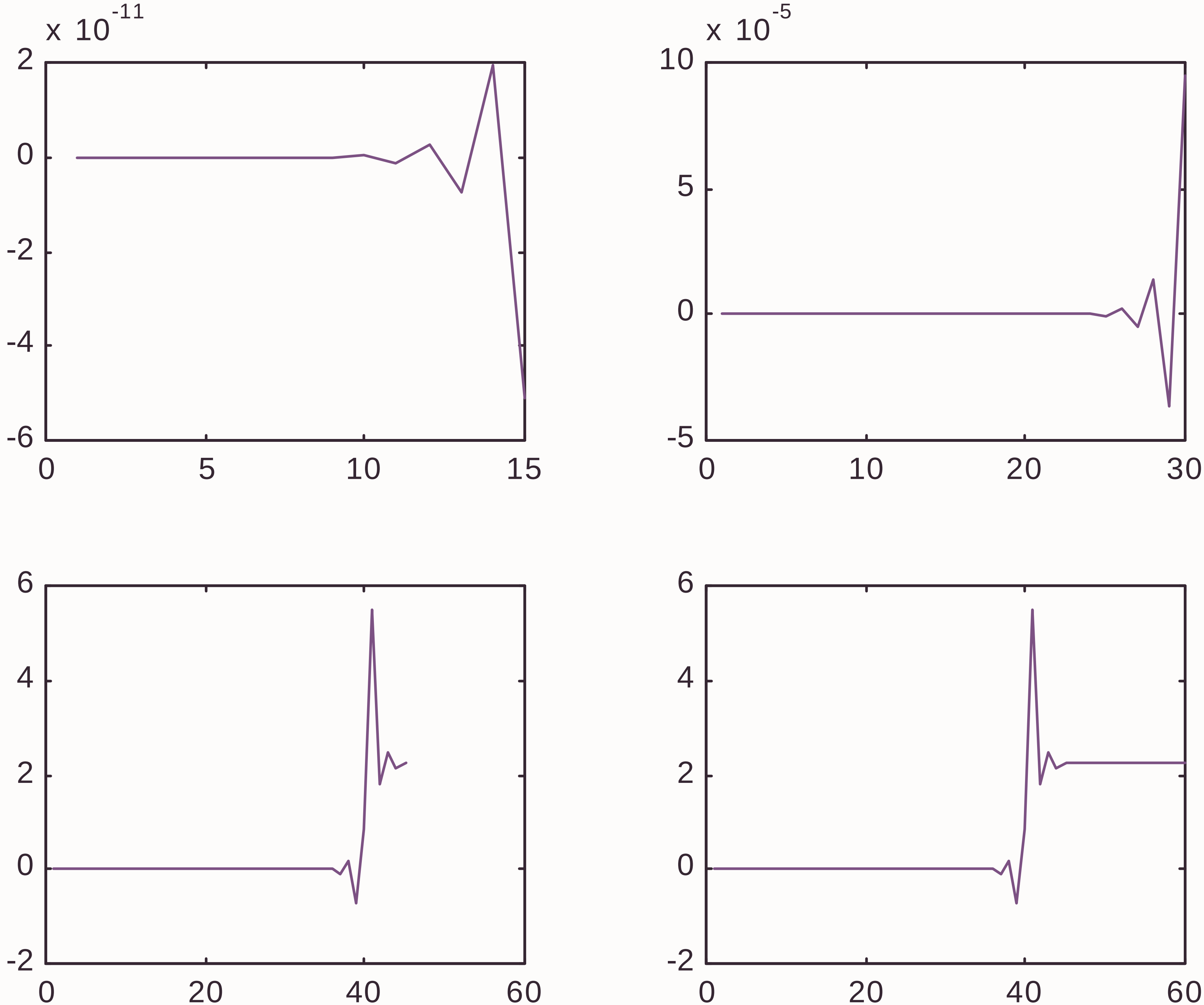
$$x_n^+ \rightarrow x_+$$

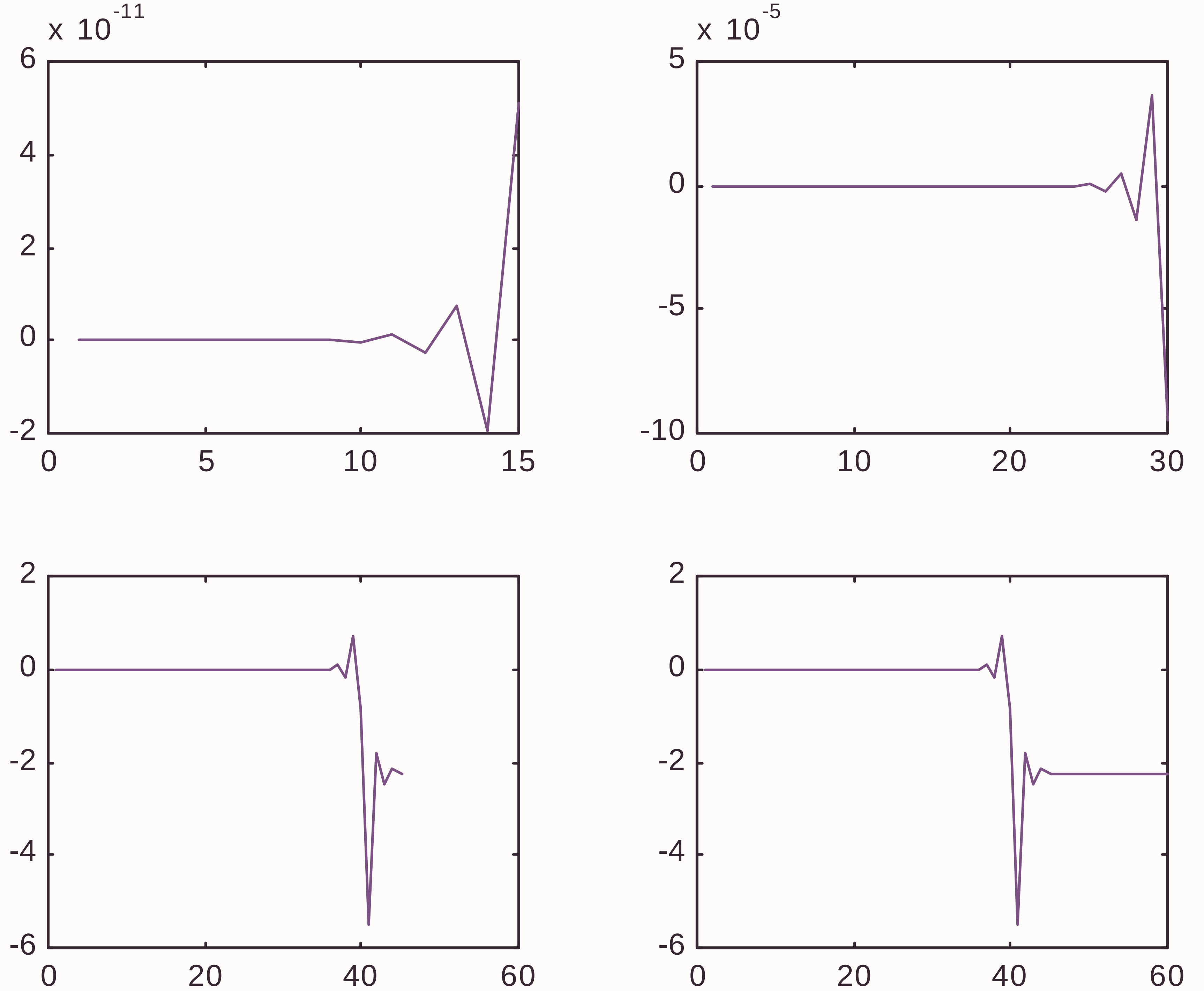
$$x_{n+1} = \frac{1}{x_n - 1}$$

$$x_n^- \rightarrow x_-$$



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

APPENDIX

- NUMERICAL MONSTERS:

  a collection of curious phenomena exhibited by computations at the limit of machine precision

The first monster:

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

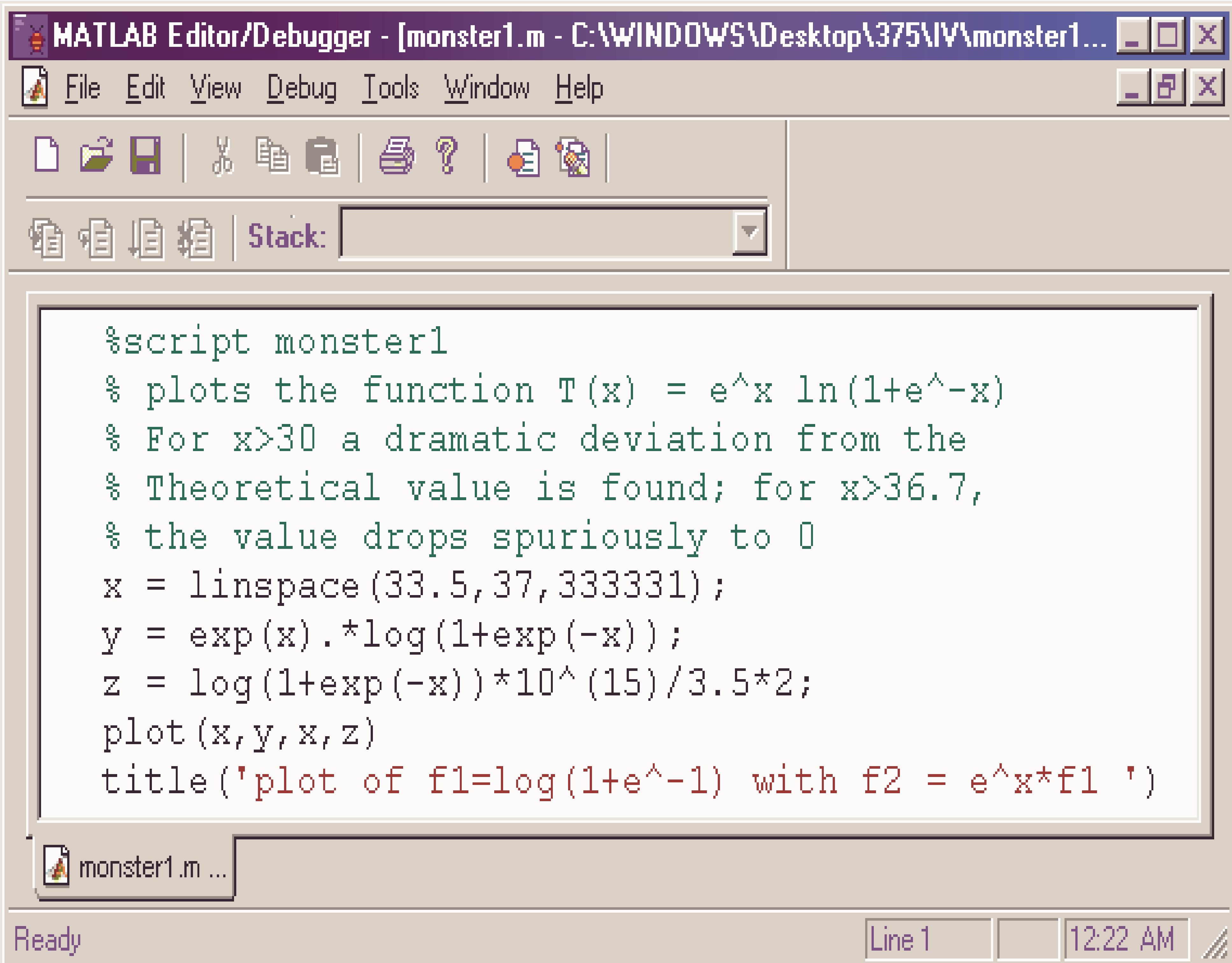
For x>30 a dramatic deviation from the theoretical value 1 is found.

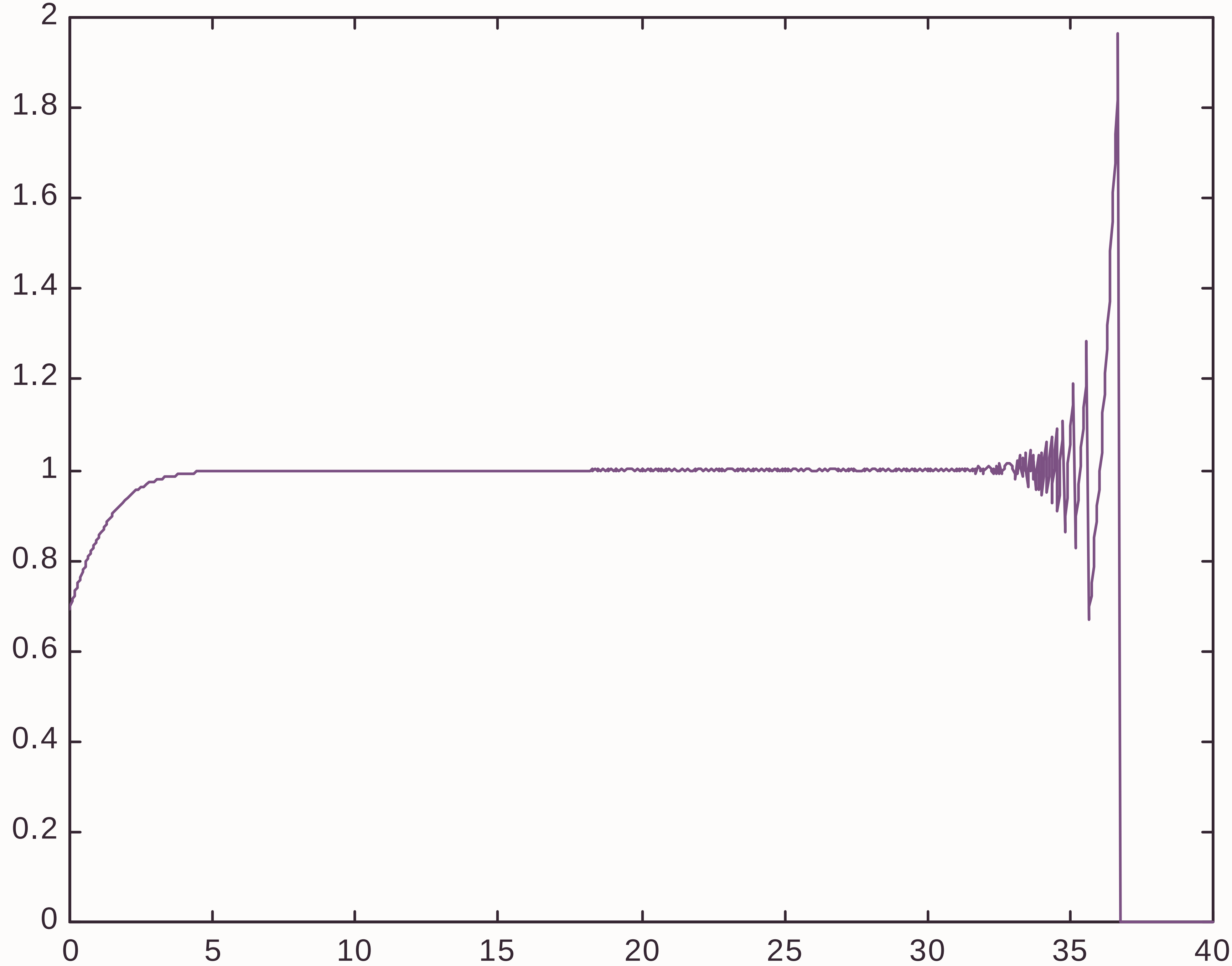
For x>36.7 the plotted value drops (incorrectly) to zero

Here:

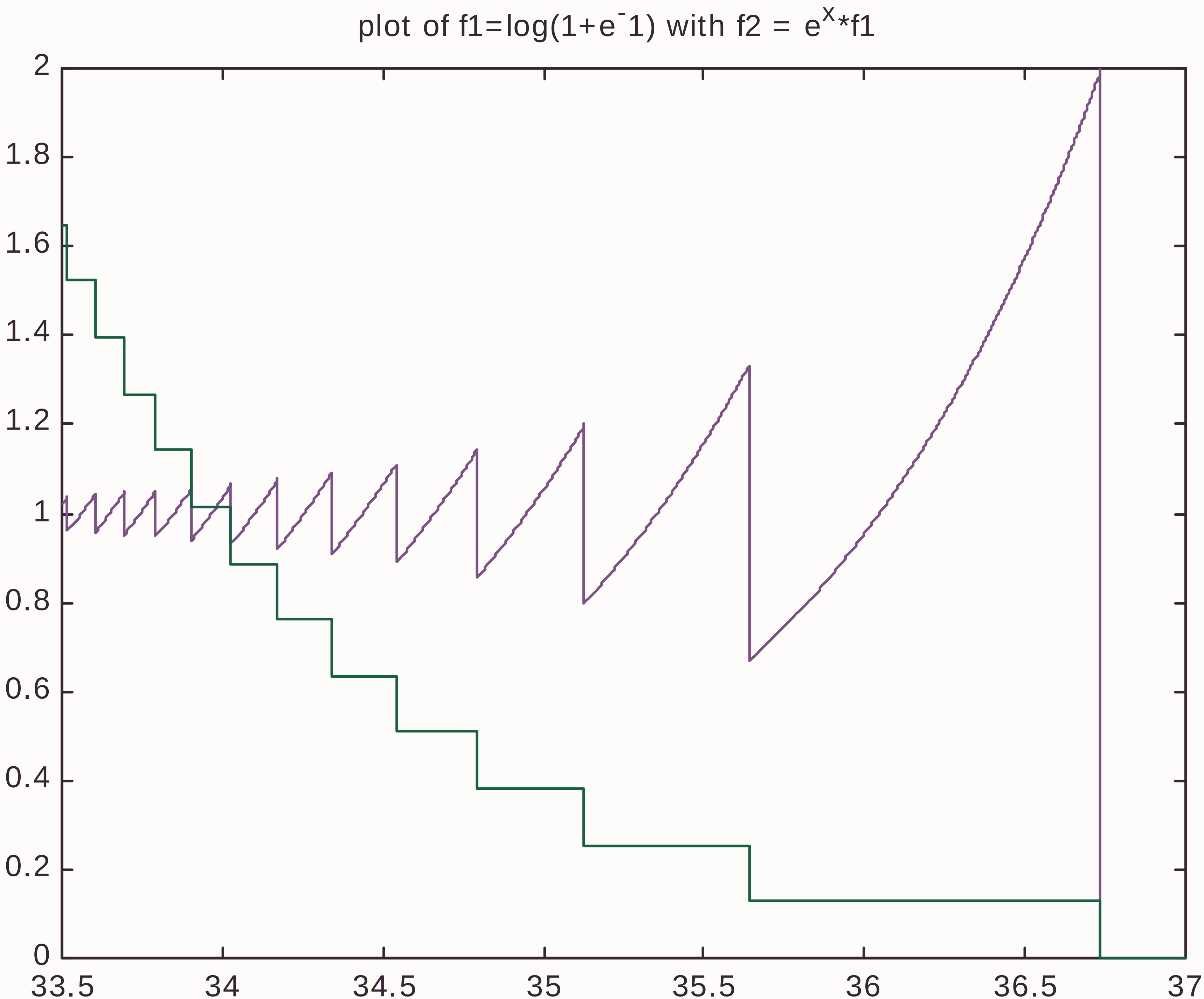
```
x = linspace(0,40,1000);
y = exp(x).*log(1+exp(-x));
```

```
for k=1:50, term=
```

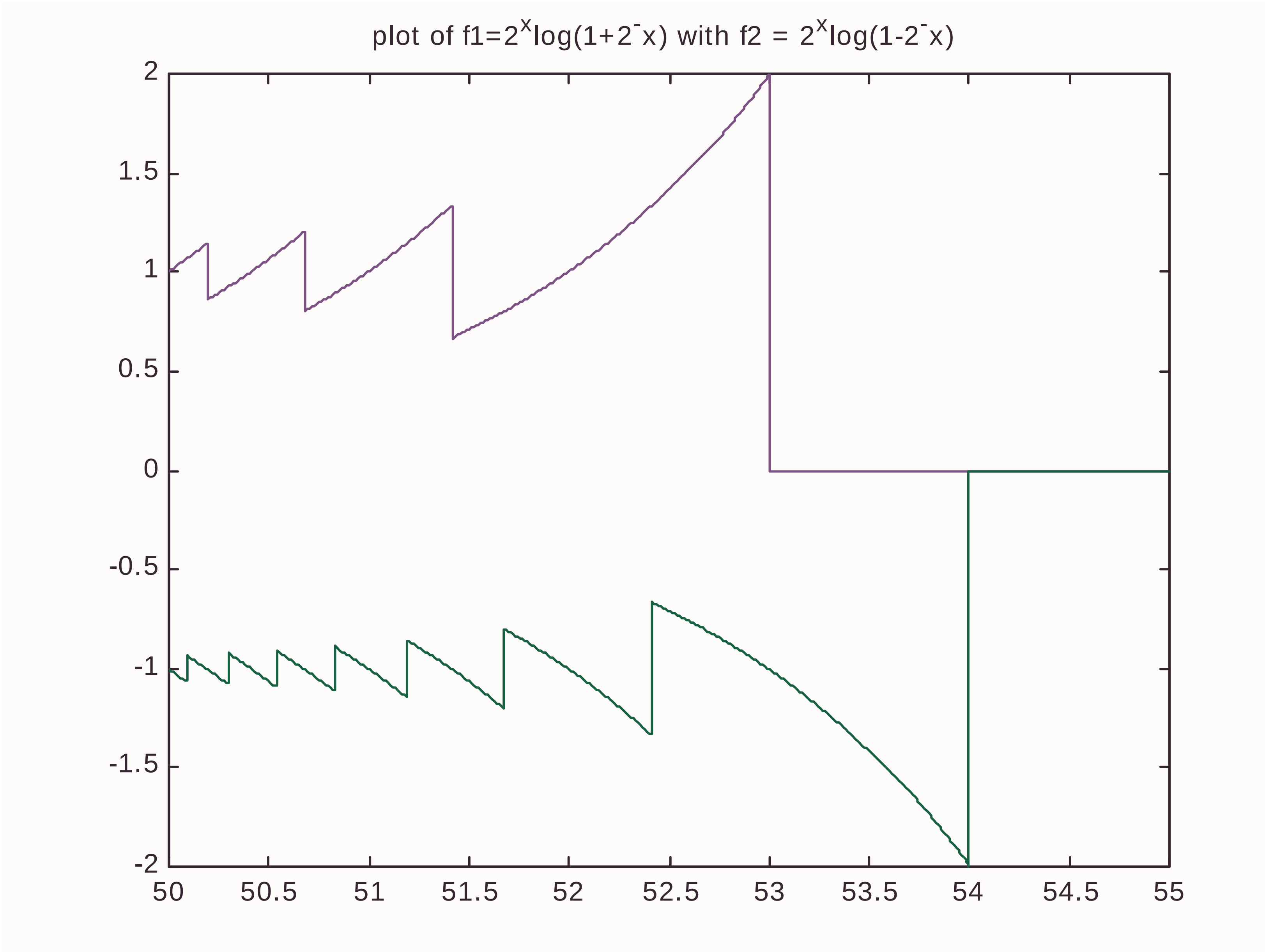

MATLAB Editor/Debugger - [monster1.m - C:\WINDOWS\Desktop\375\IV\monster1...
File Edit View Debug Tools Window Help
Stack:
%script monster1
% plots the function T(x) = e^x ln(1+e^-x)
% For x>30 a dramatic deviation from the
% Theoretical value is found; for x>36.7,
% the value drops spuriously to 0
x = linspace(33.5,37,333331);
y = exp(x).*log(1+exp(-x));
z = log(1+exp(-x))*10^(15)/3.5*2;
plot(x,y,x,z)
title('plot of f1=log(1+e^-1) with f2 = e^x*f1 ')
monster1.m ...
Ready
Line 1
12:22 AM

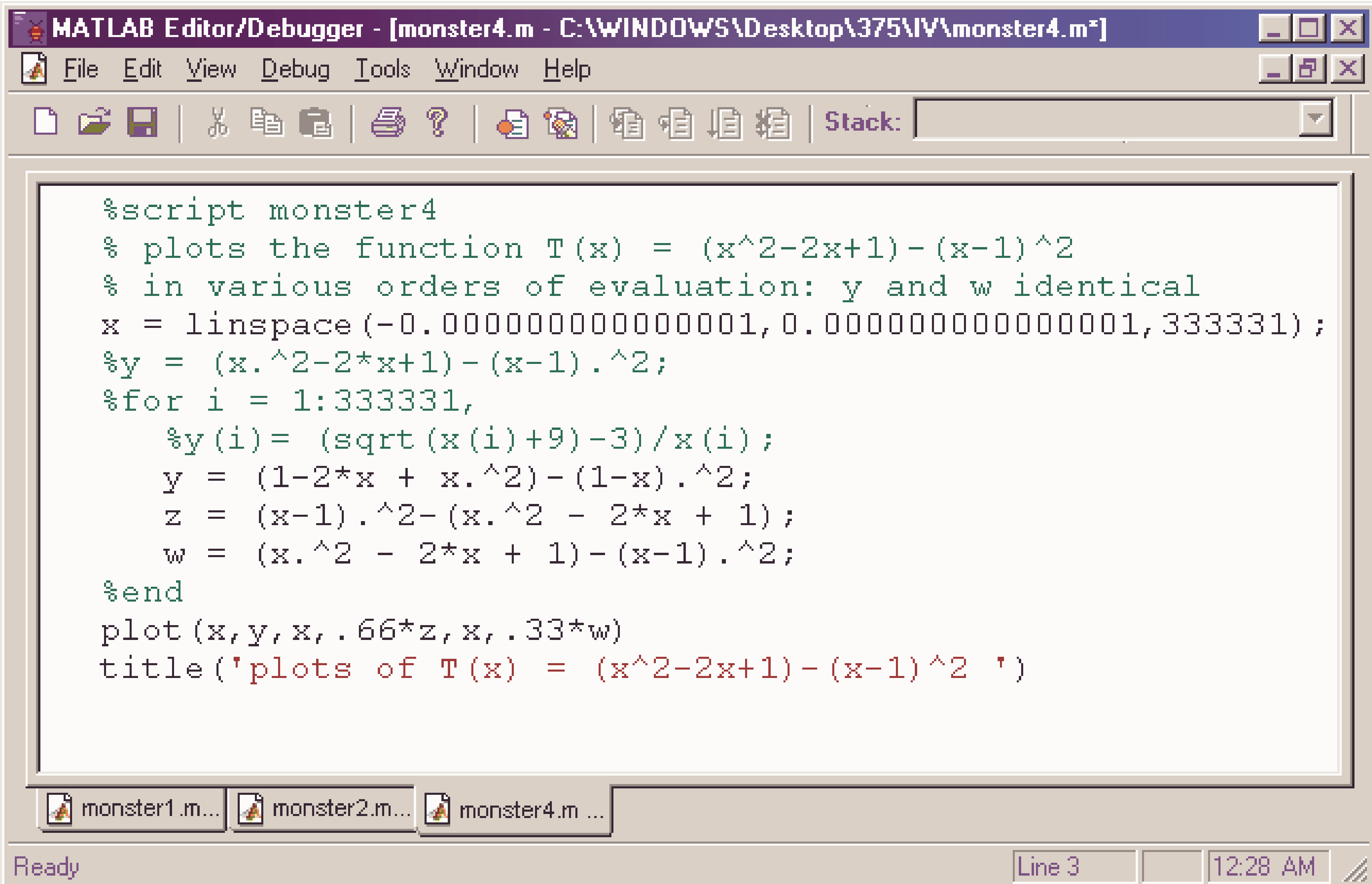


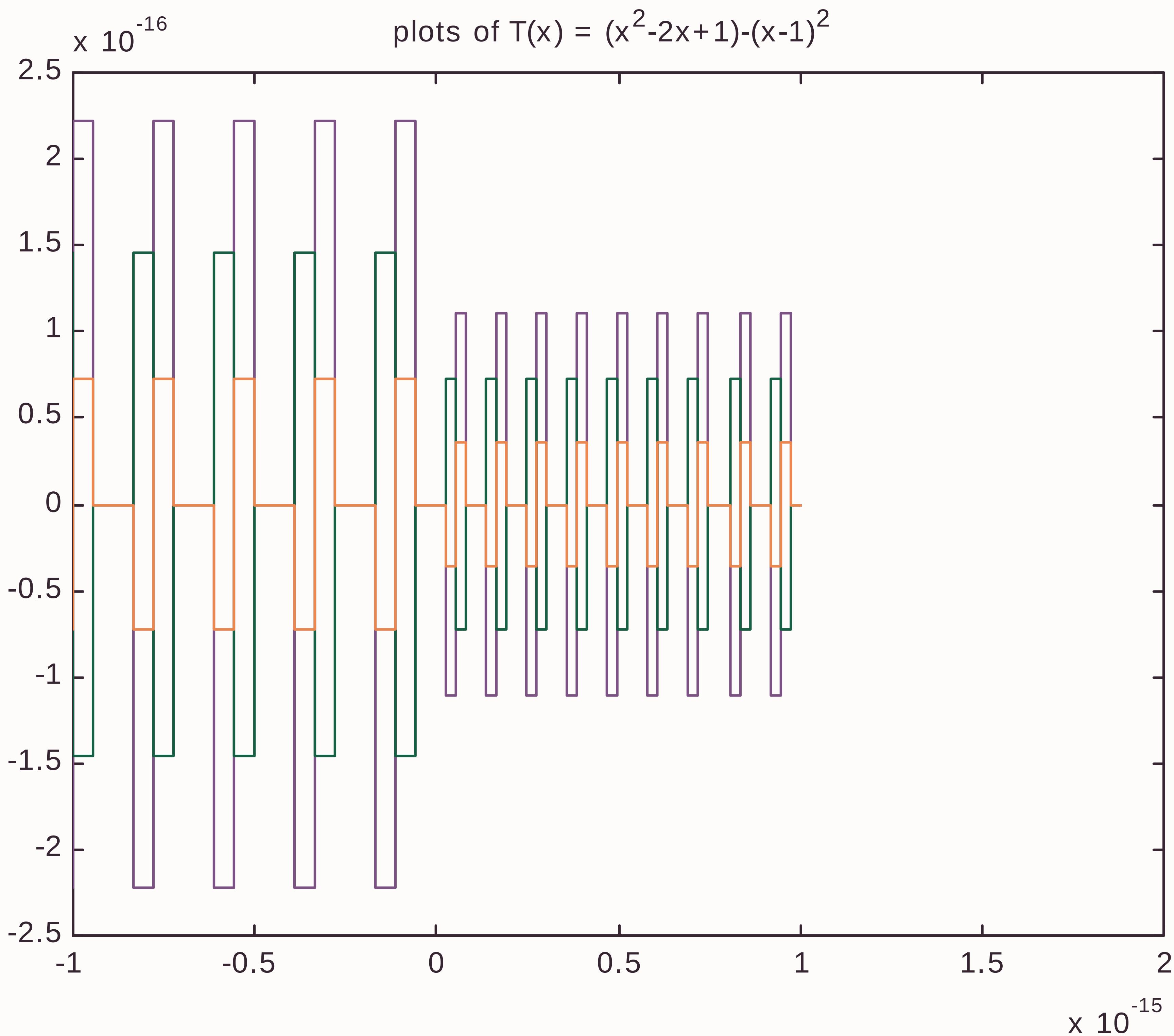
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
5
10
15
20
25
30
35
40


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

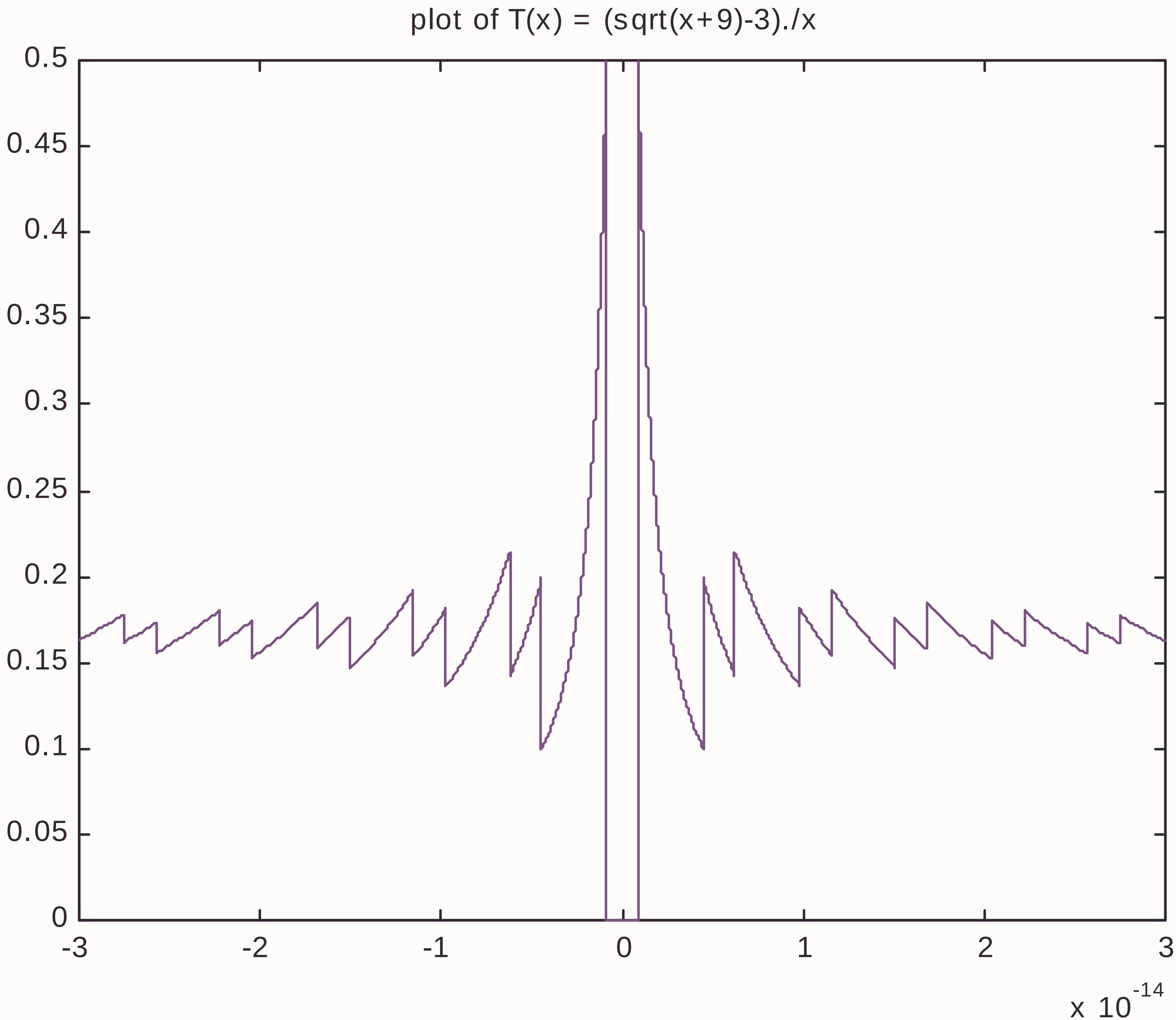
Pet Monster I $T(x) = 2^x \ln(1 + 2^{-x})$




MATLAB Editor/Debugger - [monster4.m - C:\WINDOWS\Desktop\375\IV\monster4.m*]
File Edit View Debug Tools Window Help
Stack:
%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 ')
monster1.m...
monster2.m...
monster4.m ...
Ready
Line 3
12:28 AM



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

III.MISCELLANEOUS OPERATIONS:

(1) Setting ranges for axes,

```
axis([-40 0 0 0.1])
```

(2) Superimposing plots:

```
plot(x,y,x,exp(x),'o')
```

(3) using "hold on/hold off"

(4) Subplots: `subplot(m,n,k)`

Summary

- Roundoff and other errors
- Multiple plots

References

- C. Essex, M.Davidson, C. Schulzky, "Numerical Monsters", preprint.
- Higham & Higham, Matlab Guide, SIAM
- B5 Trailer; http://www.scifi.com/b5rangers/