Math.375
II - Pandora's box

Vageli Coutsias

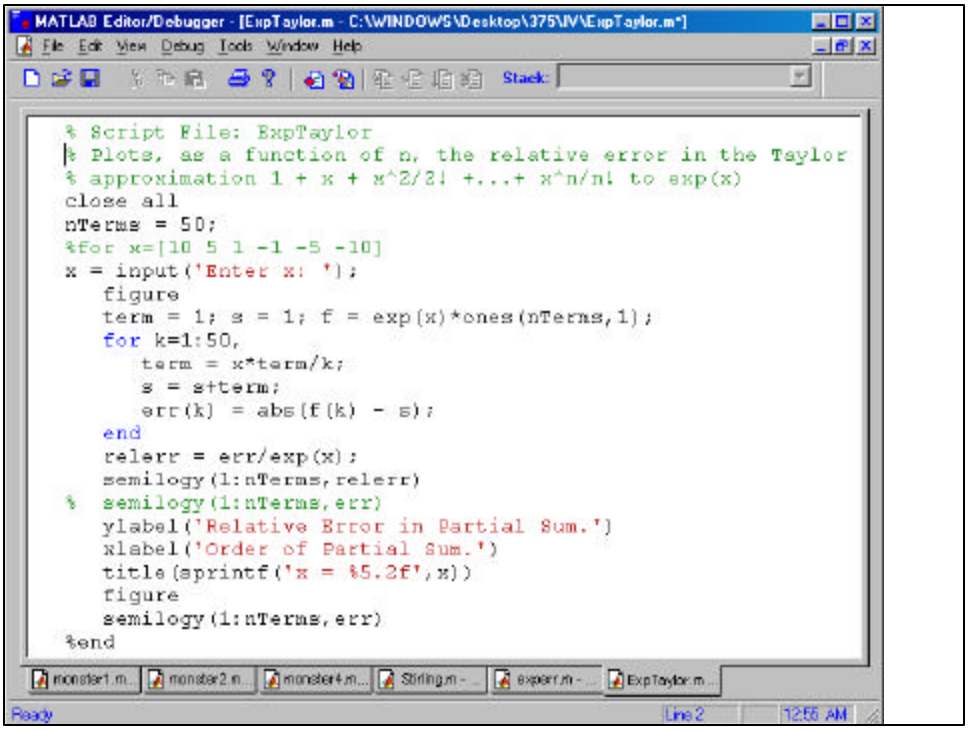
Some things do work!

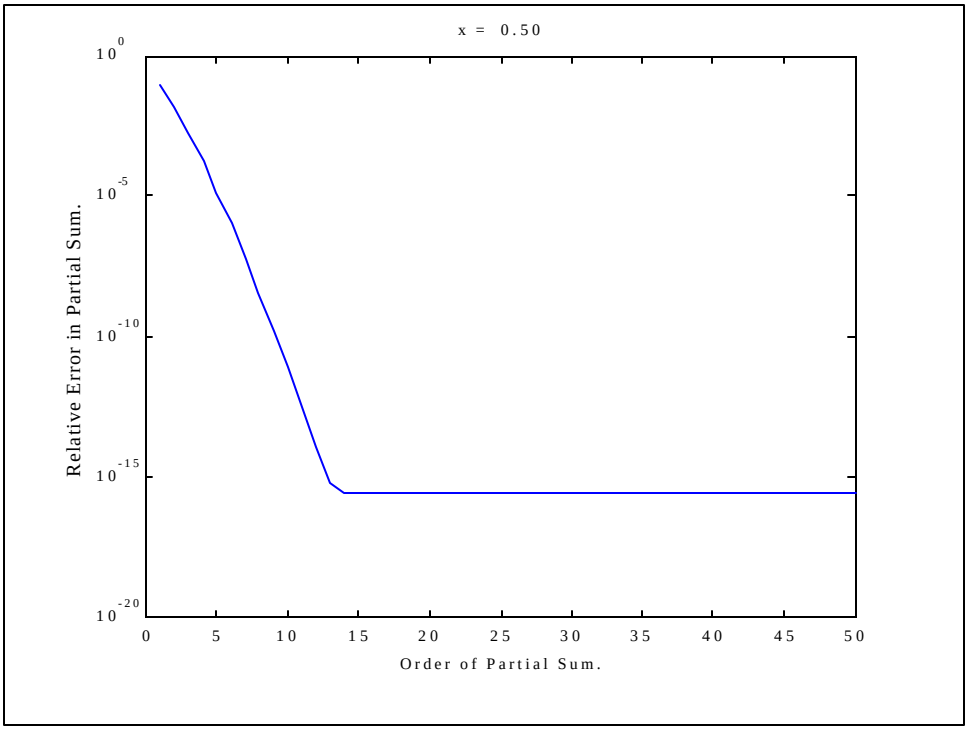
- Evaluating a convergent series
- Asymptotic formulas
- Polynomial evaluations
- Dynamical system calculations
- Recursive functions and fractals
- But: beware of errors and timing subtleties!

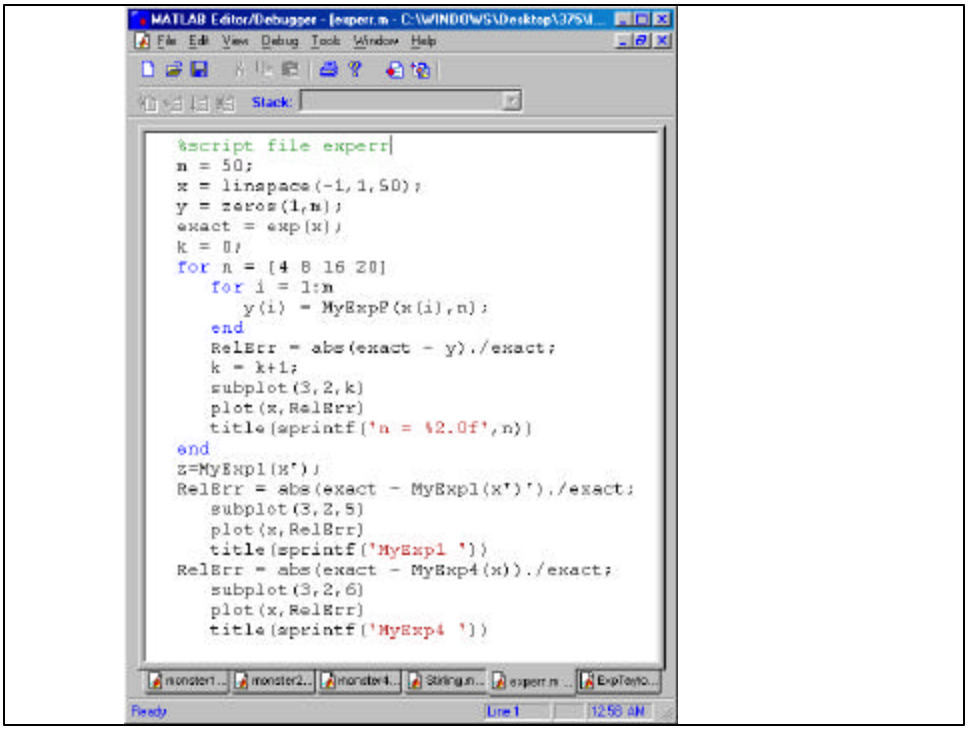
Taylor series for exp(x)

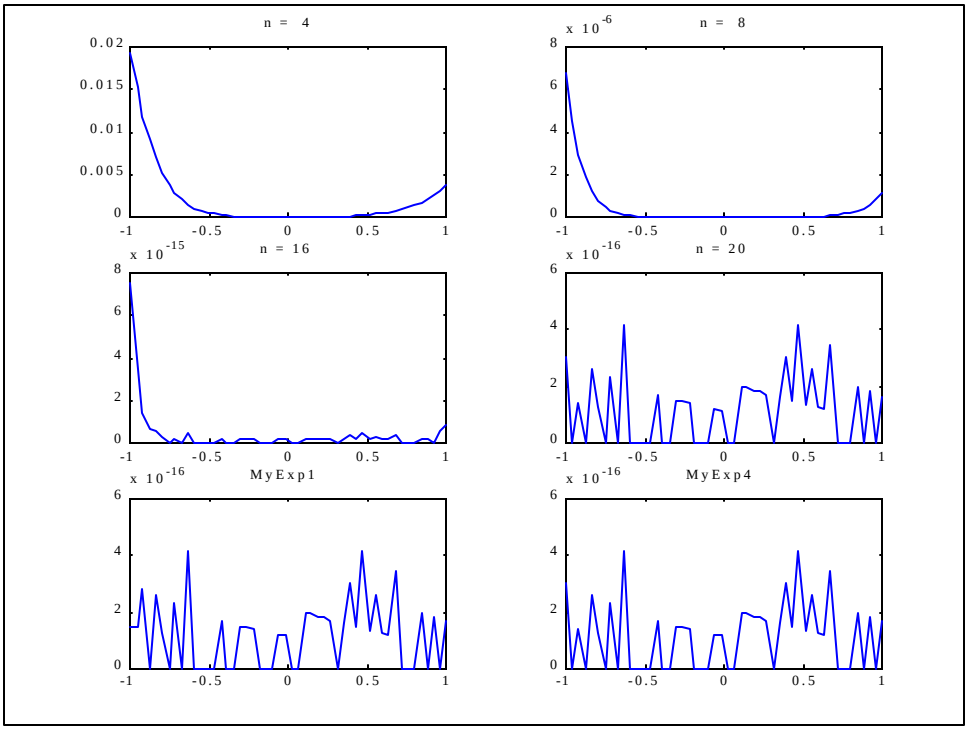
$$T_n(x) = \sum_{k=0}^{n} \frac{x^n}{k!}$$

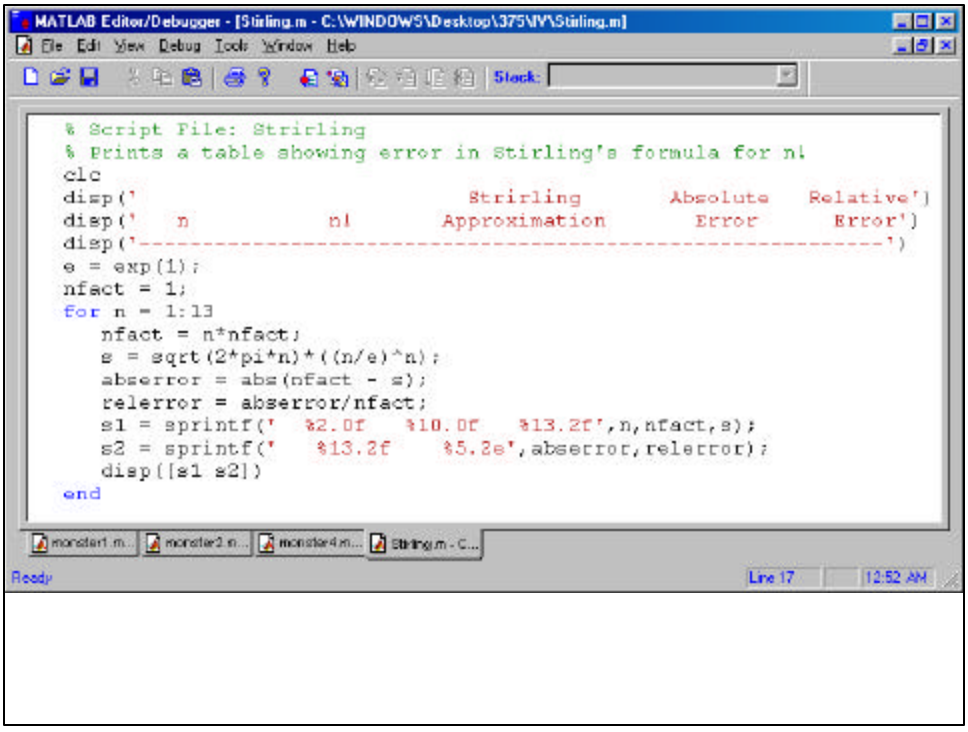
```
function y = expF(x,n)
term = 1; y = 1;
for k = 1:n
  x = x*term/k;
  y = y + term;
end
```




x = 0.50
Relative Error in Partial Sum.
Order of Partial Sum.
10^0
10^-5
10^-10
10^-15
10^-20
0
5
10
15
20
25
30
35
40
45
50


MATLAB Editor/Debugger - [experr.m - C:\WINDOWS\Desktop\375\I...
File Edit View Debug Tools Window Help
Stack:
%script file experr
n = 50;
x = linspace(-1,1,50);
y = zeros(1,n);
exact = exp(x);
k = 0;
for n = [4 8 16 20]
for i = 1:n
y(i) = MyExp2(x(i),n);
end
RelErr = abs(exact - y)./exact;
k = k+1;
subplot(3,2,k)
plot(x,RelErr)
title(sprintf('n = %2.0f',n))
end
z=MyExp1(x');
RelErr = abs(exact - MyExp1(x')')./exact;
subplot(3,2,5)
plot(x,RelErr)
title(sprintf('MyExp1 '))
RelErr = abs(exact - MyExp4(x))./exact;
subplot(3,2,6)
plot(x,RelErr)
title(sprintf('MyExp4 '))
monster1...
monster2...
monster4...
String.m...
experr.m ...
ExpTaylo...
Ready
Ln 1
12:58 AM


n = 4
n = 8
x 10^-6
n = 16
x 10^-15
n = 20
x 10^-16
MyExp1
x 10^-16
MyExp4
x 10^-16

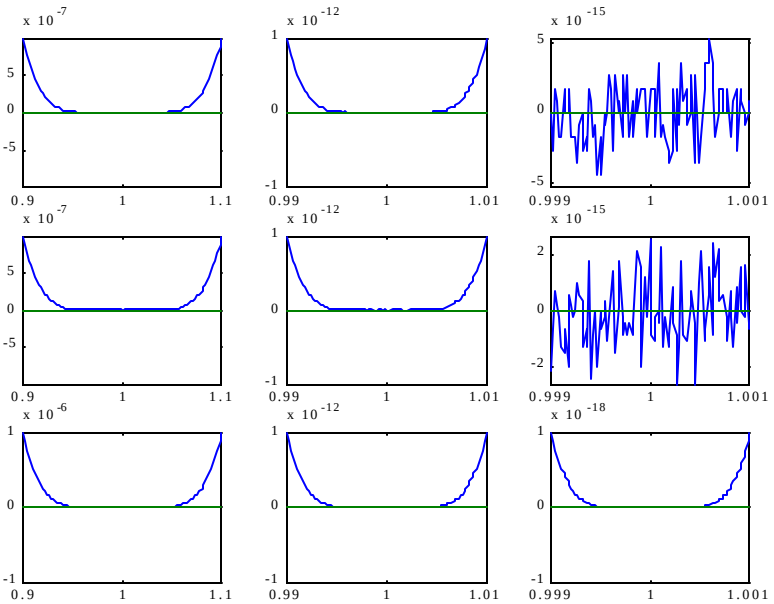

MATLAB Editor/Debugger - [Stirling.m - C:\WINDOWS\Desktop\375\IV\Stirling.m]
File Edit View Debug Tools Window Help
Stack:
% Script File: Strirling
% Prints a table showing error in Stirling's formula for n!
clc
disp('   Strirling   Absolute   Relative')
disp('   n   n!   Approximation   Error   Error')
disp('---')
e = exp(1);
nfact = 1;
for n = 1:13
   nfact = n*nfact;
   s = sqrt(2*pi*n)*((n/e)^n);
   abserror = abs(nfact - s);
   relerror = abserror/nfact;
   s1 = sprintf(' %2.0f   %10.0f   %13.2f',n,nfact,s);
   s2 = sprintf('   %13.2f   %5.2e',abserror,relerror);
   disp([s1 s2])
end
monster1.m...
monster2.m...
monster4.m...
Stirling.m - C...
Ready
Line 17
12:52 AM

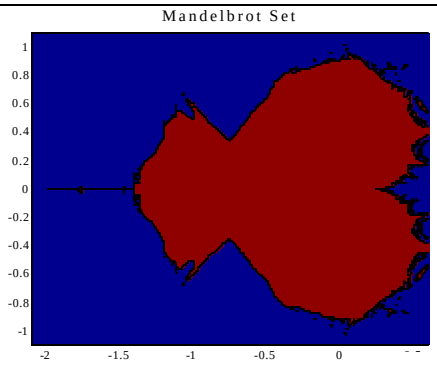
n	n!	Strirling Approximation	Absolute Error	Relative Error
1	1	0.92	0.08	7.79e-002
2	2	1.92	0.08	4.05e-002
3	6	5.84	0.16	2.73e-002
4	24	23.51	0.49	2.06e-002
5	120	118.02	1.98	1.65e-002
6	720	710.08	9.92	1.38e-002
7	5040	4980.40	59.60	1.18e-002
8	40320	39902.40	417.60	1.04e-002
9	362880	359536.87	3343.13	9.21e-003
10	3628800	3598695.62	30104.38	8.30e-003
11	39916800	39615625.05	301174.95	7.55e-003
12	479001600	475687486.47	3314113.53	6.92e-003
13	6227020800	6187239475.19	39781324.81	6.39e-003

$$S_n = \sqrt{2pn}\left(\frac{n}{e}\right)^n \approx n!$$

```
% Script File: Zoom
% Plots (x-1)^6 near x=1 with increasingly refined scale.
% Evaluation via binomial expansion leads to severe cancellation
% (x-1)^6 = x^6 - 6*x^5 + 15*x^4 - 20*x^3 + 15*x^2 -6*x + 1
close all
k = 0; n = 100;
for delta = [.1 .01 .001]
   x = linspace(1-delta,1+delta,n)';
   y = x.^6 - 6*x.^5 + 15*x.^4 - 20*x.^3 + 15*x.^2 -6*x +ones(n,1);
   z = ones(n,1) + x.*(-6 +x.*(15 +x.*(-20 +x.*(15 +x.*(-6+x)))));
   w = (x-1).^6;
   k = k+1; subplot(3,3,k); plot(x,y,x,zeros(1,n))
   axis([1-delta 1+delta -max(abs(y)) max(abs(y))])
   subplot(3,3,k+3); plot(x,z,x,zeros(1,n))
   axis([1-delta 1+delta -max(abs(z)) max(abs(z))])
   subplot(3,3,k+6); plot(x,w,x,zeros(1,n))
   axis([1-delta 1+delta -max(abs(w)) max(abs(w))])
end
```

$$f(x) = (x-1)^6$$ Polynomial, Horner, Matlab





$$\frac{dy_1}{dt} = 10*(y_2 - y_1)$$

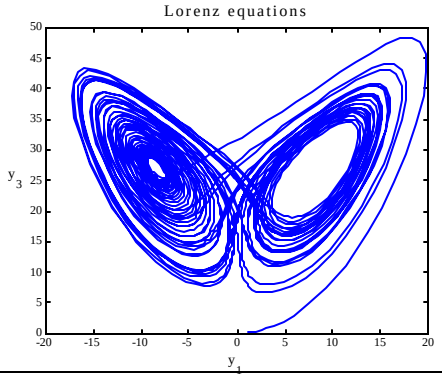
$$\frac{dy}{dt} = 28*y_1 - y_2 - y_1*y_3$$

$$\frac{dy}{dt} = y_1*y_2 - \frac{8}{3}y_3$$

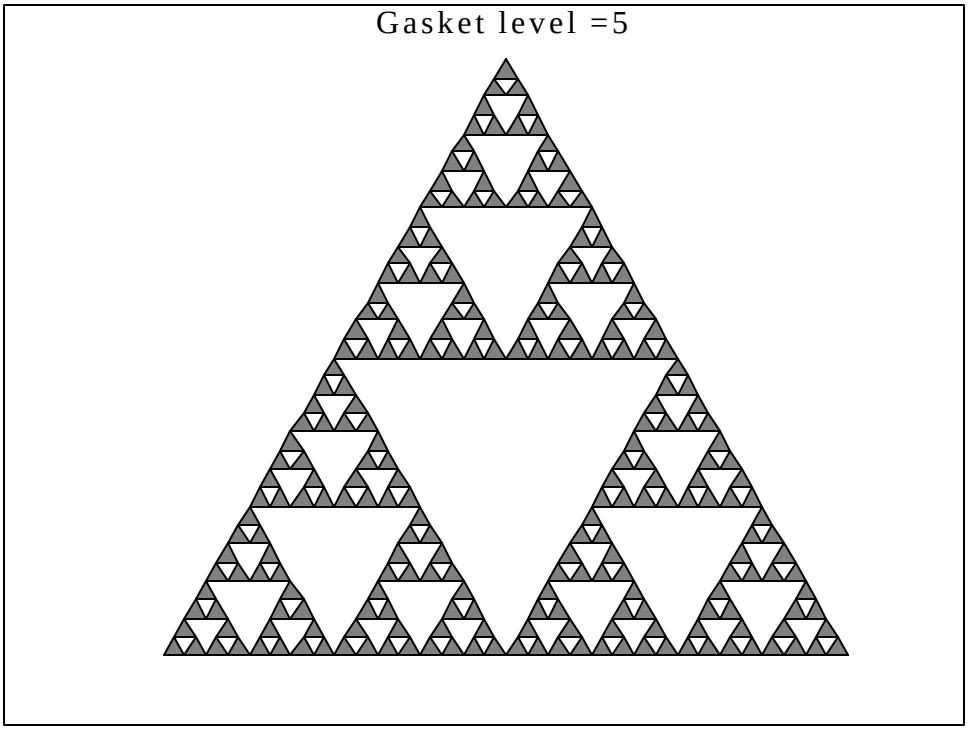
ITERATED
FUNCTIONS
(z→z^2 + c)

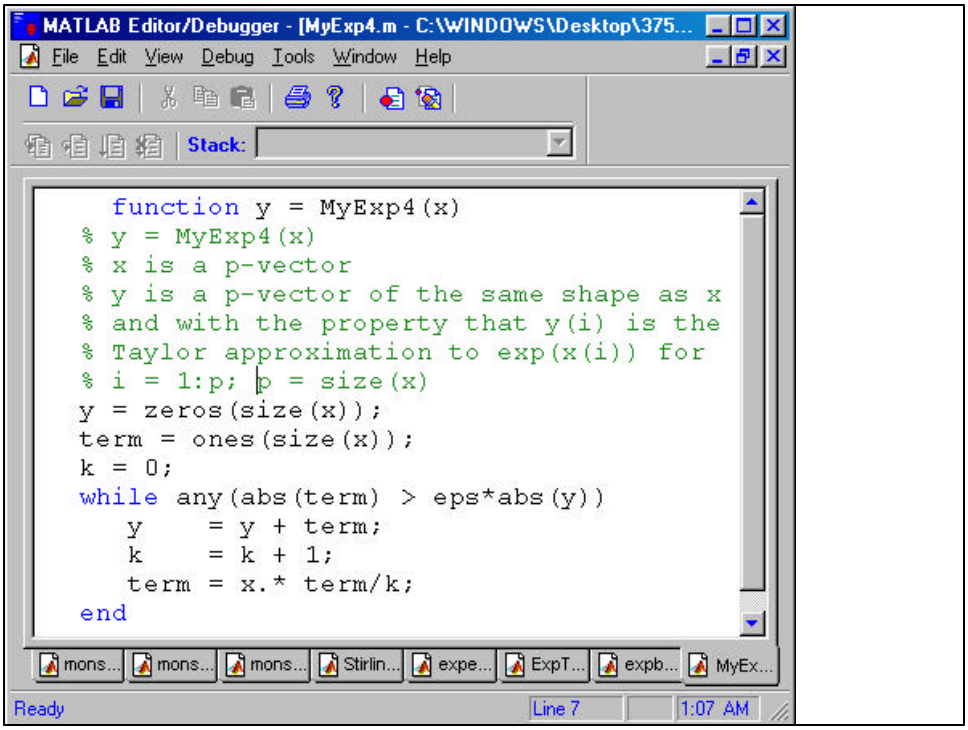
Mandel.m

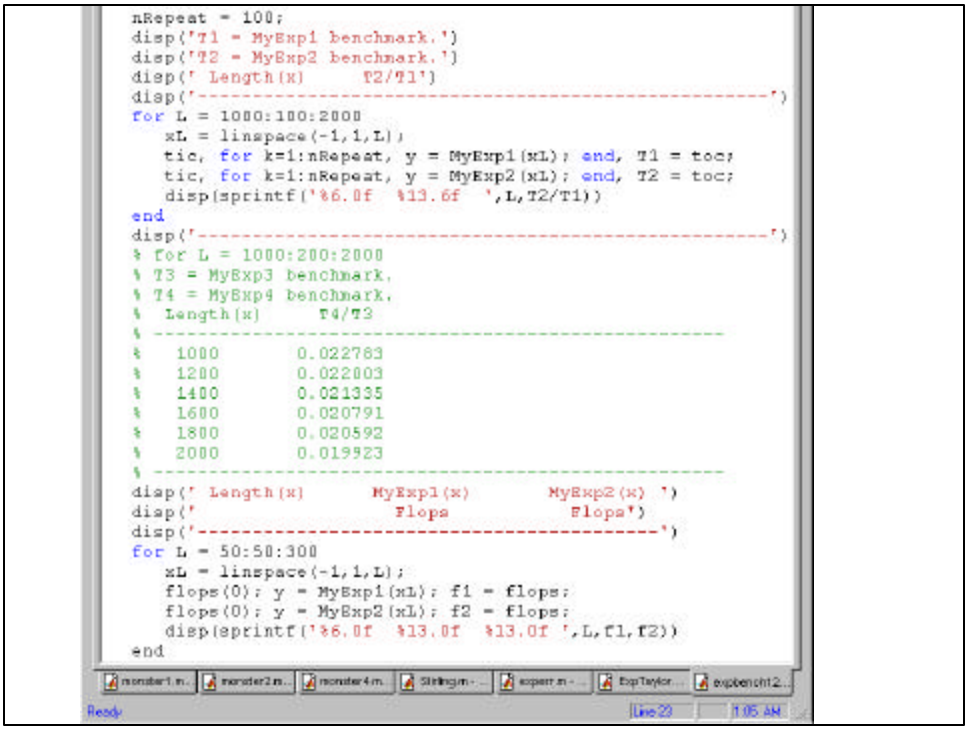
Lorenz.m/lorenzde.m



```
level = 5;
Pa=[0;0];
Pb=[1;0];
Pc=[0.5;sqrt(3)/2];
gasket(Pa,Pb,Pc,level)
hold off
title(['Gasket level =',num2str(level)],'FontSize',16)
axis('equal','off')
-------------------------
function gasket(Pa,Pb,Pc,level)
%GASKET Recursively generated Sierpinski gasket
%       GASKET(PA,PB,PC,LEVEL) generates an approximation to
%       the Sierpinski gasket, where the 2-vectors PA, PB and PC
%       define the triangle vertices.
%       LEVEL is the level of recursion
if level == 0
   % Fill the triangle with vertices Pa, Pb, Pc
   fill([Pa(1),Pb(1),Pc(1)],[Pa(2),Pb(2),Pc(2)],[0.5,0.5,0.5]);
   hold on
else
   % Recursive calls for the three subtriangles
   gasket(Pa,(Pa+Pb)/2,(Pa+Pc)/2,level-1)
   gasket(Pb,(Pb+Pa)/2,(Pb+Pc)/2,level-1)
   gasket(Pc,(Pc+Pa)/2,(Pc+Pb)/2,level-1)
end
```




MATLAB Editor/Debugger - [MyExp4.m - C:\WINDOWS\Desktop\375...
File Edit View Debug Tools Window Help
Stack:
function y = MyExp4(x)
% y = MyExp4(x)
% x is a p-vector
% y is a p-vector of the same shape as x
% and with the property that y(i) is the
% Taylor approximation to exp(x(i)) for
% i = 1:p; p = size(x)
y = zeros(size(x));
term = ones(size(x));
k = 0;
while any(abs(term) > eps*abs(y))
y = y + term;
k = k + 1;
term = x.* term/k;
end
mons... mons... mons... Stirlin... expe... ExpT... expb... MyEx...
Ready
Line 7
1:07 AM


nRepeat = 100;
disp('T1 = MyExp1 benchmark.')
disp('T2 = MyExp2 benchmark.')
disp(' Length(x) T2/T1')
disp('---')
for L = 1000:100:2000
xL = linspace(-1,1,L);
tic, for k=1:nRepeat, y = MyExp1(xL); end, T1 = toc;
tic, for k=1:nRepeat, y = MyExp2(xL); end, T2 = toc;
disp(sprintf('%6.0f %13.6f ',L,T2/T1))
end
disp('---')
% for L = 1000:200:2000
% T3 = MyExp3 benchmark.
% T4 = MyExp4 benchmark.
% Length(x) T4/T3
% ---
% 1000 0.022783
% 1200 0.022003
% 1400 0.021335
% 1600 0.020791
% 1800 0.020592
% 2000 0.019923
% ---
disp(' Length(x) MyExp1(x) MyExp2(x) ')
disp(' Flops Flops')
disp('---')
for L = 50:50:300
xL = linspace(-1,1,L);
flops(0); y = MyExp1(xL); f1 = flops;
flops(0); y = MyExp2(xL); f2 = flops;
disp(sprintf('%6.0f %13.0f %13.0f ',L,f1,f2))
end
Ready
Line 23
1:05 AM

Timing comparison: loop vs vector evaluation

Length(x)	T4/T3
1000	0.022759
1100	0.022709
1200	0.023486
1300	0.021685
1400	0.021779
1500	0.020487
1600	0.020402
1700	0.021111
1800	0.021815
1900	0.019904
2000	0.027449

Length(x)	T2/T1
1000	0.012977
1100	0.011803
1200	0.010809
1300	0.011815
1400	0.012363
1500	0.011542
1600	0.010827
1700	0.010192
1800	0.009615
1900	0.010139
2000	0.009623

Length(x)	Exp3(x)	Exp4(x) (flops)
50	3655	3869
100	7280	7719
150	10905	11569
200	14545	15419
250	18190	19269
300	21815	23119

Length(x)	Exp1(x)	Exp2(x)
50	2550	2550
100	5100	5100
150	7650	7650
200	10200	10200
250	12750	12750
300	15300	15300

Summary

- Timing and operation counts
- Matlab FUNCTIONS
- Loops vs. vectors: timing
- Multiple plots
- Examples of mathematical calculations

References

- C. Essex, M.Davidson, C. Schulzky, "Numerical Monsters", preprint.
- Higham & Higham, Matlab Guide, SIAM
- SIAM News, 29(8), 10/98 (Arianne V failure)
- find m-files at aix: ~coutsias/375/IV