```
%
 ==========================================================================
% AUTHOR ..... Lishan Huang
% DATE .... March 18
%
% Comparing different Numerical Integration
%
```

The method function of corrected trapezoidal rule

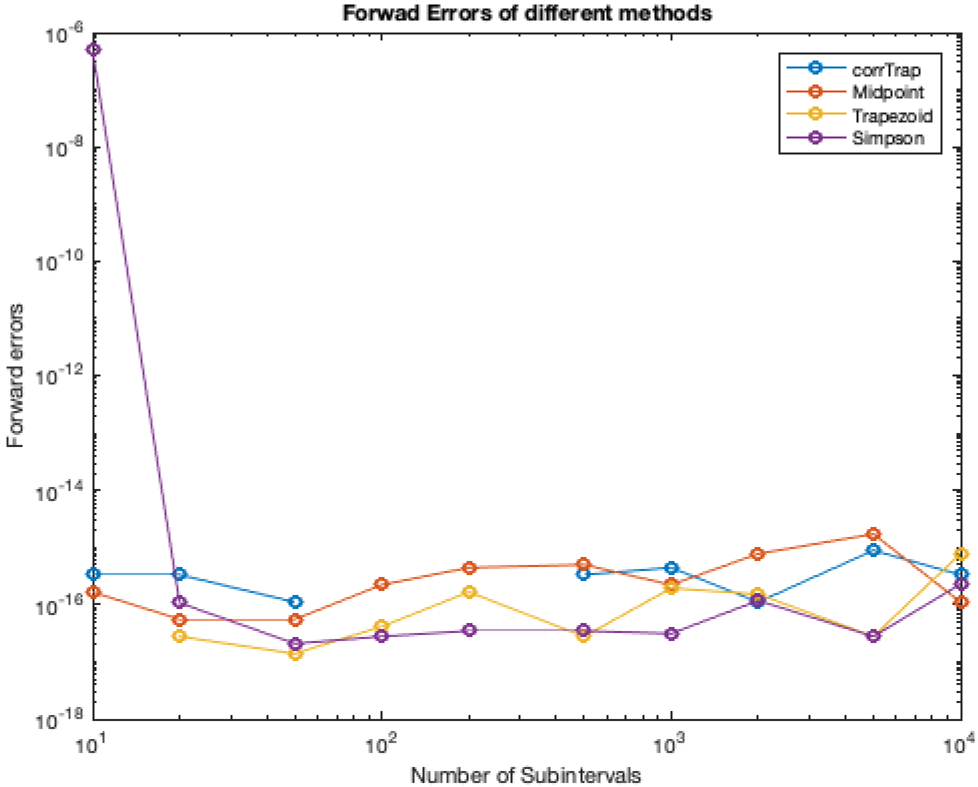
corrTrap.m

```
%
 ==========================================================================
% AUTHOR ..... Lishan Huang
% DATE .... March 18
%
% Function of the corrected trapezoidal rule
%
% INPUT
%
% A function handle for the function to be integrated
% Left endpoint of the interval to be integrated
% Right endpoint of the interval to be integrated
% The number of subdivisions to divide the interval into
% The values of f'(a) and f'(b) (if zero, just ordinary trapezoidal
 rule)
%
% OUTPUT
% The value of the integral of the input function between the endpoint
%
%
 ==========================================================================
%%
%* Script*
function m = corrTrap(f,LeftEnd,RightEnd,Num,dfa,dfb)
%store the input value
a=LeftEnd;
b=RightEnd;
n=Num;
h=(b-a)/n;
sigma=0;
%calculate sigmaf(xk) be definition
    for k=1:n-1
        sigma=sigma+f(a+k*h);
    end
    %formular of definition
m=h*(0.5*f(a)+sigma+0.5*f(b))+h^2*(dfa-dfb)/12;
end

=====================================================================
```

Problem 1 script

```
close all
clear all
%create a vector to store the number of subintervals to be evaluated
num = [10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000];
%define the Bessel function with different n value
f = @(t)cos(-sin(t))/pi;
f1 = @(t)cos(t - sin(t))/pi;
f2 = @(t)cos(2*t - sin(t))/pi;
f3 = @(t)cos(3*t - sin(t))/pi;
%the size of f(a)' and f(b)'
dfa = 0;
dfb = 0;
%left endpoint and right endpoint
a = 0;
b = pi;
%create a 4* length(num) matrix to store error for different method
Error=zeros(4,length(num));
for i = 1:length(num)
    %first row of Error stores the error of the corrected trapezoidal
 rule
    Error(1,i) = abs(corrTrap(f,a,b,num(i), dfa, dfb)-besselj(0,1));
    %sencond row of Error stores the  error of the composite midpoint
 rule
    Error(2,i) = abs(midpointRule(f1,a, b, num(i))-besselj(1,1));
     %sencond row of Error stores the  error of the composite
 trapezoidal rule
    Error(3,i) = abs(trapezoidRule(f2,a, b, num(i))-besselj(2,1));
     %sencond row of Error stores the  error of the Simpson?s rule
    Error(4,i) = abs(simpsonsRule(f3,a, b, num(i))-besselj(3,1));
end
%plot the error
loglog(num, Error(1,:),'o-', num, Error(2,:), 'o-', num,
 Error(3,:), 'o-', num, Error(4,:), 'o-')
xlabel('Number of Subintervals');
ylabel('Forward errors');
legend("corrTrap","Midpoint","Trapezoid","Simpson")
title("Forwad Errors of different methods")
%converge rate
CorrTrap=abs(Error(1,length(num))-0)/abs(Error(1,length(num-1))-0);
Midpoint=abs(Error(2,length(num))-0)/abs(Error(2,length(num-1))-0);
trapezo=abs(Error(3,length(num))-0)/abs(Error(3,length(num-1))-0);
simpson=abs(Error(4,length(num))-0)/abs(Error(4,length(num-1))-0);
fprintf('Converge rate of the corrected trapezoidal rule is %d
\n',CorrTrap)
fprintf('Converge rate of the composite midpoint rule is %d\n',
 Midpoint)
fprintf('Converge rate of the composite trapezoidal rule is %d
\n',trapezo)
fprintf('Converge rate of the Simpson?s rule rule is %d\n',simpson)

Converge rate of the corrected trapezoidal rule is 1
Converge rate of the composite midpoint rule is 1
```

Converge rate of the composite trapezoidal rule is 1
Converge rate of the Simpson?s rule rule is 1




```
%Text answer
%we can see that the graph show all the error of those method converge
 to 0
%and the converge rate is 1 by the caculation as above
```

Published with MATLAB® R2017b