EN.553.481/681 Numerical Analysis – Homework 4 Solutions

Problem 1. (a) Since $\alpha_i := \int_a^b \prod_{j\neq i}(x-x_j)/(x_i-x_j)\,\mathrm{d}x$, we have

$$\begin{aligned}
\alpha_0 &= \frac{1}{4!h^4}\int_a^b (x-x_1)(x-x_2)(x-x_3)(x-x_4)\,\mathrm{d}x\\
\alpha_1 &= -\frac{1}{3!h^4}\int_a^b (x-x_0)(x-x_2)(x-x_3)(x-x_4)\,\mathrm{d}x\\
\alpha_2 &= \frac{1}{2!2!h^4}\int_a^b (x-x_0)(x-x_1)(x-x_3)(x-x_4)\,\mathrm{d}x\\
\alpha_3 &= -\frac{1}{3!h^4}\int_a^b (x-x_0)(x-x_1)(x-x_2)(x-x_4)\,\mathrm{d}x\\
\alpha_4 &= \frac{1}{4!h^4}\int_a^b (x-x_0)(x-x_1)(x-x_2)(x-x_3)\,\mathrm{d}x.
\end{aligned}$$

From Program 1.1 we find that

$$\alpha_0 = \frac{14h}{45},\quad \alpha_1 = \frac{64h}{45},\quad \alpha_2 = \frac{8h}{15},\quad \alpha_3 = \frac{64h}{45},\quad \alpha_4 = \frac{14h}{45}.$$

Recall that $f(x) = \sum_{i=0}^4 \alpha_i f(x_i) + f[x_0,\dots,x_4,x]\prod_{i=0}^4(x-x_i)$. Apply integration by parts, the mean value theorem, and Program 1.1 again to write

$$\begin{aligned}
E_4(f) &= \int_a^b f[x_0,\dots,x_4,x]\frac{\mathrm{d}}{\mathrm{d}x}w(x)\,\mathrm{d}x\\
&= -\int_a^b w(x)\frac{\mathrm{d}}{\mathrm{d}x}f[x_0,\dots,x_4,x]\,\mathrm{d}x\\
&= -\int_a^b f[x_0,\dots,x_4,x,x]w(x)\,\mathrm{d}x\\
&= -f[x_0,\dots,x_4,\xi,\xi]\int_a^b w(x)\,\mathrm{d}x\\
&= -\frac{f^{(6)}(\xi)}{6!}\int_a^b\int_0^x \prod_{i=0}^4(x'-x_i)\,\mathrm{d}x'\,\mathrm{d}x\\
&= \frac{f^{(6)}(\xi)}{6!}\int_a^b (x-x_0)\prod_{i=0}^4(x-x_i)\,\mathrm{d}x\\
&= \frac{8h^7}{945}f^{(6)}(\xi).
\end{aligned}$$

It's just algebra to see that assembling the collected α_i, $f(x_i)$, and $E_4(f)$ fits the desired form.

(b) See Program 1.2.

(c) See Program 1.3 and its outputs Figure 1.1 and Figure 1.2 for the integral (i), and Program 1.4 and its outputs Figure 1.3 and Figure 1.4 for the integral (ii). Note the results for the composite Simpson's rule are consistent with our expectations because its error is of order 4. By a similar argument to that of the composite Simpson's rule error, we expect the error of the composite Boole's rule to be of order 6. We see this is roughly the case, better observed for larger n.

Problem 2. (a) Write

$$\begin{aligned}
\sum_{j=1}^{\infty} B_j(1-x)\frac{(-t)^j}{j!} &= \frac{-t(e^{tx-t}-1)}{e^{-t}-1} \\
&= \frac{-t(e^{tx}-e^{t})}{1-e^{t}} \\
&= \frac{t(e^{tx}-e^{t})}{e^{t}-1} \\
&= \frac{t(e^{tx}-1)}{e^{t}-1} + \frac{t(1-e^{t})}{e^{t}-1} \\
&= \sum_{j=1}^{\infty} B_j(x)\frac{t^j}{j!} - t.
\end{aligned}$$

Note the series expansion for t vanishes at the second and higher orders. Thus $B_1(1-x) = 1 - B_1(x)$ and $B_j(1-x) = B_j(x)$ for $j \geq 2$.

(b) Write

$$\begin{aligned}
\sum_{j=0}^{\infty} \frac{\mathrm{d}}{\mathrm{d}x} B_j(x)\frac{t^j}{j!} &= \frac{\mathrm{d}}{\mathrm{d}x}\frac{t(e^{tx}-1)}{e^{t}-1} \\
&= \frac{t^2 e^{tx}}{e^{t}-1} \;= \frac{t^2(e^{tx}-1)}{e^{t}-1} + \frac{t^2}{e^{t}-1} \\
&= \sum_{j=1}^{\infty} \left[B_j(x) + B_j\right]\frac{t^{j+1}}{j!} \\
&= \sum_{j=1}^{\infty} (j+1)\left[B_j(x) + B_j\right]\frac{t^{j+1}}{(j+1)!} \\
&= \sum_{j=2}^{\infty} j\left[B_{j-1}(x) + B_{j-1}\right]\frac{t^j}{j!}
\end{aligned}$$

.

For $j \geq 2$ this implies $B_j'(x) = j[B_{j-1}(x) + B_{j-1}]$, as desired.

Problem 3. (a) From the Euler-MacLaurin formula,

$$I_n^T(f) \sim I(f) - \frac{1}{12}(f'(b) - f'(a))h^2 + O(h^4).$$

Combining this result with the asymptotic formula from the problem statement

$$I_n^M(f) \sim I(f) + \frac{1}{24}(f'(b) - f'(a))h^2 + O(h^4),$$

one can see that

$$2I_n^M(f) + I_n^T(f) \sim 3I(f) + O(h^4).$$

The improved method $\tilde{I}_n(f) := \frac{2}{3}I_n^M(f) + \frac{1}{3}I_n^T(f)$ thus has 4th-order rate of convergence. This is composite Simpson's rule with increments of size $h/2$, as one can see from the result for a single interval $[a,b]$ with midpoint $c = \frac{a+b}{2}$ where

$$\frac{2}{3}hf(c) + \frac{1}{3}\frac{h}{2}[f(a) + f(b)] = \frac{h}{6}[f(a) + 4f(c) + f(b)] = \frac{h/2}{3}[f(a) + 4f(c) + f(b)].$$

(b) Write composite trapezoidal with stepsize h_k as

$$T^{(0)}(h_k) = \sum_{j=0}^{2^k-1} h_k \frac{f(x_j)+f(x_{j+1})}{2} = \sum_{j=0}^{2^{k-1}-1} h_k \frac{f(x_{2j})+2f(x_{2j+1})+f(x_{2j+2})}{2},$$

and with stepsize $h_{k-1} = 2h_k$ as

$$T^{(0)}(h_{k-1}) = \sum_{j=0}^{2^{k-1}-1} h_k \left[f(x_{2j}) + f(x_{2j+2})\right],$$

so that combining them yields

$$T^{(1)}(h_k) = \frac{1}{3}\left[4T^{(0)}(h_k) - T^{(1)}(h_{k-1})\right] = \frac{1}{3}\sum_{j=0}^{2^{k-1}-1} h_k \left[f(x_{2j}) + 4f(x_{2j+1}) + f(x_{2j+2})\right],$$

which is composite Simpson. We expand

$$\frac{1}{3}\sum_{j=0}^{2^{k-1}-1} h_k \left[f(x_{2j}) + 4f(x_{2j+1}) + f(x_{2j+2})\right] = \frac{1}{3}\sum_{j=0}^{2^{k-2}-1} h_k \left[f(x_{4j}) + 4f(x_{4j+1}) + 2f(x_{4j+2}) + 4f(x_{4j+3}) + f(x_{4j+4})\right],$$

and write

$$T^{(1)}(h_{k-1}) = \frac{1}{3}\sum_{j=0}^{2^{k-2}-1} 2h_k \left[f(x_{4j}) + 4f(x_{4j+2}) + f(x_{4j+4})\right],$$

so that combining yields

$$T^{(2)}(h_k) = \frac{16T^{(1)}(h_k) - T^{(1)}(h_{k-1})}{15} = \frac{2}{45}\sum_{j=0}^{2^{k-2}} h_k \left[7f(x_{4j}) + 32f(x_{4j+1}) + 12f(x_{4j+2}) + 32f(x_{4j+3}) + 7f(x_{4j+4})\right],$$

which is composite Boole.

Problem 4. See Figure 4.1 for a plot of the integrand in (i) and Figure 4.2 for a plot of the integrand in (ii).

Running Program 4.1 for (i) evaluates the integral to about 0.9064 for both methods, with 513 function calls in Romberg and 1797 in adaptive Simpson's. Clearly, Romberg is much superior to adaptive Simpson. This should be expected because the integrand is quite smooth, and Romberg works well to get highly accurate integrals of smooth functions.

Running Program 4.2 for (ii) evaluates the integral to about 0.7632 for both methods, with 8193 function calls in Romberg and 1405 in adaptive Simpson's. In contrast to the first case, adaptive Simpson is now greatly superior. The plot of the integrand shows a very sharp change right before $x = 1$ where derivatives are very large. Adaptive methods handle this type of integrand more efficiently, by using a small h only near the points $x = 1$, whereas Romberg with a uniform grid is forced to use a small h everywhere.

Problem 5.* First, we write

$$H_2(x) = f(a) + (x-a)f'(a) + (x-a)^2 f[a,a,b] + (x-a)^2(x-b)f[a,a,b,b],$$

where

$$f[a,a,b] = \frac{f[a,b]-f'(a)}{b-a}, \quad f[a,a,b,b] = \frac{f'(a)-2f[a,b]+f'(b)}{(b-a)^2}.$$

Thus

$$\begin{aligned}
H_2(x) &:= f(a) + sf'(a) + s^2\left[\frac{f(b)-f(a)}{h^2} - \frac{f'(a)}{h}\right] + s^2(s-h)\left[\frac{f'(a)+f'(b)}{h^2} - \frac{2(f(b)-f(a))}{h^3}\right] \\
&= \left[1 - \frac{s^2}{h^2} + \frac{2s^2(s-h)}{h^3}\right] f(a) + \left[\frac{s^2}{h^2} - \frac{2s^2(s-h)}{h^3}\right] f(b) + \left[s - \frac{s^2}{h} + \frac{s^2(s-h)}{h^2}\right] f'(a) + \frac{s^2(s-h)}{h^2} f'(b).
\end{aligned}$$

Some algebra will get us our desired form:

$$\begin{aligned}
1 - \frac{s^2}{h^2} + \frac{2s^2(s-h)}{h^3} &= \frac{h^3 - 3hs^2 + 2s^3}{h^3} \\
\frac{s^2}{h^2} - \frac{2s^2(s-h)}{h^3} &= \frac{3hs^2 - 2s^3}{h^3} \\
s - \frac{s^2}{h} + \frac{s^2(s-h)}{h^2} &= \frac{s[h^2 - 2hs + s^2]}{h^2}.
\end{aligned}$$

Let $a = x_k$ and $b = x_{k+1}$, then we're done. For computing the integral, use the obvious substitution $x \mapsto s$ and write

$$\begin{aligned}
\int_{x_k}^{x_{k+1}} H_2(x)\,\mathrm{d}x &= \int_0^h H_2(s + x_0)\,\mathrm{d}s \\
&= \frac{h^4 - h^4/2}{h^3} y_{k+1} + \frac{h^4 - h^4 + h^4/2}{h^3} y_k + \frac{h^4/4 - h^4/3}{h^2} y'_{k+1} + \frac{h^4/4 - 2h^4/3 + h^4/2}{h^2} y'_k \\
&= \frac{h}{2}\left(y_{k+1} + y_k\right) - \frac{h^2}{12}\left(y'_{k+1} - y'_k\right).
\end{aligned}$$

Problem 6.* Using the result from class,

$$\begin{aligned}
P_\ell(0; h_{k-\ell}, \ldots, h_k) &= \frac{hP_{\ell-1}(0; h_{k-\ell+1}, \ldots, h_k) - 2^{-\ell} h P_{\ell-1}(0; h_{k-\ell}, \ldots, h_{k-1})}{h - 2^{-\ell} h} \\
&= \frac{2^\ell P_{\ell-1}(0; h_{k-\ell+1}, \ldots, h_k) - P_{\ell-1}(0; h_{k-\ell}, \ldots, h_{k-1})}{2^\ell - 1},
\end{aligned}$$

where $P_0(0; h_k) = T^{(0)}(h_k)$. We thus see that $P_\ell(0; h_{k-\ell}, \ldots, h_k)$ and $T^{(\ell)}(h_k)$ satisfy identical recursion relations in ℓ and, furthermore, have identical starting values for $\ell = 0$. Thus we conclude that

$$P_\ell(0; h_{k-\ell}, \ldots, h_k) = T^{(\ell)}(h_k)$$

.
This result gives a simple interpretation of the Romberg approximation $T^{(\ell)}(h_\ell)$ as the result of using the polynomial interpolant on the data $(h_0, T^{(0)}(h_0)), ..., (h_\ell, T^{(0)}(h_\ell))$ to extrapolate to $h = 0$.

MATLAB Plots

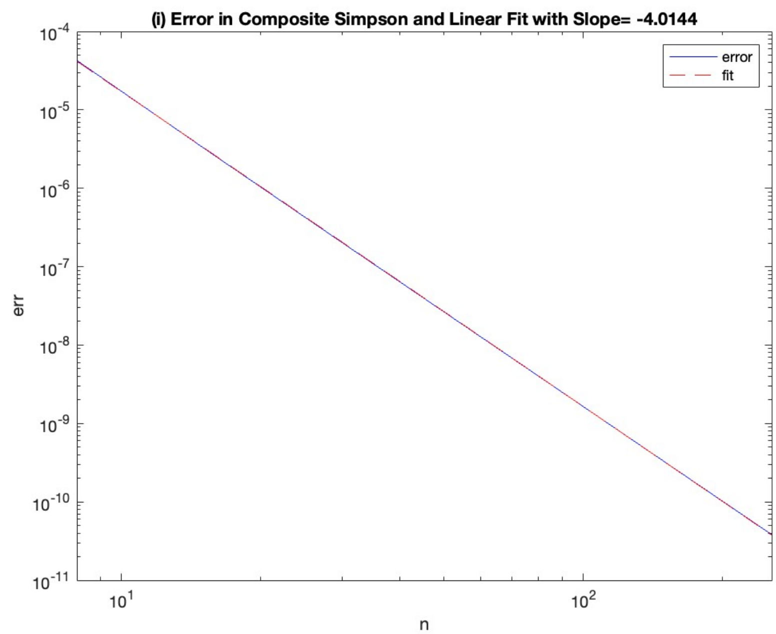

Figure 1.1. Error in composite Simpson and linear fit for integral in (i).

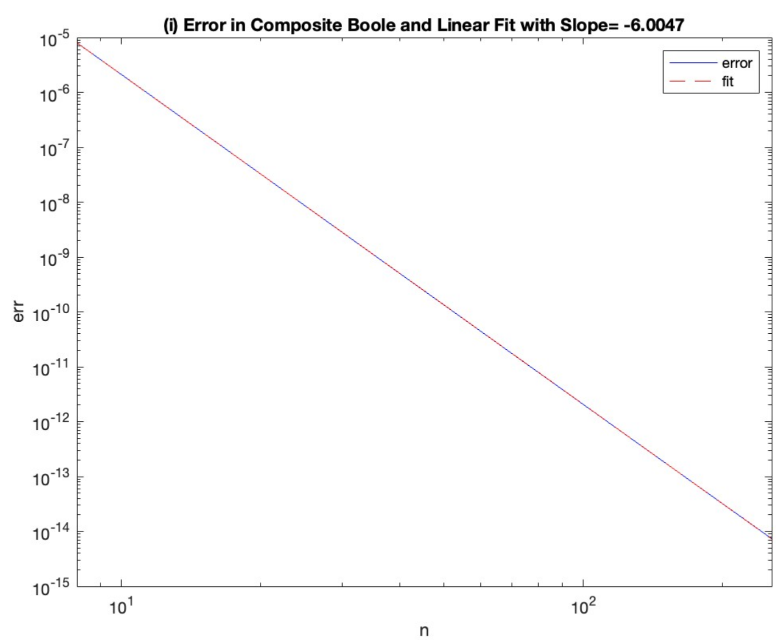

Figure 1.2. Error in composite Boole and linear fit for integral in (i).

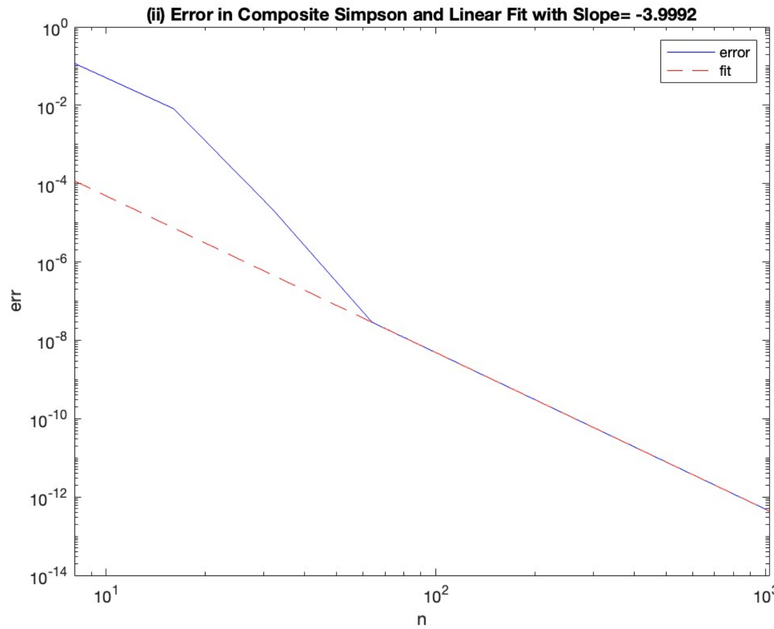

Figure 1.3. Error in composite Simpson and linear fit for integral in (ii).

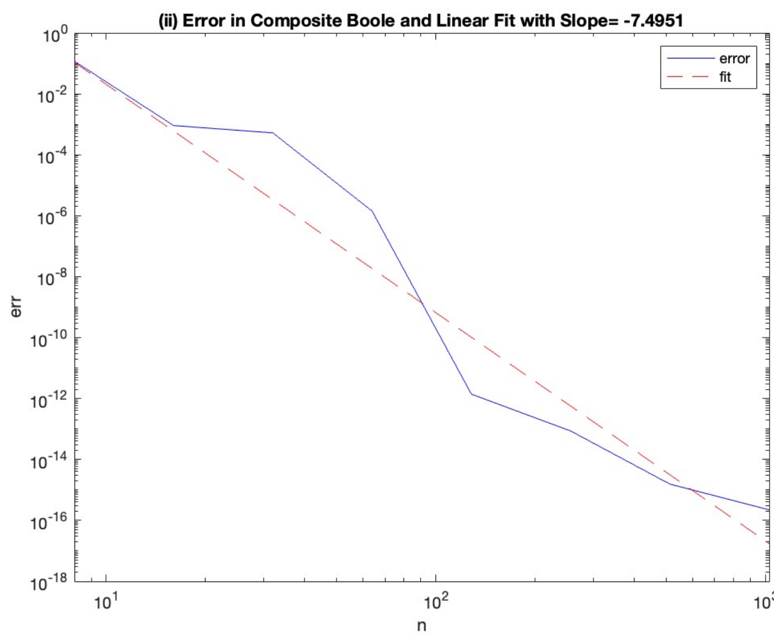

Figure 1.4. Error in composite Boole and linear fit for integral in (ii).

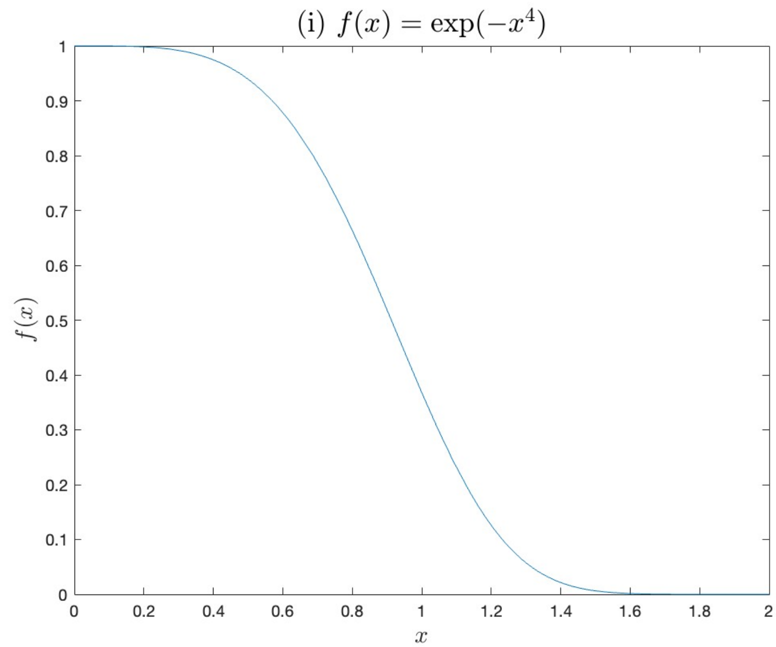

Figure 4.1. Plot of integrand $\exp(-x^4)$ in (i).

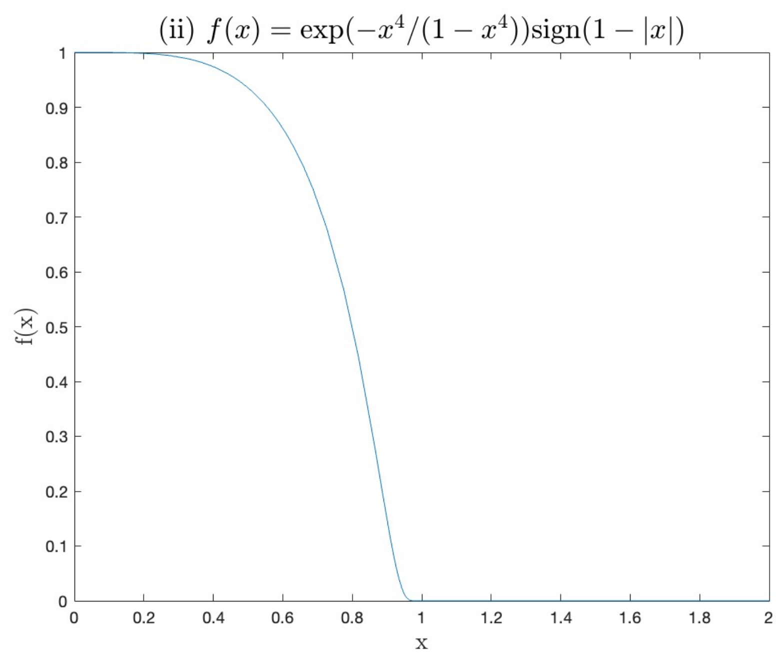

Figure 4.2. Plot of integrand $\exp(-x^4/(1-x^4))\text{sign}(1-|x|)$ in (ii).

MATLAB Code

Program 1.1. Symbolic integration for α_i and $\int_a^b (x-x_0)^2 \prod_{i=1}^4 (x-x_i)\,\mathrm{d}x$ in Problem 1(a), with the transformation $x \mapsto (x-a)/h$. Note this substitution throws a factor of h^5 and h^7 in front respectively, which we omit.

```
syms x

alp(1)=int((x-1)*(x-2)*(x-3)*(x-4),0,4)/factorial(4);
alp(2)=-int(x*(x-2)*(x-3)*(x-4),0,4)/factorial(3);
alp(3)=int(x*(x-1)*(x-3)*(x-4),0,4)/factorial(2)^2;
alp(4)=-int(x*(x-1)*(x-2)*(x-4),0,4)/factorial(3);
alp(5)=int(x*(x-1)*(x-2)*(x-3),0,4)/factorial(4)

bet=int(x^2*(x-1)*(x-2)*(x-3)*(x-4),0,4)/factorial(6)
```

Program 1.2. Implementation of `boolc.m`.

```
function I=boolc(f,a,b,n)

if mod(n,4)~=0
        'n not a multiple of 4 in Boole'
        return
end

h=(b-a)/n;
x1=a+h:4*h:b-3*h;
x2=a+2*h:4*h:b-2*h;
x3=a+3*h:4*h:b-h;
x4=a+4*h:4*h:b-4*h;
I=7*(f(a)+f(b))+32*(sum(f(x1))+sum(f(x3)))+12*sum(f(x2))+14*sum(f(x4));
I=2*h*I/45.;

end
```

Program 1.3. Comparisons of log-log-error for Simpson's and Boole's methods for integral (i).

```
f=@(x) sin(pi*x.^2/2); a=0; b=1;
It=fresnels(1);

for ii=1:6
    n=4*2^ii;
    nn(ii)=n;
    simp(ii)=simpc(f,a,b,n);
    bool(ii)=boolc(f,a,b,n);
end

Is=simp(6)
Ib=bool(6)

errs=abs(simp-It); errb=abs(bool-It);

figure; Ps=polyfit(log(nn),log(errs),1);
ords=-Ps(1)
loglog(nn,errs,'-b',nn,exp(Ps(1)*log(nn)+Ps(2)),'--r')
```

```
legend('error','fit')
xlabel('n'); ylabel('err')
title(['(i) Error in Composite Simpson and Linear Fit with Slope= -',num2str(ords)])

figure; Pb=polyfit(log(nn),log(errb),1);
ordb=-Pb(1)
loglog(nn,errb,'-b',nn,exp(Pb(1)*log(nn)+Pb(2)),'--r')
legend('error','fit')
xlabel('n'); ylabel('err')
title(['(i) Error in Composite Boole and Linear Fit with Slope= -',num2str(ordb)])
```

Program 1.4. Comparisons of log-log-error for Simpson's and Boole's methods for integral (ii).

```
f=@(x) 1./(1+x.^2).^(3/2); a=-4; b=4;
It=2*b/sqrt(1+b^2);

for ii=1:8
    n=4*2^ii;
    nn(ii)=n;
    simp(ii)=simpc(f,a,b,n);
    bool(ii)=boolc(f,a,b,n);
end

Is=simp(8)
Ib=bool(8)

errs=abs(simp-It); errb=abs(bool-It);

figure; Ps=polyfit(log(nn(5:8)),log(errs(5:8)),1);
ords=-Ps(1)
loglog(nn,errs,'-b',nn,exp(Ps(1)*log(nn)+Ps(2)),'--r')
legend('error','fit')
xlabel('n'); ylabel('err')
title(['(ii) Error in Composite Simpson and Linear Fit with Slope= -',num2str(ords)])

figure; Pb=polyfit(log(nn(4:8)),log(errb(4:8)),1);
ordb=-Pb(1)
loglog(nn,errb,'-b',nn,exp(Pb(1)*log(nn)+Pb(2)),'--r')
legend('error','fit')
xlabel('n'); ylabel('err')
title(['(ii) Error in Composite Boole and Linear Fit with Slope= -',num2str(ordb)])
```

Program 4.1. Romberg integration versus adaptive extrapolated Simpson's rule for (i).

```
f=@(x) exp(-x.^4);

figure; fplot(f,[0 2])
xlabel('$x$','FontSize',15,'Interpreter','latex');
ylabel('$f(x)$','FontSize',15,'Interpreter','latex')
title('(i)  $f(x)=\exp(-x^4)$','FontSize',18,'Interpreter','latex')

[IR,fcountR]=romberg(f,0,2,1e-14)
[IQ,fcountQ]=quad(f,0,2,1e-14)
```

Program 4.2. Romberg integration versus adaptive extrapolated Simpson's rule for (ii).

```
figure; fplot(@myfun,[0 2])

xlabel('x','FontSize',15,'Interpreter','latex');
ylabel('f(x)','FontSize',15,'Interpreter','latex')
title('(ii)  $f(x)=\exp(-x^4/(1-x^4)){\rm sign}(1-|x|)$', ...
    'FontSize',18,'Interpreter','latex')

[IR,fcountR]=romberg(@myfun,0,2,1e-14)
[IQ,fcountQ]=quad(@myfun,0,2,1e-14)
```

Program 4.3. Function needed for program 4.2.

```
function y=myfun(x)

n=length(x);
y=zeros(size(x));
for i=1:n
if abs(x(i))<1
   y(i)=exp(-x(i)^4/(1-x(i)^4));
else
   y(i)=0;
end

end
```