APMA 2560 Programming Exercise

ZHANG Zhen

We use the finite element method with continuous piecewise linear elements to solve the following equation:

$$\begin{cases} -u''(x) = \pi^2 \sin(\pi \times x), \\ u(0) = u(1) = 0 \end{cases} \tag{1}$$

We plot two tables below, first one showing the result when (ϕ_i, f) is exactly implemented, second one indicating the result when Simpson's rule is applied in each cell.

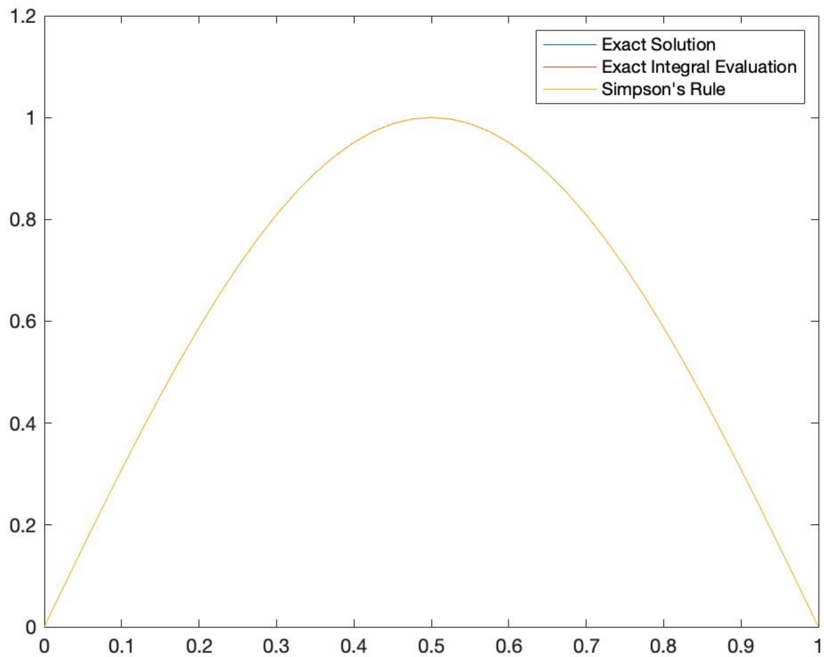
Table 1: (ϕ_i, f) exactly implemented

h	Errors at Nodes	L_2 Error	Numerical Order
$\frac{1}{20}$	2.7601×10^{-15}	0.0016	
$\frac{1}{40}$	5.3792×10^{-15}	3.9812×10^{-4}	1.9994
$\frac{1}{80}$	4.2963×10^{-15}	9.9540×10^{-5}	1.9999
$\frac{1}{160}$	1.3761×10^{-14}	2.4886×10^{-5}	2.0000

Table 2: (ϕ_i, f) using Simpson's Rule

h	Errors at Nodes	L_2 Error	Numerical Order
$\frac{1}{20}$	4.3816×10^{-7}	0.0016	
$\frac{1}{40}$	2.7691×10^{-8}	3.9815×10^{-4}	1.9997
$\frac{1}{80}$	1.7410×10^{-9}	9.9542×10^{-5}	1.9999
$\frac{1}{160}$	1.0915×10^{-10}	2.4886×10^{-5}	2.0000

It can be observed that the L_2 convergence rate is 2 for polynomial of degree 1, which is consistent with what we learned in class. Furthermore, errors at nodes is almost machine ϵ when we compute (ϕ_i, f) exactly. If Simpson's rule is applied, the error at nodes show similar pattern with L_2 error. In general, exact evaluation of integral gives a better result in terms of both L_2 error and error at nodes. The real solution, numerical solution when $h = \frac{1}{160}$ using Simpson's rule, numerical solution when $h = \frac{1}{160}$ using exact evaluation is illustrated in the following graph. We can see the three solutions overlap with each other.



Program

```
%% Preprocessing
clear
clc

%% start solving PDEs!!
ll=0;rr=1;numtrial=2;
error=zeros(numtrial,1);
nodeerror=zeros(numtrial,1);

for i=1:numtrial
    M=10*2^i-1;
    u=zeros(M+2,1);
    u(1)=0;u(M+2)=0;
    u2=u;
    h=(rr-ll)/(M+1);
    A=full(gallery('tridiag',M,-1,2,-1))/h;
    b=zeros(M,1);b2=b;
    x=linspace(ll,rr,M+2);
    for j=1:M
        b(j)=calrhs(x(j),x(j+1),x(j+2),h,1); %Calculate by hand
        b2(j)=calrhs(x(j),x(j+1),x(j+2),h,2); %Use Simpson's rule
    end
    u(2:M+1)=A\b;
```

```
    u2(2:M+1)=A\b2;

    numeval=200000;
    testx=linspace(0,1,numeval);
    l2=0;
    for p=1:numeval
        j=floor(testx(p)/h)+1;
        if (j==M+2)
            j=j-1;
        end
        uu=(u(j+1)-u(j))/(x(j+1)-x(j))*(testx(p)-x(j))+u(j);
        l2=l2+(uexact(testx(p))-uu)^2;
    end
    l2=sqrt(l2/numeval);
    error(i)=l2;
    exactu=uexact(x)';
    nodeerror(i)=sqrt(sum((u-exactu).^2)/(M+2));
end

%% Plot graph and summarize errors
plot(x,u,x,u2,x,exactu)
legend('Exact Solution','Exact Integral Evaluation','Simpson''s Rule')
savefig('result/solution.fig')
for i=2:numtrial
    rate(i-1)=log(error(i-1)/error(i))/log(2); %rate for l2
end
save('data/error_rate.fig','error','nodeerror',rate')

function y=calrhs(x0,x1,x2,h,i)
    if (i==1)
        y=(-x1/h+1)*pi/(-1)*(cos(pi*x1)-cos(pi*x0))+...
        pi^2/h*(x0/pi*cos(pi*x0)-x1/pi*cos(pi*x1)+
        1/pi^2*(sin(pi*x1)-sin(pi*x0)))+...
        (x1/h+1)*pi/(-1)*(cos(pi*x2)-cos(pi*x1))-...
        pi^2/h*(x1/pi*cos(pi*x1)-x2/pi*cos(pi*x2)+
        1/pi^2*(sin(pi*x2)-sin(pi*x1)));
    else
        y=(0.5*f((x0+x1)/2)+f(x1))/3*h+(0.5*f((x1+x2)/2)+f(x1))/3*h;
    end
end

function y=f(x)
    y=pi^2*sin(pi*x);
end

function y=uexact(x)
    y=sin(pi*x);
end
```