微分方程数值解第一次大作业

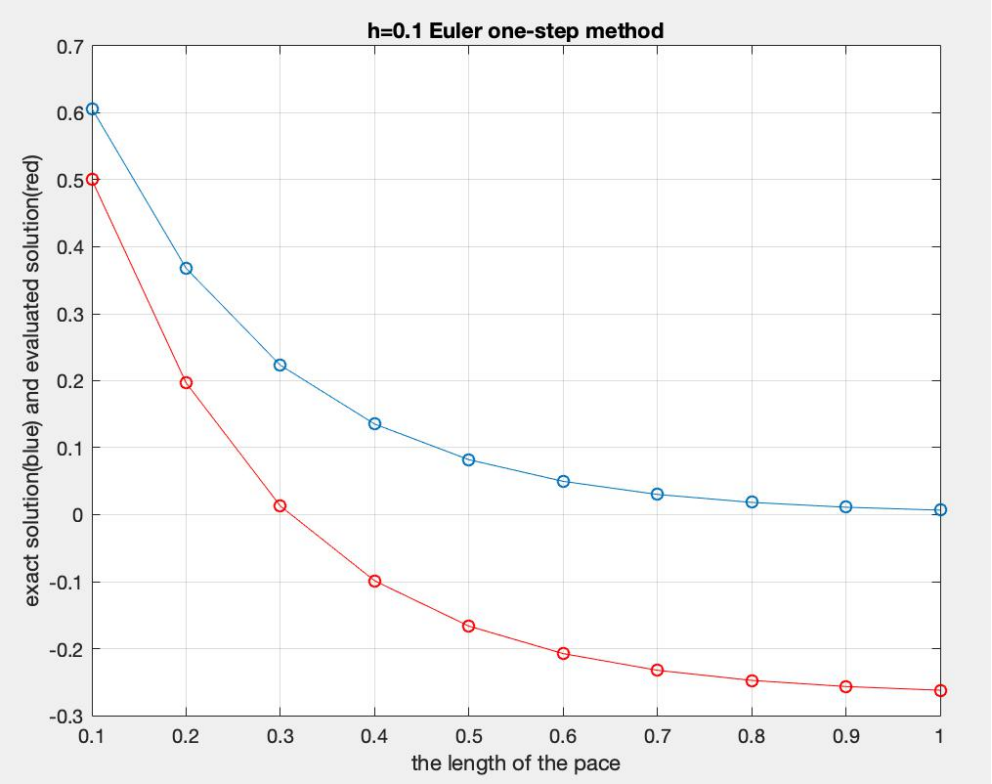
2016141211049
杜鸿宇

第一题：欧拉单步法和多步法

代码：h=0.1,欧拉单步法

```
a=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0];
u=1;
n=0;
b=[];
c=[];
i=1;
while (n>=0)&&(n<=0.9)
    u=u+0.1*(-5)*exp(-5*n);
    n=n+0.1;
    b(1,i)=u;
    c(1,i)=exp(-5*n);
    i=i+1;
end
plot(a,b,'-ro',a,c,'-o')
grid on;
title('h=0.1 Euler one-step method')
xlabel('the length of the pace')
ylabel('exact solution(blue) and evaluated solution(red)')
```

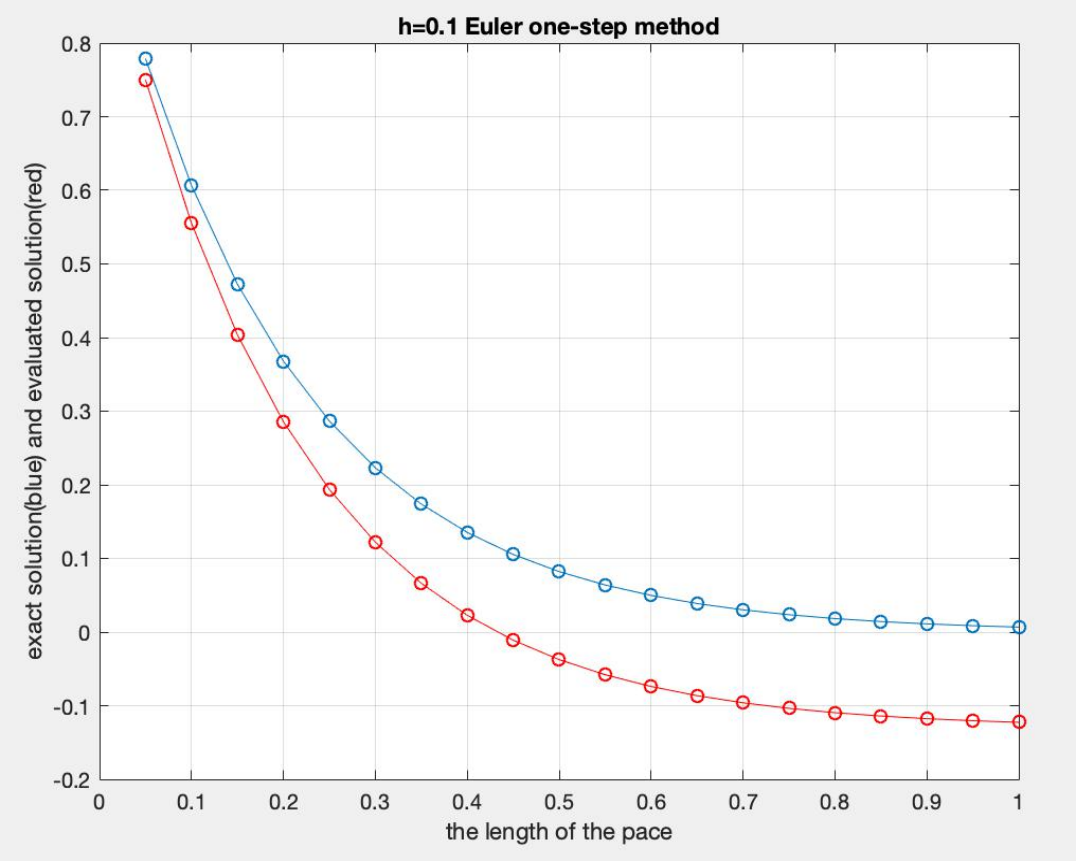
图像：



代码：h=0.05,欧拉单步法

```
a=[];
u=1;
n=0;
b=[];
c=[];
i=1;
while (n>=0)&&(n<=1)
    u=u+0.05*(-5)*exp(-5*n);
    n=n+0.05;
    b(1,i)=u;
    c(1,i)=exp(-5*n);
    a(1,i)=n;
    i=i+1;
end
plot(a,b,'-ro',a,c,'-o')
grid on;
title('h=0.1 Euler one-step method')
xlabel('the length of the pace')
ylabel('exact solution(blue) and evaluated solution(red)')
```

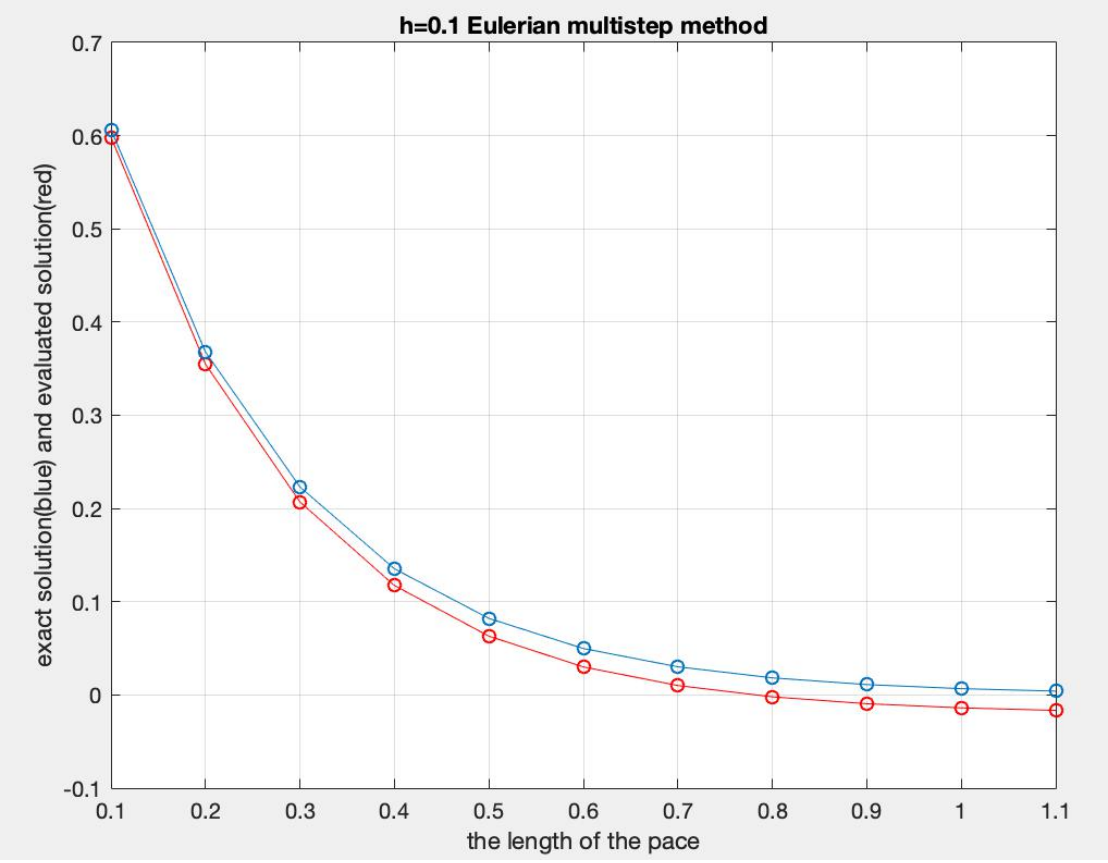
图像：



代码：h=0.01,欧拉多步法

```
a=[];
u=1;
n=0;
b=[];
c=[];
i=1;
while (n>=0)&&(n<=1)
    u=u+0.05*((-5)*exp(-5*n)+(-5)*exp(-5*(n+0.1)));
    n=n+0.1;
    b(1,i)=u;
    c(1,i)=exp(-5*n);
    a(1,i)=n;
    i=i+1;
end
plot(a,b,'-ro',a,c,'-o')
grid on;
title('h=0.1 Eulerian multistep method')
xlabel('the length of the pace')
ylabel('exact solution(blue) and evaluated solution(red)')
```

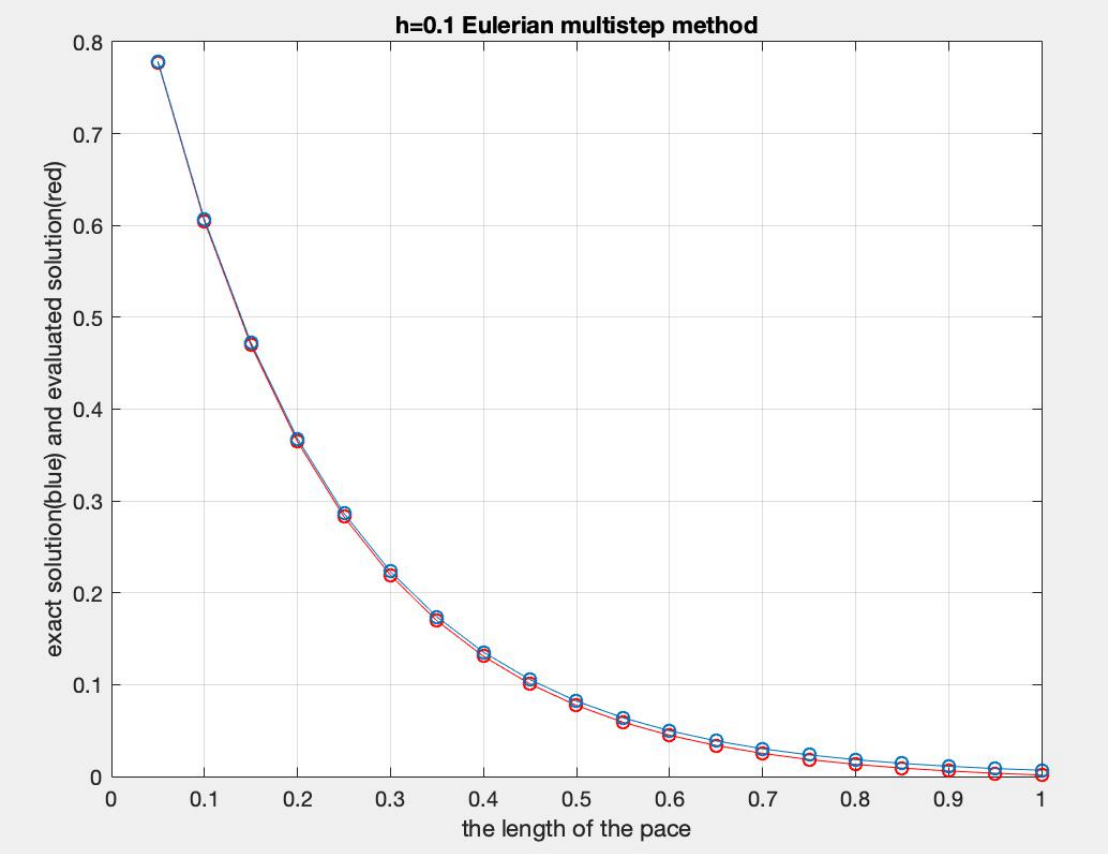
图像：



代码：h=0.05,欧拉多步法

```
a=[];
u=1;
n=0;
b=[];
c=[];
i=1;
while (n>=0)&&(n<=1)
    u=u+0.025*((-5)*exp(-5*n)+(-5)*exp(-5*(n+0.05)));
    n=n+0.05;
    b(1,i)=u;
    c (1,i)=exp(-5*n);
    a(1,i)=n;
    i=i+1;
end
plot(a,b,'-ro',a,c,'-o')
grid on;
title('h=0.1 Eulerian multistep method')
xlabel('the length of the pace')
ylabel('exact solution(blue) and evaluated solution(red)')
```

图像：



小结：从图像看出欧拉多步法明显好于欧拉单步法；步长越短越精确。

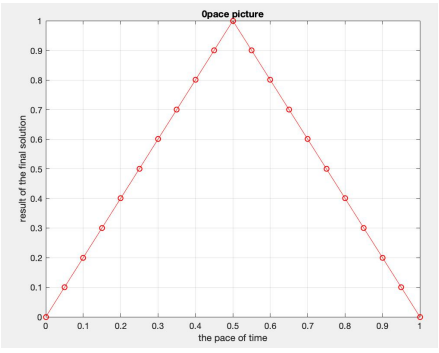
第二题：

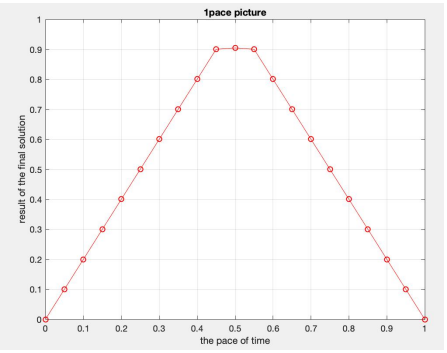
（1）显示算法：

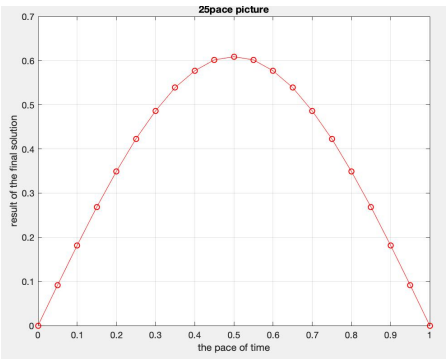
由于是显示解，所以不需要用追赶法。

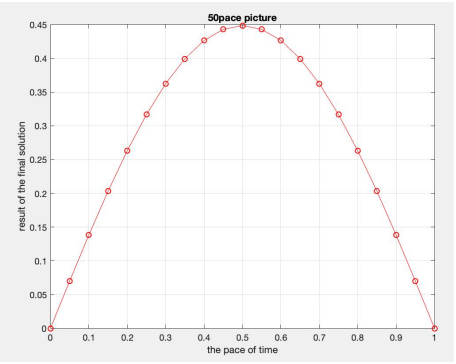
代码：t=0.0012

```
0.05;
t=0.0012;
u=t/(y^2);
%How to achieve the matrix
a=(1-2*u)*ones(1,19);
b=u*ones(1,18)';
D=diag(a);
B=diag(b,1);
C=diag(b,-1);
A=B+C+D;
%The way to calculate the pace of time
x=[];
x(1,1)=0;
x(1,21)=1;
j=2;
for i=0.05:0.05:0.95
   x(1,j)=i;
   j=j+1;
end
%How to use matrix to solve the final solution.
u=[0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1,0.0]'
for i=1:25
   u(2:20,1)=A*u(2:20,1);
end
%Final picture.
plot(x,u,'-ro');
grid on;
xlabel('the pace of time')
ylabel('result of the final solution')
title('25pace picture')
```



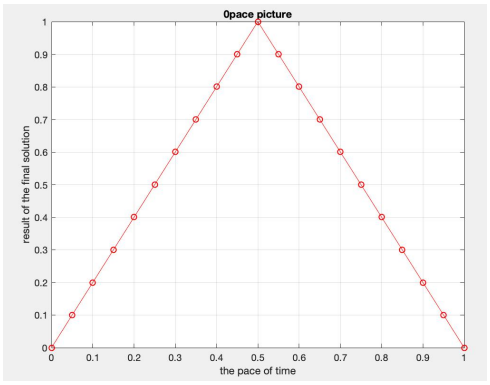


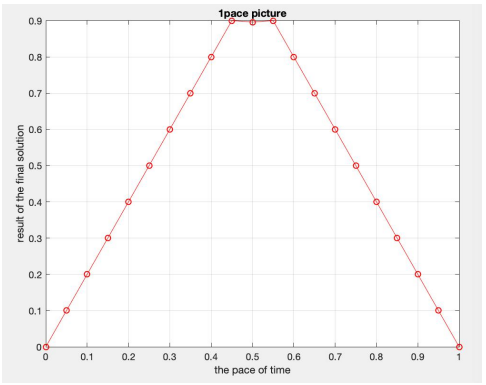


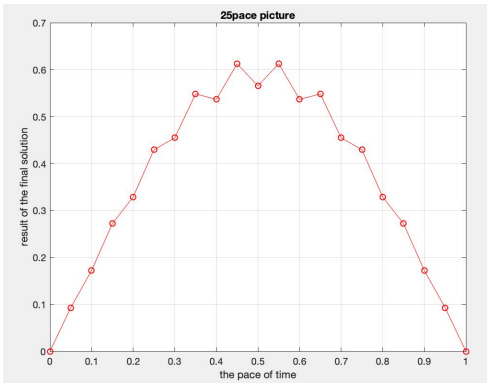


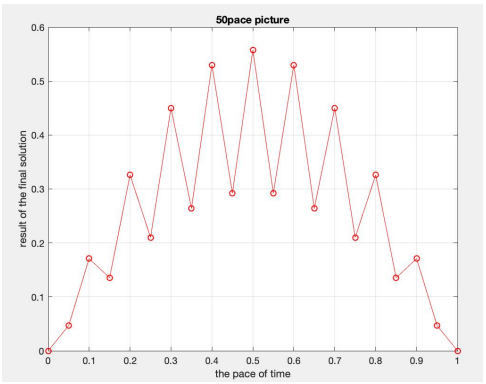
代码：t=0.0013

```
y=0.05;
t=0.0013;
u=t/(y^2);
%How to achieve the matrix
a=(1-2*u)*ones(1,19);
b=u*ones(1,18)';
D=diag(a);
B=diag(b,1);
C=diag(b,-1);
A=B+C+D;
%The way to calculate the pace of time
x=[];
x(1,1)=0;
x(1,21)=1;
j=2;
for i=0.05:0.05:0.95
    x(1,j)=i;
    j=j+1;
end
%How to use matrix to solve the final solution.
u=[0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1,0.0]'
for i=1:50
    u(2:20,1)=A*u(2:20,1);
end
%Final picture.
plot(x,u,'-ro');
grid on;
xlabel('the pace of time')
ylabel('result of the final solution')
title('50pace picture')
```









第三题：

对半隐式格式，即θ=1/2 时，用追赶法所得到的结果。

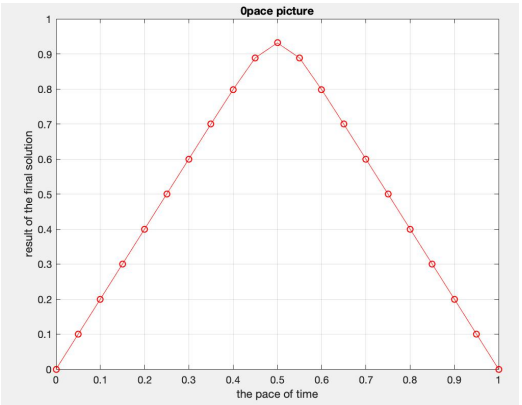
（1）t=0.0012

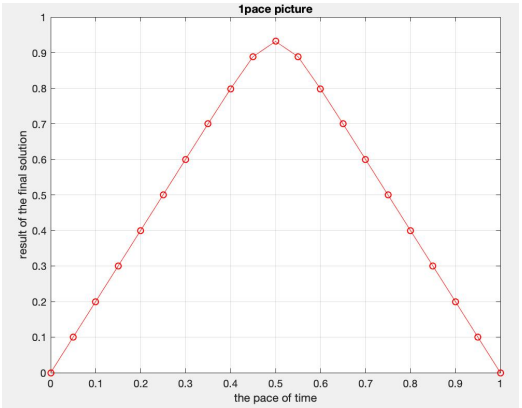
代码：

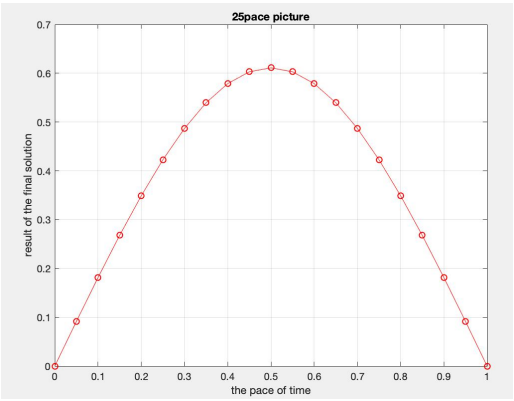
```
y=0.05;
t=0.0012;
u=t/(y^2);
%How to achieve the matrix
a=(1+u)*ones(1,19);
b=-0.5*u*ones(1,18)';
D=diag(a);
B=diag(b,1);
C=diag(b,-1);
A=B+C+D;
%chasing method
k=[0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1,0.0]'
for n=1:50
    U1=[];
    for i=1:19
        U1(i,1)=k(i+1,1)+0.5*u*(k(i,1)-2*k(i+1,1)+k(i+2,1));
    end
    e=[];
    e(1,1)=0;
```

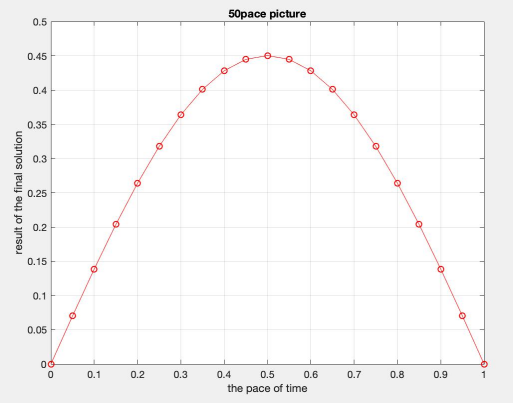
```
        for i=2:20
            e(i,1)=(0.5*u)/((1+u)-0.5*u*e(i-1,1));
        end
        f=[];
        f(1,1)=0;
        for i=2:20
            f(i,1)=(U1(i-1,1)+0.5*u*f(i-1,1))/(1+u-0.5*u*e(i-1,1));
        end
        U=[];
        U(20,1)=f(20,1);
        U(1,1)=0;
        U(21,1)=0;
        for i=19:-1:2
            U(i,1)=((U1(i-1,1)+0.5*u*f(i-1,1))/(1+u-0.5*u*e(i-1,1)))+(0.5*u*U(i+1,1))/(1+u-0.5*u*
e(i-1,1));
        end
        k(2:20,1)=U(2:20,1);
end
%The way to calculate the pace of time
x=[];
x(1,1)=0;
x(1,21)=1;
j=2;
for i=0.05:0.05:0.95
    x(1,j)=i;
     j=j+1;
end
plot(x,U','-ro')
grid on;
xlabel('the pace of time')
ylabel('result of the final solution')
title('50pace picture')
```

图像：









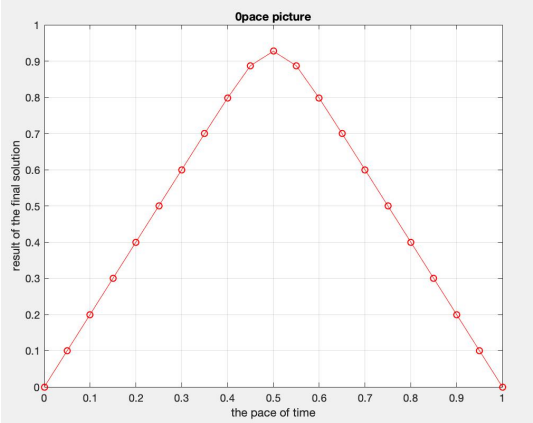
(2) t=0.0013

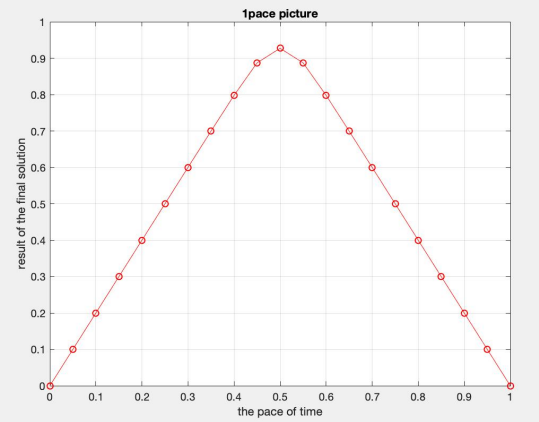
代码:

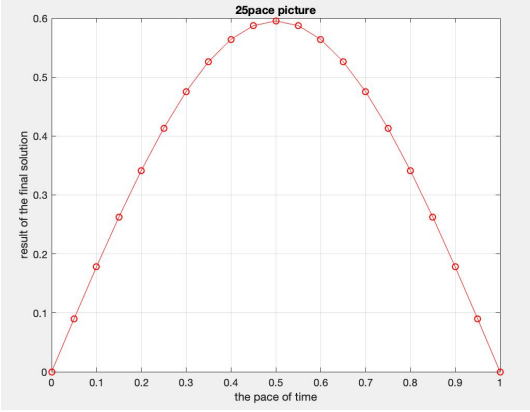
```
y=0.05;
t=0.0013;
u=t/(y^2);
%How to achieve the matrix
a=(1+u)*ones(1,19);
b=-0.5*u*ones(1,18)';
D=diag(a);
B=diag(b,1);
C=diag(b,-1);
A=B+C+D;
%chasing method
k=[0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1,0.0]'
for n=1:50
    U1=[];
    for i=1:19
        U1(i,1)=k(i+1,1)+0.5*u*(k(i,1)-2*k(i+1,1)+k(i+2,1));
    end
    e=[];
    e(1,1)=0;
    for i=2:20
        e(i,1)=(0.5*u)/((1+u)-0.5*u*e(i-1,1));
    end
    f=[];
    f(1,1)=0;
    for i=2:20
        f(i,1)=(U1(i-1,1)+0.5*u*f(i-1,1))/(1+u-0.5*u*e(i-1,1));
    end
    U=[];
    U(20,1)=f(20,1);
    U(1,1)=0;
    U(21,1)=0;
    for i=19:-1:2
```

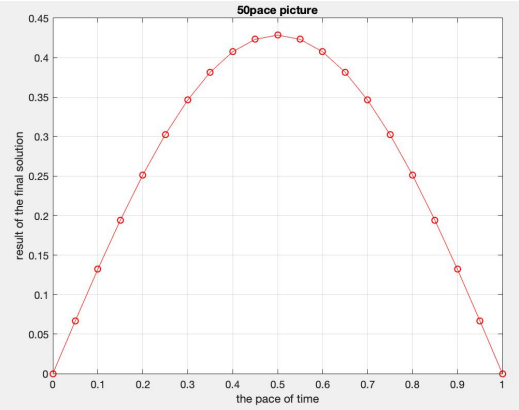
```
                U(i,1)=((U1(i-1,1)+0.5*u*f(i-1,1))/(1+u-0.5*u*e(i-1,1)))+(0.5*u*U(i+1,1))/(1+u-0.5*u*
e(i-1,1));
            end
            k(2:20,1)=U(2:20,1);
end
%The way to calculate the pace of time
x=[];
x(1,1)=0;
x(1,21)=1;
j=2;
for i=0.05:0.05:0.95
    x(1,j)=i;
     j=j+1;
end
plot(x,U','-ro')
grid on;
xlabel('the pace of time')
ylabel('result of the final solution')
title('50pace picture')
```

图像：









小结：t=0.0013 时，在显示格式下非常不稳定，但在隐式格式下就相对较稳定。所以隐式格式更好!