数值分析大作业

程梓峻

2023 年 6 月 27 日

目录

1 Programming problems

All code are aslo in GitHub with the link

1.1 Programming problem 1

The code is as below shown:

```
a=0;
c=2;
Tol=10^(-4);
C=16;
f=@(x) 2;
p=@(x) -x;
q=@(x) 2;
GL=@(x) x;
GR=@(x) x-2;
s=2;
PhiL=@(x) (x-4)/2;
PhiR=@(x) x/2;
ui=@(x) 2*x;
F=@(x) 2-2*x;
A=zeros(1,101);
for i=1:length(A)
    A(i)=(i-1)*2/(length(A)-1);
end
K=10;
Ur=directsolver(A,GL,GR,s,PhiL,PhiR,ui,F,K);
plot(Ur(1,:),Ur(2,:),'k');
```

```
function Ur=directsolver(A,GL,GR,s,PhiL,PhiR,ui,F,K)
         n=length(A)-1;
         sigma=zeros(n*K,1);
         Ur=zeros(2,n*K);
         P=zeros(n*K,n*K);
         for i=1:n
             sigma((i-1)*K+1:i*K)=Ch1([A(i),A(i+1)],K);
         end
         for j=1:n
             for i=1:K
                 t=(j-1)*K+i;
                 E=zeros(K,1);
                 E(i)=1;
                 for T=1:n*K
                     if T<(j-1)*K+1
                        P(T,t)=PhiR(sigma(T))*GR(sigma(t))*...
                            RIntegral1(A(j),A(j),A(j+1),E,K);
                     elseif T>j*K
                            P(T,t)=PhiL(sigma(T))*GL(sigma(t))*...
                                LIntegral1(A(j),A(j+1),A(j+1),E,K);
                     else
                         P(T,t)=PhiR(sigma(T))*GR(sigma(t))*...
```

```
                        RIntegral1(A(j),sigma(T),A(j+1),E,K)...
                    +PhiL(sigma(T))*GL(sigma(t))*...
                    LIntegral1(A(j),sigma(T),A(j+1),E,K);
                end
            end
        end
    end
    P=P+eye(n*K);
    Z=sigma;
    for i=1:n*K
        Z(i)=F(sigma(i));
    end
    Sigma=P\Z;
    JL=zeros(1,n+1);
    JR=zeros(1,n+1);
    EL=zeros(n,K);
    ER=zeros(n,K);
    for j=1:n
        for i=1:K
            EL(j,i)=GL(sigma((j-1)*K+i));
            ER(j,i)=GR(sigma((j-1)*K+i));
        end
    end
    for j=1:n
        JL(j+1)=JL(j)+LIntegral1(A(j),A(j+1),A(j+1),EL(j,:)'.*...
            Sigma((j-1)*K+1:j*K),K);
        JR(n+1-j)=JR(n+2-j)+RIntegral1(A(n+1-j),A(n+1-j),A(n+2-j),...
            ER(n+1-j,:)'.*Sigma((n-j)*K+1:(n-j+1)*K),K);
    end
    for j=1:n
        for i=1:K
            n=(j-1)*K+i;
            Ur(2,n)=ui(sigma(n))+ER(j,i)/s*(JL(j)+ ...
                LIntegral1(A(j),sigma(n),A(j+1),EL(j,:).*...
                Sigma((j-1)*K+1:j*K),K))+...
                EL(j,i)/s*(JR(j+1)+RIntegral1(A(j),sigma(n),...
                A(j+1),ER(j,:).*Sigma((j-1)*K+1:j*K),K));
        end
    end
    Ur(1,:)=sigma';
end
```

```
function I=LIntegral1(u,x,v,Z,K)
    I=0;
```

```
        CF=zeros(K,K);
        Theta=zeros(1,K);
        for i=1:K
            Theta(i)=(2*K-2*i+1)*pi/(2*K);
        end
        for i=1:K
            for j=1:K
                CF(i,j)=2*cos((i-1)*Theta(j))/K;
            end
        end
        CF(1,:)=CF(1,:)/2;
        L=zeros(K,K);
        for i=3:K
            L(1,i)=(-1)^(i-1)*(1/i-1/(i-2))/2;
            L(i-1,i-2)=1/(2*(i-2));
            L(i-1,i)=-1/(2*(i-2));
        end
        L(2,1)=1;
        L(1,1)=1;
        L(1,2)=-1/4;
        L(K,K-1)=1/(2*(K-1));
        L=(v-u)/2*L;
        J=L*CF*Z;
        for i=1:K
            I=I+J(i)*cos((i-1)*acos((2*x-u-v)/(v-u)));
        end
end
```

```
function I=RIntegral1(u,x,v,Z,K)
        I=0;
        CF=zeros(K,K);
        Theta=zeros(1,K);
        for i=1:K
            Theta(i)=(2*K-2*i+1)*pi/(2*K);
        end
        for i=1:K
            for j=1:K
                CF(i,j)=2*cos((i-1)*Theta(j))/K;
            end
        end
        CF(1,:)=CF(1,:)/2;
        R=zeros(K,K);
        for i=3:K
            R(1,i)=(1/i-1/(i-2))/2;
```

```
            R(i-1,i-2)=-1/(2*(i-2));
            R(i-1,i)=1/(2*(i-2));
        end
        R(2,1)=-1;
        R(1,1)=1;
        R(1,2)=1/4;
        R(K,K-1)=-1/(2*(K-1));
        R=(v-u)/2*R;
        J=R*CF*Z;
        for i=1:K
            I=I+J(i)*cos((i-1)*acos((2*x-u-v)/(v-u)));
        end
end
```

```
function Y=Ch1(X,K)
        Y=zeros(K,1);
        for i=1:K
            Y(i)=(X(2)-X(1))/2*cos((2*K-2*i+1)*pi/(2*K))+(X(2)+X(1))/2;
        end
end
```

In these codes,the functions f, p, q, correspond to the problem $u^{(2)} + pu^{(1)} + qu = f$,and functions $GL, GR, PhiL, PhiR, ui, F$ correspond to $g_l, g_r, \psi_l, \psi_r, u_i, \widetilde{f}$ (in the paper of Lee and Greengard) respectively.s is just the constant $g_l g_r^{(1)} - g_r g_l^{(1)}$.And the code gives an example of these functions and variables at the beginning.The main part of the code is function (in matlab) "directsolver".

1.2 Programming problem 2

The code is as below shown:

```
a=-1;
c=1;
Tol=10^(-5);
NN=5;
r=10^(-NN);
C=24;
```

```
f=@(x) 0;
p=@(x) 2*x/r;
q=@(x) 0;
GL=@(x) x+1;
GR=@(x) x-1;
s=2;
PhiL=@(x) x/r;
PhiR=@(x) x/r;
ui=@(x) x;
F=@(x) -2*x/r;
[Ur,eI]=linearsolver(a,c,Tol,C,GL,GR,s,PhiL,PhiR,ui,F,r);
plot(Ur(1,:),Ur(2,:),'k');
```

```
function [Ur,In]=linearsolver(a,c,Tol,C,GL,GR,s,PhiL,PhiR,ui,F,r)
          interval=struct('leftchild',0,'rightchild',0,'parent',0,...
              'content',0,'exist',1);
          interval.content=struct('Interval',[0,0],'alphaL',0,'alphaR',...
              0,'betaL',0,'betaR',0, '∆L',0,'∆R',0,'niuL',0, ...
          'niuR',0,'niu',0);
          interval(1).content.Interval=[a,c];
          interval(1).content.niuL=0;
          interval(1).content.niuR=0;
          interval(1).content.niu=1;
          N=1;
          Er=1;
          Mat2=zeros(4,3);
          Mat2(1,3)=0.5;
          while Er>Tol
                S=zeros(1,N);
                for i=1:N
                    if interval(i).leftchild==0 && interval(i).exist==1
                       S(i)=Evamonitor(interval,i,PhiL,PhiR,GL,GR,F);
                    end
                end
                M=N;
                for i=1:N
                    if interval(i).leftchild==0 && interval(i).exist==1
                       if S(i)≥max(S)/C
                          x=interval(i).content.Interval(1);
                          y=interval(i).content.Interval(2);
                          M=M+1;
                          interval(M)=struct('leftchild',0,'rightchild',...
                              0,'parent',i,'content',0,'exist',1);
                          interval(M).content=struct('Interval',[x,...
```

```
                              (x+y)/2],'alphaL',0, ...
                          'alphaR',0,'betaL',0,'betaR',0, 'ΔL',0,...
                          'ΔR',0,'niuL',0, ...
                          'niuR',0,'niu',0);
                          interval(i).leftchild=M;
                          M=M+1;
                          interval(M)=struct('leftchild',0,'rightchild',...
                              0,'parent',i,'content',0,'exist',1);
                          interval(M).content=struct('Interval',[(x+y)/2,...
                              y],'alphaL',0, ...
                          'alphaR',0,'betaL',0,'betaR',0, 'ΔL',0,...
                          'ΔR',0,'niuL',0,'niuR',0,'niu',0);
                          interval(i).rightchild=M;
                       elseif interval(i).parent>0
                              n=interval(i).parent;
                              m=interval(n).leftchild;
                              if S(m)+S(m+1)<max(S)/(2^10) && ...
                                  interval(2*m+1-i).leftchild==0
                                 interval(m).exist=0;
                                 interval(m+1).exist=0;
                                 interval(n).leftchild=0;
                                 interval(n).rightchild=0;
                              end
                        end
                    end
                end
                for i=1:M
                    if interval(i).leftchild==0 && interval(i).exist==1
                       X=ABD(interval,i,F,PhiL,PhiR,GL,GR);
                       interval(i).content.alphaL=X(1);
                       interval(i).content.alphaR=X(2);
                       interval(i).content.betaL=X(3);
                       interval(i).content.betaR=X(4);
                       interval(i).content.ΔL=X(5);
                       interval(i).content.ΔR=X(6);
                    end
                end
                for j=1:M
                    i=M+1-j;
                    n=interval(i).parent;
                    if n>0 && interval(i).exist==1
                       m=interval(n).leftchild;
                       U(1)=interval(m).content.alphaL;
                       U(2)=interval(m).content.alphaR;
                       U(3)=interval(m).content.betaL;
                       U(4)=interval(m).content.betaR;
```

```
               U(5)=interval(m).content.ΔL;
               U(6)=interval(m).content.ΔR;
               V(1)=interval(m+1).content.alphaL;
               V(2)=interval(m+1).content.alphaR;
               V(3)=interval(m+1).content.betaL;
               V(4)=interval(m+1).content.betaR;
               V(5)=interval(m+1).content.ΔL;
               V(6)=interval(m+1).content.ΔR;
               Delta=1-U(3)*V(2);
               interval(n).content.alphaL=(1-V(1))*(U(1)+...
                   Delta-1)/Delta+V(1);
               interval(n).content.alphaR=V(2)*(1-U(4))*...
                   (1-U(1))/Delta+U(2);
               interval(n).content.betaL=U(3)*(1-V(4))*...
                   (1-V(1))/Delta+V(3);
               interval(n).content.betaR=(1-U(4))*(V(4)+...
                   Delta-1)/Delta+U(4);
               interval(n).content.ΔL=(1-V(1))*U(5)/...
                   Delta+V(5)+(V(1)-1)*U(3)*V(6)/Delta;
               interval(n).content.ΔR=(1-U(4))*V(6)/...
                   Delta+U(6)+(U(4)-1)*V(2)*U(5)/Delta;
            end
        end
        for i=1:M
            n=interval(i).leftchild;
            if n>0
               x=interval(i).content.niuL;
               y=interval(i).content.niuR;
               z=interval(i).content.niu;
               u1=interval(n).content.alphaL;
               u3=interval(n).content.betaL;
               u5=interval(n).content.ΔL;
               v4=interval(n+1).content.betaR;
               v2=interval(n+1).content.alphaR;
               v6=interval(n+1).content.ΔR;
               interval(n).content.niuL=x;
               interval(n+1).content.niuR=y;
               interval(n).content.niu=z;
               interval(n+1).content.niu=z;
               NIU=[1,v2;u3,1]\[y*(1-v4)-z*v6;x*(1-u1)-z*u5];
               interval(n).content.niuR=NIU(1);
               interval(n+1).content.niuL=NIU(2);
            end
        end
        X=zeros(1,M);
        j=0;
```

```
            for i=1:M
                if interval(i).leftchild==0 && interval(i).exist==1
                   X(j+1)=i;
                   j=j+1;
                end
            end
            Num=j;
            Mat=[a-1;0;0;0];
            for i=1:Num
                K=Ch(interval(X(i)).content.Interval);
                A=P(interval,X(i),PhiL,PhiR,GL,GR);
                Z1=zeros(10,1);
                Z2=zeros(10,1);
                Z3=zeros(10,1);
                for j=1:10
                    Z1(j)=F(K(j));
                    Z2(j)=PhiL(K(j));
                    Z3(j)=PhiR(K(j));
                end
                sigma=A\Z1+A\Z2*interval(X(i)).content.niuL+...
                    A\Z3*interval(X(i)).content.niuR;
                Mat1=zeros(4,10);
                for j=1:10
                    Mat1(1,j)=K(j);
                    Mat1(2,j)=sigma(j);
                    Mat1(3,j)=X(i);
                end
                Mat=[Mat Mat1];
            end
            [¬,idx]=sort(Mat(1,:));
            Mat=Mat(:,idx);
            JL=zeros(1,Num+1);
            JR=zeros(1,Num+1);
            for i=1:Num
                n=Mat(3,10*i);
                a1=interval(n).content.alphaL;
                b1=interval(n).content.betaL;
                d1=interval(n).content.ΔL;
                n1=interval(n).content.niuL;
                n2=interval(n).content.niuR;
                JL(i+1)=JL(i)+d1+n1*a1+n2*b1;
                m=Mat(3,10*(Num+1-i));
                a2=interval(m).content.alphaR;
                b2=interval(m).content.betaR;
                d2=interval(m).content.ΔR;
                m1=interval(m).content.niuL;
```

```
                    m2=interval(m).content.niuR;
                    JR(Num+1-i)=JR(Num+2-i)+d2+m1*a2+m2*b2;
                end
                for i=1:Num
                    Y=interval(Mat(3,10*i)).content.Interval;
                    K=Ch(Y);
                    Z1=zeros(10,1);
                    Z2=zeros(10,1);
                    for t=1:10
                        Z1(t)=GL(K(t));
                        Z2(t)=GR(K(t));
                    end
                    for j=1:10
                        ur=ui(K(j))+GR(K(j))/s*(JL(i)+ ...
                        LIntegral(Y(1),K(j),Y(2),Z1.*(Mat(2,10*(i-1)+...
                        2:10*i+1))'))+GL(K(j))/s*(JR(i+1)+ ...
                        RIntegral(Y(1),K(j),Y(2),Z2.*(Mat(2,10*(i-1)+...
                        2:10*i+1))'));
                        Mat(4,10*(i-1)+1+j)=ur;
                    end
                end
                Er=Error(Mat,Mat2);
                Mat2=Mat;
                nn=size(Mat,2);
                N=M;
          end
          Ur=zeros(2,nn-1);
          Ur(1,:)=Mat(1,2:nn);
          Ur(2,:)=Mat(4,2:nn);
          w=@(x) 2*pi^(-0.5)*exp(-x.^2);
          In=0;
          for i=1:Num
              Y=interval(Mat(3,10*i)).content.Interval;
              K=Ch(Y);
              Z=zeros(10,1);
              for j=1:10
                  G=integral(@(x) w(x),0,K(j)*r^(-0.5))/...
                      integral(@(x) w(x),0,r^(-0.5));
                  Z(j)=(Mat(4,10*(i-1)+j+1)-G)^2;
              end
              In=In+LIntegral(Y(1),Y(2),Y(2),Z);
         end
end
```

```
function h=Evamonitor(str,n,phiL,phiR,gL,gR,F1)
          A=P(str,n,phiL,phiR,gL,gR);
          K=Ch(str(n).content.Interval);
          Z1=zeros(10,1);
          for i=1:10
              Z1(i)=F1(K(i));
          end
          Theta=zeros(1,10);
          for i=1:10
              Theta(i)=(20-2*i+1)*pi/20;
          end
          CF=zeros(10,10);
          for i=1:10
              for j=1:10
                  CF(i,j)=cos((i-1)*Theta(j))/5;
              end
          end
          CF(1,:)=CF(1,:)/2;
          H=CF*(A\Z1);
          h=abs(H(9))+abs(H(10)-H(8));
end
```

```
function M=P(str,n,phiL,phiR,gL,gR)
          X=str(n).content.Interval;
          K=Ch(X);
          Theta=zeros(1,10);
          for i=1:10
              Theta(i)=(20-2*i+1)*pi/20;
          end
          CF=zeros(10,10);
          for i=1:10
              for j=1:10
                  CF(i,j)=cos((i-1)*Theta(j))/5;
              end
          end
          CB=5*CF';
          CF(1,:)=CF(1,:)/2;
          L=zeros(10,10);
          for i=3:10
              L(1,i)=(-1)^(i-1)*(1/i-1/(i-2))/2;
              L(i-1,i-2)=1/(2*(i-2));
              L(i-1,i)=-1/(2*(i-2));
          end
          L(2,1)=1;
```

```
        L(1,1)=1;
        L(1,2)=-1/4;
        L(10,9)=1/18;
        L=(X(2)-X(1))/2*L;
        R=zeros(10,10);
        for i=3:10
            R(1,i)=(1/i-1/(i-2))/2;
            R(i-1,i-2)=-1/(2*(i-2));
            R(i-1,i)=1/(2*(i-2));
        end
        R(2,1)=-1;
        R(1,1)=1;
        R(1,2)=1/4;
        R(10,9)=-1/18;
        R=(X(2)-X(1))/2*R;
        IL=CB*L*CF;
        IR=CB*R*CF;
        DPL=eye(10);
        DPR=eye(10);
        DGL=eye(10);
        DGR=eye(10);
        for i=1:10
            DPR(i,i)=phiR(K(i));
            DGL(i,i)=gL(K(i));
            DGR(i,i)=gR(K(i));
            DPL(i,i)=phiL(K(i));
        end
        M=eye(10)+DPL*IL*DGL+DPR*IR*DGR;
end
```

```
function K=Ch(X)
        K=zeros(1,10);
        for i=1:10
            K(i)=(X(2)-X(1))/2*cos((20-2*i+1)*pi/20)+(X(2)+X(1))/2;
        end
end
```

```
function Z=ABD(str,n,F1,phiL,phiR,gL,gR)
        X=str(n).content.Interval;
        K=Ch(X);
        Z1=zeros(10,1);
        Z2=zeros(10,1);
```

```
        Z3=zeros(10,1);
        Z4=zeros(10,1);
        Z5=zeros(10,1);
        for i=1:10
            Z1(i)=F1(K(i));
            Z2(i)=phiL(K(i));
            Z3(i)=phiR(K(i));
            Z4(i)=gL(K(i));
            Z5(i)=gR(K(i));
        end
        A=P(str,n,phiL,phiR,gL,gR);
        Z(1)=LIntegral(X(1),X(2),X(2),Z4.*(A\Z2));
        Z(2)=LIntegral(X(1),X(2),X(2),Z5.*(A\Z2));
        Z(3)=LIntegral(X(1),X(2),X(2),Z4.*(A\Z3));
        Z(4)=LIntegral(X(1),X(2),X(2),Z5.*(A\Z3));
        Z(5)=LIntegral(X(1),X(2),X(2),Z4.*(A\Z1));
        Z(6)=LIntegral(X(1),X(2),X(2),Z5.*(A\Z1));
end
```

```
function er=Error(A,B)
        n=size(A,2);
        m=size(B,2);
        e=0;
        f=0;
        for i=3:m
            for j=3:n
                if A(1,j)>=B(1,i)
                   a=(B(1,i)-A(1,j-1))/(A(1,j)-A(1,j-1));
                   e=max(e,abs(a*A(4,j)+(1-a)*A(4,j-1)-B(4,i)));
                   f=max(f,abs(a*A(4,j)+(1-a)*A(4,j-1)+B(4,i)));
                   break
                end
            end
        end
        er=e/f;
end
```

```
function I=LIntegral(u,x,v,Z)
        I=0;
        CF=zeros(10,10);
        Theta=zeros(1,10);
        for i=1:10
```

```
            Theta(i)=(20-2*i+1)*pi/20;
        end
        for i=1:10
            for j=1:10
                CF(i,j)=cos((i-1)*Theta(j))/5;
            end
        end
        CF(1,:)=CF(1,:)/2;
        L=zeros(10,10);
        for i=3:10
            L(1,i)=(-1)^(i-1)*(1/i-1/(i-2))/2;
            L(i-1,i-2)=1/(2*(i-2));
            L(i-1,i)=-1/(2*(i-2));
        end
        L(2,1)=1;
        L(1,1)=1;
        L(1,2)=-1/4;
        L(10,9)=1/18;
        L=(v-u)/2*L;
        J=L*CF*Z;
        for i=1:10
            I=I+J(i)*cos((i-1)*acos((2*x-u-v)/(v-u)));
        end
end
```

```
function I=RIntegral(u,x,v,Z)
        I=0;
        CF=zeros(10,10);
        Theta=zeros(1,10);
        for i=1:10
            Theta(i)=(20-2*i+1)*pi/20;
        end
        for i=1:10
            for j=1:10
                CF(i,j)=2*cos((i-1)*Theta(j))/10;
            end
        end
        CF(1,:)=CF(1,:)/2;
        R=zeros(10,10);
        for i=3:10
            R(1,i)=(1/i-1/(i-2))/2;
            R(i-1,i-2)=-1/(2*(i-2));
            R(i-1,i)=1/(2*(i-2));
        end
```

```
            R(2,1)=-1;
            R(1,1)=1;
            R(1,2)=1/4;
            R(10,9)=-1/18;
            R=(v-u)/2*R;
            J=R*CF*Z;
            for i=1:10
                I=I+J(i)*cos((i-1)*acos((2*x-u-v)/(v-u)));
            end
end
```

In these codes,similar to problem 1,the functions and variables at the beginning just give an example.And the main part of the code is function (in matlab) “linearsolver”.I use struct in matlab to build binary tree structure.And the number of chebyshev points of each leaf interval is always 10.

1.3 Programming problem 3

The code is as below shown:

```
Fun=struct('U',@(x) 0);
dFun=struct('V',@(x) 0);
ddFun=struct('R',@(x) 0);
Fun(1)=struct('U',@(x) (x+2*exp(2)-2*exp(1))/(2*exp(2)-exp(1)));
dFun(1)=struct('V',@(x) 1/(2*exp(2)-exp(1)));
ddFun(1)=struct('R',@(x) 0);
f=@(x,y,z) z/(x*(y+1)^2)-y/(x^2*(y+1));
f2=@(x,y,z) -y/(x^2*(y+1)^2)-2*z/(x*(y+1)^3);
f3=@(x,y,z) 1/(x*(y+1)^2);
X=exp(1):(2*exp(2)-exp(1))/100:2*exp(2);
Y=X;
tol=10^(-4);
er=1;
N=1;
a=exp(1);
c=2*exp(2);
C=16;
while er>tol
      e=0;
      f1=0;
```

```
        p=@(x) -f3(x,Fun(N).U(x),dFun(N).V(x));
        q=@(x) -f2(x,Fun(N).U(x),dFun(N).V(x));
        F=@(x) -ddFun(N).R(x)+f(x,Fun(N).U(x),dFun(N).V(x));
        GL=@(x) x-exp(1);
        GR=@(x) x-2*exp(2);
        dGL=@(x) 1;
        dGR=@(x) 1;
        s=2*exp(2)-exp(1);
        PhiL=@(x) (p(x)+q(x)*GR(x))/s;
        PhiR=@(x) (p(x)+q(x)*GL(x))/s;
        [Ur,JL,JR,Leftendpoint]=linearsolver1(a,c,tol,C,GL,GR,s,PhiL,PhiR,F);
        d2v=@(x) ...
            f3(x,Fun(N).U(x),dFun(N).V(x))*dv(x,Ur,JL,JR,Leftendpoint...
            ,GL,GR,dGL,dGR,s,c)+ ...
                f2(x,Fun(N).U(x),dFun(N).V(x))*v(x,Ur,JL,...
            JR,Leftendpoint,GL,GR,s,c)+F(x);
        Fun(N+1)=struct('U',@(x) 0);
        Fun(N+1).U=@(x) Fun(N).U(x)+v(x,Ur,JL,JR,Leftendpoint,GL,GR,s,c);
        dFun(N+1)=struct('V',@(x) 0);
        dFun(N+1).V=@(x) dFun(N).V(x)+dv(x,Ur,JL,JR,Leftendpoint,GL,GR,...
            dGL,dGR,s,c);
        ddFun(N+1)=struct('R',@(x) 0);
        ddFun(N+1).R=@(x) ddFun(N).R(x)+d2v(x);
        for i=1:101
            e=max(e,abs(Fun(N+1).U(X(i))-Fun(N).U(X(i))));
            f1=max(f1,abs(Fun(N+1).U(X(i))+Fun(N).U(X(i))));
        end
        er=e/f1;
        N=N+1;
        if N==5
           break
        end
end
for j=1:4
    for i=1:101
        Y(j,i)=Fun(j).U(X(i));
    end
end
for i=1:101
    Y(3,i)=lambertw(X(i));
end
ER=0;
for i=1:101
    ER=max(ER,abs(Y(4,i)-lambertw(X(i))));
end
plot (X,Y(1,:),'k',X,Y(2,:),'r',X,Y(3,:),'g')
```

```

function Δu=v(x,Ur,JL,JR,Leftendpoint,GL,GR,s,c)
         n=length(Leftendpoint)+1;
         YY=zeros(1,n);
         for i=1:n-1
             YY(i)=Leftendpoint(i);
         end
         YY(n)=c;
         for i=1:n-1
             if x≥YY(i) && YY(i+1)≥x
                K=Ch([YY(i),YY(i+1)]);
                Z1=zeros(10,1);
                Z2=zeros(10,1);
                for j=1:10
                    Z1(j)=GL(K(j));
                    Z2(j)=GR(K(j));
                end
                Δu=GR(x)/s*(JL(i)+LIntegral(YY(i),x,YY(i+1),Z1.*...
                    (Ur(2,10*(i-1)+1:10*i))'))+GL(x)/s*(JR(i+1)+ ...
                       RIntegral(YY(i),x,YY(i+1),Z2.*(Ur(2,10*(i-1)+...
                       1:10*i))'));
                break
             end
         end
end

function Δdu=dv(x,Ur,JL,JR,Leftendpoint,GL,GR,dGL,dGR,s,c)
         n=length(Leftendpoint)+1;
         YY=zeros(1,n);
         for i=1:n-1
             YY(i)=Leftendpoint(i);
         end
         YY(n)=c;
         for i=1:n-1
             if x≥YY(i) && YY(i+1)≥x
                K=Ch([YY(i),YY(i+1)]);
                Z1=zeros(10,1);
                Z2=zeros(10,1);
                for j=1:10
                    Z1(j)=GL(K(j));
                    Z2(j)=GR(K(j));
                end
                Δdu=dGR(x)/s*(JL(i)+LIntegral(YY(i),x,YY(i+1),Z1.*...
                    (Ur(2,10*(i-1)+1:10*i))'))+dGL(x)/s*(JR(i+1)+ ...
                       RIntegral(YY(i),x,YY(i+1),Z2.*(Ur(2,10*(i-1)+...
                       1:10*i))'));
```

```
                break
            end
        end
end
```

```
function ...
    [Ur,JL,JR,Leftendpoint]=linearsolver1(a,c,Tol,C,GL,GR,s,PhiL,PhiR,F)
        interval=struct('leftchild',0,'rightchild',0,'parent',0,...
            'content',0,'exist',1);
        interval.content=struct('Interval',[0,0],'alphaL',0,'alphaR',...
            0,'betaL',0,'betaR',0, '∆L',0,'∆R',0,'niuL',0, ...
        'niuR',0,'niu',0);
        interval(1).content.Interval=[a,c];
        interval(1).content.niuL=0;
        interval(1).content.niuR=0;
        interval(1).content.niu=1;
        N=1;
        Er=1;
        Mat2=zeros(4,3);
        Mat2(1,3)=0.5;
        while Er>Tol
              S=zeros(1,N);
              for i=1:N
                  if interval(i).leftchild==0 && interval(i).exist==1
                     S(i)=Evamonitor(interval,i,PhiL,PhiR,GL,GR,F);
                  end
              end
              M=N;
              for i=1:N
                  if interval(i).leftchild==0 && interval(i).exist==1
                     if S(i)≥max(S)/C
                        x=interval(i).content.Interval(1);
                        y=interval(i).content.Interval(2);
                        M=M+1;
                        interval(M)=struct('leftchild',0,'rightchild',...
                            0,'parent',i,'content',0,'exist',1);
                        interval(M).content=struct('Interval',[x,...
                            (x+y)/2],'alphaL',0, ...
                        'alphaR',0,'betaL',0,'betaR',0, '∆L',0,...
                        '∆R',0,'niuL',0, ...
                        'niuR',0,'niu',0);
                        interval(i).leftchild=M;
                        M=M+1;
                        interval(M)=struct('leftchild',0,'rightchild',...
```

```
                          0,'parent',i,'content',0,'exist',1);
                      interval(M).content=struct('Interval',[(x+y)/2,...
                          y],'alphaL',0, ...
                      'alphaR',0,'betaL',0,'betaR',0, 'ΔL',0,...
                      'ΔR',0,'niuL',0,'niuR',0,'niu',0);
                      interval(i).rightchild=M;
                   elseif interval(i).parent>0
                          n=interval(i).parent;
                          m=interval(n).leftchild;
                          if S(m)+S(m+1)<max(S)/(2^10) && ...
                              interval(2*m+1-i).leftchild==0
                             interval(m).exist=0;
                             interval(m+1).exist=0;
                             interval(n).leftchild=0;
                             interval(n).rightchild=0;
                          end
                    end
                end
            end
            for i=1:M
                if interval(i).leftchild==0 && interval(i).exist==1
                   X=ABD(interval,i,F,PhiL,PhiR,GL,GR);
                   interval(i).content.alphaL=X(1);
                   interval(i).content.alphaR=X(2);
                   interval(i).content.betaL=X(3);
                   interval(i).content.betaR=X(4);
                   interval(i).content.ΔL=X(5);
                   interval(i).content.ΔR=X(6);
                end
            end
            for j=1:M
                i=M+1-j;
                n=interval(i).parent;
                if n>0 && interval(i).exist==1
                   m=interval(n).leftchild;
                   U(1)=interval(m).content.alphaL;
                   U(2)=interval(m).content.alphaR;
                   U(3)=interval(m).content.betaL;
                   U(4)=interval(m).content.betaR;
                   U(5)=interval(m).content.ΔL;
                   U(6)=interval(m).content.ΔR;
                   V(1)=interval(m+1).content.alphaL;
                   V(2)=interval(m+1).content.alphaR;
                   V(3)=interval(m+1).content.betaL;
                   V(4)=interval(m+1).content.betaR;
                   V(5)=interval(m+1).content.ΔL;
```

```
                V(6)=interval(m+1).content.ΔR;
                Delta=1-U(3)*V(2);
                interval(n).content.alphaL=(1-V(1))*(U(1)+...
                    Delta-1)/Delta+V(1);
                interval(n).content.alphaR=V(2)*(1-U(4))*...
                    (1-U(1))/Delta+U(2);
                interval(n).content.betaL=U(3)*(1-V(4))*...
                    (1-V(1))/Delta+V(3);
                interval(n).content.betaR=(1-U(4))*(V(4)+...
                    Delta-1)/Delta+U(4);
                interval(n).content.ΔL=(1-V(1))*U(5)/...
                    Delta+V(5)+(V(1)-1)*U(3)*V(6)/Delta;
                interval(n).content.ΔR=(1-U(4))*V(6)/...
                    Delta+U(6)+(U(4)-1)*V(2)*U(5)/Delta;
            end
        end
        for i=1:M
            n=interval(i).leftchild;
            if n>0
                x=interval(i).content.niuL;
                y=interval(i).content.niuR;
                z=interval(i).content.niu;
                u1=interval(n).content.alphaL;
                u3=interval(n).content.betaL;
                u5=interval(n).content.ΔL;
                v4=interval(n+1).content.betaR;
                v2=interval(n+1).content.alphaR;
                v6=interval(n+1).content.ΔR;
                interval(n).content.niuL=x;
                interval(n+1).content.niuR=y;
                interval(n).content.niu=z;
                interval(n+1).content.niu=z;
                NIU=[1,v2;u3,1]\[y*(1-v4)-z*v6;x*(1-u1)-z*u5];
                interval(n).content.niuR=NIU(1);
                interval(n+1).content.niuL=NIU(2);
            end
        end
        X=zeros(1,M);
        j=0;
        for i=1:M
            if interval(i).leftchild==0 && interval(i).exist==1
                X(j+1)=i;
                j=j+1;
            end
        end
        Num=j;
```

```
        Mat=[a-1;0;0;0];
        for i=1:Num
            K=Ch(interval(X(i)).content.Interval);
            A=P(interval,X(i),PhiL,PhiR,GL,GR);
            Z1=zeros(10,1);
            Z2=zeros(10,1);
            Z3=zeros(10,1);
            for j=1:10
                Z1(j)=F(K(j));
                Z2(j)=PhiL(K(j));
                Z3(j)=PhiR(K(j));
            end
            sigma=A\Z1+A\Z2*interval(X(i)).content.niuL+...
                A\Z3*interval(X(i)).content.niuR;
            Mat1=zeros(4,10);
            for j=1:10
                Mat1(1,j)=K(j);
                Mat1(2,j)=sigma(j);
                Mat1(3,j)=X(i);
            end
            Mat=[Mat Mat1];
        end
        [¬,idx]=sort(Mat(1,:));
        Mat=Mat(:,idx);
        JL=zeros(1,Num+1);
        JR=zeros(1,Num+1);
        for i=1:Num
            n=Mat(3,10*i);
            a1=interval(n).content.alphaL;
            b1=interval(n).content.betaL;
            d1=interval(n).content.ΔL;
            n1=interval(n).content.niuL;
            n2=interval(n).content.niuR;
            JL(i+1)=JL(i)+d1+n1*a1+n2*b1;
            m=Mat(3,10*(Num+1-i));
            a2=interval(m).content.alphaR;
            b2=interval(m).content.betaR;
            d2=interval(m).content.ΔR;
            m1=interval(m).content.niuL;
            m2=interval(m).content.niuR;
            JR(Num+1-i)=JR(Num+2-i)+d2+m1*a2+m2*b2;
        end
        for i=1:Num
            Y=interval(Mat(3,10*i)).content.Interval;
            K=Ch(Y);
            Z1=zeros(10,1);
```

```
                   Z2=zeros(10,1);
                   for t=1:10
                       Z1(t)=GL(K(t));
                       Z2(t)=GR(K(t));
                   end
                   for j=1:10
                       ur=GR(K(j))/s*(JL(i)+ ...
                       LIntegral(Y(1),K(j),Y(2),Z1.*(Mat(2,10*(i-1)+...
                       2:10*i+1))'))+GL(K(j))/s*(JR(i+1)+ ...
                       RIntegral(Y(1),K(j),Y(2),Z2.*(Mat(2,10*(i-1)+...
                       2:10*i+1))'));
                       Mat(4,10*(i-1)+1+j)=ur;
                   end
               end
               Er=Error(Mat,Mat2);
               Mat2=Mat;
               nn=size(Mat,2);
               N=M;
         end
         Ur=zeros(2,nn-1);
         Ur(1,:)=Mat(1,2:nn);
         Ur(2,:)=Mat(2,2:nn);
         Leftendpoint=zeros(1,Num);
         for i=1:Num
             Leftendpoint(i)=interval(Mat(3,10*i)).content.Interval(1);
         end
end
```

The Newton method in the paper of Starr and Rokhlin views 2-order ODE about function with image in real number as 1-order ODE about function with image in real vectors.But to decrease the complexcity of computation,I use another "Newton method" which is a little different from that in the paper of Starr and Rokhlin.Below is my method:

To solve u for $u^{(2)}(x) = f(x, u, u^{(1)})$,we can define $\mathcal{F}(u) := u^{(2)} - f(x, u, u^{(1)})$,then we only need to solve the "zero" of $\mathcal{F}$.And formally Newton's method (in its pristine form) is

$$u_{n+1} = u_n - \frac{\mathcal{F}(u_n)}{\mathcal{F}^{(1)}(u_n)}$$

Where $\mathcal{F}^{(1)}$ is just "formal dervative" .Let $v = u_{n+1} - u_n$,then

$$v\mathcal{F}^{(1)}(u_n) = -\mathcal{F}(u_n)$$

But by definition of deverivative,we have

$$v\mathcal{F}^{(1)}(u_n) \simeq \mathcal{F}(u_n+v)-\mathcal{F}(u_n) \simeq v^{(2)}-v\partial_u f(x,u_n,u_n^{(1)})-v^{(1)}\partial_{u^{(1)}} f(x,u_n,u_n^{(1)})$$

So $u_{n+1} = n_n + v$ and we only need to solve $v^{(2)} - v\partial_u f(x, u_n, u_n^{(1)}) - v^{(1)}\partial_{u^{(1)}} f(x, u_n, u_n^{(1)}) = -\mathcal{F}(u_n)$,which can be solved by direct solver. The initial guess u_1 need to satisfies the boundary condition,then in each step,we only need to solve v with homogeneous boundary condition.

In the codes above,the function f correspond to the problem $u^{(2)}(x) = f(x, u, u^{(1)})$,and functions $GL, GR, dGL, dGR, PhiL, PhiR, F$ correspond to $g_l, g_r, g_l^{(1)}, g_r^{(1)}, \psi_l, \psi_r, u_i, \widetilde{f}(= -\mathcal{F}(u_n))$ (in the paper of Lee and Greengard) respectively.s is just the constant $g_l g_r^{(1)} - g_r g_l^{(1)}$.

2 Numerical results

2.1 One example from the paper of Lee and Greengard

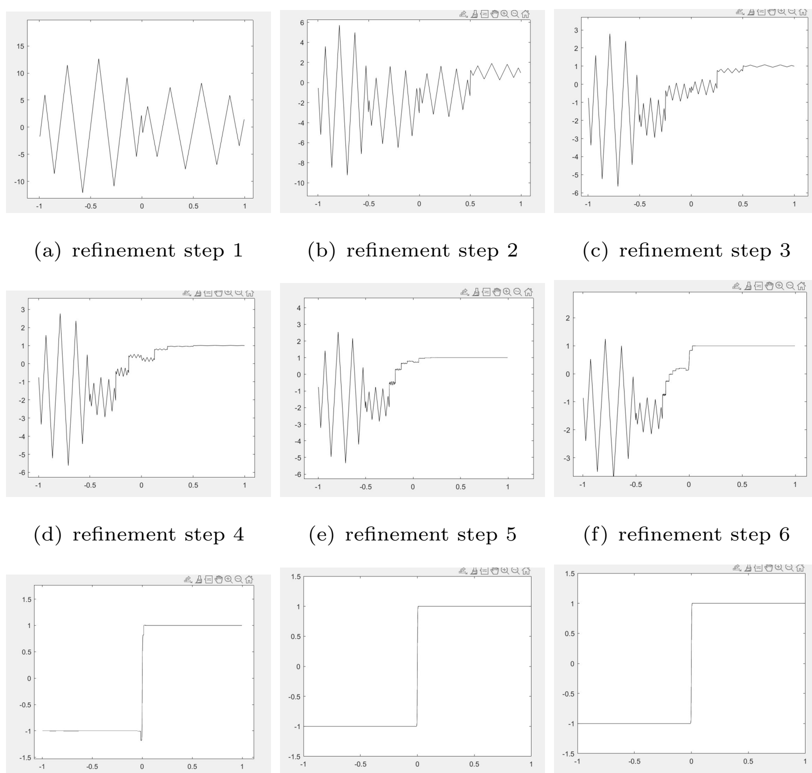
My choice is example 1 (Viscous shock) in the paper of Lee and Greengard,i.e.

$$u^{(2)}(x) + \frac{2x}{\epsilon}u^{(1)}(x) = 0$$

For $\epsilon = 10^{-5}$,the algorithm requires nine levels of mesh refinement.As figure1 shown.

And we also have the results about relative L^2 norm and CPU time,as the table below shown:

And the final graph corresponding to the table above is figure2.



(a) refinement step 1 (b) refinement step 2 (c) refinement step 3

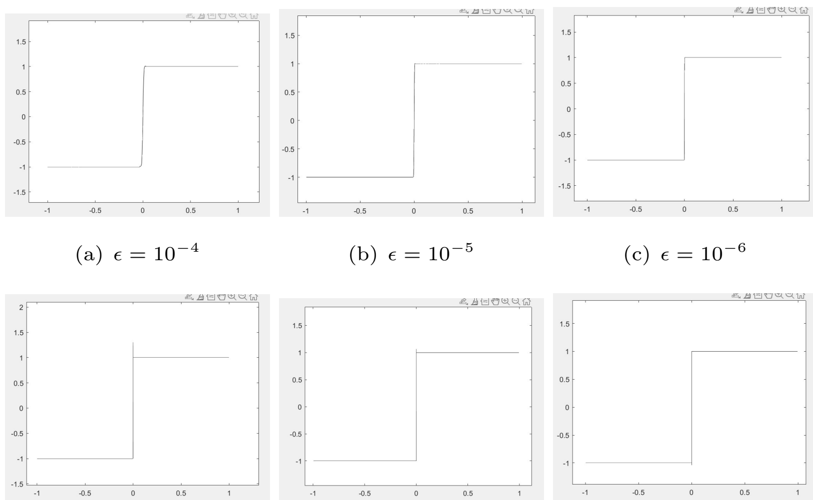
(d) refinement step 4 (e) refinement step 5 (f) refinement step 6

(g) refinement step 7 (h) refinement step 8 (i) refinement step 9

图 1: refinement process for the case $\epsilon = 10^{-5}$

ϵ	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}
C	16	16	24	24	24	32
L^2-error	$6.53 \cdot 10^{-6}$	$1.08 \cdot 10^{-7}$	$7.98 \cdot 10^{-9}$	$1.81 \cdot 10^{-4}$	$2.33 \cdot 10^{-6}$	$2.38 \cdot 10^{-7}$
CPU time (second)	0.478	0.505	0.567	1.22	0.743	1.42

表 1: results about relative L^2 norm and CPU time



(a) $\epsilon = 10^{-4}$ (b) $\epsilon = 10^{-5}$ (c) $\epsilon = 10^{-6}$

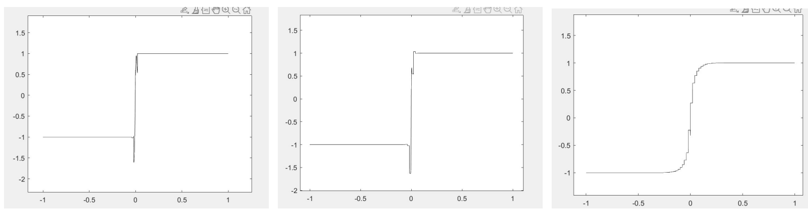
(d) $\epsilon = 10^{-7}$ (e) $\epsilon = 10^{-8}$ (f) $\epsilon = 10^{-9}$

图 2: the final graph corresponding to table1

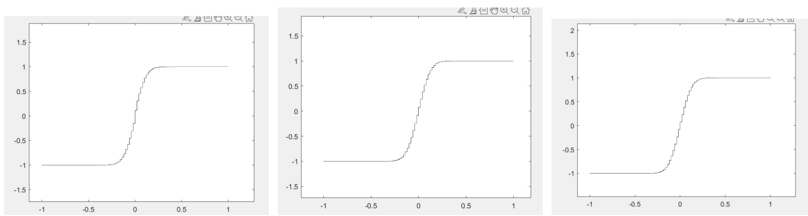
Here $\log_2 C$ is just the refinement parameter in the paper of Lee and Greengard(i.e.$S_{div} = C^{-1} \max_{i=1}^{M}$).By these results,we get that the algorithm has well numerical performance on aspects of speed,accuracy and adaptivity.

However,if we consider the direct solver,then the result is as fig3 shown:

By these results,we get the fast algorithm in problem2 is more adaptive,faster,more accurate than the direct solver in problem 1.



(a) $\epsilon = 10^{-4}, t = 3.63$ (b) $\epsilon = 10^{-5}, t - 3.73$ (c) $\epsilon = 10^{-6}, t = 3.58$



(d) $\epsilon = 10^{-7}, t = 3.61$ (e) $\epsilon = 10^{-8}, t = 3.60$ (f) $\epsilon = 10^{-9}, t = 3.62$

图 3: the graph corresponding to direct solver,where there is 100 equidistant intervals on [-1,1] and 10 chebychev points on each small interval.And t is CPU time (second)

2.2 Another two examples

Example 1:$u^{(2)} - xu^{(1)} + 2u = 2$ boundary conditions are $u(0) = 0$,$u(2) = 4$.Then the solution is $u(x) = x^2$.And for fast solver,we choose $C = 16$,then the CPU time is 0.101s,and the error(corresponding to L^1 norm) is $O(10^{-16})$.For the direct solver,we have the following table2.

n	20	40	60	80	100
L^1-error	0.009	0.0024	0.001	$6.09 \cdot 10^{-4}$	$3.91 \cdot 10^{-4}$
CPU time (second)	0.163	0.593	1.29	2.27	3.55

表 2: results about relative L^1 norm and CPU time.n is the number of equisdistant intervals,the number of chewbyshev points in each subinterval is stil 10

From the table,we can get for simple ODE,fast solver still performs much better than the direct solver.

Example2:For nonlinear eqn: $u^{(2)}(x) = -\frac{u(x)}{x^2(u(x)+1)} + \frac{u^{(1)}(x)}{x(u(x)+1)^2}$, $u(e) = 1$,$u(2e^2) = 2$.The solution is Lambert-W function.We choose $u_1 = \frac{x+2e^2-2e}{2e^2-e}$ as initial guess,and the result is as fig4 shown.

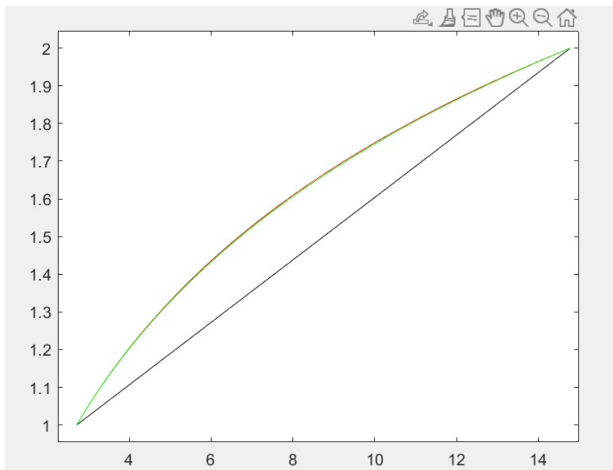


图 4: black line is initial guess u_1,red line is u_2,and green line is Lambert-W function

From the figure,we get after one iteration,what we get is very close to Lambert-W function.Moreover,if we iterate three times,then for u_4,the L^1-error is $O(10^{-6})$,and the CPU time is 2.43s.So the nonlinear solver performs well.

3 Theoretical questions

3.1 Theoretical question 1

W.L.O.G. assume the partition of interval $[a, c]$ is $a = a_1 < \cdots < a_{n+1} = c$,and assume $\forall 1 \leq i \leq n$,there are K Chebyshev nodes $\{\tau_i^j\}_{j=1}^K$on interval $[a_i, a_{i+1}]$,then let $S := \{\tau_i^j | 1 \leq i \leq n, 1 \leq j \leq K\}$,$V = \{f : S \to \mathbb{R}\}$,then the integral operator P

can be viewed as a linear map from V to V which satisfies that $P(\sum_{1\le i\le n,1\le j\le K} a_{ij}e_{ij}) = Int(\tau_i^j)e_{ij}$,where $a_{ij} \in \mathbb{R}$,$e_{ij} \in V$ s.t. $\forall 1 \le i_0 \le n, 1 \le j_0 \le K, e_{ij}(\tau_{i_0}^{j_0}) = \delta_{i,i_0}\delta_{j.j_0}$,and

$$Int : [a,c] \to \mathbb{R} \quad x \mapsto \Phi(x)+\psi_l(x)\int_a^x g_l(t)\Phi(t)dt+\psi_r(x)\int_x^c g_r(t)\Phi(t)dt$$

Here $\forall 1 \le i \le n$,$\Phi|_{[a_i,a_{i+1}]}$ is just the Chebyshev interpolant of (τ_i^j, a_{ij}) $(1 \le j \le K)$ on $[a_i, a_{i+1}]$.Then e_{ij} is a basis of V wrt which the matrix of P is $I_{nK} + \psi_L L g_L + \psi_R R g_R$,where ψ_L, g_L, ψ_R, g_R are diagonal matrix s.t.

$$\psi_L(e_{ij}) = \psi_l(\tau_i^j)e_{ij}, \psi_R(e_{ij}) = \psi_r(\tau_i^j)e_{ij}$$

$$g_L(e_{ij}) = g_l(\tau_i^j)e_{ij}, g_R(e_{ij}) = g_r(\tau_i^j)e_{ij}$$

And $L(\sum_{1\le i\le n,1\le j\le K} a_{ij}e_{ij}) = \sum_{1\le i\le n,1\le j\le K}(\int_a^{\tau_i^j}\Phi(t)dt)e_{ij}$,$R(\sum_{1\le i\le n,1\le j\le K} a_{ij}e_{ij}) = \sum_{1\le i\le n,1\le j\le K}(\int_{\tau_i^j}^b \Phi(t)dt)e_{ij}$,where definition of Φ is the same as above.So for any fixed i,

$$L(\sum_{j=1}^K r_j e_{ij}) = L_i(\sum_{j=1}^K r_j e_{ij}) + (\int_{a_i}^{a_{i+1}} F(t)dt)\sum_{k>i,1\le j\le K} e_{kj}$$

Here F is the Chebyshev interpolant of (τ_i^j, r_j) $(1 \le j \le K)$ on $[a_i, a_{i+1}]$,and L_i is just the left integration operator(this is metioned in the paper of Lee and Greengard) on $[a_i, a_{i+1}]$.So let τ_i^j be the $((i-1)K+j)$th basis element,then assume $L = (L_{ij})_{1\le i,j\le n}$,where L_{ij} are $K \times K$ matrix.Then $\forall X = (r_1, \cdots, r_K)^t \in \mathbb{R}^K$,$\forall k < i$,$L_{k,i}X = 0$;$\forall k > i$, $L_{k,i}X = (\int_{a_i}^{a_{i+1}} F(t)dt, \cdots, \int_{a_i}^{a_{i+1}} F(t)dt)^t$.So

$$L = \begin{pmatrix} L_{11} & & & \\ L_{21} & L_{22} & & \\ \vdots & \ddots & \ddots & \\ L_{n1} & \cdots & L_{n-1,n} & L_{nn} \end{pmatrix}$$

Here $L_{ii} = L_i$ and for any $i < n$,all rows of $\begin{pmatrix} L_{i+1,i} \\ \vdots \\ L_{ni} \end{pmatrix}$ are the same.And $\exists X = (r_1, \cdots, r_K)^t \in \mathbb{R}^K$ s.t. $\int_{a_i}^{a_{i+1}} F(t)dt \neq 0$,so $rk(\begin{pmatrix} L_{i+1,i} \\ \vdots \\ L_{ni} \end{pmatrix}) = 1$.Then $\psi_L L g_L = (\widetilde{L}_{ij})_{1\leq i,j\leq n}$,where $\widetilde{L}_{ij}$ are $K \times K$ matrices and $\forall 1 \leq i < j \leq n$,$\widetilde{L}_{ij} = 0$ and $rk(\begin{pmatrix} \widetilde{L}_{i+1,i} \\ \vdots \\ \widetilde{L}_{ni} \end{pmatrix}) \leq 1$.Similarly $\psi_R R g_R = (\widetilde{R}_{ij})_{1\leq i,j\leq n}$,where $\widetilde{R}_{ij}$ are $K\times K$ matrices and $\forall 1 \leq j < i \leq n$,$\widetilde{R}_{ij} = 0$ and $rk(\begin{pmatrix} \widetilde{R}_{1,i} \\ \vdots \\ \widetilde{R}_{i-1,i} \end{pmatrix}) \leq 1$.So the matrix $P = (P_{ij})_{1\leq i,j\leq n}$,where for all $i < j$,$rk(\begin{pmatrix} P_{i+1,i} \\ \vdots \\ P_{ni} \end{pmatrix}) \leq 1$ and $rk(\begin{pmatrix} P_{1,j} \\ \vdots \\ P_{j-1,j} \end{pmatrix}) \leq 1$.And P_{ii} is just the local integral operator $P_{[a_i,a_{i+1}]}$ on $[a_i, a_{i+1}]$.(This notation is the same as the one in the paper of Lee and Greengard).And when $a_{i+1} - a_i$ is small enough,the norm of R_i and L_i tends to zero,so $P_{ii} - Id = \widetilde{R}_i + \widetilde{L}_i = O(R_i) + O(L_i) = o(Id)$,hence $rk(P_{ii}) = K$ if $a_{i+1} - a_i$ is small enough. All in all,we get the rank structure of P (P is just the matrix A in Theoretical question 1).Generally,when the number of chewbyshev points of $[a_i, a_{i+1}]$ are distinct,use the same method,we can get the similar answer.

3.2 Theoretical question 2

Newton's method (in its pristine form) is very sensitive to the choice of initial guess.But my “Newton method” in Programming problem 3 avoid function whose image is vectoror matrix,and in my method the inintial guess must satisfies boundary condition,so in the process of iteration we only need to solve ODE with trival boundary condition.Then such method and initial guess can help us get better iteration results.If we want to get better initial guess,we can sketch the graph of the solution roughly and choose an initial guess which is close to the graph and satisfies boundary condition.

3.3 Theoretical question 3