

Handout 5 – Corrosion Code Example: fakephi.f

```
program fakephi

c Kreider 03/10/11
c initial code to get threshold Y0
c fix sign error in function g, to minus
c adjust right BC to phi_x(L) = -k(phi-phi_amb)

implicit none
double precision phiI(100001),x(100001),g,g1,g2,g3,g4,dx
double precision F,R,T,M,L,ip,DM,w,Y0min,Y0max,tol,FCN
double precision beta,alpha,Y0,phi0,phiRP,M0
double precision phiamb,k
double precision AL(100001),AM(100001),AR(100001),RHS(100001)
double precision dt,delta(100001),i0,Cmetal,AAM,alpham,P1,PN
double precision phi(100001,0:1),H(100001,0:1),Hx(100001)
common /param/ beta,alpha,phi0,phiRP,M0,L,k,phiamb
common /grid/ dx,im,il,x
integer i,im,il,N,kmax,kk,imax
external FCN

character*1 cflag
character*3 chan3
integer np

write(6,*) 'Enter id flag (1 character)'
read(5,77) cflag
77 format(a1)

c concentrations are in mole/cm^3

F = 9.648d4
R = 8.314472d0
T = 25.d0 + 273.15d0
alpha = F/R/T
ip = 1.d-6
DM = .72d-5
w = 1.d-3
beta = ip/(F*DM*w*alpha)
phiamb = 0.d0
k = 1.d-1
M0 = 1.d-10
c M0 = 5.d-8 is too high
Y0min = M0/2.d0 + 1.d-12
Y0max = Y0min + .1d0
tol = 1.d-12
```

```

phiRP = .31d0

i0 = 1.d-6
Dm = 1.d0
Cmetal = .47d0
AAM = 1.d0
alphan = 1.d0

c  grid parameters (x_i and t_k)
M = 1.d0
L = 3.d0
N = 3
dx = 10.d0**(-N)
im = 10**N + 1
il = int(L+.01d0)*10**N + 1
dt = 1.d-3
kmax = 10001
kmax = 2
np = (kmax-1)/100

do i=1,il
  x(i) = 0.d0 + dfloat(i-1)*dx
end do

write(6,*) alpha,beta,(F*DM*w*alpha)
write(6,*) F,DM,w,ip
write(6,*) 'fcn ',FCN(Y0min)
write(6,*) 'fcn ',FCN(Y0max)
pause

write(6,*) im,il

call bisect(FCN,Y0min,Y0max,Y0,tol)

c  now have correct Y0; find corresponding phi for IC for time stepping

phi0 = beta*L/k/(Y0+M0) + phiamb
phiI(il) = phi0

do i=il,2,-1
  g1 = g(x(i),phiI(i),Y0)
  g2 = g(x(i)-.5d0*dx,phiI(i)-.5d0*dx*g1,Y0)
  g3 = g(x(i)-.5d0*dx,phiI(i)-.5d0*dx*g2,Y0)
  g4 = g(x(i)-dx,phiI(i)-dx*g3,Y0)
  phiI(i-1) = phiI(i) - dx*( g1+2.d0*(g2+g3)+g4 )/6.d0
end do

open(8,file='phi.dat')
open(10,file='Y.dat')

```

```

open(11,file='M.dat')
open(12,file='H.dat')
do i=1,il
  write(8,*) x(i),phiI(i)
  write(10,*) x(i),Y0*exp(alpha*(phiI(i)-phi0))
  M = Y0*exp(alpha*(phiI(i)-phi0))
  - (Y0-2.d0*M0)*exp(-alpha*(phiI(i)-phi0))
  M = M/2.d0
  write(11,*) x(i),M
  write(12,*) x(i),Y0*exp(alpha*(phiI(i)-phi0))-2.d0*M
end do
write(6,*) 'Y0 phi(im) phiRP'
write(6,*) Y0,phiI(im),phiRP
write(6,*) 'Y0,H0,M0,phi0 ',Y0,Y0-2.d0*M0,M0,phi0

imax = il

c  initial conditions
do i=1,imax
  H(i,0) = 0.d0
  phi(i,0) = phiI(i)
  phi(i,0) = 0.d0
end do

do kk=1,kmax-1
c  time step from kk to kk+1
c  write(6,*) 'time step ',kk

  do i=1,imax
    H(i,1) = H(i,0)
    + dt*AAM*exp(-alphan*alpha*phi(i,0))/(2.d0*F*Cmetal)
  end do
  do i=2,imax-1
    Hx(i) = (H(i+1,1)-H(i-1,1))/(2.d0*dx)
  end do

c  build matrix
  AM(1) = -1.5d0/dx
  AR(1) = 2.d0/dx
  P1 = -.5d0/dx
  do i=2,imax-1
    AL(i) = (w+H(i,1))/dx**2 - Hx(i)/(2.d0*dx)
    AM(i) = -2.d0*(w+H(i,1))/dx**2 - i0/F/DM*AAM
    AR(i) = (w+H(i,1))/dx**2 + Hx(i)/(2.d0*dx)
  end do
  PN = 1.5d0/dx
  AL(imax) = -2.d0/dx
  AM(imax) = k + .5d0/dx

```

```

c      row reduction
      AM(1) = AM(1) - P1*AL(2)/AR(2)
      AR(1) = AR(1) - P1*AM(2)/AR(2)
      AL(imax) =
      AM(imax) =
c      t is t_{kk+1/2} for use in source term
      t = dfloat(kk-1)*dt + dt/2.d0

c      build rhs
      rhs(1) = 0.d0
      do i=2,imax-1
         rhs(i) = -i0/F/DM
      end do
      rhs(imax) = k*phiamb
c      row reduction
      rhs(1) = rhs(1) - P1*rhs(2)/AR(2)
      rhs(imax) =
         write(6,*) (rhs(i),i=1,20)
         pause
      call tridiag(AL,AM,AR,rhs,delta,imax)
      do i=1,imax
         phi(i,1) = delta(i)
      end do
      if (np*(kk/np) .eq. n) then
         call ooutfn('Fphi',chan3(kk/np),cflag,1)
         call ooutfn('Fakh',chan3(kk/np),cflag,2)
         write(6,*) kk,' ',chan3(kk/np)
         do i=1,imax
            write(1,*) x(i),phi(i,1)
            write(2,*) x(i),H(i,1)
         end do
         close(1)
         close(2)
      end if
end do

stop
end

function FCN(Y0)
implicit none
double precision FCN,phi(100001),x(100001),dx
double precision g,g1,g2,g3,g4
double precision beta,alpha,Y0,phi0,phiRP,M0,phiamb,L,k
common /param/ beta,alpha,phi0,phiRP,M0,L,k,phiamb
common /grid/ dx,im,il,x
integer i,il,im

```

```

phi0 = beta*L/k/(Y0+M0) + phiamb
phi(il) = phi0

do i=il,im+1,-1
  g1 = g(x(i),phi(i),Y0)
  g2 = g(x(i)-.5d0*dx,phi(i)-.5d0*dx*g1,Y0)
  g3 = g(x(i)-.5d0*dx,phi(i)-.5d0*dx*g2,Y0)
  g4 = g(x(i)-dx,phi(i)-dx*g3,Y0)
  phi(i-1) = phi(i) - dx*( g1+2.d0*(g2+g3)+g4 )/6.d0
end do

FCN = phi(im) - phiRP
return
end

function g(x,phi,Y0)
implicit none
double precision g,x,phi,factor
double precision beta,alpha,Y0,phi0,phiRP,M0,phiamb,L,k
common /param/ beta,alpha,phi0,phiRP,M0,L,k,phiamb

factor = alpha*(phi-phi0)
c g = -beta*x/Y0/( cosh(factor) + 2.d0*sinh(factor) )
g = -beta*x/(1.5d0*Y0*exp(factor)-.5d0*(Y0-2.d0*M0)*exp(-factor))

return
end

subroutine bisect(F,a,b,c,eps)
implicit none
double precision F,a,b,c,er,el,eps
integer n
c external F

el = a
er = b
n = 1
10 c = (el+er)/2.d0
c write(6,*) n,el,c,er,F(el),F(c),F(er)
write(6,*) n,c,F(el),F(c),F(er),abs((er-el)/c)
c pause
if ( abs((er-el)/c) .lt. eps) return
if ( F(el)*F(c) .lt. 0.0 ) er = c
if ( F(c)*F(er) .lt. 0.0 ) el = c
n = n+1
go to 10
end

```

```

SUBROUTINE tridiag(a,b,c,r,u,n)
implicit none
INTEGER n
double precision a(n),b(n),c(n),r(n),u(n)
INTEGER j
double precision bet,gam(n)

c   a is AL, b is AM, c is AR
c   r is rhs, u is output, n is dimension

    if(b(1).eq.0.)pause 'tridag: rewrite equations'
    bet=b(1)
    u(1)=r(1)/bet
    do 11 j=2,n
        gam(j)=c(j-1)/bet
        bet=b(j)-a(j)*gam(j)
        if(bet.eq.0.)pause 'tridag failed'
        u(j)=(r(j)-a(j)*u(j-1))/bet
11   continue
    do 12 j=n-1,1,-1
        u(j)=u(j)-gam(j+1)*u(j+1)
12   continue
    return
    END

    subroutine ooutfn(fpr,fid,cflag,lun)
c   usage
c   type *,' Enter file id'
c   accept (2a), id
c   call ooutfn(id,'i',1) creates output file i'id'.dat
c   taken from Nazanin Imani

    character*3 fid
    character*1 cflag
    character*4 fpr
    integer lun
    character*20 flname

    flname = fpr // cflag // fid
    open(unit=lun,file=flname,err=150)
    return
150   write(6,*) '*** ERROR IN OPENING FILE ***'
    return
    end

    function chan3(M)
c   Stephen Cardarelli, Oct 2003
c   provides time step files 000 to 999

```

```
implicit none
character*3 chan3
integer i,j,k,M,countout

countout = M
i = countout/100
countout = countout - (i*100)
j = countout/10
k = countout - (j*10)

chan3 = char(i+48)//char(j+48)//char(k+48)
return
end
```