Reissued To N. Sridhar NS
S/N No. 265 4/8/98
20-8910
20-1136

CNWRA
CONTROLLED
COPY 265 4/8/98

CNWRA
CONTROLLED
COPY 102

NOT A Controlled document
4/8/98
4/16/98

Contents

DATE
Jan 21st '94

INITIAL ENTRY:

SUBSTANTIALLY COMPLETE CONTAINMENT

– EXAMPLE PROBLEM CODE

(SCCEX)

Individuals Performing experiment –

J C Walton, Peter Lichtner,
G. Cragnolino etc.

Technical Assistance: Pavitra Ramanujan.

Description of Experiment: To evaluate the Green's integral function using Numerical Recipes in FORTRAN-77.

Routines used from Numerical Recipes were
1) qromo.f
2) ~~qsimp.f~~ midpnt.f
3) POLINT.F

8/29/00

The Green's integral was used to evaluate the temperature difference between container surface and inside of failed container. Subroutine Integral which performed the evaluation of green's integral was written in "Subroutine.f".
In the driver program "Bernie.f", Subroutine Integral was called after linear interpolation was performed to calculate average temperature. The integral was called from $i=1$ to nhist (timestep) because Green's integrals blows up for $i=0$

In "Subroutines.f", the Subroutine Integral worked as follows:

func(t) was defined as double precision function.

func = expression for convolution integral

(ie)

func = tons * Pow * grepos.

where

tons = metric tons of spent metal fuel

grepos = green's function output for repository

Pow = watts per metric ton initial heavy metal.

This func(t) was passed to Subroutine qromof which took the lower and upper limits of integration of the integral and calculated the result. For this, qromo called subroutine midpnt.f to repeatedly evaluate the value of func(t) and return the final result to "integral.".

Measurement Parameters - The time to integrate the convolution integral was measured and documented. It was not substantially different from the earlier used method for evaluating the integral

R. Pavitra

(Pavitra Ramanujan)

01/21/94

DATE Jan 31st 94

Experiment 2: To calculate the average temperature difference between the outer rim and the inner rim of the repository and also the absolute value of their difference using green's integral functions (instead of numerical recipes used earlier).

Method: Two dummy functions dumf1(t) and dumf2(t) were defined as double precision functions and were passed to qromo which again called midpnt.f to repeatedly divide the interval of integration and calculate the result and pass it back to Subroutine Integral. Sub-Routine integral is called by the main program brine.f.

The green's functions was called twice.

dumf1 was evaluated for green's parameters:

x = 0.5 ; w = width

y = 0.

z = w/2

dumf1 = tons * Pow * grepos ;

tons = metric tons of spent fuel fuel.

grepos = green's function output for repository (at outer rim) (at inner rim)

Pow = watts per metric ton initial heavy metal

dumf2 was evaluated for green's parameters

$x = 0.$
$y = 5 + (w/2)$
$z = 0.$

dumf2 = tons * pow * grepos.

tons = metric tones of spent fuel

grepos = green's function output for outer rim of repository

Pow - watts per metric ton of initial heavy metal.

Result: Using green's integral for evaluation of average temperature did not produce significant increase in precision of or significant decrease in run time.

R. Pavitra
(Pavitra Ramamyan)
02/24/94

code written in main Program brine.f.

```
c       calculate temperature from convolution integral
        if (i .gt. 1) then
          call integral(tempconv,temprim,tempinn)
        else
          tempconv = 0.d0
        endif
        tempconv = tempconv + tzero
        write(*,*) 'tempconv = ', tempconv
        write(*,*) 'temprim = ', temprim
        write(*,*) 'tempinn = ', tempinn
```

R. Pavitra

```
        time0=dtime(mytime)
        write(*,'("--> integral time: ",5(1pg12.4))') time0,mytime(1),
     .  mytime(2)
 40     continue
```

R. Pavitra

```
c This loop displays the values of the green's integral on the outer
c rim and the inner rim of the repository and also the absolute
c value of their difference
```

code written in
"SubRoutines.f"

Subroutines qromo.f, midpnt.f & polint.f are taken from Numerical Recipes in FORTRAN.

```
c*********************************************************************
c calculation of adjusted time and temperature from the convolution integral
c routines from numerical recipes were used to perform the integration
c the integral is an improper integral
c integration on an open ended interval was performed

      subroutine integral(result,result1,result2)
      implicit real*8 (a-h,o-z)
      include 'bulk.h'
      external func, midpnt, dumf1,dumf2

      call qromo(func,0.d0,tint,result,midpnt)
      call qromo(dumf1,0.d0,tint,result1,midpnt)
      call qromo(dumf2,0.d0,tint,result2,midpnt)

      return
      end

      double precision function func(t)
```

R. Parentau


```
      implicit real*8 (a-h,o-z)
      include 'bulk.h'

      x=0.d0
      y=0.d0
      z=0.d0
      alpha=alphar
      rcp=rhocpr
      w=width
      tt=tint-t
      call power(t,pow)
      tt=tint-t
      call green(tt,w,x,y,z,alpha,rcp,grepos)
      func=tons*pow*grepos
      return
      end

      double precision function dumf1(t)

      implicit real*8 (a-h,o-z)
      include 'bulk.h'

      x=0.d0
      y=0.d0
      z=0.d0
      alpha=alphar
      rcp=rhocpr
      w=width
      tt=tint-t
      call power(t,pow)
      tt=tint-t
      call green(tt,w,0.,w/2,0.,alpha,rcp,grepos)
      dumf1=tons*pow*grepos
      return
      end

      double precision function dumf2(t)

      implicit real*8 (a-h,o-z)
      include 'bulk.h'

      x=0.d0
      y=0.d0
      z=0.d0
      alpha=alphar
      rcp=rhocpr
      w=width
      tt=tint-t
      call power(t,pow)
      tt=tint-t
      call green(tt,w,0.,5+(w/2),0.,alpha,rcp,grepos)
      dumf2=tons*pow*grepos
      return
      end
*************************************************************
```

R. Parentau

```
      SUBROUTINE qromo(func,A,B,SS,midpnt)
        implicit real*8 (a-h,o-z)
      PARAMETER (EPS=1.E-6,JMAX=14,JMAXP=JMAX+1,KM=4,K=KM+1)
      DIMENSION S(JMAXP),H(JMAXP)
      H(1)=1.
      DO 11 J=1,JMAX
        CALL midpnt(FUNC,A,B,S(J),J)
        IF (J.GE.K) THEN
          CALL POLINT(H(J-KM),S(J-KM),K,0.0,SS,DSS)
          IF (ABS(DSS).LT.EPS*ABS(SS)) RETURN
        ENDIF
        S(J+1)=S(J)
        H(J+1)=H(J)/9.
11    CONTINUE
      PAUSE 'Too many steps.'
      RETURN
      END

      SUBROUTINE midpnt(func,A,B,S,N)
        implicit real*8 (a-h,o-z)
      IF (N.EQ.1) THEN
        S=(B-A)*FUNC(0.5*(A+B))
        IT=1
      ELSE
        TNM=IT
        DEL=(B-A)/(3.*TNM)
        DDEL=DEL+DEL
        X=A+0.5*DEL
        SUM=0.
        DO 11 J=1,IT
          SUM=SUM+FUNC(X)
          X=X+DDEL
          SUM=SUM+FUNC(X)
          X=X+DEL
11      CONTINUE
        S=(S+(B-A)*SUM/TNM)/3.
        IT=3*IT
      ENDIF
      RETURN
      END
```



```
      SUBROUTINE POLINT(XA,YA,N,X,Y,DY)
        implicit real*8 (a-h,o-z)
      PARAMETER (NMAX=10)
      DIMENSION XA(N),YA(N),C(NMAX),D(NMAX)
      NS=1
      DIF=ABS(X-XA(1))
      DO 11 I=1,N
        DIFT=ABS(X-XA(I))
        IF (DIFT.LT.DIF) THEN
          NS=I
          DIF=DIFT
        ENDIF
        C(I)=YA(I)
        D(I)=YA(I)
11    CONTINUE
      Y=YA(NS)

      NS=NS-1
      DO 13 M=1,N-1
        DO 12 I=1,N-M
          HO=XA(I)-X
          HP=XA(I+M)-X
          W=C(I+1)-D(I)
          DEN=HO-HP
          IF(DEN.EQ.0.)PAUSE
          DEN=W/DEN
          D(I)=HP*DEN
          C(I)=HO*DEN
12      CONTINUE
        IF (2*NS.LT.N-M)THEN
          DY=C(NS+1)
        ELSE
          DY=D(NS)
          NS=NS-1
        ENDIF
        Y=Y+DY
13    CONTINUE
      RETURN
      END
```

The SCCEX code has since been incorporated into a new code, ERSPAC. No further work will be performed in SCCEX code.

N. Fridlind

2/29/97

I have reviewed this scientific notebook and find it in compliance with QAP-001. There is sufficient information regarding procedures used for conducting tests, acquiring and analyzing data so that another qualified individual could repeat the activity.

N. Sridhar 2/24/97

Narasi Sridhar
Manager, Engineered Barrier System and Waste Solidification System