REFERENCES

Anderson, J.J. Computational Fluid Dynamics. International Ed. McGraw-Hill. (1995).

Bird, R.B., Stewart, W.E., and Lightfoot, E.N. Transport Phenomena. Wiley International Ed.(1960).

Cengel, A.Y. and Boles, M.A. Thermodynamics An Engineering Approach. Fourth Ed. McGraw-Hill. (2002).

Crowe C., Sommerfeld M., and Tsuji, Y. Multiphase Flows with Droplets and Particles. Florida: CRC Press. (1998).

Daizo Kunii and Octave Levenspiel. Fluidization Engineering-2nd ed. USA: Butterworth-Heinemann. A division of Reed Publishing. (1991).

Debenedetti, P.G. and Tom, J.W. Rapid Expansion of Supercritical Solutions (RESS): Fundamentals and applications. Fluid Phase Equilibria. 82 (1993): 311-321.

Ferziger, J.H. and Peric M. Computational Methods for Fluid Dynamics. Second Revised Ed. Springer. (1999).

Imrie, B. W. Compressible Fluid Flow. London: Butterworths. (1973).

Ksibi, H. and Tenaud, C. Numerical simulation of rapid expansion of supercritical fluid. Eur. J. Mech. B/Fluids. 15(1996): 569-596.

Miquel F. Llop, Joaquim Casal and Josep Arnaldos. Expansion of gas-solid fluidized beds at pressure and high temperature. Powder Technology. 107(2000): 212-225.

Pallado, P. and Reverchon, E. Hydrodynamic Modeling of RESS Process. The Journal of Supercritical Fluids. 9(1996): 216-221.

Patankar, S.V. Numerical Heat Transfer and Fluid Flow. Hemisphere Publishing Corporation. (1980).

Rhodes M. Introduction to Particle Technology. England. Chichester: John Willey & Sons. (1998).

Salatino, P., Poletto, M. and Massimilla, L. Fluidization of solids with CO_2 at pressures and temperatures ranging from ambient to nearly critical conditions. Chemical Engineering Science. 48(1993): 617-621.

Salatino, P. and Marzocchella, A. Fluidization of solids with CO_2 at pressures from ambient to supercritical. AIChE Journal. 46(5)(2000): 901-910.

Tsutsumi, A., Hasegawa, H., and Mineo, T. Mechanism of particle coating granulation with RESS process in a fluidized bed. Powder Technology. 118(2001): 229-235.

Versteeg, H.K. and Malalasekera, W. An introduction to computational fluid dynamics: The finite volume method. Longman Scientific & Technical. (1998).

Walas, Stanley M. Phase equilibria in chemical engineering. USA. Butterworth Publishing. (1985).

Welty, James R., Wicks, Charles E. and Wilson, Robert E. Fundamentals of Momentum, Heat, and Mass Trasfer. 3th ed. England. Chichester: John Willey & Sons. (1984).

APPENDICES

APPENDIX A

THE COMPUTER PROGRAM
BY EXPLICIT FINITE DIFFERENCE TECHNIQUE

The Computer Program by Explicit Finite Difference Technique

The computer program, which use explicit finite difference technique, for determination of the velocity profiles, the temperature and pressure profiles of rapid expansion of supercritical solution process, especially near the nozzle area, are presented.

Table A.1 Objective of Subroutines

Subroutines	Objective
MAIN PROGRAM	Open files, Close files, Call Subroutines
CONST	Input all essential constant values, the grid size, maximum time step, iteration number and basic properties of the flow
IC	Input initial condition of system in the first interval time
IC_DATA	Input initial condition of system from the last interval
BC	Input boundary condition of system
DYNVIS	Evaluate the viscosity of the fluid in every single location at time 't' according to the Sutherland's law
THERMC	Evaluate thermal conductivity of the fluid in every single location at time 't' from the Prandtl number
MAC	Calculate various properties of the fluid : velocity, pressure, temperature and density of the fluid in every single location at time 't'
EOS	Calculate the density of fluid with the Equation of State
SORT	Sort the values of parameter

Table A.2 Description of Parameters

Parameters	Description
keizoku	Running status : continue running(1) or first time(0)
nmax	Maximum time steps of calculation
delz	Grid size in z-direction
dely	Grid size in y-direction
delt	Value of time step
R	Gas constant
vo	Inlet velocity of fluid
Gram	Ratio of specific heat
T0	Inlet temperature of fluid
P0	Inlet pressure of fluid
Visrf	Reference viscosity of fluid at reference temperature
Trf	Reference temperature fluid
Pr	Prandtl number
rho	Density of fluid
T	Temperature of fluid
Pre	Pressure of fluid
vy	Y-direction velocity of fluid
vz	Z-direction velocity of fluid
visco	Viscosity of fluid
k	Thermal conductivity of fluid

Table A.2 Description of Parameters (continue)

Parameters	Description
Tc	Critical temperature of CO_2
Pc	Critical pressure of CO_2
Zc	Critical compressible factor of CO_2
Cv	Heat capacity of fluid at constant volume
Cp	Heat capacity of fluid at constant pressure

```
**************************************************************
PROGRAM FOR RESS PROCESS USING EXPLICITFINITE DIFFERENCE
**************************************************************
**************************************************************
*---+----10---+----20---+----30---+----40---+----50---+----60---+----70-
**************************************************************
MAIN PROGRAM
IMPLICIT NONE
integer nmax,n,keizoku
common/timest/n,nmax
common/running/keizoku
open(unit=1,file='E:\ress\ress01.txt',status='unknown')
open(unit=2,file='E:\ress\ress02.txt',status='unknown')
open(unit=3,file='E:\ress\ress03.txt',status='unknown')
open(unit=4,file='E:\ress\ress04.txt',status='unknown')
open(unit=5,file='E:\ress\rs01.txt',status='unknown')
open(unit=6,file='E:\ress\rs02.txt',status='unknown')
open(unit=7,file='E:\ress\rs03.txt',status='unknown')
```

```
open(unit=8,file='E:\ress\rs04.txt',status='unknown')
open(unit=13,file='E:\ress\checktemp.txt',status='unknown')
open(unit=14,file='E:\ress\checkpre.txt',status='unknown')
open(unit=15,file='E:\ress\checkrho.txt',status='unknown')
call CONST
call EOS
if (keizoku.eq.0) then
        call IC
else
        call IC_DATA
end if
DO 10 n=1,nmax
        if (mod(n,10).eq.0) then
                write(*,*) 'itime=',n
        end if
        call BC
        call DYNVIS
        call THERMC
        call MAC
10 continue
close(unit=1)
close(unit=2)
close(unit=3)
close(unit=4)
close(unit=5)
close(unit=6)
close(unit=7)
close(unit=8)
close(unit=13)
```

```
close(unit=14)
close(unit=15)
stop
end
```

****

Subroutine CONST

```
IMPLICIT NONE
integer cmax,qmax,nmax,n,keizoku
parameter(cmax=25,qmax=101)
double precision delt,dely,delz,vo,T0,P0,rho0,V0,Pr,
&                R,visrf,Trf,Gram
common/CELLSIZE/dely,delz
common/TIME_STEP/delt
common/CO2_CONST/Pr,visrf,Trf,Gram
common/SUPERFICIAL/vo
common/CO2_IN/T0,P0,rho0,V0
common/CO2/R
common/timest/n,nmax
common/running/keizoku
keizoku=0                          ! continue running(1) or first time(0)
nmax=7000                          ! maximum time steps of calculation
delz=0.40d-4                       ! Grid size in z-direction
dely=0.10d-4                       ! Grid size in y-direction
delt=1.0d-11                       ! value of time step
R=8.314*1000.0/44.01               ! specific gas constant of CO2
vo=10                              ! inlet velocity of CO2
Gram=1.304                         ! ratio of specific heat
```

```
T0=300.0                    ! inlet temperature of CO2
P0=90.0d+5                  ! inlet pressure of CO2
visrf=1.463d-4              ! reference viscosity of CO2 at Trf
Trf=293.0                   ! reference Temperature
Pr=0.710                    ! prandtl number
return
end

**********************************************************************

Subroutine IC
IMPLICIT NONE
integer cmax,qmax,p,q
double precision vy,vz,T,rho,pre
parameter(cmax=25,qmax=101)
common/CO2_DENSITY/rho(0:cmax,0:qmax)
common/CO2_TEMP/T(0:cmax,0:qmax)
common/CO2_VEl/vy(0:cmax,0:qmax),vz(0:cmax,0:qmax)
common/PRESS/pre(0:cmax,0:qmax)
DO 20 q=0,qmax
DO 20 p=0,cmax
        rho(p,q)= 5.30778d-3
        T(p,q)= 298.0
        pre(p,q)= 1.0d+5
        vy(p,q)= 0.0
        vz(p,q)= 0.0
20 continue
return
end
```

```
Subroutine IC_DATA
IMPLICIT NONE
integer cmax,qmax,p,q
double precision vy,vz,T,rho,pre
parameter(cmax=25,qmax=101)

common/CO2_DENSITY/rho(0:cmax,0:qmax)
common/CO2_TEMP/T(0:cmax,0:qmax)
common/CO2_VEl/vy(0:cmax,0:qmax),vz(0:cmax,0:qmax)
common/PRESS/pre(0:cmax,0:qmax)
open(unit=9,file='E:\ress\rss01.txt',status='unknown')
open(unit=10,file='E:\ress\rss02.txt',status='unknown')
open(unit=11,file='E:\ress\rss03.txt',status='unknown')
open(unit=12,file='E:\ress\rss04.txt',status='unknown')
DO 30 q=1,qmax-1
DO 40 p=1,cmax-1
        read(9,*) T(p,q)
        read(10,*) pre(p,q)
        read(11,*) rho(p,q)
        read(12,*) vy(p,q),vz(p,q)
40 continue
30 continue
close(unit=9)
close(unit=10)
close(unit=11)
close(unit=12)

return
end
```

```
Subroutine BC
IMPLICIT NONE
integer cmax,p,q,qmax
double precision pre,vo,vy,vz,P0,T0,rho0,rho,T,V0
parameter(cmax=25,qmax=101)
common/SUPERFICIAL/vo
common/CO2_VEl/vy(0:cmax,0:qmax),vz(0:cmax,0:qmax)
common/PRESS/pre(0:cmax,0:qmax)
common/CO2_IN/T0,P0,rho0,V0
common/CO2_DENSITY/rho(0:cmax,0:qmax)
common/CO2_TEMP/T(0:cmax,0:qmax)
DO 50 q=21,qmax
        T(0,q)=T(1,q)
        pre(0,q)=pre(1,q)
        vy(0,q)=vy(1,q)
        vz(0,q)=vz(1,q)
        rho(0,q)=rho(1,q)
        T(cmax,q)=T(cmax-1,q)
        pre(cmax,q)=pre(cmax-1,q)
        vy(cmax,q)=vy(cmax-1,q)
        vz(cmax,q)=vz(cmax-1,q)
        rho(cmax,q)=rho(cmax-1,q)
 50 continue
 DO 90 p=0,cmax
        T(p,qmax)=T(p,qmax-1)
        pre(p,qmax)=pre(p,qmax-1)
        vy(p,qmax)=vy(p,qmax-1)
        vz(p,qmax)=vz(p,qmax-1)
        rho(p,qmax)=rho(p,qmax-1)
```

```
90 continue
DO 100 q=0,20
DO 110 p=0,10
        T(p,q)=T0
        pre(p,q)=P0
        vy(p,q)=0.0
        vz(p,q)=0.0
        rho(p,q)=rho0
110 continue
DO 120 p=15,cmax
        T(p,q)=T0
        pre(p,q)=P0
        vy(p,q)=0.0
        vz(p,q)=vo
        rho(p,q)=rho0
120 continue
100 continue
DO 130 p=11,14
        T(p,0)=T0
        pre(p,0)=P0
        vy(p,0)=0.0
        vz(p,0)=0.0
        rho(p,0)=rho0
130 continue

return
end
```

```
Subroutine DYNVIS
IMPLICIT NONE
double precision visco,T,visrf,Trf,Gram,Pr
integer cmax,p,q,qmax
parameter(cmax=25,qmax=101)
common/CO2_VIS/visco(0:cmax,0:qmax)
common/CO2_CONST/Pr,visrf,Trf,Gram
common/CO2_TEMP/T(0:cmax,0:qmax)
DO 110 p=0,cmax
DO 110 q=0,qmax
      visco(p,q)=visrf*((T(p,q)/Trf)**1.5)*((Trf+110.0)/(T(p,q)+110.0))
110 continue

return
end

*****************************************************************

Subroutine THERMC
IMPLICIT NONE
double precision T,Cp,Cv,Gram,a,b,c,d,k,visrf,Trf,Pr,visco
integer cmax,p,q,qmax
parameter(cmax=25,qmax=101)
common/CO2_TEMP/T(0:cmax,0:qmax)
common/CO2_THER/k(0:cmax,0:qmax)
common/CO2_CONST/Pr,visrf,Trf,Gram
common/CO2_VIS/visco(0:cmax,0:qmax)
common/CO2_HEAT/Cv(0:cmax,0:qmax),Cp(0:cmax,0:qmax)
a=22.26
b=5.981d-2
```

```
c=-3.501d-5
d=7.469d-9
DO 120 p=0,cmax
DO 120 q=0,qmax
      Cp(p,q)=(a+(b*T(p,q))+(c*T(p,q)**2.0)+(d*T(p,q)**3.0))/44.01*1000
      Cv(p,q)=Cp(p,q)/Gram
      k(p,q)=visco(p,q)*Cp(p,q)/Pr
120 continue

return
end

*****************************************************************

Subroutine MAC
IMPLICIT NONE
integer cmax,qmax,p,q,nmax,n
parameter(cmax=25,qmax=101)
dimension A1(0:cmax,0:qmax),A2(0:cmax,0:qmax)
dimension A3(0:cmax,0:qmax),A5(0:cmax,0:qmax)
dimension B1(0:cmax,0:qmax),B2(0:cmax,0:qmax)
dimension B3(0:cmax,0:qmax),B5(0:cmax,0:qmax)
dimension C1(0:cmax,0:qmax),C2(0:cmax,0:qmax)
dimension C3(0:cmax,0:qmax),C5(0:cmax,0:qmax)
dimension U1(0:cmax,0:qmax),U2(0:cmax,0:qmax)
dimension U3(0:cmax,0:qmax),U5(0:cmax,0:qmax)
dimension dU1(0:cmax,0:qmax),dU2(0:cmax,0:qmax)
dimension dU3(0:cmax,0:qmax),dU5(0:cmax,0:qmax)
double precision delt,dely,delz,rho,R,vy,vz,T,Cv,Cp,k,pre,vo,
&                A1,A2,A3,A5,B1,B2,B3,B5,U1,U2,U3,U5,
```

```
&                C1,C2,C3,C5,dU1,dU2,dU3,dU5
common/timest/n,nmax
common/CELLSIZE/dely,delz
common/CO2_THER/k(0:cmax,0:qmax)
common/CO2_DENSITY/rho(0:cmax,0:qmax)
common/PRESS/pre(0:cmax,0:qmax)
common/CO2_VEl/vy(0:cmax,0:qmax),vz(0:cmax,0:qmax)
common/CO2_TEMP/T(0:cmax,0:qmax)
common/CO2_HEAT/Cv(0:cmax,0:qmax),Cp(0:cmax,0:qmax)
common/TIME_STEP/delt
common/CO2/R
common/SUPERFICIAL/vo
DO 130 q=1,qmax-1
DO 140 p=1,cmax-1

******** Predicted *********
      U1(p,q)=rho(p,q)
      U2(p,q)=vy(p,q)*rho(p,q)
      U3(p,q)=vz(p,q)*rho(p,q)
      U5(p,q)=(rho(p,q)*T(p,q)*Cv(p,q))+(rho(p,q)
&                *(vy(p,q)**2.0+vz(p,q)**2.0)/2.0)

      A1(p,q)=-(rho(p,q)*(vy(p+1,q)-vy(p,q))/dely)
&              -(vy(p,q)*(rho(p+1,q)-rho(p,q))/dely)
&              -(rho(p,q)*(vz(p,q+1)-vz(p,q))/delz)
&              -(vz(p,q)*(rho(p,q+1)-rho(p,q))/delz)
      A2(p,q)=-(2.0*rho(p,q)*vy(p,q)*(vy(p+1,q)-vy(p,q))/dely)
&              -((vy(p,q)**2.0)*(rho(p+1,q)-rho(p,q))/dely)
&              -(rho(p,q)*R*(T(p+1,q)-T(p,q))/dely)
```

```
&                -(T(p,q)*R*(rho(p+1,q)-rho(p,q))/dely)
&                -(rho(p,q)*vz(p,q)*(vy(p,q+1)-vy(p,q))/delz)
&                -(rho(p,q)*vy(p,q)*(vz(p,q+1)-vz(p,q))/delz)
&                -(vy(p,q)*vz(p,q)*(rho(p,q+1)-rho(p,q))/delz)

      A3(p,q)=-(rho(p,q)*vy(p,q)*(vz(p+1,q)-vz(p,q))/dely)
&                -(rho(p,q)*vz(p,q)*(vy(p+1,q)-vy(p,q))/dely)
&                -(vy(p,q)*vz(p,q)*(rho(p+1,q)-rho(p,q))/dely)
&                -(2.0*rho(p,q)*vz(p,q)*(vz(p,q+1)-vz(p,q))/delz)
&                -((vz(p,q)**2.0)*(rho(p,q+1)-rho(p,q))/delz)
&                -(rho(p,q)*R*(T(p,q+1)-T(p,q))/delz)
&                -(T(p,q)*R*(rho(p,q+1)-rho(p,q))/delz)

      A5(p,q)=-(rho(p,q)*vy(p,q)*T(p,q)*(Cv(p+1,q)-Cv(p,q))/dely)
&                -(rho(p,q)*vy(p,q)*Cv(p,q)*(T(p+1,q)-T(p,q))/dely)
&                -(rho(p,q)*T(p,q)*Cv(p,q)*(vy(p+1,q)-vy(p,q))/dely)
&                -(vy(p,q)*Cv(p,q)*T(p,q)*(rho(p+1,q)-rho(p,q))/dely)
&                -((1.5)*rho(p,q)*(vy(p,q)**2.0)*(vy(p+1,q)-vy(p,q))/dely)
&                -((0.5)*(vy(p,q)**3.0)*(rho(p+1,q)-rho(p,q))/dely)
&                -((0.5)*rho(p,q)*(vz(p,q)**2.0)*(vy(p+1,q)-vy(p,q))/dely)
&                -((0.5)*vy(p,q)*(vz(p,q)**2.0)*(rho(p+1,q)-rho(p,q))/dely)
&                -(rho(p,q)*vy(p,q)*vz(p,q)*(vz(p+1,q)-vz(p,q))/dely)
&                -(rho(p,q)*vy(p,q)*R*(T(p+1,q)-T(p,q))/dely)
&                -(rho(p,q)*T(p,q)*R*(vy(p+1,q)-vy(p,q))/dely)
&                -(vy(p,q)*T(p,q)*R*(rho(p+1,q)-rho(p,q))/dely)
&                +(k(p,q)*(T(p+1,q)-T(p,q))/(dely**2.0))
&                -(rho(p,q)*vz(p,q)*T(p,q)*(Cv(p,q+1)-Cv(p,q))/delz)
&                -(rho(p,q)*vz(p,q)*Cv(p,q)*(T(p,q+1)-T(p,q))/delz)
&                -(rho(p,q)*T(p,q)*Cv(p,q)*(vz(p,q+1)-vz(p,q))/delz)
```

```
&                    -(vz(p,q)*Cv(p,q)*T(p,q)*(rho(p,q+1)-rho(p,q))/delz)
&                    -((0.5)*vz(p,q)*(vy(p,q)**2.0)*(rho(p,q+1)-rho(p,q))/delz)
&                    -((0.5)*rho(p,q)*(vy(p,q)**2.0)*(vz(p,q+1)-vz(p,q))/delz)
&                    -(vz(p,q)*vy(p,q)*rho(p,q)*(vy(p,q+1)-vy(p,q))/delz)
&                    -((0.5)*(vz(p,q)**3.0)*(rho(p,q+1)-rho(p,q))/delz)
&                    -((1.5)*rho(p,q)*(vz(p,q)**2.0)*(vz(p,q+1)-vz(p,q))/delz)
&                    -(rho(p,q)*vz(p,q)*R*(T(p,q+1)-T(p,q))/delz)
&                    -(rho(p,q)*T(p,q)*R*(vz(p,q+1)-vz(p,q))/delz)
&                    -(vz(p,q)*T(p,q)*R*(rho(p,q+1)-rho(p,q))/delz)
&                    +(k(p,q)*(T(p,q+1)-T(p,q))/(delz**2.0))

******* U' at t + dt  *******
        U1(p,q)=U1(p,q)+(A1(p,q)*delt)
        U2(p,q)=U2(p,q)+(A2(p,q)*delt)
        U3(p,q)=U3(p,q)+(A3(p,q)*delt)
        U5(p,q)=U5(p,q)+(A5(p,q)*delt)
        rho(p,q)=U1(p,q)
        vy(p,q)=U2(p,q)/U1(p,q)
        vz(p,q)=U3(p,q)/U1(p,q)
        T(p,q)=((U5(p,q)/U1(p,q))-(0.5*(vy(p,q)**2
 &                  +vz(p,q)**2)))/Cv(p,q)
        pre(p,q)=rho(p,q)*R*T(p,q)

 ******* Rearward Difference *******
        C1(p,q)=-(rho(p,q)*(vy(p,q)-vy(p-1,q))/dely)
 &                  -(vy(p,q)*(rho(p,q)-rho(p-1,q))/dely)
 &                  -(rho(p,q)*(vz(p,q)-vz(p,q-1))/delz)
 &                  -(vz(p,q)*(rho(p,q)-rho(p,q-1))/delz)
```

```
      C2(p,q)=-(2.0*rho(p,q)*vy(p,q)*(vy(p,q)-vy(p-1,q))/dely)
     &                -((vy(p,q)**2.0)*(rho(p,q)-rho(p-1,q))/dely)
     &                -(rho(p,q)*R*(T(p,q)-T(p-1,q))/dely)
     &                -(T(p,q)*R*(rho(p,q)-rho(p-1,q))/dely)
     &                -(rho(p,q)*vz(p,q)*(vy(p,q)-vy(p,q-1))/delz)
     &                -(rho(p,q)*vy(p,q)*(vz(p,q)-vz(p,q-1))/delz)
     &                -(vy(p,q)*vz(p,q)*(rho(p,q)-rho(p,q-1))/delz)

      C3(p,q)=-(rho(p,q)*vy(p,q)*(vz(p,q)-vz(p-1,q))/dely)
     &                -(rho(p,q)*vz(p,q)*(vy(p,q)-vy(p-1,q))/dely)
     &                -(vy(p,q)*vz(p,q)*(rho(p,q)-rho(p-1,q))/dely)
     &                -(2.0*rho(p,q)*vz(p,q)*(vz(p,q)-vz(p,q-1))/delz)
     &                -((vz(p,q)**2.0)*(rho(p,q)-rho(p,q-1))/delz)
     &                -(rho(p,q)*R*(T(p,q)-T(p,q-1))/delz)
     &                -(T(p,q)*R*(rho(p,q)-rho(p,q-1))/delz)

      C5(p,q)=-(rho(p,q)*vy(p,q)*T(p,q)*(Cv(p,q)-Cv(p-1,q))/dely)
     &                -(rho(p,q)*vy(p,q)*Cv(p,q)*(T(p,q)-T(p-1,q))/dely)
     &                -(rho(p,q)*T(p,q)*Cv(p,q)*(vy(p,q)-vy(p-1,q))/dely)
     &                -(vy(p,q)*Cv(p,q)*T(p,q)*(rho(p,q)-rho(p-1,q))/dely)
     &                -((1.5)*rho(p,q)*(vy(p,q)**2.0)*(vy(p,q)-vy(p-1,q))/dely)
     &                -((0.5)*(vy(p,q)**3.0)*(rho(p,q)-rho(p-1,q))/dely)
     &                -((0.5)*rho(p,q)*(vz(p,q)**2.0)*(vy(p,q)-vy(p-1,q))/dely)
     &                -((0.5)*vy(p,q)*(vz(p,q)**2.0)*(rho(p,q)-rho(p-1,q))/dely)
     &                -(rho(p,q)*vy(p,q)*vz(p,q)*(vz(p,q)-vz(p-1,q))/dely)
     &                -(rho(p,q)*vy(p,q)*R*(T(p,q)-T(p-1,q))/dely)
     &                -(rho(p,q)*T(p,q)*R*(vy(p,q)-vy(p-1,q))/dely)
     &                -(vy(p,q)*T(p,q)*R*(rho(p,q)-rho(p-1,q))/dely)
     &                +(k(p,q)*(T(p,q)-T(p-1,q))/(dely**2.0))
```

```
&                -(rho(p,q)*vz(p,q)*T(p,q)*(Cv(p,q)-Cv(p,q-1))/delz)
&                -(rho(p,q)*vz(p,q)*Cv(p,q)*(T(p,q)-T(p,q-1))/delz)
&                -(rho(p,q)*T(p,q)*Cv(p,q)*(vz(p,q)-vz(p,q-1))/delz)
&                -(vz(p,q)*Cv(p,q)*T(p,q)*(rho(p,q)-rho(p,q-1))/delz)
&                -((0.5)*vz(p,q)*(vy(p,q)**2.0)*(rho(p,q)-rho(p,q-1))/delz)
&                -((0.5)*rho(p,q)*(vy(p,q)**2.0)*(vz(p,q)-vz(p,q-1))/delz)
&                -(vz(p,q)*vy(p,q)*rho(p,q)*(vy(p,q)-vy(p,q-1))/delz)
&                -((0.5)*(vz(p,q)**3.0)*(rho(p,q)-rho(p,q-1))/delz)
&                -((1.5)*rho(p,q)*(vz(p,q)**2.0)*(vz(p,q)-vz(p,q-1))/delz)
&                -(rho(p,q)*vz(p,q)*R*(T(p,q)-T(p,q-1))/delz)
&                -(rho(p,q)*T(p,q)*R*(vz(p,q)-vz(p,q-1))/delz)
&                -(vz(p,q)*T(p,q)*R*(rho(p,q)-rho(p,q-1))/delz)
&                +(k(p,q)*(T(p,q)-T(p,q-1))/(delz**2.0))

******** (dU/dt)av ***********
        dU1(p,q)=(A1(p,q)+C1(p,q))/2.0
        dU2(p,q)=(A2(p,q)+C2(p,q))/2.0
        dU3(p,q)=(A3(p,q)+C3(p,q))/2.0
        dU5(p,q)=(A5(p,q)+C5(p,q))/2.0

******** Corrected ********
        U1(p,q)=U1(p,q)+(dU1(p,q)*delt)
        U2(p,q)=U2(p,q)+(dU2(p,q)*delt)
        U3(p,q)=U3(p,q)+(dU3(p,q)*delt)
        U5(p,q)=U5(p,q)+(dU5(p,q)*delt)
        rho(p,q)=U1(p,q)
        vy(p,q)=U2(p,q)/U1(p,q)
        vz(p,q)=U3(p,q)/U1(p,q)
        T(p,q)=((U5(p,q)/U1(p,q))-(0.5*(vy(p,q)**2.0
```

```
&                    +vz(p,q)**2.0)))/Cv(p,q)
if (mod(n,500).eq.0) then
        write(*,*) T(p,q)
end if
if (mod(n,1000).eq.0) then
        write(1,*) T(p,q),p,q
        write(2,*) pre(p,q),p,q
        write(3,*) rho(p,q),p,q
        write(4,*) vy(p,q),vz(p,q),p,q
end if
if (n.EQ.nmax) then
        write(5,*) T(p,q),p,q
        write(6,*) pre(p,q),p,q
        write(7,*) rho(p,q),p,q
        write(8,*) vy(p,q),vz(p,q),p,q
end if
140 continue
130 continue
   DO 300 q=1,qmax-1
   DO 400 p=12,12
        if (mod(n,500).eq.0) then
              write(13,*) T(p,q),p,q
              write(14,*) pre(p,q),p,q
              write(15,*) rho(p,q),p,q
         end if
400   continue
300   continue
return
end
```

```
Subroutine EOS
IMPLICIT NONE
double precision aa,bb,cc,dd,f,g,h,pp,qq,ppp,qqq,rr,rrr,
&                v3,z,ss,sss,y,u,V,w,v1,v2,X,R,Tc,Pc,V0,
&                Zc,rho0,T0,P0,M,lpa,lpo,A,B,xx,P
common/CO2_IN/T0,P0,rho0,V0
common/VOL_fluid/v1,v2,v3,V
R=8.3144                        ! Gas constant
Tc=304.2                        ! Critical temperature of CO2
Pc=72.9d+5                      ! Critical pressure of CO2
Zc=0.277                        ! Critical compressible of CO2
lpa=3.996
lpo=1.1717
M=44.01/1000.0
xx=1.0/3.0
P=P0/1.0d+5
W=(0.2905-Zc)/0.085
X=(1.0+(0.48508+(1.5517*W)-(0.15613*W**2))*(1-(T0/Tc)**0.5))**2.0
A=0.42747*(R**2.0)*(Tc**2.0)/Pc
B=0.08664*R*Tc/Pc
aa=P
bb=-(R*T0)
cc=-(P*B**2.0)-(B*R*T0)+(A*X)
dd=-(A*X*B)
f=((3.0*cc/aa)-(bb**2.0/aa**2.0))/3.0
g=((2.0*bb**3.0/aa**3.0)-(9.0*bb*cc/aa**2.0)+(27.0*dd/aa))/27.0
h=((g**2.0/4.0)+(f**3.0/27.0))
if (h.LT.0) then
        pp=(((g**2.0)/4.0)-h)**0.5
```

```
        qq=pp**xx
        z=-(g/(2.0*pp))
        rr=acos(z)
        ss=qq*(-1.0)
        y=cos(rr/3.0)
        u=(3**0.5)*sin(rr/3.0)
        w=(bb/(3.0*aa))*(-1.0)
        v1=(2.0*qq*cos(rr/3.0))-(bb/(3.0*aa))
        v2=(ss*(y+u))+w
        v3=(ss*(y-u))+w
        call SORT
else if (h.GT.0) then
        ppp=(h**0.5)-(g/2.0)
        qqq=ppp**xx
        rrr=(h**0.5)+(g/2.0)
        if (rrr.GT.0) then
                sss=rrr**xx
                V=qqq-sss-(bb/(3.0*aa))
        else
                sss=(-rrr)**xx
                V=qqq-sss-(bb/(3.0*aa))
        end if
else
        V=-(dd/aa)**xx
end if
V0=V
rho0=M/V
 return
end
```

```
Subroutine SORT
IMPLICIT NONE
double precision v1,v2,v3,V
common/VOL_fluid/v1,v2,v3,V
if (v1.LT.0.0) go to 150
        V=v1
        if (v2.LT.V) then
                V=v2
        end if
        if (v3.LT.V) then
                V=v3
end if
150     if (v2.GT.0.0) then
                if (v3.GT.0.0) then
                        V=v2
                        if (v3.LT.V) then
                                V=v3
                        end if
                end if
        else if (v3.GT.0.0) then
                V=v3
        else
                write(1,*) ' Error .. have no positive root'
        end if
return
end
```

APPENDIX B

THE COMPUTER PROGRAM
BY IMPLICIT FINITE DIFFERENCE TECHNIQUE

The Computer Program by Implicit Finite Difference Technique

The computer program, which use implicit finite difference technique, for determination of the velocity profiles, the temperature and pressure profiles of rapid expansion of supercritical solution process, especially near the nozzle area, are presented.

Table B.1 Objective of Subroutines

Subroutines	Objective
MAIN PROGRAM	Call Subroutines
O_FILE	Open files
C_FLIE	Close files
CONST	Input all essential constant values, the grid size, maximum time step, iteration number and basic properties of the flow
INITIAL	Input initial condition of system in the first interval time
INITIAL_SAVE	Input initial condition of system from the last interval
FLUID	Call subroutines for calculating the fluid flow
BC	Input boundary condition of system
MEM	Storage the value of fluid properties: temperature, pressure, density and velocity
MEM_PRE	Storing Previous Pressure values for next loop
EOS	Calculate the density of fluid with the Equation of State
SORT	Sort the values of parameter

Table B.1 Objective of Subroutines (continue)

Subroutines	Objective
AP_BC	Calculation of Coefficient (ap) on Boundary
MOMENTUM	Calculation of fluid velocity
CON_DEN	Calculation of fluid density
HEAT	Evaluate the heat capacity and thermal conductivity of the fluid in every single location at time 't'
ENERGY	Calculation of fluid temperature
PRESSURE	Calculation of fluid pressure
ERROR	Checking the conservation of mass in each cell
TDMA	TDMA (Tri-Diagonal Matrix Algorithm)
SAVEDATA	Saving the calculated fluid properties

Table B.2 Description of Parameters

Parameters	Description
keizoku	Running status : continue running(1) or first time(0)
interval	Maximum time steps of calculation
delx	Grid size in x-direction
dely	Grid size in y-direction
delt	Value of time step
R	Gas constant

Table B.2 Description of Parameters (continue)

Parameters	Description
vo	Inlet velocity of fluid
Gram	Ratio of specific heat
Tin	Inlet temperature of fluid
prein	Inlet pressure of fluid
rhoin	Inlet density of fluid
Visrf	Reference viscosity of fluid at reference temperature
Trf	Reference temperature fluid
Pr	Prandtl number
alpha	
rho	Density of fluid
T	Temperature of fluid
Pre	Pressure of fluid
vy	Y-direction velocity of fluid
vx	X-direction velocity of fluid
visco	Viscosity of fluid
k	Thermal conductivity of fluid
Tc	Critical temperature of CO_2
Pc	Critical pressure of CO_2
Zc	Critical compressible factor of CO_2

Table B.2 Description of Parameters (continue)

Parameters	Description
Cv	Heat capacity of fluid at constant volume
Cp	Heat capacity of fluid at constant pressure

```
**************************************************************
  PROGRAM FOR RESS PROCESS USING IMPLICITFINITE DIFFERENCE
**************************************************************
---+----10---+----20---+----30---+----40---+----50---+----60---+----70-
**************************************************************
Main Program
implicit none
integer n,interval,keizoku
common/running/keizoku
common/timest/n,interval
call CONSTANT
call O_FILE
if (keizoku.eq.0) then
        call INITIAL
else
        call INITIAL_SAVE
end if
do 100 n=1,interval
        if (mod(n,5).eq.0) then
                write (*,*) 'interval time =',n
```

```
        end if
        call FLUID
        call SAVEDATA
100     continue
call C_FILE

stop
end

**************************************************************

Subroutine O_FILE
implicit none
open(unit=1,file='E:\ress\ress01.txt',status='unknown')
open(unit=2,file='E:\ress\ress02.txt',status='unknown')
open(unit=3,file='E:\ress\ress03.txt',status='unknown')
open(unit=4,file='E:\ress\ress04.txt',status='unknown')
open(unit=5,file='E:\ress\rs01.txt',status='unknown')
open(unit=6,file='E:\ress\rs02.txt',status='unknown')
open(unit=7,file='E:\ress\rs03.txt',status='unknown')
open(unit=8,file='E:\ress\rs04.txt',status='unknown')
open(unit=13,file='E:\ress\checktemp.txt',status='unknown')
open(unit=14,file='E:\ress\checkpre.txt',status='unknown')
open(unit=15,file='E:\ress\checkrho.txt',status='unknown')

return
end
```

```
Subroutine C_FILE
implicit none
close(unit=1)
close(unit=2)
close(unit=3)
close(unit=4)
close(unit=5)
close(unit=6)
close(unit=7)
close(unit=8)
close(unit=13)
close(unit=14)
close(unit=15)

return
end

**************************************************************

Subroutine CONSTANT
implicit none
integer xmax,ymax,interval,keizoku,n
parameter(xmax=25,ymax=521)
double precision delt,delx,dely,vo,Tin,prein,alpha,
&                R,Pr,visrf,Trf,Gram,rhoin

common/CELLSIZE/delx,dely
common/TIME_STEP/delt
common/SUPERFICIAL/vo
```

```
common/CO2_IN/Tin,prein,rhoin
common/CO2/R
common/timest/n,interval
common/running/keizoku
common/ALPHA/alpha
common/CO2_CONST/Pr,visrf,Trf,Gram
interval=15000
keizoku=0
delt=1.0d-13
delx=0.10d-4
dely=0.40d-4
Tin=413.0
prein=260.0d+5
rhoin=0.460395
R=8.314*1000.0/44.01
alpha=0.05
Pr=0.710
visrf=1.463d-4
Trf=293.0
Gram=1.3040
vo=0

return
end
```

```
Subroutine INITIAL
implicit none
integer xmax,ymax,x,y
double precision vx,vy,T,rho,pre
parameter(xmax=25,ymax=521)
common/CO2_DENSITY/rho(0:xmax,0:ymax)
common/CO2_TEMP/T(0:xmax,0:ymax)
common/CO2_VEl/vx(0:xmax,0:ymax),vy(0:xmax,0:ymax)
common/PRESS/pre(0:xmax,0:ymax)
DO 100 y=1,ymax-1
DO 200 x=1,xmax-1
        rho(x,y)= 5.30778d-3
        T(x,y)= 298.0
        pre(x,y)= 1.0d+5
        vx(x,y)= 0.0
        vy(x,y)= 0.0
200     continue
100     continue
return
end

**************************************************************

Subroutine INITIAL_SAVE
implicit none
integer xmax,ymax,x,y
double precision vx,vy,T,rho,pre
parameter(xmax=25,ymax=521)
common/CO2_DENSITY/rho(0:xmax,0:ymax)
```

```
common/CO2_TEMP/T(0:xmax,0:ymax)
common/CO2_VEl/vx(0:xmax,0:ymax),vy(0:xmax,0:ymax)
common/PRESS/pre(0:xmax,0:ymax)
open(unit=9,file='E:\ress\rss01.txt',status='unknown')
open(unit=10,file='E:\ress\rss02.txt',status='unknown')
open(unit=11,file='E:\ress\rss03.txt',status='unknown')
open(unit=12,file='E:\ress\rss04.txt',status='unknown')
DO 100 y=1,ymax-1
DO 200 x=1,xmax-1
        read(9,*) T(x,y)
        read(10,*) pre(x,y)
        read(11,*) rho(x,y)
        read(12,*) vx(x,y),vy(x,y)
200   continue
100   continue
close(unit=9)
close(unit=10)
close(unit=11)
close(unit=12)
return
end

**************************************************

Subroutine FLUID
implicit none
integer iteration
double precision bbb
common/iterate/iteration
```

```
call BC
call HEAT
call MEM
iteration=0
100   iteration=iteration+1
call AP_BC
call MOMENTUM
call CON_DEN
call HEAT
call ENERGY
if (iteration.eq.1.0) then
        call MEM_PRE
end if
call PRESSURE
call ERROR(bbb)
if (bbb.lt.100.0) go to 200
if (iteration.gt.100000) then
        write(*,*) 'It is Diverge!!'
        stop
end if
call MEM_PRE
call BC
go to 100
200   continue

return
end
```

```
Subroutine BC
implicit none
integer xmax,ymax,x,y
double precision vx,vy,T,rho,pre,vo,Tin,prein,rhoin,dely,delx,delt
parameter(xmax=25,ymax=521)
common/CELLSIZE/delx,dely
common/TIME_STEP/delt
common/SUPERFICIAL/vo
common/CO2_IN/Tin,prein,rhoin
common/CO2_DENSITY/rho(0:xmax,0:ymax)
common/CO2_TEMP/T(0:xmax,0:ymax)
common/CO2_VEl/vx(0:xmax,0:ymax),vy(0:xmax,0:ymax)
common/PRESS/pre(0:xmax,0:ymax)
DO 50 y=21,ymax
        T(0,y)=T(1,y)
        pre(0,y)=pre(1,y)
        vx(0,y)=vx(1,y)
        vy(0,y)=vy(1,y)
        rho(0,y)=rho(1,y)
        T(xmax,y)=T(xmax-1,y)
        pre(xmax,y)=pre(xmax-1,y)
        vx(xmax,y)=vx(xmax-1,y)
        vy(xmax,y)=vy(xmax-1,y)
        rho(xmax,y)=rho(xmax-1,y)
50 continue
DO 90 x=0,xmax
        T(x,ymax)=T(x,ymax-1)
        pre(x,ymax)=pre(x,ymax-1)
        vx(x,ymax)=vx(x,ymax-1)
```

```
        vy(x,ymax)=vy(x,ymax-1)
        rho(x,ymax)=rho(x,ymax-1)
90 continue
DO 100 y=0,20
DO 110 x=0,10
        T(x,y)=Tin
        pre(x,y)=prein
        vx(x,y)=0.0
        vy(x,y)=0.0
        rho(x,y)=rhoin
110 continue
DO 120 x=15,xmax
        T(x,y)=Tin
        pre(x,y)=prein
        vx(x,y)=0.0
        vy(x,y)=0.0
        rho(x,y)=rhoin
120 continue
100 continue
DO 130 x=11,14
        T(x,0)=Tin
        pre(x,0)=prein
        vx(x,0)=0.0
        vy(x,0)=0.0
        rho(x,0)=rhoin
130 continue

return
end
```

```
Subroutine MEM
implicit none
integer xmax,ymax,x,y
double precision vx,vy,T,rho,pre,vx0,vy0,T0,pre0,rho0,
&                Cv,Cv0,Cp,Cp0
parameter(xmax=25,ymax=521)
common/CO2_DENSITY/rho(0:xmax,0:ymax)
common/CO2_TEMP/T(0:xmax,0:ymax)
common/CO2_VEl/vx(0:xmax,0:ymax),vy(0:xmax,0:ymax)
common/PRESS/pre(0:xmax,0:ymax)
common/CO2_HEAT/Cv(0:xmax,0:ymax),Cp(0:xmax,0:ymax)
common/CO2_DENSITY0/rho0(0:xmax,0:ymax)
common/CO2_TEMP0/T0(0:xmax,0:ymax)
common/CO2_VEl0/vx0(0:xmax,0:ymax),vy0(0:xmax,0:ymax)
common/PRESS0/pre0(0:xmax,0:ymax)
common/CO2_HEAT0/Cv0(0:xmax,0:ymax),Cp0(0:xmax,0:ymax)
do 100 y=0,ymax
do 200 x=0,xmax
        T0(x,y)=T(x,y)
        pre0(x,y)=pre(x,y)
        rho0(x,y)=rho(x,y)
        vx0(x,y)=vx(x,y)
        vy0(x,y)=vy(x,y)
        Cv0(x,y)=Cv(x,y)
200   continue
100   continue
return
end
```

```
Subroutine MEM_PRE
implicit none
integer xmax,ymax,x,y
double precision prelp,pre
parameter(xmax=25,ymax=521)
common/PRESS/pre(0:xmax,0:ymax)
common/PRESSLOOP/prelp(0:xmax,0:ymax)
do 100 y=0,ymax
do 200 x=0,xmax
        prelp(x,y)=pre(x,y)
200   continue
100   continue
return
end

**************************************************************

Subroutine EOS
implicit none
integer xmax,ymax,x,y
parameter(xmax=25,ymax=521)
double precision aa,bb,cc,dd,f,g,h,pp,qq,ppp,qqq,rr,rrr,
&                v3,z,ss,sss,yy,u,V,ww,v1,v2,xxx,R,Tc,Pc,
&                Zc,rho,T,pre,M,A,B,xx,P,W
common/CO2_DENSITY/rho(0:xmax,0:ymax)
common/CO2_TEMP/T(0:xmax,0:ymax)
common/PRESS/pre(0:xmax,0:ymax)
common/VOL_fluid/v1,v2,v3,V
```

```
do 100 y=1,ymax-1
do 200 x=1,xmax-1
R=83.144                        ! Gas constant
Tc=304.2                        ! Critical temperature of CO2
Pc=72.9                         ! Critical pressure of CO2
Zc=0.277                        ! Critical compressible of CO2
M=44.01
xx=1.0/3.0
P=pre(x,y)/1.0d+5
W=(0.2905-Zc)/0.085
xxx=(1.0+(0.48508+(1.5517*W)-(0.15613*W**2))*(1-(T(x,y)
&               /Tc)**0.5))**2.0
A=0.42748*(R**2.0)*(Tc**2.0)/Pc
B=0.08664*R*Tc/Pc
aa=P
bb=-(R*T(x,y))
cc=-(P*B**2.0)-(B*R*T(x,y))+(A*xxx)
dd=-(A*xxx*B)
f=((3.0*cc/aa)-(bb**2.0/aa**2.0))/3.0
g=((2.0*bb**3.0/aa**3.0)-(9.0*bb*cc/aa**2.0)+(27.0*dd/aa))/27.0
h=((g**2.0/4.0)+(f**3.0/27.0))
if (h.LT.0) then
        pp=(((g**2.0)/4.0)-h)**0.5
        qq=pp**xx
        z=-(g/(2.0*pp))
        rr=dacos(z)
        ss=qq*(-1.0)
        yy=dcos(rr/3.0)
        u=(3**0.5)*dsin(rr/3.0)
```

```
        ww=(bb/(3.0*aa))*(-1.0)
        v1=(2.0*qq*dcos(rr/3.0))-(bb/(3.0*aa))
        v2=(ss*(y+u))+ww
        v3=(ss*(y-u))+ww
        call SORT
else if (h.GT.0) then
        ppp=(h**0.5)-(g/2.0)
        qqq=ppp**xx
        rrr=(h**0.5)+(g/2.0)
        if (rrr.GT.0) then
                sss=rrr**xx
                V=qqq-sss-(bb/(3.0*aa))
        else
                sss=(-rrr)**xx
                V=qqq-sss-(bb/(3.0*aa))
        end if
else
        V=-(dd/aa)**xx
end if
rho(x,y)=M/V
200     continue
100     continue
return
end
```

```
Subroutine SORT
implicit none
double precision v1,v2,v3,V
common/VOL_fluid/v1,v2,v3,V
if (v1.LT.0.0) go to 100
        V=v1
if (v2.LT.V) then
        V=v2
end if
if (v3.LT.V) then
        V=v3
end if
  100 if (v2.GT.0.0) then
        if (v3.GT.0.0) then
                V=v2
                if (v3.LT.V) then
                        V=v3
                end if
        end if
else if (v3.GT.0.0) then
        V=v3
else
        write(1,*) ' Error .. have no positive root'
end if

return
end
```

```
 Subroutine AP_BC
implicit none
integer x,y,xmax,ymax
parameter(xmax=25,ymax=521)
double precision an,ae,aw,as,apx,apy,fe,fw,fn,fs,
&                vx,vy,dely,delx,delt,rho
common/COEF/apx(0:xmax,0:ymax),apy(0:xmax,0:ymax)
common/CELLSIZE/delx,dely
common/TIME_STEP/delt
common/CO2_VEl/vx(0:xmax,0:ymax),vy(0:xmax,0:ymax)
common/CO2_DENSITY/rho(0:xmax,0:ymax)

********** Coefficient(ap) of left and right hand side ***********
do 100 y=1,ymax-1
        fe=0.0
        fw=0.0
        fn=rho(1,y)*(vy(0,y)+vy(1,y))*0.5*delx
        fs=rho(1,y-1)*(vy(1,y-1)+vy(2,y-1))*0.5*delx
        ae=dmax1(-fe,0.0d+0)
        aw=dmax1(fw,0.0d+0)
        an=dmax1(-fn,0.0d+0)
        as=dmax1(fs,0.0d+0)
        apx(1,y)=ae+aw+an+as+delx*dely/delt
        fe=rho(xmax,y)*(vx(xmax-1,y)+vx(xmax,y))*0.5*dely
        fw=rho(xmax-1,y)*(vx(xmax-2,y)+vx(xmax-1,y))*0.5*dely
        fn=rho(xmax,y)*(vy(xmax-1,y)+vy(xmax,y))*0.5*delx
        fs=rho(xmax,y-1)*(vy(xmax-1,y-1)+vy(xmax,y-1))*0.5*delx
        ae=dmax1(-fe,0.0d+0)
        aw=dmax1(fw,0.0d+0)
```

```
        an=dmax1(-fn,0.0d+0)
        as=dmax1(fs,0.0d+0)
        apx(xmax-1,y)=ae+aw+an+as+delx*dely/delt
100   continue

**********Coefficient (ap) of lower boundary  **********
do 200 x=1,xmax-1
        fe=rho(x,1)*(vx(x,0)+vx(x,1))*0.5*dely
        fw=rho(x-1,1)*(vx(x-1,0)+vx(x-1,1))*0.5*dely
        fn=0.0
        fs=0.0
        ae=dmax1(-fe,0.0d+0)
        aw=dmax1(fw,0.0d+0)
        an=dmax1(-fn,0.0d+0)
        as=dmax1(fs,0.0d+0)
        apy(x,1)=ae+aw+an+as+delx*dely/delt
200   continue
return
end

**********************************************************

Subroutine MOMENTUM
implicit none
integer x,y,xmax,ymax,i,j
parameter(xmax=25,ymax=521)
double precision apx,apy,ae,aw,an,as,bb,aa(4,0:xmax),
&                alpha,delx,dely,delt,vy0,rho,rho0,
&                fe,fw,fn,fs,pre,vx,vx0,vy
```

```
common/CO2_DENSITY/rho(0:xmax,0:ymax)
common/CO2_DENSITY0/rho0(0:xmax,0:ymax)
common/CO2_VEl/vx(0:xmax,0:ymax),vy(0:xmax,0:ymax)
common/CO2_VEl0/vx0(0:xmax,0:ymax),vy0(0:xmax,0:ymax)
common/PRESS/pre(0:xmax,0:ymax)
common/COEF/apx(0:xmax,0:ymax),apy(0:xmax,0:ymax)
common/CELLSIZE/delx,dely
common/TIME_STEP/delt
common/ALPHA/alpha
do 100 y=1,ymax-1
do 200 x=1,xmax-2
      fe=rho(x+1,y)*(vx(x,y)+vx(x+1,y))*0.5*dely
      fw=rho(x,y)*(vx(x,y)+vx(x-1,y))*0.5*dely
      fn=(rho(x,y)+rho(x+1,y)+rho(x,y+1)+rho(x+1,y+1))*0.25
&                 *(vy(x,y)+vy(x+1,y))*0.5*delx
      fs=(rho(x,y)+rho(x+1,y)+rho(x,y-1)+rho(x+1,y-1))*0.25
&                 *(vy(x,y-1)+vy(x+1,y-1))*0.5*delx
      ae=dmax1(-fe,0.0d+0)
      aw=dmax1(fw,0.0d+0)
      an=dmax1(-fn,0.0d+0)
      as=dmax1(fs,0.0d+0)
      apx(x,y)=ae+aw+an+as+((rho0(x,y)+rho0(x+1,y))
&                 *0.5*delx*dely/delt)
      bb=vx0(x,y)*(rho0(x,y)+rho0(x+1,y))*0.5*delx*dely/delt
      aa(1,x)=-aw
      aa(2,x)=apx(x,y)
      aa(3,x)=-ae
      aa(4,x)=an*vx(x,y+1)+as*vx(x,y-1)+bb+
&                 dely*(pre(x,y)-pre(x+1,y))
```

```
200   continue
      aa(1,0)=0.0
      aa(2,0)=1.0
      aa(3,0)=0.0
      aa(4,0)=vx(1,y)
      aa(1,xmax-1)=0.0
      aa(2,xmax-1)=1.0
      aa(3,xmax-1)=0.0
      aa(4,xmax-1)=vx(xmax-1,y)
      call TDMA(0,xmax-1,aa)
do 300 x=1,xmax-2
      vx(x,y)=(1-alpha)*vx(x,y)+alpha*aa(4,x)
300   continue
do 400 x=1,xmax-1
      fe=(rho(x,y)+rho(x+1,y)+rho(x,y+1)+rho(x+1,y+1))*0.25
&                 *(vx(x,y)+vx(x+1,y))*0.5*dely
      fw=(rho(x,y)+rho(x-1,y)+rho(x,y+1)+rho(x-1,y+1))*0.25
&                 *(vx(x-1,y)+vx(x-1,y+1))*0.5*dely
      fn=rho(x,y+1)*(vy(x,y)+vy(x,y+1))*0.5*delx
      fs=rho(x,y)*(vy(x,y)+vy(x,y-1))*0.5*delx
      ae=dmax1(-fe,0.0d+0)
      aw=dmax1(fw,0.0d+0)
      an=dmax1(-fn,0.0d+0)
      as=dmax1(fs,0.0d+0)
      apy(x,y)=ae+aw+an+as+((rho0(x,y)+rho0(x,y+1))*0.5
&                 *delx*dely/delt)
      bb=vy0(x,y)*(rho0(x,y)+rho0(x,y+1))*0.5*delx*dely/delt
      aa(1,x)=-aw
      aa(2,x)=apy(x,y)
```

```
        aa(3,x)=-ae
        aa(4,x)=an*vy(x,y+1)+as*vy(x,y-1)+bb+
&               delx*(pre(x,y)-pre(x,y+1))
400   continue
        aa(1,0)=0.0
        aa(2,0)=1.0
        aa(3,0)=0.0
        aa(4,0)=vy(1,y)
        aa(1,xmax)=0.0
        aa(2,xmax)=1.0
        aa(3,xmax)=0.0
        aa(4,xmax)=vy(xmax,y)
        call TDMA(0,xmax,aa)
do 500 x=1,xmax-1
        vy(x,y)=(1-alpha)*vy(x,y)+alpha*aa(4,x)
500   continue
100   continue
return
end

**********************************************************

Subroutine CON_DEN
implicit none
integer x,y,xmax,ymax,i,j
parameter(xmax=25,ymax=521)
double precision ap(0:xmax,0:ymax),ae,aw,an,as,bb,aa(4,0:xmax),
&                rho0,delx,dely,rho,fe,fw,fn,fs,vx,vy,delt
common/CO2_VEl/vx(0:xmax,0:ymax),vy(0:xmax,0:ymax)
common/CO2_DENSITY0/rho0(0:xmax,0:ymax)
```

```
common/CO2_DENSITY/rho(0:xmax,0:ymax)
common/CELLSIZE/delx,dely
common/TIME_STEP/delt
do 100 y=1,ymax-1
do 200 x=1,xmax-1
        fe=vx(x,y)*0.5*dely
        fw=vx(x-1,y)*0.5*dely
        fn=vy(x,y)*0.5*delx
        fs=vy(x,y-1)*0.5*delx
        ae=dmax1(-fe,0.0d+0)
        aw=dmax1(fw,0.0d+0)
        an=dmax1(-fn,0.0d+0)
        as=dmax1(fs,0.0d+0)
        ap(x,y)=vx(x,y)*0.5*dely-vx(x-1,y)*0.5*dely+vy(x,y)
&                       *0.5*delx-vy(x,y-1)*0.5*delx+(delx*dely/delt)
        bb=rho0(x,y)*delx*dely/delt
        aa(1,x)=-aw
        aa(2,x)=ap(x,y)
        aa(3,x)=-ae
        aa(4,x)=an+as+bb
200   continue
        aa(1,0)=0.0
        aa(2,0)=1.0
        aa(3,0)=0.0
        aa(4,0)=rho(1,y)
        aa(1,xmax)=0.0
        aa(2,xmax)=1.0
        aa(3,xmax)=0.0
        aa(4,xmax)=rho(xmax,y)
```

```
        call TDMA(0,xmax-1,aa)
do 300 x=1,xmax-2
        rho(x,y)=aa(4,x)
300   continue
100     continue
return
end

**************************************************************

Subroutine HEAT
implicit none
integer x,y,xmax,ymax
parameter(xmax=25,ymax=521)
double precision T,Cp,Cv,Gram,a,b,c,d,k,visrf,
&                 Trf,Pr,visco(0:xmax,0:ymax)
common/CO2_TEMP/T(0:xmax,0:ymax)
common/CO2_CONST/Pr,visrf,Trf,Gram
common/CO2_HEAT/Cv(0:xmax,0:ymax),Cp(0:xmax,0:ymax)
common/CO2_THER/k(0:xmax,0:ymax)
a=22.26
b=5.981d-2
c=-3.501d-5
d=7.469d-9
DO 100 x=0,xmax
DO 100 y=0,ymax
        visco(x,y)=visrf*((T(x,y)/Trf)**1.5)*((Trf+110.0)/(T(x,y)+110.0))
        Cp(x,y)=(a+(b*T(x,y))+(c*T(x,y)**2.0)+(d*T(x,y)**3.0))/44.01*1000
        Cv(x,y)=Cp(x,y)/Gram
```

```
      k(x,y)=visco(x,y)*Cp(x,y)/Pr
100   continue

return
end

*****************************************************************

Subroutine ENERGY
implicit none
integer x,y,xmax,ymax,i,j
parameter(xmax=25,ymax=521)
double precision ap(0:xmax,0:ymax),ae,aw,an,as,bb,aa(4,0:xmax),
&                rho0,delx,dely,rho,T,fe,fw,fn,fs,vx,vy,delt,T0,
&                vx0,vy0,k,Cv,Cp,Cp0,Cv0,pre
common/CELLSIZE/delx,dely
common/TIME_STEP/delt
common/CO2_VEl/vx(0:xmax,0:ymax),vy(0:xmax,0:ymax)
common/CO2_DENSITY/rho(0:xmax,0:ymax)
common/CO2_TEMP/T(0:xmax,0:ymax)
common/CO2_DENSITY0/rho0(0:xmax,0:ymax)
common/CO2_TEMP0/T0(0:xmax,0:ymax)
common/CO2_VEl0/vx0(0:xmax,0:ymax),vy0(0:xmax,0:ymax)
common/CO2_HEAT/Cv(0:xmax,0:ymax),Cp(0:xmax,0:ymax)
common/CO2_HEAT0/Cv0(0:xmax,0:ymax),Cp0(0:xmax,0:ymax)
common/CO2_THER/k(0:xmax,0:ymax)
common/PRESS/pre(0:xmax,0:ymax)
do 100 y=1,ymax-1
do 200 x=1,xmax-2
```

```
      fe=((rho(x,y)+rho(x+1,y))*0.5*(Cv(x,y)+Cv(x+1,y))*0.5*vx(x,y)
&                 *0.5*dely)+((k(x,y)+k(x+1,y))*0.25*dely/delx)
      fw=((rho(x,y)+rho(x-1,y))*0.5*(Cv(x,y)+Cv(x-1,y))*0.5
&                 *vx(x-1,y)*0.5*dely)+((k(x,y)+k(x-1,y))*0.25*dely/delx)
      fn=((rho(x,y)+rho(x,y+1))*0.5*(Cv(x,y)+Cv(x,y+1))*0.5*vy(x,y)
&                 *0.5*delx)+((k(x,y)+k(x,y+1))*0.25*delx/dely)
      fs=((rho(x,y)+rho(x,y-1))*0.5*(Cv(x,y)+Cv(x,y-1))*0.5
&                 *vy(x,y-1)*0.5*delx)+((k(x,y)+k(x,y-1))*0.25*delx/dely)
      ae=dmax1(-fe,0.0d+0)
      aw=dmax1(fw,0.0d+0)
      an=dmax1(-fn,0.0d+0)
      as=dmax1(fs,0.0d+0)
      ap(x,y)=(rho(x,y)*Cv(x,y)*delx*dely/delt)+((rho(x,y)
&                 +rho(x+1,y))*0.5*(Cv(x,y)+Cv(x+1,y))*0.5*vx(x,y)*0.5
&                 *dely)-((rho(x,y)+rho(x-1,y))*0.5*(Cv(x,y)+Cv(x-1,y))
&                 *0.5*vx(x-1,y)*0.5*dely)+((k(x,y)+k(x+1,y))*0.25*dely
&                 /delx)-((k(x,y)+k(x-1,y))*0.25*dely/delx)+((rho(x,y)
&                 +rho(x,y+1))*0.5*(Cv(x,y)+Cv(x,y+1))*0.5*vy(x,y)*0.5
&                 *delx)-((rho(x,y)+rho(x,y-1))*0.5*(Cv(x,y)+Cv(x,y-1))
&                 *0.5*vy(x,y-1)*0.5*delx)+((k(x,y)+k(x,y+1))*0.25*delx
&                 /dely)-((k(x,y)+k(x,y-1))*0.25*delx/dely)
      bb=((rho0(x,y)*Cv0(x,y)*T0(x,y))+(0.5*rho0(x,y)
&                 *((vx0(x,y)**2.0)+(vy0(x,y)**2.0))))*delx*dely/delt
      aa(1,x)=-aw
      aa(2,x)=ap(x,y)
      aa(3,x)=-ae
      aa(4,x)=an+as+bb-(rho(x,y)*((vx(x,y)**2.0)+(vy(x,y)**2.0))
&                 *0.5*delx*dely/delt)-((rho(x,y)+rho(x+1,y))*0.5
&                 *(vx(x,y)**3.0)*0.5*dely)+((rho(x,y)+rho(x-1,y))
```

```
&                *0.5*(vx(x-1,y)**3.0)*0.5*dely)-((rho(x,y)+rho(x+1,y))
&                *0.5*(vy(x,y)**2.0)*0.5*vx(x,y)*dely)+((rho(x,y)
&                +rho(x-1,y))*0.5*(vy(x-1,y)**2.0)*0.5*vx(x-1,y)*dely)
&                -((pre(x,y)+pre(x+1,y))*0.5*vx(x,y)*dely)+((pre(x,y)
&                +pre(x-1,y))*0.5*vx(x-1,y)*dely)-((rho(x,y)
&                +rho(x,y+1))*0.5*(vx(x,y)**2.0)*0.5*vy(x,y)*delx)
&                +((rho(x,y)+rho(x,y-1))*0.5*(vx(x,y-1)**2.0)*0.5
&                *vy(x,y-1)*delx)-((rho(x,y)+rho(x,y+1))*0.5*(vy(x,y)
&                **3.0)*0.5*delx)+((rho(x,y)+rho(x,y-1))*0.5*(vy(x,y-1)
&                **3.0)*0.5*delx)-((pre(x,y)+pre(x,y+1))*0.5*vy(x,y)
&                *delx)+((pre(x,y)+pre(x,y-1))*0.5*vy(x,y-1)*delx)
&                -(pre(x,y)+pre(x+1,y))*0.5*vx(x,y)*dely
&                +(pre(x,y)+pre(x-1,y))*0.5*vx(x-1,y)*dely
&                -(pre(x,y)+pre(x,y+1))*0.5*vy(x,y)*delx
&                +(pre(x,y)+pre(x,y-1))*0.5*vy(x,y-1)*delx
200   continue
      aa(1,0)=0.0
      aa(2,0)=1.0
      aa(3,0)=0.0
      aa(4,0)=T(1,y)
      aa(1,xmax-1)=0.0
      aa(2,xmax-1)=1.0
      aa(3,xmax-1)=0.0
      aa(4,xmax-1)=T(xmax-1,y)
      call TDMA(0,xmax-1,aa)
do 300 x=1,xmax-2
            T(x,y)=aa(4,x)
300   continue
100    continue
```

```
return
end

**********************************************************************

Subroutine PRESSURE
implicit none
integer x,y,xmax,ymax
parameter(xmax=25,ymax=521)
double precision V(0:xmax,0:ymax),W,xxx,R,Tc,Pc,Zc,rho,T,
&                pre,M,A,B,xx
common/CO2_DENSITY/rho(0:xmax,0:ymax)
common/CO2_TEMP/T(0:xmax,0:ymax)
common/PRESS/pre(0:xmax,0:ymax)
do 100 y=0,ymax
do 200 x=0,xmax
R=83.144                    ! Gas constant
Tc=304.2                    ! Critical temperature of CO2
Pc=72.9                     ! Critical pressure of CO2
Zc=0.274                    ! Critical compressible factor of CO2
M=44.01
xx=1.0/3.0
W=(0.2905-Zc)/0.085
xxx=(1.0+(0.48508+(1.5517*W)-(0.15613*W**2.0))
&           *(1-(T(x,y)/Tc)**0.5))**2.0
A=0.42748*xxx*(R**2.0)*(Tc**2.0)/Pc
B=0.08664*R*Tc/Pc
V(x,y)=M/rho(x,y)
```

```
pre(x,y)=((R*T(x,y)/(V(x,y)-B))-(A/(V(x,y)*(V(x,y)+B))))*1.0d+5
pre(x,0)=pre(x,1)
200     continue
100     continue
return
end

**********************************************************************

Subroutine ERROR(bbb)
implicit none
integer xmax,ymax,x,y
double precision bbb,prelp,pre,sum
parameter(xmax=25,ymax=521)
common/PRESS/pre(0:xmax,0:ymax)
common/PRESSLOOP/prelp(0:xmax,0:ymax)
sum=0.0d+0
do 100 x=1,xmax-1
do 100 y=1,ymax-1
        sum=sum+(prelp(x,y)-pre(x,y))
100   continue
        bbb=dabs(sum)/(xmax*ymax)
return
end
```

```
Subroutine TDMA(jj,ii,aa)
implicit none
integer ii,jj,k,xmax
parameter (xmax=25)
double precision w,aa(4,0:xmax)
do 100 k=jj,ii
        w=aa(2,k)
        aa(4,k)=aa(4,k)/w
        if (k.lt.ii) then
                aa(3,k)=aa(3,k)/w
                aa(2,k+1)=aa(2,k+1)-aa(1,k+1)*aa(3,k)
                aa(4,k+1)=aa(4,k+1)-aa(1,k+1)*aa(4,k)
        end if
100   continue
do 200 k=ii,jj+1,-1
        aa(4,k-1)=aa(4,k-1)-aa(3,k-1)*aa(4,k)
200   continue
 return
end

**********************************************************

 Subroutine SAVEDATA
 implicit none
 integer xmax,ymax,interval,n,x,y
 parameter(xmax=25,ymax=521)
 double precision rho,T,vx,vy,pre

 common/timest/n,interval
```

```
common/CO2_DENSITY/rho(0:xmax,0:ymax)
common/CO2_TEMP/T(0:xmax,0:ymax)
common/CO2_VEl/vx(0:xmax,0:ymax),vy(0:xmax,0:ymax)
common/PRESS/pre(0:xmax,0:ymax)
do 100 y=1,ymax-1
do 200 x=1,xmax-1
        if (mod(n,2000).eq.0) then
                write(*,*) T(x,y)
        end if
        if (mod(n,10000).eq.0) then
                write(1,*) T(x,y),x,y
                write(2,*) pre(x,y),x,y
                write(3,*) rho(x,y),x,y
                write(4,*) vx(x,y),vy(x,y),x,y
        end if
        if (n.EQ.interval) then
                write(5,*) T(x,y),x,y
                write(6,*) pre(x,y),x,y
                write(7,*) rho(x,y),x,y
                write(8,*) vx(x,y),vy(x,y),x,y
        end if
200   continue
100    continue
do 300 y=1,ymax-1
do 400 x=12,12
        if (mod(n,2000).eq.0) then
                write(13,*) T(x,y),x,y
                write(14,*) pre(x,y),x,y
                write(15,*) rho(x,y),x,y
```

```
        end if
400   continue
300     continue
return
end
```

APPENDIX C

ISENTROPIC FLOW

Isentropic flow

Flow fields from which all molecular transport effects are absent form an important special case to which constant reference is made in analytical fluid dynamics. The total energy equation shows that $\frac{DS}{Dt} = 0$; and the flow is said to be isentropic. The remainder of energy equation yields

$$C_p \frac{DT}{Dt} = \frac{\beta T}{\rho} \frac{Dp}{Dt}$$

Maybe regarded as being combined with the equation of state $f(p, \rho, T) = 0$ to give an equation relating ρ and p for isentropic changes of a material element.

$$\rho = \rho(p, S)$$

in which the appearance of the entropy S is a reminder that when the flow field is not homentropic. ρ is a different function of p for different material elements. The expression for the entropy of a perfect gas

$$S = C_p \log T - R \log p + S_0$$

with $R = C_p - C_v$ is the special form of this equation of state for a gas with constant specific heats. The mass and momentum conservation equations, when supplemented by this relation between ρ and p, are now sufficient to determine the flow field, and the energy equation for isentropic flow serves to determine the associated temperature distribution. The simplifying feature of isentropic flow is that exchanges between the internal energy and other forms of energy are reversible, and internal energy and temperature play passive roles, merely changing in response to the compression of a material element.

The equations governing isentropic flow may thus be written as

$$\frac{1}{\rho c^2} \frac{Dp}{Dt} + \nabla \cdot u = 0$$

$$\rho \frac{Du}{Dt} = \rho F - \nabla p$$

together with the energy equation, where $c^2 = \left(\frac{\partial p}{\partial \rho}\right)_s$ is a function of ρ (or, alternatively, of p) of a form which may be different for different material elements.

These equations may be handled more easily in the important case of a homentropic flow field, for which ρ, and hence also c is a function of p alone.

The physical significance of the parameter c, which has the dimensions of velocity, may be seen in the following way. Suppose that a mass of fluid of uniform density ρ_0 is initially at rest, in equilibrium, so that the pressure p_0 is given by $\rho_0 F = \nabla p_0$

The fluid is then disturbed slightly (all changes being isentropic), by some or all material elements being compressed and their density changed by small amounts, and is subsequently allowed to return freely to equilibrium and to oscillate about it. The fluid is elastic, and no energy is dissipated, so oscillations about the equilibrium are to be expected.

The perturbation quantities $\rho_1 = \rho - \rho_0$ and $p_1 = p - p_0$ and u are all small in magnitude, and a consistent approximation to the equations governing isentropic flow is

$$\frac{1}{\rho_0 c_0^{\,2}} \frac{Dp_1}{Dt} + \nabla \cdot u = 0$$

$$\rho_0 \frac{Du}{Dt} = \rho_1 F - \nabla p_1$$

where c_0 is the value of c at $\rho=\rho_0$. On eliminating u by taking the temporal and spatial gradients of these two equations respectively, then

$$\frac{1}{c_0^2}\frac{\partial^2 p_1}{\partial t^2} = \nabla^2 \cdot p_1 - \rho_1 \nabla \cdot F - \frac{F \cdot \nabla p_1}{c_0^2}$$

If the body force commonly arises from a uniform gravitational field, in which case $F = g$, $\nabla \cdot F = 0$, and the last term is negligible except in the event of the length scale of the pressure variations not being small compared with $\frac{c_0^2}{g}$. Thus in these common circumstances this equation reduces to the wave equation for p_1 and ρ_1 satisfies the same equation. There exist solutions of this equation representing plane compression waves, which propagate with phase velocity c_0 and in which the fluid velocity u is parallel to the direction of propagation. In other words, c_0 is the speed of propagation of sound waves in a fluid whose undisturbed density is ρ_0 . Not all solutions of the equations of isentropic flow represent compression waves of small amplitude, but it is useful nevertheless to keep in mind the interpretation of c as the local value of the speed with which sound waves would propagate through the fluid.

APPENDIX D

THE JOULE – THOMPSON EXPANSION

The Joule-Thompson Expansion

It soon became apparent that the result of the Joule expansion experiment was not valid for real gases. A more accurate experiment, slightly different, was carried out by Joule and J.J. Thompson to further elucidate the properties on real gases under expansion.

A sample of a gas, initially at p_1, V_1, and T_1 was forced through a porous plug at constant pressure, p_1. The gas came out of the other side of the plug at p_2, V_2, and T_2. The apparatus was insulated so that $q = 0$. The work has two terms, the work done on the system to force the gas through the plug and the work done by the system on the surroundings as it came out the other side of the plug.

The total work is

$$w = -p_1(0 - V_1) = p_2(V_2 - 0) = p_1V_1 - p_2V_2$$

Since $q = 0$, the change in internal energy of the gas is,

$$\Delta U = q + w$$

$$= 0 + p_1V_1 - p_2V_2 \neq 0$$

This process, unlike the Joule expansion, is not at constant internal energy.

The enthalpy, however, is given by,

$$\Delta H = \Delta U + \Delta(pV)$$

$$\Delta H = p_1V_1 - p_2V_2 + p_2V_2 - p_1V_1$$

$$\Delta H = 0$$

So the Joule Thompson experiment is a process at constant enthalpy. In the experiment they could select a value for Dp, and then measure DT. The ratio of these two quantities is an approximation to a derivative,

$$\frac{\Delta T}{\Delta p} \approx \left(\frac{\partial T}{\partial p}\right)_H = \mu_{JT} \qquad (1)$$

μ_{JT} is called the "coefficient of the Joule-Thompson effect." This coefficient is not zero for a real gas (or for realistic equations of state like the van der Waals equation of state), but we will now show that it *is* zero for an ideal gas. Applying the Euler chain rule to Equation (1) we obtain,

$$\left(\frac{\partial T}{\partial p}\right)_H = -\frac{\left(\frac{\partial H}{\partial p}\right)_T}{\left(\frac{\partial H}{\partial p}\right)_p} = -\frac{\left(\frac{\partial H}{\partial p}\right)_T}{C_p} \qquad (2)$$

The numerator in Equation (2) is zero for an ideal gas, but not necessarily zero for a real gas.

The coefficient of the Joule-Thompson effect is important in the liquefaction of gases because it tells whether a gas cools or heats on expansion. It turns out that this coefficient is a decreasing function of temperature and it passes through zero at the Joule-Thompson inversion temperature, T_I. In an expansion $dp < 0$. Whether dT is positive or negative depends on the sign of μ_{JT}. Looking at the definition of μ_{JT},

$$\left(\frac{\partial T}{\partial p}\right)_H = \mu_{JT}$$

we see that if μ_{JT} is positive then dT is negative upon expansion so that the gas cools. On the other hand, if μ_{JT} is negative, then dT is positive so that the gas warms upon expansion. In order to liquefy a gas by a Joule-Thompson expansion the gas must first be cooled to below the J-T inversion temperature. Some inversion temperatures are:

He 40 K N_2 621 K O_2 764 K Ne 231 K

We see that N_2 and O_2 will cool upon expansion at room temperature, but He and Ne will warm upon expansion at room temperature.

APPENDIX E

COURANT-FRIEDRICHS-LEWY (CFL) CRITERION

Courant-Friedrichs-Lewy (CFL) criterion

Because this method is an explicit formulation, the time step is subject to a stability criterion. To determine the size of time step, the following version of the Courant-Friedrichs-Lewy (CFL) criterion is used for the convergence of an explicit finite difference scheme. The Courant-Friedrichs-Lewy (CFL) is the domain of dependence of the discrete problem includes the domain of dependence of the differential equation in the limit as the length of the finite difference steps goes to zero. CFL condition is an important stability that must be less than or at most equal to unity.

$A_{I,j}$ = local speed of sound (m/s)

K = the Courant number ($0.5 \le K \le 0.8$), K acts as a "fudge factor"

$$(\Delta t_{CFL})_{i,j} = \left[\frac{|u_{i,j}|}{\Delta x} + \frac{|v_{i,j}|}{\Delta y} + A_{i,j}\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}} + 2v'_{i,j}\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right) \right]^{-1}$$

$$v'_{i,j} = \max\left[\frac{\frac{4}{3}\mu_{i,j}\left(\gamma\mu_{i,j}/\mathrm{Pr}\right)}{\rho_{i,j}} \right]$$

$$\Delta t = \min\left[K(\Delta t_{CFL})_{i,j}\right]$$

$$\mu = \mu_0\left(\frac{T}{T_0}\right)^{\frac{3}{2}}\frac{T_0 + 110}{T + 110}$$

VITA

Ms.Kanokwan Kanno, the third child of Mr. Saneha and Mrs. Somsamai Kanno, was born on November 14, 1977. She attended Sawangsuksa School in Sakonakhon and graduated in 1996. In April 2000, she received her Bachelor Degree of Engineering in Chemical Engineer from Faculty of Engineering, KhonKaen University. In 2000 she gained admission to the Graduate School of Chulalongkorn University and awarded the degree of Master of Engineering in Chemical Engineering in 2004.