LANDMARK UNIVERSITY, OMU-ARAN

LECTURE NOTE

COLLEGE: COLLEGE OF SCIENCE AND ENGINEERING

DEPARTMENT: MECHANICAL ENGINEERING

*Course code: MCE521*

*Course title:* ADVANCED COMPUTATIONAL DYNAMICS.

Credit unit: *2 UNITS.*

Course status: *compulsory*

**ENGR. ALIYU, S.J**

**CFD CLASS WORK EXAMPLES.**

**CFD PROGRAM, ONED**

**1D Conduction Code.**

**1 Structure of the Code**

The 1D conduction code is divided into two parts:

1. a *user part* containing files COM1D.FOR and USER1D.FOR and
2. a *library part* containing file LIB1D.FOR.

The user part is *problem dependent*. Therefore, the two files in this part are used to specify the problem to be solved. In contrast, the library part is *problem independent*. Thus, the LIB1D.FOR file remains unaltered for *all* problems. In this sense, the library part may be called the *solver* whereas the user part may be called the *pre*- and *postprocessor*. This structure is central to creation of a generalised code. To *execute* the code, USER1D.FOR and LIB1D.FOR files are compiled separately and then *linked* before execution. The COM1D.FOR is common to both parts and its contents are brought into each subroutine or function via the "INCLUDE" statement in FORTRAN. Variable names starting with I, J, K, L, M, and N are *integers* whereas all others are *real* by default. The list of variable names with their meanings is given in Table 1. The listings of each file are given at the end of this lecture.

**2 File COM1D.FOR**

In this file, logical, real, and integer variables are included. The PARAMETER statement is used to specify the maximum array dimension IT and values of $\pi$, GREAT, and SMALL. The latter are frequently required for generalised coding. The variable names are given in a *labelled* COMMON as in COMMON/BOUND/. . . , where BOUND is the label. Here, variables of relevance to boundary conditions are included. If required, the user may add more variable names or arrays for the specific problem at hand as shown at the bottom of the file.

Table 1. **List of variables 1D for conduction code.**

| Variable | Meaning |
|---|---|
| ACF | Array containing cross-sectional area ($m^2$) at cell face w |
| AE, AW | Array containing east and west coefficients |
| AL | Domain length (m) |

| | |
|---|---|
| AP | Array containing coefficient of variable $\Phi_p$ |
| COND | Array containing conductivity (W/m-K) at node P |
| CONDREF | Reference conductivity |
| CC | Convergence criterion |
| DELT | Time step (s) |
| DUM1,DUM2 | Dummy arrays |
| FCMX | Maximum absolute fractional change |
| GAUSS | Logical – refers to Gauss–Seidel method |
| GREAT | Parameter having a large value $10^{30}$ |
| H1SPEC | Logical – refers to *h*-boundary condition at node 1 |
| HB1 | Heat transfer coefficient (W/$m^2$-K) at node 1 |
| HB1O | Heat transfer coefficient at node 1 at old time |
| HBN | Heat transfer coefficient at node $N$ |
| HBNO | Heat transfer coefficient at node $N$ at old time |
| HNSPEC | Logical – refers to *h*-boundary condition at node $N$ |
| HPREF | Heat transfer coefficient at any $x$ |
| HPREFO | Heat transfer coefficient at any $x$ at old time |
| ISTOP | STOP index – used in unsteady problems |
| IT | Parameter containing array size |
| ITER | Iteration counter |
| ITERMX | Maximum number of allowable iterations |
| N | Total number of nodes |
| NTIME | Current time counter |
| PERIM | Array containing perimeter (m) at any $x$ |
| PI | Value of $\pi$ |
| PSI | Variable $\psi$ for choosing explicit/implicit scheme |
| Q1SPEC | Logical – refers to *q*-boundary condition at node 1 |
| QB1 | Heat flux (W/$m^2$) at node 1 |
| QB1O | Heat flux at node 1 at old time |
| T | Array containing temperature (°C or K) |
| T1 | Temperature at node 1 |
| T1O | Temperature at node 1 at old time |
| T1SPEC | Logical – refers to $T$ -boundary condition at node 1 |
| THOMAS | Logical – refers to TDMA |
| TIMEMX | Maximum allowable time |
| TINF | Temperature $T_\infty$ |
| TINFO | Temperature $T_\infty$ at old time |
| TINF1 | Temperature $T_\infty$ near node 1 |
| TINFN | Temperature $T_\infty$ near node $N$ |
| TINF1O | Temperature $T_\infty$ near node 1 at old time |
| TINFNO | Temperature $T_\infty$ near node $N$ at old time |
| TN | Temperature at node $N$ |
| TNO | Temperature at node $N$ at old time |
| TNSPEC | Logical – refers to $T$ -boundary condition at node $N$ |

| | |
|---|---|
| TO | Array containing temperature at old time |
| TTIME | Total current time |
| UNSTEADY | Logical ‑ refers to unsteady-state calculation |
| VOL | Array containing cell volume ($m^3$) |
| X | Coordinate of node *P* (m) |
| XCELL | Logical ‑ refers to cell-face coordinate specification |
| XCF | Coordinate of cell face at w |
| XNODE | Logical ‑ refers to node coordinate specification |

### 3 File USER1D.FOR

This is the main control file at the command of the user. The first routine PROGRAM ONED is the command routine from where subroutine MAIN is called. The latter is the first subroutine of the LIB1D.FOR file. When all operations are completed, PROGRAM ONED calls the RESULT subroutine, which is a part of the USER1D.FOR file. Following the listing of the COM1D.FOR file, listings of two USER1D.FOR files are given. They correspond to the two solved problems in the lecture. The reader is advised to refer to these files as well as to Table 1 to understand the description of each routine in USER1D.FOR file.

**BLOCK DATA.** This routine at the end of the USER1D.FOR file specifies all the problem-dependent data such as properties, boundary conditions, and other control parameters. It is assumed that all data are given in consistent units. Here, SI units are used except for the grid data XCF or X, which are dimensionless. The physical coordinates in meters are then evaluated by multiplying by AL (the domain length) in PROGRAM ONED. Dimensionless specification provides better appreciation of non-uniformity (if any) in the specified grid. When a non-uniform grid is specified, it is advisable to ensure that the ratio of two consecutive cell sizes does not exceed 2.

**Subroutine INIT.** In this routine, an initial guess for *T* at ITER = 0 in a steady-state problem or at *t* = 0 in an unsteady-state problem is given. In a steady-state problem, the number of iterations (and hence the computer time) greatly depends on how close the initial guess is to the final converged solution. In the fin problem (Problem 2), a linear temperature profile is given with $T$ 1 = 225 (given) and TN = 205 (which is guessed) although the converged solution is nonlinear.

**Subroutine NEWVAL.** In this routine, boundary conditions at a new time (if different from the initial time) are specified.

**Subroutine PROPS** In this routine, thermal conductivity and specific heat are given. They may be functions of *x*, *t*, or T. The density is of course constant in our formulation.

**Subroutine SORCE.** A problem-dependent source ($q^m \Delta V$) is given in this routine. It may be a function of *T*, *x*, and/or ψ.

**Subroutine INTPRI.** This routine prints the converged solution at the current time step. The routine can also be used to store current values in dummy arrays DUM1 and DUM2 for later printing or plotting. Here, the STOP condition may be given.

**Functions HPERI, AREA, and PERI.** These function routines calculate heat transfer coefficient at node I and area and perimeter at location X or XCF as per the specifications in their arguments. Note that heat transfer coefficients may be functions of $T$, $x$, and/or $t$.

**Subroutine RESULT.** In this last routine, the converged solution is printed along with evaluation and printing of derived parameters. For example, in Problem 2, it is of interest to calculate heat loss from the fin as well as fin effectiveness and compare them with the exact solutions. This routine can also be used to create files containing results for post-processing using graphics packages such as GNUPLOT or GRAPHER.

## 4 File LIB1D.FOR

**Subroutine MAIN.** All subroutines in the code are called from this subroutine. First, subroutines GRID and INIT are called. Then, starting with TTIME = 0, an *outer* DO loop (3000) is initiated to begin calculations at a time step NTIME and TTIME is incremented by DELT. Subroutine NEWVAL is called to set boundary conditions at a new time step. Then, iterations are carried out in an *inner* loop (1000) in which subroutines PROPS, COEF, SORCE, BOUND, and SOLVE are called in turn. The SOLVE routine returns the value of FCMX. If this value is less than $10^{-4}$, the inner loop is exited; otherwise a further iteration is carried out by returning to “1000 ITER = ITER + 1.” In a steady-state problem, a minimum of two iterations are performed. If the problem is steady, there is no need to carry out calculations at a new time step and, therefore, the outer loop is also now exited and control is transferred to statement “5000 CONTINUE.” If the problem is unsteady, subroutines UPDATE and INTPRI are called and the outer loop continues.

**Subroutine GRID.** In this routine, depending on logical XCELL or XNODE, coordinates XCF or X are set and area, perimeter, and cell volume are calculated and printed. It is always desirable to check these specifications in the output file OO (see PROGRAM ONE-D BELOW).

**Subroutine COEF.** In this routine, coefficients AE and AW are evaluated. Note that cell-face conductivities are evaluated by harmonic mean.

**Subroutine BOUND.** This routine implements specified boundary conditions at I = 1 and I = N. The implementation is carried out by updating Su and Sp at near-boundary nodes as explained in the lecture.

**Subroutine SOLVE.** In this routine, Su and Sp are further updated if the problem is unsteady. Also, if the stability criterion is violated, a warning message is printed. AP and Su are further augmented to take account of the underrelaxation factor. Thus, all coefficients are ready to solve the discretised equations. This is done by GS or by TDMA depending on the user choice specified in the BLOCK DATA routine.

**Subroutine UPDATE.** This routine sets all new variables to their “OLD” counterparts.

**Subroutine PRINT** The arguments of this general routine carry the variable F and its logical name “HEADER” specified from point-of-call. The routine is written to print six variables on a line. If N > 6, the next six variables are printed on the next line, and so on. The values are printed in E-format but the user may change to F-format, if desired.

## COMMON BLOCK COM1D.FOR

```
C *** THIS IS COMMON BLOCK FOR 1-D CONDUCTION PROGRAM
      PARAMETER (IT=50, PI=3.1415927, SMALL=1E-30, GREAT=1E30)
      LOGICAL T1SPEC, H1SPEC, Q1SPEC, TNSPEC, HNSPEC, QNSPEC
      LOGICAL STEADY, UNSTEADY, GAUSS, THOMAS, XCELL, XNODE
      COMMON/BOUNDS/T1SPEC, H1SPEC, Q1SPEC, TNSPEC, HNSPEC, QNSPEC
      COMMON/STATE/STEADY, UNSTEADY, GAUSS, THOMAS, XCELL, XNODE
      COMMON/CVAR/T (IT), TO (IT), SPH (IT), COND (IT), RHO (IT)
      COMMON/COORDS/X (IT), XCF(IT), ACF(IT), PERIM(IT),VOL(IT), AL
      COMMON/COEFF/AP(IT), AE(IT), AW(IT), SU(IT), SP(IT), STAB(IT)
      COMMON/CONTRO/ITERMX, N, RP, RSU, FCMX, CC, ISTOP
      COMMON/CTRAN/DELT, TIMEMX, MXSTEP, PSI, ITER, NTIME, TTIME
      COMMON/CPROPS/CONDREF, RHOREF, SPHREF
        COMMON/CDAT1/T1, TN, QB1, QBN, HB1, HBN, TINF1, TINFN, HPREF, TINF
        COMMON/CDAT1/QB1O, QBNO, HB1O, HBNO, TINF1O, TINFNO, HPREFO, TINFO
        COMMON/CDUM/DUM1 (5000), DUM2 (5000), DUM3 (5000)
C  ADDITIONAL PROBLEM-DEPENDENT VARIABLES
C  VARIABLES FOR PROB2
         COMMON/CP2/BREADTH, THICK
C  VARIABLES FOR PROB3
         COMMON/CRADS/R1, R2, R3
```

## USER File for Problem 1

```
C ***************************************************
       PROGRAM ONED
       INCLUDE 'COM1D.FOR'
C ***************************************************
       OPEN (6, FILE='OO')
       WRITE (6,*)' **********************************************'
       WRITE (6,*)' ADHESION OF PLASTIC SHEETS - PROB1'
       WRITE (6,*)' **********************************************'
```

```
      DO 1 I = 1, N
1     XCF (I) = XCF (I) * AL
      CALL MAIN
      CALL RESULT
      STOP
      END
C ************************************************
        SUBROUTINE INIT
        INCLUDE 'COM1D.FOR'
C ************************************************
C  GIVE INITIAL GUESS AT TIME=0.0 OR AT ITER=0 FOR STEADY STATE
       TIN = 30
       DO 1 I = 1, N
       T (I) = 30
       IF (I.EQ.1.OR.I.EQ.N) T (I) = 250
1      CONTINUE
        RETURN
        END
C ************************************************
        SUBROUTINE NEWVAL
        INCLUDE 'COM1D.FOR'
C ************************************************
C  SET NEW VALUES OF HB1, HBN, QB1, QBN, TINF1, TINFN OR SOURCES
         RETURN
         END
C ************************************************
         SUBROUTINE PROPS
         INCLUDE 'COM1D.FOR'
C ************************************************
C  COND (I) AND SPH (I) ARE DEFINED AT NODE P
        DO 1 I = 1, N
        RHO (I) = RHOREF
        COND (I) = CONDREF
1       SPH (I) =SPHREF
```

```
      RETURN
      END
C ************************************************
      SUBROUTINE SORCE
      INCLUDE 'COM1D.FOR'
C ************************************************
C  FORM PROBLEM DEPENDENT SOURCE TERM INCLUDING SU AND SP
      DO 1 I = 2, N-1
      SU (I) = SU (I) + 0.0
1     CONTINUE
      RETURN
      END
C ************************************************
      SUBROUTINE INTPRI
      INCLUDE 'COM1D.FOR'
      CHARACTER*20 HEADER
C ************************************************
      WRITE (6,*)' TIMESTEP = ', NTIME,' TOTAL TIME = ', TTIME
C  PRINT TEMPERATURES AT THE CURRENT STEP
      HEADER=' TEMP '
      CALL PRINT (T, HEADER)
C  STORE MID-POINT TEMPERATURE
      DUM1 (NTIME) = T (4)
C  GIVE STOP CONDITION
      IMID = 4
      IF (T(IMID).GT.140)ISTOP=1
      RETURN
      END
C ************************************************
C     FUNCTION ROUTINES
C ************************************************
      FUNCTION HPERI (II)
      INCLUDE 'COM1D.FOR'
C  H AT PERIMETER
```

```
      I = II
      HPERI=HPREF*0.0+X (I) * 0.0+T (I)*0.0
      RETURN
      END
C --------------------------------------------
        FUNCTION AREA (XX)
        INCLUDE 'COM1D.FOR'
C  AREA OF CROSS-SECTION
        AREA=1.0 + 0.0 * XX
        RETURN
        END
C --------------------------------------------
        FUNCTION PERI (XX)
        INCLUDE 'COM1D.FOR'
C  PERIMETER
       PERI= 0 * XX
       RETURN
       END
C ************************************************
        SUBROUTINE RESULT
        INCLUDE 'COM1D.FOR'
        CHARACTER*20 HEADER
C ************************************************
        HEADER=' FINAL-TEMP '
       CALL PRINT (T, HEADER)
       HEADER=' X (I) '
       CALL PRINT(X, HEADER)
      HEADER=' XCF (I) '
      CALL PRINT (XCF, HEADER)
C EXTRACT PROBLEM DEPENDENT PARAMETERS IF ANY
        WRITE (6,*)' PRINT MID-POINT TEMPERATURE'
        DO 1 I=1, NTIME
        TT= FLOAT (I) * DELT
1       WRITE (6,*) TT, DUM1 (I)
```

```
      TNOW=DUM1 (NTIME)
      TOLD = DUM1 (NTIME-1)
      TT= FLOAT (NTIME-1) * DELT
      TIME = (140-TOLD)/(TNOW-TOLD)*DELT+TT
      WRITE (6,*)' TIME FOR ADHESION = ', TIME
      RETURN
      END
C ************************************************
      BLOCK DATA
      INCLUDE 'COM1D.FOR'
C ************************************************
C  LOGICAL DECLARATIONS
      DATA STEADY,UNSTEADY,GAUSS,THOMAS/.FALSE.,.TRUE.,.TRUE.,.FALSE./
C --------------------------------------------
C  CONTROL PARAMETERS
C  FULLY IMPLICIT (PSI=1), FULLY EXPLICIT (PSI=0), SEMI IMPLICIT (0 < PSI < 1)
      DATA PSI,DELT,MXSTEP,ITERMX,RP,CC/0.0,10,10000,500,1.0,1E-5/
C --------------------------------------------
C  BOUNDARY SPECIFICATION
      DATA T1SPEC,Q1SPEC,H1SPEC/.TRUE.,2*.FALSE./
      DATA TNSPEC,QNSPEC,HNSPEC/.TRUE.,2*.FALSE./
      DATA T1,TN,QB1,QBN,HB1,HBN/250.0,250.0,0.0,0.0,0.0,0.0/
C       DATA TINF,TINF1,TINFN,HPREF/25,150,250,12.0/
      DATA CONDREF,RHOREF,SPHREF/0.25,1300,2000.0/
C --------------------------------------------
C GRID SPECIFICATION
      DATA XCELL,XNODE/.TRUE.,.FALSE./
      DATA N,AL/7,0.01/
      DATA XCF/0.0,0.0,0.2,0.4,0.6,0.8,1.0,43*1.0/
      END
```

## USER File for Problem 1

```
C ************************************************
      PROGRAM ONED
      INCLUDE 'COM1D.FOR'
```

```
C ************************************************
        OPEN (6, FILE='OO')
        WRITE (6,*)' ********************************************'
        WRITE (6,*)' ADHESION OF PLASTIC SHEETS - PROB1-CHAPTER2'
        WRITE (6,*)' ********************************************'
        DO 1 I=1, N
1       XCF (I)=XCF(I)*AL
        CALL MAIN
        CALL RESULT
        STOP
        END
C ************************************************
        SUBROUTINE INIT
        INCLUDE 'COM1D.FOR'
C ************************************************
C GIVE INITIAL GUESS AT TIME=0.0 OR AT ITER=0 FOR STEADY STATE
        TIN=30
        DO 1 I=1, N
        T (I) =30
        IF (I.EQ.1.OR.I.EQ.N)T(I)=250
1       CONTINUE
        RETURN
        END
C ************************************************
        SUBROUTINE NEWVAL
        INCLUDE 'COM1D.FOR'
C ************************************************
C SET NEW VALUES OF HB1, HBN, QB1, QBN, TINF1, TINFN OR SOURCES
        RETURN
        END
C ************************************************
        SUBROUTINE PROPS
        INCLUDE 'COM1D.FOR'
C ************************************************
```

```
C COND (I) AND SPH (I) ARE DEFINED AT NODE P
       DO 1 I=1, N
       RHO (I) = RHOREF
      COND (I) = CONDREF
1     SPH (I) = SPHREF
      RETURN
      END
C ************************************************
        SUBROUTINE SORCE
        INCLUDE 'COM1D.FOR'
C ************************************************
C FORM PROBLEM DEPENDENT SOURCE TERM INCLUDING SU AND SP
        DO 1 I=2, N-1
        SU (I) = SU (I) + 0.0
1       CONTINUE
       RETURN
       END
C ************************************************
       SUBROUTINE INTPRI
       INCLUDE 'COM1D.FOR'
       CHARACTER*20 HEADER
C ************************************************
       WRITE (6,*)' TIMESTEP = ', NTIME,' TOTAL TIME = ', TTIME
C PRINT TEMPERATURES AT THE CURRENT STEP
       HEADER=' TEMP '
       CALL PRINT (T, HEADER)
C STORE MID-POINT TEMPERATURE
       DUM1 (NTIME) =T (4)
C GIVE STOP CONDITION
      IMID=4
      IF (T (IMID).GT.140)ISTOP=1
      RETURN
      END
C ************************************************
```

```
C                 FUNCTION ROUTINES
C ************************************************
                  FUNCTION HPERI (II)
                  INCLUDE 'COM1D.FOR'
C H AT PERIMETER
                 I=II
                 HPERI=HPREF*0.0+X(I)*0.0+T(I)*0.0
                 RETURN
                 END
C -------------------------------------------
                  FUNCTION AREA (XX)
                  INCLUDE 'COM1D.FOR'
C AREA OF CROSS-SECTION
                   AREA=1.0+0.0*XX
                   RETURN
                    END
C -------------------------------------------
                     FUNCTION PERI (XX)
                     INCLUDE 'COM1D.FOR'
C PERIMETER
                    PERI=0*XX
                    RETURN
                    END
C ************************************************
                    SUBROUTINE RESULT
                    INCLUDE 'COM1D.FOR'
                   CHARACTER*20 HEADER
C ************************************************
                   HEADER=' FINAL-TEMP '
                  CALL PRINT (T, HEADER)
                  HEADER=' X (I) '
                 CALL PRINT(X, HEADER)
                 HEADER=' XCF (I) '
                CALL PRINT (XCF, HEADER)
```

```
C EXTRACT PROBLEM DEPENDENT PARAMETERS IF ANY
        WRITE (6,*)' PRINT MID-POINT TEMPERATURE'
        DO 1 I=1, NTIME
        TT= FLOAT (I)*DELT
1       WRITE (6,*) TT, DUM1 (I)
        TNOW=DUM1 (NTIME)
       TOLD=DUM1 (NTIME-1)
       TT=FLOAT (NTIME-1)*DELT
       TIME = (140-TOLD)/(TNOW-TOLD)*DELT+TT
       WRITE (6,*)' TIME FOR ADHESION = ', TIME
       RETURN
       END
C ************************************************
        BLOCK DATA
        INCLUDE 'COM1D.FOR'
C ************************************************
C LOGICAL DECLARATIONS
      DATA STEADY,UNSTEADY,GAUSS,THOMAS/.FALSE.,.TRUE.,.TRUE.,.FALSE./
C -------------------------------------------
C CONTROL PARAMETERS
C FULLY IMPLICIT (PSI=1), FULLY EXPLICIT (PSI=0), SEMI IMPLICIT (0 < PSI < 1)
       DATA PSI,DELT,MXSTEP,ITERMX,RP,CC/0.0,10,10000,500,1.0,1E-5/
C -------------------------------------------
C BOUNDARY SPECIFICATION
       DATA T1SPEC,Q1SPEC,H1SPEC/.TRUE.,2*.FALSE./
       DATA TNSPEC,QNSPEC,HNSPEC/.TRUE.,2*.FALSE./
      DATA T1,TN,QB1,QBN,HB1,HBN/250.0,250.0,0.0,0.0,0.0,0.0/
C      DATA TINF,TINF1,TINFN,HPREF/25,150,250,12.0/
      DATA CONDREF,RHOREF,SPHREF/0.25,1300,2000.0/
C -------------------------------------------
C GRID SPECIFICATION
     DATA XCELL,XNODE/.TRUE.,.FALSE./
     DATA N,AL/7,0.01/
     DATA XCF/0.0,0.0,0.2,0.4,0.6,0.8,1.0,43*1.0/
```

```
      END
```

## USER File for Problem 3

```
C ***************************************************
C THIS IS USER FILE USER1D.FOR - A. W. DATE
C ***************************************************
      PROGRAM ONED
      INCLUDE 'COM1D.FOR'
C ***************************************************
      OPEN (6, FILE='OO')
      WRITE (6,*)' ***********************************************'
      WRITE (6,*)' ANNULAR COMPOSITE FIN - PROB3'
      WRITE (6,*)' SOLVE BY TDMA'
      WRITE (6,*)' ***********************************************'
      DX = (R3-R1)/FLOAT (N-2)
      XCF (1) = 0
      XCF (2) = 0.0
      DO 1 I=3, N
1     XCF (I) = XCF (I-1) + DX
      CALL MAIN
      CALL RESULT
      STOP
      END
C ***************************************************
      SUBROUTINE INIT
      INCLUDE 'COM1D.FOR'
C ***************************************************
C GIVE INITIAL GUESS AT TIME=0.0 OR AT ITER=0 FOR STEADY STATE
      T (1)=T1
      T (N) =TN
      RETURN
      END
C ***************************************************
```

```
      SUBROUTINE NEWVAL
      INCLUDE 'COM1D.FOR'
C ************************************************
C SET NEW VALUES OF HB1, HBN, QB1, QBN, TINF1, TINFN OR SOURCES
      RETURN
      END
C ************************************************
      SUBROUTINE PROPS
      INCLUDE 'COM1D.FOR'
C ************************************************
C COND (I) AND SPH (I) ARE DEFINED AT NODE P
      RR=R2-R1
      DO 1 I=1, N
      IF(X (I).LT. RR) COND (I) = 200
      IF(X (I).GT. RR) COND (I) = 40
1     SPH (I) = SPHREF
      RETURN
      END
C ************************************************
      SUBROUTINE SORCE
      INCLUDE 'COM1D.FOR'
C ************************************************
C FORM PROBLEM DEPENDENT SOURCE TERM INCLUDING SU AND SP
      DO 1 I=2, N-1
      TERM=HPERI (I)*PERIM (I)*(XCF (I+1)-XCF (I))
      SU (I) =SU (I) + TERM*TINF
      SP (I) = SP (I) + TERM
1     CONTINUE
      RETURN
      END
C ************************************************
      SUBROUTINE INTPRI
      INCLUDE 'COM1D.FOR'
C ************************************************
```

```
      RETURN
      END
C ************************************************
C           FUNCTION ROUTINES
C ************************************************
      FUNCTION HPERI (II)
      INCLUDE 'COM1D.FOR'
C H AT PERIMETER
      I=II
      HPERI=HPREF+X (I)*0.0+T(I)*0.0
      RETURN
      END
C -------------------------------------------
      FUNCTION AREA (XX)
      INCLUDE 'COM1D.FOR'
C AREA OF CROSS-SECTION
      AREA=2*PI*(R1+XX)*THICK
      RETURN
      END
C -------------------------------------------
      FUNCTION PERI (XX)
      INCLUDE 'COM1D.FOR'
C PERIMETER
      PERI=4*PI*(R1+XX)
      RETURN
      END
C ************************************************
      SUBROUTINE RESULT
      INCLUDE 'COM1D.FOR'
      CHARACTER*20 HEADER
C ************************************************
      HEADER=' FINAL-TEMP '
      CALL PRINT (T, HEADER)
C EXTRACT PROBLEM DEPENDENT PARAMETERS IF ANY
```

```
      QLOSS=ACF (2)*COND (1)*(T (1)-T (2))/(X (2)-X (1))
      QMAX=2*PI*(R3**2-R1**2)*HPREF*(T (1)-TINF)
      EFF=QLOSS/QMAX
      WRITE (6,*)' NUMERICAL SOLUTION '
      WRITE (6,*)' QLOSS = ', QLOSS,' EFF = ', EFF
C PLOT TEMP PROFILE
      OPEN (12, FILE='TEXT3.DAT')
      WRITE (12,*)'TITLE = ANNULAR FIN'
      WRITE (12,*)'VARIABLES = XX TT '
      WRITE (12,*)'ZONE T = ZONE1, I = ', N,', F = POINT'
      DO 51 J=1, N
51    WRITE (12,*) X (J),T (J)
      CLOSE (12)
      RETURN
      END
C ************************************************
      BLOCK DATA
      INCLUDE 'COM1D.FOR'
C ************************************************
C LOGICAL DECLARATIONS
C ***    DECLARE STEADY OR UNSTEADY AND SOLUTION METHOD
      DATA STEADY,UNSTEADY,GAUSS,THOMAS/.TRUE.,.FALSE.,.FALSE.,.TRUE./
C -------------------------------------------
C CONTROL PARAMETERS
C FULLY IMPLICIT (PSI=1), FULLY EXPLICIT (PSI=0),SEMI IMPLICIT (0<PSI<1)
      DATA PSI, DELT, MXSTEP, ITERMX, RP, CC/1.0,5,100,500,1.0,1E-5/
C -------------------------------------------
C BOUNDARY SPECIFICATION
      DATA T1SPEC,Q1SPEC,H1SPEC/.TRUE.,2*.FALSE./
      DATA TNSPEC,QNSPEC,HNSPEC/.FALSE.,.TRUE.,.FALSE./
      DATA T1,TN,QB1,QBN,HB1,HBN/200.0,150.0,0.0,0.0,0.0,0.0/
      DATA TINF,TINF1,TINFN,HPREF/25,0.0,0.0,20.0/
      DATA CONDREF,RHOREF,SPHREF/1.0,1.0,1.0/
C -------------------------------------------
```

```
C GRID SPECIFICATION
      DATA XCELL,XNODE/.TRUE.,.FALSE./
      DATA N/8/
C PROBLEM DEPENDENT PARAMETERS (IF ANY)
      DATA THICK/0.001/
      DATA R1,R2,R3/0.0125,0.025,0.0375/
      END
```

Library File LIB1D.FOR

```
C ************************************************
C THIS IS LIBRARY LIB1D.FOR - A. W. DATE
C ************************************************
      SUBROUTINE MAIN
      INCLUDE 'COM1D.FOR'
C ************************************************
      WRITE (6,*)' **************************************'
      IF (THOMAS) WRITE (6,*)' SOLUTION BY TDMA'
      IF (GAUSS) WRITE (6,*)' SOLUTION BY GAUSS SIEDEL'
      WRITE (6,*)' **************************************'
C***  CALCULATE CELL FACE COORDINATES, AREA AND VOLUME.
      CALL GRID
C***  SPECIFY INITIAL TEMPERATURE DISTRIBUTION (USER FILE)
      CALL INIT
      ISTOP=0
      IF (STEADY) PSI=1.0
      IF (UNSTEADY) THEN
      DO 101 I=1, N
101   TO (I) =T (I)
      IF (PSI.EQ.0.0) ITERMX=0
      ENDIF
      TTIME=0.0
C***  BEGIN TIME STEP
      TIMEMX=MXSTEP*DELT
      DO 3000 NTIME=1, MXSTEP
      TTIME=TTIME+DELT
```

```
C SET NEW VALUES AT THE BOUNDARY OR SOURCES (USER FILE)
      IF (UNSTEADY) CALL NEWVAL
C***  BEGIN ITERATIONS AT A TIME STEP
      IF (PSI.NE.0.0) WRITE (6,*)' ITER FCMX '
      ITER=0
1000  ITER=ITER+1
C     CALL PROPERTIES ROUTINE (USER FILE)
      CALL PROPS
C***  CALCUALTE THE COEFFICIENTS AW AND AE
      CALL COEF
C***  CALCULATE THE SOURCE TERMS SU AND SP (USER FILE)
      CALL SORCE
C***  SPECIFY THE BOUNDARY CONDITIONS
      CALL BOUND
C***  SOLVE THE DISCRETISED EQUATION
      CALL SOLVE
C***  WRITE RESIDUAL, CHECK CONVERGENCE
      WRITE (6,500) ITER, FCMX
      IF (ITER.GT.ITERMX) GO TO 2000
      IF (STEADY.AND.ITER.EQ.1) GO TO 1000
      IF (FCMX.GT.CC) GO TO 1000
2000  CONTINUE
      IF (STEADY) GO TO 5000
C END OF TIME STEP
C UPDATE OLD TEMPERATURES AND PRINT OUT VARIABLES (USER FILE)
      CALL INTPRI
      CALL UPDATE
      IF (ISTOP.EQ.1) GO TO 5000
      IF (TTIME.GT.TIMEMX) GO TO 5000
3000  CONTINUE
5000  CONTINUE
500   FORMAT (I5, 6X, E10.3)
      RETURN
      END
```

```
C ************************************************
        SUBROUTINE GRID
         INCLUDE 'COM1D.FOR'
        CHARACTER*20 HEADER
C ************************************************
C GRID DATA ARE GIVEN IN BLOCK DATA (USER FILE)
        IF (XCELL) THEN
        XCF (2) = XCF (1)
         X (1) = XCF (1)
         DO 1 I=2, N-1
1         X (I) = 0.5*(XCF (I) + XCF (I+1))
          X (N) = XCF (N)
          ELSE
          XCF (1) = X (1)
          XCF (2) = X (1)
         DO 2 I=3, N-1
2        XCF (I) = 0.5*(X (I) + X (I-1))
          XCF (N) = X (N)
          ENDIF
C CALCULATE PERIMETER, CELL-FACE AREA AND CELL VOLUME
C AREA AND PERI ARE FUNCTION ROUTINES (USER FILE)
       DO 3 I=1, N
      ACF (I) = AREA (XCF (I))
      PERIM (I) = PERI(X (I))
3     CONTINUE
      DO 4 I=2, N-1
4     VOL (I) =AREA(X (I))*(XCF (I+1)-XCF (I))
      HEADER=' X (I) '
      CALL PRINT(X, HEADER)
      HEADER=' XCF (I) '
      CALL PRINT (XCF, HEADER)
     HEADER=' CELL FACE AREA '
     CALL PRINT (ACF, HEADER)
     HEADER=' PERIMETER '
```

```
      CALL PRINT (PERIM, HEADER)
      HEADER=' CELL-VOLUME '
      CALL PRINT (VOL, HEADER)
      RETURN
      END
C ************************************************
      SUBROUTINE COEF
      INCLUDE 'COM1D.FOR'
C ************************************************
      DO 1 I=2, N-1
C INITIALISE SU ANS SP
      STAB (I) = 0.0
      SU (I) = 0.0
      SP (I) = 0.0
      LW=0
      LE=0
      IF (I.EQ.2)LW=1
      IF (I.EQ.N-1) LE=1
      DXE=X (I+1)-X (I)
      DXEP=X (I+1)-XCF (I+1)
      DXEM=XCF (I+1)-X (I)
      DXW=X (I)-X (I-1)
      DXWP=X (I)-XCF (I)
      DXWM=XCF (I)-X (I-1)
C***  CALCULATE CELL FACE CONDUCTIVITY BY HARMONIC MEAN.
      CONDSME=DXE / (DXEM/COND (I) + DXEP/COND (I+1))*(1-LE) + LE*COND (I+1)
      CONDSMW=DXW / (DXWP/COND (I) + DXWM/COND (I-1))*(1-LW) + LW*COND (I-1)
      AW (I) = CONDSMW*ACF (I)/DXW
      AE (I) = CONDSME*ACF (I+1)/DXE
1     CONTINUE
      RETURN
      END
C ************************************************
      SUBROUTINE BOUND
```

```
                        INCLUDE 'COM1D.FOR'
C ************************************************
                       STAB (2) = AW (2)
                      STAB (N-1) =AE (N-1)
C***                  FOR I=1 BOUNDARY
                      IF (T1SPEC) THEN
                      SU (2) = SU (2) + AW (2)*(PSI*T (1) + (1-PSI)*(TO(1)-TO (2)))
                       SP (2) = SP (2) + AW (2)*PSI
                      AW (2) = 0.0
                      ELSE IF (Q1SPEC) THEN
                      SU (2) = SU (2) + ACF (2)*(PSI*QB1 + (1-PSI)*QB1O)
                       T (1) = QB1*ACF (2) / (AW (2) + SMALL) + T (2)
                       AW (2) = 0.0
                       ELSE IF (H1SPEC) THEN
                       TERM1=HB1*ACF (2) + SMALL
                      TERM2=AW (2) + SMALL
                     TERM=1 / (1/TERM1+ 1/TERM2)
                     SU (2) =SU (2) + PSI*TERM*TINF1 + TERM1*(1-PSI)*(TINF1O-TO (1))
                      SP (2) = SP (2) + PSI*TERM
                        T (1) = (T (2) + TERM1/TERM2*TINF1) / (1+TERM1/TERM2)
                        AW (2) = 0.0
                        ENDIF
C***            FOR I=N BOUNDARY
                IF (TNSPEC) THEN
                SU (N-1) = SU (N-1) + AE (N-1)*(PSI*T (N) + (1-PSI)*(TO (N)-TO (N-1)))
                SP (N-1) = SP (N-1) +AE (N-1)*PSI
                AE (N-1) = 0.0
                ELSE IF (QNSPEC) THEN
                 SU (N-1) = SU (N-1) + ACF (N)*(PSI*QBN + (1-PSI)*QBNO)
                 T (N) = QBN*ACF (N) / (AE (N-1) + SMALL) + T (N-1)
                AE (N-1) = 0.0
                ELSE IF (HNSPEC) THEN
                TERM1=HBN*ACF (N) + SMALL
                TERM2 =AE (N-1) + SMALL
```

```
      TERM=1 / (1/TERM1+ 1/TERM2)
      SU (N-1) = SU (N-1) + PSI*TERM*TINFN+TERM1*(1-PSI)*(TINFNO-TO (N))
      SP (N-1) = SP (N-1) + PSI*TERM
       T (N) = (T (N-1) +TERM1/TERM2*TINFN) / (1+TERM1/TERM2)
       AE (N-1) = 0.0
      ENDIF
      RETURN
      END
C ************************************************
        SUBROUTINE SOLVE
        INCLUDE 'COM1D.FOR'
C ************************************************
        DIMENSION AA (IT), BB (IT)
C***    ASSEMBLE SU AND SP TERMS
        DO 1 I=2, N-1
        IF (UNSTEADY) THEN
        BP=RHO (I)*SPH (I)/DELT*VOL (I)
        SP (I)=SP (I)+BP
        SU (I) = SU (I)+(1-PSI)*(AE(I)*TO(I+1)+AW (I)*TO(I-1))
       SU (I) =SU (I) + (BP-(1-PSI)*(AE (I)+AW (I)))*TO (I)
C CHECK FOR STABILITY CONDITION
       TERM = BP-(1-PSI)*(AE (I)+AW (I)+STAB (I))
        IF (TERM.LT.0.0) WRITE(*,*)' COEF OF TPOLD IS NEGATIVE AT I = ',I
        ENDIF
       AP (I)=PSI*(AE (I)+AW (I))+SP (I)
C UNDER-RELAX
      E= (1.-RP)/RP*AP (I)
      AP (I) =AP (I) + E
       SU (I) = SU (I) + E*T (I)
1       CONTINUE
        FCMX=0.0
C -----------------------------------------------
C***  SOLVE BY GAUSS-SIEDEL METHOD
C -----------------------------------------------
```

```
      IF (GAUSS) THEN
      DO 2 I=2, N-1
      TL=T (I)
      ANUM=PSI*(AE (I)*T (I+1) + AW (I)*T (I-1)) + SU (I)
      T (I) =ANUM/AP (I)
      DIFF= (T (I)-TL) / (TL+SMALL)
      IF (ABS (DIFF).GT.FCMX) FCMX=ABS (DIFF)
2     CONTINUE
      ENDIF
C ------------------------------------------------
C***  SOLVE BY TDMA
C ------------------------------------------------
      IF (THOMAS) THEN
C CALCULATE COEFFICIENTS BY RECURRENCE
      AA (2) = PSI*AE (2)/AP (2)
      BB (2) = SU (2)/AP (2)
      DO 3 I=3, N-1
      DEN=1.0-PSI*AW (I)/AP (I)*AA (I-1)
      AA (I) = PSI*AE(I)/AP(I)/(DEN+SMALL)
3     BB (I) = (PSI*AW(I)*BB(I-1)+SU(I))/AP(I)/(DEN+SMALL)
C BACK SUBSTITUTION
      DO 4 I=N-1, 2,-1
      TL=T (I)
      T (I) =AA (I)*T(I+1)+BB(I)
      DIFF= (T (I)-TL)/(TL+SMALL)
      IF (ABS (DIFF).GT.FCMX) FCMX=ABS (DIFF)
4     CONTINUE
      ENDIF
      RETURN
      END
C *************************************************
      SUBROUTINE UPDATE
      INCLUDE 'COM1D.FOR'
C *************************************************
```

```
C RESET OLD VALUES
      DO 200 I=1, N
200   TO (I) =T (I)
      QB1O=QB1
      QBNO=QBN
      HB1O=HB1
      HBNO=HBN
      TINF1O=TINF1
      TINFNO=TINFN
      HPREFO=HPREF
      TINFO=TINF
      RETURN
      END
C ************************************************
      SUBROUTINE PRINT (F, HEADER)
      INCLUDE 'COM1D.FOR'
      CHARACTER*20 HEADER
C ************************************************
      DIMENSION F (IT)
      WRITE (6,*)'*****************************************'
      WRITE (6,*)'DISTRIBUTION OF ', HEADER
      IB=1
      IE=IB+6
      IF (IE.GT.N) IE=N
100   CONTINUE
      WRITE (6,500) (F (I), I = IB, IE)
      WRITE (6,600) (I, I=IB, IE)
      IF (IE.LT.N) THEN
      IB=IE+1
      IE=IB+6
      IF (IE.GT.N) IE=N
      GO TO 100
      ENDIF
      WRITE (6,*)'*****************************************'
```

```
500       FORMAT (7E10.3)
600       FORMAT (4X, I3,6I10)
          RETURN
           END
```

ASSIGNMENT.

From the lecture notes, develop the CFD program for example two (**Rectangular Fin)** on how to determine the heat loss from the fin and its effectiveness.