NAME

PHASE2 − Pivot a canonical-form linear programming tableau to optimality.

SYNOPSIS

CALL PHASE2(T,LDT,BASIS,MP,N,ROWS,MR, RC)

T(LDT,*)	is the REAL*8 input LP tableau
LDT	is the INTEGER*4 leading dimension of T
BASIS(N)	is the INTEGER*4 input/output vector of basic column row indices, 0 for nonbasic
MP	is the INTEGER*4 total number of rows in the LP including the objective
N	is the INTEGER*4 number of variables in the LP
ROWS(MR)	is the INTEGER*4 input vector of the row indices in the problem
MR	is the INTEGER*4 input number of rows in use including the objective
RC	is the INTEGER*4 return code; 0 => optimal; >0 => unbounded in column RC

DESCRIPTION

First the routine finds the most negative cost entry in T. Then it uses MINR to find the minimum-ratio row in that column, or returns with RC set to that column index if no element in the column is greater than 1.D-06. Finally it pivots on the minimum-ratio element, updating T and BASIS. This process is repeated until no negative cost entries remain, when it returns with RC=0.

DIAGNOSTICS

On output these are the possible RC values:

0 all went well
J the index of a column in which the linear program is unbounded

LINKAGE

gfortran source.f −L${HOME}/lib −lmisc

AUTHOR

Michael Kupferschmid

REFERENCE

[1] Kupferschmid, Michael, "Introduction to Mathematical Programming"

EXAMPLE

```
      PARAMETER(LDT=4,MP=4,N=7,MR=4)
      INTEGER*4 ROWS(MR)/1,2,3,4/,BASIS(7),RC
      REAL*8 T(LDT,1+N)/0.0D0, 1.6D2, 5.0D1, 6.0D1,
     ;                 -9.0D1, 7.0D0, 1.0D0, 2.0D0,
     ;                 -1.5D2, 1.0D1, 3.0D0, 4.0D0,
     ;                 -6.0D1, 8.0D0, 1.0D0, 1.0D0,
     ;                 -7.0D1, 1.2D1, 1.0D0, 3.0D0,
     ;                  0.0D0, 1.0D0, 0.0D0, 0.0D0,
     ;                  0.0D0, 0.0D0, 1.0D0, 0.0D0,
     ;                  0.0D0, 0.0D0, 0.0D0, 1.0D0/
      DO 1 I=1,MP
           WRITE(6,901) (T(I,J),J=1,1+N)
  901      FORMAT(8(1X,F7.1))
    1 CONTINUE
      CALL PHASE2(T,LDT,BASIS,MP,N,ROWS,MR, RC)
      WRITE(6,900)
  900 FORMAT(' ')
      DO 2 I=1,MP
           WRITE(6,901) (T(I,J),J=1,1+N)
    2 CONTINUE
      STOP
      END
```

This example (which is from [1] page 72) produced the following output:

```
unix[1] a.out
     0.0   -90.0  -150.0   -60.0   -70.0     0.0     0.0     0.0
   160.0     7.0    10.0     8.0    12.0     1.0     0.0     0.0
    50.0     1.0     3.0     1.0     1.0     0.0     1.0     0.0
    60.0     2.0     4.0     1.0     3.0     0.0     0.0     1.0

  2325.0     0.0     0.0    18.8    76.3     7.5     0.0    18.8
     5.0     1.0     0.0     2.8     2.3     0.5     0.0    -1.3
     7.5     0.0     0.0     1.6    -0.1     0.3     1.0    -1.4
    12.5     0.0     1.0    -1.1    -0.4    -0.3     0.0     0.9
unix[2]
```