

數值方法範例集—— FORTRAN 版本

■ 配合著名教本——

Numerical Recipes:

The Art of Scientific Computing

陳祝美 編譯

數值方法範例集—— FORTRAN 版本

陳祝美 編譯

儒林圖書公司 印行

版權所有
翻印必究

數值方法範例集—FORTRAN版本

編譯者：陳 祝 美

發行人：楊 鏡 秋

出版者：儒林圖書有限公司

地 址：台北市重慶南路一段 111 號

電 話：3118971-3 3144000

郵政劃撥：0106792-1 號

吉豐印刷廠有限公司承印

板橋市三民路二段居仁巷一弄五十三號

行政院新聞局局版台業字第 1492 號

中華民國七十七年七月初版

定價新台幣 180 元正

前言

本「數值方法範例程式集」(Numerical Recipes Example Book)是設計由William H. Press、Brian P. Flannery、Saul A. Tenkolsky 和William T 著作之「數值方法；科學運算之技巧」(Numerical Recipes : The Art of Scientific Computing) (1985, 劍橋大學出版) 一書的輔助教材及參考書。在那本書中，作者經周詳的考慮而發展出各種科學運算的演算法，並由基本的數學分析開始，進而實際地將之設計為FORTRAN之副程式 (subroutine) 和Pascal之程序 (procedure) 的形式。「數值方法；科學運算之技巧」一書中所含之常式 (routine) 幾近 200 個。同時，爲了方便使用者將其併入到自己的應用程式中，它們都被寫成副程式 (或函數 function)，而非獨立的程式 (stand-alone program)。

當使用者想要直接將他人之副程式併入到自己的應用程式前，通常希望先觀察此副程式在一簡單之範例程式中執行的情況。同時，即使是「數值方法」一書中的副程式使用說明，偶而也會出錯。而解決這些問題的最好方法是用一實際的FORTRAN示範程式顯示並說明如何將資料輸入到副程式中、如何呼叫副程式及如何傳回和解釋其結果。

另一較不尋常的情形是，使用者爲了某種目的而修改「外來」(foreign) 之副程式。在此情況下，使用者應會希望在將自己的資料輸入到該修改過之副程式前，先在一已知能正確執行的範例程式上測試其執行結果，以免此副程式在處理你自己的資料時出錯。而另一個相關的情形是，常式原始碼本身並可能遭到破壞。例如：當副程式由一機器傳送到另一機器時，可能會偏失某些列或字元，此時便需要作簡單的測試，以確認其正確性。

對於本數值方法範例程式集在此時特別提出幾點說明。本書的分章與「數值方法；科學運算之技巧」一致。本書所包含的由FORTRAN語言寫成之示範程式可以實際地演練及描述「數值方法」一書中所有的副程式和函數

。同時每個程式都附有註解，其起始處並簡單扼要地說明該程式的功能及所欲演練的「數值方法」常式。本書亦印出示範程式所需的輸入資料。在某些情況下，當示範程式不容易「自行確認」(self-validating)其正確性時，本書亦會印出其範例的輸出結果。

爲了清楚起見，本書以簡單的方式示範「數值方法」一書中的副程式和函數。這使得本書的示範程式有時無法測試所有可能的輸入資料。本書中的示範程式絕非發展「數值方法」一書中的副程式和函數時所需通過的唯一有效之驗證標準。在本書中的程式是使用於著作「數值方法：科學運算之技巧」一書的後期階段以維護「數值方法」一書中程式的完整性，它在此所擔任的角色是非常珍貴的。

目 錄

前言	IV
第一章 緒論	1
第二章 線性代數方程式	7
第三章 內插法和外插法	25
第四章 函數的積分	39
第五章 函數的求值	51
第六章 特殊函數	61
第七章 亂數	89
第八章 排序	111
第九章 求根及方程組	123
第十章 函數的極小化及極大化	133
第十一章 特徵系統	149
第十二章 傅立葉方法	161
第十三章 資料的統計敘述	177
第十四章 資料模式	199
第十五章 常微分方程	215
第十六章 兩點邊界值問題	225
第十七章 偏微分方程	233
中英名詞對照表	240

第一章 緒 論

「數值方法」(Numerical Recipes) 一書第一章中的常式是介紹性的，且其用途與該書其餘部份的常式比起來較不常用。此章的常式主要是用來介紹此書慣用的表示法，而以實例說明控制結構，或者是純供娛樂之用。您甚至會發現它們還很有用。我們希望你能在較輕鬆的場合使用 BADLUK 程式。

Date	Time(EST)	Phase
1 9 1982	3 PM	full moon
1 16 1982	7 PM	last quarter
1 24 1982	11 PM	new moon
2 1 1982	10 AM	first quarter
2 8 1982	2 AM	full moon
2 15 1982	3 PM	last quarter
2 23 1982	4 PM	new moon
3 2 1982	6 PM	first quarter
3 9 1982	3 PM	full moon
3 17 1982	12 AM	last quarter
3 25 1982	5 AM	new moon
4 1 1982	0 AM	first quarter
4 8 1982	5 AM	full moon
4 16 1982	7 AM	last quarter
4 23 1982	4 PM	new moon
4 30 1982	7 AM	first quarter
5 7 1982	8 PM	full moon
5 16 1982	0 AM	last quarter
5 23 1982	0 AM	new moon
5 29 1982	3 PM	first quarter

2 数值方法範例集—FORTRAN 版本

```

PROGRAM DIR1
C   Driver for routine FLMOON
PARAMETER(TZONE=5.0)
CHARACTER PHASE(4)*15,TIMSTR(2)*3
DATA PHASE/'new moon','first quarter',
*         'full moon','last quarter'/
DATA TIMSTR/' AM',' PM'/
WRITE(*,*) 'Date of the next few phases of the moon'
WRITE(*,*) 'Enter today''s date (e.g. 1,31,1982)'
TIMZON=-TZONE/24.0
READ(*,*) IM,ID,IY
C   Approximate number of full moons since January 1900
N=12.37*(IY-1900+(IM-0.5)/12.0)
NPH=2
J1=JULDAY(IM,ID,IY)
CALL FLMOON(N,NPH,J2,FRAC)
N=N+(J1-J2)/28.0
WRITE(*, '(/1X,T6,A,T19,A,T32,A)') 'Date','Time(EST)','Phase'
DO 11 I=1,20
    CALL FLMOON(N,NPH,J2,FRAC)
    IFRAC=NINT(24.*(FRAC+TIMZON))
    IF (IFRAC.LT.0) THEN
        J2=J2-1
        IFRAC=IFRAC+24
    ENDIF
    IF (IFRAC.GE.12) THEN
        J2=J2+1
        IFRAC=IFRAC-12
    ELSE
        IFRAC=IFRAC+12
    ENDIF
    IF (IFRAC.GT.12) THEN
        IFRAC=IFRAC-12
        ISTR=2
    ELSE
        ISTR=1
    ENDIF
    CALL CALDAT(J2,IM,ID,IY)
    WRITE(*, '(1X,2I3,I5,T20,I2,A,5X,A)') IM,ID,IY,
*         IFRAC,TIMSTR(ISTR),PHASE(NPH+1)
    IF (NPH.EQ.3) THEN
        NPH=0
        N=N+1
    ELSE
        NPH=NPH+1
    ENDIF
11  CONTINUE
END

```

副程式 FLMOON 將計算月亮的盈虧現象；更精確地說，即計算某一給定之將發生或已發生的盈虧現象的儒略日期 (Julian day) 和時段。程式 D1R1 詢及目前的日期並收集一連串預定的盈虧現象。我們已比較過這些預測和陰曆表並得到相當滿意的結果。在此列出您已重新檢查過一遍的測試結果。請注意，在此程式中，我們設定 TZONE (time zero ，零時，為 5.0 以表示其與英國格林威治標準零時相差 5 個小時。如果是在格林威治的東邊，則應使用負值的 TZONE 透過 CALDAT 常式將儒略日轉換為曆法日期，本章稍後會介紹該程式。由分時段的儒略日和時區的組合，可修正其所轉換成的曆法日期，其曆法日期可能差上一天。

在 IF 控制結構的實例——JULDAY 程式中，可將曆法日期 (calendar date) 轉變為儒略日期。人們大多不知道他們生日或其它方便之參考日子的儒略日期。為了彌補此一缺失，我們在本章後的 DATES DAT 檔中提供一些檢查

```

PROGRAM D1R2
C   Driver for JULDAY
    CHARACTER TXT*40,NAME(12)*15
    DATA NAME/'January','February','March','April','May','June',
*             'July','August','September','October','November',
*             'December'/
    OPEN(5,FILE='DATES.DAT',STATUS='OLD')
    READ(5,'(A)') TXT
    READ(5,*) N
    WRITE(*,'(/1X,A,T12,A,T17,A,T23,A,T37,A/)') 'Month','Day','Year',
*           'Julian Day','Event'
    DO 11 I=1,N
        READ(5,'(I2,I3,I5,A)') IM,ID,IY,TXT
        WRITE(*,'(1X,A10,I3,I6,3X,I7,5X,A)') NAME(IM),ID,IY,
*           JULDAY(IM,ID,IY),TXT
11   CONTINUE
    CLOSE(5)
    WRITE(*,'(/1X,A/)') 'Month,Day,Year (e.g. 1,13,1905)'
    DO 12 I=1,20
        WRITE(*,*) 'MN,DD,YYYY'
        READ(*,*) IM,ID,IY
        IF(IM.LT.0) STOP
        WRITE(*,'(1X,A12,I8/)') 'Julian Day: ',JULDAY(IM,ID,IY)
12   CONTINUE
END

```

4 數值方法範例集—FORTRAN 版本

點 (checkpoints) 。 D1R2 程式將列出各歷史事件的儒略日期以供比較與選擇。

「數值方法」的下一個程式是 BADLUK，這是一個結合人們最佳及最壞之直覺而寫成之不甚理想的程式。本書並未納入 BADLUK 常式的示範程式，不只是因為我們怕它，也是因為經由文中的範例結果就可以很容易地了解它的執行情形。

第一章最後所介紹的是 CALDAT 常式，它並未介紹新的觀點，但它可將給定的儒略日數字轉變成其開始的年、月、日以補充 JULDAY 常式的不足。如示範程式 D1R4 所提供的：將日期連續輸入到 JULDAY 和 CALDAT 常式以觀察它們是否能保持原狀；當然，這只能測試作者的能力，但我們希望你可分享在此提及這個常式之正確結果的樂趣（我們已使用其它方式更仔細地檢查過了）。

```
PROGRAM D1R4
C   Driver for routine CALDAT
CHARACTER NAME(12)*10
C   Check whether CALDAT properly undoes the operation of JULDAY
DATA NAME/'January','February','March','April','May',
*         'June','July','August','September','October',
*         'November','December'/
OPEN(5,FILE='DATES.DAT',STATUS='OLD')
READ(5,*)
READ(5,*) N
WRITE(*, '(/1X,A,T40,A)') 'Original Date:', 'Reconstructed Date:'
WRITE(*, '(1X,A,T12,A,T17,A,T25,A,T40,A,T50,A,T55,A/)')
*   'Month','Day','Year','Julian Day','Month','Day','Year'
DO 11 I=1,N
  READ(5, '(I2,I3,I5)') IM,ID,IY
  IYCOP Y=IY
  J=JULDAY(IM,ID,IYCOP Y)
  CALL CALDAT(J,IMM,IDD,IYY)
  WRITE(*, '(1X,A,I3,I6,4X,I9,6X,A,I3,I6)') NAME(IM),ID,
*   IY,J,NAME(IMM),IDD,IYY
11 CONTINUE
END
```

附 錄

檔案 DATES·DAT :

List of dates for testing routines in Chapter 1
16 entries

12 31 -1 End of millennium
01 01 1 One day later
10 14 1582 Day before Gregorian calendar
10 15 1582 Gregorian calendar adopted
01 17 1706 Benjamin Franklin born
04 14 1865 Abraham Lincoln shot
04 18 1906 San Francisco earthquake
05 07 1915 Sinking of the Lusitania
07 20 1923 Pancho Villa assassinated
05 23 1934 Bonnie and Clyde eliminated
07 22 1934 John Dillinger shot
04 03 1936 Bruno Hauptman electrocuted
05 06 1937 Hindenburg disaster
07 26 1956 Sinking of the Andrea Doria
06 05 1976 Teton dam collapse
05 23 1968 Julian Day 2440000

第二章 線性代數方程式

「數值方法」第二章藉著討論線性代數方程式的解以開始數值分析的正式介紹。首先介紹高斯 - 喬敦消去法 (Gauss-Jordan elimination) (GAUSS J 常式)，然後介紹 LU 前向和後向代入分解法 (LU decomposition with forward and backward substitution) (LUDCMP 和 LUBKSB 常式)。對於奇異 (singular) 或幾近於奇異之矩陣，最佳的選擇是使用後向代入奇異值分解法 (SVDcmp 和 SVBKSB 常式)。至於其他一些特定形式的線性系統；如三對角線 (tridiagonal)、凡得蒙第 (Vandermonde) 和陶普立茲 (Toeplitz) 矩陣可分別由副程式 TRIDAG、VANDER 和 TOEPLZ 來處理。線性系統中由少數非零係數所構成之稀疏矩陣 (sparse matrices) 則由 SPARSE 常式來處理。

* * * *

GAUSS J 常式利用完全旋轉 (full pivoting) 之高斯 - 喬敦消去法來求得一組線性方程式的解。輸入一群測試矩陣至示範程式 D2R1 以檢查其操作。這些測試矩陣存於本章後的 MATRX1.DAT 檔內。首先由 GAUSS J 常式求得各矩陣的反矩陣 (inverse matrix)。然後將矩陣及其反矩陣相乘並檢查其結果是否為單位矩陣 (unit matrix)。然後再將原先的矩陣乘以解答向量並將結果和產生該解答向量的「右手邊」 (right-hand side) 向量相比較。

LUDCMP 常式之示範程式的使用同於 D2R1 範例程式的測試矩陣。

8 数值方法範例集—FORTRAN 版本

```

PROGRAM D2R1
C   Driver program for subroutine GAUSSJ
C   Reads Matrices from file MATRIX.DAT and feeds them to GAUSSJ
PARAMETER(NP=20)
DIMENSION A(NP,NP),B(NP,NP),AI(NP,NP),X(NP,NP)
DIMENSION U(NP,NP),T(NP,NP)
CHARACTER DUMMY*3
OPEN(5,FILE='MATRIX1.DAT',STATUS='OLD')
10  READ(5,'(A)') DUMMY
    IF (DUMMY.EQ.'END') GOTO 99
    READ(5,*)
    READ(5,*) N,M
    READ(5,*)
    READ(5,*) ((A(K,L), L=1,N), K=1,N)
    READ(5,*)
    READ(5,*) ((B(K,L), K=1,N), L=1,M)
C   Save Matrices for later testing of results
DO 13 L=1,N
    DO 11 K=1,N
        AI(K,L)=A(K,L)
11    CONTINUE

        DO 12 K=1,M
            X(L,K)=B(L,K)
12    CONTINUE
13  CONTINUE
C   Invert Matrix
CALL GAUSSJ(AI,N,NP,X,M,NP)
WRITE(*,*) 'Inverse of Matrix A : '
DO 14 K=1,N
    WRITE(*, '(1H,(6F12.6))') (AI(K,L), L=1,N)
14  CONTINUE
C   Test Results
C   Check Inverse
WRITE(*,*) 'A times A-inverse (compare with unit matrix)'
DO 17 K=1,N
    DO 16 L=1,N
        U(K,L)=0.0
        DO 15 J=1,N
            U(K,L)=U(K,L)+A(K,J)*AI(J,L)
15    CONTINUE
16    CONTINUE
    WRITE(*, '(1H,(6F12.6))') (U(K,L), L=1,N)
17  CONTINUE
C   Check Vector Solutions
WRITE(*,*) 'Check the following vectors for equality:'
WRITE(*, '(T12,A8,T23,A12)') 'Original', 'Matrix*Sol','n'
DO 21 L=1,N
    WRITE(*, '(1X,A,I2,A)') 'Vector ',L,': '
    DO 19 K=1,N

```

```

          T(K,L)=0.0
          DO 18 J=1,N
              T(K,L)=T(K,L)+A(K,J)*X(J,L)
18      CONTINUE
          WRITE(*,'(8X,2F12.6)') B(K,L),T(K,L)
19      CONTINUE
21      CONTINUE
          WRITE(*,*) '*****'
          WRITE(*,*) 'Press RETURN for next problem:'
          READ(*,*)
          GOTO 10
99      CLOSE(5)
          END

```

LUDCMP 常式之績效 (performance) 的檢查方法是：將分解所得之上、下矩陣 (lower and upper matrices) 相乘的結果和原先之矩陣比較，看看它們是否相等。INDX 陣列記錄 LUDCMP 常式的分解步驟，在這我們必須用它來完成還原的工作。但通常是無須如此的，因為 LUDCMP 常式通常和含有還原能力的 LUBKSB 常式一起使用。

```

PROGRAM D2R2
C      Driver for routine LUDCMP
PARAMETER(NP=20)
DIMENSION A(NP,NP),XL(NP,NP),XU(NP,NP)
DIMENSION INDX(NP),JNDX(NP),X(NP,NP)
CHARACTER TXT*3

OPEN (5,FILE='MATRX1.DAT',STATUS='OLD')
READ(5,*)
10  READ(5,*)
    READ(5,*) N,M
    READ(5,*)
    READ(5,*) ((A(K,L), L=1,N), K=1,N)
    READ(5,*)
    READ(5,*) ((X(K,L), K=1,N), L=1,M)
C      Print out A-matrix for comparison with product of lower
C      and upper decomposition matrices.
    WRITE(*,*) 'Original matrix:'
    DO 11 K=1,N
        WRITE(*,'(1X,6F12.6)') (A(K,L), L=1,N)
11  CONTINUE
C      Perform the decomposition
    CALL LUDCMP(A,N,NP,INDX,D)
C      Compose separately the lower and upper matrices
    DO 13 K=1,N
        DO 12 L=1,N

```

10 数值方法范例集—FORTRAN 版本

```

      IF (L.GT.K) THEN
        XU(K,L)=A(K,L)
        XL(K,L)=0.0
      ELSE IF (L.LT.K) THEN
        XU(K,L)=0.0
        XL(K,L)=A(K,L)
      ELSE
        XU(K,L)=A(K,L)
        XL(K,L)=1.0
      ENDIF
12      CONTINUE
13      CONTINUE
C      Compute product of lower and upper matrices for
C      comparison with original matrix.
      DO 16 K=1,N
        JNDX(K)=K
        DO 15 L=1,N
          X(K,L)=0.0
          DO 14 J=1,N
            X(K,L)=X(K,L)+XL(K,J)*XU(J,L)
14          CONTINUE
15          CONTINUE
16          CONTINUE
          WRITE(*,*) 'Product of lower and upper matrices (unscrambled):'
          DO 17 K=1,N
            DUM=JNDX(INDX(K))
            JNDX(INDX(K))=JNDX(K)
            JNDX(K)=DUM
17          CONTINUE
          DO 19 K=1,N
            DO 18 J=1,N
              IF (JNDX(J).EQ.K) THEN
                WRITE(*, '(1X,6F12.6)') (X(J,L), L=1,N)
              ENDIF
18          CONTINUE
19          CONTINUE
          WRITE(*,*) 'Lower matrix of the decomposition:'
          DO 21 K=1,N
            WRITE(*, '(1X,6F12.6)') (XL(K,L), L=1,N)
21          CONTINUE
          WRITE(*,*) 'Upper matrix of the decomposition:'
          DO 22 K=1,N
            WRITE(*, '(1X,6F12.6)') (XU(K,L), L=1,N)
22          CONTINUE
          WRITE(*,*) '*****'
          WRITE(*,*) 'Press RETURN for next problem:'
          READ(*,*)

```

```

READ(5, '(A3)') TXT
IF (TXT.NE. 'END') GOTO 10
CLOSE(5)
END

```

在 LUBKSB 常式之範例驅動程式 (example driver) 將呼叫 LUDCMP 和 LUBKSB 常式以解答存於 MATRX1.DAT 檔中的線性方程式問題 (參考 GAUSS J 常式中的討論)，並將解答向量代入原先的係數矩陣以檢查其結果是否和各問題中之右手邊向量相等。雖然，我們已單獨地測試過 LUDCMP 常式，但在測試 LUBKSB 常式時仍必須呼叫 LUDCMP 常式，這是因為若所處理的問題不是這種型式，則 LUBKSB 常式將毫無意義可言，所以應連合兩個副程式一起測試。

```

PROGRAM D2R3
C Driver for routine LUBKSB
PARAMETER (NP=20)
DIMENSION A(NP,NP), B(NP,NP), INDX(NP)
DIMENSION C(NP,NP), X(NP)
CHARACTER TXT*3
OPEN(5, FILE='MATRX1.DAT', STATUS='OLD')
READ(5, *)
10 READ(5, *)
READ(5, *) N, M
READ(5, *)
READ(5, *) ((A(K,L), L=1,N), K=1,N)
READ(5, *)
READ(5, *) ((B(K,L), K=1,N), L=1,M)
C Save matrix A for later testing
DO 12 L=1, N
DO 11 K=1, N
C(K,L)=A(K,L)
11 CONTINUE
12 CONTINUE
C Do LU decomposition
CALL LUDCMP(C,N,NP,INDX,P)
C Solve equations for each right-hand vector
DO 16 K=1, M
DO 13 L=1, N
X(L)=B(L,K)
13 CONTINUE
CALL LUBKSB(C,N,NP,INDX,X)
C Test results with original matrix
WRITE(*,*) 'Right-hand side vector:'
WRITE(*, '(1X,6F12.6)') (B(L,K), L=1,N)

```