Міністерство науки і освіти України
Національний університет “Львівська політехніка”
Інститут прикладної математики та фундаментальних наук

Кафедра прикладної математики

Звіт

про виконання лабораторної роботи №4
з курсу “Чисельні методи, частина 2”
на тему
“Лінійні багатокрокові методи чисельного розв’язування задачі Коші для системи звичайних диференціальних рівнянь”
Варіант 9

Виконав:
студент групи ПМ-42
Сватюк Д.Р.
Перевірила:
Візнович В.О.

Львів 2017

Мета: навчитися чисельно знаходити розв'язок задачі Коші для системи ЗДР методом Адамса, використовуючи програму STIFF, написану на мові FORTRAN.

Постановка задачі

На відрізку $t \in (0,20]$ потрібно чисельно розв'язати задачу Коші

$$\begin{cases} u_1' = -u_1 + u_2^2 + u_3^2 + u_4^2, \\ u_2' = -10u_2 + 10(u_3^2 + u_4^2), \\ u_3' = -40u_3 + 40u_4^2, \\ u_4' = -100u_4 + 2, \quad t \in (0,20], \end{cases}$$

$$u_1(0) = u_2(0) = u_3(0) = u_4(0) = 1.$$

з точністю $\varepsilon = 10^{-2}, 10^{-4}, 10^{-6}$ та початковим кроком інтегрування $h_0 = 10^{-2}$.

Матриця Якобі правих частин цієї системи ЗДР має вигляд

$$\begin{pmatrix} -1 & 2u_2 & 2u_3 & 2u_4 \\ 0 & -10 & 20u_3 & 20u_4 \\ 0 & 0 & -40 & 80u_4 \\ 0 & 0 & 0 & -100 \end{pmatrix}$$

Написати та відлагодити програму STIFF для розв'язування заданої задачі Коші.

Методи Адамса

Введемо на інтервалі $[t_0, T]$ рівномірну сітку $\overline{\omega}_\tau = \{t_n = t_0 + n\tau, n = \overline{0, n_0}\}$ з кроком $\tau = (T - t_0)/n_0$. Якщо рівняння (1) §2 проінтегрувати на відрізку $[t_n, t_{n+1}]$, то одержимо

$$u_{n+1} = u_n + \int_{t_n}^{t_{n+1}} f(t, u(t))dt. \qquad (2)$$

Тоді чисельний аналог буде задаватися формулою

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} P(t)dt.$$

Після заміни змінної $s = (t - t_n)/\tau$ в останньому інтегралі та підставляння виразу (2) будемо мати

$$y_{n+1} = y_n + \tau \int_0^1 P(t_n + s\tau)ds = y_n + \tau \sum_{j=0}^{k-1} \gamma_j \nabla^j f_n, \qquad (3)$$

де коефіцієнти γ_j обчислюються за формулами

$$\gamma_0 = 1, \ \gamma_j = \frac{1}{j!} \int_0^1 s(s+1)\dots(s+j-1)ds, \ j = \overline{1, k-1}.$$

Формули (3) одержані при інтегруванні інтерполяційного многочлена від t_n до t_{n+1}, тобто зовні інтервалу інтерполяції (t_{n-k+1}, t_n). Добре відомо, що зовні цього інтервалу інтерполяційний многочлен дає досить погане наближення. Тому дослідимо також методи, що грунтуються на інтерполяційному многочлені, який додатково використовує точку (t_{n+1}, f_{n+1}), тобто

$$P^*(t) = P^*(t_n + s\tau) = \nabla^0 f_{n+1} + \frac{s-1}{1!}\nabla f_{n+1} + \frac{(s-1)s}{2!}\nabla^2 f_{n+1} + \\ + \ldots + \frac{(s-1)s\ldots(s+k-2)}{k!}\nabla^k f_{n+1}. \tag{4}$$

Підставляючи цей многочлен у (1), одержимо наступний неявний метод:

$$y_{n+1} = y_n + \tau \sum_{j=0}^{k} \gamma_j^* \nabla^j f_{n+1}, \tag{5}$$

де коефіцієнти γ_j^* визначаються за формулами

$$\gamma_0^* = 1, \quad \gamma_j^* = \frac{1}{j!}\int_0^1 (s-1)s\ldots(s+j-2)ds, \quad j = \overline{1,k}.$$

Формули (5) визначають y_{n+1} неявно (на кожному кроці для обчислення y_{n+1} необхідно розв’язати нелінійне рівняння), а тому вони називаються неявними методами Адамса.

Неявні формули Адамса мають загальний вигляд:

$$y_{n+1} = y_n + \tau \sum_{j=0}^{k} \beta_j f_{n-j+1}.$$

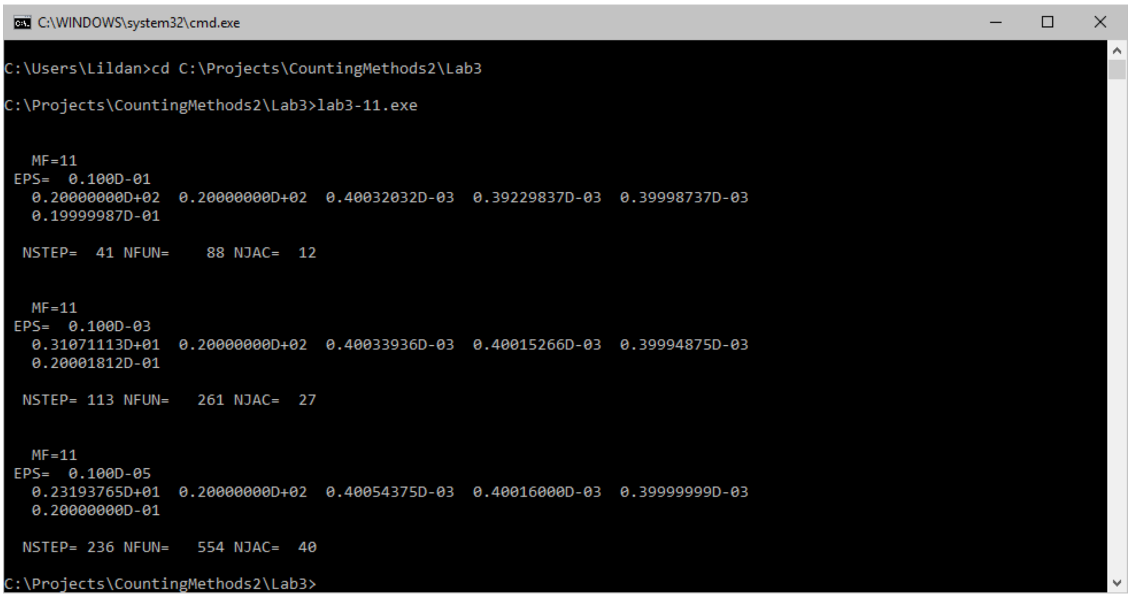
Програмний код

```
 PROGRAM LAB4
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(10,13),YMAX(10),ERROR(10),PW(100),
     1          FSAVE(20),IWORK(10)
      COMMON/STCOM1/T,H,HMIN,HMAX,EPS,N,MF,KFLAG,JSTART,MAXORD
      COMMON/STCOM2/HUSED,NQUSED
      COMMON/STCOM3/ML,MU
      COMMON/STCOM4/NSTEP,NFUN,NJAC
      NYDIM=10
      EPS=1.D-2
      KB=0
401   CONTINUE
      N=4
      T=0.0D0
      TEND=20.D0
      Y(1,1)=1.D0
      Y(2,1)=1.D0
      Y(3,1)=1.D0
      Y(4,1)=1.D0
      H=1.D-1
      HMAX=TEND
      HMIN=1.D-15
      JSTART=0
      MF=12
      MAXORD=5
      WRITE(0,20) MF,EPS
20    FORMAT(//3X,'MF=',I2/,' EPS='D11.3)
      NSTEP=0
      NFUN=0
      NJAC=0
      DO 30 I=1,N
30    YMAX(I)=1.D0
40    CONTINUE
      CALL STIFF(Y,YMAX,ERROR,PW,FSAVE,IWORK,NYDIM)
      IF(KFLAG.EQ.0)GO TO 60
      WRITE(0,50) KFLAG
50    FORMAT(/'  KFLAG=',I2/)
      STOP
60    CONTINUE
      IF(DABS(TEND-T).LE.1.D-15) GO TO 90
      IF(TEND-T-H) 80,40,40
80    E=TEND-T
      S=E/H
      DO 85 I=1,N
      DO 85 J=1,JSTART
85    Y(I,1)=Y(I,1)+Y(I,J+1)*S**J
      T=T+E
      GO TO 60
90    CONTINUE
      WRITE(0,556) H,T,(Y(I,1),I=1,N)
556   FORMAT(1X,5D16.8)
```

```
      WRITE(0,95) NSTEP,NFUN,NJAC
95    FORMAT(/'  NSTEP=',I4,' NFUN= ',I5,' NJAC=',I4)
      KB=KB+1
      IF(KB.GE.3) GO TO 402
      EPS=EPS*1.D-2
      GO TO 401
402   CONTINUE
      STOP
      END

SUBROUTINE DIFFUN (N,T,Y,YDOT)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(1),YDOT(1)
      YDOT(1)=-Y(1)+Y(2)*Y(2)+Y(3)*Y(3)+Y(4)*Y(4)
      YDOT(2)=-10.D0*Y(2)+10.D0*(Y(3)*Y(3)+Y(4)*Y(4))
      YDOT(3)=-40.D0*Y(3)+40.D0*Y(4)*Y(4)
      YDOT(4)=-100.0D0*Y(4)+2
      RETURN
      END

SUBROUTINE PEDERV(N,T,Y,PW,NYDIM)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(1),PW(1)
      PW(1)=-1.0D0
      PW(2)=0.D0
      PW(3)=0.D0
      PW(4)=0.D0
      PW(NYDIM+1)=-2.0D0*Y(2)
      PW(NYDIM+2)=-1.0D1
      PW(NYDIM+3)=0.D0
      PW(NYDIM+4)=0.D0
      N2=NYDIM*2
      PW(N2+1)=2.0D0*Y(3)
      PW(N2+2)=2.0D1*Y(3)
      PW(N2+3)=-40.D0
      PW(N2+4)=0.D0
      N2=NYDIM*3
      PW(N2+1)=2.0D0*Y(4)
      PW(N2+2)=2.0D1*Y(4)
      PW(N2+3)=8.0D1*Y(4)
      PW(N2+4)=-1.0D2
      RETURN
      END
```

Результати виконання

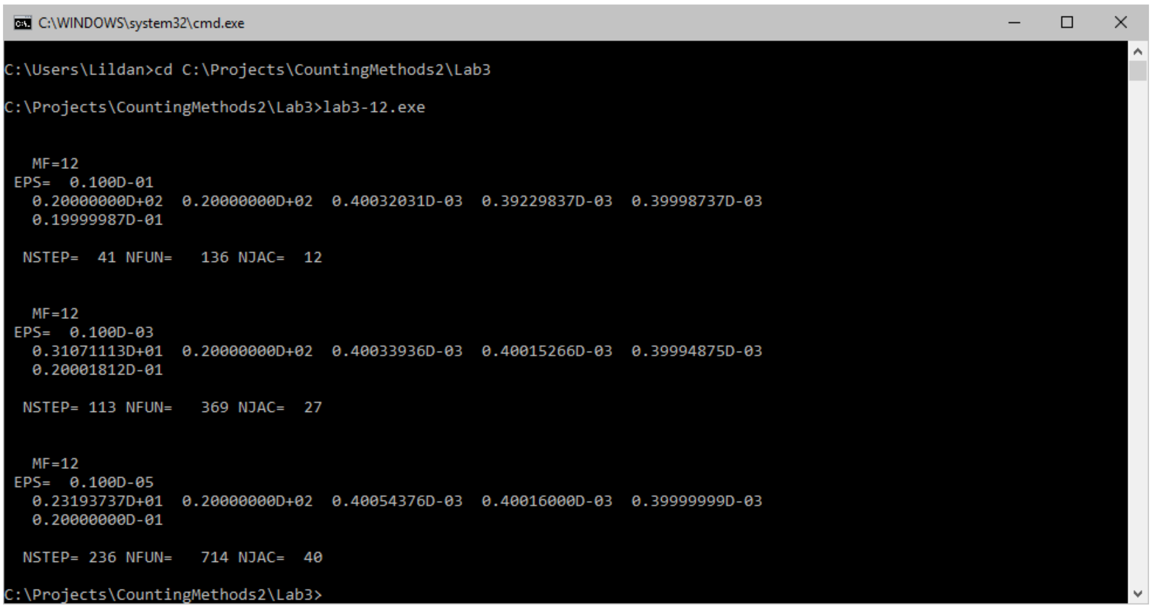
1) Використовується модифікований ітераційний метод Ньютона, матриця Якобі обчислюється аналітично підпрограмою PEDERV.



```
C:\WINDOWS\system32\cmd.exe

C:\Users\Lildan>cd C:\Projects\CountingMethods2\Lab3

C:\Projects\CountingMethods2\Lab3>lab3-11.exe


  MF=11
 EPS=  0.100D-01
  0.20000000D+02  0.20000000D+02  0.40032032D-03  0.39229837D-03  0.39998737D-03
  0.19999987D-01

 NSTEP=  41 NFUN=    88 NJAC=  12


  MF=11
 EPS=  0.100D-03
  0.31071113D+01  0.20000000D+02  0.40033936D-03  0.40015266D-03  0.39994875D-03
  0.20001812D-01

 NSTEP= 113 NFUN=   261 NJAC=  27


  MF=11
 EPS=  0.100D-05
  0.23193765D+01  0.20000000D+02  0.40054375D-03  0.40016000D-03  0.39999999D-03
  0.20000000D-01

 NSTEP= 236 NFUN=   554 NJAC=  40

C:\Projects\CountingMethods2\Lab3>
```

Рис: 1

2) Використовується модифікований ітераційний метод Ньютона, матриця Якобі обчислюється чисельним диференціюванням.



```
C:\WINDOWS\system32\cmd.exe

C:\Users\Lildan>cd C:\Projects\CountingMethods2\Lab3

C:\Projects\CountingMethods2\Lab3>lab3-12.exe


  MF=12
 EPS=  0.100D-01
  0.20000000D+02  0.20000000D+02  0.40032031D-03  0.39229837D-03  0.39998737D-03
  0.19999987D-01

 NSTEP=  41 NFUN=   136 NJAC=  12


  MF=12
 EPS=  0.100D-03
  0.31071113D+01  0.20000000D+02  0.40033936D-03  0.40015266D-03  0.39994875D-03
  0.20001812D-01

 NSTEP= 113 NFUN=   369 NJAC=  27


  MF=12
 EPS=  0.100D-05
  0.23193737D+01  0.20000000D+02  0.40054376D-03  0.40016000D-03  0.39999999D-03
  0.20000000D-01

 NSTEP= 236 NFUN=   714 NJAC=  40

C:\Projects\CountingMethods2\Lab3>
```

Рис: 2

Висновок

Під час виконання лабораторної роботи, я навчився розв’язувати задачу Коші для системи ЗДР методом Адамса використовуючи програму STIFF.