

Venlig hilsen Ole Trinhammer

7/5-82

RUN OF BACKFIT WITH N= 3
NUMBER OF STEPS PR OSCILLATION 75
AND STEPWISE FLUCTUATION BELOW 0.001

LEVEL NUMBER	ENERGY E(I)	BROKEN(I)
1	0.499691925667126	0.499691925667126
2	1.50279860419355	1.49643122781618
3	2.47045362864153	2.47045362864153

✓

relative breakdown= 1.42556071191857E-3

Gem udskriften til mig

Venlig hilsen Ole Trinhammer

7/5-92

RUN OF BACKFIT WITH N= 3
NUMBER OF STEPS PR OSCILLATION 300
AND STEPWISE FLUCTUATION BELOW 1.0E-5

LEVEL NUMBER	ENERGY E(I)	BROKEN(I)
1	0.499804249284485	0.499804249284485
2	1.50299068291251	1.49643528450246
3	2.47137228333872	2.47137228333872

5-10 min

relative breakdown= 1.46731602439656E-3

Gem udskriften til mig

Venlig hilsen Ole Trinhammer

7/5-92

RUN OF BACKFIT WITH N= 3
NUMBER OF STEPS PR OSCILLATION 900
AND STEPWISE FLUCTUATION BELOW 1.0E-6

LEVEL NUMBER	ENERGY E(I)	BROKEN(I)
1	0.499804697384023	0.499804697384023
2	1.50298913256649	1.49643422557105
3	2.47137899239047	2.47137899239047

15-25 min

relative breakdown= 1.46720402676481E-3

```

0010 //
0020 // BACKFIT
0030 //
0040 PRINT CHR$(12)
0050 INPUT "N= ": n
0055 INPUT "NUMBER OF LEVELS TO CALCULATE= ": m
0060 INPUT "ITERATION STOPS WHEN FLUCTUATION IS BELOW EPSILON= ": epsilon
0070 INPUT "INTEGRATIONSTEP PR OSCILLATION= ": skridt
0080 PRINT "N= ";n;"epsilon= ";epsilon;"SKRIDT/OSC (N*SK)= ";n*skridt
0084 PRINT
0085 PRINT "MASKINEN REGNER KÆRE KOLLEGA"
0086 PRINT "Færdig ved 'program afsluttet' - gem venligst udskrift til OT"
0087 PRINT
0088 PRINT
0090
0100 DIM e(m)
0110 DIM broken(m)
0120
0130
0140
0150 // BARE LINES CARRY FILING STRUCTURE FOR INTERRUPTED CALCULATIONS
0160
0170
0180 l:=n //  $l=g^2*N$  is a lattice reminiscence
0190 lc:=PI^2/(2*SQR(2))
0200 //
0209 // EIGENVALUE GUESS FROM A SUITABLY WEITHTED AVERAGE OF HARMONIC OSCILLATOR
0210 // AND STURM-LOUVILLE THEORY FOR ZERO POTENTIAL
0220 //
0250 deltae:=2*SQR(2)*lc*lc/(lc*lc+l*l)+4*4*PI*PI*l*l/(lc*lc+l*l)
0260 //          EHARMON          VÆGT          ESTURM          VÆGT
0320 FOR i:=1 TO m DO
0330
0340
0350
0360 //
0370 // EIGENVALUE LIMITED ABOVE FROM STURM-THEORY FOR NEXT FOLLOWING LEVEL
0380 //
0390 eimax:=2*(i+1)*(i+1)*4*PI*PI+PI*PI/2
0400 //
0410 //          ESTURM(NEXT)          + POTMAX
0420 //
0430 IF i>1 THEN
0440     eimin:=e(i-1)
0450 ELSE
0460     eimin:=0
0470 ENDIF
0480 ei:=(eimax+eimin)/2
0500 //
0510 //
0520 //
0530 //
0540 //
0550 //
0590 // Iteration until correct number of nodes
0600 //
0601 zeros:=0
0620 IF i MOD 2=1 THEN
0622     node(ei,eimax,eimin,zeros)
0625 REPEAT
0630     IF 2*zeros>i-1 THEN
0640         eimax:=ei
0650         ei:=(eimax+eimin)/2
0670     ENDIF
0680     IF 2*zeros<i-1 THEN
0690         eimin:=ei
0700         ei:=(eimax+eimin)/2
0720     ENDIF
0722     node(ei,eimax,eimin,zeros)
0725 UNTIL 2*zeros=i-1

```

```

0730      //
0740      // Finetuning via backward integration
0750      //
0770      backfit(ei,eimax,eimin)
0790      e(i):=ei
0800  ENDIF
0802  //
0804  // Iteration of even states
0806  //
0810  IF i MOD 2=0 THEN
0815      node(ei,eimax,eimin,zeros)
0820      REPEAT
0840          IF 2*zeros>i-2 THEN
0850              eimax:=ei
0860              ei:=(eimax+eimin)/2
0870          ENDIF
0880          IF 2*zeros<i-2 THEN
0890              eimin:=ei
0900              ei:=(eimax+eimin)
0910          ENDIF
0912          node(ei,eimax,eimin,zeros)
0915      UNTIL 2*zeros=i-2
0920      //
0922      // Finetuning of even states
0924      //
0930      backfit(ei,eimax,eimin)
0950      e(i):=ei
0960  ENDIF
0970
1030
1040  ENDFOR i
1050
1060  //
1070  //
1080  // THIS PART OF THE PROGRAMME FINDS EIGENVALUES FOR BROKEN LEVELS
1090  //
1100  //
1110  FOR i:=1 TO m DO
1120      IF i MOD 2=1 THEN
1130          broken(i):=e(i)
1140      ELSE
1150          bimax:=e(i)
1160          bimin:=e(i-1)
1170          bi:=(bimax+bimin)/2
1180          zeros:=0
1200
1210
1220
1240
1250
1260
1270      node(bi,bimax,bimin,zeros)
1275      REPEAT
1280          IF 2*zeros>i-2 THEN
1290              bimax:=bi
1300              bi:=(bimax+bimin)/2
1310          ENDIF
1320          IF 2*zeros<i-2 THEN
1330              bimin:=bi
1340              bi:=(bimax+bimin)/2
1350          ENDIF
1450
1460
1470          node(bi,bimax,bimin,zeros)
1480      UNTIL 2*zeros=i-2
1485      backbrok(bi,bimax,bimin)
1490      broken(i):=bi
1500
1510
1520

```

```

1530   ENDIF
1540   ENDFOR i
1545   SELECT OUTPUT "prn"
1550   PRINT
1555   PRINT "Gem udskriften til mig"
1556   PRINT
1557   PRINT "Venlig hilsen      Ole Trinhammer"
1558   PRINT
1560   PRINT
1570   PRINT "RUN OF BACKFIT WITH N= ",n
1571   PRINT "NUMBER OF STEPS PR OSCILLATION ",n*skridt
1575   PRINT " AND STEPWISE FLUCTUATION BELOW ",epsilon
1580   PRINT
1590   PRINT "LEVEL NUMBER      ENERGY E(I)                BROKEN(I)"
1600   PRINT
1610
1620   FOR i:=1 TO m DO
1630
1640
1650     PRINT TAB(5);i;TAB(17),e(i);TAB(40),broken(i)
1660   ENDFOR i
1670
1672   IF m=3 THEN
1673     break:=(e(2)-broken(2))/(broken(1)+broken(2)+broken(3))
1674     PRINT
1675     PRINT "relative breakdown= ",break
1676   ENDIF
1678   SELECT OUTPUT "con"
1680
1690
1700   //
1710   //
1720   // RUNGE KUTTA 4.ORDEN
1730   //
1740   PROC node(REF ei,REF eimax,REF eimin,REF zeros)
1750     x:=0
1760     IF i MOD 2=1 THEN
1770       y:=1
1780       v:=0
1790     ELSE
1800       y:=0
1810       v:=4*i*skridt
1820     ENDIF
1830     zeros:=0
1850     max:=n*i*skridt
1860     xturn:=SQR(2*ei)
1870     h:=xturn/max
1880     FOR j:=1 TO max+1 DO
1890       k1:=v
1900       l1:=y*(x*x-ei*2)
1910       k2:=v+h*l1/2
1920       l2:=(y+h*k1/2)*((x+h/2)*(x+h/2)-ei*2)
1930       k3:=v+h*l2/2
1940       l3:=(y+h*k2/2)*((x+h/2)*(x+h/2)-ei*2)
1950       k4:=v+h*l3
1960       l4:=(y+h*k3)*((x+h)*(x+h)-ei*2)
1970       yny:=y+h*(k1+2*k2+2*k3+k4)/6
1980       vny:=v+h*(l1+2*l2+2*l3+l4)/6
1985       IF j<=max THEN
1990         IF y*yny<0 THEN
2000           zeros:=zeros+1
2010         ENDIF
2020         IF y=0 AND x<>0 THEN
2030           zeros:=zeros+1
2040         ENDIF
2160       //
2170       // Interrupted return in case of overestimated eigenvalue
2180       //

2190     IF i MOD 2=1 AND 2*zeros>i-1 OR i MOD 2=0 AND 2*zeros>i-2 THEN
2200       RETURN

```

```

2210         ENDIF
2260     ENDIF
2270     x:=x+h
2320     y:=yny
2330     v:=vny
2340     ENDFOR j
2360 ENDPROC node
3000 PROC backfit(REF ei,REF eimax,REF eimin)
3010     max:=n*i*skridt
3015 again:
3020     x:=PI
3030     IF i MOD 2=1 THEN
3040         y:=1
3045         IF i=3 THEN y:=-y
3050         v:=0
3060     ELSE
3070         y:=0
3080         v:=1
3090     ENDIF
3100     xturn:=SQR(2*ei)
3110     h:=-PI/max
3120     FOR j:=1 TO max DO
3130         k1:=v
3140         l1:=y*(x*x-ei*2)
3150         k2:=v+h*l1/2
3160         l2:=(y+h*k1/2)*((x+h/2)*(x+h/2)-ei*2)
3170         k3:=v+h*l2/2
3180         l3:=(y+h*k2/2)*((x+h/2)*(x+h/2)-ei*2)
3200         k4:=v+h*l3
3210         l4:=(y+h*k3)*((x+h)*(x+h)-ei*2)
3220         yny:=y+h*(k1+2*k2+2*k3+k4)/6
3230         vny:=v+h*(l1+2*l2+2*l3+l4)/6
3240         //
3250         // Fitting boundary conditions at 0
3260         //
3270         IF j=max THEN
3274             WHILE eimax-eimin>epsilon*ei DO
3275
3280                 IF i MOD 2=1 THEN
3290                     IF vny>0 THEN
3300                         eimax:=ei
3310                         ei:=(eimax+eimin)/2
3320                     ELSE
3330                         eimin:=ei
3340                         ei:=(eimax+eimin)/2
3350                     ENDIF
3355                 ENDIF
3360                 IF i MOD 2=0 THEN
3370                     IF yny>0 THEN
3380                         eimax:=ei
3390                         ei:=(eimax+eimin)/2
3400                     ELSE
3410                         eimin:=ei
3420                         ei:=(eimax+eimin)/2
3430                     ENDIF
3440                 ENDIF
3443                 GOTO again
3445             ENDWHILE
3450         ENDIF
3452         x:=x+h
3453         y:=yny
3454         v:=vny
3455     ENDFOR j
3460 ENDPROC backfit
4000 PROC backbrok(REF bi,REF bimax,REF bimin)
4010     max:=n*i*skridt
4015 again:
4020     x:=PI

```

```

4030  y:=1
4040  v:=0

```

```

4110  h:=-PI/max
4120  FOR j:=1 TO max DO
4130      k1:=v
4140      l1:=y*(x*x-bi*2)
4150      k2:=v+h*l1/2
4160      l2:=(y+h*k1/2)*((x+h/2)*(x+h/2)-bi*2)
4170      k3:=v+h*l2/2
4180      l3:=(y+h*k2/2)*((x+h/2)*(x+h/2)-bi*2)
4200      k4:=v+h*l3
4210      l4:=(y+h*k3)*((x+h)*(x+h)-bi*2)
4220      yny:=y+h*(k1+2*k2+2*k3+k4)/6
4230      vny:=v+h*(l1+2*l2+2*l3+l4)/6
4240      //
4250      // Fitting boundary conditions at 0
4260      //
4261      WHILE bimax-bimin>epsilon*bi DO < DP j 2 max THEN
4262          IF yny<0 THEN
4263              bimax:=bi
4264              bi:=(bimax+bimin)/2
4265          ELSE
4266              bimin:=bi
4267              bi:=(bimax+bimin)/2
4270          ENDIF
4370          GOTO again
4380
4390
4400
4410
4420
4430
4435
4440
4445      ENDWHILE < DP IF
4452      x:=x+h
4453      y:=yny
4454      v:=vny
4455  ENDFOR j
4460 ENDPROC backbrok
4470 END

```