15. Programele de calcul şi ... câteva sfaturi utile

Un set de programe scrise în FORTRAN pentru compilatorul Compaq's Digital Fortran 6 şi pentru pachetul de programe *Mathematica* sunt prezentate mai jos. Sunt descrise codurile sursă (*source code*), astfel încât informaţia prezentată referitor la modelele analitice este completată în continuare cu tehnicile de calcul corespunzătoare.

Se va ţine cont de următoarele aspecte:

1. Dacă utilizaţi un compilator FORTRAN diferit de Compaq's Digital Fortran 6, atunci, din cauza incompatibilităţii bibliotecilor de programe, ar putea fi necesar de a modifica codul în concordanţă cu biblioteca proprie. Sau instalaţi Compaq's Digital Fortran!

2. Atunci când se importă/exportă date sau se crează fişiere output noi, modificaţi adresa din program în corespundere cu parametrii proprii. Adresa din program este adresa proprie a autorului!

3. Dacă folosiţi executabile “*.exe” obţinute cu ajutorul compilatorului Fortran, plasaţi-le în aceiaşi mapă cu fişierile de date iniţiale atunci când programul conţine o asemenea opţiune.

4. Iar dacă aţi ţinut cont de toate sfaturile de mai sus, dar programul, totuşi, nu funcţionează, atunci ... mai treceţi o dată cu privirea peste cod şi comparaţi-l cu cel din lucrarea dată. Buna funcţionare a programelor a fost testată de autor.

(** **DYNAMIC CLUSTER MODEL. MASTER EQUATIONS** **)

(** PROBABILITIES: **)
(* p1 - growth (i=1) *)
(* p21 - new cluster creation (innovation) *)
(* p22 - addition (hiring 1 agent) *)
(* p23 - coagulation (merger), r≥2 *)
(* p31 - fragmentation (closure) *)
(* p32 - 1-agent split (firing 1 unit) *)
(* p33 - r-split, r≥2 *)
(* p4 - removal (j=1) *)
(* M=100 *)

(* k=1 *)

$$\mathbf{dz1} = -\frac{n_1 * Z_1}{M * M}\left(p21 * n_1 + p22 * \left(\sum_{r=2}^{M-2} r * n_r\right)\right) + \frac{p31}{M} * \left(\sum_{r=2}^{M-1} r * n_r * Z_r\right) + \frac{p33}{M} * \left(\sum_{r=3}^{M-1} n_r * Z_r\right) + \frac{p1 * \text{KroneckerDelta}[1, i] - p4 * \text{KroneckerDelta}[1, j]}{N}$$

$$\frac{p1\ \text{KroneckerDelta}[1, i] - p4\ \text{KroneckerDelta}[1, j]}{N} - \frac{n_1 Z_1 \left(p21\ n_1 + p22\ \sum_{r=2}^{-2+M} r\ n_r\right)}{M^2} + \frac{p33\ \sum_{r=3}^{-1+M} n_r Z_r}{M} + \frac{p31\ \sum_{r=2}^{-1+M} r\ n_r Z_r}{M}$$

(* k=2 *)

$$dz2 = \frac{1}{M*M}\left(p21*n_1*n_1*Z_1 - 2*p22*n_1*n_2*Z_2 - 2*p23*n_2*Z_2*\left(\sum_{r=2}^{M-3} r*n_r\right)\right) - \frac{2*p31*n_2*Z_2}{M} + \frac{3*p32*n_3*Z_3}{M} + \frac{2*p33}{M}*\left(\sum_{r=4}^{M-1} n_r*Z_r\right) + \frac{p1*\text{KroneckerDelta}[2,i] - p4*\text{KroneckerDelta}[2,j]}{N}$$

$$\frac{p1\ \text{KroneckerDelta}[2,i] - p4\ \text{KroneckerDelta}[2,j]}{N} - \frac{2\,p31\,n_2\,Z_2}{M} + \frac{3\,p32\,n_3\,Z_3}{M} + \frac{p21\,n_1^2\,Z_1 - 2\,p22\,n_1\,n_2\,Z_2 - 2\,p23\,n_2\,Z_2\sum_{r=2}^{-3+M} r\,n_r}{M^2} + \frac{2\,p33\sum_{r=4}^{-1+M} n_r\,Z_r}{M}$$

(* k=3 *)

$$dz3 = \frac{p22*n_1}{M*M}(2*n_2*Z_2 + 2*n_2*Z_1 - 3*n_3*Z_3) - \frac{3*p23*n_3*Z_3}{M*M}*\left(\sum_{r=2}^{M-4} r*n_r\right) - \frac{3*p31*n_3*Z_3}{M} + \frac{p32}{M}(4*n_4*Z_4 - 3*n_3*Z_3) + \frac{3*p33}{M}*\left(\sum_{r=5}^{M-1} n_r*Z_r\right) - \frac{p33*n_3*Z_3}{M} + \frac{p1*\text{KroneckerDelta}[3,i] - p4*\text{KroneckerDelta}[3,j]}{N}$$

$$\frac{p1\ \text{KroneckerDelta}[3,i]-p4\ \text{KroneckerDelta}[3,j]}{N}-\frac{3\,p31\ n_3 Z_3}{M}-\frac{p33\ n_3 Z_3}{M}+\frac{p22\ n_1(2n_2Z_1+2n_2Z_2-3n_3Z_3)}{M^2}+\frac{p32\ (-3n_3Z_3+4n_4Z_4)}{M}-\frac{3\,p23\ n_3Z_3\sum_{r=2}^{-4+M} r\,n_r}{M^2}+\frac{3\,p33\ \sum_{r=5}^{-1+M} n_r Z_r}{M}$$

(* k=4,5,...,M-3 *)

dzk =

$$\sum_{k=4}^{M-3}\left(\frac{p22*n_1}{M*M}((k-1)*n_{k-1}*Z_{k-1}+(k-1)*n_{k-1}*Z_1-k*n_k*Z_k)+\frac{p23}{M*M}*\left(\sum_{r=2}^{k-2} r*n_r*(k-r)*n_{k-r}*Z_r-k*n_k*Z_k*\left(\sum_{r=2}^{M-k-1} r*n_r\right)\right)-\frac{k*p31*n_k*Z_k}{M}+\frac{p32}{M}((k+1)*n_{k+1}*Z_{k+1}-k*n_k*Z_k)+\frac{p33}{M}*\left(k*\left(\sum_{r=k+2}^{M-1} n_r*Z_r\right)-n_k*Z_k*\left(\sum_{r=1}^{k-2} r\right)\right)+\frac{p1*\text{KroneckerDelta}[k,i]-p4*\text{KroneckerDelta}[k,j]}{N}\right)$$

$$\sum_{k=4}^{-3+M}\left(\frac{\text{p1 KroneckerDelta}[i,k]-\text{p4 KroneckerDelta}[j,k]}{N}-\frac{k\,\text{p31}\,n_k\,Z_k}{M}+\frac{\text{p22}\,n_1\left((-1+k)\,n_{-1+k}\,Z_1+(-1+k)\,n_{-1+k}\,Z_{-1+k}-k\,n_k\,Z_k\right)}{M^2}+\frac{\text{p32}\left(-k\,n_k\,Z_k+(1+k)\,n_{1+k}\,Z_{1+k}\right)}{M}+\frac{\text{p33}\left(-\frac{1}{2}(-2+k)(-1+k)\,n_k\,Z_k+k\sum_{r=2+k}^{-1+M}n_r\,Z_r\right)}{M}+\frac{\text{p23}\left(-k\,n_k\,Z_k\sum_{r=2}^{-1-k+M}r\,n_r+\sum_{r=2}^{-2+k}(k-r)\,r\,n_{k-r}\,n_r\,Z_r\right)}{M^2}\right)$$

(* k=M-2 *)

dzm2 =

$$\frac{\text{p22}*n_1}{M*M}\left((M-3)*n_{M-3}*Z_{M-3}+(M-3)*n_{M-3}*Z_1-(M-2)*n_{M-2}*Z_{M-2}\right)+\frac{\text{p23}}{M*M}*\left(\sum_{r=2}^{M-4}r*n_r*(M-r-2)*n_{M-r-2}*Z_r\right)-\frac{(M-2)*\text{p31}*n_{M-2}*Z_{M-2}}{M}+\frac{\text{p32}}{M}\left((M-1)*n_{M-1}*Z_{M-1}-(M-2)*n_{M-2}*Z_{M-2}\right)-\frac{\text{p33}*n_{M-2}*Z_{M-2}}{M}*\left(\sum_{r=1}^{M-4}r\right)+\frac{1}{N}\left(\text{p1}*\text{KroneckerDelta}[M-2,i]-\text{p4}*\text{KroneckerDelta}[M-2,j]\right)$$

$$\frac{p1\ \text{KroneckerDelta}[i, -2+M] - p4\ \text{KroneckerDelta}[j, -2+M]}{N} - \frac{(-2+M)\ p31\ n_{-2+M} Z_{-2+M}}{M} - \frac{(-4+M)(-3+M)\ p33\ n_{-2+M} Z_{-2+M}}{2M} + \frac{1}{M^2}(p22\ n_1((-3+M)\ n_{-3+M} Z_1 + (-3+M)\ n_{-3+M} Z_{-3+M} - (-2+M)\ n_{-2+M} Z_{-2+M})) + \frac{p32\ (-(-2+M)\ n_{-2+M} Z_{-2+M} + (-1+M)\ n_{-1+M} Z_{-1+M})}{M} + \frac{p23 \sum_{r=2}^{-4+M}(-2+M-r)\ r\ n_{-2+M-r} n_r Z_r}{M^2}$$

(* k=M-1 *)

$$\mathbf{dzm1} = \frac{p22 * n_1}{M * M}((M-2) * n_{M-2} * Z_{M-2} + (M-2) * n_{M-2} * Z_1) + \frac{p23}{M * M} * \left(\sum_{r=2}^{M-3} r * n_r * (M-r-1) * n_{M-r-1} * Z_r\right) - \frac{(M-1) * p31 * n_{M-1} * Z_{M-1}}{M} - \frac{(M-1) * p32 * n_{M-1} * Z_{M-1}}{M} - \frac{p33 * n_{M-1} * Z_{M-1}}{M} * \left(\sum_{r=1}^{M-3} r\right) + \frac{1}{N}(p1 * \text{KroneckerDelta}[M-1, i] - p4 * \text{KroneckerDelta}[M-1, j])$$

$$\frac{p1\ \text{KroneckerDelta}[i, -1+M] - p4\ \text{KroneckerDelta}[j, -1+M]}{N} + \frac{p22\ n_1((-2+M)\ n_{-2+M} Z_1 + (-2+M)\ n_{-2+M} Z_{-2+M})}{M^2} - \frac{(-1+M)\ p31\ n_{-1+M} Z_{-1+M}}{M} - \frac{(-1+M)\ p32\ n_{-1+M} Z_{-1+M}}{M} - \frac{(-3+M)(-2+M)\ p33\ n_{-1+M} Z_{-1+M}}{2M} + \frac{p23 \sum_{r=2}^{-3+M}(-1+M-r)\ r\ n_{-1+M-r} n_r Z_r}{M^2}$$

s=dz1+dz2+dz3+dzk+dzm2+dzm1

(* For closed system: $\sum_{k=1}^{M} Z_k(t)=$
1 → $\sum_{k=1}^{M}(dZ_k(t)/dt)=d(\sum_{k=1}^{M} Z_k(t))/dt)=0$ *)

M=100;
i=1;
j=1;
Simplify[st]

$$\frac{p1 - p4}{N}$$

12345	*6*	*Programul*
c		General model of cluster formation and non-stationary
c		cluster size distribution
c		************* Z(k,t) ***************
c		Raw data at t=0, tab1_t0.xls
c		M(t)=var
c		Preferential addition
c		p1 - growth by single monomers (i=1)
c		p21 - new cluster creation (innovation)
c		p22 - addition (hiring 1 agent)
c		p23 - coagulation (merger), r>=2
c		p31 - fragmentation of a cluster into individual agents
c		(firm closure)
c		p32 - 1-monomer split
c		p33 - r-split, r>=2
c		p4 - removal of single monomers (j=1)
c		
		INTEGER init, timax, tmax, ar, iar
		DOUBLE PRECISION A, B, C, MS, NTS, NS, MSN,
	&	MD, NV, TS, p1, p2cr, p21, p2, p3cl, p31, p3s, p4, p
		PARAMETER (timax=50)
		PARAMETER (p1=0.0)
		PARAMETER (p21=0.2, p22=0.3, p23=0.1)
		PARAMETER (p31=0.1, p32=0.3, p33=0.0)

```
              PARAMETER (init=114823, ar=116749, iar=1507)
              DOUBLE PRECISION n(ar, timax), z(ar, timax),
        &     sz(timax), M(timax), NT(timax), MNT(timax), ik(iar),
        &     n0(iar)
c
c             open data files and get initial data for simulation
c             11 = output file for [ t, M(t), N(t), M/N, S(t) ]
c             12 = output file for [ t, k, n(k,t), z(k,t) ]
c             13, 14 = initial empirical (raw) data for n(k,1) different
c             0.
c             15 = output file for all n(k,1) up to maximum k (cut-off)
c
              open(11, file='mnt.txt')
              open(12, file='nz1t.txt')
              open(13, file='tab1_k.txt')
              open(14, file='tab1_nk.txt')
              open(15, file='tab1_all.txt')
c
              READ(13, 110) (ik(k),k=1,1507)
              READ(14, 110) (n0(k),k=1,1507)
110           FORMAT(F10.0)
c
              DO k = 1, ar
              n(k, 1) = 0.0
              DO j = 1, iar
              IF (k.EQ.ik(j)) THEN
              n(k, 1) = n0(j)
              END IF
              END DO
              END DO
c             M(1) = number of agents at initial moment (a priori
c             initial condition)
c             tmax   = maximum length of computation
c             p1 + p2cr + p21 + p2 + p3cl + p31 + p3s + p4 = 1
c             n(ar,t)=0 -> Z(ar,t)=0 boundary condition
c             ('ar'(not 'M'!) is cut-off from raw data)
c             Z(k,t) non-stationary cluster size distribution
```

```
c           sz(t) = S(t) entropy
c
            t=1
            MS=0
            NTS=0
            DO k = 1, ar
            MS=MS+k*n(k, t)
            NTS=NTS+n(k, t)
            END DO
c
            M(t)=MS
            NT(t)=NTS
            MNT(t)=M(t)/NT(t)
c
            DO k = 1, ar
            z(k, t) = n(k, t)/NT(t)
            END DO
c
            tmax=timax-1
c
            p4=1.0-p1-p21-p22-p23-p31-p32-p33
c
            p=p1+p21+p22+p23+p31+p32+p33+p4
            IF (p.LE.1) THEN
            GOTO 101
            ELSE
            WRITE (*, 11)
11          FORMAT (/5x,'TOTAL PROBABILITY IS > 1 -> re-
      &     define Pi !')
            GOTO 201
            END IF
101         CONTINUE
c
            DO k = 1, ar
            WRITE (*, 33) k, n(k,1), z(k,1)
            WRITE (15, 34) k, n(k,1), z(k,1)
            END DO
```

```
33          FORMAT (/2x,'k:', I7,
     &              5x, 'nk:', F10.0,
     &              5x, 'zk:', F10.5)
34          FORMAT (I7, F10.0, F10.5)
c
            NV=0.0
            DO k = 1, ar
            TS=k*n(k, t)/M(t)
            IF (TS.EQ.0) THEN
            TS=0.000000000001
            END IF
            NV=NV+TS*LOG(TS)
            END DO
            sz(t)=-NV
c
            WRITE (*, 112) p1, p21, p22, p23, p31, p32, p33, p4
            WRITE (*, 114) t, M(t), NT(t), MNT(t), sz(t)
            WRITE (11, 113) p1, p21, p22, p23, p31, p32, p33, p4
            WRITE (11, 115) t, M(t), NT(t), MNT(t), sz(t)
112         FORMAT (/2x,'p1 :', F3.1,
     &             5x, 'p21:', F3.1,
     &              5x, 'p22:', F3.1,
     &              5x, 'p23:', F3.1,
     &              5x, 'p31:', F3.1,
     &              5x, 'p32:', F3.1,
     &              5x, 'p33:', F3.1,
     &              5x,'p4 :', F3.1)
113         FORMAT (F5.1, F5.1, F5.1, F5.1, F5.1, F5.1, F5.1,
     &      F5.1)
114         FORMAT (/2x,'step:', F7.1,
     &             5x, 'M:', F10.1,
     &             5x,'N:', F8.1,
     &             5x,'M/N:', F9.3,
     &             5x,'S:', F9.5)
115         FORMAT (F7.1, F12.1, F12.1, F9.3, F9.5)
c
c           main loop 2 <= t <= tmax
```

```
c        tmax=timax-1
c
         DO t=2, tmax
c
c        compute z(1, t)
c
         A=0.0
         B=0.0
         C=0.0
         DO r = 2, ar-2
         A = A + r*n(r,t-1)
         END DO
         DO r = 2, ar-1
         B = B + r*n(r,t-1)*z(r,t-1)
         END DO
         DO r = 3, ar-1
         C = C + n(r,t-1)*z(r,t-1)
         END DO
         z(1,t) = z(1,t-1) + (p1 - p4)/NT(t-1)
     &   - n(1,t-1)*z(1,t-1)*(p21*n(1,t-1)
     &   +A*p22)/(M(t-1)*M(t-1))
     &   + B*p31/M(t-1) + C*p33/M(t-1)
c
c        compute z(2, t)
c
         A=0.0
         B=0.0
         DO r = 2, ar-3
         A = A + r*n(r,t-1)
         END DO
         DO r = 4, ar-1
         B = B + n(r,t-1)*z(r,t-1)
         END DO
         z(2,t) = z(2,t-1) + (p21*n(1,t-1)*n(1,t-1)*z(1,t-1)
     &    - 2*p22*n(1,t-1)*n(2,t-1)*z(2,t-1)
     &    - 2*A*p23*n(2,t-1)*z(2,t-1))/(M(t-1)*M(t-1))
     &    - 2*n(2,t-1)*z(2,t-1)*p31/M(t-1)
```

```
     &  + 3*n(3,t-1)*z(3,t-1)*p32/M(t-1) + 2*B*p33/M(t-1)
c
c       compute z(3, t)
c
        A=0.0
        B=0.0
        DO r = 2, ar-4
        A = A + r*n(r,t-1)
        END DO
        DO r = 5, ar-1
        B = B + n(r,t-1)*z(r,t-1)
        END DO
        z(3,t) = z(3,t-1) + (p22*n(1,t-1)*(2*n(2,t-1)*z(2,t-1)
     &  + 2*n(2,t-1)*z(1,t-1) - 3*n(3,t-1)*z(3,t-1))
     &  - 3*A*p23*n(3,t-1)*z(3,t-1))/(M(t-1)*M(t-1))
     &  - 3*n(3,t-1)*z(3,t-1)*p31/M(t-1)
     &  + p32*(4*n(4,t-1)*z(4,t-1)-3*n(3,t-1)*z(3,t-1))/M(t-1)
     &  + 3*B*p33/M(t-1)-n(3,t-1)*z(3,t-1)*p33/M(t-1)
c
c       compute z(k, t), 4 <= k <= ar-3
c
        DO k = 4, ar-3
        A=0.0
        B=0.0
        C=0.0
        DO r = 2, k-2
        A = A + r*n(r,t-1)*(k-r)*n(k-r,t-1)*z(r,t-1)
        END DO
        DO r = 2, ar-k-1
        B = B + r*n(r,t-1)
        END DO
        DO r = k+2, ar-1
        C = C + n(r,t-1)*z(r,t-1)
        END DO
        z(k,t) = z(k,t-1)
     &  + (p22*n(1,t-1)*((k-1)*n(k-1,t-1)*z(k-1,t-1)
     &  + (k-1)*n(k-1,t-1)*z(1,t-1) - k*n(k,t-1)*z(k,t-1))
```

```
     &  + p23*(A-B*k*n(k,t-1)*z(k,t-1)))/(M(t-1)*M(t-1))
     &  - k*n(k,t-1)*z(k,t-1)*p31/M(t-1)
     &  + p32*((k+1)*n(k+1,t-1)*z(k+1,t-1)
     &  - k*n(k,t-1)*z(k,t-1))/M(t-1)
     &  + C*k*p33/M(t-1)
     &  - (k-1)*(k-2)*n(k,t-1)*z(k,t-1)*p33/(2*M(t-1))
c
        END DO
c
c       compute z(ar-2, t)
c
        k=ar-2
        A=0.0
        DO r = 2, k-2
        A = A + r*n(r,t-1)*(k-r)*n(k-r,t-1)*z(r,t-1)
        END DO
        z(k,t) = z(k,t-1)
     &  + (p22*n(1,t-1)*((k-1)*n(k-1,t-1)*z(k-1,t-1)
     &  + (k-1)*n(k-1,t-1)*z(1,t-1) - k*n(k,t-1)*z(k,t-1))
     &  + p23*A)/(M(t-1)*M(t-1))
     &  - k*n(k,t-1)*z(k,t-1)*p31/M(t-1)
     &  + p32*((k+1)*n(k+1,t-1)*z(k+1,t-1)
     &  - k*n(k,t-1)*z(k,t-1))/M(t-1)
     &  -(k-1)*(k-2)*n(k,t-1)*z(k,t-1)*p33/(2*M(t-1))
c
c       compute z(ar-1, t)
c
        k=ar-1
        A=0.0
        DO r = 2, k-2
        A = A + r*n(r,t-1)*(k-r)*n(k-r,t-1)*z(r,t-1)
        END DO
        z(k,t) = z(k,t-1)
     &  + (p22*n(1,t-1)*((k-1)*n(k-1,t-1)*z(k-1,t-1)
     &  + (k-1)*n(k-1,t-1)*z(1,t-1)) + p23*A)/(M(t-1)*M(t-1))
     &  - k*n(k,t-1)*z(k,t-1)*p31/M(t-1)
     &  - k*n(k,t-1)*z(k,t-1)*p32/M(t-1)
```

```
     &  - (k-1)*(k-2)*n(k,t-1)*z(k,t-1)*p33/(2*M(t-1))
c
c       Boundary condition
c
        z(ar,t)=0.0
c
c       compute, show and save z(k,t), n(k,t), M(t), N(t), M/N
c
        NS=0.0
        DO k = 1, ar
        NS=NS+k*z(k, t)
        END DO
c
        MNT(t)=NS
c
        M(t)=M(t-1)+p1-p4
c
        NT(t)=M(t)/MNT(t)
c
        DO k = 1, ar
        n(k, t)=z(k, t)*NT(t)
        END DO
c
        NV=0.0
        DO k = 1, ar
        TS=k*n(k, t)/M(t)
        IF (TS.EQ.0) THEN
        TS=0.000000000001
        END IF
        NV=NV+TS*LOG(TS)
        END DO
        sz(t)=-NV
c
        IF (M(t).LE.1) THEN
        WRITE (*, 22)
22      FORMAT (/5x,'!!!AT THIS STEP YOU HAVE NO
     &  AGENTS IN THE SYSTEM!!!')
```

```
              GOTO 201
              ELSEIF (NT(t).LE.1) THEN
              WRITE (*, 24)
24            FORMAT (/5x,'!!!THERE IS A SINGLE CLUSTER IN
          &   THE SYSTEM!!!')
              GOTO 201
              ELSE
              GOTO 132
              END IF
132           CONTINUE
c
              WRITE (*, 214) t, M(t), NT(t), MNT(t), sz(t)
              WRITE (11, 215) t, M(t), NT(t), MNT(t), sz(t)
214           FORMAT (/2x,'step:', F7.1,
          &                 5x, 'M:', F10.1,
          &                 5x, 'N:', F8.1,
          &                 5x, 'M/N:', F9.3,
          &                 5x, 'S:', F9.5)
215           FORMAT (F7.1, F12.1, F12.1, F9.3, F9.5)
c
              DO k = 1, ar
              WRITE (12, 216) t, k, n(k,t), z(k,t)
              END DO
216           FORMAT (F8.1, I7, F9.3, F7.3)
c
              END DO
c
201           CONTINUE
c
              STOP
              END
```

12345	*6*	*Programul*
c		**Programul de calcul pentru modelul probabilistic**
c		Probability Model * Step=0.05 * Non-zero Pi, i=1-6 *
c		* J-rule=c, H-rule=h *
c		
		INTEGER t, q, qp, y
		PARAMETER (tmax=53130, st=19)
		REAL n(tmax), nc(tmax), nh(tmax), d(tmax), vc(tmax),
	&	vh(tmax), var(tmax)
		DOUBLE PRECISION p1, p2, p3, p4, p5, p6, p, pr1,
	&	pr2, pr3, pr4, pr5, pr6, z, ne, zv, nev, np, npv, r, rv,
	&	dmax, di, dj1, dj2, init
c		
c		open data files
c		
c		11 = output file for probabilities, payoffs n(p1,...,p6),
c		variance, total ratio r
c		12 = output file for payoffs at the same distance on the
c		simplex
c		13 = output file for ratio as function of distance from
c		O(0,0,0)
c		14 = output file for distance at each step t of run
c		
		open(11, file='payoffs.txt')
		open(12, file='simplex.txt')
		open(13, file='ratio.txt')
		open(14, file='distance.txt')
c		
c		pi = probabilities: i=1,...,6 SUM(pi)= 1 (x 20 for step
c		0.05)
c		n = equations for difference nc-nh
c		var=vc-vh = variance at each step t of run
c		r=(ne+z)/(ne+np+z) = ratio non-positive/total number of
c		payoffs
c		
		p=0.0
		p6=0

```
            t=0
            z=0
            ne=0
            zv=0
            nev=0
c
            DO p1 = 0, 100, 5
            DO p2 = 0, 100, 5
            DO p3 = 0, 100, 5
            DO p4 = 0, 100, 5
            DO p5 = 0, 100, 5
c
            p=p1+p2+p3+p4+p5
            p6=100-p
c
            IF (p.LE.100) THEN
            GOTO 101
            ELSEIF (p1.LT.100) THEN
            GOTO 102
            ELSEIF (p2.LT.100) THEN
            GOTO 102
            ELSEIF (p3.LT.100) THEN
            GOTO 102
            ELSEIF (p4.LT.100) THEN
            GOTO 102
            ELSEIF (p5.LT.100) THEN
            GOTO 102
            ELSE
            WRITE (*, 11)
11          FORMAT (/5x,'THE RUN IS FINISHED')
            GOTO 201
            END IF
101         CONTINUE
c
            t=t+1
c
            nc(t)=((p1+p2)*(p1+p2-p3-p5)+(p3+p4)*(p3+p4-p1-p6)
```

```
     &  + (p5+p6)*(p5+p6-p2-p4))/10000
c
        nh(t)=(p1*(p1+p6-p3-p5)+p2*(p2+p4-p3-p5)
     &  + p3*(p3+p5-p1-p6)+p4*(p2+p4-p1-p6)
     &  + p5*(p3+p5-p2-p4)+p6*(p1+p6-p2-p4))/20000
c
        n(t)=nc(t)-nh(t)
c
c       variance
c
        vc(t)=(p1/100)*((p1+p2-p3-p4)/100-nc(t))*((p1+p2-p3-
     &  p4)/100-nc(t)) + (p2/100)*((p1+p2-p5-p6)/100-
     &  nc(t))*((p1+p2-p5-p6)/100-nc(t)) + (p3/100)*((p3+p4-
     &  p1-p2)/100-nc(t))*((p3+p4-p1-p2)/100-nc(t))
     &  + (p4/100)*((p3+p4-p5-p6)/100-nc(t))*((p3+p4-p5-
     &  p6)/100-nc(t)) + (p5/100)*((p5+p6-p1-p2)/100-
     &  nc(t))*((p5+p6-p1-p2)/100-nc(t)) + (p6/100)*((p5+p6-
     &  p3-p4)/100-nc(t))*((p5+p6-p3-p4)/100-nc(t))
c
        vh(t)=(p1/100)*((p1+p6-p3-p5)/200-nh(t))*((p1+p6-p3-
     &  p5)/200-nh(t)) + (p2/100)*((p2+p4-p3-p5)/200-
     &  nh(t))*((p2+p4-p3-p5)/200-nh(t)) + (p3/100)*((p3+p5-
     &  p1-p6)/200-nh(t))*((p3+p5-p1-p6)/200-nh(t))
     &  + (p4/100)*((p2+p4-p1-p6)/200-nh(t))*((p2+p4-p1-
     &  p6)/200-nh(t)) + (p5/100)*((p3+p5-p2-p4)/200-
     &  nh(t))*((p3+p5-p2-p4)/200-nh(t)) + (p6/100)*((p1+p6-
     &  p2-p4)/200-nh(t))*((p1+p6-p2-p4)/200-nh(t))
c
        var(t)=vc(t)-vh(t)
c
        IF (var(t).EQ.0) THEN
        zv=zv+1
        END IF
        IF (var(t).LT.0) THEN
        nev=nev+1
        END IF
        IF (n(t).EQ.0) THEN
```

```
      z=z+1
      END IF
      IF (n(t).LT.0) THEN
      ne=ne+1
      END IF
c
c     distance from the origin of simplex
c
      l=(p1+p2-p3-p5)/100
      c=(p3+p4-p1-p6)/100
      ri=(p5+p6-p2-p4)/100
c
      d(t)=SQRT(l*l+c*c+ri*ri)
c
      WRITE (*, 30) t, d(t)
      WRITE (14, 31) d(t)
30    FORMAT (I9, F10.7)
31    FORMAT (F10.7)
c
c     show on screen and save results in *.txt files
c
      pr1=p1/100
      pr2=p2/100
      pr3=p3/100
      pr4=p4/100
      pr5=p5/100
      pr6=p6/100
      WRITE (*, 12) t, pr1, pr2, pr3, pr4, pr5, pr6, nc(t),
     & nh(t), n(t), vc(t), vh(t), var(t)
      WRITE (11, 13) t, pr1, pr2, pr3, pr4, pr5, pr6, nc(t),
     & nh(t), n(t), vc(t), vh(t), var(t)
12    FORMAT (/2x,' t:', I9,
     &               4x, 'p1:', F5.2,
     &               4x, 'p2:', F5.2,
     &               4x, 'p3:', F5.2,
     &               4x, 'p4:', F5.2,
     &               4x, 'p5:', F5.2,
```

```
        &             4x, 'p6:', F5.2,
        &             4x, 'nc:', F7.3,
        &             4x, 'nh:', F7.3,
        &             4x, ' n:', F7.3,
        &             4x, 'vc:', F7.3,
        &             4x, 'nh:', F7.3,
        &             4x, 'var:', F7.3)
13          FORMAT (I9, F8.2, F7.2, F7.2, F7.2, F7.2, F7.2,
        &   F15.10, F15.10, F15.10, F15.10, F15.10, F15.10)
c
102         CONTINUE
            END DO
            END DO
            END DO
            END DO
            END DO
c
201         CONTINUE
c
            r=(ne+z)/t
c
            WRITE (*, 14) t, ne, z, r
            WRITE (11, 15) t, ne, z, r
14          FORMAT (//2x, ' t:', I9,
        &             5x, 'ne:', F7.1,
        &             5x, ' z: ', F7.1,
        &             5x, ' r:', F7.3)
15          FORMAT (I9, F10.1, F7.1, F7.3)
c
            rv=(nev+zv)/t
c
            WRITE (*, 25) t, nev, zv, rv
            WRITE (11, 26) t, nev, zv, rv
25          FORMAT (//2x,'  t:', I9,
        &             5x, 'nev:', F7.1,
        &             5x, ' zv: ', F7.1,
        &             5x, ' rv:', F7.3)
```

```
26          FORMAT (I9, F10.1, F7.1, F7.3)
c
c           compute and plot ratio as function of distance
c
            dmax=0
            DO t = 1, tmax
            IF (d(t).GT.dmax) THEN
            dmax=d(t)
            END IF
            END DO
c
            init=0
            z=0
            ne=0
            np=0
            zv=0
            nev=0
            npv=0
            DO t = 1, tmax
            IF (d(t).EQ.0) THEN
            CONTINUE
            IF (n(t).EQ.0) THEN
            z=z+1
            END IF
            IF (n(t).LT.0) THEN
            ne=ne+1
            END IF
            IF (n(t).GT.0) THEN
            np=np+1
            END IF
            END IF
            IF (d(t).EQ.0) THEN
            CONTINUE
            IF (var(t).EQ.0) THEN
            zv=zv+1
            END IF
            IF (var(t).LT.0) THEN
```

```
            nev=nev+1
            END IF
            IF (var(t).GT.0) THEN
            npv=npv+1
            END IF
            END IF
            END DO
c
            r=(ne+z)/(ne+np+z)
            rv=(nev+zv)/(nev+npv+zv)
c
            WRITE (*, 27) init, r, rv, z, ne, np, zv, nev, npv
            WRITE (13, 28) init, r, rv, z, ne, np, zv, nev, npv
27          FORMAT (//2x, 'd:', F7.3,
        &               5x, 'r:', F7.3,
        &               5x, 'rv:', F7.3,
        &               5x, 'z: ', F7.1,
        &               5x, 'ne: ', F7.1,
        &               5x, 'np: ', F7.1,
        &               5x, 'zv: ', F7.1,
        &               5x, 'nev: ', F7.1,
        &               5x, 'npv:', F7.1)
28          FORMAT (F7.5, F7.3, F7.3, F10.1, F7.1, F7.1, F7.1,
        &   F7.1, F7.1)
c
            di=dmax/(st+1)
c
            DO j = 0, st
            dj1=j*di
            dj2=(j+1)*di
            z=0
            ne=0
            np=0
            zv=0
            nev=0
            npv=0
            DO t = 1, tmax
```

```
          IF (d(t).GT.dj1 .AND. d(t).LE.dj2) THEN
          GOTO 203
          ELSE
          GOTO 204
          END IF
203       CONTINUE
c
          IF (n(t).EQ.0) THEN
          z=z+1
          END IF
          IF (n(t).LT.0) THEN
          ne=ne+1
          END IF
          IF (n(t).GT.0) THEN
          np=np+1
          END IF
204       CONTINUE
c
          IF (d(t).GT.dj1 .AND. d(t).LE.dj2) THEN
          GOTO 303
          ELSE
          GOTO 304
          END IF
303       CONTINUE
c
          IF (var(t).EQ.0) THEN
          zv=zv+1
          END IF
          IF (var(t).LT.0) THEN
          nev=nev+1
          END IF
          IF (var(t).GT.0) THEN
          npv=npv+1
          END IF
304       CONTINUE
c
          END DO
```

```
c
        r=(ne+z)/(ne+np+z)
        rv=(nev+zv)/(nev+npv+zv)
c
        WRITE (*, 21) dj2, r, rv, z, ne, np, zv, nev, npv
        WRITE (13, 22) dj2, r, rv, z, ne, np, zv, nev, npv
21      FORMAT (//2x, 'd:', F7.5,
     &              5x, 'r:', F7.3,
     &              5x, 'rv:', F7.3,
     &              5x, 'z: ', F7.1,
     &              5x, 'ne: ', F7.1,
     &              5x, 'np: ', F7.1,
     &              5x, 'zv: ', F7.1,
     &              5x, 'nev: ', F7.1,
     &              5x,'npv:', F7.1)
22      FORMAT (F7.5, F7.3, F7.3, F10.1, F7.1, F7.1, F7.1,
     &  F7.1, F7.1)
c
        END DO
c
c       payoffs at the same distance on the simplex
c
        dd=d(1)
        qp=1
        q=2
202     CONTINUE
        y=0
        DO t = q, tmax
        IF (d(t).EQ.dd) THEN
        y=y+1
c
        WRITE (*, 16) y, qp, d(qp), nc(qp), nh(qp), n(qp)
        WRITE (*, 17) t, d(t), nc(t), nh(t), n(t)
        WRITE (12, 18) y
        WRITE (12, 19) qp, d(qp), nc(qp), nh(qp), n(qp)
        WRITE (12, 20) t, d(t), nc(t), nh(t), n(t)
16      FORMAT (//5x, 'step:', I7,
```

```
          &      //5x, 't-1:', I5,
          &        5x, 'd:', F7.3,
          &        5x, 'nc:', F7.3,
          &        5x, 'nh:', F7.3,
          &        5x, 'n:', F7.3)
17             FORMAT ( /5x, '  t:', I5,
          &        5x, 'd:', F7.3,
          &        5x, 'nc:', F7.3,
          &        5x, 'nh:', F7.3,
          &        5x, 'n:', F7.3)
18             FORMAT (/, I7)
19             FORMAT (I5, F7.3, F7.3, F7.3, F7.3)
20             FORMAT (I5, F7.3, F7.3, F7.3, F7.3)
               END IF
               END DO
205            CONTINUE
c
               rem=0
               DO t = 1, qp
               IF (d(q).EQ.d(t)) THEN
               rem=rem+1
               END IF
               END DO
c
               qp=q
               dd=d(q)
               q=q+1
               IF (rem.EQ.0) THEN
               GOTO 206
               ELSEIF (q.EQ.tmax) THEN
               GOTO 207
               ELSE
               GOTO 205
               END IF
206            CONTINUE
c
               IF (q.LT.tmax) THEN
```

		GOTO 202
		END IF
207		CONTINUE
c		
		STOP
		END

*** Probability Model * 3D Plots ***

<<Graphics`Graphics3D`

plot1_data=Import["C://Documents and Settings/florentin/Desktop/payoff2.txt","Table"]

(* show your own pass above ! *)

{{0.,0.,1.},{0.,0.1,0.64},{0.,0.2,0.36},{0.,0.3,0.16},{0.,0.4,0.04},{0.,0.5,0.},{0.,0.6,0.04},{0.,0.7,0.16},{0.,0.8,0.36},{0.,0.9,0.64},{0.,1.,1.},{0.1,0.,0.73},{0.1,0.1,0.4},{0.1,0.2,0.15},{0.1,0.3,-0.02},{0.1,0.4,-0.11},{0.1,0.5,-0.12},{0.1,0.6,-0.05},{0.1,0.7,0.1},{0.1,0.8,0.33},{0.1,0.9,0.64},{0.2,0.,0.52},{0.2,0.1,0.22},{0.2,0.2,0.},{0.2,0.3,-0.14},{0.2,0.4,-0.2},{0.2,0.5,-0.18},{0.2,0.6,-0.08},{0.2,0.7,0.1},{0.2,0.8,0.36},{0.3,0.,0.37},{0.3,0.1,0.1},{0.3,0.2,-0.09},{0.3,0.3,-0.2},{0.3,0.4,-0.23},{0.3,0.5,-0.18},{0.3,0.6,-0.05},{0.3,0.7,0.16},{0.4,0.,0.28},{0.4,0.1,0.04},{0.4,0.2,-0.12},{0.4,0.3,-0.2},{0.4,0.4,-0.2},{0.4,0.5,-0.12},{0.4,0.6,0.04},{0.5,0.,0.25},{0.5,0.1,0.04}, {0.5,0.2,-0.09},{0.5,0.3,-0.14},{0.5,0.4,-0.11},{0.5,0.5,0.},{0.6,0.,0.28},{0.6,0.1,0.1},{0.6,0.2,0.},{0.6,0.3,-0.02},{0.6,0.4,0.04},{0.7,0.,0.37},{0.7,0.1,0.22},{0.7,0.2,0.15},{0.7,0.3,0.16},{0.8,0.,0.52},{0.8,0.1,0.4},{0.8,0.2,0.36},{0.9,0.,0.73},{0.9,0.1,0.64},{1.,0.,1.}}

plot1=ScatterPlot3D[plot1_data, PlotStyle → {Hue[0], PointSize[0.03]}, ViewPoint → {1.2,1.0,-0.26}];

Show[%, ViewVertical → {1,0,0}];

plot2_data=Import["C://Documents and Settings/florentin/Desktop/bar.txt","Table"]

{{0.014,0.046,0.057,0.049,0.021,-0.028,-0.096,-0.184,-0.293,-0.421,-0.569},{-0.024,0.002,0.009,-0.004,-0.038,-0.091,-0.164,-0.258,-0.371,-0.504,0.},{-0.023,-0.001,0.001,-0.018,-0.056,-0.114,-0.193,-0.291,-0.409,0.,0.},{0.019,0.036,0.032,0.009,-0.034,-0.098,-0.181,-0.284,0.,0.,0.},{0.101,0.112,0.104,0.076,0.027,-0.041,-0.129,0.,0.,0.,0.},{0.222,0.229,0.216,0.182,0.129,0.056,0.,0.,0.,0.,0.},{0.384,0.386,0.367,0.329,0.271,0.,0.,0.,0.,0.,0.},{0.586,0.582,0.559,0.516,0.,0.,0.,0.,0.,0.,0.},{0.827,0.819,0.791,0.,0.,0.,0.,0.,0.,0.,0.},{1.109,1.096,0.,0.,0.,0.,0.,0.,0.,0.,0.},{1.431,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.}}

plot2a=BarChart3D[plot2_data, AxesLabel → {"sample chart", "", "payoffs"}];
plot2b=BarChart3D[plot2_data, ViewPoint → {-2,-2,2}];
plot2c=BarChart3D[plot2_data, ViewPoint → {-2,-2,-1}];
Exemplu: