APPENDIX to:

Molecules and Thoughts

Pattern Complexity and Evolution in Chemical Systems and the Mind

http://spirospero.net/mindscale.pdf ; http://spirospero.net/MINDSCALE.pdf

http://www.scribd.com/doc/11576667/-Yuri-Tarnopolsky-MOLECULES-AND-THOUGHTS

Yuri Tarnopolsky

PROTO

WORKSPACE: **PROTO.mat**
http://spirospero.net/PROTO.mat

The MATLAB program PROTO requires workspace **PROTO.mat**, is stored as scripts **proto.m** (main), **proto1.m**, and **proto2.m**, and generates a figure of the tree with red asterisks meaning acts of selection (actons).

To activate the program, save workspace (matrices **P**, **T**, and **GXY**) and the scripts. The main program offers a choice of starting from equal probability distribution (proto1) or continuing from the previous workspace, parameters, and probability distribution (proto2).

proto.m

```
%PROTO; script proto; can be renamed


S=input('1 to start, 2 to continue   ');

if S==1
   proto1
   end
if S==2
   proto2
   end
```

proto1.m

```
%PROTO1; TO START; file proto1.m; workspace proto.mat
%SETS PARAMETERS AND STARTS FIRST BATCH
% can be followed by proto2.m
%Workspace: P(32x11), T(31x31), GXY (31,2).

%Initial settings:

H=0.7; F=0.9; C=1; gup=0.5; w=2; a=2; t=0.1;

%%%%%%%PARAMETERS SECTION %%%%%%%%%%%%%%%%

n=input ('Enter n  ');%n=20;
a=input('Enter a ');% a=2;
%b=input('Enter b '); %Optional second acton
F=input('Enter F ' );%F=0.8;
C=input('Enter C ');%C=1;
H=input('Enter H ');%H=0.8;

%gup=input('Enter gup  ');
gup=0.5;
%w=input('Enter w ');
w=2;
gdown=w*gup;

        %gdown=input('Enter gdown ');
        %gdown=2;

%t=input('Enter pause t ' );
%t=0.1;
tra=[];          % tra: trajectory
ar=0;            %ar: arity
aa=a;    %aa: cell for the previous acton;

%%%%%%%%%%  END of PARAMETERS SECTION
%%%%%%%%%%%%%%%%%

PAR12=['a=',num2str(a)];
text (0.01,0.3,PAR12); %displays starting acton


PX=P(1:31,2)/14; PY=P(1:31,3)/8.5; P(32,7)=0;
%factors 14 and 8.8 to fit the figure

%COLUMNS in matrix P: 1: Node #; 2,3: Node coordinate, 4-6: Node neighborhood;
```

```
% 7: Probability,p; 8: Influence,v; 9: Active node,A; 10: Memory,M;
%11: Spare column.  Line 32:zeros.

%INITIAL MATRIX P. Equal probabilities.
        P(1:31,7)=1/31; P(:,8:11)=0; %Columns 7 to 11 contain variables

  % Data for the acton:
  P(a,7)=C; P(a,9)=1; P(a,10)=C;%ACTON ENTERED

  AA=zeros(n,1); %array for the trajectory prepared;
                                              %contains subsequent actons

%%%%%%%%%%%%%%% DISPLAY SECTION %%%%%%%%%%%%%%%
%CONNECTOR GRAPH DISPLAY
                        gplot(T,GXY);

%NODE NUMBERS DISPLAY;
                        for i=1:31
                        No=int2str(i);
                        text(PX(i), PY(i), No );
                        end
%ACTON ASTERISK DISPLAY
                        text(PX(a),PY(a)+0.01,'*','color', [0 1 0],'FontSize',24);

     %text(PX(b),PY(b)+0.01,'*','color', [0 1 1],'FontSize',24);
                % b is the optional second acton

%PARAMETERS DISPLAY
PAR1=['n= ',num2str(n)]; PAR11=['H= ', num2str(H)];
PAR2=['C= ',num2str(C),'; F= ',num2str(F)];
PAR3=['gup= ',num2str(gup),'; gdown= ',num2str(gdown)];
text (0.01,0.25,PAR1);
text (0.01,0.2,PAR2);
text(0.07, 0.28,' *','color',[0 1 0],'FontSize',24);
text(0.7,0.15,PAR3);text (0.01, 0.15,PAR11);

%%%%%%%%%%%%%%% END OF DISPLAY SECTION
%%%%%%%%%%%%%

%PROTO CONTINUES, PR STARTS; PR adds n cycles.

for jj=1:n

 % START MEMORY
        for ii=1:31
        P(ii,10)=F*P(ii,10);
```

```
        end
        P(a,10)=F*C; P(a,7)=C;%special values for the acton

  % END MEMORY

%START INFLUENCE

                                P(a,7)=H*C; %to exert influence

     P(:,8)=0;
        for I=1:31
                for j=4:6
                        if P(I,1)<P(I,j)
        g=gdown; else g=gup;
                        end
                if I<16 ar=1; end
        if I>16 & I<31 ar=1/3; end
        if I==31 ar=1/2; end

  %P(I,8)=P(I,8)+P(P(I,j),7)*g*ar;
  %previous influence remembered
  P(I,8)=P(P(I,j),7)*g*ar;
        end
                end
%END INFLUENCE
%START PROBABILITY
                P(a,7)=C; %just a formality
        for I=1:31  P(I,7)=P(I,7)+P(I,8)+P(I,10);  end

  %%%%%%%%%%%%%%%
%END PROBABILITY

% START       ACTON SELECTION

        P(a,7)=0; %cannot be selected again
        SUM=0;
        ss=0; S=P(1:31,7); SUM=sum(S); P(1:31,7)=P(1:31,7)/SUM;
        r=rand;
        for i=1:31, ss=ss+P(i,7);
         if ss>r, a=i; break,
         end
 end
 P(:,9)=0;P(a,9)=1; %information about acton; not used
 P(aa,7)=C; %probability for the previous acton;
 % END ACTON SELECTION
```

```
% ACTON ASTERISK DISPLAY; Randomization of the position of ASTERISK.
        %This is done redundantly,with possible other uses in mind
        %COORDINATES
        LXx=P(1:31,2);LX=LXx'/14;
        LYy= P(1:32,3);LY=LYy'/8.5;

        for  I=1:31
        %RANDOM SHIFT OF X
        u=rand/40;
        %RANDOMIZATION OF SIGN
        uu=rand; if uu<=0.5, uuu=1; else uuu=-1; end
        u=u*uuu; LX(1,I)=LX(1,I)+u; end
        for  I=1:31
        %RANDOM SHIFT OF Y
        u=rand/40;
        %RANDOMIZATION OF SIGN
        uu=rand; if uu<=0.5, uuu=1; else uuu=-1; end
        u=u*uuu; LY(1,I)=LY(1,I)+u; end

        text(LX(a), LY (a), '*', 'color',[1 0 0],'FontSize',12);
%END ASTERISK DISPLAY

        % END OF PROTO FOLLOWS
pause (t);
a;AA(jj,1)=a; tra=AA';
end
PP=zeros(31,2);PP(:,1)=P(1:31,1); PP(:,2)=P(1:31,7);
disp ('type PP for probability distribution ');
disp ('type tra for trajectory');
```

proto2.m

```
%PROTO2; TO CONTINUE; file proto2.m; workspace proto.mat
%CONTINUES WITH PREVIOUS WORKSPACE AND
%PROBABILITY DISTRIBUTION; Prompts for n
%%PROTO2 is done redundant,with possible other uses in mind

%Workspace: P(32x11), T(31x31), GXY (31,2).

 %%%%%%%PARAMETERS SECTION %%%%%%%%%%%%%%%%%%
n=input ('Enter n  ');                %n=20;

%a=input('Enter a ');% a=2; %F=input('Enter F ' );%F=0.8;
%C=input('Enter C ');%C=1;  %H=input('Enter H ');%H=0.8;
```

```
%gup=input('Enter gup  ');%gup=0.5;
%w=input('Enter w ');%w=2; %gdown=w*gup;
%t=input('Enter pause t ' );%t=0.1;

tra=[];            % tra: trajectory

%%%%%%%^^^ END of PARAMETERS SECTION
^^^%%%%%%%%%%%%%%

%COLUMNS in matrix P: 1: Node #; 2,3: Node coordinate, 4-6: Node neighborhood;
% 7: Probability,p; 8: Influence,v; 9: Active node,A; 10: Memory,M;
%11: Spare column.  Line 32:zeros.

%INITIAL MATRIX P. Equal probabilities.
        %P(1:31,7)=1/31; P(:,8:11)=0; %Columns 7 to 11 contain variables

  % Data for the acton:
  P(a,7)=C; P(a,9)=1; P(a,10)=C;%ACTON ENTERED

  AA=zeros(n,1); %array for the trajectory prepared;
                                          %contains subsequent actons

%%%%%%%%%%%%%%% DISPLAY SECTION %%%%%%%%%%%%%%%
%CONNECTOR GRAPH DISPLAY
                        %gplot(T,GXY);
%NODE NUMBERS DISPLAY;
                        %for i=1:31
                        %No=int2str(i);
                        %text(PX(i), PY(i), No );
                        %end
%NEW ACTON ASTERISK DISPLAY (THE ACTON LEFT BY PROTO1)
                %text(PX(a),PY(a)+0.01,'*','color', [0 1 0],'FontSize',24);

  rectangle ( 'Position', [0, 0.1, 0.12, 0.25], 'FaceColor', 'w')
  rectangle ( 'Position', [0.7, 0.1, 0.3, 0.1], 'FaceColor', 'w')

  %PARAMETERS DISPLAY
PAR1=['n= ',num2str(n)];
PAR11=['H= ', num2str(H)];
PAR2=['C= ',num2str(C)];
PAR22=['F= ',num2str(F)];
PAR3=['gup=',num2str(gup),'; gdown=',num2str(gdown)];
text (0.01,0.27,PAR1);
text (0.01,0.22,PAR11);
text (0.01,0.17,PAR2);
text (0.01, 0.12,PAR11);
```

```
text(0.72,0.15,PAR3);
%%%%%%%%%%%%%%^^^ END OF DISPLAY SECTION
^^^%%%%%%%%%%

%PROTO CONTINUES, adds n cycles.

for jj=1:n

 % START MEMORY
        for ii=1:31
        P(ii,10)=F*P(ii,10);
        end
        P(a,10)=F*C; P(a,7)=C;%special values for the acton

  % END MEMORY

%START INFLUENCE

                                P(a,7)=H*C; %to exert influence

     P(:,8)=0;
        for I=1:31
                for j=4:6
                        if P(I,1)<P(I,j)
        g=gdown; else g=gup;
                        end
                if I<16 ar=1; end
        if I>16 & I<31 ar=1/3; end
        if I==31 ar=1/2; end

  %P(I,8)=P(I,8)+P(P(I,j),7)*g*ar;
  %previous influence remembered; optional

  P(I,8)=P(P(I,j),7)*g*ar;

        end
                end
%END INFLUENCE
%START PROBABILITY
P(a,7)=C; %just a formality; will be zero anyway

        for I=1:31  P(I,7)=P(I,7)+P(I,8)+P(I,10);  end

  %%%%%%%%%%%%%%
%END PROBABILITY
```

```
% START       ACTON SELECTION
        P(a,7)=0; %cannot be selected again
        SUM=0;
        ss=0; S=P(1:31,7); SUM=sum(S); P(1:31,7)=P(1:31,7)/SUM;
        r=rand;
        for i=1:31, ss=ss+P(i,7);
         if ss>r, a=i; break,
         end
 end
 P(:,9)=0;P(a,9)=1; %information about acton; not used
 P(aa,7)=C; %probability for the previous acton;
 % END ACTON SELECTION

% START ACTON ASTERISK DISPLAY;
%Randomize the position of ASTERISK.

        %COORDINATES
        LXx=P(1:31,2);LX=LXx'/14;
        LYy= P(1:32,3);LY=LYy'/8.5;

        for  I=1:31
        %RANDOM SHIFT OF X
        u=rand/40;
        %RANDOMIZATION OF SIGN
        uu=rand; if uu<=0.5, uuu=1; else uuu=-1; end
        u=u*uuu; LX(1,I)=LX(1,I)+u; end
        for  I=1:31
        %RANDOM SHIFT OF Y
        u=rand/40;
        %RANDOMIZATION OF SIGN
        uu=rand; if uu<=0.5, uuu=1; else uuu=-1; end
        u=u*uuu; LY(1,I)=LY(1,I)+u; end

        text(LX(a), LY (a), '*', 'color',[1 0 0],'FontSize',12);
%END ASTERISK DISPLAY

        % END OF PROTO FOLLOWS
pause (t);
a;AA(jj,1)=a; tra=AA';
end
PP=zeros(31,2);PP(:,1)=P(1:31,1); PP(:,2)=P(1:31,7);
disp ('type PP for probability distribution ');
disp ('type tra for trajectory');
```

==========.

BIRDS

WORKSPACE: **BIRD.mat**
http://spirospero.net/BIRD.mat

To use BIRDS, copy the workspace BIRDS.mat , copy scripts **bf.m** , **bf2.m**, and **bq.m**. Start by entering **bf**. After that, use **bf2**. For bf2, n=1 to 10 is recommended. In this mode, drastic changes of direction can be observed even after one cycle. However, only after a certain number of cycles (50-100), the system enters the collective mode. In terms of physics, it reaches a steady state far from equilibrium. At C=1, coherence is low, but at C=2 it is clearly seen.

bf.m

```
%SCRIPT bf
%Initial parameters.

nf=0; b=0; q=0; gg=0; gin=0; gout=0;gd=0;q=0; w=0;
SUM=0;  s=0; ss=0;  S=0; nf=0; gr=0;

%%%%%%%. Input:
n=5; d=16;
F=0.5; C=1;H=1; gin=1; gr=0.8; gd=1;
%%%%%%%%
gout=gin*gr;
FL(:,:,8)=1/32;
FL(:,33,:)=0;
FL(37,:,:)=0;
FL(:,:,7)=0;
FL(1:36,33,7)=d;
FL(1:36,33,9)= C;
FL(:,:,9:10)=0;
L(1:36);
```

```
%START
for nf=1:n,  for  b=1:36

%MEMORY
%[Forget]
FL(b, :,9)= F* FL(b,:,9);
%[update last direction]
FL(b, FL(b,33,7), 9)=C;
FL(b, FL(b,33,7), 8)=H;

%INFLUENCE, INDIVIDUAL
FL(b,:, 10)=0;
for ii=1:32
   gf =abs(ii-FL(b,33,7));
   if gf>17, gf=32-gf; end
   %[gf=distance between directions ]
          if gf==1
    FL(b,33,10)=FL(b,33,10)+ FL(b,ii,8)*gd;
     end

end;

% INFLUENCE, FLOCK
INF=0;
for w=1:32
          for q=2:9

                              if FLA(b,1)< FLA(b,q)
                    gg=gin; else gg=gout;  end

%[read neighbor's b]
x=FLA(b,q);

%x can be 37

%[read it's d(past){7}]
xx=FL(x,33, 7);
%[read it's P]
PPP=FL(b,w, 8);
%[correct this Infl {10}]
PPP=PPP*gg ;
INF=INF + PPP;
end
FL(b,w,10)= FL(b, w, 10)+PPP;
end
```

```
%PROBABILITY
for ii=1:32
FL(b,ii, 8)=FL(b,ii,8)+FL(b,ii,9)+FL(b,ii,10);
end

FL(b, FL(b, 33,7), 8)=C;

%SELECTION
SUM=0; ss=0;S=FL(b,1:32, 8); SUM=sum(S);
FL(b,:, 8)=FL(b,:,8)/SUM;
r=rand;
 for i=1:32, ss=ss+FL(b,i, 8);
if ss>r, d=i; break,
end
  end

%write new d
          FL(b,33,6)=d;
end
FL(:,33,7)=FL(:,33,6);FL(b,33,6)=0;
A=FL(:,33,7);gplot(TOP,COORD);
AH=['H =   ',num2str(H)];AC=['C =   ',num2str(C)];
AF=['F =   ',num2str(F)];An=['n =  ',num2str(n)];
Ag=['gd =   ',num2str(gd)];Agin=['gin =   ',num2str(gin)];
Agout=['gout=   ',num2str(gout)];
 text (1.5,5.5,AC); text (1.5,4.5,AH);
text (1.5,3.5,AF);text (1.5,2.5,Ag);text (1.5,1.5,Agin);
text (3.4,1.5,Agout); text(3.5,2.5,An);
for u=1:36
   X=COORD(u,1); Y=COORD(u,2);
   AA=A(u);AAA=[' ',num2str(AA)];
   text(X,Y,AAA,'color',[1 0 0],'FontSize',14);

    end
end

%quiver plot: bq.m
%for ll=1:36, L(ll)=2*pi*A(ll)/32;end
%sL=sin(L);cL=cos(L);sL(37)=[];cL(37)=[];
%quiver(COORD(:,1),COORD(:,2),sL',cL')


%pause(2);
bq;     %if blocked, the grid output will be displayed
```

bq.m

```
for ll=1:36, L(ll)=2*pi*A(ll)/32;end
sL=sin(L);cL=cos(L);sL(37)=[];cL(37)=[];
quiver(COORD(:,1),COORD(:,2),sL',cL')
```

bf2.m

```
%%%%%%%. Input:
%n=3; d=16; F=0.3; C=2;H=1; gin=1; gr=0.8; gd=1;
%%%%%%%

%START
gout=gin*gr;
FL(:, : , 8)=1/32;
FL(:,33,:)=0;
FL(37,:,:)=0;
FL(37,:,:)=0;
FL(1:36,33,7)=d;
FL(1:36,33,9)= C;
FL(:,:,9:10)=0;
FL(1:36,33,7)=d;
L(1:36);

for nf=1:n,  for  b=1:36

%MEMORY
%[Forget]
FL(b, :,9)= F* FL(b,:,9);
%[update last direction]
FL(b, FL(b,33,7), 9)=C;
FL(b, FL(b,33,7), 8)=H;

%INFLUENCE, INDIVIDUAL
FL(b,:, 10)=0;
for ii=1:32
   gf =abs(ii-FL(b,33,7));
   if gf>17, gf=32-gf; end
   %[gf=distance ]
          if gf==1
    FL(b,33,10)=FL(b,33,10)+ FL(b,ii,8)*gd;
     end

end;
```

```
% INFLUENCE, FLOCK
INF=0;
for w=1:32
        for q=2:9

                        if FLA(b,1)< FLA(b,q)
                gg=gin; else gg=gout;  end

%[read neighbor's b]
x=FLA(b,q);

%x can be 37

%[read it's d(past){7}]
xx=FL(x,33, 7);
%[read it's P]
PPP=FL(b,w, 8);
%[correct this Infl {10}]
PPP=PPP*gg ;
INF=INF + PPP;
end
FL(b,w,10)= FL(b,w,10)+PPP;
end

%PROBABILITY
for ii=1:32
FL(b,ii, 8)=FL(b,ii,8)+FL(b,ii,9)+FL(b,ii,10);
end

FL(b, FL(b, 33,7), 8)=C;

%SELECTION
SUM=0; ss=0;S=FL(b,1:32, 8); SUM=sum(S);
FL(b,:, 8)=FL(b,:,8)/SUM;
r=rand;
 for i=1:32, ss=ss+FL(b,i, 8);
if ss>r, d=i; break,
end
  end

%write new d
        FL(b,33,6)=d;
end
FL(:,33,7)=FL(:,33,6);FL(b,33,6)=0;
A=FL(:,33,7);gplot(TOP,COORD);
```

```
for u=1:36
  X=COORD(u,1); Y=COORD(u,2);
  AA=A(u);AAA=[' ',num2str(AA)];
  text(X,Y,AAA,'color',[1 0 0],'FontSize',14);
end
%DISPLAY PARAMETERS
AH=['H =  ',num2str(H)];AC=['C =  ',num2str(C)];
AF=['F =  ',num2str(F)];An=['n =  ',num2str(n)];
Ag=['gd =  ',num2str(gd)];Agin=['gin =  ',num2str(gin)];
Agout=['gout=  ',num2str(gout)];
 text (1.5,5.5,AC); text (1.5,4.5,AH);
text (1.5,3.5,AF);text (1.5,2.5,Ag);text (1.5,1.5,Agin);
text (3.4,1.5,Agout); text(3.5,2.5,An);

end
%QUIVER DISPLAY :  bq.m

%for ll=1:36, L(ll)=2*pi*A(ll)/32;end
%sL=sin(L);cL=cos(L);sL(37)=[];cL(37)=[];
%quiver(COORD(:,1),COORD(:,2),sL',cL')

%pause(2);
bq;         %if blocked, the grid output will be displayed
```

==========.

SCALE

sc.m (main)

```
% PROGRAM  SCALE, script sc; RELATED: link; space lin.mat
% 1. INITIALIZE

S=input('1 to start, 2 to continue  ');
nn=input('enter number of cycles (nn):  ');
disp( '  ATTENTION: Interspaced components should be entered')
disp( '  between apostrophes as single string of characters');
disp( '  ');
disp( '             press any key to continue') ; pause;

% 2. START
for n=1:nn, if  (S==1)&(n==1), sw=1; W=zeros(sw, 16,8);
                          NAMES=['!']; WW=zeros(sw,sw);NM=[ ];
```

```
                    end
% 3.    INPUT
I=input('enter components  ');

% 4.    CONVERT  INPUT into PIN;
PIN=zeros(8,8); word=[];jj=1; I=[I,' ']; ab=abs(I); lab=length(ab);
for i=1:lab,  if (ab(i))~=32, word=[word,ab(i)]; else
lw=length(word); PIN(jj,1:lw)=word;  word=[]; jj=jj+1; end, end
% 5.    CHECKING novelty of PIN against WORLD
 old=0;  wiold=0; comp=[]; % PIN presumed new

 for wi=1:sw %wi: length of World
% 5.1   CUTTING FLAT SLICE OF WORLD W(wi,:,:)
fW=zeros(16,8); fW(:,:)=W(wi,:,:);

%       5.2     COMPARE flat W and PIN as ordered sets %
eqid=isequal(fW(9:16,:),PIN);          if eqid==1, old=1; %PIN is  OLD
disp ('This is old  '),disp(NAMES((wi),:)), %  wiold=wi; list=find(WW(:,wiold));
leli=length(list); SN=[];for k=1:leli, SN=strvcat(SN,NAMES((leli),:));end
end,

%       5.3     COMPARE flat W and PIN as sets
set=0;

if isempty(setxor(fW((1:8),:),PIN,'rows')),if old==0,
if eqid~=1,
note = ['Looks like W' ,int2str(wi),': ', NAMES(wi,:)];
disp(note); set=1;   end,          end,old=1; end
%%%%%%
if set==1,
nnew=input(' Is it new?  1/0  ');
if nnew==1,old=0; end, end
%%%%%
%       5.4.  CHECK for SET MEMBERS to create SPECTRUM;
eqcomp = intersect(PIN,fW ,'rows');
h=size(eqcomp,1) ; test=zeros(1,8);
for hh=1:h,
if ((~isempty(eqcomp))&(~isequal(eqcomp(hh,:),test))),
comp=[comp,wi];end, end
end %for wi=1:sw
  spectrum=unique(comp);les=length(spectrum);
  %disp('spectrum: ');   %for r=1:les, disp(NAMES(spectrum(r),:));  end
  %if old==1,SNU=unique(SN,'rows');disp('OLD spectrum '),disp(SNU),end,
% 6. IF NEW, EXPAND THE WORLD:
%       6.1. NAME THE NEW ENTRY
```

```
if old==0, NAME=input('This is new. Name it   ') ;
   NAMES=strvcat(NAMES, NAME);
% 6.2. CREATE NewPIN containing NAME
NAME=[NAME,' ']; NewPIN=zeros(8,8);word=[];jj=1;  ab=abs(NAME);
 lab=length(ab); for i=1:lab,  if (ab(i))~=32, word=[word,ab(i)];else
    lw=length(word); NewPIN(jj,1:lw)=word;  word=[];jj=jj+1; end,
                    end

% 6.3.ADD PIN and NewPIN to WORLD

sw=sw+1; WW(sw,:)=0; WW(:,sw)=0;% PLACE IN THE WORLD

W(sw,9:16,:)=NewPIN(:,:);    W(sw,1:8,:)=PIN(:,:);
                    % All 16 lines in WORLD are filled up

     % 6.4. RECORD NEW LINKS
for ll=1:les, if ~isempty(spectrum(ll)),
WW(sw,spectrum(ll))=1; WW(spectrum(ll),sw)=1;end, end,

end % if old==0,

%To read an entry from W(x,y,:)  %D=nonzeros(W(x,y,:)); D=D'; setstr(D);

end % for n=1:nn
NM=NAMES; world=WW;
disp('If you want to check WORLD for a name, type: link ')
disp('To see NAMES, type: NM  ');
disp('To display the 3D world, type plotW ');
```

sct.m

```
% PROGRAM  SCALE, script sc; RELATED: link; space lin.mat
% 1. INITIALIZE
S=input('1 to start, 2 to continue   ');nn=input('nn  ');

% 2. START
for n=1:nn, if  (S==1)&(n==1), sw=1; W=zeros(sw, 16,8);
                            NAMES=['!']; WW=zeros(sw,sw);
                       end
% 3.     INPUT
I=input('type interspaced components as single character string  ');

% 4.     CONVERT  INPUT into PIN;
PIN=zeros(8,8); word=[];jj=1; I=[I,' ']; ab=abs(I); lab=length(ab);
```

```
for i=1:lab,  if (ab(i))~=32, word=[word,ab(i)]; else
lw=length(word); PIN(jj,1:lw)=word;  word=[]; jj=jj+1; end, end
% 5.    CHECKING novelty of PIN against WORLD
 old=0;  wiold=0; comp=[]; % PIN presumed new

 for wi=1:sw %wi: length of World
% 5.1  CUTTING FLAT SLICE OF WORLD W(wi,:,:)
fW=zeros(16,8); fW(:,:)=W(wi,:,:);

%       5.2      COMPARE flat W and PIN as ordered sets %
eqid=isequal(fW(9:16,:),PIN);          if eqid==1, old=1; %PIN is  OLD
disp ('This is old  '),disp(NAMES((wi),:)), %  wiold=wi; list=find(WW(:,wiold));
leli=length(list); SN=[];for k=1:leli, SN=strvcat(SN,NAMES((leli),:));end
end,

%       5.3      COMPARE flat W and PIN as sets
set=0;

if isempty(setxor(fW((1:8),:),PIN,'rows')),if old==0,
if eqid~=1,
note = ['Looks like W' ,int2str(wi),': ', NAMES(wi,:)];
disp(note); set=1;   end,          end,old=1; end
%%%%%%
if set==1,
nnew=input(' Is it new?  1/0  ');
if nnew==1,old=0; end, end
%%%%%
%       5.4.  CHECK for SET MEMBERS to create SPECTRUM;
eqcomp = intersect(PIN,fW ,'rows');
h=size(eqcomp,1) ; test=zeros(1,8);
for hh=1:h,
if ((~isempty(eqcomp))&(~isequal(eqcomp(hh,:),test))),
comp=[comp,wi];end, end
end %for wi=1:sw
   spectrum=unique(comp);les=length(spectrum);
   %disp('spectrum: ');   %for r=1:les, disp(NAMES(spectrum(r),:));  end
   %if old==1,SNU=unique(SN,'rows');disp('OLD spectrum '),disp(SNU),end,
% 6. IF NEW, EXPAND THE WORLD:
%       6.1. NAME THE NEW ENTRY

if old==0, NAME=input('This is new. Name it   ') ;
   NAMES=strvcat(NAMES, NAME);
% 6.2. CREATE NewPIN containing NAME
NAME=[NAME,' ']; NewPIN=zeros(8,8);word=[];jj=1;  ab=abs(NAME);
 lab=length(ab); for i=1:lab,  if (ab(i))~=32, word=[word,ab(i)];else
    lw=length(word); NewPIN(jj,1:lw)=word;  word=[];jj=jj+1; end,
```

```
                end

% 6.3.ADD PIN and NewPIN to WORLD

sw=sw+1; WW(sw,:)=0; WW(:,sw)=0;% PLACE IN THE WORLD

W(sw,9:16,:)=NewPIN(:,:);    W(sw,1:8,:)=PIN(:,:);
                % All 16 lines in WORLD are filled up

     % 6.4. RECORD NEW LINKS
for ll=1:les, if ~isempty(spectrum(ll)),
WW(sw,spectrum(ll))=1; WW(spectrum(ll),sw)=1;end, end,

end % if old==0,

%To read an entry from W(x,y,:)  %D=nonzeros(W(x,y,:)); D=D'; setstr(D);

end % for n=1:nn
NAMES; world=WW;
disp('If you want to check WORLD for a name, type: link ')
disp('To see NAMES, type: NM  ');
disp('To display the 3D world, type plotW ');
```

flatW.m

```
%script flatW
cla
om=length(WW);
PX=zeros(om,1);PY=zeros(om,1);XYP=zeros(om,2);
angles=(2*pi./100).*[0:100];
plot(cos(angles),sin(angles))
hold on
angles=(2*pi/om).*[0:om-1];
for j=1:om
  %plot(cos(angles(j)).*[1 1.5],sin(angles(j)).*[1 1.5])
  text(cos(angles(j)),sin(angles(j)),[num2str(j),' ',NAMES(j,:)]),
PX(j)=cos(angles(j));PY(j)=sin(angles(j)); end
XYP(:,1)=PX; XYP(:,2)=PY; gplot(WW,XYP);
%end of script flatW
```

link.m

```
% CHECKING THE WORLD FOR NAME; SCRIPT: link
 I=input('NAME to check ');
 CI=cellstr(I);
 SC=nonzeros(strmatch(CI,NAMES));%find NAME's numbers
 LINKS=find(WW(:,SC)); links=LINKS';
 LL= [];%list of links
 NAMELIST=[];
 for i=1:length(SC)
   LL= [LL,(find(WW(:,SC(i))))'];
 end

 lel=length(LL); for r=1:lel, NAM=(NAMES(LL(r),:));
 NAMELIST=strvcat(NAMELIST,NAM);end,
 NAMELIST=unique(NAMELIST,'rows'); disp(NAMELIST)
```

plotW.m

```
%script: plotW, plot world
close;
om=length(WW);
PX=zeros(om,1);PY=zeros(om,1); XYP=zeros(om,3);  PZ=[1:om]; PZ=PZ'/om;
angles=(2*pi/om).*[0:om-1];
for  j=1:om
PX(j)=cos(angles(j));PY(j)=sin(angles(j));
end
XYP(:,1)=PX; XYP(:,2)=PY; XYP(:,3)=PZ;

[iW,jW]=find(WW);
liW=length(iW);
MW=zeros(liW,2);
MW(:,1)=iW; MW(:,2)=jW;
for i=1:liW
line([XYP(MW(i,1),1),XYP(MW(i,2),1)],...
    [XYP(MW(i,1),2),XYP(MW(i,2),2)],...
    [XYP(MW(i,1),3),XYP(MW(i,2),3)]);
end
hold on,
stack
```

stack.m

```
%script: stack
ZZ=zeros(1,41);
tt=0;
```

```
for zz=0:0.1:1
t = 0:pi/20:(2*pi); ZZ(1,:)=zz; tt=ZZ;
    plot3(sin(t),cos(t),tt),hold on,
    grid on
    axis square
tt=tt+1;
end
```

==========.

email: simp@spirospero.net