Paper 242-2008

How to Get Phase Portraits Animated with SAS®

Kristan A. Schneider, University of Vienna, Austria
Barbara G. Schneider, Medical University of Vienna, Austria

Abstract

In this video-poster we exemplify how to create animated phase portraits of systems of difference or differential equations, by using the SAS® GIF Animator. More precisely, we consider a system of difference or differential equations and want to illustrate how a change in one or more parameters changes the phase portrait. This is of great value e.g. when studying bifurcations. We exemplify this by solving a difference equation that has applications in biomathematics and game theory. First, we implement and solve the system of difference or equations for different initial conditions and different values for one of the model parameters using SAS/IML. Next, we create phase portraits from this data by using SAS/GRAPH for each parameter value. Then we use the GIF Animator to concatenate the plots in order to obtain an animated GIF file that shows how the phase portrait changes if one varies the regarded parameter. Finally, we give some hints how to improve the plots using various goptions and the Annotate facility, and how to improve your SAS/IML code.

Model description

We consider the following system of difference equations known as the pairwise interaction model. This model has various applications in biomathematics and game theory.

$$p_i(t+1) = p_i(t)\frac{W_i(t)}{\overline{W}(t)} \quad \text{for} \quad i = 1,2,3, \tag{1}$$

where

$$W_i(t) = \sum_{j,k,l=1}^{3} \alpha_{ij,kl} p_j(t) p_k(t) p_l(t) \tag{2}$$

and

$$\overline{W}(t) = \sum_{i=1}^{3} W_i(t) p_i(t). \tag{3}$$

We assume that $\alpha_{ij,kl} = \alpha_{ji,kl}$, $\alpha_{ij,kl} = \alpha_{ij,lk}$, $\alpha_{ij,kl} = \alpha_{ji,lk}$ and $\alpha_{ij,kl} > 0$ for all $i,j,k,l = 1,2,3$. Hence, we can represent the coefficients $\alpha_{ij,kl}$ by the matrix:

$$A = \begin{pmatrix} \alpha_{11,11} & \alpha_{11,12} & \alpha_{11,13} & \alpha_{11,22} & \alpha_{11,23} & \alpha_{11,33} \\ \alpha_{12,11} & \alpha_{12,12} & \alpha_{12,13} & \alpha_{12,22} & \alpha_{12,23} & \alpha_{12,33} \\ \alpha_{13,11} & \alpha_{13,12} & \alpha_{13,13} & \alpha_{13,22} & \alpha_{13,23} & \alpha_{13,33} \\ \alpha_{22,11} & \alpha_{22,12} & \alpha_{22,13} & \alpha_{22,22} & \alpha_{22,23} & \alpha_{22,33} \\ \alpha_{23,11} & \alpha_{23,12} & \alpha_{23,13} & \alpha_{23,22} & \alpha_{23,23} & \alpha_{23,33} \\ \alpha_{33,11} & \alpha_{33,12} & \alpha_{33,13} & \alpha_{33,22} & \alpha_{33,23} & \alpha_{33,33} \end{pmatrix}.$$

Moreover, the system (1) is invariant on the simplex $\mathcal{S}^2 = \{(x,y,z) \in \mathbb{R}^3 \,|\, x,\, y,\, z \geq 0 \wedge x+y+z = 1\}$, i.e., if $(p_1(t), p_2(t), p_3(t)) \in \mathcal{S}^2$ then $(p_1(t+1), p_2(t+1), p_3(t+1)) \in \mathcal{S}^2$.

The system can be interpreted as follows. Consider a sufficiently large randomly mating diploid population. Furthermore, assume discrete, non-overlapping generations. Assume that 3 alleles (version of a gene) A_1, A_2, A_3 occur at a single autosomal locus (specific position in the genome). Hence, the genotypes A_iA_j $i, j = 1, 2, 3$ occur in the population. The fitness (probability to survive until the reproductive age) of an individual (genotype) is the average amount of interaction experienced by this individual. The interaction coefficient $\alpha_{ij,kl}$ expresses how much an individual of genotype A_iA_j is affected by interaction with an individual of genotype A_kA_l. Moreover, let p_1, p_2, and p_3 denote the frequencies of the alleles A_1, A_2 and A_3, respectively. The system (1) of difference equations describes the evolutionary dynamics of the frequencies of the alleles A_1, A_2 and A_3, i.e., the changes in allele frequencies in the population from one generation to the next. Moreover, W_i is the so called "marginal fitness" of allele A_i and $\overline{W}$ is the mean fitness of the population.

Implementation in SAS/IML

Now, we implement and solve the system of differential equations using SAS/IML, i.e., for a given coefficient matrix A and given starting values $p_1(0)$, $p_2(0)$, $p_3(0)$, we want to iterate the system of difference equations until an equilibrium (fixed point) is numerically reached or a maximum number of iteration steps is reached.

First, we create an input data set that contains one or more coefficient matrices. Then, we create a data set that contains the starting values for each coefficient matrix.
In our example, we chose the coefficient matrices

$$A_b = \begin{pmatrix} 0.7 & 1 & 1 & 1 & 1 & 1 \\ b & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & b \\ 1 & 1 & 1 & 0.7 & 1 & 1 \\ 1 & 1 & 1 & b & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0.7 \end{pmatrix},$$

where $b = 1,\ 1.5,\ 2, 3,\ 4,\ 5,\ 10,\ 30$. The matrices in the input data set are vertically concatenated, i.e., in our example the input data set contains 48 rows and 6 columns, it is named 'incoeff' and stored in the folder 'gf2008'.

Furthermore, we create a data set that contains the starting values for the respective coefficient matrices. We use the same four sets of initial values for each coefficient matrix, namely,

	$p_1(0)$	$p_2(0)$	$p_3(0)$
1	0.3333	0.3333	0.3334
2	0.6	0.39	0.01
3	0.01	0.6	0.39
4	0.39	0.01	0.6

.

We store the initial values in the data set 'freq1' in the folder 'gf2008'. The set 'freq1' contains 32 rows and 3 columns.

Next, we write a SAS/IML program that iterates the system of difference equations for the coefficient matrices and the respective starting values. The SAS/IML program stores all iteration steps in an output file. This is the code:

```
%let n=3;
%let maxit=500;
%let eps=1e-12;
%let icoeff=%str(gf2008.coeff2);
%let freq=%str(gf2008.freq1);
%let outdata=%str(out2);
data &outdata;
array x p3 p2 p1 gen freqnum coeffnum;
delete;
run;
```

```
proc iml;
n=&n;
maxit=&maxit;
eps=(&eps)##2;
mat=J(n,n,.);
do l=1 to n;
  mat[l,1:l]=J(1,l,1);
end;
use &icoeff;
read all into alpha;
nn=ncol(alpha);
na=nrow(alpha)/nn;
use &freq;
read all into infreq;
np=nrow(infreq)/na;
out=J(maxit,n+1,.);
do u=1 to na;
  aij=alpha[(u-1)*nn+1:u*nn,];
  Do  v=1 to np;
    p=t(infreq[(u-1)*np+v,]);
    gennum=maxit;
    Do k=1 to maxit;
      pp=mat#p;
      pmat1=t(p)# pp ;
      pvec1=symsqr(pmat1);
      pmat=2#sqrsym(symsqr(pmat1))-diag(pmat1);
      pvec=symsqr(pmat);
      Wij=pvec1#(aij*pvec);
      Wij=sqrsym(Wij);
      Wi=Wij[,+];
      W=sum(Wi);
      pnew=Wi/W;
      out[k,]=t(pnew)||k;
      if ssq(p-pnew)<eps then do;
        gennum=k;
        k=maxit;
      end;
      p=pnew;
    end;
    out1=J(gennum,n+3,.);
    out1[,1:n+1]=out[1:gennum,];
    out1[,n+2]=J(gennum,1,v);
    out1[,n+3]=J(gennum,1,u);
    edit &outdata;
    append from out1;
  end;
end;
quit;
```

Creating phase portraits using SAS/Graph

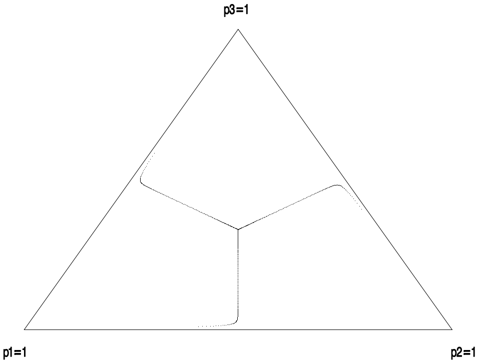
Next, we use the output data set to create the respective phase portraits. First, note that the simplex $\mathcal{S}^2$ corresponds to an equilateral-triangle plane in the three-dimensional space, and can thus be represented as an equilateral triangle in the real plane $\mathbb{R}^2$. Hence, we use a coordinate transformation in order to represent the phase portraits in an equilateral triangle. Furthermore, in order to show the edges of the triangle representing the simplex, we have to add some additional data to the data set that is ultimately plotted. This is the code:

```
%let outdata1=%str(out2trans);

data &outdata1;
array x p3 p2 p1 gen freqnum coeffnum p1trans p2trans ;
delete;
run;

proc iml;
use &outdata;
read all into A;
nnew=nrow(A);
uu=(1/Sqrt(2))#(1||Sqrt(3));
m={1 -1, -1 -1};
out2=J(nnew,2,.);
do l= 1 to nnew;
  out2[l,1:2]=uu#({1 1}+A[l,2:3]*m);
end;
B=A||out2;
edit &outdata1;
append from B;
quit;


data outtemp;
set &outdata1;
keep freqnum p1trans p2trans coeffnum;
label coeffnum='b';
run;

data outtemp1;
array x freqnum coeffnum p1trans p2trans;
delete;
label coeffnum='b';
run;
```

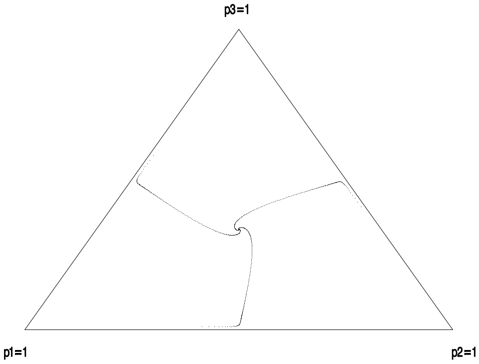
```
proc iml;
use outtemp;
read all into aa;
n=nrow(aa);
b={0 0}//(sqrt(2)||0)//((1||sqrt(3))/sqrt(2))//{0 0};
cc=J(4,4,1);
cc[1:4,3:4]=b;
cc[1:4,1]=J(4,1,5);
do k=1 to 8;
 cc[1:4,2]=J(4,1,k);
  edit outtemp1;
  append from cc;
end;
  edit outtemp1;
  append from aa;
quit;

proc sort data=outtemp1;
 by coeffnum freqnum;
run;
```

For each coefficient matrix A we want to create a phase portrait by plotting the trajectories starting at the respective initial values.

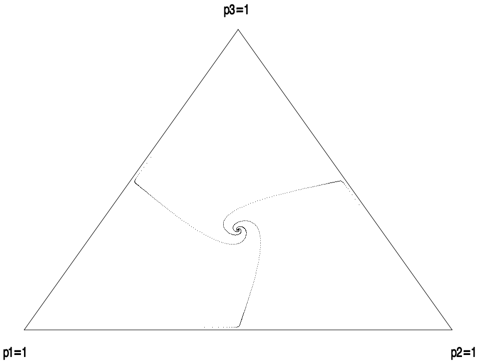
For different values of b we obtain the following phase portraits:



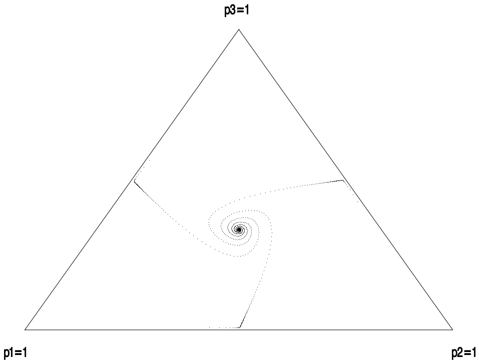
$b = 1$



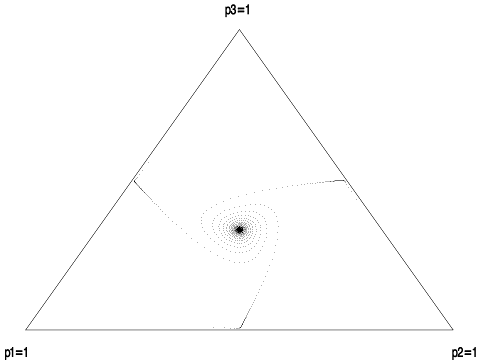
$b = 1.5$



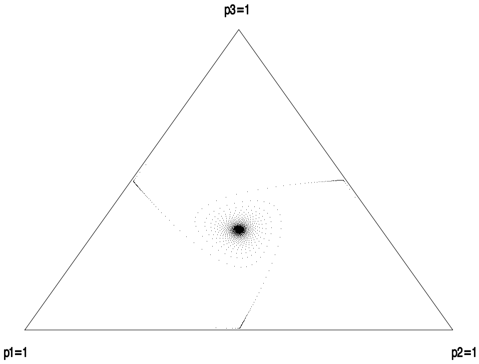
$b = 2$



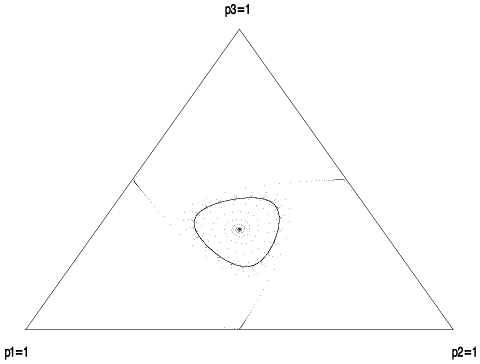
$b = 3$



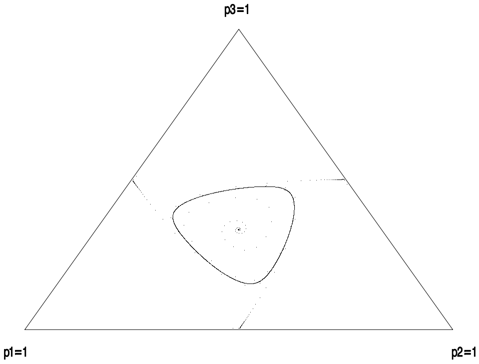
$b = 4$



$b = 5$



$b = 10$



$b = 30$

Using the GIF Animator

We now link the various phase portraits together to create an animated GIF file.

The gifanim driver

The GIFANIM driver provides a mechanism for combining GIF images created with SAS/GRAPH procedures that allow you to create GIF animations. The behavior of the driver is controlled by graphics options that enable you to set such things as delay time, iteration count, transparency, and disposal methods. Before using the GIFANIM device driver, you should become familiar with the animation process controls. The process involved with creating an animated GIF file with the GIFANIM driver requires that you, the animator, take control of the job sequence and ensure that the resulting data stream is constructed properly. The GIFANIM data stream has three parts: Header, Body, and Trailer. Each portion of the data stream is equally important and must be present. Otherwise, an incomplete or unreadable animation sequence will result.

Preparing the header

When creating a new animated GIF data stream, you must issue GOPTIONS GSFMODE=REPLACE prior to the invocation of the first SAS/GRAPH procedure in the SAS job. The driver will then construct a new data stream by writing a valid GIF header and graphic data from the first procedure (see below for the code for our example).

Preparing the body

After the first SAS/GRAPH procedure has been executed, you must construct the body of the GIF animation. You can think of the Body as all the graphic images between the first and last image. Set GOPTIONS GSFMODE=APPEND to signal the GIFANIM driver to suppress the header information and to begin appending graphic data to the current data stream. The GOPTIONS GSFMODE=APPEND statement must appear somewhere between the first and second SAS/GRAPH procedures.

Note: If you use BY-group processing on the first graphics procedure to generate multiple graphs (as in our example), they are automatically appended to the same GIF file. Thus, you do not need GSFMODE=APPEND for that first procedure. If you do not use a second graphics procedure to append additional graphs to the GIF file, you do not need this Body section in your program.

Preparing the trailer

The final step is to mark the end of the animation by appending a GIF trailer ('3B'x) to the data stream. The way to do this depends on whether the last procedure uses BY-group processing.

- Without BY-group processing, set GOPTIONS GEPILOG='3B'X before the last SAS/GRAPH procedure.

- With BY-group processing (as in our example), do not assign a value to GEPILOG; otherwise your GIF animation sequence will be incomplete. Because a GEPILOG is written after each graph in a BY-group, the GIF decoder will interpret the first '3B'x as the end of the animation. Instead, you should use a DATA step to add the trailer to the data stream.

Here is the code for our example:

```
goptions RESET=all ;

/* assign the destination for the output file*/
filename out 'C:\SAS_GlobalForum_2008\anim08_4.gif';
/*set the graphics environment and assign the appropriate
graphics options for the animation */
goptions ftext=swiss  HSIZE=17 CM VSIZE=12 CM
device=gifanim gsfname=out gsfmode=replace iteration=0 delay=80;
symbol1 color=black value=point height=3;
symbol2 color=black value=point height=1;
symbol3 color=black value=point height=1;
symbol4 color=black value=point height=1;
symbol5 color=black interpol=join value=point height=1;
axis1 major=none minor=none color=white label=( angle=0 rotate=0 h=1)
width=1  value=( angle=0 rotate=0 h=0.01);
axis2  label=(h=1)  color=white  minor=none  major=none value=(
angle=0 rotate=0 h=1);
options nobyline;

proc gplot data=outtemp1 uniform; by coeffnum;
plot p2trans*p1trans =freqnum/ nolegend vaxis=axis1 haxis=axis2 anno=anno;
run;
quit;
/* end the animation */
data _null_;
   file out recfm=n mod;
   put '3B'x;
run;

/*disassociates one or more currently assigned filerefs*/
filename out clear;
```

The careful reader might have noticed that we use the an annotate data set in the gplot statement. The data set 'anno' is used to label the vertices of the equilateral-triangle representing the two-dimensional simplex, $\mathcal{S}^2$. The data set is created as follows:

```
data anno;
coeffnum=1;
size=1;
position='0';
x=4 ; y=3 ; text='p1=1'; WHEN='A'; output;
x=73; y=3; text='p2=1'; WHEN='A'; output;
x=38.5; y=37.5; text='p3=1'; WHEN='A'; output;
coeffnum=2;
size=1;
position='0';
x=4; y=3; text='p1=1'; WHEN='A'; output;
x=73; y=3; text='p2=1'; WHEN='A'; output;
x=38.5; y=37.5; text='p3=1'; WHEN='A'; output;
coeffnum=3;
size=1;
position='0';
x=4; y=3; text='p1=1'; WHEN='A'; output;
x=73; y=3; text='p2=1'; WHEN='A'; output;
x=38.5; y=37.5; text='p3=1'; WHEN='A'; output;
coeffnum=4;
size=1;
position='0';
x=4; y=3; text='p1=1'; WHEN='A'; output;
x=73; y=3; text='p2=1'; WHEN='A'; output;
x=38.5; y=37.5; text='p3=1'; WHEN='A'; output;
coeffnum=5;
size=1;
position='0';
x=4; y=3; text='p1=1'; WHEN='A'; output;
x=73; y=3; text='p2=1'; WHEN='A'; output;
x=38.5; y=37.5; text='p3=1'; WHEN='A'; output;
coeffnum=6;
size=1;
position='0';
x=4; y=3; text='p1=1'; WHEN='A'; output;
x=73; y=3; text='p2=1'; WHEN='A'; output;
x=38.5; y=37.5; text='p3=1'; WHEN='A'; output;
coeffnum=7;
size=1;
position='0';
x=4; y=3; text='p1=1'; WHEN='A'; output;
x=73; y=3; text='p2=1'; WHEN='A'; output;
x=38.5; y=37.5; text='p3=1'; WHEN='A'; output;
coeffnum=8;
size=1;
position='0';
x=4; y=3; text='p1=1'; WHEN='A'; output;
x=73; y=3; text='p2=1'; WHEN='A'; output;
x=38.5; y=37.5; text='p3=1'; WHEN='A'; output;
run;
```

Conclusions

The GIFANIM device driver is an excellent tool to create animated graphics, which can be used in presentations. In our example we use the animated GIFANIM device driver to study how the phase portrait of a system of differential equations changes if we change a parameter. It is easily seen that a bifurcation occurs. More precisely, first a globally stable fixed point exists, that is reached by all trajectories. As the parameter b increases the fixed point becomes unstable and a globally stable periodic attractor emerges. Animated graphics are predestinated to illustrate such a situation. However, animated graphics are not limited to the study of bifurcations, the applications of animated graphics are as manifold as your ideas.

References

SAS/IML User's Guide, `http://support.sas.com/documentation/onlinedoc/sas9doc.html`

SAS/GRAPH 9.1 Reference, `http://support.sas.com/documentation/onlinedoc/sas9doc.html`

Cockerham, C. C., Burrows, P. M., Young, S. S. & Prout, T. 1972. Frequency-dependent selection in randomly mating populations. Amer. Naturalist 106, 439-515.

Contact Information

Dr. Kristan Schneider
Department of Mathematics, University of Vienna
Nordbergstr. 15, UZA 4
A-1090 Vienna, Austria
Phone: +4314277 507 74
E-mail: kristan.schneider@univie.ac.at