Paper 260-28

How to Use the SAS® System as a Powerful Tool in Biomathematics

Kristan A. Schneider, University of Vienna, Austria
Georg T. Schneider, Institute of Mathematics, University of Vienna, Austria

Abstract

This work is devoted to show how the SAS® system can be used in applied biomathematics. Our intention is to demonstrate with the aid of two biologically important examples how SAS® Software products can be used as a powerful tool in biological and mathematical research. We will implement the discrete logistic equation on SAS/IML® and use SAS/GRAPH® to visualize the data. The second example is the selection-mutation equation. In the first step we use SAS/IML® along with Base SAS® Software to create data, reflecting a biological or genetic background. Afterwards we use SAS/GRAPH® to display data, and Base SAS® Software as well as SAS/STAT® to proceed statistical evaluation. Efficient implementation of difference equations, expressing genetic models on SAS/IML®, and data management on as well as SAS/STAT® are discussed. Further, we show how SAS® Macro Language can be used to improve the efficiency of the simulations and to connect the various SAS® tools. We also make suggestions how to improve the graphics using the Annotate facility and how to export graphics into external graphic files. The skill level of the audience is assumed to be advanced.

1 Introduction

In biomathematics mathematical models reflecting a biological background are studied. Biomathematics is focused to the investigation of ecological or genetical systems. In biomathematics biological situations are described as mathematical models. Usually these models are expressed through difference equation, differential equations or stochastic processes. Since even for simple models a global analysis of the dynamical behavior of such models including all biological relevant factors seems to be impossible, usually an approach using computer simulations is made, in order to get a better understanding of the biological background represented by the model. In this paper we show how SAS® software can be used for computer simulations and exemplify this for a basic population growth model and a selection-mutation model, which can be expressed by difference equations. The main tool for the simulations is SAS/IML®.

2 A population growth model

In this section we want to show how to use the SAS® system in biomathematics. We chose the discrete logistic equation to exemplify how to perform simulation reflecting a biological background and how to exhibit the created data in terms of useful graphics (sometimes a picture says more than a thousand words).

To obtain a biologically useful model, we define

$$N' = \begin{cases} N\left[1+\rho\left(1-\frac{N}{K}\right)\right] & if N \le K\frac{\rho+1}{\rho} \\ 0 & otherwise \end{cases} . \quad (1)$$

This type of equation has been subject to intensive mathematical research because of its complex dynamical properties. To simplify (1) we make the following transformation: $x = \frac{\rho}{1+\rho}\frac{N}{K}$ $(0 \le x \le 1 \Leftrightarrow N \le K\frac{\rho+1}{\rho})$. Therefore,

$$x' = Rx(1-x) \text{ , where } R = \rho + 1 \quad (2)$$

If $0 \le R \le 4$, it follows that $0 \le x' \le 1$ provided $0 \le x \le 1$.

Here is our SAS® code:

```
%let start=0.1;
%let num=1000;
%let rmin=1;
%let rmax=4;
%let stepwidth=0.1;
%let data=%str(temp);
data &data; array xx x gen R;
delete;
run;
proc iml;
a=J(&num,3,.);
Do R=& rmin to & rmax by & stepwidth;
x=&start;
do i=1 to &num;
a[1,1]=x;
x=R*x*(1-x);
a[i,1]=x;
```

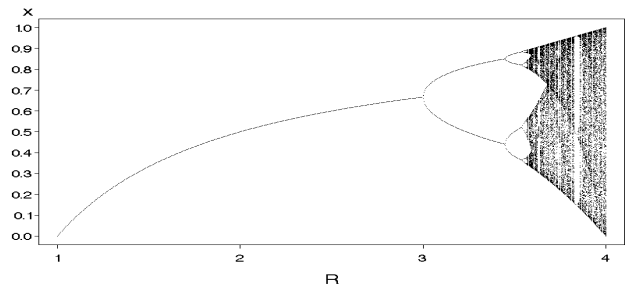

Figure 1: *Bifurcation structure of the logistic equation. For $R \leq 3$ convergence toward an equilibrium occurs and there exist no limiting cycles. For $R > 3$ a cascade of pitchfork bifurcations occurs and convergence toward limiting cycles can be observed.*

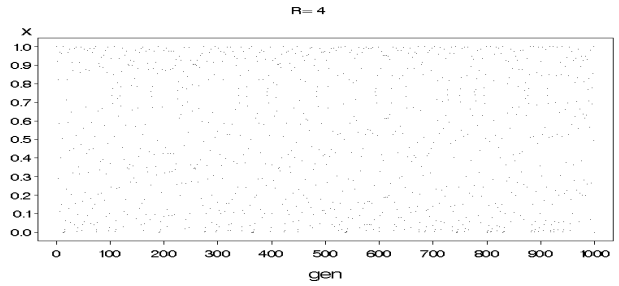

Figure 2: *The figure shows chaotic behavior.*

```
end;
a[1:&num,2]=t((1:&num));
a[1:&num,3]=J(&num,1,R);
edit &data;
append from a;
end;
quit;
goptions ftext=SWISS ctext=BLACK
htext=1 cells ;
symbol1 c=black ci=black v=point height=1
cells interpol=NONE l=1 w=1;
proc gplot data=&data;
plot x*gen=1;
by R;
run;
quit;
```

The dynamics of the equation (2) depend much on the parameter R. For $0 \leq R \leq 3$ an equilibrium is ultimately reached. For $3 < R \leq 4$ the dynamics start to get more complicated, stable limiting cycles occur (see Figures 8 - 3). Finally for $R = 4$ the system leads to chaotic behavior (see Figure 2). Figure 1 illustrates the bifurcation structure.

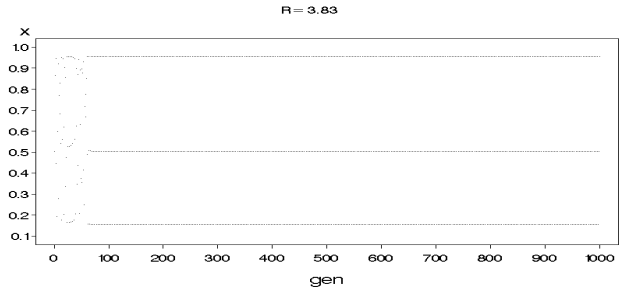

Figure 3: *The figure shows convergence toward a limiting cycle of period three.*

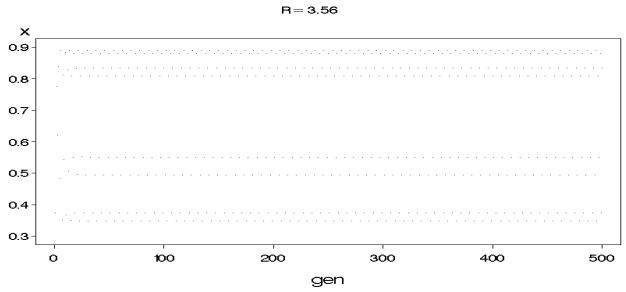

Figure 4: *The figure shows convergence toward a limiting cycle of period eight.*

3 Brief description of a basic population genetic model

In this section we will introduce the basic discrete, deterministic one-locus selection mutation model for arbitrary changing environments.

We suppose that the alleles $\mathcal{A}_1, \ldots, \mathcal{A}_n$ can occur on a single autosomal locus of a sufficiently large, random mating population (so that random genetic drift can be neglected). Generations are assumed to be discrete and nonoverlapping. We further assume that genotype frequencies are the same in both sexes, that selection acts solely through differential viabilities and that mutation only occurs during reproduction. We denote the (relative) frequency of unordered $\mathcal{A}_i\mathcal{A}_j$ heterozygotes at the zygote stage of generation t by $2P_{ij} = 2P_{ij}(t)$ and that of $\mathcal{A}_i\mathcal{A}_i$ homozygotes by $P_{ii} = P_{ii}(t)$. Therefore, the frequency of the allele $\mathcal{A}_i$ is

$$p_i = \sum_{j=1}^{k} P_{ij} \ . \tag{3}$$

Let μ_{ij} denote the mutation rate from $\mathcal{A}_i$ to $\mathcal{A}_j$ if $i \neq j$ and $\mu_{ii} = 0$. Since we assume random mating the P_{ij} are in Hardy-Weinberg proportions, i.e., $P_{ij} = p_i p_j$. We denote the fitness (viability) of $\mathcal{A}_i\mathcal{A}_j$ individuals by W_{ij}, satisfying $W_{ij} \geq 0$ and $W_{ij} = W_{ji}$. Hence, the frequency of $\mathcal{A}_i\mathcal{A}_j$ genotypes among adults that have

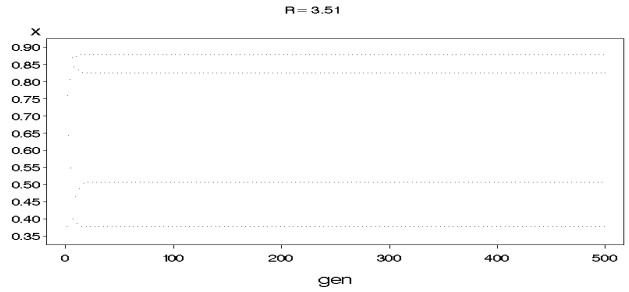

Figure 5: *The figure shows convergence toward a limiting cycle of period four.*

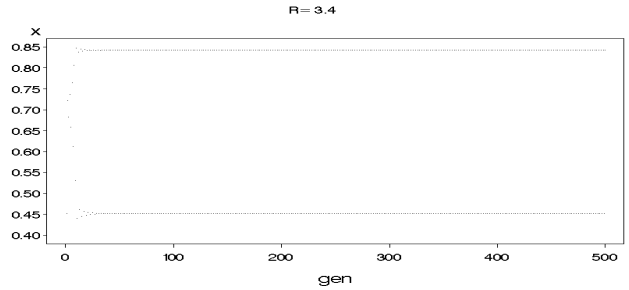

Figure 6: *The figure shows convergence toward a limiting cycle of period two.*

survived selection is

$$P_{ij}^* = P_{ij}^*(t) = \frac{W_{ij}P_{ij}}{\overline{W}} = \frac{W_{ij}p_ip_j}{\overline{W}} \ , \tag{4}$$

where

$$\overline{W} = \overline{W}(t) = \sum_{i,j=1}^{k} W_{ij}(t)P_{ij}(t) \tag{5}$$

$$= \sum_{i,j=1}^{k} W_{ij}p_ip_j = \sum_{i=1}^{k} W_ip_i \tag{6}$$

is the *mean fitness* of the population and

$$W_i = \sum_{j=1}^{k} W_{ij}p_j \tag{7}$$

is the *marginal fitness* of allele $\mathcal{A}_i$. Thus, the frequency of $\mathcal{A}_i$ after selection is $p_i^* = \sum_{j=1}^{k} P_{ij}^* = p_iW_i/\overline{W}$. Then during recombination mutation occurs, thus the fraction of $\mathcal{A}_i$ genes that do not mutate is $1-\sum_{j=1}^{n}\mu_{ij}$, and $\mathcal{A}_j$ genes give rise to a mutant $\mathcal{A}_i$ with probability μ_{ji}. Therefore the frequency p_i' of $\mathcal{A}_i$ in the next generation is

$$\begin{aligned} p_i' &= p_i^*\Big(1-\sum_{j=1}^{n}\mu_{ij}\Big) + \sum_{j=1}^{n} p_j^*\mu_{ji} \\ &= p_i\frac{W_i}{\overline{W}}\Big(1-\sum_{j=1}^{n}\mu_{ij}\Big) + \frac{1}{\overline{W}}\sum_{j=1}^{n}\Big(p_jW_j\Big)\mu_{ji} \ . \end{aligned}$$

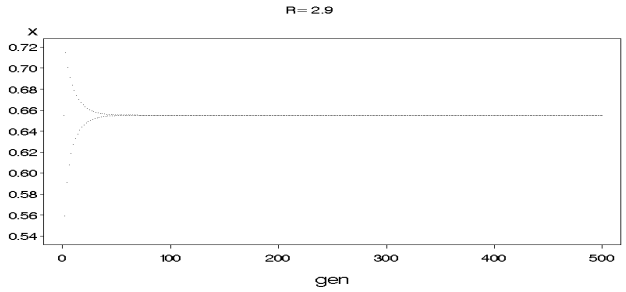

Figure 7: *The figure shows convergence toward an equilibrium.*

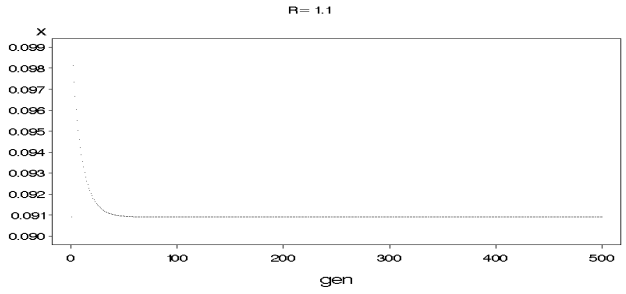

Figure 8: *The figure shows convergence toward an equilibrium.*

Therefore we obtain the *selection-mutation equation*

$$p_i' = p_i\frac{W_i}{\overline{W}} + \frac{1}{\overline{W}}\sum_{j=1}^{n}(p_jW_j\mu_{ji} - p_iW_i\mu_{ij}) \ . \tag{8}$$

In the following we will assume cyclical selection, i.e., fitnesses change periodically in time. We can express this mathematically as $W_{ij}(t) = W_{ij}(t+L)$ $\forall\, t \in \mathbb{N}$, where L is the period length.

For given initial distributions of allele frequencies, fitnesses and mutation rates, we want to observe the the long time behavior of the system (8). The trajectories may converge to a fixed point or a limiting cycle or may behave chaotically.

4 Implementation on SAS/IML®

First we write the fitnesses in generation $s = t + kL$ ($k \in \mathbb{N}$, $t \in \{0, \ldots, L-1\}$) as a symmetric $n \times n$ matrix

$$W(t) = \begin{pmatrix} W_{11}(t) & W_{12}(t) & \ldots & W_{1n}(t) \\ W_{12}(t) & W_{22}(t) & \ldots & W_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ W_{1n}(t) & W_{2n}(t) & \ldots & W_{nn}(t) \end{pmatrix} \ . \tag{9}$$

Furthermore we successive append these matrixes and obtain a $n \times Ln$ matrix W. Further let

$$\text{one} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \tag{10}$$

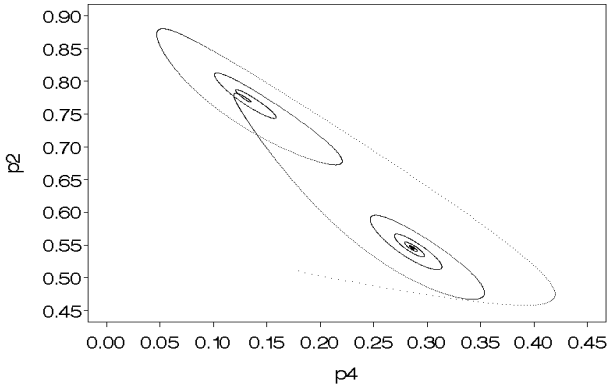

Figure 9: *The figure shows a spiral trajectory in the* (p_2, p_4)*-plane converging to an equilibrium.*

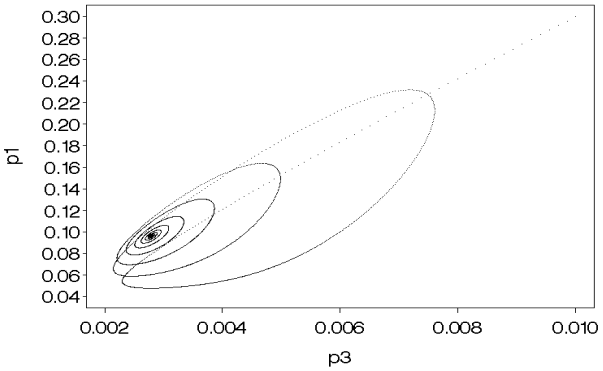

Figure 10: *The figure shows a spiral trajectory in the* (p_1, p_3)*-plane converging to an equilibrium.*

and

$$\mathrm{K1} = \left(\sum_{j=1}^{n} \mu_{1j} \ , \quad \cdots \quad , \sum_{j=1}^{n} \mu_{nj} \right) . \qquad (11)$$

We fix the number n of alleles and the length L of the environmental period. Then we create a SAS® data set containing the fitness matrices for several independent environmental cycles - be aware that we assume cyclical selection. Thus, each environmental cycle is represented by a $n \times Ln$-matrix. For instance, suppose $n = 4$, $L = 15$ and we want to create 10 independent environmental cycles the data set containing the fitness matrices has 60 variables and 40 observations. Furthermore, we create a SAS® data set containing several mutation matrices, and a SAS® data set containing several initial distributions of allele frequencies as row vectors. For instance, let us assume we have 5 different initial distributions and the number of alleles is $n = 6$ then we have a data set containing six columns and five observations. For each combination of initial frequencies, mutation matrices and environmental cycles we want to observe the long time behavior of the trajectory. Therefore, we iterate the selection cycle several times and store the distribution of allele frequencies at the beginning of the cycles. We stop the iterations if an equilibrium is numerically reached, i.e., the euclidian distance between the allele vectors of allele frequencies of two consecutive cycles is less than a given size `eps` or if no equilibrium is reached within `num` generations. The frequencies are stored in a SAS® data set.

We want to consider an example: Suppose we have three alleles and the environmental cycles contains two environments. We generate three independent environmental cycles. As an example consider the following matrices:

1st environment			2nd environment		
1.240	1.647	1.470	1.963	0.822	0.480
1.647	1.623	0.561	0.822	0.257	0.856
1.470	0.561	2.000	0.480	0.856	0.531
0.919	0.340	0.099	0.624	0.337	0.024
0.340	0.277	1.490	0.337	1.913	0.136
0.099	1.490	0.055	0.024	0.136	0.412
1.158	0.036	0.555	1.147	1.688	1.785
0.036	0.959	0.652	1.688	1.745	0.940
0.555	0.652	0.153	1.785	0.940	0.526

Now we assign in the names of the needed data sets and the other quantities to macro variables.

```
/*maximum number of iterations*/
%let num=50000000;
/*accuracy of convergence */
%let eps=1e-15;
/*length of environmental cycles*/
%let L=2;
/*number of alleles*/
%let n=3;
/*name of data set containing the initial
frequencies*/
%let pname=%str(cycles.initialfreq9);
/*name of data set containing the
fitnesses*/
%let wname=%str(cycles.fit6a);
/*name of data set containing the
mutation rates*/
%let mname=%str(cycles.M3);
/*name of output data set*/
%let simulationdata=%str(cycles.conv6a);
```

The output data set should contain a variable that indicates if an equilibrium is reached or not, it should group the data set for each combination of environmental cycles, mutationrates and initial frequencies. For each of this combination the numbering of observations is reset.

The following datastep creates the output data set:

```
data &simulationdata;
array x p1-p&n generation iternum conv;
```

```
delete;
run;
```

Now we are able to implement the recursion relation (8) on SAS/IML®, with the given parameters and initial conditions.

The programm begins by initializing the matrices and macro variables:

```
proc iml worksize=400000000;
n=&n;
eps=&eps;
use &pname;
read all into p;
use &wname;
read all into wmat;
use &mname;
read all into m;
iteratio=0;
anzahl=ncol(p);
L=ncol(wmat)/n;
W=J(n,L*n,.);
aa=J(1,n+3,.);
one=J(n,1,1);
a=J(&num,n+3,.);
hy=J(n,1,.);
hz=J(n,1,.);
x=J(n,1,.);
K1=J(n,1,.);
```

In the next step the mutation rates, fitness matrices and initial frequencies are determined in iterative loops:

```
do mm=1 to nrow(m)/ncol(m);
m1=m[(1+(mm-1)*ncol(m)):mm*ncol(m),];
K1=m1[,+];
do wm=1 to nrow(w)/n;
W=wmat[(1+(wm-1)*n):(wm*n),1:L*n];
do pm=1 to nrow(p);
p1=p[pm,];
iteratio=iteratio+1;
a[1,1:n]=p1[1,1:n];
a[1,n+1]=1;
a[1,n+2]=iteratio;
x[1:n,1]=t(a)[1:n,1];
xt=T(x);
knew=&num;
```

Now the main part of the loop is explained. In this part for a given environmental cycle, given mutation rates and given initial frequencies the selection-mutation cycle is iterated with respect to (8). Also the frequencies on the beginning of the environmental cycles are stored. The iteration stops if an equilibrium is numerically reached or if no equilibrium is established within num generations.

```
do k=2 to &num;
do t=0 to L-1;
w1=w[1:n,1+t*n:(t+1)*n];
hy=w1*x;
hz=t(m1)*(hy#x);
x=(1/(xt*hy))*(x#(hy#(one-K1))+hz);
xt=T(x);
end;
a[k,1:n]=xt[1,1:n];
a[k,n+1]=k;
a[k,n+2]=iteratio;
if ssq(a[k,1:n]-a[k-1,1:n])<eps**2
then do;
knew=k;
k=&num;
end;
end;
```

Finally the quantities of interest are written to the output data set.

```
if knew<&num then do;
a[1:knew,n+3]=0;
end;
else do;
a[1:knew,n+3]=1;
end;
b=J(knew,n+3,.);
b[1:knew,1:n+3]=a[1:knew,1:n+3];
edit &simulationdata;
append from b;
end;
end;
end;
quit;
```

5 Visualization

Now we show how SAS® procedures can be used to visualize the created data, and how to export the created graphic into an external file.

We can plot the phase portraits of the system (8) at

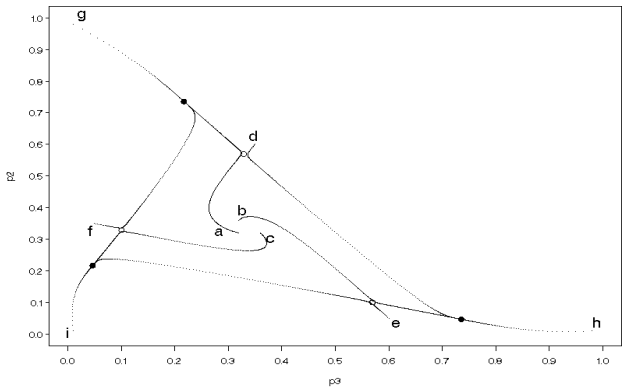

Figure 11: *Phase portrait: the labels are created using the Annotate facility.*

the beginning of the consecutive environmental cycles by using `proc gplot`. It is often desirable to label the trajectories occurring in phase portrait. We will show how this can be done using the Annotate facility (see Figure 11. Since the starting points of the trajectories are known we can place the label near them. We consider a simple example how to generate an Annotate data set:

```
data anno;
length text $9.  color style function $8.;
retain xsys '1' ysys '1' hsys '1' when 'a';
function='symbol'; size=4;
function='label';
size=4;
ftext=SWISS;
position='6';
x=29; y=35; text='a'; output;
x=33; y=41; text='b'; output;
x=38; y=33; text='c'; output;
x=35; y=63; text='d'; output;
x=60; y=8; text='e'; output;
x=7; y=35; text='f'; output;
x=5; y=99; text='g'; output;
x=95; y=8; text='h'; output;
x=3; y=5; text='i'; output;
run;
```

In order to obtain a graphic, appropriate for presentations, we set some graphical options and since we want to export the graphic we have to specify the output device, e.g., a bitmap file (Figures 9 - 11 were created in a similar way):

```
goptions ftext=SWISS ctext=BLACK
htext=1 cells gsfname=grafout
gsfmode=replace device=bmp;
axis1 width=1 offset=(3 pct)
label=(a=90 r=0);
axis2 width=1 offset=(3 pct);
symbol1 c=black ci=black v=point
height=1 cells interpol=NONE l=1 w=1;
symbol2 f=marker c=black ci=blue v='P'
height=1 cells interpol=NONE l=1 w=1;
symbol3 f=marker c=black ci=blue v='C'
height=1 cells interpol=NONE l=1 w=1;
goptions reset=f;
```

We then have to set a path where to store the graphic output:

```
filename grafout 'C:\ pictures \ pict.bmp';
```

Finally we use the gplot procedure:

```
proc gplot data=&simulationdata ;
plot p2 * p3 =conv /
nolegend anno=anno
name='SCAT'
caxis = BLACK
ctext = BLACK
```

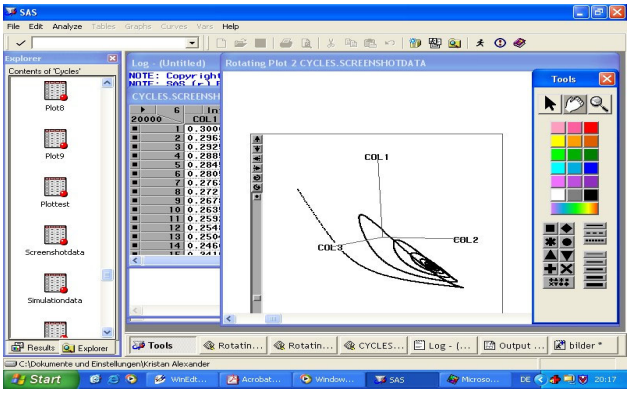

Figure 12: *Rotating plot*

```
cframe =white
hminor = 0
vminor = 0
vaxis = axis1
haxis = axis2;
run;
quit;
```

Furthermore, the Interactive Data Analysis allows you to create rotating plots. This is a useful tool for investigating phase portraits (see Figure 12).

6 Conclusions

SAS/IML® can be used for numerical iterations of recurrence relations reflecting a biological background, as the above examples illustrate. Afterwards various SAS® tools can be used to handle the created data and to apply statistical analysis. Furthermore it turns out that SAS/GRAPH® is an appropriate tool to illustrate the created data and the applied statistics. However, the efficiency of your SAS/IML® code depends to a great extent on the implementation. Thus, for extensive numerical simulations it seems to be inevitable to focus on efficient programming in order to perform your simulations in adequate time. For data intensive simulation you data management is another important factor.

7 Contact Information

Kristan Schneider
address: Münichreiterstr.12 A-1130 Vienna Austria Europe
e-mail: a9900273@ unet.univie.ac.at
phone: +43 1 0664 19 62 741
fax: +43 1 877 48 50