TFY4235/FY8904 Computational Physics, Spring 2013

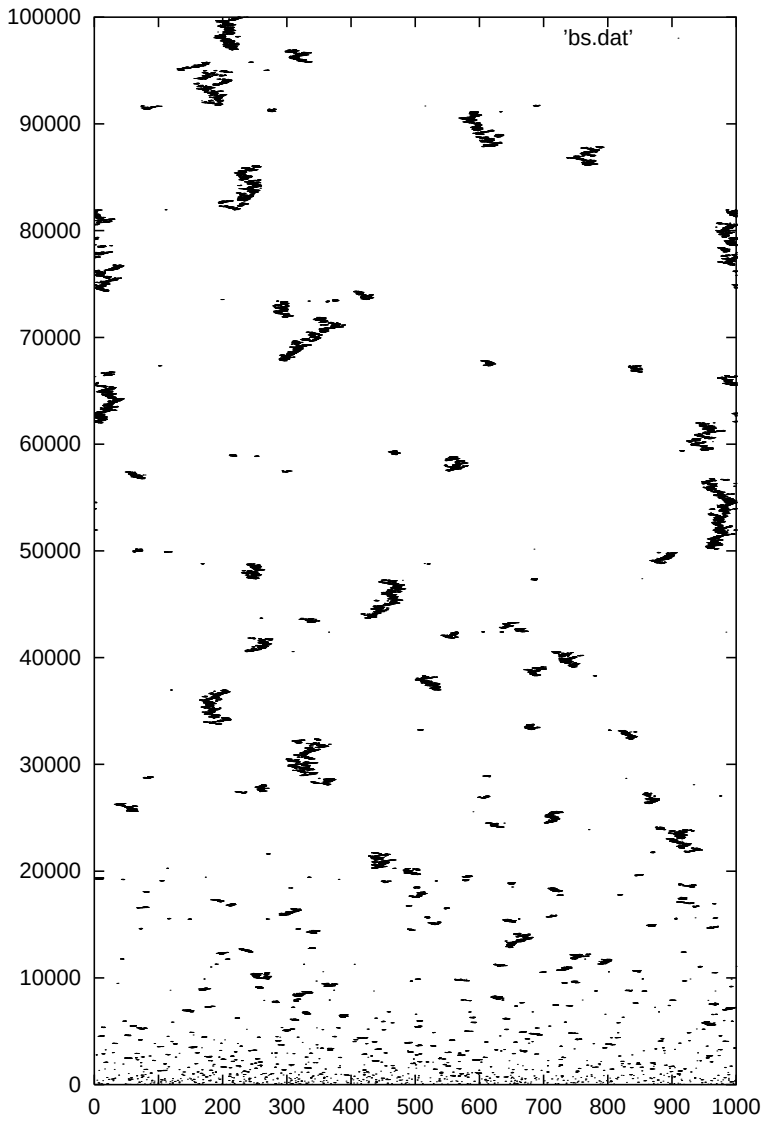
Solution Set 1

Problem 1.

I have implemented a program simulating the Bak-Sneppen model as follows:

```
      program bs
c Program implementing Bak-Sneppen model
      parameter(l=1000,mtime=100000,idum=4711)
      dimension fitnes(l),ir(l),il(l)
c Heating up random number generator
      ibm=idum
      rinv=0.5/(2.**31-1.)
      do i=1,1000
      ibm=ibm*16807
      enddo
c Implementing periodic boundary conditions
      do i=1,l
      ir(i)=i+1
      il(i)=i-1
      enddo
      ir(l)=1
      il(1)=l
c Initializing fitness vector
      do i=1,l
      ibm=ibm*16807
      fitnes(i)=(ibm*rinv+0.5)
      enddo
c Loop over time
      do itime=1,mtime
c finding smallest fitness factor
      is=0
      fs=2.0
      do i=1,l
      if(fitnes(i).le.fs) then
      is=i
      fs=fitnes(i)
      endif
      enddo
c Updating element with smallest fitness and neighbors
      ibm=ibm*16807
      fitnes( is )=(ibm*rinv+0.5)
      ibm=ibm*16807
```

```
      fitnes(ir(is))=(ibm*rinv+0.5)
      ibm=ibm*16807
      fitnes(il(is))=(ibm*rinv+0.5)
c Writing position of least fit element
      write(*,*) is,itime
      enddo
      end
```



There are two aspects of this program that is not straight forward: 1) The implementation of the random number generator. This is a so-called portable generator that goes under the name "16807" and which works by overflow. We will describe this in detail later on in the course. The second aspect is the implementation of the periodic boundary conditions. The way I have implemented them is the most efficient: There are no logical tests to perform. In the plot below I have reproduced Fig. 3a in the Bak-Sneppen paper. It shows clearly

how the activity in the model is "clumped" together. This clumping corresponds to the avalanches of change (think about this!). Here the x-axis shows position and the y-axis shows time.