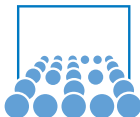


# Scientific Computing I

## Module 2: Population Modelling – Discrete Models

Tobias Neckel

Winter 2015/2016



# Outline – Algebraic Population Models

## Fibonacci's Rabbits

## Game of Life

## PageRank

Ranking of Websites

Stochastic Matrices, Markov Chains

PageRank in Practice: Vector Iteration

Random Surfer Model

# Fibonacci's Rabbits

*A pair of rabbits are put in a field.  
If rabbits take a month to become mature  
and then produce a new pair every month,  
how many pairs will there be in twelve months time?*

Leonardo Pisano ("Fibonacci"), A.D. 1202

# Model Assumptions

Which assumptions or simplifications have been made?

# Model Assumptions

Which assumptions or simplifications have been made?

- we consider pairs of rabbits

# Model Assumptions

Which assumptions or simplifications have been made?

- we consider pairs of rabbits
- rabbits reproduce exactly once a month

# Model Assumptions

Which assumptions or simplifications have been made?

- we consider pairs of rabbits
- rabbits reproduce exactly once a month
- female rabbits always give birth to a pair of rabbits

# Model Assumptions

Which assumptions or simplifications have been made?

- we consider pairs of rabbits
- rabbits reproduce exactly once a month
- female rabbits always give birth to a pair of rabbits
- newborn rabbits require one month to become mature



# Model Assumptions

Which assumptions or simplifications have been made?

- we consider pairs of rabbits
- rabbits reproduce exactly once a month
- female rabbits always give birth to a pair of rabbits
- newborn rabbits require one month to become mature
- rabbits don't die

# Model Assumptions

Which assumptions or simplifications have been made?

- we consider pairs of rabbits
- rabbits reproduce exactly once a month
- female rabbits always give birth to a pair of rabbits
- newborn rabbits require one month to become mature
- rabbits don't die
- ... ?

# The Fibonacci Numbers

How many pairs of rabbits are there?

# The Fibonacci Numbers

How many pairs of rabbits are there?

- we start with a newborn pair of rabbits

# The Fibonacci Numbers

How many pairs of rabbits are there?

- we start with a newborn pair of rabbits
- after one month: still 1 pair of rabbits (now mature)

# The Fibonacci Numbers

How many pairs of rabbits are there?

- we start with a newborn pair of rabbits
- after one month: still 1 pair of rabbits (now mature)
- after two months: 2 pairs of rabbits (one mature)

# The Fibonacci Numbers

How many pairs of rabbits are there?

- we start with a newborn pair of rabbits
- after one month: still 1 pair of rabbits (now mature)
- after two months: 2 pairs of rabbits (one mature)
- after three months: 3 pairs of rabbits (two mature)

# The Fibonacci Numbers

How many pairs of rabbits are there?

- we start with a newborn pair of rabbits
- after one month: still 1 pair of rabbits (now mature)
- after two months: 2 pairs of rabbits (one mature)
- after three months: 3 pairs of rabbits (two mature)



# The Fibonacci Numbers

How many pairs of rabbits are there?

- we start with a newborn pair of rabbits
- after one month: still 1 pair of rabbits (now mature)
- after two months: 2 pairs of rabbits (one mature)
- after three months: 3 pairs of rabbits (two mature)
- after  $n$  months:

$$f_n = f_{n-1} + f_{n-2}, \quad f_0 = f_1 = 1$$

# The Fibonacci Numbers

How many pairs of rabbits are there?

- we start with a newborn pair of rabbits
- after one month: still 1 pair of rabbits (now mature)
- after two months: 2 pairs of rabbits (one mature)
- after three months: 3 pairs of rabbits (two mature)
- after  $n$  months:

$$f_n = f_{n-1} + f_{n-2}, \quad f_0 = f_1 = 1$$

# The Fibonacci Numbers

How many pairs of rabbits are there?

- we start with a newborn pair of rabbits
- after one month: still 1 pair of rabbits (now mature)
- after two months: 2 pairs of rabbits (one mature)
- after three months: 3 pairs of rabbits (two mature)
- after  $n$  months:

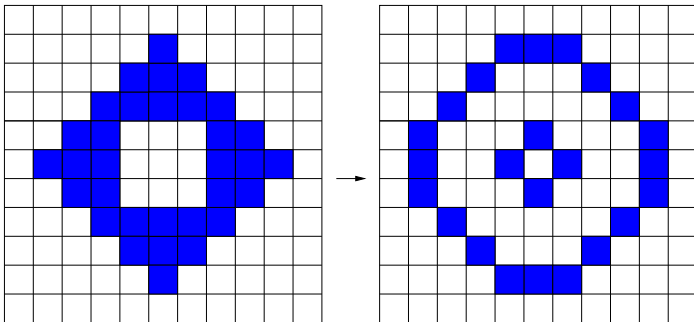
$$f_n = f_{n-1} + f_{n-2}, \quad f_0 = f_1 = 1$$

- exponential growth of rabbits (see tutorials):

$$f_n = \frac{1}{\sqrt{5}} (\phi^n - (1 - \phi)^n),$$

where  $\phi = \frac{1}{2} (1 + \sqrt{5}) \approx 1.61 \dots$  is the golden section number.

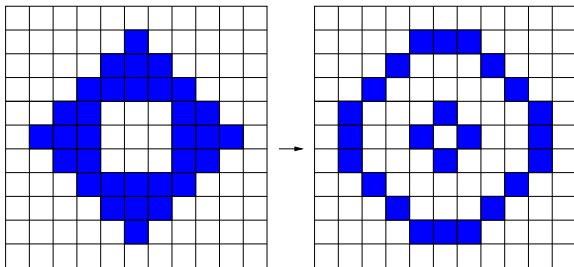
# Conway's Game of Life



## Cellular Automaton:

- cells are either “alive” or “dead”
- synchronous status update of all cells
- depending on the status of the neighbour cells

# Game of Life – Update Rules



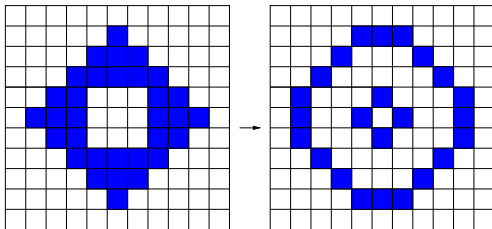
## A living cell:

- stays alive, if it has exactly 2 or 3 living neighbours
- dies, if it has more or less neighbours

## A dead cell:

- comes alive, if it has exactly 3 living neighbours

# Game of Life – Modelling Questions



## Modelling Questions:

- are there any stable states? (or cycles?)
- is extinction or infinite growth of the population possible?
- how do such scenarios look like?
- is it possible to explicitly compute such scenarios?

# PageRank: Website Ranking and “Random Surfer” Models

- given:  $n$  websites connected by hyperlinks
- wanted: rank websites according to “importance”
- idea: rank depends on links to a website

# PageRank: Website Ranking and “Random Surfer” Models

- given:  $n$  websites connected by hyperlinks
- wanted: rank websites according to “importance”
- idea: rank depends on links to a website

## Quantitative approach – count the links:

- Graph model:  
websites  $\rightarrow$  nodes, links  $\rightarrow$  edges
- represented as *adjacency matrix*:  
 $A_{ij} = 1$  if an edge exists from  $i$  to  $j$ , (else  $A_{ij} = 0$ )
- ranking depends on number of edges to  $j$

$$r(j) := \sum_{i \neq j} A_{ij} \quad (\text{column sum})$$



# PageRank

## Qualitative approach:

- Goal: links from "important" website have higher impact
- Step 1: add weights (rank) instead of number of links
- Example: page 3 and 4 link to page 2  
⇒ impact  $x_2$  of page 2 is  $x_2 = x_3 + x_4$

# PageRank

## Qualitative approach:

- Goal: links from "important" website have higher impact
- Step 1: add weights (rank) instead of number of links
- Example: page 3 and 4 link to page 2  
⇒ impact  $x_2$  of page 2 is  $x_2 = x_3 + x_4$

## Modelled by adjacency matrix:

- leads to system of equations:

$$x_j = \sum_{i \neq j} A_{ij} x_i = \sum_{i \neq j} (A^T)_{ji} x_i$$

- in matrix-vector notation:  $x = A^T x$   
(search eigenvector for eigenvalue 1)

## PageRank (2)

- Goal: reduce influence of pages with many links
- Step 2: weights divided by number of outgoing links (each website has a "total impact"/sum of weights of 1)
- Example: page 3 (three outgoing links) and 4 (two links) link to page 2  
⇒ impact  $x_2$  of page 2 is  $x_2 = \frac{1}{3}x_3 + \frac{1}{2}x_4$

## PageRank (2)

- Goal: reduce influence of pages with many links
- Step 2: weights divided by number of outgoing links (each website has a "total impact"/sum of weights of 1)
- Example: page 3 (three outgoing links) and 4 (two links) link to page 2  
⇒ impact  $x_2$  of page 2 is  $x_2 = \frac{1}{3}x_3 + \frac{1}{2}x_4$

### Modelled by adjacency matrix:

- $n_i$ : number of outgoing links of page  $i$ ;  $n_i = \sum_j A_{ij}$
- Resulting system of equations:

$$x_j = \sum_{i \neq j} \frac{1}{n_i} A_{ij} x_i = \sum_{i \neq j} \left( \frac{1}{n_i} (A^T)_{ji} \right) x_i$$

# Page-Rank Matrix

- set  $B_{ji} := \frac{1}{n_i}(A^T)_{ji} \rightarrow$  leads to system of equations:

$$x_j = \sum_{i \neq j} \frac{1}{n_i} A_{ij} x_i = \sum_{i \neq j} B_{ji} x_i$$

- search eigenvector for eigenvalue 1:  $x = Bx$

# Page-Rank Matrix

- set  $B_{ji} := \frac{1}{n_i}(A^T)_{ji} \rightarrow$  leads to system of equations:

$$x_j = \sum_{i \neq j} \frac{1}{n_i} A_{ij} x_i = \sum_{i \neq j} B_{ji} x_i$$

- search eigenvector for eigenvalue 1:  $x = Bx$

## Properties of the page-rank matrix:

- all column sums are 1
- all  $B_{ji} \geq 0$ , diagonal elements  $B_{jj} = 0$   
(linking to your own page is not counted)
- $B$  is a so-called (left) **stochastic matrix**

# Stochastic Matrices – Properties

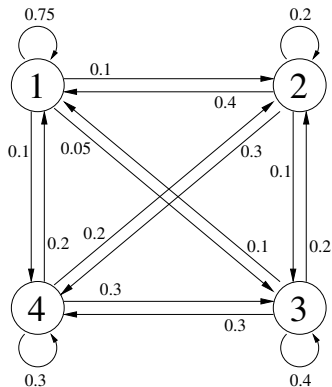
$B$  a stochastic matrix, then:

1.  $B$  has 1 as eigenvalue;  
all elements of the corresp. eigenvectors  $b^{(1)} \geq 0$   
 $\rightarrow$  normalise  $b^{(1)}$ , such that  $\sum b_j^{(1)} = 1$
2. element sum of  $y = Bx$  is equal to the element sum of  $x$ ;  
if  $x \geq 0$  (element-wise), then also  $y \geq 0$
3.  $v$  an eigenvector of  $B$  with eigenvalue  $\neq 1$ ,  
 $\Rightarrow$  element sum of  $v$  equal to 0
4.  $\lambda$  eigenvalue of  $B$ , then  $|\lambda| \leq 1$

(without proofs  $\rightarrow$  see a resp. textbook)

# Compare: Markov Chain

- finite set of states with certain possible state transitions
- change of states subject to given probability
- probability only depends on current state (“memoryless”)





# Vector Iteration with Stochastic Matrices

- examine iteration  $x(m) = Bx(m-1)$ ,  
start vector  $x(0) \geq 0$  has element sum 1
- use eigenvector decomposition of  $x(0)$ :

$$x(0) = \sum_j \gamma_j b^{(j)}$$

- then:  $x(m) = B^m x(0) = \sum_j \gamma_j \lambda_j^m b^{(j)}$
- if  $\lambda_1 = 1$  and all other  $0 < \lambda_j < 1$ , then:

$$x(m) = \sum_j \gamma_j \lambda_j^m b^{(j)} \rightarrow \gamma_1 b^{(1)} \quad \text{for } m \rightarrow \infty$$

# PageRank in Practice

- use a start vector  $x(0)$  with element sum 1  
(then:  $\gamma_1 = 1$  can be assumed via normalisation of  $b^{(1)}$ )
- vector iteration  $x(m) = Bx(m-1)$  converges to ranking vector  $\gamma_1 b^{(1)} = b^{(1)}$  (with element sum 1)
- as every page has only few outgoing links  
→  $B$  a sparse matrix
- $n$  pages with an average of  $k$  links per page:  
→  $kn$  mult/add operations per iteration
- convergence faster for smaller values of the largest eigenvalue  
(except  $\lambda_1 = 1$ )

# Vector Iteration in Practice

## Problem: 2 separate partitions

- consider the following page-rank matrix:

$$B = \begin{pmatrix} B_I & 0 \\ 0 & B_{II} \end{pmatrix}$$

(web divided into two non-linked partitions)

# Vector Iteration in Practice

## Problem: 2 separate partitions

- consider the following page-rank matrix:

$$B = \begin{pmatrix} B_I & 0 \\ 0 & B_{II} \end{pmatrix}$$

(web divided into two non-linked partitions)

- $B_I$  and  $B_{II}$  are stochastic matrices, each with eigenvectors  $b_I$  and  $b_{II}$  for eigenvalue 1
- $(b_I \ b_{II})^T$ , but also  $(b_I \ 0)^T$  and  $(0 \ b_{II})^T$  are eigenvectors of  $B$  (for eigenvalue 1)
- consequences for convergence and ranking?

## Vector Iteration in Practice (2)

### Problem: slow convergence

- happens, if at least one  $\lambda \approx 1$  (but  $\neq \lambda_1 = 1$ )
- modify page-rank matrix  $B$ :

$$\tilde{B} \rightsquigarrow \alpha B + (1 - \alpha) \frac{1}{n} \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$$

- new system of equations  $x = \tilde{B}x$ ,  
or:  $x = \alpha Bx + (1 - \alpha) \frac{1}{n} \mathbf{e} \mathbf{e}^T x$ , with  $\mathbf{e} = (1, \dots, 1)^T$
- equivalent to:  $x - \alpha Bx = (1 - \alpha) \frac{1}{n} \mathbf{e} \mathbf{e}^T x$
- $\frac{1}{n} \mathbf{e} \mathbf{e}^T$  stochastic, therefore  $\tilde{B}$  a stochastic matrix, as well

# Vector Iteration in Practice (3)

## Regularisation

- $x = \tilde{B}x$  iff  $x - \alpha Bx = (1 - \alpha)\frac{1}{n}ee^T x$
- as  $e^T x = 1$  (element sum = 1):

$$(I - \alpha B)x = \frac{1}{n}(1 - \alpha)e \quad \text{where } 0 < \alpha < 1$$

- eigenvalues of  $\alpha B$  are  $\leq \alpha$   
 $\Rightarrow (I - \alpha B)$  not singular (all eigenvalues  $\geq 1 - \alpha$ )
- leads to **unique solution**

# Vector Iteration in Practice (4)

## Convergence

- compute solution via vector iteration:

$$x(m) = \alpha Bx(m-1) + (1-\alpha)\frac{1}{n}e$$

- corresponds to iteration for error vector  $\epsilon(m) = x(m) - x$ :

$$\epsilon(m) = \alpha B\epsilon(m-1)$$

- now: all eigenvalues of  $\alpha B$  are  $\leq \alpha$
- therefore  $\|\epsilon(m)\| \sim \alpha^m \|\epsilon(0)\|$   
(convergence faster for smaller  $\alpha$ )

# Vector Iteration in Practice (5)

## Regularisation and Convergence

- Vector iteration converges faster for smaller  $\alpha$
- solution is better, the closer  $\alpha$  is to 1  
(then  $\tilde{B} \approx B$ )
- task: find an optimal  $\alpha$   
(common page-rank choice:  $\alpha = 0.85$ )
- regularisation parameter balances between exact solution and “well-behaved” problem
- regularisation therefore a frequent technique for ill-posed problems



# PageRank as a Population Model: Random Surfer

- each website “populated” by  $x_i$  web surfers
- total population:  $\sum x_i = 1$  (normalised)
- population corresponds to page rank: how many surfer are expected to be on each site?

# PageRank as a Population Model: Random Surfer

- each website “populated” by  $x_i$  web surfers
- total population:  $\sum x_i = 1$  (normalised)
- population corresponds to page rank: how many surfer are expected to be on each site?

## PageRank: “Random Surfer”

- surfers randomly follow a link from the current page (and change to a different website)
- thus:  $\frac{1}{n_i} A_{ij} x_i$  surfers change to website  $j$
- regularisation: with probability  $(1 - \alpha)$ , a surfer will jump to another (random) page in the internet
- vector iteration  $\rightarrow$  population evolves towards an equilibrium

