Evaluating stationary and quasi-stationary distributions for Markov chains with a large sparse transition structure

Phil Pollett

AUSTRALIAN RESEARCH COUNCIL
Centre of Excellence for Mathematics
and Statistics of Complex Systems

Markovian chain

Let $(X(t),\, t \geq 0)$ is be a continuous time Markov chain with transition rates

$$Q = (q_{ij},\, i, j \in S),$$

so that q_{ij} represents the rate of transition from state i to state j, for $j \neq i$, and $q_{ii} = -q_i$, where

$$q_i := \sum_{j \neq i} q_{ij} \ (< \infty)$$

represents the total rate out of state i.

Equilibrium distribution

This is a probability distribution $\pi = (\pi_i,\, i \in S)$ satisfying the balance equations

$$\sum_{i \neq j} \pi_i q_{ij} = \pi_j \sum_{i \neq j} q_{ji}, \qquad j \in S,$$

Equilibrium distribution

This is a probability distribution $\pi = (\pi_i,\ i \in S)$ satisfying the balance equations

$$\sum_{i \neq j} \pi_i q_{ij} = \pi_j \sum_{i \neq j} q_{ji} \quad (= \pi_j q_j = -\pi_j q_{jj}\ ), \quad j \in S,$$

Equilibrium distribution

This is a probability distribution $\pi = (\pi_i,\, i \in S)$ satisfying the balance equations

$$\sum_{i \neq j} \pi_i q_{ij} = \pi_j \sum_{i \neq j} q_{ji} \quad \left(= \pi_j q_j = -\pi_j q_{jj} \right), \quad j \in S,$$

that is, $\sum_{i \in S} \pi_i q_{ij} = 0$, $j \in S$.

Equilibrium distribution

This is a probability distribution $\pi = (\pi_i,\, i \in S)$ satisfying the balance equations

$$\sum_{i \neq j} \pi_i q_{ij} = \pi_j \sum_{i \neq j} q_{ji} \quad \left(= \pi_j q_j = -\pi_j q_{jj} \right), \quad j \in S,$$

that is, $\sum_{i \in S} \pi_i q_{ij} = 0$, $j \in S$. If, for example, S is irreducible and finite, then the equilibrium distribution exists uniquely and, for all $j \in S$,

$$\Pr(X(t) = j) \to \pi_j \quad \text{as } t \to \infty.$$

Equilibrium distribution

This is a probability distribution $\pi = (\pi_i,\ i \in S)$ satisfying the balance equations

$$\sum_{i \neq j} \pi_i q_{ij} = \pi_j \sum_{i \neq j} q_{ji} \quad \left(= \pi_j q_j = -\pi_j q_{jj} \right), \quad j \in S,$$

that is, $\sum_{i \in S} \pi_i q_{ij} = 0$, $j \in S$. If, for example, S is irreducible and finite, then the equilibrium distribution exists uniquely and, for all $j \in S$,

$$\Pr(X(t) = j) \to \pi_j \quad \text{as } t \to \infty.$$

$\boxed{\text{We need to be able to solve } \pi Q = 0}$

Example

A frog hops about on n stones, which are labelled in order of increasing temperature (he leaves the hotter ones more quickly). When he hops, he moves to any of other the $n-1$ stones with equal probability. Suppose he leaves stone i at rate $i(n-1)$.

Example

A frog hops about on n stones, which are labelled in order of increasing temperature (he leaves the hotter ones more quickly). When he hops, he moves to any of other the $n-1$ stones with equal probability. Suppose he leaves stone i at rate $i(n-1)$.

So, $S = \{1, 2, \ldots, n\}$ and

$$Q = \begin{pmatrix} -(n-1) & 1 & 1 & \cdots & 1 & 1 \\ 2 & -2(n-1) & 2 & \cdots & 2 & 2 \\ 3 & 3 & -3(n-1) & \cdots & 3 & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ n & n & n & \cdots & n & -n(n-1) \end{pmatrix}$$

Example

The balance equations are

$$\sum_{i \neq j} \pi_i q_{ij} = \pi_j q_j, \qquad j = 1, 2, \ldots, n,$$

Example

The balance equations are

$$\sum_{i \neq j} \pi_i q_{ij} = \pi_j q_j, \qquad j = 1, 2, \ldots, n,$$

that is,

$$\sum_{i \neq j} \pi_i \, i = \pi_j \, j(n-1), \qquad j = 1, 2, \ldots, n.$$

Example

The balance equations are

$$\sum_{i\neq j} \pi_i q_{ij} = \pi_j q_j, \qquad j = 1, 2, \ldots, n,$$

that is,

$$\sum_{i\neq j} \pi_i\, i = \pi_j\, j(n-1), \qquad j = 1, 2, \ldots, n.$$

Therefore, $i\pi_i =$ constant.

Example

The balance equations are

$$\sum_{i \neq j} \pi_i q_{ij} = \pi_j q_j, \qquad j = 1, 2, \ldots, n,$$

that is,

$$\sum_{i \neq j} \pi_i \, i = \pi_j \, j(n-1), \qquad j = 1, 2, \ldots, n.$$

Therefore, $i\pi_i =$ constant. And so,

$$\pi_i = \frac{1/i}{\sum_{j=1}^{n} 1/j}, \qquad i = 1, 2, \ldots, n.$$

Example

Let’s do this numerically:

```
n=5;
for i=1:n
  for j=1:n
    if (j ~= i) Q(i,j)=i; else Q(i,j)=-i*(n-1); end
  end
end

disp(Q)
        -4     1     1     1     1
         2    -8     2     2     2
         3     3   -12     3     3
         4     4     4   -16     4
         5     5     5     5   -20
```

Example

```
A=Q'; % Matlab calculates right eigenvectors
[V,D]=eig(A);

disp(D)
      0        0         0         0         0
      0  -6.7778         0         0         0
      0        0  -12.2804         0         0
      0        0         0  -23.2222         0
      0        0         0         0  -17.7196

disp(V)
   -0.8266  -0.8516  -0.2260   0.0831  -0.1294
   -0.4133   0.4699  -0.7216   0.1145  -0.2132
   -0.2755   0.1841   0.6051   0.1841  -0.6051
   -0.2066   0.1145   0.2132   0.4699   0.7216
   -0.1653   0.0831   0.1294  -0.8516   0.2260
```

Example

Extract the eigenvector corresponding to the eigenvalue with maximum real part (which is $\nu = 0$):

```
[nu,I]=max(real(diag(D)));
m=V(:,I);
pi=m/sum(m);

disp(pi')
   0.4380    0.2190    0.1460    0.1095    0.0876
```

Compare this with the one evaluated analytically:

```
a=1./(1:n);

disp(a/sum(a))
   0.4380    0.2190    0.1460    0.1095    0.0876
```

Quasi-equilibrium distribution

This is a probability distribution $\pi = (\pi_i,\, i \in C)$ satisfying

$$\sum_{i \in C} \pi_i q_{ij} = -\left(\sum_{i \in C} \pi_i q_{i0}\right) \pi_j, \qquad j \in C,$$

where C is an irreducible transient class and $S = \{0\} \cup C$, where 0 is an absorbing state which is accessible from C.

Quasi-equilibrium distribution

This is a probability distribution $\pi = (\pi_i,\, i \in C)$ satisfying

$$\sum_{i \in C} \pi_i q_{ij} = -\left(\sum_{i \in C} \pi_i q_{i0} \right) \pi_j, \qquad j \in C,$$

where C is an irreducible transient class and $S = \{0\} \cup C$, where 0 is an absorbing state which is accessible from C.

If, for example, C is finite, the quasi-equilibrium distribution exists uniquely and, for all $j \in C$,

$$\Pr(X(t) = j | X(t) \neq 0) \to \pi_j \quad \text{as } t \to \infty.$$

Quasi-equilibrium distribution

We need to be able to solve $\pi Q_C = \nu\pi$

Here Q_C is the restriction of Q to the transient states C (first row and column of Q removed).

Quasi-equilibrium distribution

$$\boxed{\text{We need to be able to solve } \pi Q_C = \nu\pi}$$

Here Q_C is the restriction of Q to the transient states C (first row and column of Q removed). Notice that

$$\sum_{i\in C} \pi_i q_{ij} = \nu\pi_j, \qquad j \in C,$$

implies $\nu = \nu \sum_{j\in C} \pi_j = \sum_{i\in C} \pi_i \sum_{j\in C} q_{ij} = -\sum_{i\in C} \pi_i q_{i0}.$

Quasi-equilibrium distribution

$\boxed{\text{We need to be able to solve } \pi Q_C = \nu\pi}$

Here Q_C is the restriction of Q to the transient states C (first row and column of Q removed). Notice that

$$\sum_{i\in C} \pi_i q_{ij} = \nu\pi_j, \qquad j \in C,$$

implies $\nu = \nu \sum_{j\in C} \pi_j = \sum_{i\in C} \pi_i \sum_{j\in C} q_{ij} = -\sum_{i\in C} \pi_i q_{i0}$. Compare with

$$\sum_{i\in C} \pi_i q_{ij} = -\left(\sum_{i\in C} \pi_i q_{i0}\right) \pi_j, \qquad j \in C.$$

The SIS model

Let $X(t)$ be the number of occupied patches at time t in a metapopulation consisting of n patches.

The SIS model

Let $X(t)$ be the number of occupied patches at time t in a metapopulation consisting of n patches.

Then, $S = \{0, 1, \ldots, n\}$, and $q_{i,i+1} = ci(1 - i/n)$ and $q_{i,i-1} = ei$, where c is the colonization rate and e is the local extinction rate.

The SIS model

Let $X(t)$ be the number of occupied patches at time t in a metapopulation consisting of n patches.

Then, $S = \{0, 1, \ldots, n\}$, and $q_{i,i+1} = ci(1 - i/n)$ and $q_{i,i-1} = ei$, where c is the colonization rate and e is the local extinction rate.

$$Q = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ e & -e - c(1 - 1/n) & c(1 - 1/n) & \cdots & 0 & 0 \\ 0 & 2e & -2e - c(1 - 2/n) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & en & -en \end{pmatrix}$$

The SIS model

Let $X(t)$ be the number of occupied patches at time t in a metapopulation consisting of n patches.

Then, $S = \{0, 1, \ldots, n\}$, and $q_{i,i+1} = ci(1 - i/n)$ and $q_{i,i-1} = ei$, where c is the colonization rate and e is the local extinction rate.

$$Q_C = \begin{pmatrix} -e - c(1-1/n) & c(1-1/n) & \cdots & 0 & 0 \\ 2e & -2e - c(1-2/n) & \cdots & 0 & 0 \\ 0 & 3e & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & en & -en \end{pmatrix}$$

$C = \{1, 2, \ldots, n\}$.

The SIS model

Evaluate the quasi-stationary distribution:

```
n=5; c=2; e=1; Q=zeros(n,n);
Q(1,2)=c*(1-1/n); Q(1,1)=-(c*(1-1/n)+e);
for i=2:(n-1)
  Q(i,i+1)=c*i*(1-i/n); Q(i,i-1)=e*i;
  Q(i,i)=-i*(c*(1-i/n)+e);
end
Q(n,n-1)=e*n; Q(n,n)=-e*n;

disp(Q)
   -2.6000    1.6000         0         0         0
    2.0000   -4.4000    2.4000         0         0
         0    3.0000   -5.4000    2.4000         0
         0         0    4.0000   -5.6000    1.6000
         0         0         0    5.0000   -5.0000
```

The SIS model

```
A=Q'; % Matlab calculates right eigenvectors
[V,D]=eig(A);

disp(D)
  -10.0783        0        0        0        0
         0  -6.8050        0        0        0
         0        0  -0.2350        0        0
         0        0        0  -4.0381        0
         0        0        0        0  -1.8436

disp(V)
    0.1054   0.3523  -0.4876   0.6927  -0.8441
   -0.3942  -0.7406  -0.5766  -0.4981  -0.3192
    0.6899   0.4059  -0.5404  -0.4295   0.1781
   -0.5703   0.3018  -0.3519   0.1526   0.3499
    0.1797  -0.2675  -0.1182   0.2538   0.1774
```

The SIS model

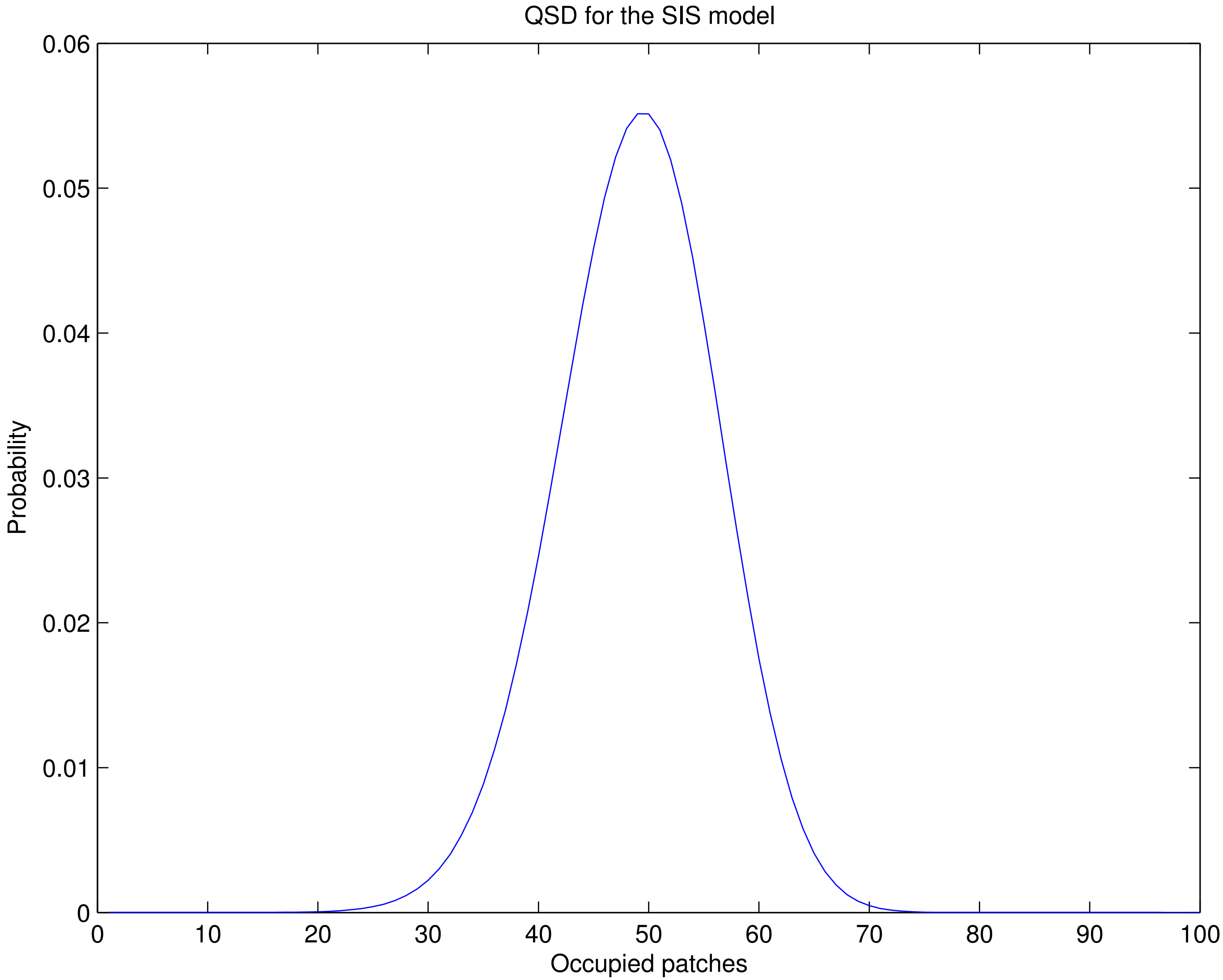
```
[nu,I]=max(real(diag(D)));
m=V(:,I);
pi=m/sum(m);

disp(pi');
   0.2350   0.2779   0.2605   0.1696   0.0570
```

The SIS model

Evaluate the quasi-stationary distribution for $n = 100$:

```
n=100; c=2; e=1; Q=zeros(n,n);
Q(1,2)=c*(1-1/n); Q(1,1)=-(c*(1-1/n)+e);
for i=2:(n-1)
  Q(i,i+1)=c*i*(1-i/n);
  Q(i,i-1)=e*i;
  Q(i,i)=-i*(c*(1-i/n)+e);
end
Q(n,n-1)=e*n; Q(n,n)=-e*n;
[V,D]=eig(Q’);
[nu,I]=max(real(diag(D)));
m=V(:,I); pi=m/sum(m);
plot(pi);
title(’QSD for the SIS model’);
xlabel(’Occupied patches’);
ylabel(’Probability’);
```

The SIS model



An epidemic model

Let $X(t) = (S(t), I(t))$, where $S(t)$ is the number of susceptibles at time t and $I(t)$ is the number of infectives at time t.

An epidemic model

Let $X(t) = (S(t), I(t))$, where $S(t)$ is the number of susceptibles at time t and $I(t)$ is the number of infectives at time t.

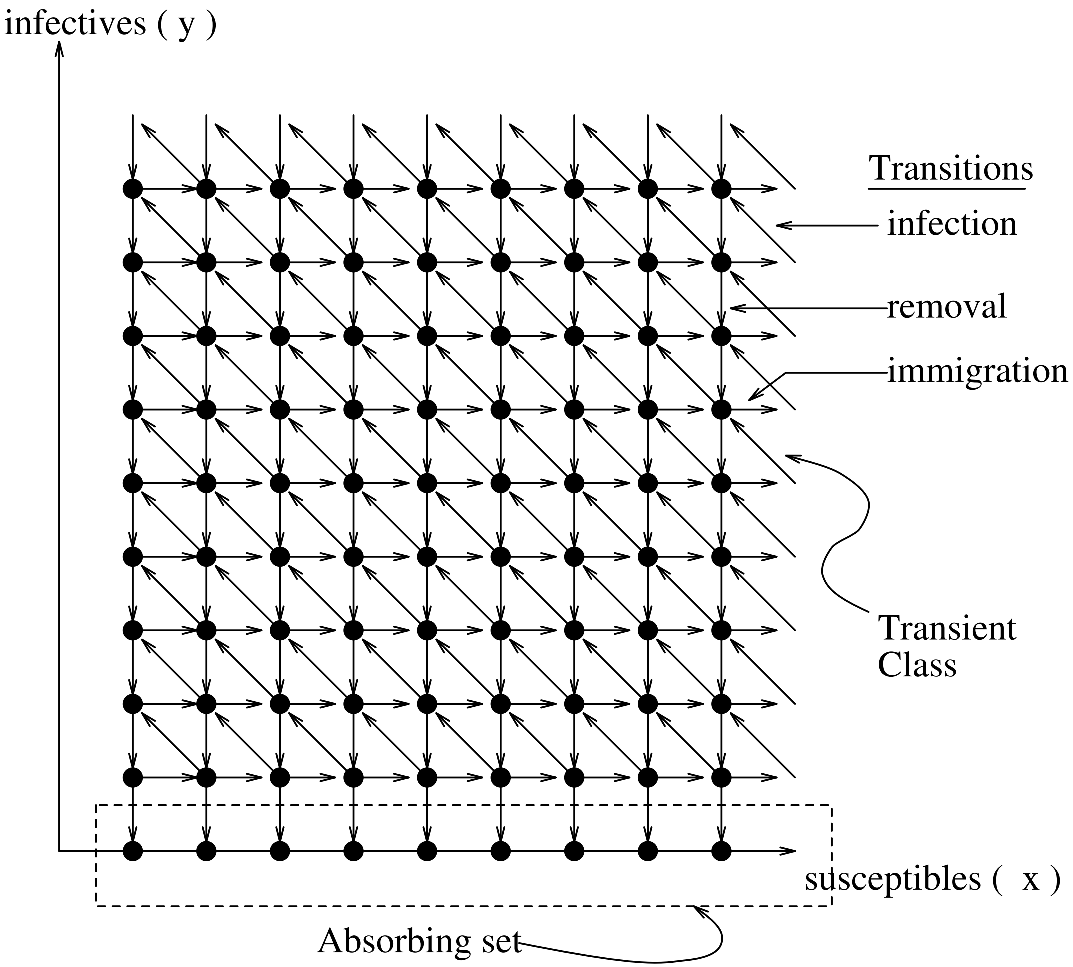
The state space is $S = \{(x, y) : x, y = 0, 1, \ldots\}$ and the transition rates are given by

$$q_{(x\ \ y),(x+1\ \ y)} = \alpha, \qquad q_{(x\ \ y),(x\ \ y-1)} = \gamma y,$$

$$q_{(x\ \ y),(x-1\ \ y+1)} = \beta x y,$$

where $\alpha, \gamma, \beta > 0$ are the immigration, removal and infection rates.

An epidemic model



An epidemic model

Clearly,

$$C = \{(x, y) : x = 0, 1, \ldots ; y = 1, 2, \ldots\}$$

is an irreducible transient class, and the abscissa is absorbing.

An epidemic model

Clearly,

$$C = \{(x,y) : x = 0, 1, \ldots; y = 1, 2, \ldots\}$$

is an irreducible transient class, and the abscissa is absorbing.

Ridler-Rowe (1967) proved that Q is regular (non-explosive) and absorption occurs with probability 1.

An epidemic model

Clearly,

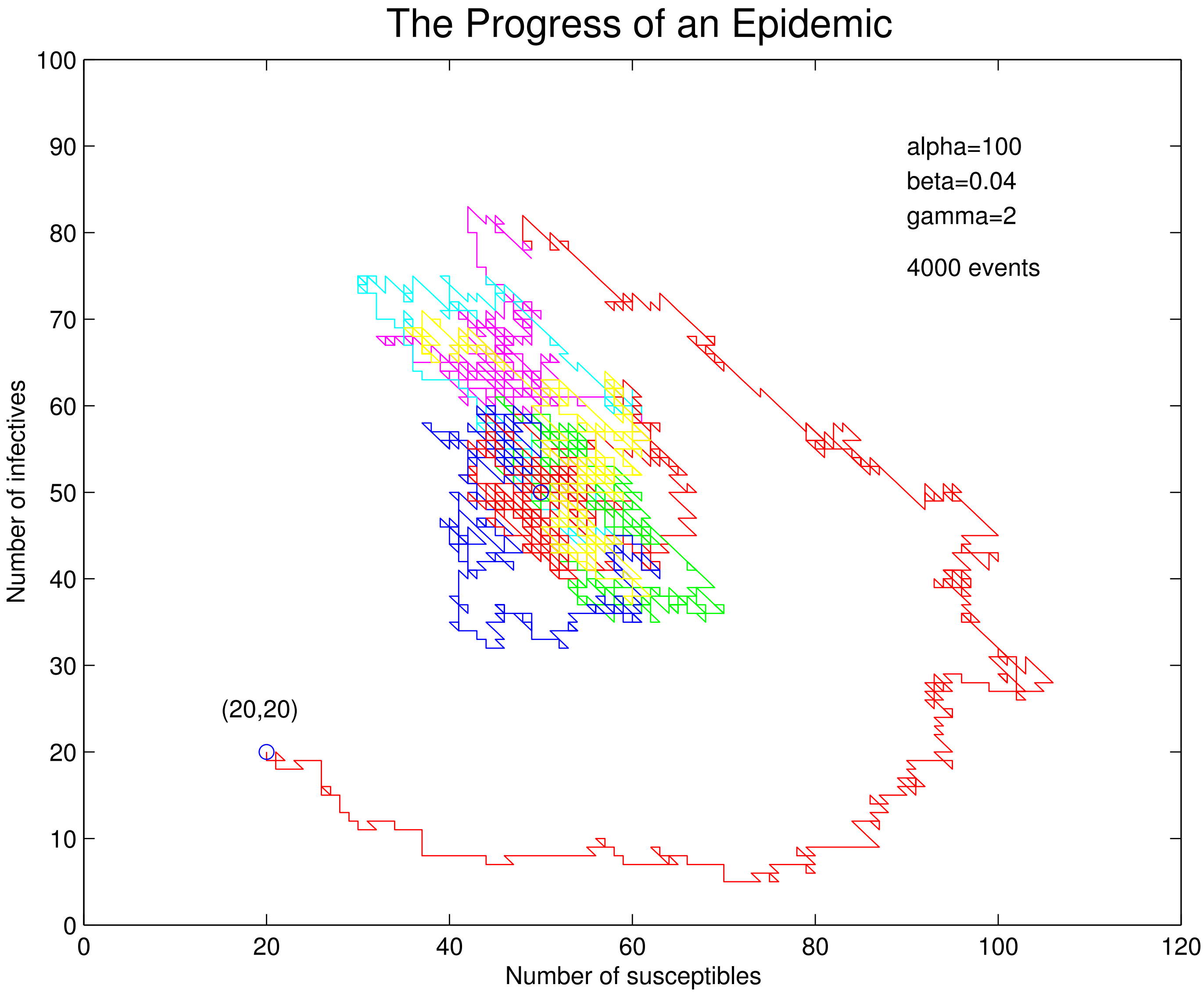
$$C = \{(x, y) : x = 0, 1, \ldots ; y = 1, 2, \ldots\}$$

is an irreducible transient class, and the abscissa is absorbing.

Ridler-Rowe (1967) proved that Q is regular (non-explosive) and absorption occurs with probability 1.

However, absorption is not observed over any reasonable time scale.

An epidemic model



Evaluate a QSD

We must solve

$$\pi_{(0\ y+1)}\gamma(y+1)+\pi_{(0\ y-1)}\beta(y-1)$$
$$=\pi_{(0\ y)}(\alpha+\gamma y-\lambda), \qquad y=1,2,\ldots$$

$$\pi_{(x-1\ y)}\alpha+\pi_{(x\ y+1)}\gamma(y+1)+\pi_{(x+1\ y-1)}\beta(x+1)(y-1)$$
$$=\pi_{(x\ y)}(\alpha+(\beta x+\gamma)y-\lambda),$$
$$x=1,2,\ldots;\ y=1,2,\ldots$$

for $(\pi_{(x\ y)},\ x=1,2,\ldots;\ y=1,2,\ldots)$, where $\lambda>0$.

Evaluate a QSD

We must solve

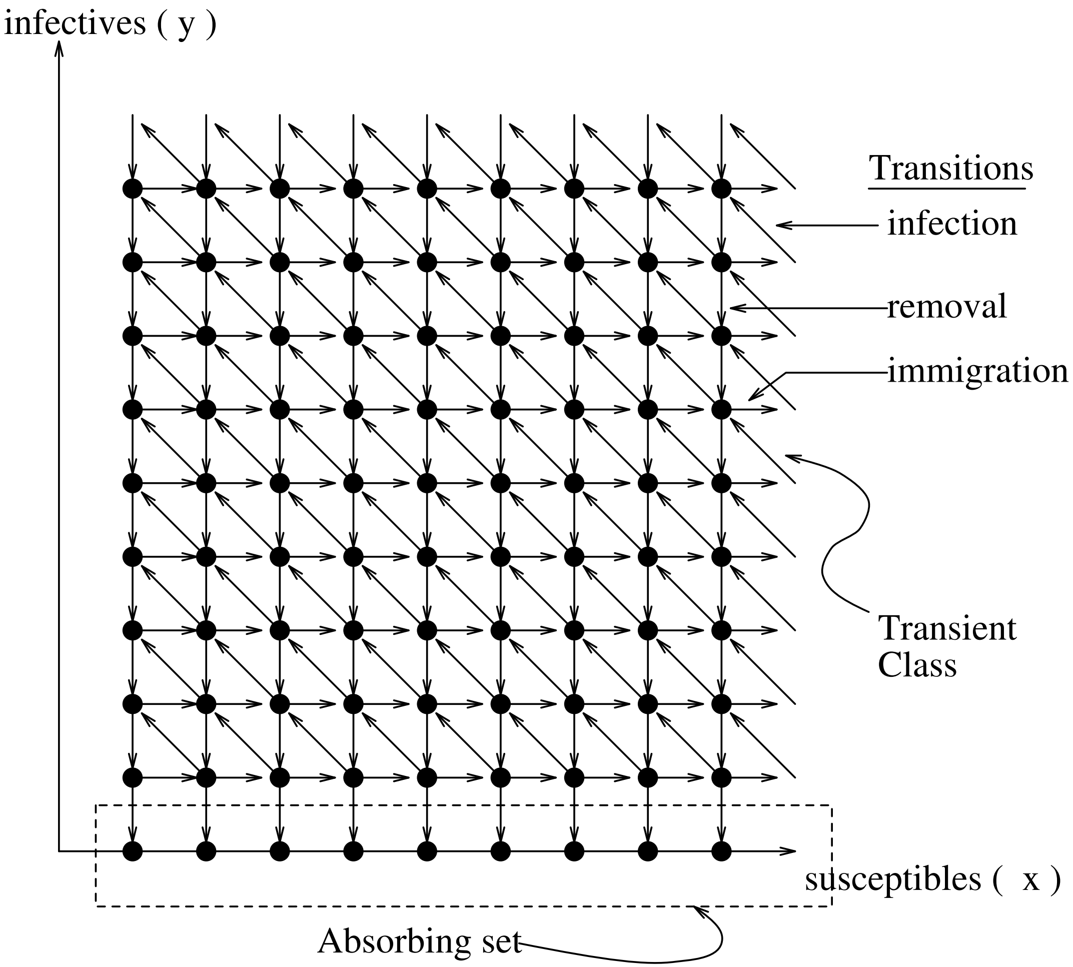
$$\pi_{(0\ y+1)}\gamma(y+1)+\pi_{(0\ y-1)}\beta(y-1)$$
$$=\pi_{(0\ y)}(\alpha+\gamma y-\lambda),\qquad y=1,2,\dots$$

$$\pi_{(x-1\ y)}\alpha+\pi_{(x\ y+1)}\gamma(y+1)+\pi_{(x+1\ y-1)}\beta(x+1)(y-1)$$
$$=\pi_{(x\ y)}(\alpha+(\beta x+\gamma)y-\lambda),$$
$$x=1,2,\dots;\ y=1,2,\dots$$

for $(\pi_{(x\ y)},\ x=1,2,\dots;\ y=1,2,\dots)$, where $\lambda>0$.

(In our dreams)

An epidemic model



How to evaluate the QSD

First truncate C to

$$C_N = \{(x, y) : x = 0, \ldots, N - 1; y = 1, \ldots, N\}$$

and restrict Q to C_N.

How to evaluate the QSD

First truncate C to

$$C_N = \{(x, y) : x = 0, \ldots, N-1; y = 1, \ldots, N\}$$

and restrict Q to C_N.

Then, use the transformation $i = y + Nx$ to convert the restricted transition matrix into an $n \times n$ matrix, $R = (q_{ij},\ i, j = 1, 2, \ldots, n)$, where $n = N^2$.

How to evaluate the QSD

First truncate C to

$$C_N = \{(x, y) : x = 0, \ldots, N-1; y = 1, \ldots, N\}$$

and restrict Q to C_N.

Then, use the transformation $i = y + Nx$ to convert the restricted transition matrix into an $n \times n$ matrix, $R = (q_{ij},\ i, j = 1, 2, \ldots, n)$, where $n = N^2$.

Construct a sequence $\{\pi^{(n)}\}$ of normalized eigenvectors and hope that this converges to the quasi-stationary distribution of the full epidemic model. (In practice, we choose N as large as possible.)

How to evaluate the QSD

Open questions

- Does a quasi-stationary distribution π exist for the epidemic model?

How to evaluate the QSD

Open questions

- Does a quasi-stationary distribution π exist for the epidemic model?
- Does a limiting-conditional distribution exist?

How to evaluate the QSD

Open questions

- Does a quasi-stationary distribution π exist for the epidemic model?
- Does a limiting-conditional distribution exist?
- Is C λ-positive recurrent?

How to evaluate the QSD

Open questions

- Does a quasi-stationary distribution π exist for the epidemic model?
- Does a limiting-conditional distribution exist?
- Is C λ-positive recurrent?
- Does $\{\pi^{(n)}\} \to \pi$?

How to evaluate the QSD

Open questions

- Does a quasi-stationary distribution π exist for the epidemic model?
- Does a limiting-conditional distribution exist?
- Is C λ-positive recurrent?
- Does $\{\pi^{(n)}\} \to \pi$?
- Pointwise? Or, only in the likelihood ratio sense?

How to evaluate the QSD

Implement the transformation $i = y + Nx$:

```
function i=index(st)
% (x,y) -> i
x=st(1); y=st(2); N=st(3);
i=y+x*N;
```

Implement the inverse transformation:

```
function state=state(index)
% i -> (x,y)
i=index(1); N=index(2);
x=fix((i-1)/N); y=i-N*x;
state=[x,y];
```

How to evaluate the QSD

Set up the truncated transition rate matrix and evaluate the dominant eigenvalue:

```
N=100; n=N^2;
a=1.0; b=4.0; c=2.0; alpha=a*N; beta=b/N; gamma=c;
R=zeros(n,n);
for x=0:(N-1)
  for y=1:N
    i=index([x,y,N]);
    if x<(N-1) R(i,index([x+1,y,N]))=alpha; end
    if ((x>0) & (y<N)) R(i,index([x-1,y+1,N]))=beta*x*y; end
    if y>1 R(i,index([x,y-1,N]))=gamma*y; end
    R(i,i)=-(alpha+(beta*x+gamma)*y);
  end
end
[V,D]=eig(R');
[nu,in]=max(real(diag(D))); m=V(:,in);
```

Preliminary numerical results

For $N = 92$ we get

```
??? Error using ==> zeros
Out of memory. Type HELP MEMORY for your options.

Error in ==> C:\docs\talks\UQ2004c\quasi.m
On line 3  ==> R=zeros(n,n);
```

For $N = 70$ we get

```
??? Error using ==> eig
Out of memory. Type HELP MEMORY for your options.

Error in ==> C:\docs\talks\UQ2004c\quasi.m
On line 13  ==> [V,D]=eig(R');
```

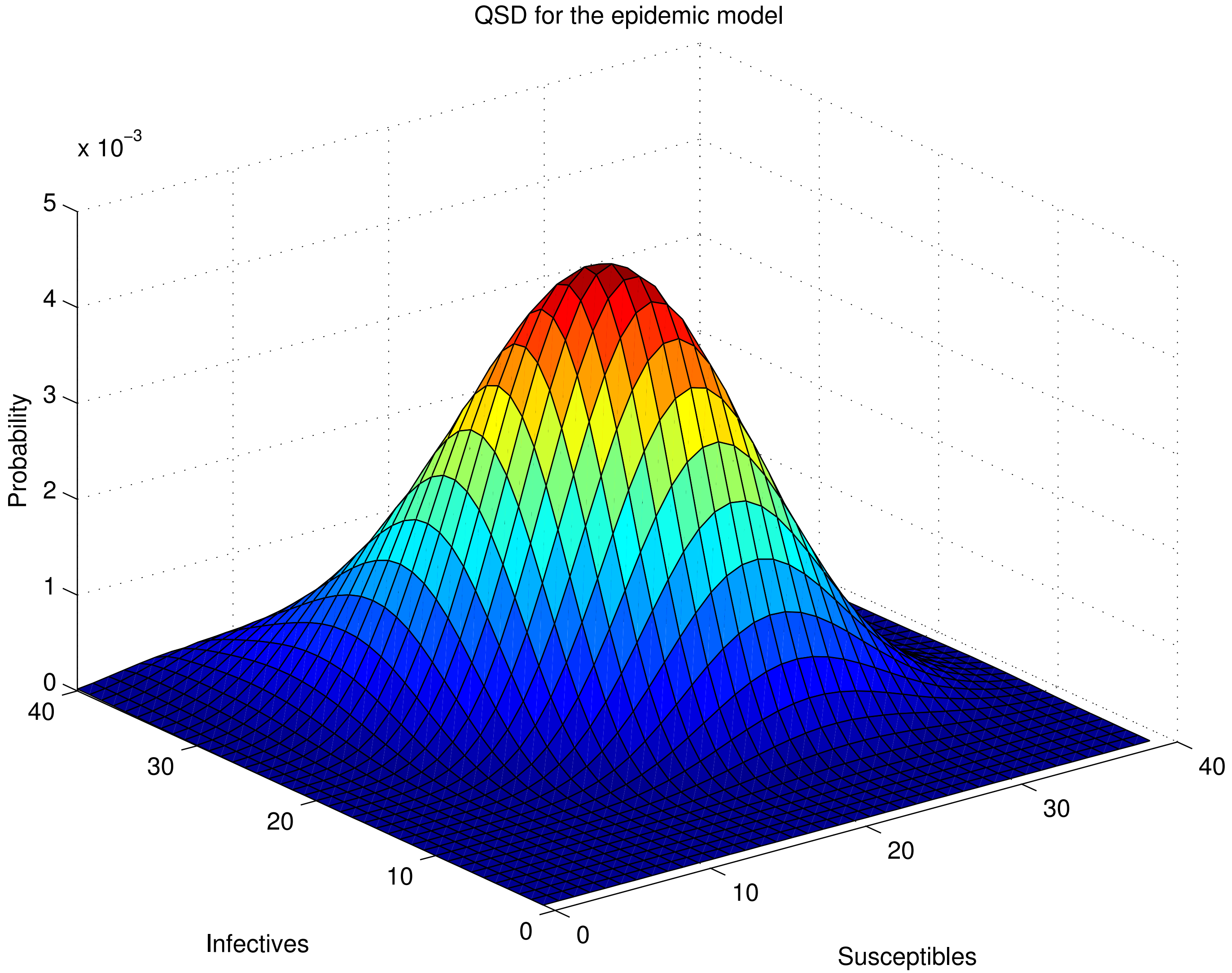
How to evaluate the QSD

Normalize the dominant eigenvalue and transform its support back to two dimensions:

```
pi0=m/sum(m);
for x=0:(N-1)
  for y=1:N
    i=index([x,y,N]);
    pi1(x+1,y)=pi0(i);
  end
end
surf(0:(N-1),1:N,pi1)
title('QSD for the epidemic model');
xlabel('Susceptibles');
ylabel('Infectives');
zlabel('Probability');
```

For $N = 40$ it took about 20 minutes to produce the graph.

The SIS model



How to evaluate the QSD

Recall that we restricted Q to

$$C_N = \{(x, y) : x = 0, \ldots, N-1; y = 1, \ldots, N\}$$

and then used the transformation $i = y + Nx$ to convert this to an $n \times n$ matrix, $R = (q_{ij},\ i, j = 1, 2, \ldots, n)$, where $n = N^2$.

How to evaluate the QSD

Recall that we restricted Q to

$$C_N = \{(x, y) : x = 0, \ldots, N-1; y = 1, \ldots, N\}$$

and then used the transformation $i = y + Nx$ to convert this to an $n \times n$ matrix, $R = (q_{ij},\, i, j = 1, 2, \ldots, n)$, where $n = N^2$.

Numerical evaluation of the eigenvectors of R is obviously a non-trivial problem when N is large.

How to evaluate the QSD

Recall that we restricted Q to

$$C_N = \{(x, y) : x = 0, \ldots, N - 1; y = 1, \ldots, N\}$$

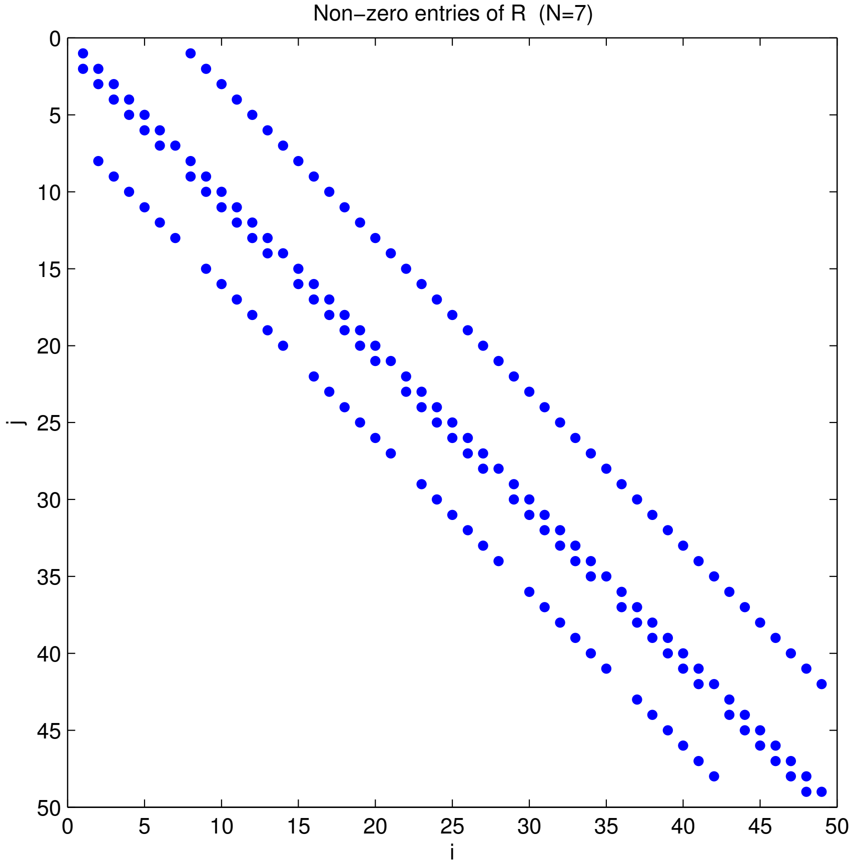
and then used the transformation $i = y + Nx$ to convert this to an $n \times n$ matrix, $R = (q_{ij},\ i, j = 1, 2, \ldots, n)$, where $n = N^2$.

Numerical evaluation of the eigenvectors of R is obviously a non-trivial problem when N is large.

For example, if $N = 100$, that is $n = 10^4$, so that Q has 10^8 entries, we would need 400 Mbytes of storage to even store Q, let alone evaluate its eigenvectors.

How to evaluate the QSD

R is a sparse matrix: the number of of non-zero entries of R is $(2N-1)^2$ and so the proportion is $O(1/N^2) = O(1/n)$.



The Arnoldi Method

We need to solve $Ax = \nu x$ (at least for dominant eigenvectors), where A is $n \times n$ and n is large.

The Arnoldi Method

We need to solve $Ax = \nu x$ (at least for dominant eigenvectors), where A is $n \times n$ and n is large.

Using an initial estimate of x, the basic Arnoldi method produces an $m \times m$ (upper-Hessenberg) matrix H_m and an $n \times m$ matrix V_m with

$$V_m^T A V_m = H_m.$$

The Arnoldi Method

We need to solve $Ax = \nu x$ (at least for dominant eigenvectors), where A is $n \times n$ and n is large.

Using an initial estimate of x, the basic Arnoldi method produces an $m \times m$ (upper-Hessenberg) matrix H_m and an $n \times m$ matrix V_m with

$$V_m^T A V_m = H_m.$$

It has the property that if z_m is an eigenvector of H_m, then, for m large, $V_m z_m$ is close to an eigenvector of A.

The Arnoldi Method

We need to solve $Ax = \nu x$ (at least for dominant eigenvectors), where A is $n \times n$ and n is large.

Using an initial estimate of x, the basic Arnoldi method produces an $m \times m$ (upper-Hessenberg) matrix H_m and an $n \times m$ matrix V_m with

$$V_m^T A V_m = H_m.$$

It has the property that if z_m is an eigenvector of H_m, then, for m large, $V_m z_m$ is close to an eigenvector of A.

We solve for z_m using standard (dense-matrix) methods. For example, n might be $100,000$ and m might be 20.

The Basic Arnoldi Method

The basic Arnoldi method starts with an arbitrary “seed” vector $v_1 \in \mathbb{R}^n$ from which a sequence, $v_1, v_2, \ldots$, of orthonormal vectors is constructed as follows.

The Basic Arnoldi Method

The basic Arnoldi method starts with an arbitrary "seed" vector $v_1 \in \mathbb{R}^n$ from which a sequence, $v_1, v_2, \ldots$, of orthonormal vectors is constructed as follows.

First, the vector $w_1 = Av_1$ is computed.

The Basic Arnoldi Method

The basic Arnoldi method starts with an arbitrary “seed” vector $v_1 \in \mathbb{R}^n$ from which a sequence, $v_1, v_2, \ldots$, of orthonormal vectors is constructed as follows.

First, the vector $w_1 = Av_1$ is computed. Then, the components of w_1 in the direction of v_1 are subtracted to give the “residual” $r_1 = w_1 - (v_1^T w_1)v_1$.

The Basic Arnoldi Method

The basic Arnoldi method starts with an arbitrary “seed” vector $v_1 \in \mathbb{R}^n$ from which a sequence, $v_1, v_2, \ldots$, of orthonormal vectors is constructed as follows.

First, the vector $w_1 = Av_1$ is computed. Then, the components of w_1 in the direction of v_1 are subtracted to give the “residual” $r_1 = w_1 - (v_1^T w_1)v_1$. This vector is normalized, using the Euclidean norm to form v_2: $v_2 = r_1/\|r_1\|_2$.

The Basic Arnoldi Method

The basic Arnoldi method starts with an arbitrary “seed” vector $v_1 \in \mathbb{R}^n$ from which a sequence, $v_1, v_2, \ldots$, of orthonormal vectors is constructed as follows.

First, the vector $w_1 = Av_1$ is computed. Then, the components of w_1 in the direction of v_1 are subtracted to give the “residual” $r_1 = w_1 - (v_1^T w_1)v_1$. This vector is normalized, using the Euclidean norm to form v_2: $v_2 = r_1/\|r_1\|_2$. Next, $w_2 = Av_2$ is computed and then the components of w_2 in the directions of v_1 and v_2 are subtracted to give the second residual r_2.

The Basic Arnoldi Method

The basic Arnoldi method starts with an arbitrary “seed” vector $v_1 \in \mathbb{R}^n$ from which a sequence, $v_1, v_2, \ldots,$ of orthonormal vectors is constructed as follows.

First, the vector $w_1 = Av_1$ is computed. Then, the components of w_1 in the direction of v_1 are subtracted to give the “residual” $r_1 = w_1 - (v_1^T w_1)v_1$. This vector is normalized, using the Euclidean norm to form v_2: $v_2 = r_1/\|r_1\|_2$. Next, $w_2 = Av_2$ is computed and then the components of w_2 in the directions of v_1 and v_2 are subtracted to give the second residual r_2. This is normalized to give v_3, and so on.

The Basic Arnoldi Method

The procedure described gives

```
**for** $j = 1, 2, \ldots \{$
    $w_j \leftarrow A v_j$
    $h_{ij} \leftarrow v_i^T w_j$        (**for** $i = 1, 2, \ldots, j$)
    $r_j \leftarrow w_j - \sum_{i=1}^{j} h_{ij}\, v_i$
    $v_{j+1} \leftarrow r_j / \|r_j\|_2$
    $h_{j+1,j} \leftarrow \|r_j\|_2$
    $\}$
```

The Basic Arnoldi Method

If the procedure is halted at say $j = m$, then we shall have that

$$Av_k = \begin{cases} \sum_{i=1}^{k+1} h_{ik} v_i & \text{for } k < m \\ \sum_{i=1}^{m} h_{ik} v_i + r_m & \text{for } k = m. \end{cases}$$

The Basic Arnoldi Method

If the procedure is halted at say $j = m$, then we shall have that

$$Av_k = \begin{cases} \sum_{i=1}^{k+1} h_{ik} v_i & \text{for } k < m \\ \sum_{i=1}^{m} h_{ik} v_i + r_m & \text{for } k = m. \end{cases}$$

Here $H_m = (h_{ij})$ is an $m \times m$ upper-Hessenberg matrix given by

$$h_{ij} = \begin{cases} v_i^T w_j & \text{for } i = 1, 2, \ldots, j \\ \|r_j\|_2 & \text{for } i = j + 1 \\ 0 & \text{otherwise.} \end{cases}$$

The Basic Arnoldi Method

If the procedure is halted at say $j = m$, then we shall have that

$$Av_k = \begin{cases} \sum_{i=1}^{k+1} h_{ik} v_i & \text{for } k < m \\ \sum_{i=1}^{m} h_{ik} v_i + r_m & \text{for } k = m. \end{cases}$$

Thus, if we let $V_m = [v_1, v_2, \ldots, v_m]$ (columns), then

$$AV_m = V_m H_m + r_m e_m^T,$$

where e_m is the unit vector with a 1 as its m^{th} entry, and so, since the columns of V_m are orthonormal and r_m is orthogonal to each of them, we deduce that $V_m^T A V_m = H_m$.

The Basic Arnoldi Method

Claim. If z_m is an eigenvector of H_m, then, for m sufficiently large, $V_m z_m$ should be close to an eigenvector of A.

The Basic Arnoldi Method

Claim. If z_m is an eigenvector of H_m, then, for m sufficiently large, $V_m z_m$ should be close to an eigenvector of A.

Suppose that z_m satisfies $H_m z_m = \hat{\nu}_m z_m$, for some $\hat{\nu}_m$, and let $x_m = V_m z_m$. Then, on multiplying $AV_m = V_m H_m + r_m e_m^T$ (just obtained) by z_m, we get

$$\begin{aligned} Ax_m &= V_m(H_m z_m) + r_m(z_m)_m = V_m(\hat{\nu}_m z_m) + r_m(z_m)_m \\ &= \hat{\nu}_m x_m + r_m(z_m)_m. \end{aligned}$$

The Basic Arnoldi Method

Claim. If z_m is an eigenvector of H_m, then, for m sufficiently large, $V_m z_m$ should be close to an eigenvector of A.

Suppose that z_m satisfies $H_m z_m = \hat{\nu}_m z_m$, for some $\hat{\nu}_m$, and let $x_m = V_m z_m$. Then, on multiplying $AV_m = V_m H_m + r_m e_m^T$ (just obtained) by z_m, we get

$$
\begin{aligned}
Ax_m &= V_m(H_m z_m) + r_m(z_m)_m = V_m(\hat{\nu}_m z_m) + r_m(z_m)_m \\
&= \hat{\nu}_m x_m + r_m(z_m)_m.
\end{aligned}
$$

Thus, $(A - E_m)x_m = \hat{\nu}_m x_m$, where E_m is given by

$$E_m = r_m(z_m)_m x_m^T / \|x_m\|_2^2.$$

The Basic Arnoldi Method

$(A - E_m)x_m = \hat{\nu}_m x_m$, where E_m is given by

$$E_m = r_m (z_m)_m x_m^T / \|x_m\|_2^2.$$

It follows, from standard sensitivity analysis (see, for example, Section 7.2 of Golub and Van Loan*), that the error in the eigenvalue can be estimated by $\|r_m\|_2 \, |(z_m)_m| / \|x_m\|_2$.

*Golub, G.H. and Van Loan, C. (1996) Matrix Computations, 3rd Edition, John Hopkins Press.

The Basic Arnoldi Method

$(A - E_m)x_m = \hat{\nu}_m x_m$, where E_m is given by

$$E_m = r_m (z_m)_m x_m^T / \|x_m\|_2^2.$$

It follows, from standard sensitivity analysis (see, for example, Section 7.2 of Golub and Van Loan*), that the error in the eigenvalue can be estimated by $\|r_m\|_2 \, |(z_m)_m| / \|x_m\|_2$.

*Golub, G.H. and Van Loan, C. (1996) Matrix Computations, 3rd Edition, John Hopkins Press.

Hence, if the residual vector r_m is small or $|(z_m)_m|$ is small, then the approximation will be good.

Which eigenvectors does it give?

For simplicity, suppose A is symmetric, so that all its eigenvalues are real. The Arnoldi method reduces to the Lanczos method, and H is a symmetric (of necessity tridiagonal) matrix.

Which eigenvectors does it give?

For simplicity, suppose A is symmetric, so that all its eigenvalues are real. The Arnoldi method reduces to the Lanczos method, and H is a symmetric (of necessity tridiagonal) matrix. However, let $V_m = [v_1, v_2, \ldots, v_m]$, where $v_1, v_2, \ldots$ is any sequence of orthonormal vectors in $\mathbb{R}^n$, and let $H_m = V_m^T A V_m$.

Which eigenvectors does it give?

For simplicity, suppose A is symmetric, so that all its eigenvalues are real. The Arnoldi method reduces to the Lanczos method, and H is a symmetric (of necessity tridiagonal) matrix. However, let $V_m = [v_1, v_2, \ldots, v_m]$, where $v_1, v_2, \ldots$ is any sequence of orthonormal vectors in $\mathbb{R}^n$, and let $H_m = V_m^T A V_m$. Let $\nu_1(A)$ and $\nu_n(A)$ be the maximum and minimum eigenvalues of A.

Which eigenvectors does it give?

For simplicity, suppose A is symmetric, so that all its eigenvalues are real. The Arnoldi method reduces to the Lanczos method, and H is a symmetric (of necessity tridiagonal) matrix. However, let $V_m = [v_1, v_2, \ldots, v_m]$, where $v_1, v_2, \ldots$ is any sequence of orthonormal vectors in $\mathbb{R}^n$, and let $H_m = V_m^T A V_m$. Let $\nu_1(A)$ and $\nu_n(A)$ be the maximum and minimum eigenvalues of A.

Claim. $\nu_n(A) \leq \nu_m(H_m) \leq \nu_1(H_m) \leq \nu_1(A)$.

Which eigenvectors does it give?

For simplicity, suppose A is symmetric, so that all its eigenvalues are real. The Arnoldi method reduces to the Lanczos method, and H is a symmetric (of necessity tridiagonal) matrix. However, let $V_m = [v_1, v_2, \ldots, v_m]$, where $v_1, v_2, \ldots$ is any sequence of orthonormal vectors in $\mathbb{R}^n$, and let $H_m = V_m^T A V_m$. Let $\nu_1(A)$ and $\nu_n(A)$ be the maximum and minimum eigenvalues of A.

Claim. $\nu_n(A) \leq \nu_m(H_m) \leq \nu_1(H_m) \leq \nu_1(A)$.

The proof uses the fact that the Rayleigh quotient $r(x) = x^T Ax / x^T x$, $x \neq 0$, is maximized (resp. minimized) by the maximum and (resp. minimum) eigenvalue of A.

Which eigenvectors does it give?

Next, it can be shown that $\nu_{m+1}(H_{m+1}) < \nu_m(H_m)$ and $\nu_1(H_m) < \nu_1(H_{m+1})$ (that is, we move closer to the maximum and minimum eigenvalues of A) if

$$\text{span}\{v_1, v_2, v_3, \ldots, v_k\} = \text{span}\{v_1, Av_1, A^2v_1, \ldots, A^kv_1\}$$

for both $k = m$ and $k = m + 1$. The Arnoldi method (Lanczos method) achieves this.

Which eigenvectors does it give?

Next, it can be shown that $\nu_{m+1}(H_{m+1}) < \nu_m(H_m)$ and $\nu_1(H_m) < \nu_1(H_{m+1})$ (that is, we move closer to the maximum and minimum eigenvalues of A) if

$$\text{span}\{v_1, v_2, v_3, \ldots, v_k\} = \text{span}\{v_1, Av_1, A^2v_1, \ldots, A^kv_1\}$$

for both $k = m$ and $k = m + 1$. The Arnoldi method (Lanczos method) achieves this. Note that

$$\mathcal{K}(A, v, m) = \text{span}\{v, Av, A^2v, \ldots, A^mv\}, \qquad m = 1, 2, \ldots, n,$$

are called the Krylov subspaces of A generated by v.

Which eigenvectors does it give?

Next, it can be shown that $\nu_{m+1}(H_{m+1}) < \nu_m(H_m)$ and $\nu_1(H_m) < \nu_1(H_{m+1})$ (that is, we move closer to the maximum and minimum eigenvalues of A) if

$$\text{span}\{v_1, v_2, v_3, \ldots, v_k\} = \text{span}\{v_1, Av_1, A^2v_1, \ldots, A^kv_1\}$$

for both $k = m$ and $k = m + 1$. The Arnoldi method (Lanczos method) achieves this. Note that

$$\mathcal{K}(A, v, m) = \text{span}\{v, Av, A^2v, \ldots, A^mv\}, \qquad m = 1, 2, \ldots, n,$$

are called the Krylov subspaces of A generated by v. The Arnoldi method provides a means of computing a set of orthonormal bases for these subspaces.

Some properties

The most important property of the Arnoldi method is:

Some properties

The most important property of the Arnoldi method is:

- The Arnoldi method **does not work**.

Some properties

The most important property of the Arnoldi method is:

- The Arnoldi method **does not work.**

Why? The algorithm is vulnerable to round off error: in particular, loss of orthogonality of the columns of V_m.

Some properties

The most important property of the Arnoldi method is:

- The Arnoldi method **does not work**.

Why? The algorithm is vulnerable to round off error: in particular, loss of orthogonality of the columns of V_m.

This was addressed in

Pollett, P.K. and Stewart, D.E. (1994) An efficient procedure for computing quasistationary distributions of Markov chains with sparse transition structure. *Advances in Applied Probability* 26, 68–79.

The Iterative Arnoldi Method

Take m small (we found that $m = 20$ worked best). Then, using an initial estimate v_1 of the eigenvector x, apply the Basic Arnoldi Method (to obtain H_m and V_m) and set $\hat{\nu}$ to be the dominant eigenvalue of H_m if this is real, or set $\hat{\nu}$ equal to zero otherwise.

The Iterative Arnoldi Method

Take m small (we found that $m = 20$ worked best). Then, using an initial estimate v_1 of the eigenvector x, apply the Basic Arnoldi Method (to obtain H_m and V_m) and set $\hat{\nu}$ to be the dominant eigenvalue of H_m if this is real, or set $\hat{\nu}$ equal to zero otherwise.

Now solve

$$(H_m - \hat{\nu} I) u_1 = z$$

with z chosen at random and repeat the procedure with a new initial estimate, given by

$$v_1 = V_m u_1 / \|V_m u_1\|_2.$$

The Iterative Arnoldi Method

Take m small (we found that $m = 20$ worked best). Then, using an initial estimate v_1 of the eigenvector x, apply the Basic Arnoldi Method (to obtain H_m and V_m) and set $\hat{\nu}$ to be the dominant eigenvalue of H_m if this is real, or set $\hat{\nu}$ equal to zero otherwise.

Now solve

$$(H_m - \hat{\nu} I) u_1 = z$$

with z chosen at random and repeat the procedure with a new initial estimate, given by

$$v_1 = V_m u_1 / \|V_m u_1\|_2.$$

Continue until the residual $\|Av_1 - \hat{\nu} v_1\|_2$ is sufficiently small.

The Iterative Arnoldi Method

An explanation of why this works is that the computed $\hat{u}_1$ is an exact solution of a perturbed system

$$(H_m + E - \hat{\lambda} I)\hat{u}_1 = z,$$

where $\|E\|_2 \approx c_m \boldsymbol{u} \|H_m - \hat{\lambda} I\|_2$, $\{c_m\}$ is a sequence of constants that grows slowly and $\boldsymbol{u}$ is the “machine epsilon” or “unit roundoff” for the arithmetic used; see Section 3.3 of Golub and Van Loan.

In Matlab use `eigs` instead of `eig`

Replace the command

```
R=zeros(n,n);
```

by

```
R=sparse([]);
```

Replace the commands

```
[V,D]=eig(R’); [nu,I]=max(real(diag(D))); m=V(:,I);
```

by

```
[m,nu,FL]=eigs(R’,1,’lr’);
if FL==1 disp(’ Warning - did not converge’); end
```

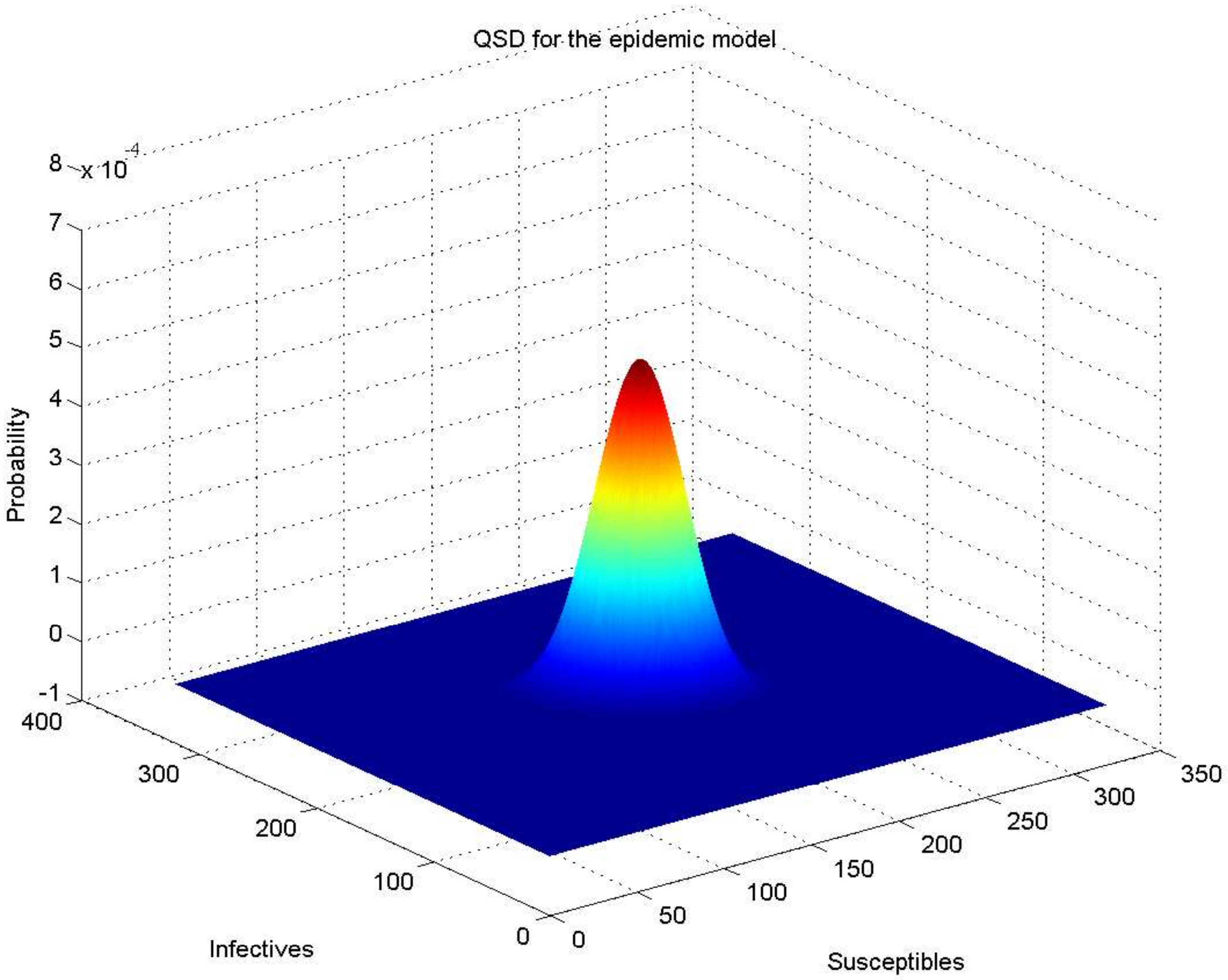
There are many other options, including the ability to control the value of m.

The Arnoldi Method

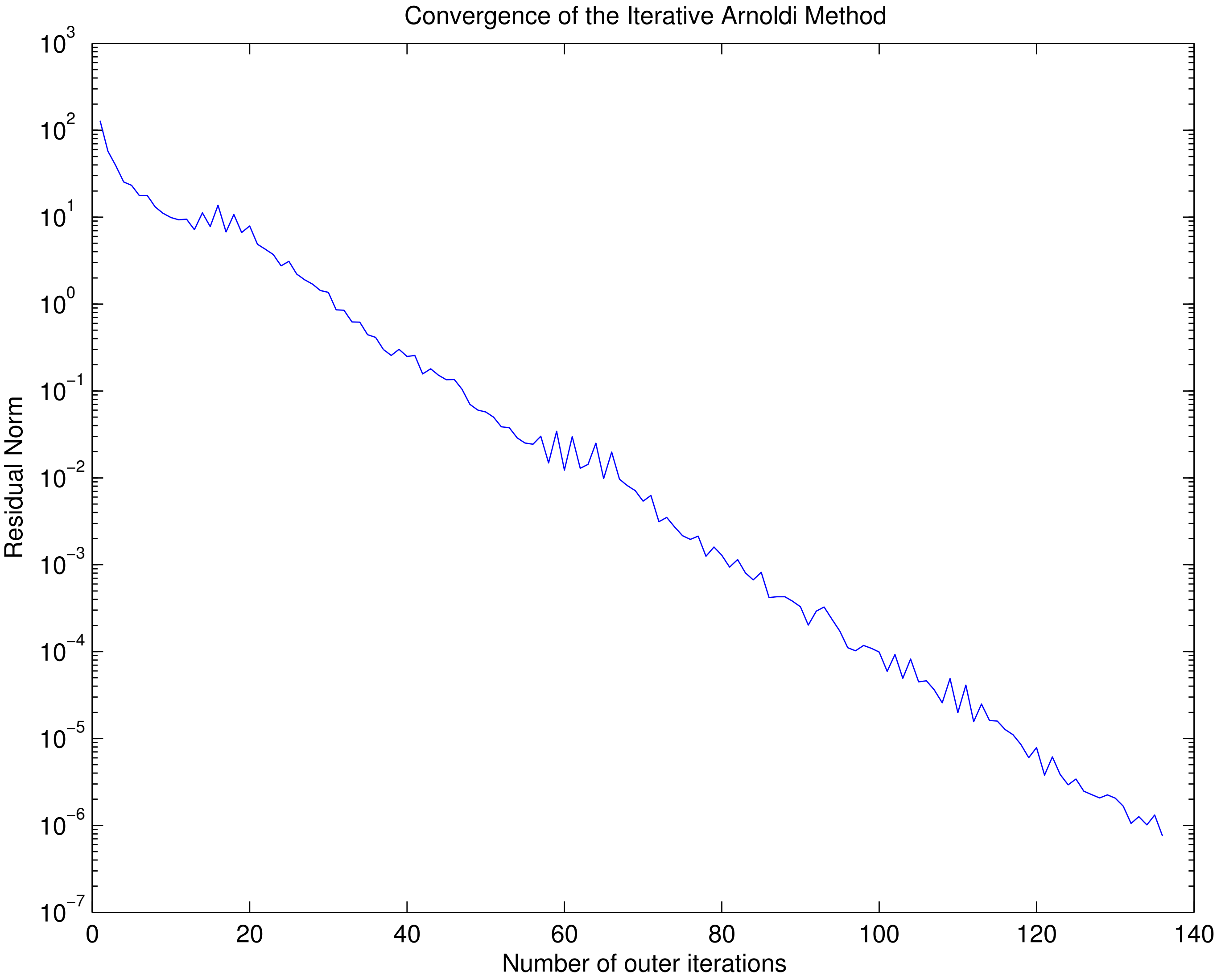
For $N = 320$ my code successfully evaluated the quasi-stationary distribution in about 40 minutes (the iterative Arnoldi method converged).

Remember that the system had $102,400$ states!

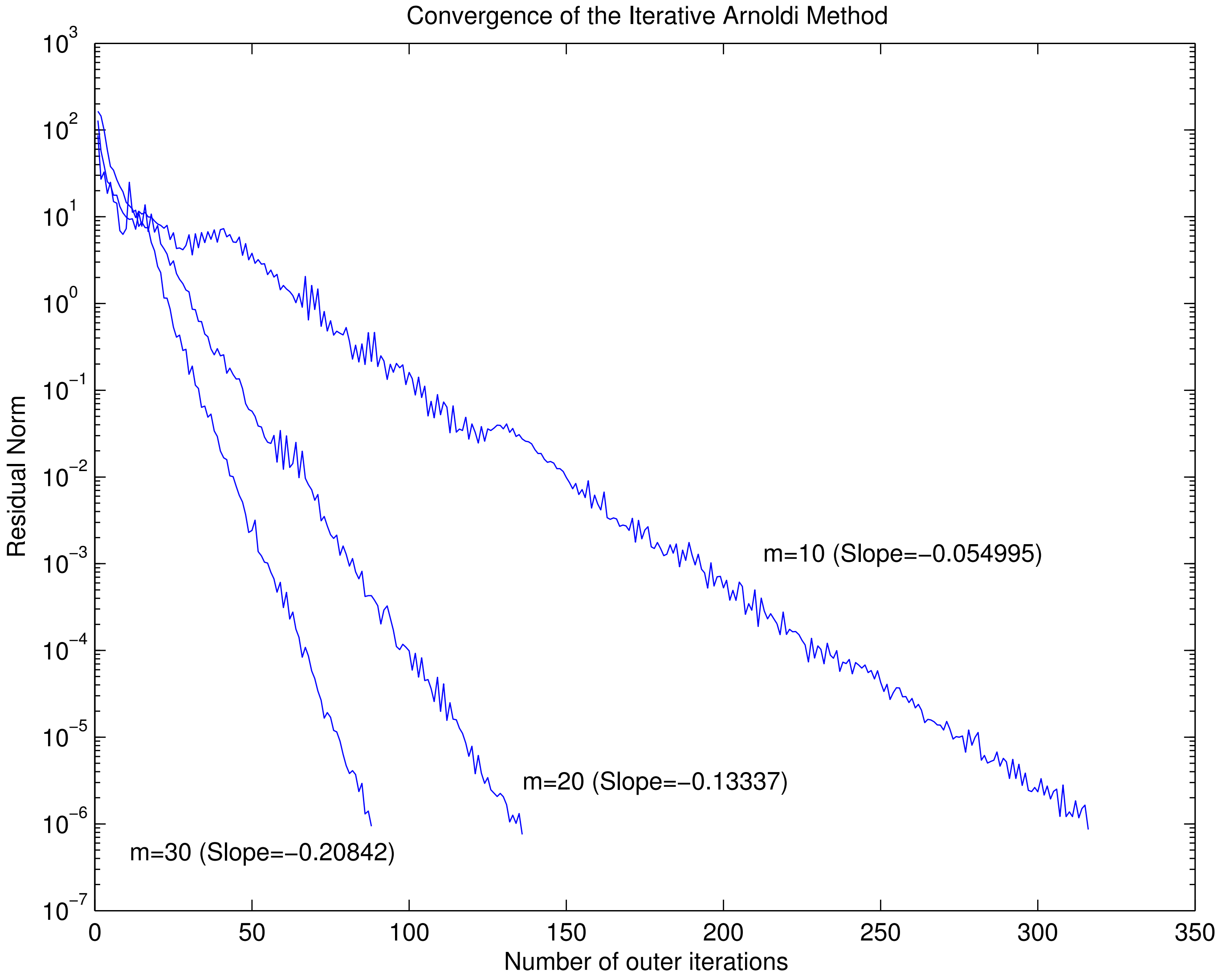
The Arnoldi Method



Convergence



Convergence


Convergence of the Iterative Arnoldi Method
Residual Norm
Number of outer iterations
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
m=10 (Slope=−0.054995)
m=20 (Slope=−0.13337)
m=30 (Slope=−0.20842)