A numerical search for intertwining relations

Jan M. Swart

Singapore, August 13, 2024

Work in progress!

Intertwining of matrices

A relation between square matrices of the form

$$PK = KQ \qquad (\star)$$

is called an *intertwining relation* and K is the *intertwiner*.
If K is invertible, then we can rewrite $(\star)$ as

$$Q = K^{-1}PK \quad \text{or} \quad P = KQK^{-1}.$$

This says that P and Q are *similar*.

By definition, P is diagonalisable if
K can be chosen such that Q is diagonal.

Probability kernels and Markov chains

An $d \times d$ matrix P is a *probability kernel* if

(i) $P(x,y) \geq 0 \qquad (1 \leq x, y \leq d)$,

(ii) $\displaystyle\sum_{y=1}^{d} P(x,y) = 1 \qquad (1 \leq x \leq d)$.

Its n-th power P^n describes the n-step transition probabilities of the Markov chain with transition kernel P.

If we can diagonalise P, then we have good control over its powers, since

$$P^n = KQ^nK^{-1} \qquad (t \geq 0),$$

and it is trivial to calculate the n-th power of a diagonal matrix.

Intertwining of probability kernels

In practice, it can be hard to have good control over the eigenvalues of P and the intertwiner K that diagonalises P.

Also, by diagonalising P, we leave the space of probability kernels, so in a sense we forget about the special property that P is a probability kernel (in particular, the nonnegativity of its elements).

In view of this, as an alternative to diagonalisation, we can look for intertwining relations of the form

$$\text{(i)}\ PK = KQ \quad \text{or} \quad \text{(ii)}\ KP = QK,$$

where P, Q, K are all probability kernels, and Q is "as simple as possible".

Intertwining of probability kernels

Note that

$$\left.\begin{array}{llll} \text{(i)} \ PK = KQ & \quad \text{implies} \quad & P^n K = KQ^n \\ \text{(ii)} \ KP = QK & \quad \text{implies} \quad & KP^n = Q^n K \end{array}\right\} \qquad (n \geq 0).$$

In case (i), let's say that Q is intertwined *on top of P* and in case (ii), let's say that Q is intertwined *below P*.

$PK = KQ$ implies $K^{-1}P = QK^{-1}$,
but K^{-1} is in general not a probability kernel.

Continuous time

A *Markov semigroup* is a family $(P_t)_{t\geq 0}$ of square probability kernels such that $t \mapsto P_t$ is continuous,
$P_0 = 1$, and $P_s P_t = P_{s+t}$ $(s, t \geq 0)$.

Each Markov semigroup is of the form

$$P_t = e^{tG} := \sum_{k=0}^{\infty} \frac{1}{k!} t^k G^k,$$

where the *generator* G satisfies

(i) $G(x, y) \geq 0 \quad \forall x \neq y$,

(ii) $\sum_{y=1}^{n} G(x, y) = 0 \quad \forall x$.

For $x \neq y$, we call $G(x, y)$ the *rate* of jumps from x to y.

Intertwining of probability kernels

For semigroups $(P_t)_{t\geq 0}$ and $(Q_t)_{t\geq 0}$ with generators G, H, one has

$$\left.\begin{array}{llll} \text{(i)} & GK = KH & \text{implies} & P_t K = K Q_t \\ \text{(ii)} & KG = HK & \text{implies} & K P_t = Q_t K \end{array}\right\} \qquad (t \geq 0).$$

In case (i), we say that H is intertwined *on top of* G and in case (ii), we say that H is intertwined *below* G.

Intertwining and coupling

Assume that Q is intertwined on top of P, i.e., $PK = KQ$.

Then it is possible to construct a Markov chain $(X_n, Y_n)_{n\geq 0}$ such that

$$\mathbb{P}\big[Y_0 \in \cdot\,\big|\,X_0\big] = K(X_0,\,\cdot\,) \quad \text{a.s.}$$

implies that

$$\mathbb{P}\big[Y_n \in \cdot\,\big|\,(X_k)_{0\leq k\leq n}\big] = K(X_n,\,\cdot\,) \quad \text{a.s.} \quad (n \geq 0).$$

Moreover (note that Y is *autonomous* but X is not):

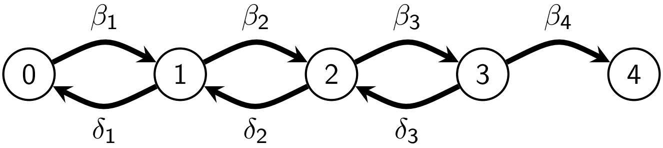
$$\begin{aligned}\mathbb{P}\big[X_{n+1} = x\big|\,(X_k)_{0\leq k\leq n}\big] &= P(X_n, x) \quad \text{a.s.},\\ \mathbb{P}\big[Y_{n+1} = y\big|\,(X_k, Y_k)_{0\leq k\leq n}\big] &= Q(Y_n, y) \quad \text{a.s.}\end{aligned}$$

An analogue result holds on the continuous-time case.
[Rogers & Pitman '81, Fill '92]

Birth-and-death chains

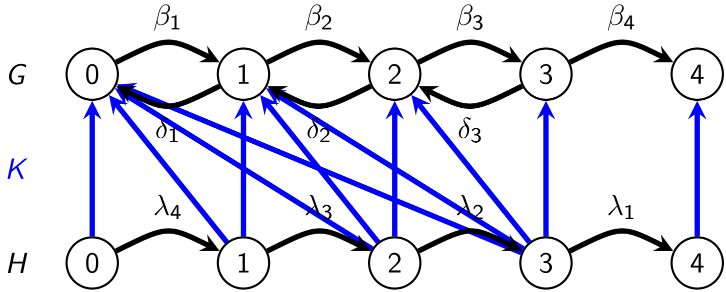
Consider a continuous-time process on $\{0, \ldots, d\}$ that jumps with rate β_k from $k-1$ to k and
with rate δ_k from k to $k-1$ $(1 \leq k \leq d)$.

Assume that $\delta_d = 0$ so that d is a trap.



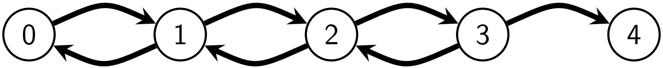
Birth-and-death chains

In **[Diaconis & Miclos '09]** It has been shown that it is possible to intertwine the generator H of a pure birth process below G, whose jump rates $\lambda_d > \cdots > \lambda_1$ are the negatives of the nontrivial eigenvalues of G.



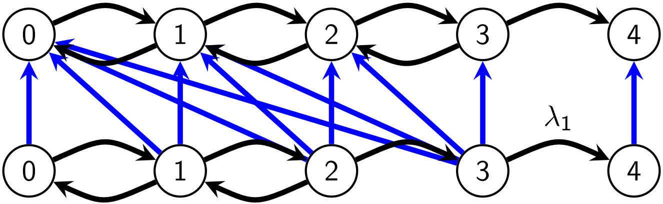
Birth-and-death chains

The proof is based on a repeated application of the Perron-Frobenius theorem.



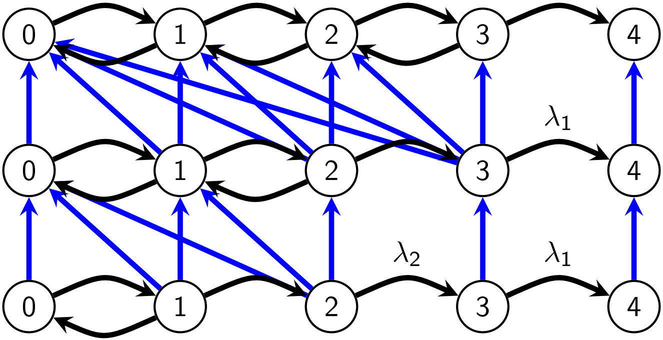
Birth-and-death chains

The proof is based on a repeated application of the Perron-Frobenius theorem.



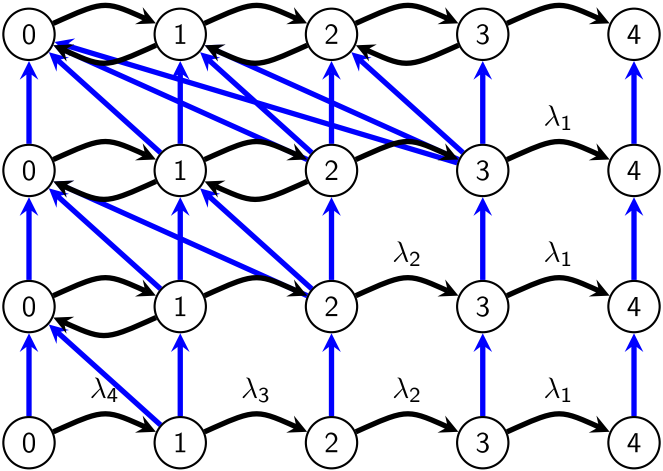
Birth-and-death chains

The proof is based on a repeated application of the Perron-Frobenius theorem.



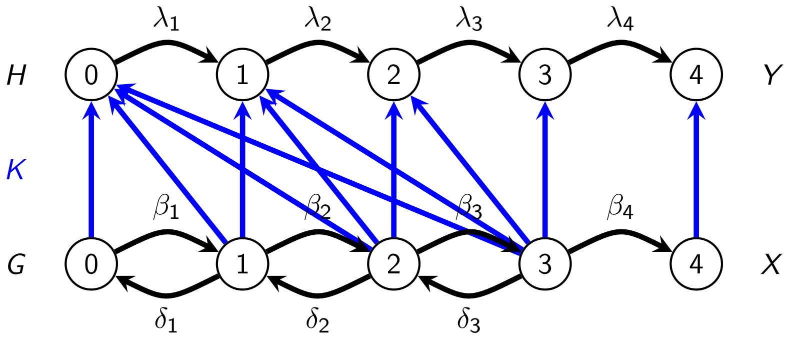
Birth-and-death chains

The proof is based on a repeated application of the Perron-Frobenius theorem.



Birth-and-death chains

In '10, I showed that it is similarly possible to intertwine the generator H of a pure birth process *on top of* the generator G of a birth-and-death process.



Trapping time $P^x[\tau_X = t] = \sum_{y=0}^{d-1} K(x,y) P^y[\tau_Y = t].$

The general picture

The example of birth-and-death chains shows that:

- For some transition kernels P, it is possible to find a simpler transition kernel Q and an intertwining kernel K such that $PK = KQ$ (Q on top of P) or $KP = QK$ (Q below P).
- Using the simpler kernel Q, it is possible to get information about the long-time behaviour of the Markov chain with transition kernel P (such as the time till absorption).
- Even though Q is not diagonal, there is a relation between Q and the eigenvalues of P.

(In our example, $P = P_t$ and $Q = Q_t$ are the transition kernels of a continuous-time Markov process.)

The general picture

Questions remain:

- Is there a similar picture for discrete-time birth-and-death chains?
- To what extent does this generalise beyond birth-and-death chains?
- For example, if P has one absorbing state, then can one always choose K and Q so that they are triangular?
- What if P does not have an absorbing state but is ergodic?

To investigate these and related questions, we will look at numerical methods that aim to find Q and K given P.

We will focus on the problem of finding Q that are intertwined *on top* of P.

An evolution equation

A simple idea:

Let P be a probability kernel of size $d \times d$.
Let 1 denote the identity matrix of size $d \times d$.
Let $K_0, K_1, \ldots$ be inductively defined by

$$K_0 := 1 \quad \text{and} \quad K_{s+1} := K_s + PK_s - K_s Q_s \quad (s \geq 0),$$

where $Q_s = \mathcal{Q}(P, K_s)$ is some function of P and K_s.
Assume that

$$K_s \underset{s\to\infty}{\longrightarrow} K \quad \text{and} \quad Q_s \underset{s\to\infty}{\longrightarrow} Q.$$

Then $PK - KQ = \lim_{s\to\infty}(PK_s - K_s Q_s) = \lim_{s\to\infty}(K_{s+1} - K_s) = 0.$

How to choose the function $\mathcal{Q}(P, K_s)$?

An evolution equation

Here's an idea:

Let $\mathcal{K}$ be the space of probability kernels of size $d \times d$.
Let $[d] := \{1, \ldots, d\}$. Fix $Z \subset \big\{(x, y) \in [d]^2 : x \neq y\big\}$.
Set

$$\mathcal{K}_Z := \big\{K \in \mathcal{K} : K(x, y) = 0 \ \forall (x, y) \in Z\big\},$$
$$\mathcal{C}_Z(P, K) := \big\{Q \in \mathcal{K} : K' := K + PK - KQ \in \mathcal{K}_Z\big\},$$

and define

$$\mathcal{Q}_Z(P, K) := \text{the unique minimiser of}$$
$$Q \mapsto \sum_{x \neq y} Q(x, y) \text{ on } \mathcal{C}_Z(P, K).$$

Assuming the minimiser exists and is unique!

An evolution equation

Recall $K_{s+1} := K_s + PK_s - K_s Q_s$.

- Need to choose Q_s such that $K_{s+1} := K_s + PK_s - K_s Q_s$ is a probability kernel.
- By minimising $Q \mapsto \sum_{x \neq y} Q(x,y)$ we try to choose Q_s as "simple" as possible.
- Without further restrictions on K_{s+1}, the minimiser is $Q_s = 1$, which gives the trivial evolution $K_{s+1} := PK_s$.
- By requiring that $K_{s+1}(x,y) = 0$ for $(x,y) \in Z$, we can use our intuition about what the intertwiner should look like.

This is (so far) *nonrigorous:* no proof that the minimiser exists, or is unique, or that the limits $K := \lim_{s\to\infty} K_s$ and $Q := \lim_{s\to\infty} Q_s$ exist.

An evolution equation

Given probability kernels $P \in \mathcal{K}$ and $K \in \mathcal{K}_Z$,
$\mathcal{C}_Z(P, K)$ is the space of all $d \times d$ matrices such that:

$$
\begin{array}{lll}
\text{(i)} & Q(x,y) \geq 0 & \forall x, y, \\
\text{(ii)} & \displaystyle\sum_{y=1}^{d} Q(x,y) = 1 & \forall x, \\
\text{(iii)} & KQ(x,y) = K(x,y) + PK(x,y) & \forall (x,y) \in Z, \\
\text{(iv)} & KQ(x,y) \leq K(x,y) + PK(x,y) & \forall (x,y) \notin Z.
\end{array}
$$

Here (i) and (ii) say that Q is a probability kernel, while (iii) and (iv) say that $K' := K + PK - KQ$ is nonnegative with $K'(x,y) = 0$ for $(x,y) \in Z$.

The fact that $\sum_y K'(x,y) = 1 \ \forall x$ follows from the fact that P, K, and Q have this property.

An evolution equation

To calculate $Q = \mathcal{Q}_Z(P, K)$, we have to minimise

$$Q \mapsto \sum_{x \neq y} Q(x, y)$$

subject to the constraints

$$\begin{array}{lll}
\text{(i)} & Q(x,y) \geq 0 & \forall x, y, \\
\text{(ii)} & \sum_{y=1}^{d} Q(x,y) = 1 & \forall x, \\
\text{(iii)} & KQ(x,y) = K(x,y) + PK(x,y) & \forall (x,y) \in Z, \\
\text{(iv)} & KQ(x,y) \leq K(x,y) + PK(x,y) & \forall (x,y) \notin Z.
\end{array}$$

This is a standard exercise in linear optimisation.
There exist fast algorithms that give you a solution, if it exists.

The implementation

I have written a couple of scripts in the scientific programming language GNU Octave that numerically solve the equation

$$K_0 := 1 \quad \text{and} \quad K_{s+1} := K_s + PK_s - K_sQ_s \quad (s \geq 0) \qquad (\star).$$

with $Q_s = \mathcal{Q}_Z(P, K_s)$.

The input are a probability kernel $\mathtt{P}$ of size $d \times d$
and a matrix $\mathtt{Z}$ of size $d \times d$ containing only zeros and ones,
where $\mathtt{Z}(x, y) = 1$ means that $(x, y) \in Z$.

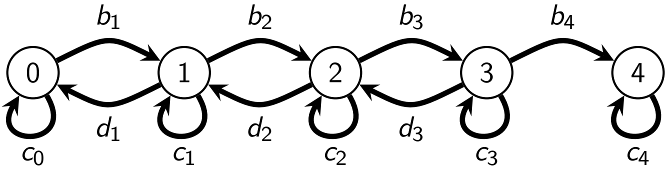
The program runs $(\star)$ until $PK_s - K_sQ_s$ is close enough to zero.

These scripts are available from my homepage
with instructions on how to use them,
so you can give them a try if you wish.

Numerical results

Let P be the transition kernel of a *discrete*-time Markov chain on $\{0, \ldots, d\}$ that jumps
with probability b_k from $k-1$ to k,
with probability c_k from k to k and
with probability d_k from k to $k-1$.

Assume that $c_d = 1$ so that d is a trap.

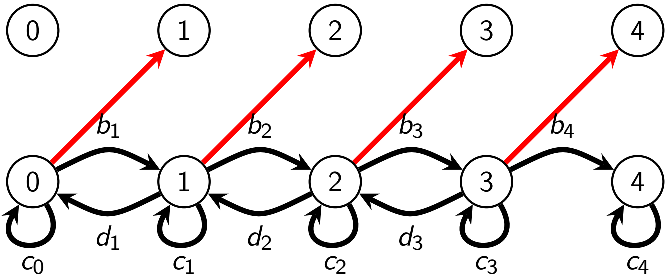


Numerical results

We are looking for an intertwining relation of the form

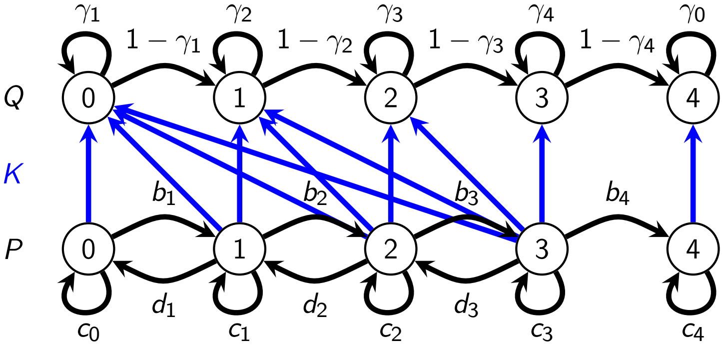
$$PK = KQ$$

where $K(k, k+1) = 0$ for $k = 0, \ldots, d-1$
and Q is as simple as possible.



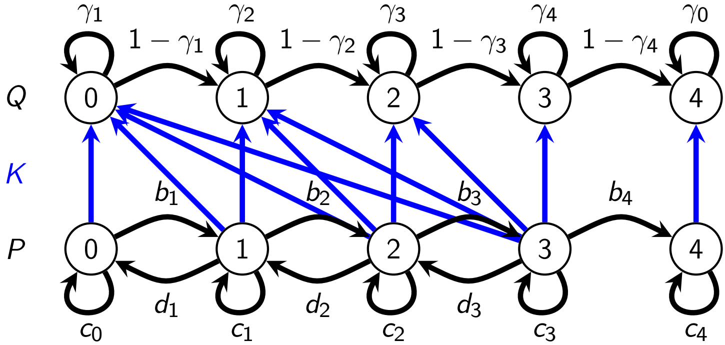
Numerical results

Numerically, we find that such an intertwining really exists.
Here $1 = \gamma_0 \geq \cdots \geq \gamma_d$ are the eigenvalues of P.



Numerical results

Numerically, we find that such an intertwining really exists.
Here $1 = \gamma_0 \geq \cdots \geq \gamma_d$ are the eigenvalues of P.
I am cheating a bit: this works if P is a *lazy kernel* which means that $P = \frac{1}{2}(1 + P')$ for some kernel P', which guarantees that $\gamma_i \geq 0\ \forall i = 0, \ldots, d$.



Numerical results

kern : octave-gui — Konsole

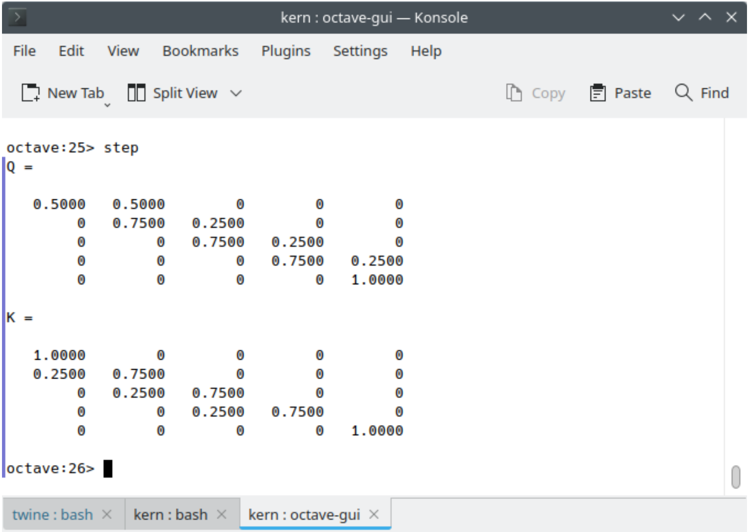
File Edit View Bookmarks Plugins Settings Help

New Tab Split View Copy Paste Find

```
octave:23> P
P =

   0.5000   0.5000        0        0        0
   0.2500   0.5000   0.2500        0        0
        0   0.2500   0.5000   0.2500        0
        0        0   0.2500   0.5000   0.2500
        0        0        0        0   1.0000

octave:24> Z
Z =

   0   1   0   0   0
   0   0   1   0   0
   0   0   0   1   0
   0   0   0   0   1
   0   0   0   0   0

octave:25>
```

twine : bash × kern : bash × kern : octave-gui ×

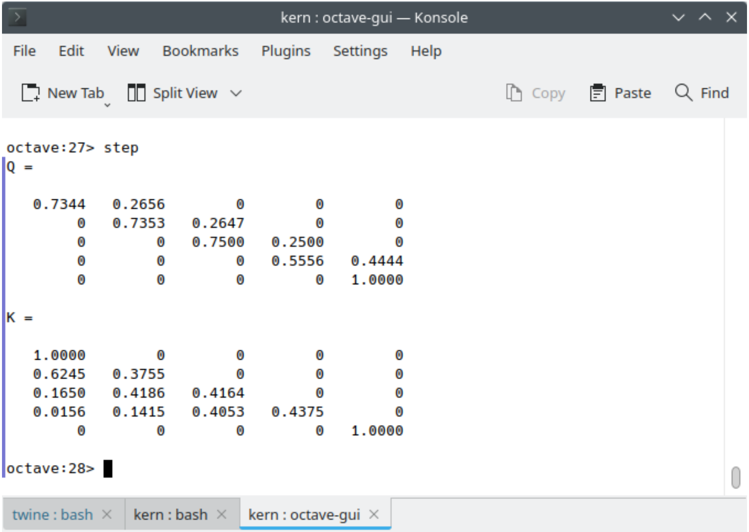
Numerical results



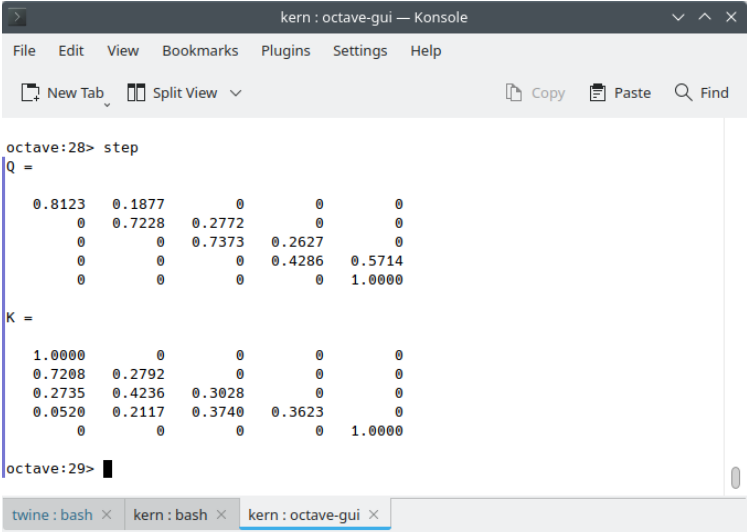
Numerical results

```
octave:26> step
Q =

   0.6250   0.3750        0        0        0
        0   0.7500   0.2500        0        0
        0        0   0.7500   0.2500        0
        0        0        0   0.6667   0.3333
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.4687   0.5312        0        0        0
   0.0625   0.3750   0.5625        0        0
        0   0.0625   0.3750   0.5625        0
        0        0        0        0   1.0000

octave:27>
```

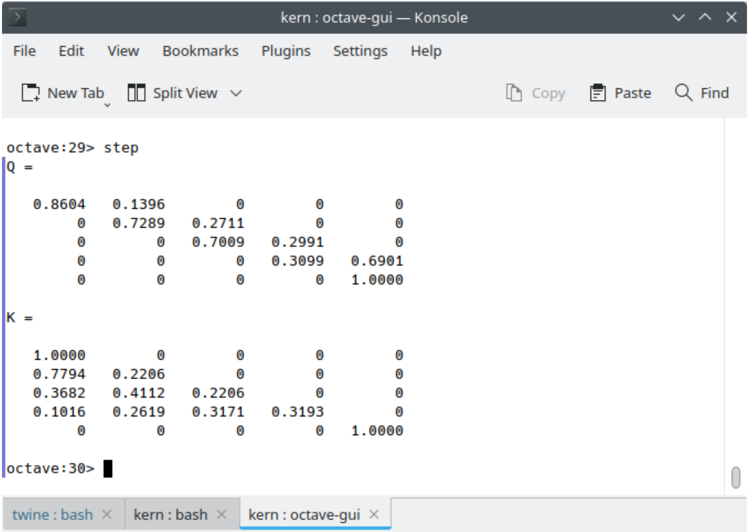
Numerical results



Numerical results



Numerical results



Numerical results

kern : octave-gui — Konsole

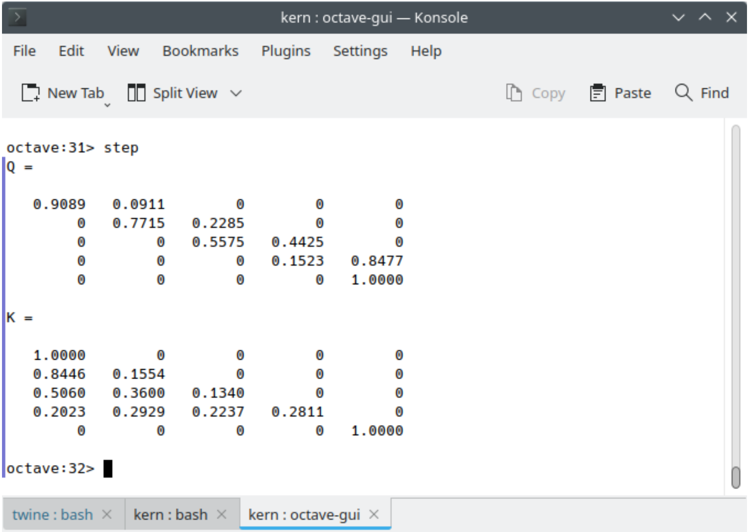
File Edit View Bookmarks Plugins Settings Help

New Tab Split View Copy Paste Find

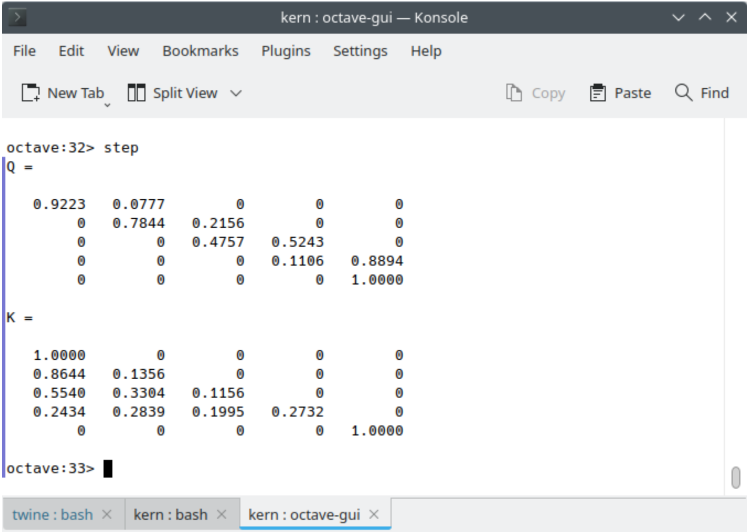
```
octave:30> step
Q =

   0.8897   0.1103        0        0        0
        0   0.7500   0.2500        0        0
        0        0   0.6382   0.3618        0
        0        0        0   0.2170   0.7830
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.8177   0.1823        0        0        0
   0.4449   0.3884   0.1666        0        0
   0.1541   0.2880   0.2630   0.2949        0
        0        0        0        0   1.0000

octave:31>
```

twine : bash | kern : bash | kern : octave-gui

Numerical results

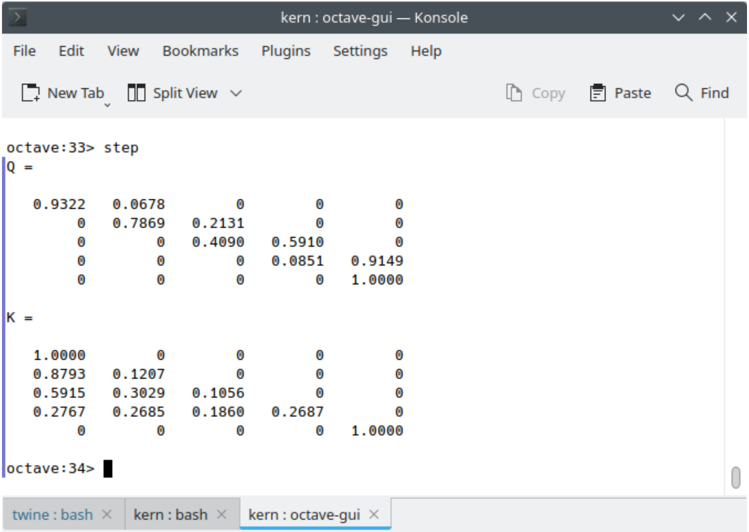


Numerical results




```
octave:32> step
Q =

   0.9223   0.0777        0        0        0
        0   0.7844   0.2156        0        0
        0        0   0.4757   0.5243        0
        0        0        0   0.1106   0.8894
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.8644   0.1356        0        0        0
   0.5540   0.3304   0.1156        0        0
   0.2434   0.2839   0.1995   0.2732        0
        0        0        0        0   1.0000

octave:33>
```

Numerical results



Numerical results

kern : octave-gui — Konsole

File Edit View Bookmarks Plugins Settings Help

New Tab Split View Copy Paste Find

```
octave:34> step
Q =

   0.9397   0.0603        0        0        0
        0   0.7813   0.2187        0        0
        0        0   0.3636   0.6364        0
        0        0        0   0.0696   0.9304
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.8906   0.1094        0        0        0
   0.6205   0.2793   0.1002        0        0
   0.3029   0.2520   0.1791   0.2660        0
        0        0        0        0   1.0000

octave:35>
```

twine : bash | kern : bash | kern : octave-gui

Numerical results

kern : octave-gui — Konsole

File Edit View Bookmarks Plugins Settings Help

New Tab Split View Copy Paste Find

```
octave:35> step
Q =

   0.9453   0.0547        0        0        0
        0   0.7710   0.2290        0        0
        0        0   0.3366   0.6634        0
        0        0        0   0.0600   0.9400
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.8991   0.1009        0        0        0
   0.6425   0.2601   0.0974        0        0
   0.3231   0.2370   0.1757   0.2642        0
        0        0        0        0   1.0000

octave:36>
```

twine : bash | kern : bash | kern : octave-gui

Numerical results

kern : octave-gui — Konsole

File Edit View Bookmarks Plugins Settings Help

New Tab Split View Copy Paste Find

```
octave:36> step
Q =

   0.9496   0.0504        0        0        0
        0   0.7586   0.2414        0        0
        0        0   0.3219   0.6781        0
        0        0        0   0.0537   0.9463
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.9055   0.0945        0        0        0
   0.6592   0.2449   0.0959        0        0
   0.3385   0.2244   0.1741   0.2630        0
        0        0        0        0   1.0000

octave:37>
```

twine : bash × kern : bash × kern : octave-gui ×

Numerical results

kern : octave-gui — Konsole

File Edit View Bookmarks Plugins Settings Help

New Tab Split View Copy Paste Find

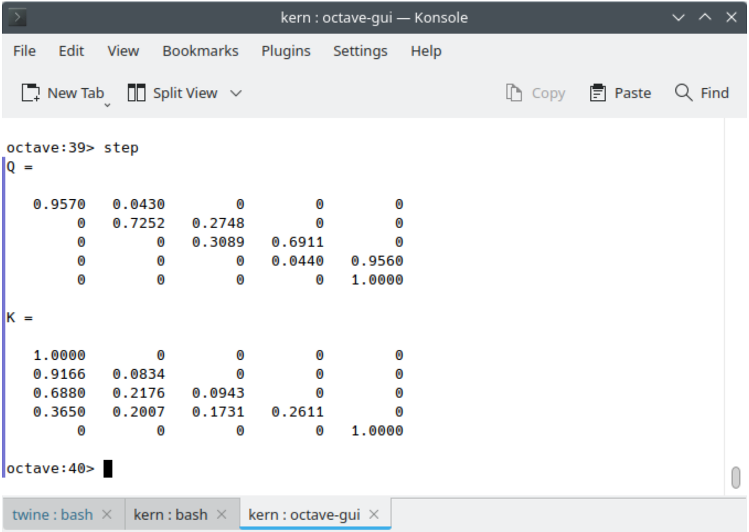
```
octave:37> step
Q =

   0.9528   0.0472        0        0        0
        0   0.7462   0.2538        0        0
        0        0   0.3144   0.6856        0
        0        0        0   0.0493   0.9507
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.9104   0.0896        0        0        0
   0.6718   0.2332   0.0951        0        0
   0.3501   0.2144   0.1734   0.2621        0
        0        0        0        0   1.0000

octave:38>
```

twine : bash kern : bash kern : octave-gui

Numerical results

```
octave:38> step
Q =

   0.9552   0.0448        0        0        0
        0   0.7349   0.2651        0        0
        0        0   0.3107   0.6893        0
        0        0        0   0.0462   0.9538
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.9139   0.0861        0        0        0
   0.6811   0.2243   0.0946        0        0
   0.3587   0.2066   0.1732   0.2615        0
        0        0        0        0   1.0000

octave:39>
```

Numerical results



Numerical results

kern : octave-gui — Konsole

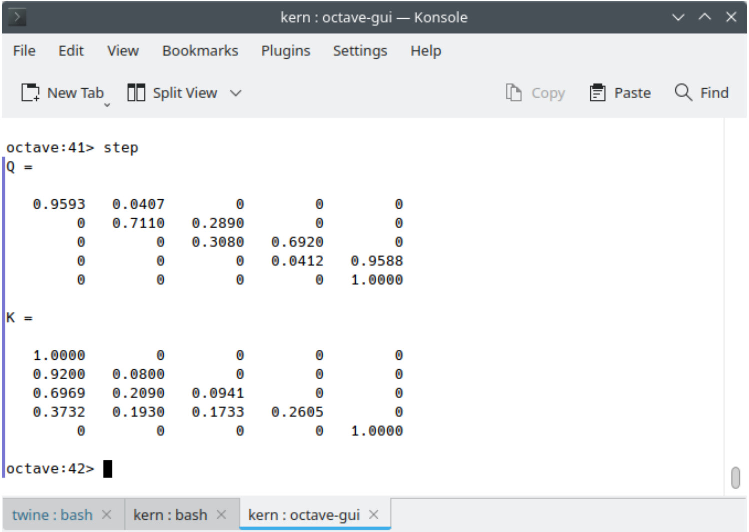
File Edit View Bookmarks Plugins Settings Help

New Tab Split View Copy Paste Find

```
octave:40> step
Q =

   0.9583   0.0417        0        0        0
        0   0.7173   0.2827        0        0
        0        0   0.3082   0.6918        0
        0        0        0   0.0424   0.9576
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.9185   0.0815        0        0        0
   0.6931   0.2127   0.0942        0        0
   0.3697   0.1963   0.1732   0.2608        0
        0        0        0        0   1.0000

octave:41>
```

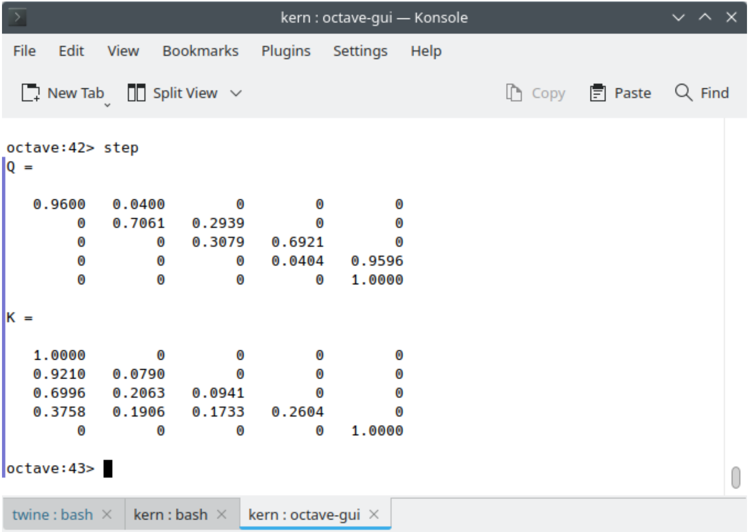
twine : bash | kern : bash | kern : octave-gui

Numerical results



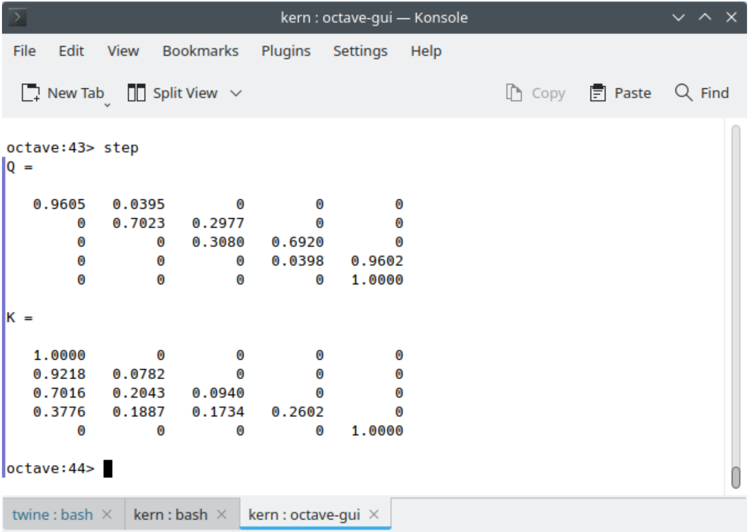

```
octave:41> step
Q =

   0.9593   0.0407        0        0        0
        0   0.7110   0.2890        0        0
        0        0   0.3080   0.6920        0
        0        0        0   0.0412   0.9588
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.9200   0.0800        0        0        0
   0.6969   0.2090   0.0941        0        0
   0.3732   0.1930   0.1733   0.2605        0
        0        0        0        0   1.0000

octave:42>
```

Numerical results

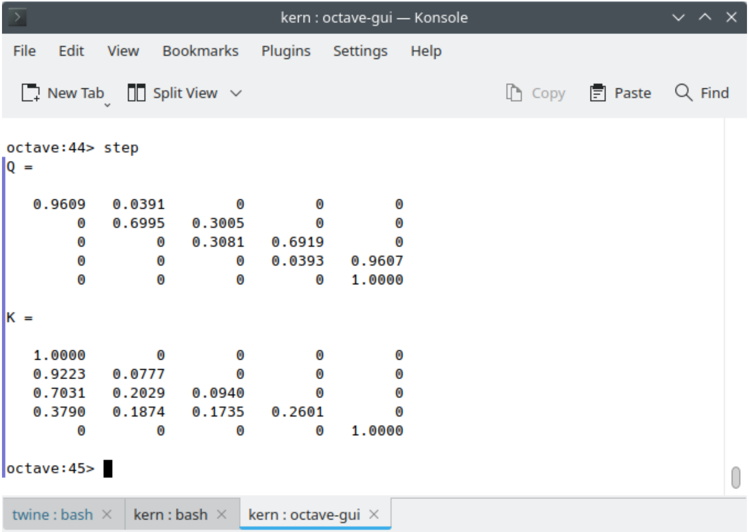



```
octave:42> step
Q =

   0.9600   0.0400        0        0        0
        0   0.7061   0.2939        0        0
        0        0   0.3079   0.6921        0
        0        0        0   0.0404   0.9596
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.9210   0.0790        0        0        0
   0.6996   0.2063   0.0941        0        0
   0.3758   0.1906   0.1733   0.2604        0
        0        0        0        0   1.0000

octave:43>
```

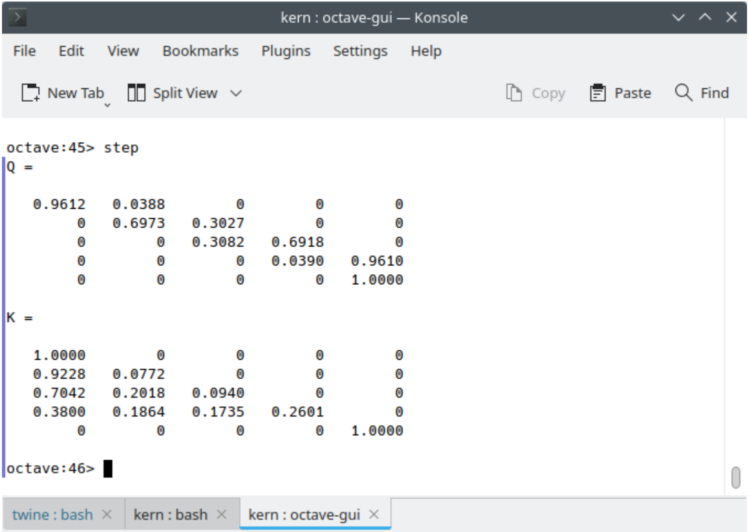
Numerical results



Numerical results

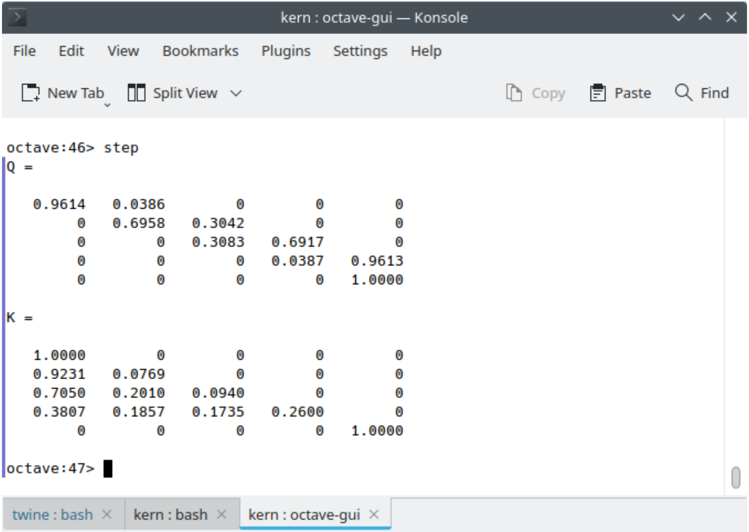


Numerical results

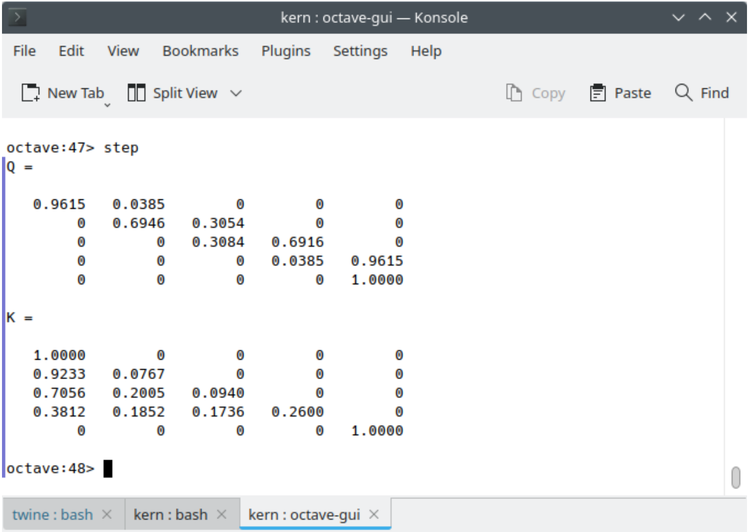



```
octave:45> step
Q =

   0.9612   0.0388        0        0        0
        0   0.6973   0.3027        0        0
        0        0   0.3082   0.6918        0
        0        0        0   0.0390   0.9610
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.9228   0.0772        0        0        0
   0.7042   0.2018   0.0940        0        0
   0.3800   0.1864   0.1735   0.2601        0
        0        0        0        0   1.0000

octave:46>
```

Numerical results



Numerical results



Numerical results

kern : octave-gui — Konsole

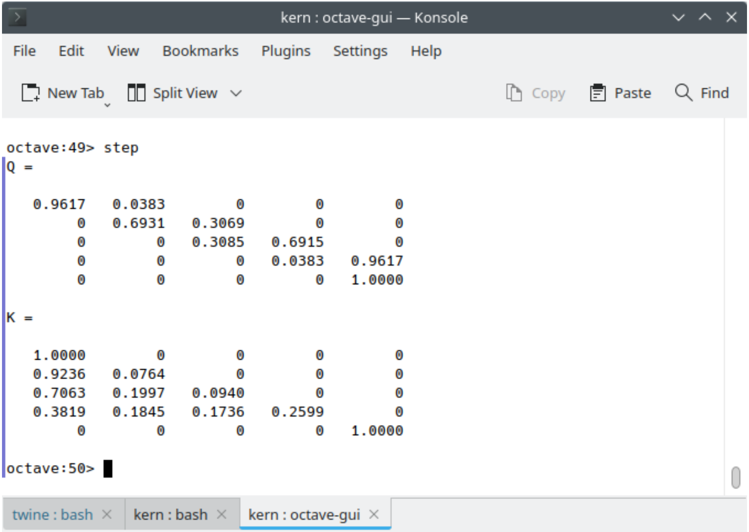
File Edit View Bookmarks Plugins Settings Help

New Tab Split View Copy Paste Find

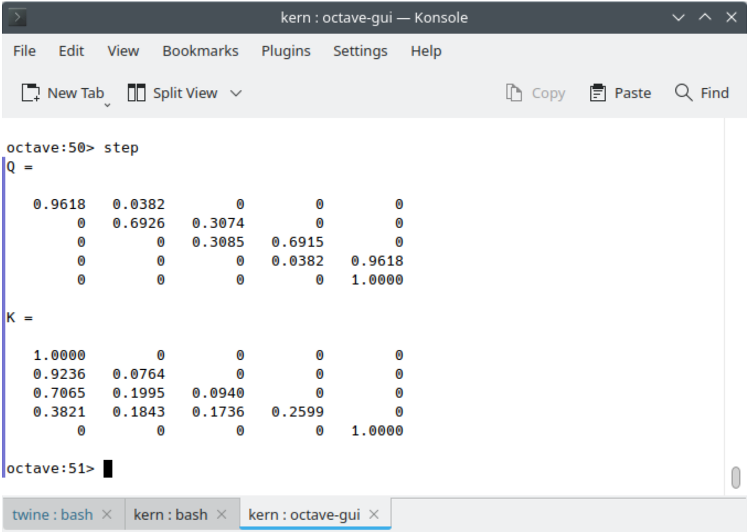
```
octave:48> step
Q =

   0.9616   0.0384        0        0        0
        0   0.6937   0.3063        0        0
        0        0   0.3085   0.6915        0
        0        0        0   0.0384   0.9616
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.9234   0.0766        0        0        0
   0.7060   0.2000   0.0940        0        0
   0.3816   0.1848   0.1736   0.2600        0
        0        0        0        0   1.0000

octave:49>
```

twine : bash | kern : bash | kern : octave-gui

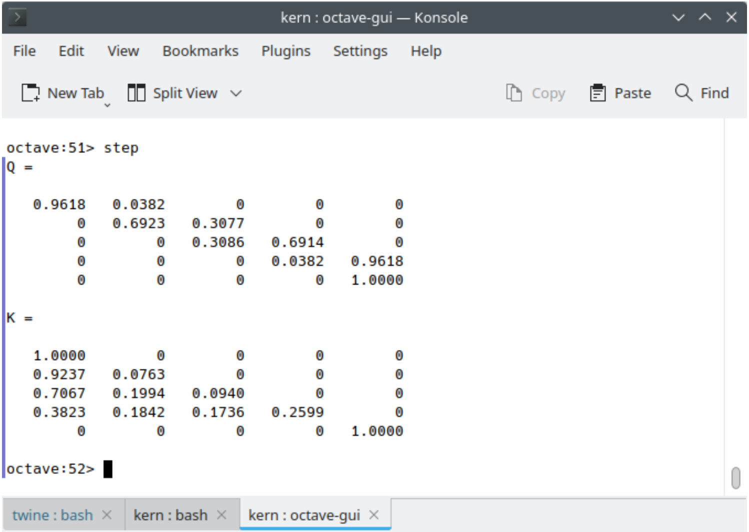
Numerical results



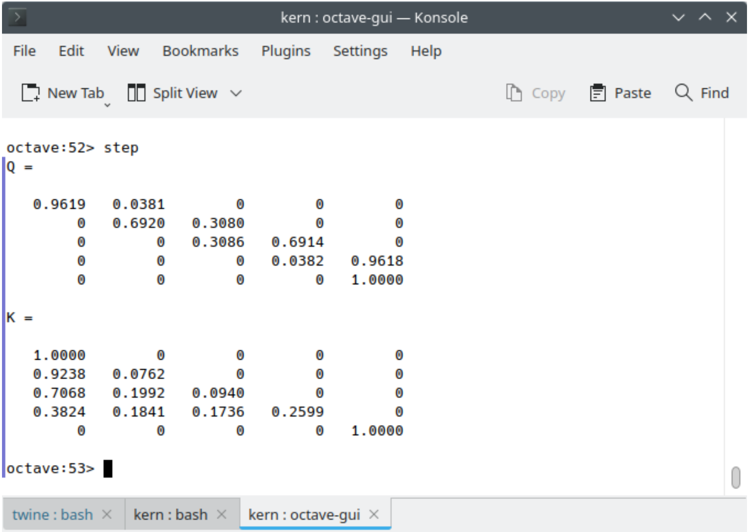
Numerical results



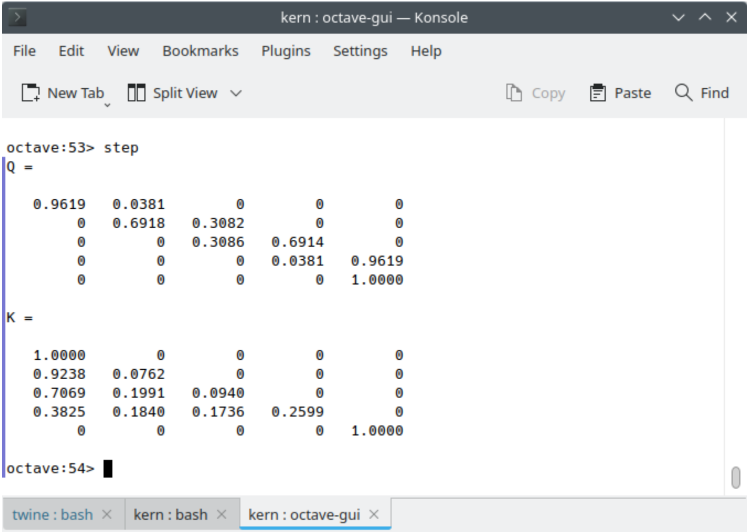
Numerical results



Numerical results



Numerical results



Numerical results

kern : octave-gui — Konsole

File Edit View Bookmarks Plugins Settings Help

New Tab | Split View | Copy | Paste | Find

```
octave:54> step
Q =

   0.9619   0.0381        0        0        0
        0   0.6917   0.3083        0        0
        0        0   0.3086   0.6914        0
        0        0        0   0.0381   0.9619
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.9238   0.0762        0        0        0
   0.7069   0.1991   0.0940        0        0
   0.3825   0.1839   0.1736   0.2599        0
        0        0        0        0   1.0000

octave:55>
```

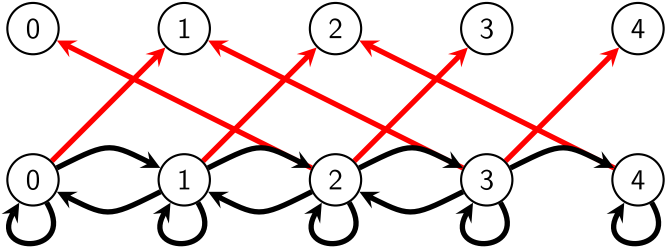
twine : bash | kern : bash | kern : octave-gui

Numerical results

```
octave:55> step
Q =

   0.9619   0.0381        0        0        0
        0   0.6916   0.3084        0        0
        0        0   0.3086   0.6914        0
        0        0        0   0.0381   0.9619
        0        0        0        0   1.0000

K =

   1.0000        0        0        0        0
   0.9238   0.0762        0        0        0
   0.7070   0.1990   0.0940        0        0
   0.3826   0.1839   0.1736   0.2599        0
        0        0        0        0   1.0000

octave:56>
```

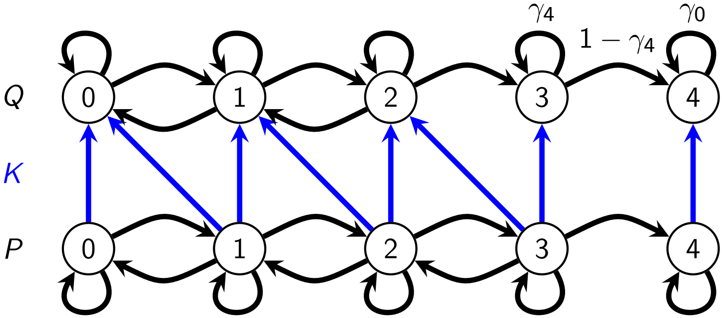
Numerical results

Instead of requiring only $K(k, k+1) = 0$ for $k = 0, \ldots, d-1$ it is interesting to also require that $K(k, k-2) = 0$ for $k = 2, \ldots, d$.



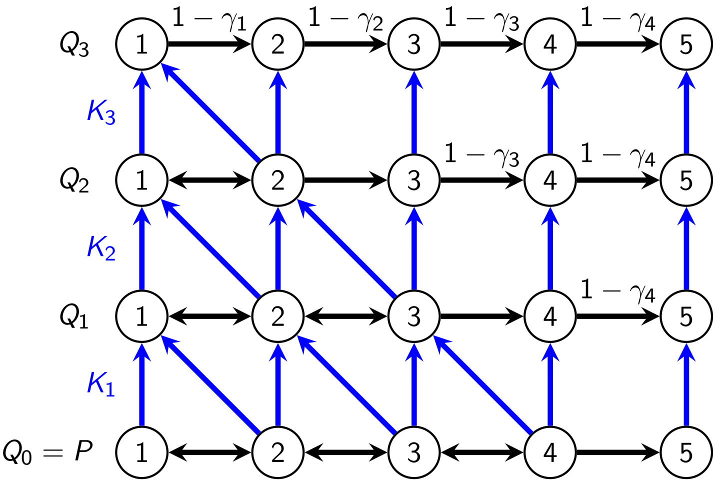
Numerical results

In this case we find an intertwining $PK = KQ$ where K is particularly simple.



Numerical results

This construction can be repeated, leading to the intertwining we have already seen.



Note 1 This inductive construction is *not* how I proved the existence of the intertwining in the continuous-time setting.

Note 2 The eigenvalue γ_4 is the *smallest* eigenvalue, so this is *not* the Perron-Frobenius eigenvalue associated with the killed process on $\{0, \ldots, d-1\}$.

Numerical results

Numerically, we have found that a known intertwining for continuous-time birth-and-death chains also holds for lazy discrete-time birth-and-death chains.

Moreover, we have numerically found a hitherto unknown intertwining for such birth-and-death chains.

This asks for a rigorous proof.

It also makes one wonder what other intertwinings wait to be discovered numerically.

I encourage everybody to try out my scripts.

However, the first indications are that outside of the world of birth-and-death chains, things may not always work so smoothly.

The contact process

Let Λ be a finite set and let $S := \{0,1\}^\Lambda$ be the set of functions $x : \Lambda \to \{0,1\}$.
For each $i, j \in \Lambda$, define $\mathtt{birth}_{ij} : S \to S$ and $\mathtt{dth}_i : S \to S$ by

$$\mathtt{birth}_{ij}(x)(k) := \begin{cases} 1 & \text{if } x(i) = 1,\ k = j, \\ x(k) & \text{otherwise} \end{cases}$$

and

$$\mathtt{dth}_i(x)(k) := \begin{cases} 0 & \text{if } k = i, \\ x(k) & \text{otherwise.} \end{cases}$$

Let p be a probability kernel on Λ and let $0 \leq \lambda \leq 1$.
Let P be the probability kernel on S defined as

$$P(x,y) := \frac{1}{|\Lambda|} \sum_{i \in \Lambda} \Big[\lambda \sum_{j \in \Lambda} p(i,j) 1_{\{y = \mathtt{birth}_{ij}(x)\}} + (1-\lambda) 1_{\{y = \mathtt{dth}_i(x)\}} \Big].$$

The contact process

The Markov chain with transition kernel P has the following description:

- In each step, we first choose a site i uniformly from Λ.
- Next, we choose to *give birth* with probability λ or to *die* with probability $1-\lambda$.
- In case of birth, we choose j according to $p(i,\cdot\,)$ and apply $\mathtt{birth}_{ij}$.
- In case of death, we apply $\mathtt{dth}_i$.

If Λ is large, then it makes sense to rescale time by $|\Lambda|^{-1}$.

Often, there is a limit process as Λ increases to an infinite lattice: the contact process with birth rate λ and death rate $1-\lambda$.

The contact process

On finite lattices, the Markov chain eventually gets trapped in the all-zero state $\underline{0}$.

We would like to understand how fast.

For many large lattices, it has been proved that there is a sharp transition at some $0 < \lambda_{\rm c} < 1$.

For $\lambda < \lambda_{\rm c}$, the time till extinction is of order $\log|\Lambda|$.
For $\lambda > \lambda_{\rm c}$, the time till extinction is of order $e^{|\Lambda|}$.

Can intertwining help us understand this better?

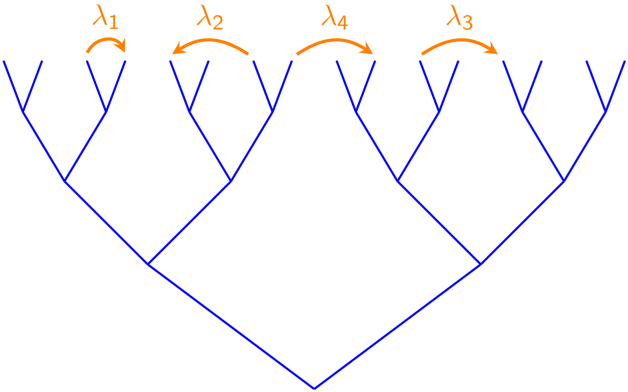
A hierarchical contact process

Let us look at a continuous-time contact process with state space $S_m := \{0,1\}^{\Lambda_m}$ where $\Lambda_m := \{1,2\}^m$.

We denote the death rate by δ and assume that the birth rates $\lambda(i,j) = \lambda_{|i-j|}$ depend only on the *hierarchical distance* $|i-j| := \inf\{k : i_k \neq j_k\}$ between i and j.

A hierarchical contact process

It is useful to picture Λ_m as the set of leaves of a binary tree.



Intertwining and coupling

Assume that H is intertwined on top of G, i.e., $GK = KH$.

Then by **[Fill '92]** it is possible to construct a Markov process $(X_t, Y_t)_{t\geq 0}$ such that

$$\mathbb{P}\big[Y_0 \in \cdot \,\big|\, X_0\big] = K(X_0, \cdot\,) \quad \text{a.s.}$$

implies that

$$\mathbb{P}\big[Y_t \in \cdot \,\big|\, (X_s)_{0\leq s\leq t}\big] = K(X_t, \cdot\,) \quad \text{a.s.} \quad (t \geq 0).$$

Moreover

$$\begin{aligned}\mathbb{P}\big[X_{t+\varepsilon} = x \big| \, (X_s)_{0\leq s\leq t}\big] &= 1(X_t, x) + \varepsilon G(X_t, x) + O(\varepsilon^2),\\ \mathbb{P}\big[Y_{t+\varepsilon} = y \big| \, (X_s, Y_s)_{0\leq s\leq t}\big] &= 1(Y_t, y) + \varepsilon H(Y_t, y) + O(\varepsilon^2).\end{aligned}$$

Intertwining and coupling

For each x, let H_x be a Markov generator. Assume that

$$GK = \hat{K}\hat{H},$$

where

$$\hat{K}f(x) := \sum_y K(x,y)f(x,y) \quad \text{and} \quad \hat{H}f(x,y) := \sum_{y'} H_x(y,y')f(y').$$

Then by **[Athreya & S. '10]** the result of **[Fill '92]** remains true except that now

$$\mathbb{P}\big[Y_{t+\varepsilon} = y \big| (X_s, Y_s)_{0 \le s \le t}\big] = 1(Y_t, y) + \varepsilon H_{X_t}(Y_t, y) + O(\varepsilon^2).$$

This is especially useful if H_x "does not depend too much" on x.

A multiscale argument

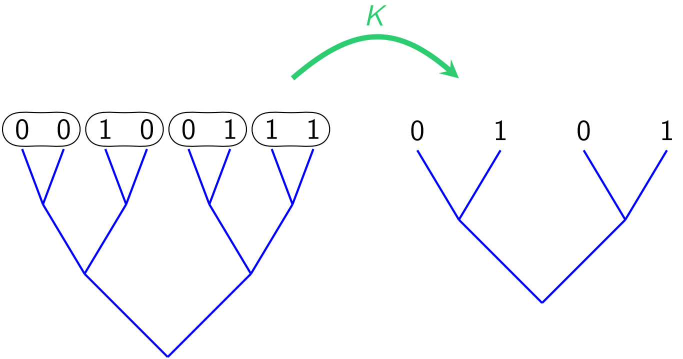
Let $S_m := \{0,1\}^{\Lambda_m}$ with $\Lambda_m := \{1,2\}^m$.

We define a kernel K from S_m to S_{m-1} by independently replacing blocks consisting of two sites by a single site according to the following stochastic rules:

$$00 \mapsto 0, \qquad 11 \mapsto 1,$$
$$01 \text{ or } 10 \mapsto \begin{cases} 0 & \text{with probability } \xi, \\ 1 & \text{with probability } 1-\xi, \end{cases}$$

where $\xi \in (0, \frac{1}{2}]$ is a constant, to be determined later.

A multiscale argument



The probability of this transition is $1 \cdot (1-\xi) \cdot \xi \cdot 1$.

A multiscale argument

We let X be the contact process with state space S_m, birth rates $\lambda_1, \ldots, \lambda_m$, and death rate δ.
We define K from S_m to S_{m-1} as described with

$$\xi := \gamma - \sqrt{\gamma^2 - \tfrac{1}{2}} \quad \text{with} \quad \gamma := \frac{1}{4}\left(3 + \frac{\lambda_1}{2\delta}\right).$$

Then we can construct a Markov process $(X_t, Y_t)_{t \geq 0}$ such that the generator H_x of Y in the presence of X does not depend too much on x.
In particular, Y can stochastically be estimated from below by a contact process Y' on S_{m-1} with birth rates $\lambda'_1, \ldots, \lambda'_{m-1}$ and death rate δ', where

$$\lambda'_k := \tfrac{1}{2}\lambda_{k+1} \quad \text{and} \quad \delta' := 2\xi\delta.$$

A multiscale argument

We may view the map

$$(\delta, \lambda_1, \ldots, \lambda_m) \mapsto (\delta', \lambda'_1, \ldots, \lambda'_{m-1})$$

as an approximate renormalisation transformation.

This can be used to derive lower bounds on the probability that the contact process survives for a long time.

Open problem Do something similar on $\Lambda = \{1, \ldots, d\}$.

Back to numerics

Let P be the transition kernel of the discrete-time contact process with state space $S = \{0,1\}^2$, birth probability $0 \leq \lambda \leq 1$, and death probability $\delta := 1 - \lambda$.

$$P \qquad \begin{array}{c|c|c|c|c|}
 & 00 & 01 & 10 & 11 \\
\hline
00 & & 0 & 0 & 0 \\
\hline
01 & \frac{1}{2}\delta & & 0 & \frac{1}{2}\lambda \\
\hline
10 & \frac{1}{2}\delta & 0 & & \frac{1}{2}\lambda \\
\hline
11 & 0 & \frac{1}{2}\delta & \frac{1}{2}\delta & \\
\hline
\end{array}$$

Back to numerics

We are looking for an intertwining of the form $PK = KQ$ where

$$K(00,00) = 1 \quad \text{and} \quad K(x,\cdot\,) \text{ concentrated on } \{y : x \le y\}.$$

This means that we choose the set Z (or its indicator $\mathtt{Z}$) as follows:

$\mathtt{Z}$

	00	01	10	11
00		1	1	1
01	1		1	
10	1	1		
11	1	1	1	

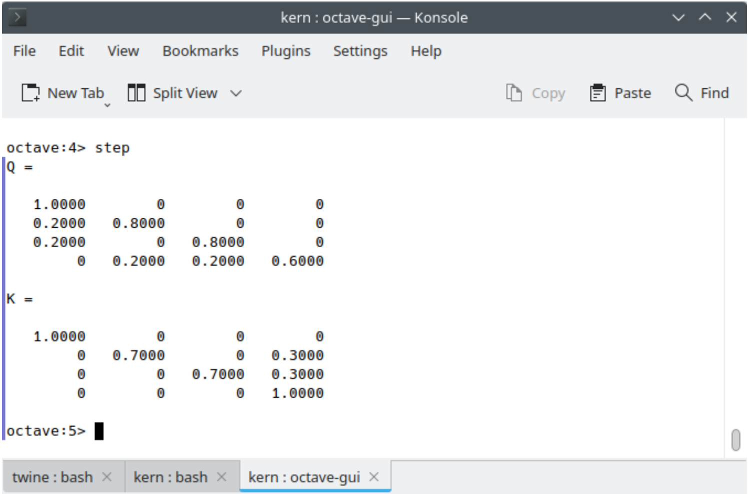
Back to numerics

Let's try this for $\lambda = 0.6$.

```
octave:2> P
P =

   1.0000        0        0        0
   0.2000   0.5000        0   0.3000
   0.2000        0   0.5000   0.3000
        0   0.2000   0.2000   0.6000

octave:3> Z
Z =

   0   1   1   1
   1   0   1   0
   1   1   0   0
   1   1   1   0

octave:4>
```

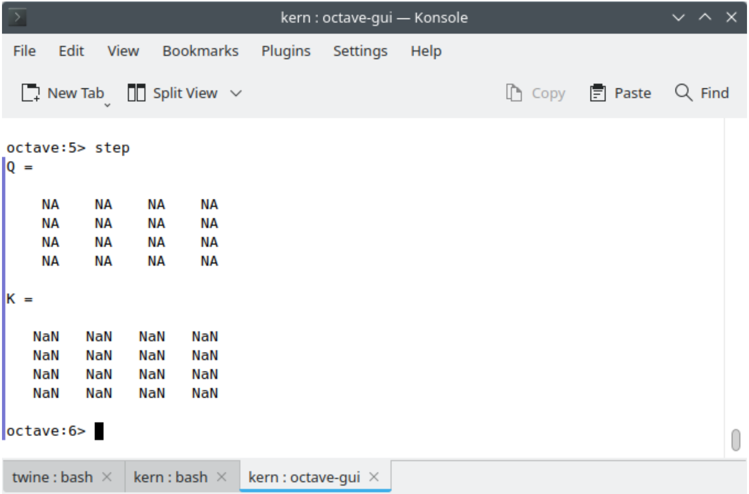
Back to numerics

Let's try this for $\lambda = 0.6$.



Back to numerics

Let's try this for $\lambda = 0.6$.



Back to numerics

What happens here is that after setting $K_0 := 1$ we can calculate $Q_1 := \mathcal{Q}_Z(P, K_0)$ and $K_1 := K_0 + PK_0 - K_0 Q_1$ all right, but when we try to calculate $Q_2 := \mathcal{Q}_Z(P, K_1)$ we run into the problem that $\mathcal{C}_Z(P, K_1) = \emptyset$ so there is no minimiser.

Work in progress...