```
> restart:with(linalg):with(combinat, numbcomb):
```

Warning, the protected names norm and trace have been redefined and unprotected

```
> N:=2:m:=2*N:
```

```
> # we need to index from 1 to 2N+1, instead of "0 to 2N", so a bit messy:
```

```
> H:=matrix(m+1,m+1,(i,k) -> (1/m)^m * numbcomb(m,k-1) * (i-1)^(k-1)*(m-i+1)^(m-k+1));
```

$$H := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{81}{256} & \frac{27}{64} & \frac{27}{128} & \frac{3}{64} & \frac{1}{256} \\ \frac{1}{16} & \frac{1}{4} & \frac{3}{8} & \frac{1}{4} & \frac{1}{16} \\ \frac{1}{256} & \frac{3}{64} & \frac{27}{128} & \frac{27}{64} & \frac{81}{256} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> evalm(H^2);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{3795}{8192} & \frac{477}{2048} & \frac{729}{4096} & \frac{189}{2048} & \frac{275}{8192} \\ \frac{85}{512} & \frac{27}{128} & \frac{63}{256} & \frac{27}{128} & \frac{85}{512} \\ \frac{275}{8192} & \frac{189}{2048} & \frac{729}{4096} & \frac{477}{2048} & \frac{3795}{8192} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> # fractions are hard to read, so from now on print as float with 2 significant digits:
```

```
> evalf(evalm(H^2),2);
```

$$\begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0.46 & 0.23 & 0.18 & 0.092 & 0.034 \\ 0.17 & 0.21 & 0.25 & 0.21 & 0.17 \\ 0.034 & 0.092 & 0.18 & 0.23 & 0.46 \\ 0. & 0. & 0. & 0. & 1. \end{bmatrix}$$

```
> evalf(evalm(H^3),2);
```

$$\begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0.55 & 0.15 & 0.14 & 0.094 & 0.075 \\ 0.25 & 0.16 & 0.18 & 0.16 & 0.25 \\ 0.075 & 0.094 & 0.14 & 0.15 & 0.55 \\ 0. & 0. & 0. & 0. & 1. \end{bmatrix}$$

```
> evalf(evalm(H^4),2);
```

$$\begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0.60 & 0.10 & 0.10 & 0.081 & 0.11 \\ 0.31 & 0.12 & 0.14 & 0.12 & 0.31 \\ 0.11 & 0.081 & 0.10 & 0.10 & 0.60 \\ 0. & 0. & 0. & 0. & 1. \end{bmatrix}$$

```
> evalf(evalm(H^5),2);
```

$$\begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0.64 & 0.072 & 0.076 & 0.064 & 0.15 \\ 0.36 & 0.090 & 0.10 & 0.090 & 0.36 \\ 0.15 & 0.064 & 0.076 & 0.072 & 0.64 \\ 0. & 0. & 0. & 0. & 1. \end{bmatrix}$$

```
> evalf(evalm(H^10),2);
```

$$\begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0.72 & 0.016 & 0.018 & 0.016 & 0.22 \\ 0.47 & 0.021 & 0.024 & 0.021 & 0.47 \\ 0.22 & 0.016 & 0.018 & 0.016 & 0.72 \\ 0. & 0. & 0. & 0. & 1. \end{bmatrix}$$

```
> evalf(evalm(H^20),2);
```

$$\begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0.75 & 0.00091 & 0.0010 & 0.00091 & 0.25 \\ 0.50 & 0.0012 & 0.0014 & 0.0012 & 0.50 \\ 0.25 & 0.00091 & 0.0010 & 0.00091 & 0.75 \\ 0. & 0. & 0. & 0. & 1. \end{bmatrix}$$

```
> evalf(evalm(H^30),2);
```

$$\begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0.75 & 0.000051 & 0.000057 & 0.000051 & 0.25 \\ 0.50 & 0.000068 & 0.000077 & 0.000068 & 0.50 \\ 0.25 & 0.000051 & 0.000057 & 0.000051 & 0.75 \\ 0. & 0. & 0. & 0. & 1. \end{bmatrix}$$

```
> # now suppose that initial frequency of A alleles is
  (deterministic) "i/2N"
> # e.g. if p(0) = (1,0,0,0,0), then p(infty) = p(0) times
  H^infty = (1,0,0,0,0)
> # this means that P(f_A(infinity)=0)=1, P(f_A(infinity)=i/2N)=1
  for i>0, i.e. all A if start all A
> # if p(0)=(0,1,0,0,0), then p(infinity) = (3/4,0,0,0,1/4)
  (this is when f_A(0)=1/2N = 1/4)
> # which means that for large t, P(f_A(t)=0) = 3/4 (same as
  P(f_a=1)) and P(f_A(t)=4/4=1) = 1/4, rest=0
> # in general, if initial is i/N, we get P(f_A(infty)=1) = i/N
  and P(f_a(infty)=1) = 1-i/N, rest=0
```

```
> # consistent with "pure" population (since with prob=1 will
  have some generation with no a's or A's)
> # now let us try computing eigens:
> E:=eigenvectors(transpose(H));
```

$$E := \left[\frac{3}{32}, 1, \{[1, -4, 6, -4, 1]\}\right], \left[\frac{3}{8}, 1, \{[-1, 2, 0, -2, 1]\}\right], \left[\frac{3}{4}, 1, \left\{\left[\frac{-25}{16}, 1, \frac{9}{8}, 1, \frac{-25}{16}\right]\right\}\right], [1, 2, \{[0, 0, 0, 0, 1], [1, 0, 0, 0, 0]\}]$$

```
> # I used the transpose because we are representing vectors as
  ROWS, not columns as in the previous theory!
> # let's pick up the eigenvals equal to 1 (the rest are all
  strictly <1)
> F:=E[4];
```

$$F := [1, 2, \{[0, 0, 0, 0, 1], [1, 0, 0, 0, 0]\}]$$

```
> G:=F[3];
```

$$G := \{[0, 0, 0, 0, 1], [1, 0, 0, 0, 0]\}$$

```
> V1:=G[1];
```

$$V1 := [1, 0, 0, 0, 0]$$

```
> V2:=G[2];
```

$$V2 := [0, 0, 0, 0, 1]$$

```
> # so, all solutions will end up being combinations of these two
  (meaning prob=1 of all a or all A);
> # we still need the rest to set constants for initial
  conditions
> V3:=E[1][3][1];
```

$$V3 := [1, -4, 6, -4, 1]$$

```
> V4:=E[2][3][1];
```

$$V4 := [-1, 2, 0, -2, 1]$$

```
> V5:=E[3][3][1];
```

$$V5 := \left[\frac{-25}{16}, 1, \frac{9}{8}, 1, \frac{-25}{16}\right]$$

```
> # given initial condition P, find c_i so that  P = c_1 V1 +
  ..., i.e. solve "P = CV" for vector "C"
> V:=matrix(5,5,[V1,V2,V3,V4,V5]);
```

$$V := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & -4 & 6 & -4 & 1 \\ -1 & 2 & 0 & -2 & 1 \\ \frac{-25}{16} & 1 & \frac{9}{8} & 1 & \frac{-25}{16} \end{bmatrix}$$

```
> P:=array([0,1,0,0,0]);
```

$$P := [0, 1, 0, 0, 0]$$

```
> C:=evalm(P*V^(-1));
```

$$C := \left[\frac{3}{4}, \frac{1}{4}, \frac{-3}{56}, \frac{1}{4}, \frac{2}{7} \right]$$

```
> # this means that for p(0)=(0,1,0,0,0) we get p(infinity) =
  (3/4)V1 + (1/2)V2 = (3/4,0,0,0,1/4)
> # just like we concluded from the "brute force" taking of
  powers
>
> # NOTE: there is a nice theoretical way (involving the theory
  of "martingales") to get all this without computation!
>
```