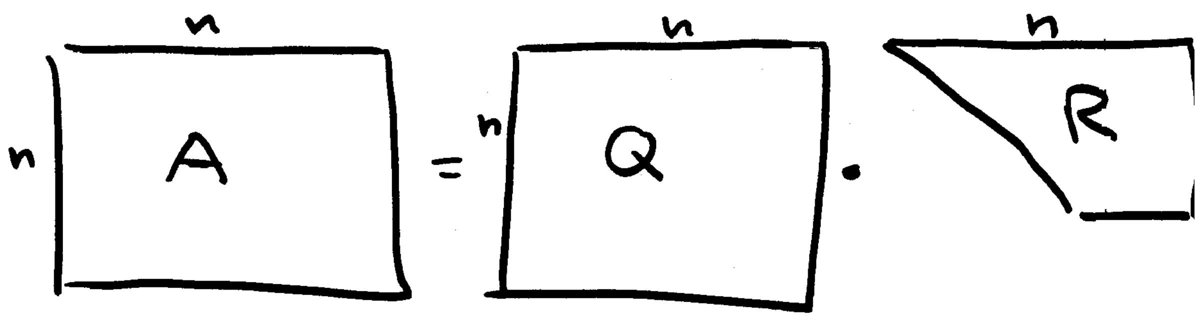
The QR-Decomposition:

Lemma: Any matrix (A) can be decomposed into a unitary matrix (Q) and an uppertriangular matrix (R):

$$\boxed{A = Q * R} .$$

Assume: $\text{Rank}(A) = r < n$

$\Rightarrow \text{Rank}(R) = r$



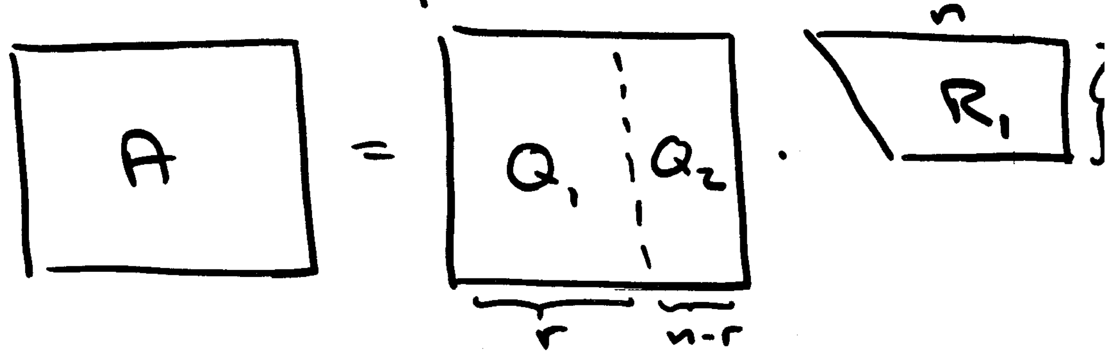
- $\text{Rank}(Q) = n$, as Q is unitary.
- Warning: $\text{Eig}(R) \neq \text{Eig}(A)$!!

- Let us take a matrix A

$$A = [a_1, a_2, \ldots, a_n]$$

with $\text{Rank}(A) = r < n$

- We want to find an Image and a Nullspace for A.



$$A = Q \cdot R = Q_1 \cdot R_1 + Q_2 \cdot \emptyset = Q_1 \cdot R_1$$

$$\boxed{A = Q_1 \cdot R_1}$$

$\Rightarrow$ The q_i-vectors are linear combinations of the $\underline{a}_i$-vectors. They are of length 1, and are orthogonal to each other

$\Rightarrow$ Q_1 is a <u>column image</u> of A.

- As the vectors in Q_2 are of length 1, and are orthogonal to all vectors of Q_1

$\Rightarrow$ Q_2 is a <u>columnnullspace</u> of A.

In Matlab:

```
k = rank (A);
[n,n] = size(A);
[q,r] = qr (A);
q1 = q(:, 1:k);
q2 = q(:, k+1,n);
```

- Sometimes, we want to find the row image and row nullspace of A.

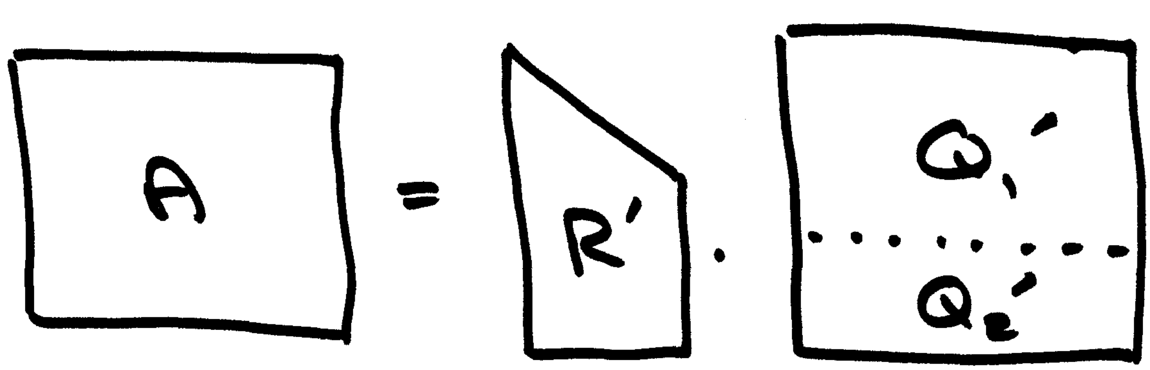
$$A = [\underline{a}_1'; \underline{a}_2'; \ldots; \underline{a}_n']$$

with $\operatorname{rank}(A) = r < n$

$$\Rightarrow A' = Q \cdot R$$

is the same problem as before.

$$\Rightarrow A = R' \cdot Q'$$



$\Rightarrow$ Q_1' is a row image of A,
Q_2' is a row nullspace of A.

In Matlab:

```
k=rank(A);
[n,n]=size(A);
[q,r]=qr(A');
q1 = q(:,1:k)';
q2= q(:,k+1:n)';
```

This is an efficient way to compute these quantities.

- How the QR-decomposition can be found manually is more advanced, and will therefore be discussed later.

Example:

We want to determine the solution to the problem:

$$(A-\lambda_i I)^k \cdot \underline{v}_{ik} = \emptyset$$

Let us abbreviate:

$$A_i = (A - \lambda_i I)$$

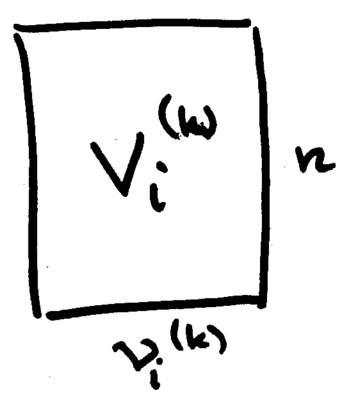
and: $$A_i^{(k)} = A_i^{\,k} = (A - \lambda_i I)^k$$

There exist $\nu_i^{(k)}$ vectors $\underline{v}_{i_k}$ that satisfy the condition:

$$A_i^{(k)} \cdot \underline{v}_{i_k} = \phi$$

Let: $$V_i^{(k)} = [\, \underline{v}_{i_{k_1}}, \ldots, \underline{v}_{i_{k_{\nu_i^{(k)}}}} \,]$$

be the matrix of these vectors:



We want to find $V_i^{(k)}$.

$\Leftrightarrow$ The $\underline{v}_{ik}$-vectors are all perpendicular to all row vectors of $A_i^{(k)}$.

$\Leftrightarrow$ $V_i^{(k)}$, is in the rownullspace of $A_i^{(k)}$.

$\Rightarrow$
```
Ai = A - li * eye(n);
Aik = Ai^k;
rik = rank(Aik);
nik = n - rik;
[q,r] = qr(Aik');
vik = q(:, rik+1:n);
```

is the desired matrix.

<u>Example</u>:

$$\left| \begin{array}{l} A_i^{(k)} \cdot \underline{v}_{ik} = \emptyset \\ A_i^{(k-1)} \cdot \underline{v}_{ik} \neq \emptyset \end{array} \right)$$

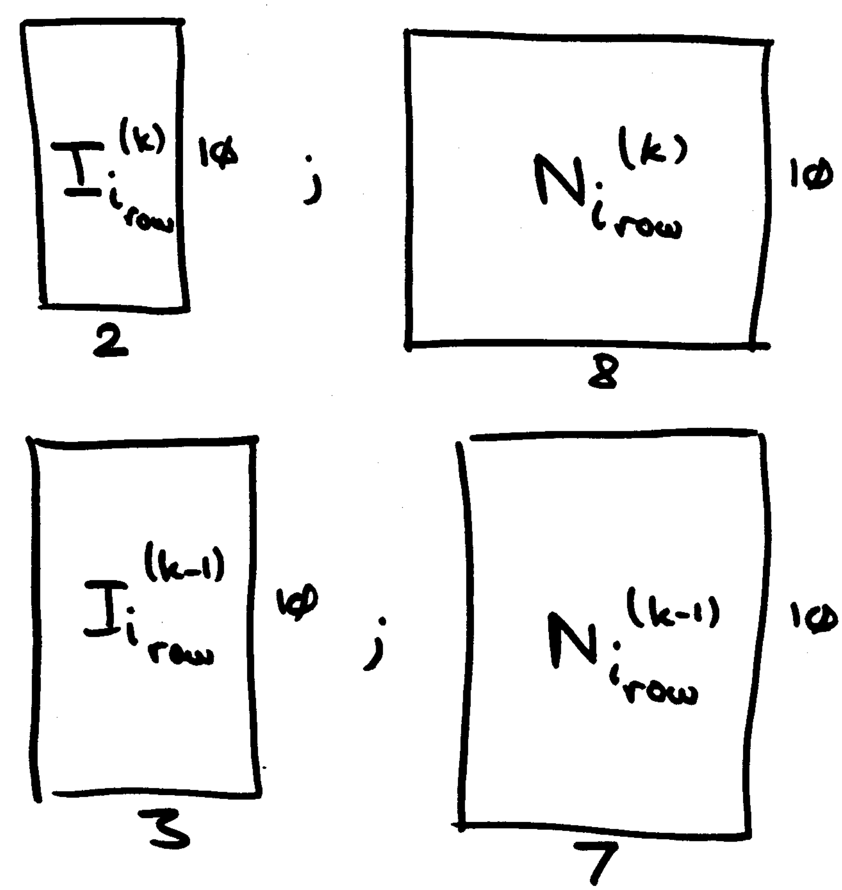
We want to find a sub= set of the previous matrix where $\underline{v}_{i_k}$ are in the (row-) Nullspace of $A_i^{(k)}$, but in the Image of $A_i^{(k-1)}$.

Assume: $n = 10$; $m_i = 8$

$\rho_i^{(k)} = 2$; $\nu_i^{(k)} = 8$

$\rho_i^{(k-1)} = 3$; $\nu_i^{(k-1)} = 7$

$\Rightarrow$



Idea: Find $I_{row}^{(k)}$ and $N_{row}^{(k-1)}$, and build the matrix:

$$X = \begin{array}{c} \left[\begin{array}{c|c|c} I_{row}^{(k)} & N_{row}^{(k-1)} & \emptyset \end{array}\right] 10 \\ \quad 2 \qquad\quad 7 \qquad\quad 1 \end{array}$$

$\Rightarrow \underline{v}_{ik} = N_{col}(X)$

In Matlab:

```
Ai = A - li*eye(n);
Aikm1 = Ai^(k-1);
Aik = Aikm1*Ai;
rikm1 = rank(Aikm1);
nikm1 = n-rikm1;
rik = rank(Aik);
nik = n-rik;
[qk,r] = qr(Aik');
[qkm1,r] = qr(Aikm1');
iik = qk(:,1:rik);
nikm1 = qkm1(:,rikm1+1:n);
```

```
X = zeros(n);
X(:, 1:rik) = iik;
X(:, rik+1, rik+nikm1) = mikm1;
[q,r] = qr(X);
v = q(:, rik+nikm1+1:n);
```

v is the desired generalized eigenvector of grade k.

Transcendental Functions:

- We want to study some properties of the spectral decomposition:

$$\boxed{A = V \cdot \Lambda \cdot V^{-1}}$$

(a) $A^k = ?$

$$A^k = \underbrace{A \cdot A \cdot \; \dots \; \cdot A}_{k \text{ times}}$$

$$\Rightarrow A^k = (V\Lambda V^{-1}) * (V\Lambda V^{-1}) * \cdots * (V\Lambda V^{-1})$$

$$= V \cdot \Lambda * \Lambda * \cdots * \Lambda \cdot V^{-1}$$

$$= V \cdot \Lambda^k \cdot V^{-1}$$

$$\Rightarrow \boxed{A^k = V \cdot \Lambda^k \cdot V^{-1}}$$

Of course:

$$\Lambda = \begin{bmatrix} \lambda_1 & & \emptyset \\ & \ddots & \\ \emptyset & & \lambda_j \end{bmatrix} \Rightarrow \Lambda^k = \begin{bmatrix} \lambda_1^k & & \emptyset \\ & \ddots & \\ \emptyset & & \lambda_j^k \end{bmatrix}$$

$$\underline{\underline{\Lambda_i^k = ?}}$$

Example:

$$\Lambda_i = \begin{bmatrix} \lambda_i & 1 & & & \emptyset \\ & \lambda_i & 1 & & \\ & & \lambda_i & 1 & \\ & \emptyset & & \lambda_i & 1 \\ & & & & \lambda_i \end{bmatrix}$$

$$\Rightarrow \Lambda_i^4 = \begin{bmatrix} \lambda_i^4 & 4\lambda_i^3 & 6 \cdot \lambda_i^2 & 4 \cdot \lambda_i & 1 \\ \emptyset & \lambda_i^4 & 4 \cdot \lambda_i^3 & 6 \cdot \lambda_i^2 & 4 \cdot \lambda_i \\ \emptyset & \emptyset & \lambda_i^4 & 4 \cdot \lambda_i^3 & 6 \cdot \lambda_i^2 \\ \emptyset & \emptyset & \emptyset & \lambda_i^4 & 4 \cdot \lambda_i^3 \\ \emptyset & \emptyset & \emptyset & \emptyset & \lambda_i^4 \end{bmatrix}$$

$$\Rightarrow \Lambda_i^k = T_{up}\left\{\left[\lambda_i^k, \binom{k}{1}\lambda_i^{k-1}, \binom{k}{2}\lambda_i^{k-2}, \dots\right]\right\}$$

which can be written down at once.

- If we want e.g. to compute

$$B = A^{1000}$$

(which is a bad idea in the first place!), we do <u>not</u> write:

```
B = eye(size(A));
for i = 1:1000,
    B = B*A;
end
```

<u>Instead:</u>

```
[V,L] = eig(A);
l = diag(L);
for i = 1:n,
    l(i) = l(i)^1000;
end
```

L=diag(l);
B=V*L/V;

assuming that all eigenvalues of A are distinct.

In Matlab:

B=A^1000

uses exactly this method, and generates an error message if A has multiple eigenvalues!

(6) $e^{At} = ?$

Remember:

$$e^{At} = I + At + \frac{A^2t^2}{2!} + \dots$$
$$= VIV^{-1} + V\Lambda t V^{-1} + \frac{V\Lambda^2 t^2 V^{-1}}{2!} + \dots$$
$$= V\left[I + \Lambda t + \frac{\Lambda^2 t^2}{2!} + \dots\right] \cdot V^{-1}$$

$$\Rightarrow \boxed{e^{At} = V \cdot e^{\Lambda t} \cdot V^{-1}}$$

of course :

$$\Lambda = \begin{bmatrix} \lambda_1 & & \emptyset \\ & \ddots & \\ \emptyset & & \lambda_j \end{bmatrix} \Rightarrow e^{\Lambda t} = \begin{bmatrix} e^{\Lambda_1 t} & & \emptyset \\ & \ddots & \\ \emptyset & & e^{\Lambda_j t} \end{bmatrix}$$

$$\Lambda_i = \begin{bmatrix} \lambda_i & 1 & & \emptyset \\ & \lambda_i & 1 & \\ & & \ddots & 1 \\ \emptyset & & & \lambda_i \end{bmatrix}$$

$$\Rightarrow e^{\Lambda_i t} = \begin{bmatrix} e^{\lambda_i t} & te^{\lambda_i t} & \cdots & \frac{t^{m_i-1}}{(m_i-1)!}e^{\lambda_i t} \\ & e^{\lambda_i t} & te^{\lambda_i t} & \vdots \\ & & \ddots & te^{\lambda_i t} \\ \emptyset & & & e^{\lambda_i t} \end{bmatrix}$$

$$\Rightarrow e^{\Lambda_i t} = \mathcal{T}_{up}\left\{ \left[1, t, \frac{t^2}{2!}, \dots \frac{t^{m_i-1}}{(m_i-1)!}\right] e^{\lambda_i t} \right\}$$

which is another way to compute the exponential of a matrix.

In Matlab: This is the algorithm that will be used when you specify:

$$F = \mathrm{expm}(A)$$

that is:

```
[V,L] = eig(A);
l = diag(L);
xl = exp(l);      ← elementwise
Xl = diag(xl);
F = V*Xl/V;
```

The function "expm" generates an error message if A has multiple eigenvalues.

Let us take <u>any</u> continuous and continuously differentiable function, and develop it into a Taylor series, e.g.

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + - \dots$$

We define:

$$\sin(A) = A - \frac{A^3}{3!} + \frac{A^5}{5!} - \frac{A^7}{7!} + - \dots$$

for any square matrix.

$$\Rightarrow \quad \boxed{\sin(A) = V \cdot \sin(\Lambda) \cdot V^{-1}}$$

That is also what CTRL-C does. Such functions are called <u>transcendental functions</u>

$$\boxed{f(A) = V \cdot f(\Lambda) \cdot V^{-1}}$$

Cayley Hamilton Theorem:

Given a square matrix A.
Its characteristic polynomial
$$q(\lambda) = \det(\lambda I - A)$$
has a set of roots : λ_i
which are also the eigen=
values of A.

$$\Rightarrow \quad q(\lambda_i) \equiv \phi \; ; \; \forall i$$

Example:
$$A = \begin{bmatrix} 1 & \phi & \phi \\ 2 & 3 & \phi \\ 4 & 5 & 6 \end{bmatrix}$$

$$\Rightarrow (\lambda I - A) = \begin{bmatrix} (\lambda-1) & \phi & \phi \\ -2 & (\lambda-3) & \phi \\ -4 & -5 & (\lambda-6) \end{bmatrix}$$

$$\Rightarrow q(\lambda) = (\lambda-1)(\lambda-3)(\lambda-6)$$
$$\Rightarrow q(\lambda) = \lambda^3 - 1\phi\lambda^2 + 27\lambda - 18$$

$$\lambda_1 = 1 \; ; \; \lambda_2 = 3 \; ; \; \lambda_3 = 6$$

<u>Lemma</u>: Each matrix satisfies its own characteristic polynomia[l] (Cayley Hamilton).

$$\Rightarrow \quad q(\lambda_i) = \lambda_i^3 - 10\lambda_i^2 + 27\lambda_i - 18 \equiv 0$$

<u>and</u>: $$q(A) = A^3 - 10A^2 + 27A - 18I \equiv 0$$

<u>Proof</u>:

$$q(A) = V \cdot q(\Lambda) \cdot V^{-1} \quad ; \quad \Lambda = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

$$\Rightarrow q(\Lambda) = \begin{bmatrix} q(1) & 0 & 0 \\ 0 & q(3) & 0 \\ 0 & 0 & q(6) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Rightarrow \quad q(A) \equiv 0$$ q.e.d.

<u>Lemma</u>: The Markov vector satisfies the characteristic polynomial.

Proof (Example):

$$q(A) = A^3 - 10A^2 + 27A - 18 = 0$$

$$\Rightarrow \underline{c}'\left[A^3 - 10A^2 + 27A - 18\right]\underline{b} = 0$$

$$\Rightarrow \underline{c}'A^3\underline{b} - 10\,\underline{c}'A^2\underline{b} + 27\,\underline{c}'A\underline{b} - 18\,\underline{c}'\underline{b}$$

$$\Rightarrow \boxed{\beta_4 - 10\beta_3 + 27\beta_2 - 18\beta_1 = 0}$$

q.e.d.

$\Rightarrow$ The higher Markov parameters are linearly dependent on the lower Markov parameters. The coefficients of this relation are the coefficients of the char. Polynomial.