Eigenvalue

wiki: https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

definition: a non-zero vector that changes by only a scalar factor when that linear transformation is applied to it

$A\overrightarrow{v}_i = \lambda\overrightarrow{v}_i$

In matrix form: $AV = V\Lambda$

if invertible: $A = V\Lambda V^{-1}$

property:

1. $(A - \lambda_i I)\overrightarrow{v}_i = 0$
2. $\det(A - \lambda_i I) = 0$
3. $tr(A) = \lambda_1 + \lambda_2 + \cdots + \lambda_n$
4. $\det(A) = \lambda_1\lambda_2\cdots\lambda_n$

real symmetric matrix:

1. eigenvalues are all real
2. eigenvectors are perpendicular to each other (orthogonality property): $(\lambda_i - \lambda_j)\overrightarrow{v}_i^T\overrightarrow{v}_j = 0$
3. (still orthogonality): $VV^T = I$

parameter

```
N0 = 5;
tmp1 = randn(N0,N0);
matA = tmp1 + tmp1.';
```

builtin (always check built-in method first)

```
% abbreviation: EVC is short for EigenVeCtor, EVL is short for EigenVaLue
[EVC,EVL] = eig(matA);
EVL = diag(EVL);
disp(EVC)
```

```
   -0.2108    0.0517    0.6826   -0.4707   -0.5151
    0.8028    0.3476    0.0556    0.2249   -0.4254
    0.3893   -0.7627   -0.2346   -0.4351   -0.1492
    0.0256    0.5416   -0.4665   -0.6962    0.0619
    0.3985    0.0375    0.5082   -0.2322    0.7263
```

```
disp(EVL)
```

```
   -4.6560
   -2.3943
    1.0714
    2.2270
    4.8616
```

```
ind_pick = 3;

% definition
disp([matA*EVC(:,ind_pick), EVL(ind_pick) * EVC(:,ind_pick)])
```

```
    0.7313    0.7313
    0.0596    0.0596
   -0.2513   -0.2513
   -0.4998   -0.4998
    0.5445    0.5445
```

```
%property-1
disp((matA-EVL(ind_pick)*eye(N0))*EVC(:,ind_pick))
```

```
   1.0e-15 *
   -0.4408
    0.3133
    0.3963
    0.4282
    0.1833
```

```
% property-2
disp(det(matA - EVL(ind_pick)*eye(N0)))
```

```
  -5.5625e-14
```

```
% property-3
disp([trace(matA), sum(EVL)])
```

```
    1.1097    1.1097
```

```
% property-4
disp([det(matA), prod(EVL)])
```

```
  129.3101  129.3101
```

Jacobi Method

wiki: https://en.wikipedia.org/wiki/Jacobi_eigenvalue_algorithm

Givens matrix: https://en.wikipedia.org/wiki/Givens_rotation

$$G_{i,j,\theta} = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \cos(\theta) & \cdots & -\sin(\theta) & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & \sin(\theta) & \cdots & \cos(\theta) & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}$$

procedure

1. Givens rotation: $A' = G_{i,j,\theta} A G_{i,j,\theta}^T$
2. $i(j)$ is the off-diagonal element row (column) with the largest absolute value (called the pivot)
3. $$\tan(2\theta) = \frac{2A_{ij}}{A_{jj} - A_{ii}}$$
4. repeatedly performs this rotation until the matrix becomes almost diagonal

property

1. A and A' have the Frobenius norm (the square-root sum of squares of all components)
2. $A'_{ij} = A'_{ji} = 0$
3. A' has a larger sum of squares on the diagonal

```
[EVC,EVL] = eig_jacobi(matA,30);
disp(EVC)
```

```
    0.6826   -0.2108   -0.0517    0.4707   -0.5151
    0.0556    0.8028   -0.3476   -0.2249   -0.4254
   -0.2346    0.3893    0.7627    0.4351   -0.1492
   -0.4665    0.0256   -0.5416    0.6962    0.0619
    0.5082    0.3985   -0.0375    0.2322    0.7263
```

```
disp(EVL)
```

```
    1.0714
   -4.6560
   -2.3943
    2.2270
    4.8616
```