Quadratic Eigenvalue Problems

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

Definition: solve $\left(M_0 + aM_1 + a^2M_2\right)\phi = 0$

TODO: what properties should matrix M_0, M_1, M_2 have.

equivalent to $\begin{cases} a\phi = a(\phi) \\ M_0\phi + aM_1\phi = -a^2M_2\phi \end{cases}$

write in matrix form: $\begin{bmatrix} & I \\ M_0 & M_1 \end{bmatrix}\begin{bmatrix} \phi \\ a\phi \end{bmatrix} = a\begin{bmatrix} I & \\ & -M_2 \end{bmatrix}\begin{bmatrix} \phi \\ a\phi \end{bmatrix}$, which is generalized eigenvalue equation (see wiki).

a more symmetry form: $\begin{bmatrix} & M_0 \\ M_0 & M_1 \end{bmatrix}\begin{bmatrix} \phi \\ a\phi \end{bmatrix} = a\begin{bmatrix} M_0 & \\ & -M_2 \end{bmatrix}\begin{bmatrix} \phi \\ a\phi \end{bmatrix}$.

a normal form: $\begin{bmatrix} & I \\ -M_2^{-1}M_0 & -M_2^{-1}M_1 \end{bmatrix}\begin{bmatrix} \phi \\ a\phi \end{bmatrix} = a\begin{bmatrix} \phi \\ a\phi \end{bmatrix}$.

solve it by call MATLAB built-in directly.

example

```
hf_symmetry = @(x) x + x.';
N0 = 3;
M0 = hf_symmetry(rand(N0));
M1 = hf_symmetry(rand(N0));
M2 = hf_symmetry(rand(N0)) + eye(N0)*N0/2; %make sure inversible

[EVC,EVL] = eig_quadratic(M0,M1,M2);
tmp1 = M0*EVC + (M1*EVC).*EVL.' + (M2*EVC).*(EVL.').^2;
disp(['eig_quadratic:: maximum absolute error: ', num2str(max(max(abs(tmp1))))])
```

```
eig_quadratic:: maximum absolute error: 2.979e-15
```

```
function [EVC,EVL] = eig_quadratic(M0,M1,M2)
% solve eigenvalue problem (M0 + a*M1 + a^2*M2)*phi = 0
% M0/M1/M2(N0,N0)
% EVC(N0,2*N0) EVL(2*N0,1)
% reference:
%   https://en.wikipedia.org/wiki/Quadratic_eigenvalue_problem
%   https://prod.sandia.gov/techlib-noauth/access-control.cgi/2007/072072.pdf
N0 = size(M0, 1);
tmp1 = eye(N0,N0);
matA = [zeros(N0,N0), tmp1; M0, M1];
matB = blkdiag(tmp1, -M2);
[tmp1,tmp2] = eig(matA,matB);
```

```
EVC = tmp1(1:N0,:);
EVL = diag(tmp2);
end
```