This bunch of codes solve the following modal equation of the system with continuation method under Harmonic Balance Method (HBM) assumption:

$$\begin{pmatrix} M_{ee} & M_{ec} \\ M_{ce} & M_{cc} \end{pmatrix} \begin{pmatrix} \ddot{e} \\ \ddot{x}_c \end{pmatrix} + \xi \begin{pmatrix} K_{ee} & K_{ec} \\ K_{ce} & K_{cc} \end{pmatrix} \begin{pmatrix} \dot{e} \\ \dot{x}_c \end{pmatrix} + \begin{pmatrix} K_{ee} & K_{ec} \\ K_{ce} & K_{cc} \end{pmatrix} \begin{pmatrix} e \\ x_c \end{pmatrix} + \begin{pmatrix} 0 \\ g(x_c) \end{pmatrix} = \begin{pmatrix} F_e \\ F_c \end{pmatrix} \cdot f(t) + \begin{pmatrix} P_e \\ P_c \end{pmatrix} \tag{1}$$

where: e are displacements of elastic part, x_c are the displacements of contact part, M and K are the mass and stiffness matrices from Finite Element Method (FEM), $g(x_c)$ are the Coulomb friction forces acting only on contact part, F_e and F_c are the amplitudes coefficients of external harmonic forces acting on elastic and contact part, P_e and P_c are the preload forces acting on elastic and contact part. The equation (1) should be modified before solving it.

First, calculate the elastic preload displacement x_e^0:

$$\begin{pmatrix} K_{ee} & K_{ec} \\ K_{ce} & K_{cc} \end{pmatrix} \begin{pmatrix} x_e^0 \\ 0 \end{pmatrix} = \begin{pmatrix} F_e \\ F_c \end{pmatrix} + \begin{pmatrix} 0 \\ R \end{pmatrix} \tag{2}$$

where R is the reactions at the contacts.

So the elastic displacement x_e based on x_e^0 can be written as:

$$\begin{cases} x_e = e - x_e^0 \\ \dot{x}_e = \dot{e} \\ \ddot{x}_e = \ddot{e} \end{cases} \tag{3}$$

substitute equation (3) to (1), we get:

$$\begin{pmatrix} M_{ee} & M_{ec} \\ M_{ce} & M_{cc} \end{pmatrix} \begin{pmatrix} \ddot{x}_e \\ \ddot{x}_c \end{pmatrix} + \xi \begin{pmatrix} K_{ee} & K_{ec} \\ K_{ce} & K_{cc} \end{pmatrix} \begin{pmatrix} \dot{x}_e \\ \dot{x}_c \end{pmatrix} + \begin{pmatrix} K_{ee} & K_{ec} \\ K_{ce} & K_{cc} \end{pmatrix} \begin{pmatrix} x_e \\ x_c \end{pmatrix} + \begin{pmatrix} 0 \\ g(x_c) \end{pmatrix} = \begin{pmatrix} F_e \\ F_c \end{pmatrix} \cdot f(t) + \begin{pmatrix} 0 \\ -R \end{pmatrix} \tag{4}$$

Second, do the Criag-Bampton Reduction to the equation (4) above, apply:

$$\begin{pmatrix} x_e \\ x_c \end{pmatrix} = \begin{pmatrix} \Phi & \Psi \\ 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} a \\ x_c \end{pmatrix} = T_{cb} \begin{pmatrix} a \\ x_c \end{pmatrix} \tag{5}$$

to equation (4), where Ψ is constrained modes, $\Phi$is full-stuck elastic modes, we get:

$$\begin{pmatrix} \mathbf{I} & M_{ax} \\ M_{xa} & M_{xx} \end{pmatrix} \begin{pmatrix} \ddot{a} \\ \ddot{x}_c \end{pmatrix} + \xi \begin{pmatrix} K_{aa} & 0 \\ 0 & K_{xx} \end{pmatrix} \begin{pmatrix} \dot{a} \\ \dot{x}_c \end{pmatrix} + \begin{pmatrix} K_{aa} & 0 \\ 0 & K_{xx} \end{pmatrix} \begin{pmatrix} a \\ x_c \end{pmatrix} + \begin{pmatrix} 0 \\ g(x_c) \end{pmatrix} = \begin{pmatrix} F_a \\ F_x \end{pmatrix} \cdot f(t) + \begin{pmatrix} 0 \\ -R \end{pmatrix} \tag{6}$$

Third, calculate the contact preload displacement x_p:

$$\begin{pmatrix} K_{aa} & 0 \\ 0 & K_{xx} \end{pmatrix} \begin{pmatrix} a \\ x_p \end{pmatrix} + \begin{pmatrix} 0 \\ g(x_p) \end{pmatrix} = \begin{pmatrix} 0 \\ -R \end{pmatrix} \tag{7}$$

So the contact displacement x based on x_p is:

$$\begin{cases} x = x_c - x_p \\ \dot{x} = \dot{x_c} \\ \ddot{x} = \ddot{x_c} \end{cases} \tag{8}$$

substitute equation (8) to (6), we get the final form of the modal equation to be solved:

$$\begin{pmatrix} \mathbf{I} & M_{ax} \\ M_{xa} & M_{xx} \end{pmatrix} \begin{pmatrix} \ddot{a} \\ \ddot{x} \end{pmatrix} + \xi \begin{pmatrix} K_{aa} & 0 \\ 0 & K_{xx} \end{pmatrix} \begin{pmatrix} \dot{a} \\ \dot{x} \end{pmatrix} + \begin{pmatrix} K_{aa} & 0 \\ 0 & K_{xx} \end{pmatrix} \begin{pmatrix} a \\ x \end{pmatrix} + \begin{pmatrix} 0 \\ g(x+x_p) - g(x_p) \end{pmatrix} = \begin{pmatrix} F_a \\ F_x \end{pmatrix} \cdot f(t) \tag{9}$$

Fourth, apply the Fourier Transformation to CB modes a, contact displacement x and friction forces $g(x + x_p) - g(x_p)$:

$$\begin{cases} a(t) = \frac{A_c^0}{2} + \sum_{k=1}^{\infty} A_c^k \cdot \cos(k\Omega t) + A_s^k \cdot \sin(k\Omega t) \\ x(t) = \frac{X_c^0}{2} + \sum_{k=1}^{\infty} X_c^k \cdot \cos(k\Omega t) + X_s^k \cdot \sin(k\Omega t) \\ g(x(t) + x_p) - g(x_p) = \frac{G_c^0}{2} + \sum_{k=1}^{\infty} G_c^k \cdot \cos(k\Omega t) + G_s^k \cdot \sin(k\Omega t) \end{cases} \tag{10}$$

External forces are assumed harmonic forces, so we can define the Fourier coefficients directly. Substitute (10) to equation (9), we can get following functions, equation (11)-(13), for continuation method:

$$FUN(X) = \begin{pmatrix} K_{aa} - k^2\Omega^2 \mathrm{I} & k\Omega\xi K_{aa} & -k^2\Omega^2 M_{ax} & 0 \\ -k\Omega\xi K_{aa} & K_{aa} - k^2\Omega^2 \mathrm{I} & 0 & -k^2\Omega^2 M_{ax} \\ -k^2\Omega^2 M_{xa} & 0 & K_{xx} - k^2\Omega^2 M_{xx} & k\Omega\xi K_{xx} \\ 0 & -k^2\Omega^2 M_{xa} & -k\Omega\xi K_{xx} & K_{xx} - k^2\Omega^2 M_{xx} \end{pmatrix} \begin{pmatrix} A_c^k \\ A_s^k \\ X_c^k \\ X_s^k \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ G_c^k \\ G_s^k \end{pmatrix} - \begin{pmatrix} F_{ac}^k \\ F_{as}^k \\ F_{xc}^k \\ F_{xs}^k \end{pmatrix} \tag{11}$$

where X in $FUN(X)$ are the Fourier coefficients which are $A_c^k, A_s^k, X_c^k, X_s^k$.

$$\frac{\partial FUN(X)}{\partial X} = \begin{pmatrix} K_{aa} - k^2\Omega^2 \mathrm{I} & k\Omega\xi K_{aa} & -k^2\Omega^2 M_{ax} & 0 \\ -k\Omega\xi K_{aa} & K_{aa} - k^2\Omega^2 \mathrm{I} & 0 & -k^2\Omega^2 M_{ax} \\ -k^2\Omega^2 M_{xa} & 0 & K_{xx} - k^2\Omega^2 M_{xx} & k\Omega\xi K_{xx} \\ 0 & -k^2\Omega^2 M_{xa} & -k\Omega\xi K_{xx} & K_{xx} - k^2\Omega^2 M_{xx} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\partial G}{\partial X} \end{pmatrix} \tag{12}$$

$$\frac{\partial FUN(X)}{\partial \Omega} = \begin{pmatrix} -2k^2\Omega \mathrm{I} & k\xi K_{aa} & -2k^2\Omega M_{ax} & 0 \\ -k\xi K_{aa} & -2k^2\Omega \mathrm{I} & 0 & -2k^2\Omega M_{ax} \\ -2k^2\Omega M_{xa} & 0 & -2k^2\Omega M_{xx} & k\xi K_{xx} \\ 0 & -2k^2\Omega M_{xa} & -k\xi K_{xx} & -2k^2\Omega M_{xx} \end{pmatrix} \begin{pmatrix} A_c^k \\ A_s^k \\ X_c^k \\ X_s^k \end{pmatrix} \tag{13}$$

So the codes are basically to implement above equations. In code structure, each model to be analyzed has its own folder, such as "two-solid system" and "dof5_model" in the examples. Inside the model's folder, there is a main.m file to call every other functions, the structure of the codes is showed in Figure 1. To run the code successfully, we need to add "HBM Continuation Code mex" to matlab path.

In main.m file, there are 5 main parts.

First, the data.m file which in the same folder with main.m, we finish parameters setting. Including:

- FEM matrices and vectors: M and K matrices, F and P vectors from the equation(1). We can write them directly inside the data.m file or read them from other file like csv file. Finally the M, K, F and P should be stored in the form of Matlab struct named as FEM in Workspace. Because $M_{ec} = M_{ce}^{\mathrm{T}}$, $K_{ec} = K_{ce}^{\mathrm{T}}$, we only keep M_{ec} and K_{ec} to save the memory. They can be accessed like $FEM.Mee$, $FEM.Mec$, $FEM.Mcc$, $FEM.Kee$, $FEM.Kec$, $FEM.Kcc$, $FEM.Fe$, $FEM.Fc$, $FEM.Pe$, $FEM.Pc$.

- Harmonics of external forces: named as H_F_ext in Matlab Workspace. The elements inside the vector in the order of Fourier coefficient $H\_F\_ext = [a_0, a_1, b_1, a_2, b_2, ..., a_n, b_n]$. For example, if the external

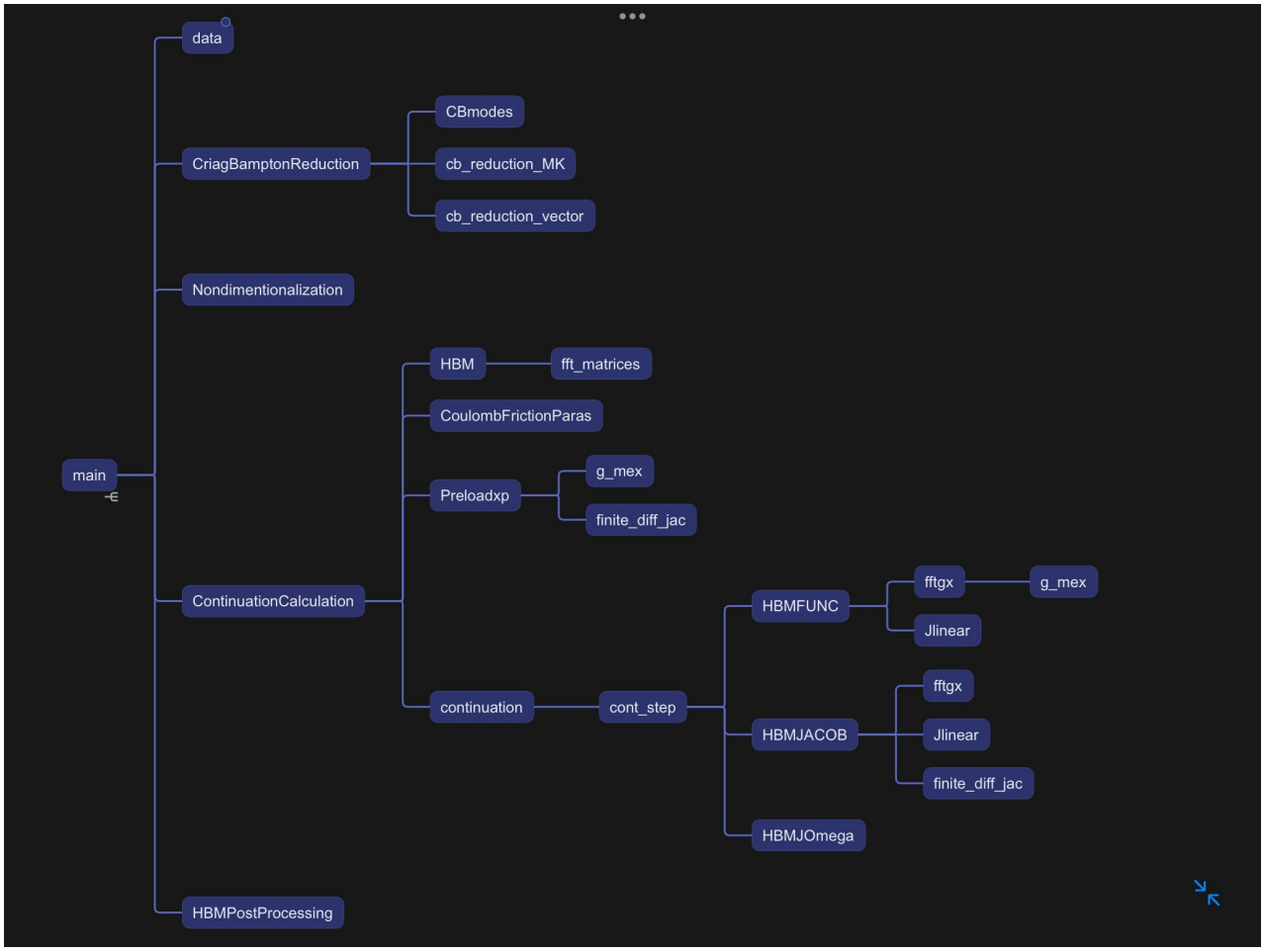

Figure 1: Structure of the codes

forces $f(t) = \cos(\Omega t) + \sin(2\Omega t)$, then H_F_ext= $[0, 1, 0, 0, 1]$.

- HBM parameters: H, number of harmonics assumption, $H = k$ in equation (10), should be bigger than H_F_ext; N, discretized time points per external forces cycle, $t = [0, N-1] \cdot \frac{2\pi}{k\Omega}$ in equation (10); Nx, number of contact points, means having $3Nx$ dofs (unknows) in contact part; Na, number of CB modes; xi is ξ in modal equation (1).

- Newton Method parameters: $epsx$ and $epsy$ are the tolerances of independent variables x and dependent variables$y$ in Newton iteration , $maxiter$ is the maximum iterations in continuation method and preload calculation.

- Coulomb friction parameters: kn, stiffness in normal direction of each contact point; $xn0$, normal direction gaps of each contact point; mu, friction factor in two tangential directions of each contact point; kt, stiffness in two tangential directions before sliding of each contact point. kn and $xn0$ have the size of $1 \times Nx$ (row vector) or 1×1 (scale), we can define the different parameters for each contact point in row vector, or keep the same values for each contact point in scale type. mu and kt have the size of $2 \times Nx$ (matrix) or 2×1 (column vector), the first row are the parameters of first tangential direction, the second row are for the second direction.

- Preload initial condition for Newton method: $xp0$ for contact preload calculation in equation (7), which solved by Newton iteration. It's an optional setting, if no value defined, the default value inside is 0.

- Continuation parameters: ds, step length to the next continuation step, sometimes need to set smaller if calculation diverge. $maxstep$, maximum steps of continuation; $omega\_0$ and $omega\_end$ are the beginning and the final Ω of the continuation calculation; $x0$, initial value of the X in $FUN(X)$ of the equation (11), is an optional setting. Default value of $x0$ is 0 column vector, in the size of $(Na + 3Nx) \times (2H + 1)$.

All the variables in data.m file described above should be stored in Matlab Workspace.

Second, Criag-Bampton (CB) reduction from FEM matrices. We can call the CriagBamptonReduction.m to get the M, K matrices and F vectors in equation (6). To finish the CB reduction, we need the FEM struct in the Matlab Workspace stored in data.m before. If run the CriagBamptonReduction.m successfully, we can get CB struct, $xe0$ and Rx vectors stored in Matlab Workspace and also at the same time saved in Current Folder in Matlab as ".mat" form so that can be used in next calculation. Which $xe0$ is a column vector with the size of degree of freedoms (dofs) of elastic part, Rx is a column vector with the size of $3Nx$, CB struct is illustrated in (14).

$$CB. \begin{cases} CBmodes. & \begin{cases} Phi \\ Psi \\ K_{aa} \end{cases} \\ CB\_MK. & \begin{cases} M_{ax} \\ M_{xx} \\ K_{xx} \\ K_{aa} \end{cases} \\ CB\_F. & \begin{cases} Fa \\ Fx \end{cases} \end{cases} \tag{14}$$

Inside CriagBamptonReduction.m, there are three parts:

- CBmodes function: calculate the full-stuck elastic modes and constrain modes of the equation (1), full-stuck elastic modes Φ are calculated by eigenvalue problem in equation (15), which only keeps the modes of the number Na defined in data.m, and constrained modes Ψ are calculated by static equilibrium equation (16).

$$(-\omega^2 M_{ee} + K_{ee})\Phi = 0 \tag{15}$$

$$\begin{aligned} &K_{ee}e + K_{ec}x_c = 0 \\ &e = -K_{ee} \backslash K_{ec}x_c = \Psi x_c \end{aligned} \tag{16}$$

- cb_reduction_MK function: make the Braig-Bampton reduction of M and K matrices in the equation (1) using the transformation matrix T_{CB} in equation (5), which:

$$\begin{aligned} M_{ax} &= M_{xa}^{\mathrm{T}} = \Phi^{\mathrm{T}} M_{ee}\Psi + \Phi^{\mathrm{T}} M_{ec} \\ M_{xx} &= \Psi^{\mathrm{T}} M_{ee}\Psi + \Psi^{\mathrm{T}} M_{ec} + M_{ce}\Psi + M_{cc} \\ K_{xx} &= K_{cc} - \Psi^{\mathrm{T}} K_{ee}\Psi \\ K_{aa} &= \mathrm{diag}(\omega_j^2) \end{aligned} \tag{17}$$

- cb_reduction_vector function: make the Briag-Bampton reduction of column vector F in the equation (1), which:

$$\begin{aligned} F_a &= \Phi^{\mathrm{T}} F_e \\ F_x &= \Psi^{\mathrm{T}} F_e + F_c \end{aligned} \tag{18}$$

If CB matrices are already calculated, we can directly read them from Current Folder or other file. In this case, FEM matrices are not needed in previous processes.

Third, Nondimensionalization. We can define the different dimensionless parameters to different model, so this file stays together with main.m file. If previous processes implement successfully, we have CB struct (illustrated in (14)) , R, kn, kt and xi in Workspace of Matlab, and can directly run Nondimensionalization to make the nondimensionalization of equation (9) using the following changes:

$$\begin{aligned} t &\to \tfrac{1}{\omega_0} t \\ a &\to \alpha a \\ x &\to \beta x \end{aligned} \tag{19}$$

where ω_0 is selected to match that of the resonance of interest, $\alpha = Fa/\omega_0^2$, $\beta = \mu \max(|R|)/k_t$, then we change the equation (6) to equation :

$$\begin{aligned} &\begin{pmatrix} \mathbf{I} & \frac{\beta}{\alpha} M_{ax} \\ \frac{\beta}{\alpha} M_{xa} & \frac{\beta^2}{\alpha^2} M_{xx} \end{pmatrix} \begin{pmatrix} \ddot{a} \\ \ddot{x} \end{pmatrix} + (\xi\omega_0)\xi \begin{pmatrix} \frac{1}{\omega_0^2} K_{aa} & 0 \\ 0 & \frac{\beta^2}{\alpha^2\omega_0^2} K_{xx} \end{pmatrix} \begin{pmatrix} \dot{a} \\ \dot{x} \end{pmatrix} \\ &+ \begin{pmatrix} \frac{1}{\omega_0^2} K_{aa} & 0 \\ 0 & \frac{\beta^2}{\alpha^2\omega_0^2} K_{xx} \end{pmatrix} \begin{pmatrix} a \\ x \end{pmatrix} + \frac{\beta^2}{\alpha^2\omega_0^2} \begin{pmatrix} 0 \\ g(x + x_p) - g(x_p) \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha{\omega_0}^2} F_a \\ \frac{\beta}{\alpha^2\omega_0^2} F_x \end{pmatrix} \cdot f(t) \end{aligned} \tag{20}$$

Fourth, continuation calculation part, after all the parameters setting, we have all we need in Workspace of Matlab. So we can do the calculation and get the x_cont and $omega\_cont$ frequency domain results in Workspace for post processing in the fifth part, where x_cont is a matrix with row size $(2H+1)\cdot(Na+3Nx)$ in the order of dofs, each column stores the the Fourier coefficients results under different frequency of external forces; $omega\_cont$ is a row vector stores accordingly the frequency of external forces of each continuation step.

Fifth, in the HBM post processing, we do the ifft to convert the result in time domain to make the analysis. In the file HBMPostProcessing.m, we can define which DOF to plot by setting the variable $Ndof$ to plot the amplitude over each Ω.

Except data.m, Nondimensionalization.m and HBMPostProcessing.m, other functions and files are located in the folder "HBM Continuation Code mex", which using g_mex function to calculate the friction forces coded in C++ and compiled by matlab.

Other folders, "HBM Continuation Code Matlab" has the g function coded in 100% matlab language which we can see the calculation logic. But it's much slower than using g_mex.

We can create a now folder for the now model each time, then copy the main.m, data.m, Nondimensionalization.m and HBMPostProcessing.m as templates that paste them to the folder and modify.