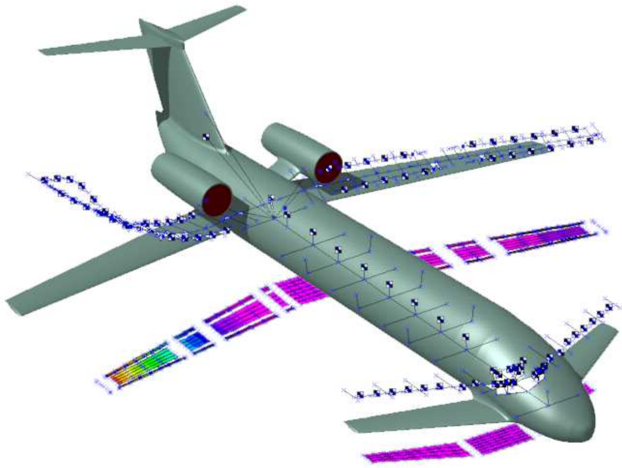
NeoCASS

Next generation Conceptual Aero Structural Sizing



Dipartimento di Scienze e Tecnologie Aerospaziali
Politecnico di Milano

Report name	NeoCASS Next generation Conceptual Aero Structural Sizing
Date	December 11, 2013
Partner Politecnico di Milano	Dipartimento di Scienze e Tecnologie Aerospaziali
Author	L. Cavagna, S. Ricci
Delivery number	D.XXXX
Dissemination level	Public

Contents

List of Figures

List of Tables

Appendix A

NeoRESP: aeroelastic dynamic response

A.1 Flight condition

Three parameters are required in order to define the flight reference condition:

1. VREF: flight speed V_∞ (TAS) in m/s;
2. RHOREF: air density ρ_∞ in Kg/m^3.
3. MACH: flight Mach number M_∞.

NeoRESP does not check for the consistency of the parameters given, i.e. Mach number, velocity and density need not have any dependency.
The parameters are required to define all the terms for the aeroeasltic response:

$$\left(-\omega^2\mathbf{M} + i\omega\mathbf{C} + \mathbf{K} - q_\infty \mathbf{H}_{mm}(k, M)\right)\mathbf{q}(\omega) = q_\infty \mathbf{H}_{mg}(k, M)\,\mathbf{V}\mathbf{g}(\omega) \tag{A.1}$$

Indeed, the dynamic pressure $q_\infty = \frac{1}{2}\rho_\infty V_\infty^2$ is determined by VREF and RHOREF while the reduced frequnecy k in $\mathbf{H}_{mm}(k, M)$ and $\mathbf{H}_{mg}(k, M)$ depends on VREF as $k = \frac{\omega\,\hat{c}}{V_\infty}$.
The parameter MACH is used only to define $H_{mm}(k, M)$ and $H_{mg}(k, M)$. A zeroth order approximation is adopted by NeoRESP which selects among the aerodynamic matrices available in the database those having Mach number M closest to MACH. No interpolation is carried out at the moment.
An example of reference condition is the following:

```
PARAM   VREF    238.0
PARAM   RHOREF  1.225
PARAM   MACH    0.7
```

for a case of flight at $M_\infty = 0.7$, $\rho_\infty = 1.225$ Kg/m^3 and $V_\infty = 238.0$ m/s.

A.2 Gust input

NeoRESP has a very flexbile and general way to prescribe a gust. The gust can be lateral or vertical and can assume any kind of profile. The card GUST provides all the data required. The parameter SFUN allows to define how the gust is distributed spanwise.

GUST	ID	WG	T	X0	AX				
	SFUN								
	TFUN								

ID	Card identifier
MAG	Gust scale factor
T	Gust period [s]
X0	Streamwise position of the gust reference point (spatial offset)
AX	2 for lateral gust along y axis, 3 for vertical gust along z axis
SFUN	General user-defined function to provide non-uniform gust. Function of y or x if AX=3 or AX=2 respectively.
TFUN	General user-defined function of time t to model gust profile

Table A.1: GUST definition

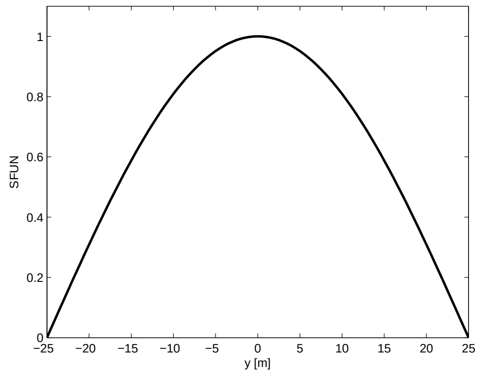


Figure A.1: Gust spatial scaling factor SFUN.

For the case of uniform gust, SFUN== 1. For non-uniform gust, consider for example an aircraft with wingspan of 40 m travelling across a gust with spatial length of 50 m. The function will be evaluated along the aerodynamic mesh which is symmetric ans span from -20 m to 20 m. A half-sine gust spanning from to -25 m to 25 m can be provided as:

$$\mathrm{SFUN} = \sin\left(\pi\ \cdot (x + 25)/50\right) \tag{A.2}$$

SFUN is supposed to be a scaling factor, thus its magintude should be unity.
The parameters MAG, T and TFUN concurr in defining the gust velocity profile VG as:

$$\mathrm{VG} = \mathrm{MAG} \cdot \mathrm{TFUN}, \quad \text{for} \quad 0 \leq t \leq \mathrm{T} \tag{A.3}$$

where V_∞ is the flight speed. VG is a function of time and defines how the gust velocity changes on each aerodynamic panel. The function is evaluated from $0 < leq$T. If X0 is given (usually a negative value), the gust origin will be moved in that point as depicted in Fig. A.2. There will be hence a time delay equal to the ratio between X0 and the flight speed V_∞ before the gust hits the aircraft. Suppose the following uniform vertical gust has to be modelled:

$$VG = 17.07 \left(\frac{1}{2}\left(1 - \cos\left(2 \cdot \pi/0.12 \cdot t\right)\right)\right), \quad 0 \leq t \leq 0.12 \tag{A.4}$$

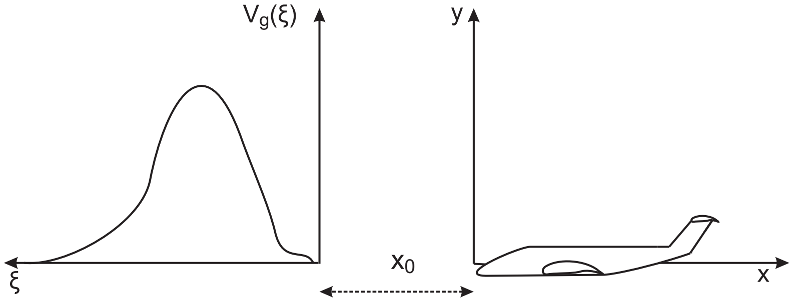


Figure A.2: Gust offset X0.

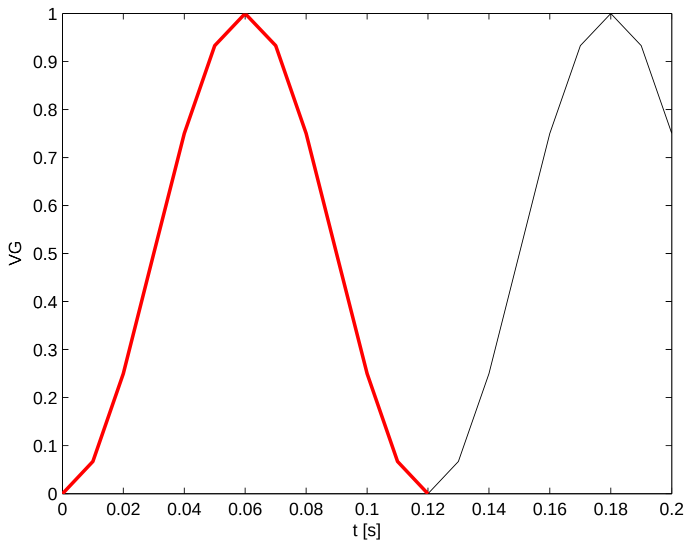


Figure A.3: Gust profile VG.

with a period $\tau = 0.12$ s and a velocity magnitude of 17.07 m/s. For this case, SFUN will be defined as:

$$\text{SFUN} = (1 - \cos(2 \cdot \pi/0.12 \cdot t)) \cdot 0.5 \tag{A.5}$$

while T will be set to 0.12 s and MAG to 17.07 m/s. The GUST card will be:

```
GUST    1       17.07   0.12    0.0     3
        1
        (1-cos(2*pi/0.12*t))*0.5
```

The function given in Eq. (A.4) will be evaluated from 0 to T. Thus, referring to Fig. A.3, only the red line will be considered and the input signal will be padded with zeros.

To require the application of the gust, the command line `GUST= ID`, where ID is the identification number of the GUST card, is given. The case of multiple gusts is at the moment not allowed.

A.3 Control surface input

NeoRESP can solve for control surface prescribed motion. The card SURDEF provides the required information, in a similar way to the GUST case: NAM is the name of the

SURFDEF	ID	NAM	MAG	T	DLY				
	TFUN								

ID	Card identifier
NAM	Master surface name
MAG	Deflection scale factor (degrees)
T	Function period [s]
DLY	Time delay [s]
TFUN	General user-defined function to provide deflection in time

Table A.2: SURFDEF definition

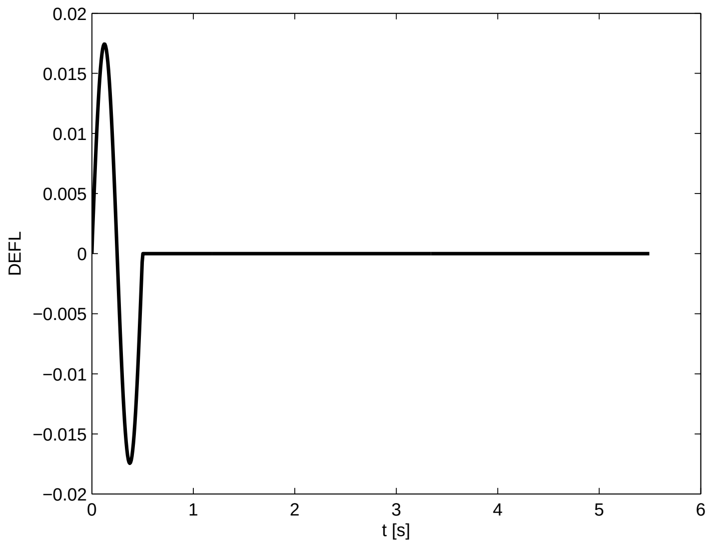


Figure A.4: Elevator deflection DEFL (in radians).

master control surface reported in `beam_model.Aero.Trim.Link.Master`.
T is the time duration of the function. When $t >$T, the input signal will be padded with zeros. DLY introduce a delay in the signal which will be applied after DLY seconds. Finally, MAG is the magnitude of the deflection in degrees and scales the function TFUN. According to the information reported, the deflection DEFL will be recovered as:

$$\mathrm{DEFL} = \mathrm{MAG} \cdot \mathrm{TFUN}, \quad \text{for} \quad \mathrm{DLY} \leq t \leq \mathrm{T} + \mathrm{DLY} \tag{A.6}$$

Considering an example of elevator deflection, the SURFDEF card will be:

```
SURFDEF 1       elev1r  1.0     0.5     0.0
        sin(2*pi/0.5*t)
```

In this case one unitary sinusoidal deflection is given, with period T= 0.5 s, as reported in Fig. A.4. The card is select by the command `SURFDEF= ID`, where `ID` is the identification number of the SURFDEF card. Similarly to the case of gust, multiple control inputs is at the moment not allowed. A second option allows to provide the law as a time-dependent piece-wise function. NAM is the name of the master control surface reported in `beam_model.Aero.Trim.Link.Master`.
Ti is the time duration of each function. Each function is evaluated between 0 and Ti.

SURFDEF	ID	NAM	MAG		NP				
	DLY1	T1	FUN1						
	DLY2	T2	TF2						
	DLYN	TN	TFN						

ID	Card identifier
NAM	Master surface name
MAG	Deflection scale factor (degrees)
NP	Number of piece-wise functions
DLYi	Time offset (seconds) for piece i
Ti	Time duration (seconds) for piece i
TFi	General user-defined function to provide deflection in time. Time goes from 0 to Ti.

Table A.3: SURFDEF definition

The result is then shifted in time at DLYi. When $t >$T, the input signal will be padded with zeros.
Finally, MAG is the magnitude of the deflection in degrees and scales all the functions provided. According to the information reported, the deflection DEFL will be recovered as:

$$\begin{aligned} \text{DEFL} &= \text{MAG} \cdot \text{TF1}, \quad \text{for} \quad \text{DLY1} \leq t \leq \text{T1} + \text{DLY1} \\ \text{DEFL} &= \text{MAG} \cdot \text{TF2}, \quad \text{for} \quad \text{DLY2} \leq t \leq \text{T2} + \text{DLY1} \\ \text{DEFL} &= \text{MAG} \cdot \text{TFN}, \quad \text{for} \quad \text{DLYN} \leq t \leq \text{TN} + \text{DLYN} \end{aligned}$$

If the tail of the former function does not match with the begin of the latter function, DEFL will be null. Considering an example of elevator deflection, the SURFDEF card will be:

```
SURFDEF 1       rudder1 1.0     0       3
        0       1       (1-cos(2*pi*t))*0.5
        1       1       0
        2       1       -(1-cos(2*pi*t))*0.5
```

In this case one unitary 1-cos deflection is given, with period T= 1 s. The command is left unrotated for T= 1 s and finally deflected again as 1-cos in the opposite direction of the first motion. The time law is reported in Fig. A.5.

A.4 External force

NeoRESP includes dynamic external loads $\mathbf{p}(\omega)$ in the aeroelastic response:

$$\left(-\omega^2\mathbf{M} + i\omega\mathbf{C} + \mathbf{K} - q_\infty\mathbf{H}_{mm}(k, M)\right)\mathbf{q}(\omega) = \mathbf{p}(\omega) \tag{A.7}$$

ID represents the identification number of the load set. Multiple card with the same ID can hence be given. NID is the node loaded at its degree of freedom DOF. MAG scales the load function TFUN which prescribes how the load varies with time. TFUN is evaluated between 0 and T and then padded with zeros. DLY shifth in time the applied

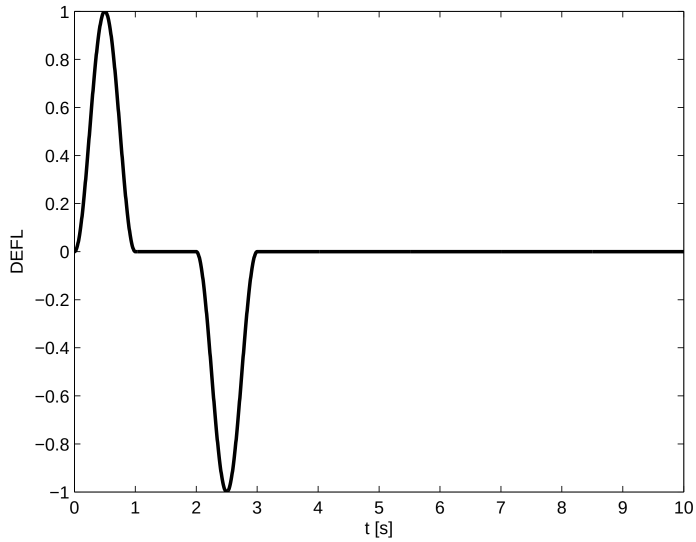


Figure A.5: Elevator deflection DEFL (in radians).

DLOAD	ID	NID	DOF	MAG	T	DLY			
	TFUN								

ID	Card identifier
NID	Node loaded
DOF	DOF loaded, from 1 to 6
MAG	Scale factor
T	Function period [s]
DLY	Time delay [s]
TFUN	General user-defined function to provide load in time

Table A.4: DLOAD definition

load.
Thus, the load will be:

$$\text{DLOAD} = \text{MAG} \cdot \text{TFUN}, \quad \text{for} \quad \text{DLY} \leq t \leq \text{T} + \text{DLY} \tag{A.8}$$

The load set ID is considered by including the command line `DLOAD= ID`.
A simple example of DLOAD is:

```
DLOAD   1       2000    3       1.0e+3  0.5     0.0
        sin(2*pi/0.5*t)
```

which loads node 2000 with a sinusolidal vertical force of 1000 N and period 0.5 s (similar to Fig. A.4).

A.5 Mode selection: U

When the eigenvalue solver in SMARTCAD is run, modes are determined and selected to form the modal base **U**. The EIGR provides inputs to the eigenvalue solver and allows a first selection of vibration modes. The selection can be defined trough:

EIGR	ID		F1	F2		ND			
	NRM	G	C						

ID	Card identifier
G	Node identifier used to normalize modes with POINT normalization
C	Generalized displacement component used to normalize modes with POINT normalization
NRM	Normalization criterion: MASS (default value), MAX, POINT MASS: normalize to unit generalized mass matrix MAX: normalize to unit value of the largest component in the analysis set POINT: normalize to unit value of the C component for node G
F1,F2	Frequency (Hz) range of interest. All the eigenmodes in this range will be used Default values are: F1=0.0, F2=∞
ND	Desidered number of modes. If null, all the modes in the frequency range will be used

Table A.5: EIGR eigenvalue solution parameters

1. providing a frequnecy range of interest limited by F1 and F2;

2. specifying the number of the first ND roots to extract.

If ND is not given, all the modes will be determined. This may be time-consuming. The modes having a natural frequnecy F1 $\leq \Omega_i \leq$ F2 will be retained. On the other hand, if ND is defined the library ARPACK will be used. ARPACK is a collection of routines available in MATLAB for large scale sparse eigenvalue problems. It allows to extract the first lowest ND roots of the problem, thus reducing computational costs. Once the eigenvalues have been extracted, only those satisfying F1 $\leq \Omega_i \leq$ F2 will be reatined. Of course if the frequency of the last ND-th root is below FMAX, no truncation will occur. The user may want to increase the value of NROOTS in this case, to be sure the desidered frequency range is fully covered.
The results from the modal analysis are saved in `beam_model.Res` as:

- `beam_model.Res.ID` with mode IDs;

- `beam_model.Res.Mmm, beam_model.Res.Mmm` with generalized mass $\mathbf{M}$ and stiffness $\mathbf{K}$ matrices;

- `beam_model.Res.Omega` with modal circular frequnecy $\mathbf{\Omega}$;

- `beam_model.Res.NDispl` with modal shapes $\mathbf{U}$.

A.5.1 User-defined modal base

After a first run, the user may want to perform a further selective choice of the modes. This is because it turns out to be impossibile to discard modes only on the basis of their natual frequnecy. A criteria based on their shapes is required.

A typical case is represented by in-plane bending modes, or by symmetric modes when the full model is considered and antysimmetric modes have to be discarded because the input force itself is symmwetric. The card MSELECT allows to discard the modes retained from the first process introduced above (based on ND and/or F1,F2). The ID of the modes forming the final modal base **U** are listed in the dedicated card MSELECT. Thus, to

MSELECT	M1	M2	M3	M4	M5	M6	M7	M8	
	M9	etc							

Mi	Mode index used (THRU and EXCEPT can be used)

Table A.6: MSELECT modal base definition.

discard the modes n.7, 8, 10 from the modal database ranging from 1 to 12, the MSELECT card will be:

```
MSELECT 1       2       3       4       5       6       9       11
        12
```

A second way to proceed is given by THRU and EXCEPT options. The former is used to provide a range from the node at its left and the node at its right. The latter is used to discard nodes from the those prescribed by THRU. Using this option the card will be:

```
MSELECT 1       THRU    12      EXCEPT  7       8       10
```

If the nodes after EXCEPT are not contained in the set given by THRU, they will be included. Thus:

```
MSELECT 1       THRU    3       EXCEPT  5
```

will give 1, 2, 3, 5.
After file reloading and solver re-launching, the new modal base **U** will be made of modes 1, 2, 3, 4, 5, 6, 9, 11, 12 and satisfy F1 $\leq \Omega_i \leq$ F2.

A.5.2 Modes for flutter assessment: $\mathbf{U}_f$

In some cases, only a subset of modes can be assessed for flutter. The flutter algorithm considers each mode separately at low speed and predicts its damping and frequency at increasing values of velocity. In some cases, the stability of few modes may not be of interest and modes can be skipped during the tracking process. This could be the case of rigid body modes or in-plane modes. In the first case, equations should be re-written in stability axes to correctly evaluate rigid dynamic modes. For the second case, the prediction of aerodynamicc force could be rather inaccurate and lead to a mode whioch has no damping. The card FMODES allows to define the modes to be assessed. The syntax is the same as the MSELECT card.

A.5.3 Modes for dynamic response: $\mathbf{U}_d$

Modes from the initial modal base **U** can be retained or discarded when performing a dynamic response problem. This new set defines the dynamic modal base $\mathbf{U}_d$. By this way, the influence of modal truncation on the final solution can be estimated by reducing or raising the modal bandwidth considered. Also, some modes can be discarded because they are considered as negligible in the dynamic solution: for example antysimmetric modes, or rigid body mode along free-stream velocity, in plane-modes.

The dynamic equations are originally written for $\mathbf{U}$ and are successively reduced in size according to $\mathbf{U}_d$. Rows and columns corresponding to the $\mathbf{U}_d$ modes are hence retained. A dedicated card UMODES is available to list the modes to be used. In principle UMODES is similar to MSELECT. The only difference is that MSELECT defines once for all the modes of interest to be considered. This means they will always appear in flutter or dynamic response equations, e.g. their aerodynamic contribution will be always evaluated. UMODES simply reduces the number of degrees of freedom before the simulation is run. The syntax is the same as MSELECT and FMODES. Rigid mode along x axis must not be included in $\mathbf{U}_d$ to avoid singularities.

A.6 Modes acceleration

Accurate dynamic load prediction is a critical process for structural design and assessment. The dynamic response analysis usually requires a reduction of the number of degrees of freedoms (DOFs) to limit the computational costs. Modal response or more superposition method is extensively used in structural dynamics and aeroelastic analisys. In this case, mode truncation enables to reduce the size of the problem according to the frequency content of interest.
Dynamic analysis based on modal superposition is tipically performed in three steps:

- a set of vibration modes of the system $\mathbf{U}$ are determined and used to define the generalized modal coordinates $\mathbf{q}$;
- dynamic response $\mathbf{q}(t)$ is calculated;
- physical responses $\mathbf{y}$ of interest are recovered from the modal solution.

In order for the accuracy of modal superposition to be accurate, corrections are usually taken into account. With mode truncation some information is necessarily lost. The load representation is altered as well as the quality of the response. The truncated information representing the high frequency modes of the structure does not have a large impact on the overall response, having a small dynamic excitation. On the other hand, the high frequency modes are essential in accurately defining the steady-state portion of the response, which affects the accuracy of element loads, stresses and displacements.
The Mode Displacement (MD) method which recovers these data from the adopted modes turns out to be inaccurate, since it cannot capture local effects associated with concentrated loads or local stresses. The choice of the number of modes to retain is tipically made in relation to the frequency content of the appliced load. This criterion addresses time resolution but not the spatial one. Inaccuracies in the response occur because the effects of modal truncation on the spatial representation of the applied load are not accounted for. The load truncation can be reduced by increasing the number of modes retained until convergence. Unfortunately, the required addition can be significant and impair the advantages of the modal approach.
The spatial load truncation $\mathbf{p}_t^*$ is the amount of load not represented by the retained modes, and is given by the difference between the applied load $\mathbf{p}^*$ and the modally represented spatial load vector $\mathbf{p}_s^*$ for the modes retained:

$$\mathbf{p}_t^* = \mathbf{p}^* - \mathbf{p}_s^* \tag{A.9}$$

where * refers to the physical domain.
A technique which improves the accuracy of the modal dynamic analysis is the Mode Acceleration method (MA). MA takes into account the portion of the solution not represented without calculating the non-retained modes. Indeed, the displacements with the non-modally represented dynamics are corrected thorugh a static solution to $\mathbf{p}_t^*$ [2].
The equation of motion for a linear dynamic system:

$$\mathbf{M}^*\ddot{\mathbf{u}} + \mathbf{C}^*\dot{\mathbf{u}} + \mathbf{K}^*\mathbf{u} = \mathbf{p}^* \tag{A.10}$$

and its response is:

$$\mathbf{u} = \mathbf{K}^{*-1}\left(\mathbf{p}^* - \mathbf{M}^*\ddot{\mathbf{u}} - \mathbf{C}^*\dot{\mathbf{u}}\right) \tag{A.11}$$

A set of normal modes $\mathbf{U}$ relating the displacement $\mathbf{u}$ to the modal coordinates $\mathbf{q}$ are introduced:

$$\mathbf{u} = \mathbf{U}\mathbf{q} \tag{A.12}$$

with the following properties:

$$\mathbf{M} = \mathbf{U}^T\mathbf{M}^*\mathbf{U}; \quad \mathbf{C} = \mathbf{U}^T\mathbf{C}^*\mathbf{U}; \quad \mathbf{K} = \mathbf{U}^T\mathbf{K}^*\mathbf{U} = \mathbf{M}\boldsymbol{\Omega}^2; \tag{A.13}$$

where $\boldsymbol{\Omega}$ is the matrix of modal circular frequencies. Applying modal tranformation to Eq. (A.10) and supposing a mode normalization to unit generalized mass, i.e. $\mathbf{M} = \mathbf{I}$, th dynamic of the system is:

$$\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\Omega}^2\mathbf{q} = \mathbf{U}^T\mathbf{p}^* \tag{A.14}$$

Recovering modal acceleration:

$$\ddot{\mathbf{q}} = \mathbf{U}^T\mathbf{p}^* - \mathbf{C}\dot{\mathbf{q}} - \boldsymbol{\Omega}^2\mathbf{q} \tag{A.15}$$

and introducing modal transformation for accelerations and velocities, i.e. $\ddot{\mathbf{u}} = \mathbf{U}\ddot{\mathbf{q}}$, $\dot{\mathbf{u}} = \mathbf{U}\dot{\mathbf{q}}$, Eq. (A.11) gives:

$$\begin{aligned}
\mathbf{K}^*\mathbf{u} &= \mathbf{p}^* - \left(\mathbf{M}^*\mathbf{U}\ddot{\mathbf{q}}\right) - \left(\mathbf{C}^*\mathbf{U}\dot{\mathbf{q}}\right) && \text{(A.16)}\\
&= \mathbf{p}^* - \left(\mathbf{M}^*\mathbf{U}\left(\mathbf{U}^T\mathbf{p}^* - \mathbf{C}\dot{\mathbf{q}} - \boldsymbol{\Omega}^2\mathbf{q}\right)\right) - \left(\mathbf{C}^*\mathbf{U}\dot{\mathbf{q}}\right) && \text{(A.17)}\\
&= \left(\mathbf{p}^* - \mathbf{M}^*\mathbf{U}\mathbf{U}^T\mathbf{p}^*\right) + \mathbf{K}^*\mathbf{U}\mathbf{q} && \text{(A.18)}
\end{aligned}$$

Thus:

$$\mathbf{u} = \mathbf{K}^{*-1}\left(\mathbf{p}^* - \mathbf{M}^*\mathbf{U}\mathbf{U}^T\mathbf{p}^*\right) + \mathbf{U}\mathbf{q} = \mathbf{K}^{*-1}\mathbf{p}_t^* + \mathbf{U}\mathbf{q} = \mathbf{u}_c + \mathbf{U}\mathbf{q} \tag{A.19}$$

A correction $\mathbf{u}_c = \mathbf{K}^{*-1}\mathbf{p}_t^*$ to the modal displacement $\mathbf{U}\mathbf{q}$ is provided by considering the load truncation $\mathbf{p}_t^*$. It can be seen that damping does not appear in the previous expression whethere is considered or not. MD and MA will give the same results during a free decay. The correction term of MA is important when the applied loads are large and the system is dominated by them [1].
The term $\mathbf{M}^*\mathbf{U}\mathbf{U}^T\mathbf{p}^*$ is the physical force represented by the adopted modes along the physical domain. Its magnitude depends on the external loads applied and by the modes adopted and should be compared to $\mathbf{U}\mathbf{q}$ to understand how much the solution is affected by the correction. When all modes are used, the correction term in Eq. (A.19) vanishes:

$$\mathbf{u}_c = \left(\mathbf{K}^{*-1} - \mathbf{K}^{*-1}\mathbf{M}^*\mathbf{U}\mathbf{U}^T\right)\mathbf{p}^* = \left(\mathbf{K}^{*-1} - \mathbf{U}\boldsymbol{\Omega}^{-2}\mathbf{U}^T\right)\mathbf{p}^* = 0 \tag{A.20}$$

because $\mathbf{U}\mathbf{\Omega}^{-2}\mathbf{U}^T$ represents the flexibility matrix $\mathbf{K}^{*-1}$. Highliting the truncated modes $\mathbf{U}_t$ in the representation of the flexibility matrix, the correction term is:

$$\mathbf{u}_c = \left(\mathbf{U}_t\mathbf{\Omega}^{-2}\mathbf{U}_t^T\right)\mathbf{p}^* \tag{A.21}$$

This term is the difference between the exact static response and the static response of the structure based on the truncated modal representation.
Generally speaking, MA accounts for both the dynamic effects of the modes adopted and the static response of the high frequency modes. Compared to a Modal Truncation (MT) augmentation methods, only the displacements are corrected for the truncated modes. For the case of MT, dynamic effects from the non-retained modes are considered as well (on final accelerations and velocities). They are recovered from a second dynamic solution with load $\mathbf{p}_t^*$ which averages the response of the unmodelled modes. With MA, physical responses are recovered as summation of their transient and steady-state contribution. The transient portion of the response is recovered from the modal responses of the truncated dynamic model. The steady-state portion is calculated using static deflection analyses of the entire finite element model with inertial relief automatically included. By definition, the modes retained accurately span the frequency range of interest, while the loading represented by the non retained modes will produce a static response. In other words, the modes not retained are supposed to cause a minor change in velocity or acceleration which are accurately predicted by the low frequency retained modes.
Unfortunately, the static corrections improve the accuracy in the low frequency range [3] Also, the application of MA can often be prohibitive because it requires several static FE solutions to be carried out for each time step $\bar{t}$ or frequency value $\bar{\omega}$. Computational savings can be given by the Unit Load Method which reduces the number of systems to be solved (forward and backward substitution) to the DOFs loaded. This is carried out prior the recovery phase. Successively, a simple product of the unit load deflections with the applied loads provides the corrected displacement field. Also, selective data recovery can be applied, instead of having to solve for all the DOFs. The unit load methiod works well when the loaded DOFs are a few. When distributed loads are considered, such as pressure or gravity loads, the computational gain needs to be well estimated. This is the case of aeroelastic response to gust excitation.
The aeroelastic response of the full original problem in the displacement space $\mathbf{u}$ is:

$$\left(-\omega^2\mathbf{M}^* + i\omega\mathbf{C}^* + \mathbf{K}^* - q_\infty{\mathbf{H}^*}_{mm}(k, M)\right)\mathbf{u}(\omega) = \mathbf{p}^*(\omega) \tag{A.22}$$

where * indicates the matrices refer to the original FE mesh. The aerodynamic term is given by the matrix $\mathbf{H}^*_{mm}$ which provides the oscillatory forces along the FE model due to nodal dynamics. The aeroelastic analysis is never carried out in the physical domain because the size of the problem would be prohibitive.
Thus the reduced modal problem is always considered:

$$\left(-\omega^2\mathbf{M} + i\omega\mathbf{C} + \mathbf{K} - q_\infty\mathbf{H}_{mm}(k, M)\right)\mathbf{q}(\omega) = \mathbf{p}(\omega) \tag{A.23}$$

Similarly as above, the aerodynamic term is given by the matrix $\mathbf{H}_{mm}$ which provides the generalzied forces due to modal displacements.
For the case of gust inputs, the previous equation becomes:

$$\left(-\omega^2\mathbf{M} + i\omega\mathbf{C} + \mathbf{K} - q_\infty\mathbf{H}_{mm}(k, M)\right)\mathbf{q}(\omega) = q_\infty\mathbf{H}_{mg}(k, M)\mathbf{v}_g(\omega) \tag{A.24}$$

The application of the acceleration modes to Eq. (A.22) leads to:

$$\mathbf{K}^*\mathbf{u}(\omega) = -\left(-\omega^2\mathbf{M}^* + i\omega\mathbf{C}^* - q_\infty\mathbf{H}^*{}_{mm}(k, M)\right)\mathbf{u}(\omega) + \mathbf{p}^*(\omega) \tag{A.25}$$

Considering the modal solution from Eq. (A.23) and a case of gust input only, the state is:

$$\mathbf{K}^*\mathbf{u}(\omega) = \left(\omega^2\mathbf{M}^* - i\omega\mathbf{C}^* + q_\infty\mathbf{H}^*{}_{mm}(k, M)\right)\mathbf{U}\mathbf{q}(\omega) + q_\infty\mathbf{H}^*_{mg}(k, M)\mathbf{v}_g(\omega) \tag{A.26}$$

Two features must highlithed:

1. the equation represetns the output of the system $\mathbf{u}(\omega)$, function of its inputs and state vector $\mathbf{q}(\omega)$;
2. the system modelled is simple proper and may result in Gibbs oscillations when step inputs are given.

From the computational point of view, the major issues are related to the aerodynamic matrices $\mathbf{H}^*{}_{mm}$ and $\mathbf{H}^*{}_{mg}$ providing the forces along the FE mesh. The former is the one commonly used in aeroelastic analysis for flutter and modal dynamic response. On the other hand, the latter is uncommon; each coefficient ij represents the force on the degree of freedom i due to a unit displacement of the degree of freedom j. This is computational demanding and requires the aerodynamic mesh discretization to be of the same size of the underlying structural model. However, NeoRESP directly evalutes the matrix $\mathbf{\Theta} = \mathbf{H}^*{}_{mm}\mathbf{U}$ which gives the nodal aerodynamic forces due to modal shapes. Thus, the process of calculating $\mathbf{H}^*{}_{mm}$ is failry similar to the one giving $\mathbf{H}_{mm}$. The same can be said for gust input matrix $\mathbf{H}^*{}_{mg}$ which gives the nodal forces due to the panel downwash.

A.6.1 Acceleration modes in NeoRESP

Acceleration modes are required in NeoRESP by including `PARAM MODACC 0` generated by GUESS. Before running the aeroelastic response analysis, the matrices appearing in Eq. (A.24),(A.26) must be determined. Once all the data are available, the solver evaluates the response $\mathbf{q}(\omega)$ for each value of frequency (Eq. (A.24)) and anti-transforms into the time domain to recover $\mathbf{q}(t)$. Finally, acceleration modes are applied to correct the solution.
The creation of the aerodynamic data is time-consuming.In this respect, a pre-processor must be run to create a database of the aeroelastic model to be used for multiple runs. By typing `init_dyn_model(filename)`, where `filename` is the main file of the numeric model created by GUESS, the pre-processor is run, .
The following steps are taken:

- the Doublet Lattice kernel is initialized;
- mass $\mathbf{M}*$, damping $\mathbf{C}^*$ and stiffness $\mathbf{K}*$ matrices are determined;
- eigenvalue analysis is carried out, providing $\mathbf{M}$, $\mathbf{C}$, $\mathbf{K}$, $\mathbf{U}$ and $\mathbf{\Omega}$
- modes $\mathbf{U}$ are appended with control surface dummy modes $\mathbf{U}_c$ and rigid modes $\mathbf{U}_r$ are corrected according to `SUPORT` entry;

- modal shapes are interpolated onto the aerodynamic mesh and boundary conditions for the DLM are applied;
- aerodynamic matrices $\mathbf{H}_{mm}$ and $\mathbf{\Theta}$ are evaluated;
- flutter analysis is carried out and rigid modes (phugoid, short-period, dutch-roll, roll, spiral) damping and frequency is predicted;
- if gust input is required, gust matrices $\mathbf{H}_{mg}$ and $\mathbf{H}_{mg}$ are created;
- matrices to recover user-define outputs $\bar{\mathbf{y}}$, e.g. nodal accelerations, velocities, displacements, forces along panels, aerodynamic force and moment coefficients, control surface hinge moments, are saved.

Further information is given in the section dedicated to NeoRESP.

A.7 Modal damping

Damping is introduced in numerical models to take into account the energy dissipation during motion. Its modelling is troublesome, thus its characterization is usually supported by vibration tests. Different models are available: they model the dissipation as a function of velocity or displacement and can eventually be combined together.
The amount of damping can be expressed as a percentage of the critical value. Considering a simple damped harmonic oscillator with mass m, damping coefficient c, and spring constant k, the damping factor η is defined as the ratio between the system damping c and the critical value c_c:

$$\eta = \frac{c}{c_c} = \frac{c}{2\sqrt{m\,k}} = \frac{c}{2m\omega_o} \tag{A.27}$$

where ω_0 is the natual frequency of the system, $\omega_0 = \sqrt{k/m}$. Common aerospace structures are weakly damped, i.e. η is small, of the order of cents or tenths. The modes of the system are hence not so much affected by damping and are very close to those of the undamped system. Resonance tests show that modes are mostly real and the damping is predominantly diagonal. Tipically, structural damping is due to friction along joints. The dissipation is due to coulombian force which is a non-linear function of velocity:

$$F_s = -f(v)\,N\,\frac{v}{|v|} \tag{A.28}$$

where N is normal load, f is the friction coefficient and v is the local relative velocity.
Of course this relation is far different from the viscous damping model $\mathbf{C}\dot{\mathbf{u}}$ which is proportional to the velocity and is usually over-abused by virtue of its linearity. However, the model is keen on being tuned so that overal dissipation can be made equivalent to the real average value. For the cases investigated in NeoCASS, the numerical damping modelling relys on:

- the calculation of undamped modes (real analysis), application of modal transformations resulting in uncoupled dynamic equations;
- adoption of a proportional damping model by imposing a-priori a diagonal modal damping matrix. The equations will be still uncoupled.

NeoCASS allows to adopt two simple modal approximations for proportional damping: viscous or structural damping. The damping can is used for:

1. flutter assessment in SMARTCAD;
2. frequency domain dynamic response in NeoRESP.

The modal viscous damping coefficient matrix is:

$$\mathbf{B} = \mathbf{U}^T \mathbf{C}^* \mathbf{U} = 2\mathbf{\Xi}\mathbf{M}\mathbf{\Omega} \tag{A.29}$$

where $\mathbf{\Xi}$ is the diagonal matrix of viscous damping ratios η_k. The real eigenvectors of the undamped system still diagonalize the problem. The damping matrix has as many damping coefficients as the number of modes used, which makes the tuning of the model possible and straightforward.
In the frequency domain, the structural model is:

$$\left(-\omega^2 \mathbf{M} + i\omega\, 2\mathbf{\Xi}\mathbf{M}\mathbf{\Omega} + \mathbf{M}\mathbf{\Omega}^2\right) \mathbf{q}(\omega) = \mathbf{p}(\omega) \tag{A.30}$$

The damping force increases with ω.
Rayleigh damping considers only two parameters α, β and models damping as a combination of mass and stifness matrices:

$$\mathbf{C}^* = \alpha \mathbf{M}^* + \beta \mathbf{K}^* \tag{A.31}$$

The model allows to have a correct tuning of two modes only, with the risk of leading to unrealistic damping levels for the remaining modes. Thus it is necessary to find a good compromise for these parameters to have a fair agreement with experiments. In this case the modal damping matrix is expressed as:

$$\mathbf{C} = \alpha \mathbf{U}^T \mathbf{M}^* \mathbf{U} + \beta \mathbf{U}^T \mathbf{K}^* \mathbf{U} \tag{A.32}$$

and the damping factor matrix is:

$$\mathbf{\Xi} = \frac{1}{2}\alpha \mathbf{M}\mathbf{\Omega}^{-1} + \frac{1}{2}\beta \mathbf{M}\mathbf{\Omega} \tag{A.33}$$

Note that Rayleigh damping is a special case of viscous damping.
The structural (or hysteretic) damping simulate the energy loss is proportional to displacement. Energy dissipation is modelled by means of a complex stiffness which introduce a phase lag for displacements. When one oscillation cycle is considered, there is net loss of energy. The model has no property derived from a physical law and can be applied only in the frequency domain because it implies non casual behavior in time.
The structural model is:

$$\left(-\omega^2 \mathbf{M} + (1 + i\omega\, g)\, \mathbf{M}\mathbf{\Omega}^2\right) \mathbf{q}(\omega) = \mathbf{p}(\omega) \tag{A.34}$$

The modes are still real in analogy with the previous methods. In this case only one parameter, g which scales the stiffness matrix, is used.
Within NeoCASS, the modes are always recovered from the undamped case. Successively the damping matric will be added to the system according to one of the models presented.

1	2	3	4	5	6	7	8	9	10
TABDMP1	TID	TYPE							
	f1	g1	f2	g2	f3	g3	f4	g4	
	f5	g5	ENDT						

TID	Table identification number ($0 < \text{Integer} < 100.000.000$)
TYPE	G, CRIT, Q to provide respectively g_k, η_k, Q_k
f_k	frequency value
g_k	damping value according to TYPE
ENDT	specifies where the table input ends

Table A.7: TABDMP1 damping definition

Damping is provided in NeoCASS through the card TABDMP1, which is a list of η_k, or $g_k = 2\eta_k$, or dynamic magnification factor $Q_k = \frac{1}{g_k}$ for different values of frequency (Hz). An interpolation process is carried out to find the value of g for a required frequency f. Damping is included in the model if the command line `SDAMPING= TID` is given, where `TID` is the identification number of a TABDMP1 card. If no `SDAMPING` is provided, no TABDMP1 will be considered and no damping will be applied. The present method is flexible. It allows to consider viscous, Rayleigh or structural damping with only one card. For the case of viscous damping (Eq. (A.29)) this is straigtforward. The user directly provides the damping coefficients at different frequency values. If natural frequencies Ω are known, Eq. (A.33) can be provided as well trough a proper data input. Supposing unit generalized mass, the table will list:

$$\eta_k = \frac{1}{2}\alpha\Omega_k^{-1} + \frac{1}{2}\beta\Omega_k \tag{A.35}$$

Finally, to model structural damping the user can specifiy a costant value of damping factor along a range of frequencies, i.e. the value of g in Eq. (A.34). The code will then apply the same value of g to each mode. By including `PARAM KDAMP 0`, the damping matrix will be included in the complex stiffness matrix. The default value for KDAMP is 1, a separate damping matrix will be assembled and viscous damping applied.

A.8 Output

Outputs are available in the struct `dyn_model.Res`. Default outputs are:

- Q, Qd, Qddot for $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$;
- Time, for simulation time t;
- Qload, sum of gust and control input generalized forces;
- Qextf, external load generalized forces;
- Extload_profile, representation of TFUN of DLOAD card;
- Gust_profile, representation of TFUN of GUST card;

- Control_profile, representation of TFUN of SURFDEF card;
- Cy_mode, Cz_mode, Cl_mode, Cm_mode, Cn_mode aerodynamic force and moment coefficients due to **q**;
- Cy_gust, Cz_gust, Cl_gust, Cm_gust, Cn_gust aerodynamic force and moment coefficients due to gust;

Some outputs can be requested through dedicated cards:

1. DISP: nodal displacmeents;
2. VELOCITY: nodal accelerations;
3. ACCELERATION: nodal accelerations;
4. IFORCE: bar internal forces;
5. CP_mode, CP_gust panel pressure coefficients due to **q** and gust respectively;
6. HINGEFORCE: control surface hinge moment;
7. MODACC: with corrected bar internal forces and nodal displacements;

The MODACC field is created when acceleration modes are used through `PARAM MODACC 0`. The other outputs require four dedicated cards respectively:

1. DISP= ID
2. VELOCITY= ID
3. ACCELERATION= ID
4. IFORCE= ID
5. AEROFORCE= ID
6. HINGEFORCE= ID

where ID is the indentification number of a SET card proving a list of identifiers for nodes and bars. The ID for control surface to be used in HINGEFORCE is the position of the control in `beam_model.Aero.Trim.Link.Master`. Alternatively, ALL can be used in place of ID, meaning all the nodes or bars will be considered. The SET card is quite similar to MSELECT card introduced in Sec. A.5.1: For example:

SET	ID	N1	N2	N3	N4	N5	N6	N7	
	N8	etc							
ID	Set identifier								
Ni	Node or Bar ID (THRU and EXCEPT can be used)								

Table A.8: SET definition of nodes and bars.

```
SET 1 =    2002   4000
```

```
SET 2 =    2000
DISP= ALL
ACCELERATION= 2
IFORCE= 1
AEROFORCE= ALL
```

Will export the displacements for all nodes and the accelerations only for nodes 2000. Internal forces will be exported only for bar elements 2002 and 4000. Finally, pressure coefficients for the whole aerodynamic mesh will be exported.

A.9 NeoRESP preprocessor

The preprocessor loads the file for NeoRESP and processes all the input provided and determined all the data required to solve:

$$\left(-\omega^2\mathbf{M} + i\omega\mathbf{C} + \mathbf{K} - q_\infty\mathbf{H}_{mm}(k,M)\right)\mathbf{q}(\omega) = q_\infty\mathbf{H}_{mg}(k,M)\mathbf{v}_g(\omega) + \mathbf{p}(\omega) \tag{A.36}$$

SMARTCAD routines are called at this stage:

- the Doublet Lattice kernel is initialized;
- mass $\mathbf{M}*$, damping $\mathbf{C}^*$ and stiffness $\mathbf{K}*$ matrices are determined;
- eigenvalue analysis is carried out, providing $\mathbf{M}$, $\mathbf{C}$, $\mathbf{K}$, $\mathbf{U}$ and $\mathbf{\Omega}$
- modes $\mathbf{U}$ are appended with control surface dummy modes $\mathbf{U}_c$ and rigid modes $\mathbf{U}_r$ are corrected according to `SUPORT` entry;
- modal shapes are interpolated onto the aerodynamic mesh and boundary conditions for the DLM are applied;
- aerodynamic matrices $\mathbf{H}_{mm}$ and $\mathbf{\Theta}$ are evaluated;
- flutter analysis is carried out and rigid modes (phugoid, short-period, dutch-roll, roll, spiral) damping and frequency is predicted;
- if gust input is required, gust matrices $\mathbf{H}_{mg}$ and $\mathbf{H}_{mg}$ are created;
- matrices to recover user-define outputs $\bar{\mathbf{y}}$ presented in Sec. A.8, e.g. nodal accelerations, velocities, displacements, forces along panels, aerodynamic force and moment coefficients, control surface hinge moments, are saved.

The input file for NeoRESP is fairly similar to the one for flutter analysis. The following cards must be included to the aero-structural model generated by GUESS:

- SOL 146: set NeoRESP as solver;
- PARAM VREF, RHOREF, MACH: defition of reference flight condition;
- EIGR: eigenvalues analysis settings;

- AERO: reference values for aerodynamics, DLM kernel settings and flutter anlysis inputs;
- MKAERO1: list of reduced frequnecies k and Mach numbers M for $\mathbf{H}_{mg}$ and $\mathbf{H}_{mg}$;
- SUPORT: require rigid body modes along coordinate axes;
- GUST, SURFDEF, DLOAD, GUST=, SURFDEF=, DLOAD=: set forcing terms.

Cards not mandatory are:

- PARAM MODACC: mode acceleration;
- TABDMP1, SDAMPING=, PARAM KDAMP 0: for modal damping;
- DISP=, ACCCELERATION=, VELOCITY=, IFORCE=, AEROFORCE=, HINGE-FORCE=: define structural and aerodynamic output;
- MSELECT, UMODES, FMODES: set modal base, modes to be used in dynamic response and to be assessed in flutter analysis.

An example with few comments is provided below:

```
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ Dynamic response solver number
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
SOL 146
PARAM   MODACC  0
PARAM   VREF    238.0
PARAM   RHOREF  1.225
PARAM   MACH    0.7
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ Output set (node list)
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
SET 1 =    2002   4000
SET 2 =    2000
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ Choose output (displacements, velocities, accelerations and internal forces)
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
DISP= ALL
ACCELERATION= 2
IFORCE= 1
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ Enable Cp export
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
AEROFORCE= ALL
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ Select gust
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
GUST= 1
```

```
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ Gust input
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
GUST    1       17.07   0.12    0.0     3
        1
        (1-cos(2*pi/0.1200*t))*0.5
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ Flutter solver parameters
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
AERO    126.8   300     3.83    1.225   0       0       2       50      36.2
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ Reduced frequency and Mach number list
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
MKAERO1 0.7     0.9
        0.001   0.01    0.05    0.1     0.2     0.5     1.0     2.0
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ Eigen-analysis cards
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
EIGR    1               0       999999          30
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ SUPORT node
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
SUPORT  1       2000    123456
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ Modal base to be used in dynamic analysis (symmetric modes)
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
UMODES  3       5       7       THRU    15      17
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ Select modes from an external database
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
MSELECT 1       THRU    28
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ Modes to follow in flutter solution
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
FMODES  7       THRU    28
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
$ Include aero-structural model
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
INCLUDE GUESS_MODEL.dat
```

Calling for the preprocessor is mandatory in order to create all the data necessary for the solution of the dynamic equations. The preprocess is called by typing `init_dyn_model(filename)` where `filename` is the main input file for NeoRESP. The preprocessor will create a database in the global variable `dyn_model` which can be saved and used for successive runs with some data manipulation.
All the matrices reported in Eq. (A.36) are stored. This process is rather time-consuming because of the aerodynamic terms which are determined for each value of k and M. A

flutter assessment is run according to the input in AERO card. Usually a sea level density is prescribed so that the flutter point can be alligned with altitude (flutter envelope). This provides the user with the warning the system may be unstable.

Bibliography

[1] Y. T. Chung, *Dynamicloads recovery using alternative mode acceleration approach*, Proceedings of the 39$^{\text{th}}$ SDM (Long Beach,CA), no. AIAA-98-1719, April 1998.

[2] Blelloch P., *Calculation of structural dynamic forces and stresses using mode acceleration*, Journal of Guidance, Control, and Dynamics **12** (1989), no. 5, 760–762.

[3] D. J. Rixen, *Generalized mode acceleration methods and modal truncation augmentation*, Proceedings of the 42$^{\text{nd}}$ SDM (Seattle,WA), no. AIAA-2001-1300, April 2001.