Outline

- Direct frequency response = slow
- Reduction principles
- Reduction illustrations

- CMS
- CMS illustrations

- Course notes : chapter 5 : model reduction methods

SDTools
VIBRATION SOFTWARE & CONSULTING

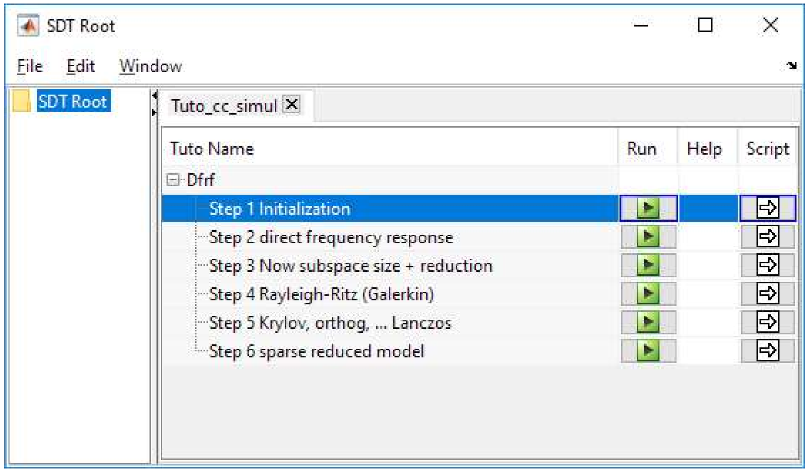
Arts et Métiers Institute of Technology

MS2SC http://savoir.ensam.eu/moodle/course/view.php?id=1874
PROVIR http://savoir.ensam.eu/moodle/course/view.php?id=9318

MATLAB Tutorial : direct frequency response issues

See cc_simul tuto

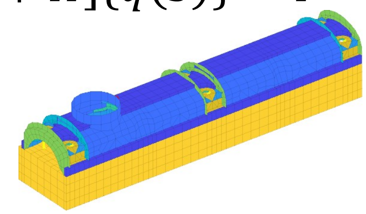
- Step1 : assembly, sparse matrices
- Step 2 : point load, collocated displacement, factorization strategies
- Step 3 : subspace around resonance, phase collinearity, SVD
- Step 4 : Rayleigh-Ritz, reduced FRF



Direct frequency response : Zq=F

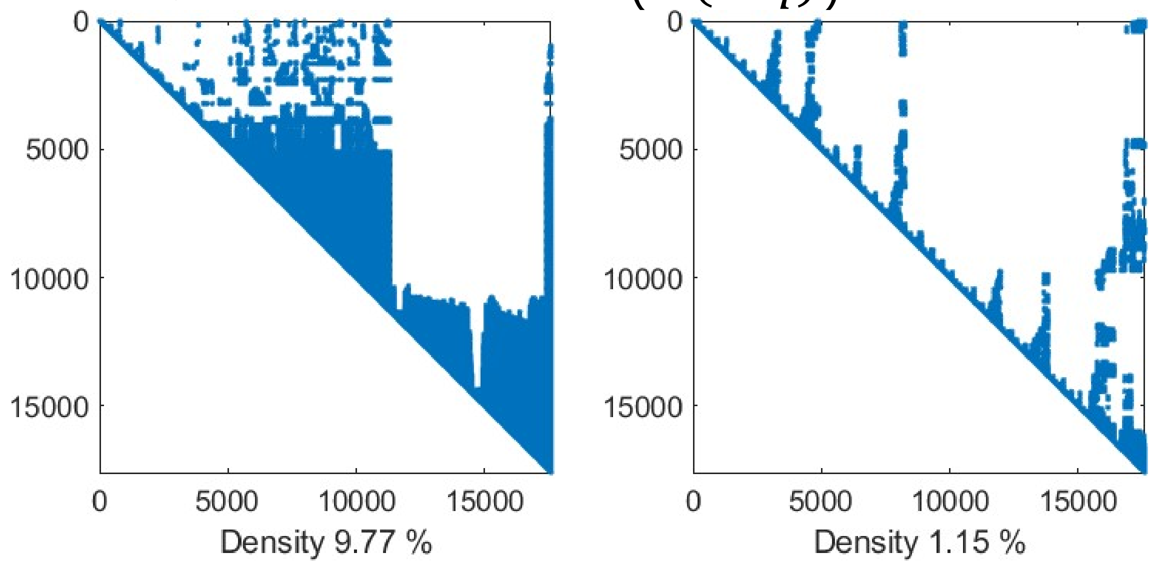
$$[Z]\{q\} = [Ms^2 + Cs + K]\{q(s)\} = F$$

1. Renumbering (fill in reduction, symbolic factorization METIS, symrcm, …)
2. Numerical factorization $Z = LU$ or $Z = LDL^T$
3. Forward/backward solve $L\left(D\left(L^T q\right)\right) = F$



17589 DOF

Fact+Solve	0.7s
FactMA57	0.8s
Solve	0.02s
FactPardiso	0.23s
Solve	0.01s



Sparse libraries : Umfpack (lu), MA57 (ldl), Pardiso, Mumps, BCS-Lib, Spooles, Taucs, …

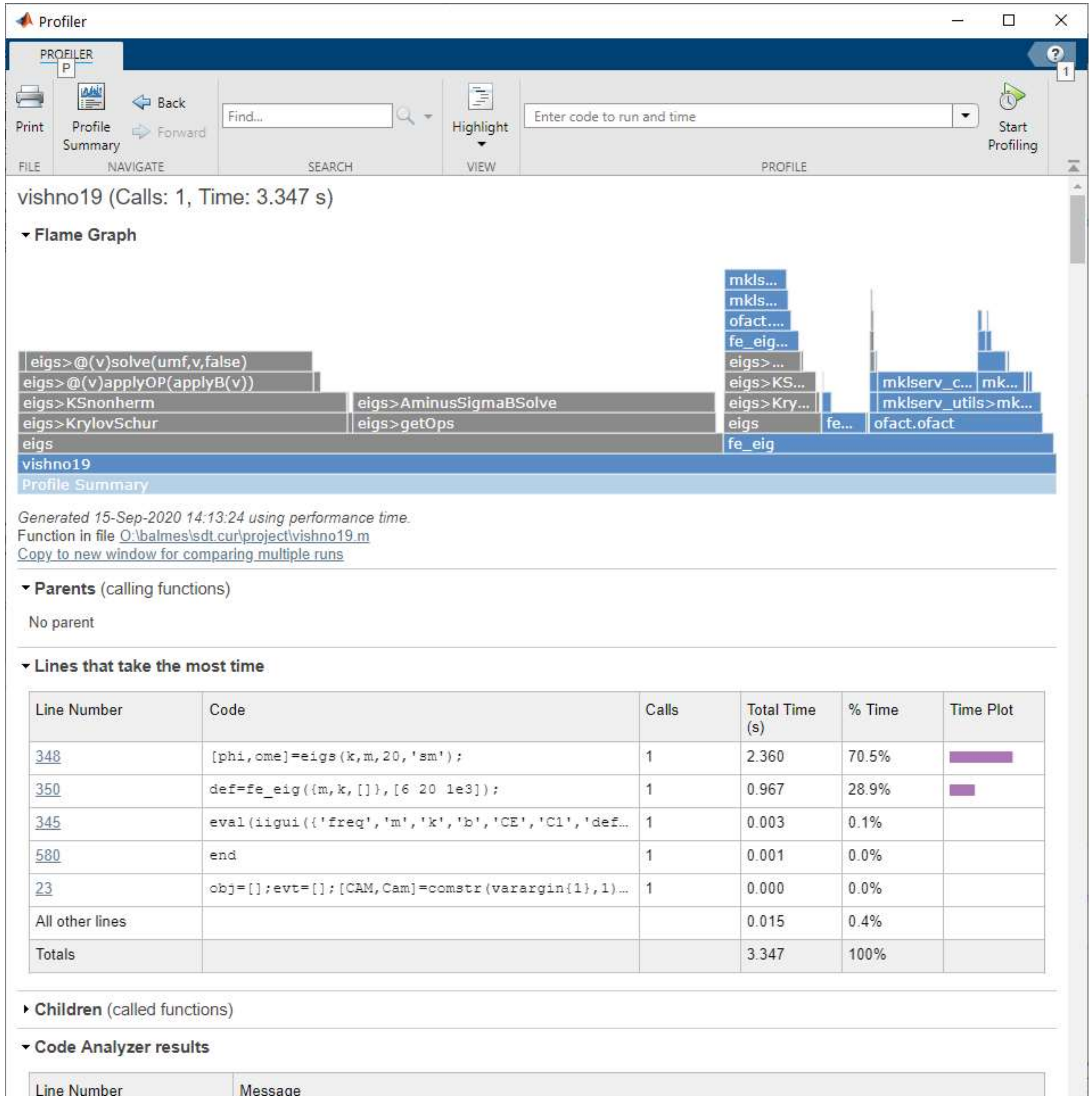
Eigenvalue computation

Sparse library choice

x2.4 speedup

Main steps :

- Factor
- Iterate



Modal frequency response : H

1. Renumbering, factorization of $Z(\omega_0)$ — 1/2 factor (60%)

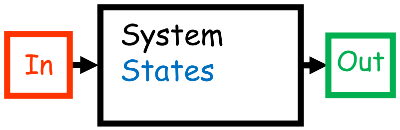
2. Partial eigenvalue solver (Lanczos, eigs Arnoldi, …) — 2 NM Solves (39%)
3. Reduction : $M_R = I, \ldots$ — NM^2 matrix/vector

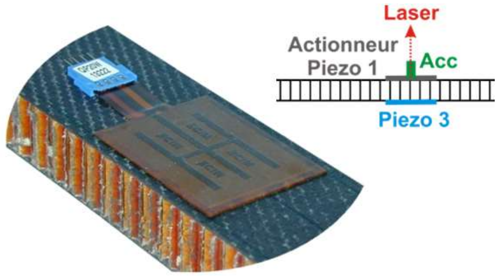
4. Modal coordinate solve — diagonal or NM^2 matrix
   much faster if NM<<Nw (1%)

$$[Ms^2 + Cs + K]\,\{q(s)\}_{Nq} = [b]\{u(s)\}$$
$$\{y(s)\} = [c]\{q(s)\}$$

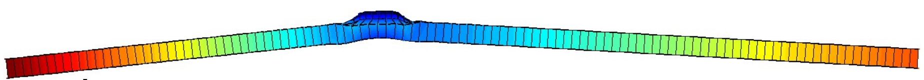
$$\left[Is^2 + \left[\ddots 2\zeta_j\omega_j \ddots\right] s + \left[\ddots \omega_j^2 \ddots\right]\right]\{q_R(s)\}_{Nqr} = [\phi_j^T b]\{u(s)\}$$
$$\{y(s)\} = [c\phi_j]\{q_R(s)\}$$

Transfers : what subspace is needed ?

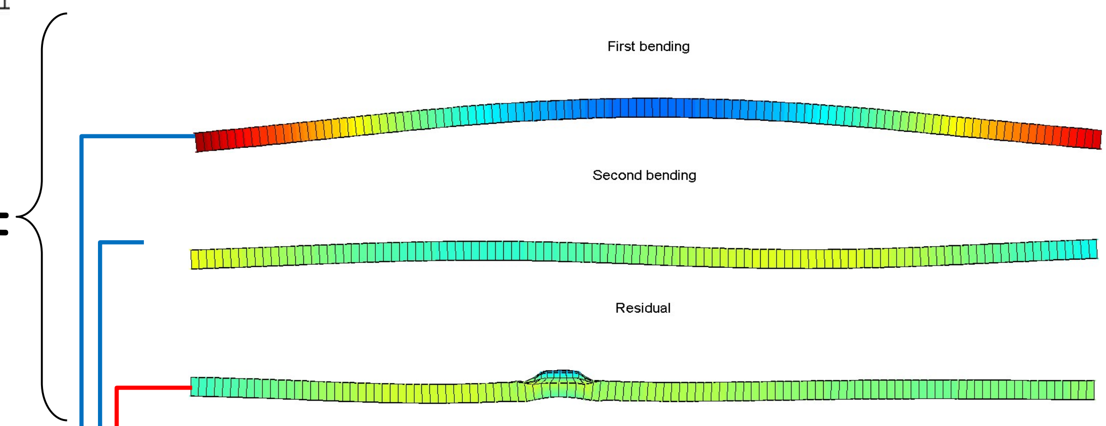


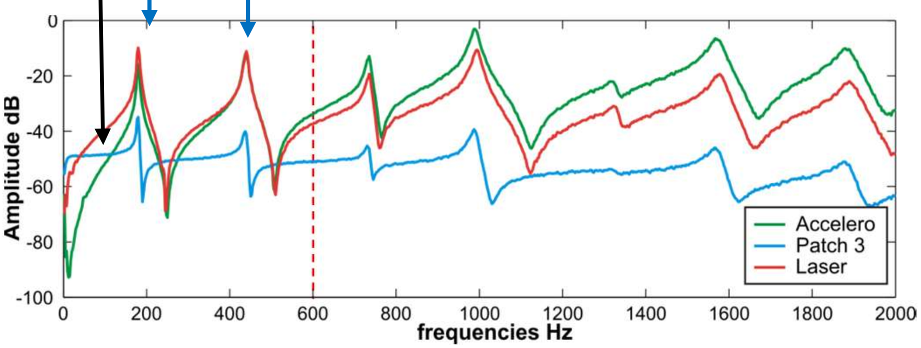


Quasi-static response @ 10Hz



=





$$H(\omega) = [c]\left[-M\omega^2 + K\right]^{-1}[b] \approx$$

$$\sum_{j=1}^{N_M} \frac{[c]\{\phi_j\}\{\phi_j\}^T[b]}{-\omega^2+\omega_j^2} + \sum_{j=N_M+1}^{N} \frac{[c]\{\phi_j\}\{\phi_j\}^T[b]}{\omega_j^2}$$

$[\phi_1 \;\ldots\; \phi_{NM} \qquad K^{-1}b]$

Modes + static correction

Reading the Abaqus documentation

Several analysis types in ABAQUS/Standard are based on the eigenmodes and eigenvalues of the system. For example, in a mode-based steady-state dynamic … (for more information, see "Linear dynamic analysis using modal superposition," Section 2.5.3 of the ABAQUS Theory Manual).

Due to cost, usually only a small subset of the total possible eigenmodes of the system are extracted, … it is usually the higher frequency modes that are left out. …

… superposition can be augmented with additional modes known as residual modes. The residual modes help correct for errors introduced by mode truncation. In ABAQUS/Standard a residual mode, R, represents the static response of the structure subjected to a **nominal (or unit) load, P**, corresponding to the actual load that will be used in the mode-based analysis orthogonalized against the extracted eigenmodes,

$$R^N = (\delta^{NJ} - \phi^N_\alpha \frac{1}{m_\alpha} \phi^I_\alpha M^{IJ})(K^{-1})^{JK} P^K,$$

followed by an orthogonalization of the residual modes against each other.

If the static responses are linearly dependent on each other or on the extracted eigenmodes, ABAQUS automatically eliminates the redundant responses for the purpose of computing the residual modes.

For the Lanczos eigensolver you must ensure that the static perturbation response of the load that will be applied in the subsequent mode-based analysis (i.e.,) is available by specifying that load in a static **perturbation** step. If multiple load cases are specified in this static perturbation analysis, one residual mode is calculated for each load case.

Reduction <-> Ritz analysis

Response is approximated

$$q(s) = \left[\phi_1 \ldots \phi_{NM} \quad \left[K_{Flex}\right]^{-1}[b]\right]_{N\times(NM+NA)} \left\{ \begin{array}{c} \vdots \\ \frac{\phi_j^T bu}{s^2+\omega_j^2} \\ \vdots \\ u \end{array} \right\}$$

- within subspace containing modes and flexibility

$$T = \left[\phi_1 \ldots \phi_{NR} \quad \left[K_{Flex}\right]^{-1}[b]\right]$$

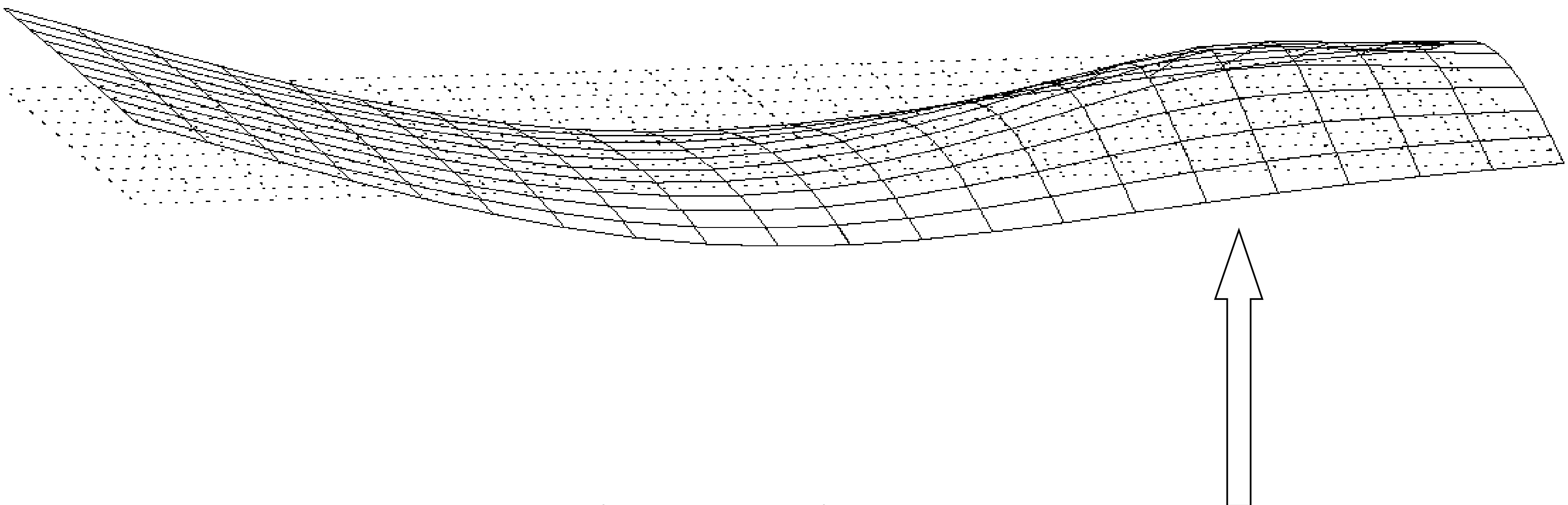
- or modes and residual flexibility

$$[T] = \left[[\phi_1 \ldots \phi_{NM}] \quad \left[[K]^{-1}_{Flex}[b] - \sum_{j=1}^{NM} \frac{\{\phi_j\}\{\phi_j\}^T [b]}{\omega_j^2} \right] \right]$$

- Prefiltering b may be necessary for numerical precision

$$T = \left[\phi_{1:NM} \quad \left[K_{Flex}\right]^{-1}\left[b - \left[M\left[\phi_{1:NM}\right]\right]\left[\left[\phi_{1:NM}\right]^T b\right]\right]\right]$$

Attachment modes



For free structure : static load implies deformation in a uniformly accelerating frame

$$\{q_F\} = [K]^{-1}_{Flex}\,[b] = \sum_{j=NB+1}^{N} \frac{\{\phi_j\}\left\{\phi_j^T b\right\}}{\omega_j^2}$$

See section 5.3.2 static response in presence of rigid body modes

Collocated transfers

- Collocated $\Leftrightarrow \{u\}^T\{\dot{y}\} = power \Leftrightarrow [c] = [b]^T$
- Modal contributions positive real

$$H_c(s) = \sum_{j=1}^{NM} \frac{(c\phi_j)^2}{s^2 + 2\zeta_j\omega_j s + \omega_j^2}$$

- Trivial ranking of mode contributions as fractions

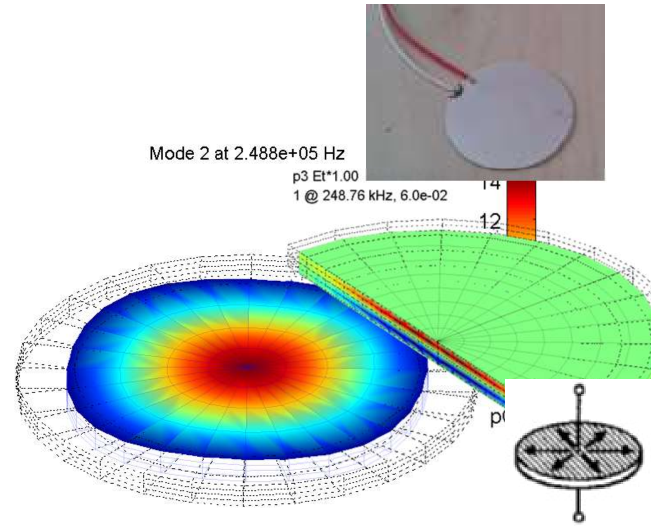
$$cont_j = \frac{(c\phi_j)^2/\omega_j^2}{\sum (c\phi_j)^2/\omega_j^2} \in [0\ \ 1]$$

Residual terms critical if number of sum of kept contributions not close to 1

Need for static correction : critical case

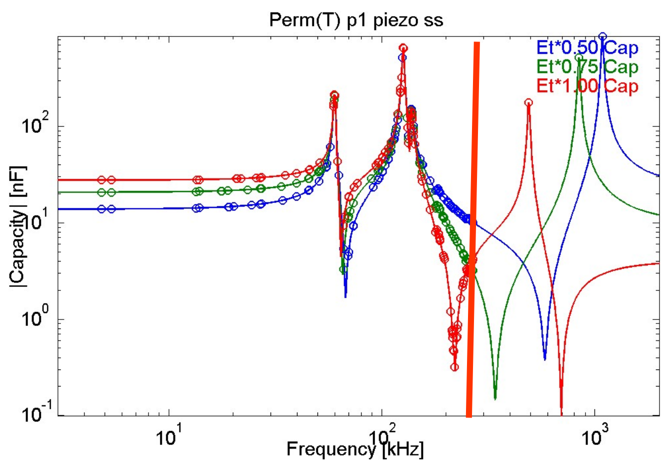
Traditional : modes + static correction

$$T = \begin{bmatrix} \phi(Zcc(\omega_j)) & K_{CC}(s)^{-1}K_{CV}(s)V_{In} \\ 0 & V_{In} \end{bmatrix}_{\perp M,K}$$

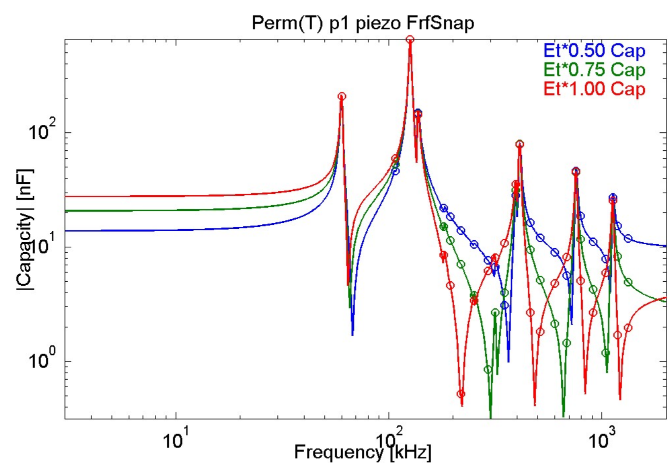


Snap-shot Ritz basis

$$T = \left[\left\{ \begin{matrix} Z_{CC}(s)^{-1}Z_{CV}(s)V_{In} \\ V_{In} \end{matrix} \right\}_{s \in i\omega_{target}} \right]_{\perp M,K}$$



3 out of 100 useful modes
Relatively close static correction



Easily captures wide range

Unit imposed displacement

Applied load : free modes + static correction = McNeal

Applied displacement : dynamic & Static/Guyan condensation

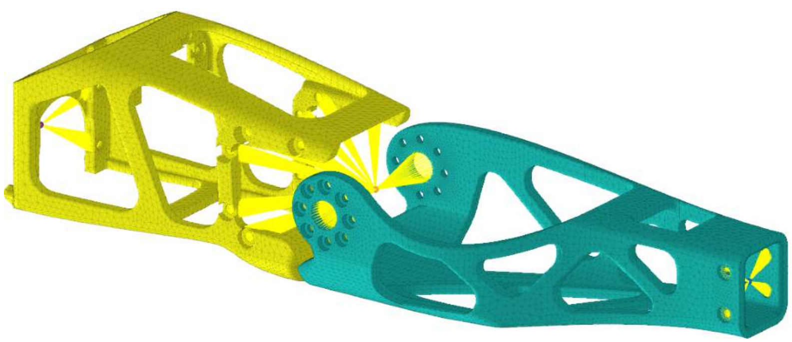
$$\begin{bmatrix} K_{II} & K_{IC} \\ K_{CI} & K_{CC} \end{bmatrix} \left\{ \begin{array}{c} < q_I(s) > \\ q_C(s) \end{array} \right\} + \left[M s^2 \right] \{q\} = \left\{ \begin{array}{c} R_I(s) \\ < 0 > \end{array} \right\}$$

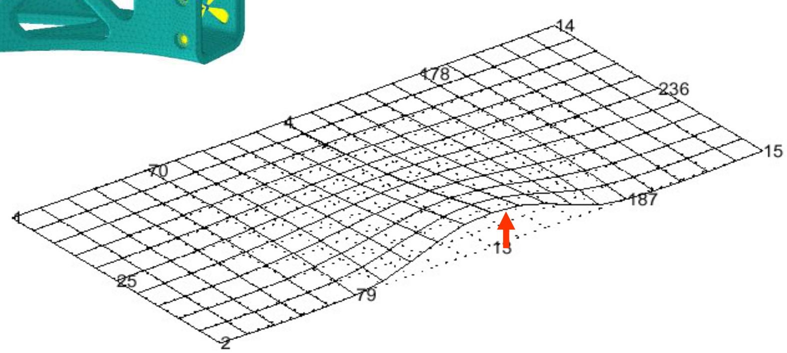
No interior load = dynamic condensation

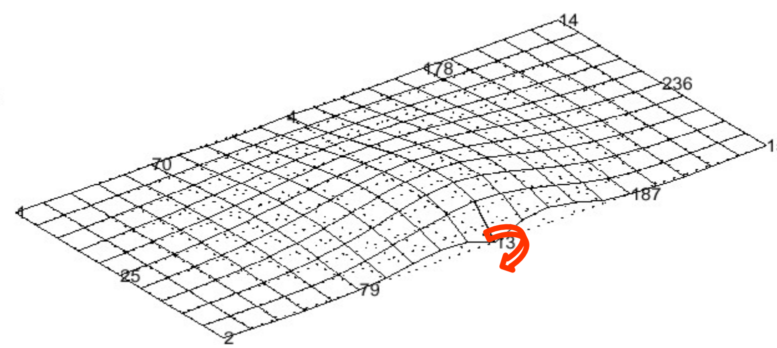
$$[T(\omega)] \{q_I\} = \begin{bmatrix} I \\ -Z_{CC}(\omega)^{-1} Z_{CI}(\omega) \end{bmatrix} \{q_I\}$$

Inertia cc neglected = static/Guyan

$$\{q\} \approx [T] \{q_I\} = \begin{bmatrix} I \\ -K_{CC}^{-1} K_{CI} \end{bmatrix} \{q_I\}$$







Frequency limit -> Craig-Bampton

Inertia neglected : error associated with $M_{cc}q_c$

$$\begin{bmatrix} K_{II} & K_{IC} \\ K_{CI} & K_{CC} \end{bmatrix} \left\{ \begin{array}{c} < q_I(s) > \\ q_C(s) \end{array} \right\} + \left[M s^2 \right] \{q\} = \left\{ \begin{array}{c} R_I(s) \\ < 0 > \end{array} \right\}$$

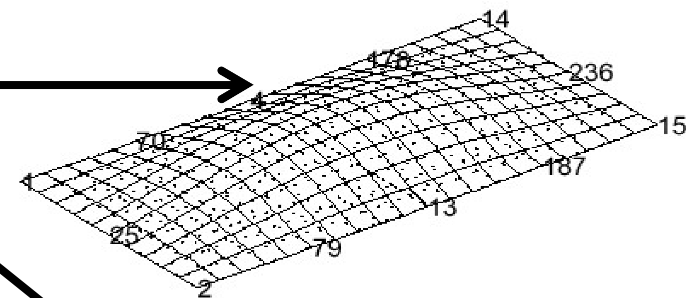
When $Z_{cc}(s)$ is singular

$\Rightarrow$

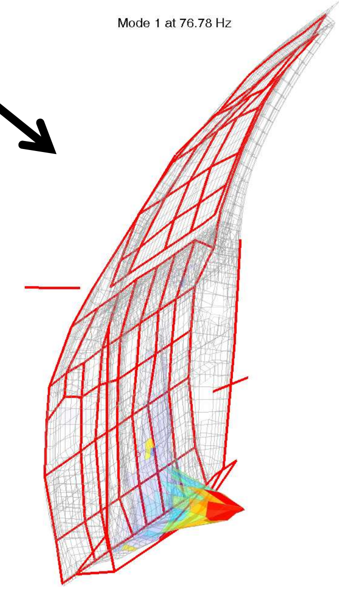
Approximation cannot be valid

$$\begin{bmatrix} 0 & 0 \\ 0 & Z_{CC}(\omega_j) \end{bmatrix} \left\{ \begin{array}{c} 0 \\ \phi_{j,c} \end{array} \right\} = \left\{ \begin{array}{c} R_I \\ 0 \end{array} \right\}$$

Fixed interface modes



Craig-Bampton = guyan/static + fixed interface



$$[T] = \left[\begin{bmatrix} I \\ K_{cc}^{-1} K_{ci} \end{bmatrix} \begin{bmatrix} 0 \\ \phi_{1:NM,c} \end{bmatrix} \right]$$

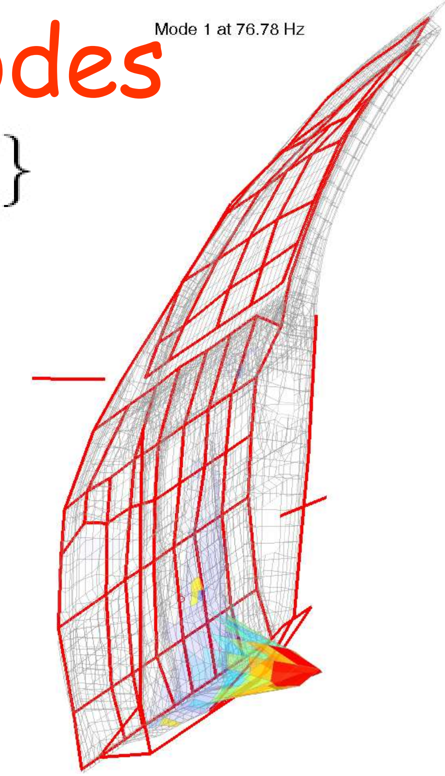
Application : fixed sensor mode

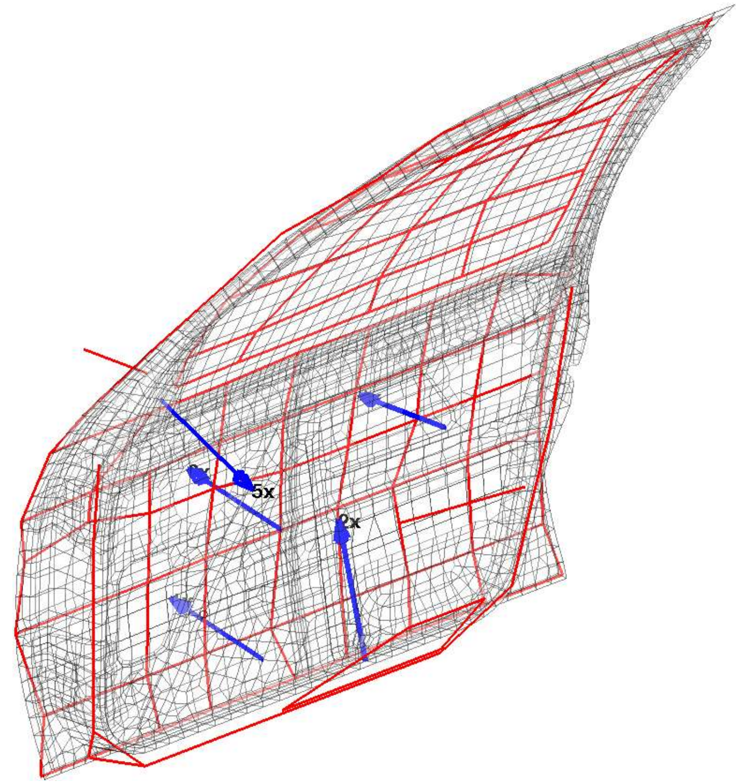
Fixed sensor modes

$$\left[K-\omega_j^2 M\right]\{\phi_j\}=\{0\}$$

With

$$[c_m]_{m\le k}\{\phi_j\}=0$$

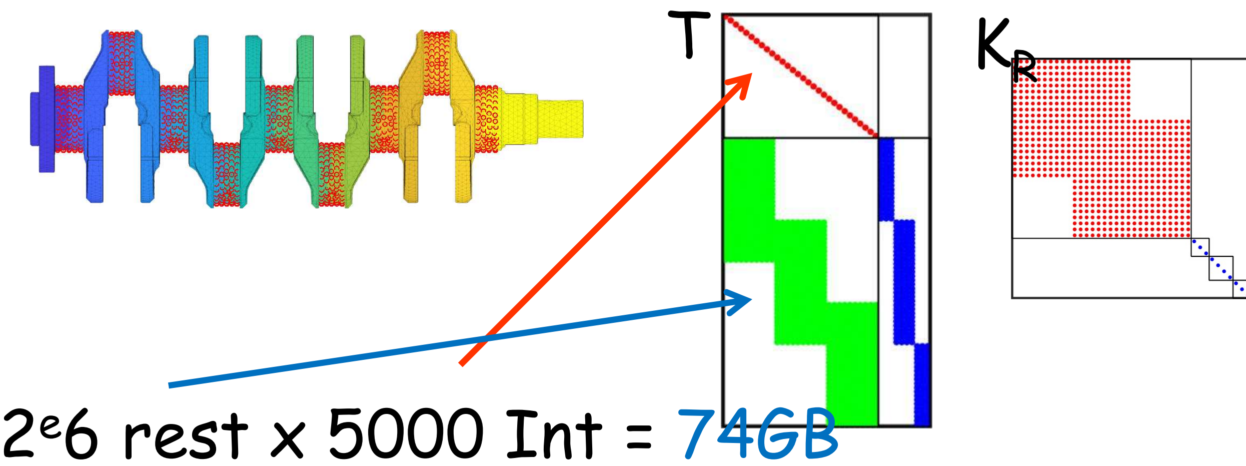




Use : place additional sensors to extend frequency band (IMAC 05)

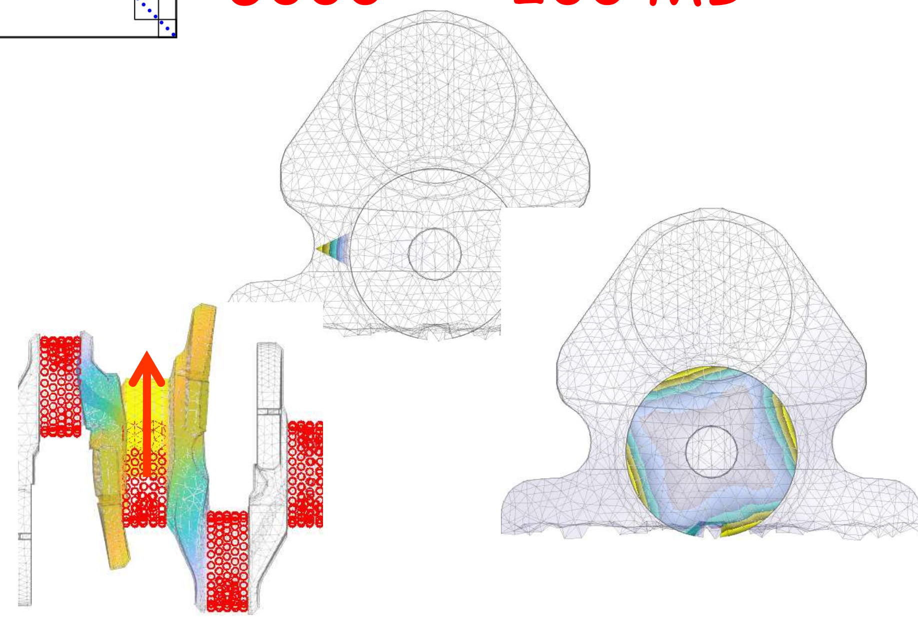
Interface reduction / model size / sparsity

- Craig-Bampton often sub-performant because of interfaces



5000^2 = 200 MB

- Unit motion can be redefined : interface modes
  Fourier, analytic polynomials, local eigenvalue
  5000 -> 500 interface DOFs.
- Disjoint internal DOF subsets

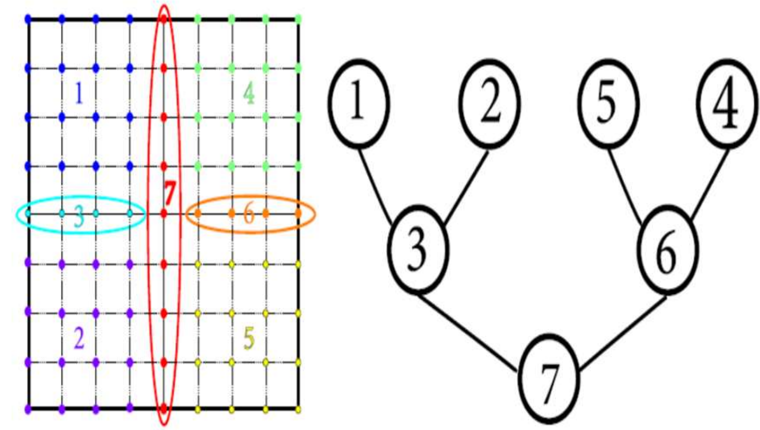


Separate requirements for learning shapes :

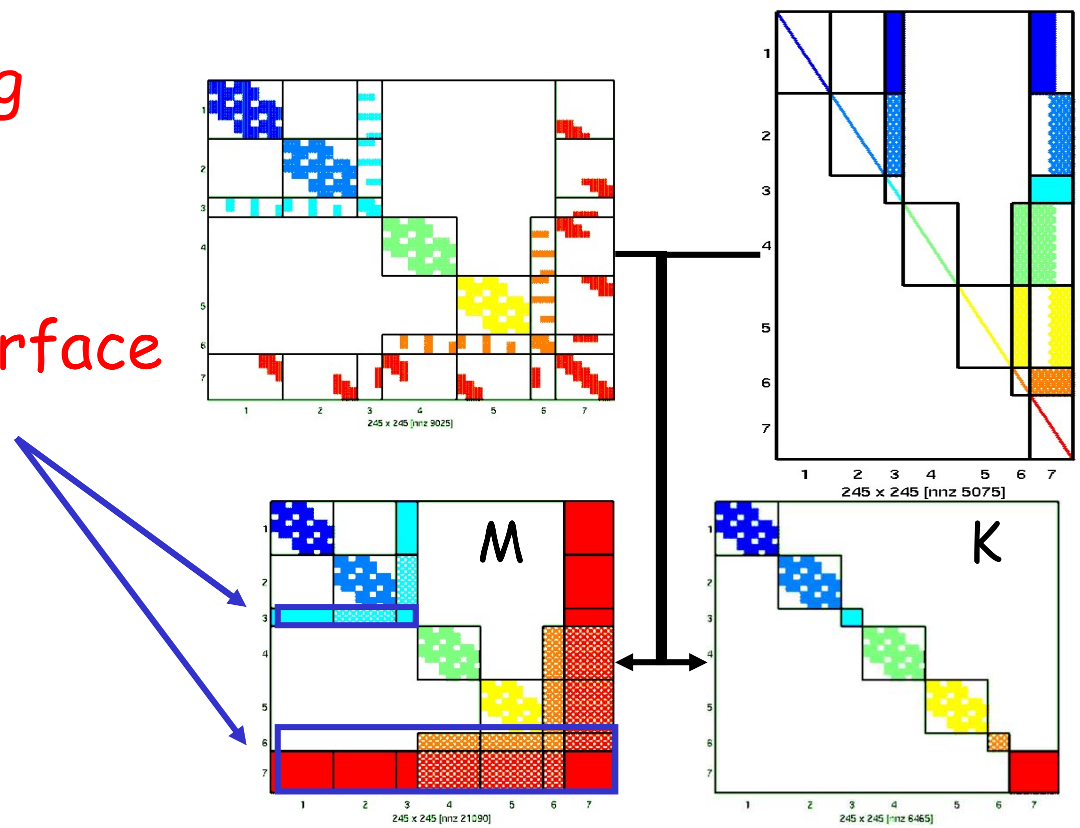
bandwidth, inputs external & parameter
truncation, sparsity

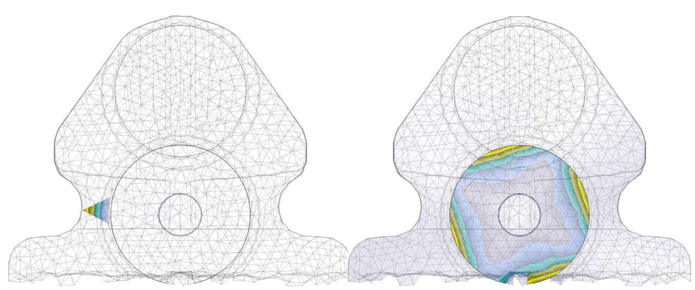
Multi-frontal solvers / AMLS

- Graph partionning methods $\Rightarrow$ group DOFs in an elimination tree with separate branches
- Block structure of reduction basis
- Block diagonal stiffness
- Very populated mass coupling



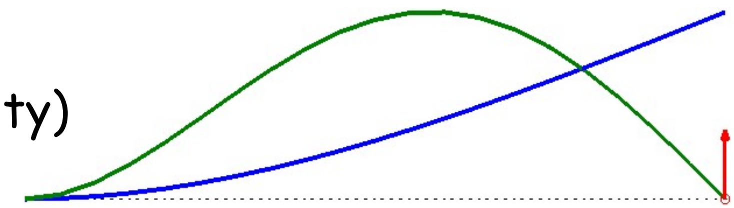
- Multi-frontal eigensolvers introduce some form of interface modes to limit size of mass coupling



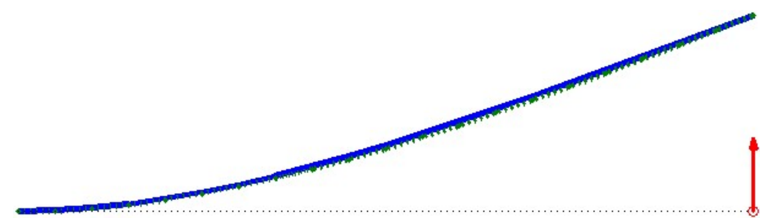


CMS current practice

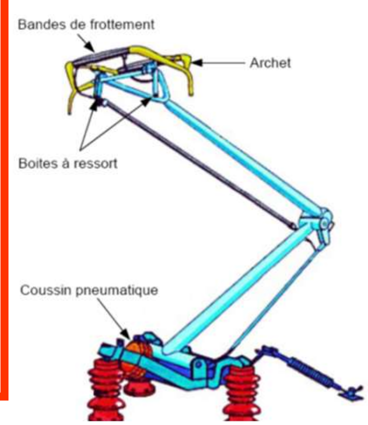
- Craig-Bampton (unit displacements + fixed interface modes)
  - Very robust, guaranteed independence
- McNeal (free modes + static response to loads)
  - Tends to have poor conditioning (residual flexibility)

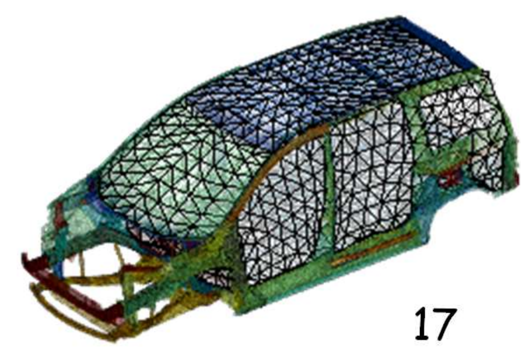


- Well established applications
  - structural vibrations
  - multi flexible-bodies
  - vibroacoustics



- Limits
  - Very large models
  - Large interfaces
  - Parametric design of component
  - Non local or strong coupling (reduction not independent)
  - Hybrid test/analysis
  - …
  - Ease of use

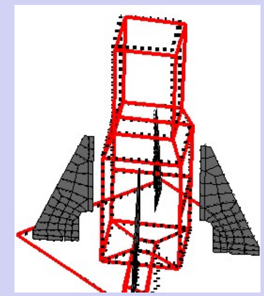


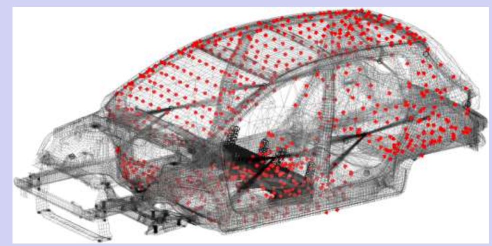


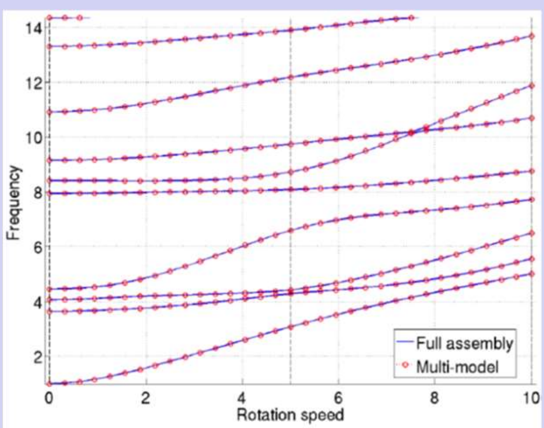
Moving complexity in the coupling part

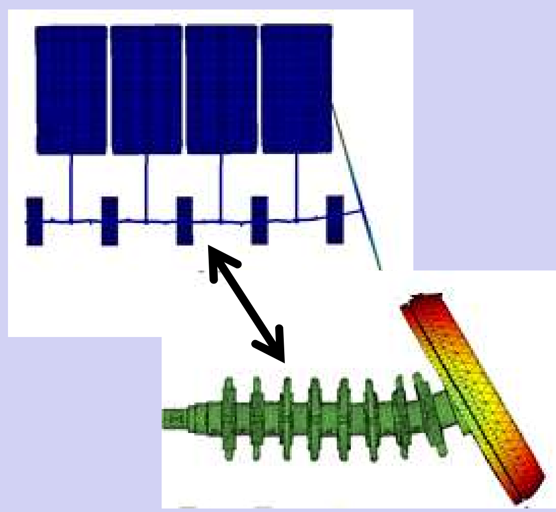
In → Reduced model → Sensors

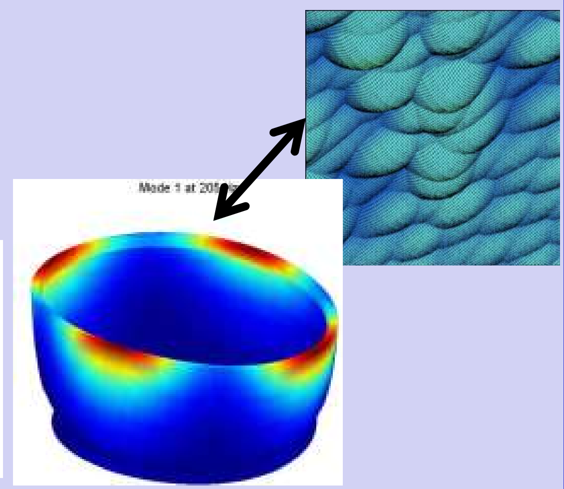
- Coupling : test/FEM, fluid/structure active control, …
- Local non-linearities : machining, bearings, contact/friction, …
- Optimization / uncertainty











Equations of motion

FEM ⇔ Reduction

	Finite elements Continuous → discrete full	Reduction Full → reduced
Support	Element: line, tria, tetra, …	FE mesh
Variable separ. Shape functions	$w(x,t) = N_i(x) q_i(t)$ $\epsilon(x,t) = B_i(x) q_i(t)$	$\{q(t)\} = \{T_i\}_{N \times N_R} \{q_i(t)\}_{N_R}$ T_i simple FE solutions
Matrix comp. Weak form	$K_{ij} = \int_{\Omega} B_i^T \Lambda B_j = \sum_g B_i^T(g) \Lambda B_j w_g J_g$ numerical integration	$K_{ijR} = T_i^T K T_j$ FEM matrix projection
Assembly	Localization matrix	Boundary continuity, CMS
Validity	Fine mesh for solution gradients	Good basis for considered loading

Target defined by load {f}=[b]{u}

- space [b]
- time/freq {u}

[1] O. C. Zienkiewicz et R. L. Taylor, *The Finite Element Method*. MacGraw-Hill, 1989
[2] J. L. Batoz et G. Dhatt, *Modélisation des Structures par Éléments Finis*. Hermès, Paris, 1990
[3] K. J. Bathe, *Finite Element Procedures in Engineering Analysis*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1982

Ritz/Galerkin reduction from full

- Basis building steps
  - FEM : cinematically admissible subspace, virtual work principle
  - Reduction : 1) learn, 2) generate basis 3) choose DOF

$$\{q(p,t)\}_N \approx [T]_{N\times NR}\{q_R(p,t)\}_{NR}$$

- Virtual work principle / reduction / Ritz-Galerkin

  Matrices $[M_R(p)] = T^T M(p) T, K_R(p) = T^T K(p) T$

  Loads $\{\mathrm{f}(p,t)\} = [b_R(p)]\{u(t)\} = [T^T b]\{u\}$

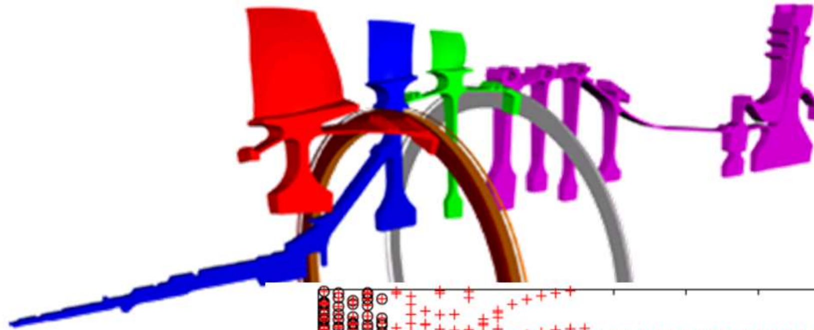
  Observations $\{\mathrm{y}(p,t)\} = [c_R(p)]\{q_R(p,t)\} = [cT]\{q_R\}$

- Solve time/freq (same model form)

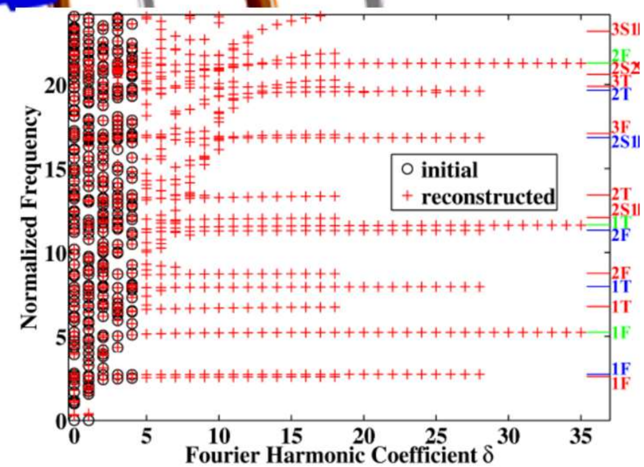
$$\begin{array}{rcl} [M_R]\{\ddot{q_R}\} + [C_R]\{\dot{q_R}\} + [K_R]\{q_R\} & = & [b]\{u(t)\} \\ \{y(t,p)\} & = & [c_R]\{q_R\} \end{array}$$

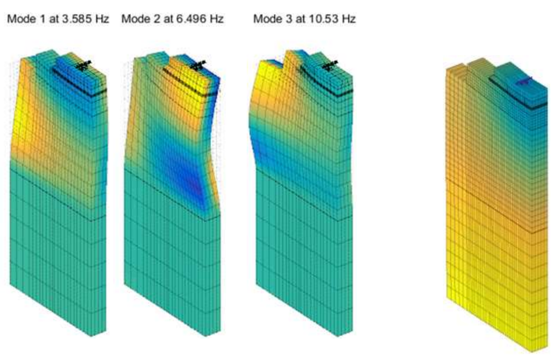
Interface reduction : wave/cyclic

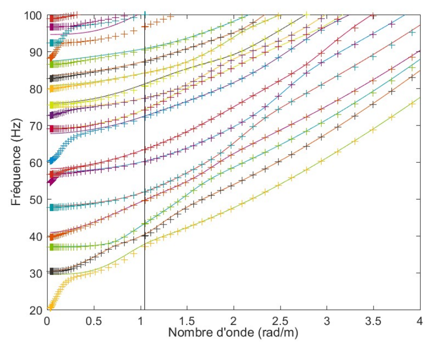
Best interface reduction = learn from full system modes

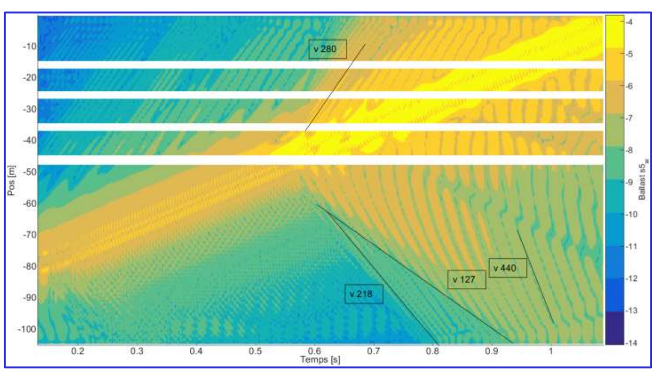


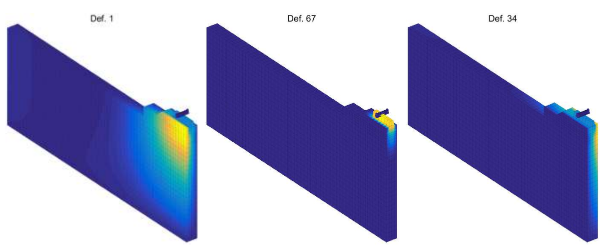
1. Learn using wave (Floquet)/cyclic solutions
2. Build basis with left/right compatibility
3. Assemble reduced model



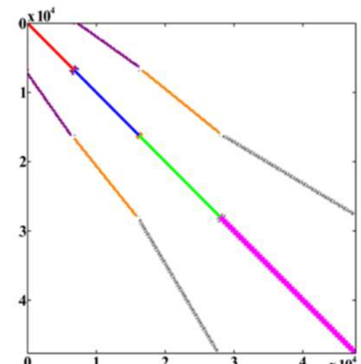








PhD Elodie Arlaud, 2016
PhD Hadrien Pinault, 2020



PhD Sternshuss 2008