I.1.(d) Algebraic Substructuring Method

Background

- Substructuring dates back to the 1960s, e.g., CMS
- Connection with domain decomposition methods
- Substructuring holds great promoise for solving extremely large scale problems, e.g., commercial AMLS
- Open questions as the techniques extended for broader applications:
  - arbitrary eigenmodes
  - accuracy
  - high frequency response analysis
  - performance optimization (memory, out-of-core, ...)

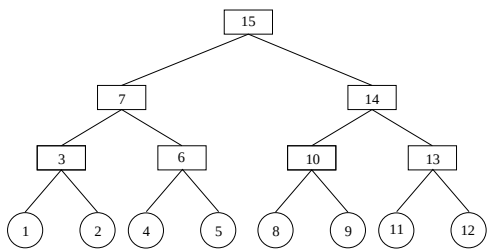
Eigenvalue problem and multi-level substructure

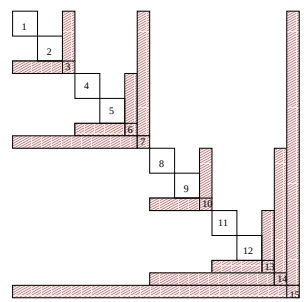
- Eigenvalue problem: $Kq = \lambda Mq$, $K^T = K$, $M^T = M > 0$
- Substructure partition

  Single-level

$$K = \begin{bmatrix} K_{11} & & K_{13} \\ & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix}, \quad M = \begin{bmatrix} M_{11} & & M_{13} \\ & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}$$

  Multi-level: nested dissection (ND)





- Can use METIS to record any sparse matrix to the ND form.

Algebra substructuring algorithm

1. Perform a congruence transformation such that

$$L^{-T}(K,M)L^{-1} = \left(\begin{bmatrix} K_{11} & & \\ & K_{22} & \\ & & \widehat{K}_{33} \end{bmatrix}, \begin{bmatrix} M_{11} & & \widehat{M}_{13} \\ & M_{22} & \widehat{M}_{23} \\ \widehat{M}_{31} & \widehat{M}_{32} & \widehat{M}_{33} \end{bmatrix} \right) \equiv (\widehat{K}, \widehat{M})$$

   this is so-called the Craig-Bampton form in structure dyanmics.

2. Compute **partial** local modes by local cutoff values:

$$\begin{aligned} K_{11}S_1 &= M_{11}S_1\Theta_1 \\ K_{22}S_2 &= M_{22}S_2\Theta_2 \end{aligned} \quad \longrightarrow \quad S_m = \begin{bmatrix} S_1 & & \\ & S_2 & \\ & & I \end{bmatrix}$$

3. Project onto the AS subspace $\{L^{-1}S_m\}$:

$$(L^{-1}S_m)^T(K,M)(L^{-1}S_m) = (K_m, M_m)$$

4. Solve the reduced eigenvalue problem:

$$K_m\Phi = M_m\Phi\Theta,$$

5. Compute the global modes $\Phi = \left[\, \Phi_l \;\; \Phi_n \;\; \Phi_r \,\right]$

   Φ_n are retained modes determined by cutoff values

   $\Phi_t = \left[\, \Phi_l \;\; \Phi_r \,\right]$: truncated modes

6. Return approximate eigenpairs

   $(\theta^\sigma + \sigma, L^{-1} S_m \phi) \approx (\lambda, q)$ of (K, M)

Why (when) AS works?

- Consider the **full** eigendecomposition of (K_{11}, M_{11}) and (K_{22}, M_{22})

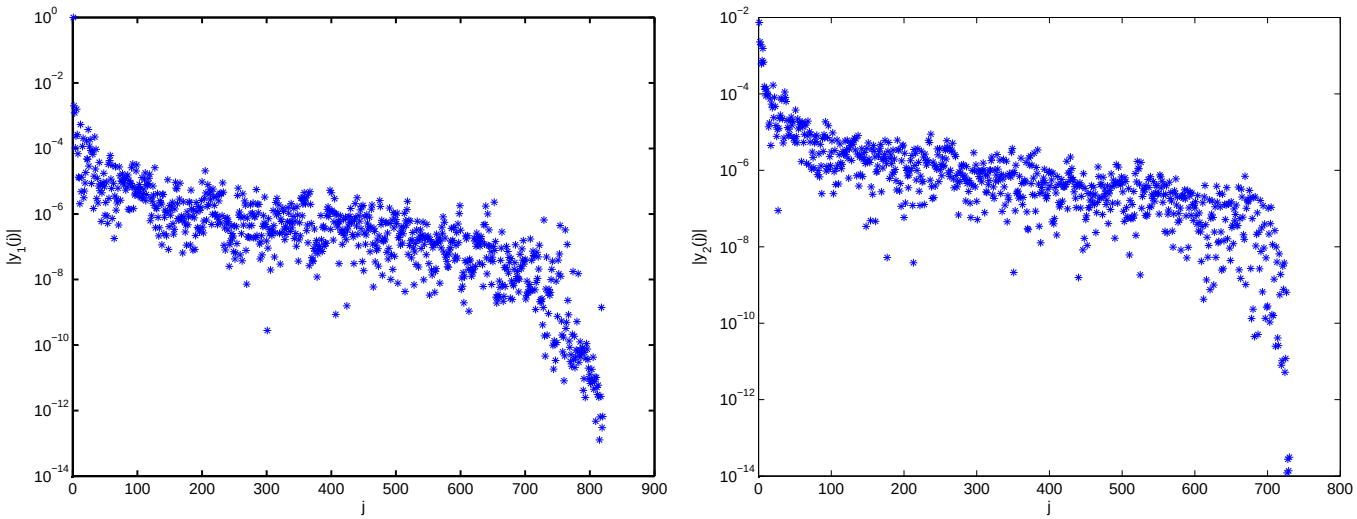
$$K_{11}V_1 = M_{11}V_1\Lambda_1, \quad K_{22}V_2 = M_{22}V_2\Lambda_2.$$

- An eigenvector $\widehat{x}$ of $(\widehat{K}, \widehat{M})$ can be expressed as the follows

$$\widehat{x} = Vy = \begin{bmatrix} V_1 & & \\ & V_2 & \\ & & I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

- If $\widehat{x}$ can be well approximated by a linear combination of $S = \begin{bmatrix} S_1 & & \\ & S_2 & \\ & & I \end{bmatrix}$, then the vectors y_1 and y_2 must contain only a few large entries, and all components of y_1 and y_2 are likely to be small and negligible – *key observation/requriement*

- An example of the magnitude of y_1 and y_2:



(a finite element model corresponding to a five-cell traveling wave accelerating structure)

- For a formal analysis to derive an *a priori* error bound of the smallest eigenpair in terms of the small components of y_1 and y_2, see [Yang *et al*'05, Voss *et al*].

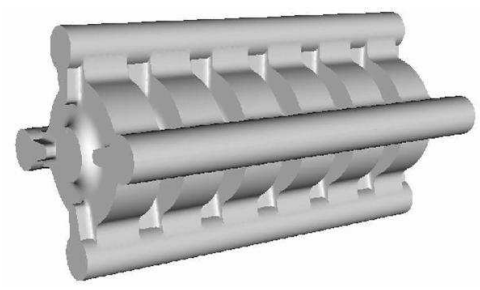
Implementation and performance evaluation

- Major operations:
  1. block elimination and projection
  2. projected eigenvalue problem
- Costs
  1. flops: more than a single sparse Cholesky factorization
  2. storage: block Cholesky factors + projected matrices + ...

  But no triangular solvers, no (re)-orthogonalization
- ASEIG package ([Gao *et al*, ACM/TOMS 2008])
  1. interleave different steps
  2. recompute some of intermediate matrix blocks instead of storing. *50% of memory saving with about extra 15% recompute time*

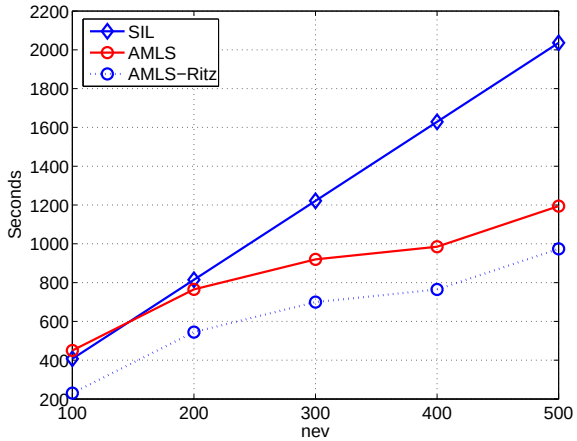
Case study: accelerator cavity design (SLAC)

- Elecromagnetic modeling of a 6-cell DDS structure for the design of next generation accelerator (SLAC)



- Computing small eigenvalues (tightly clustered) out of a large-eigenvalue dominated eigenspetrum

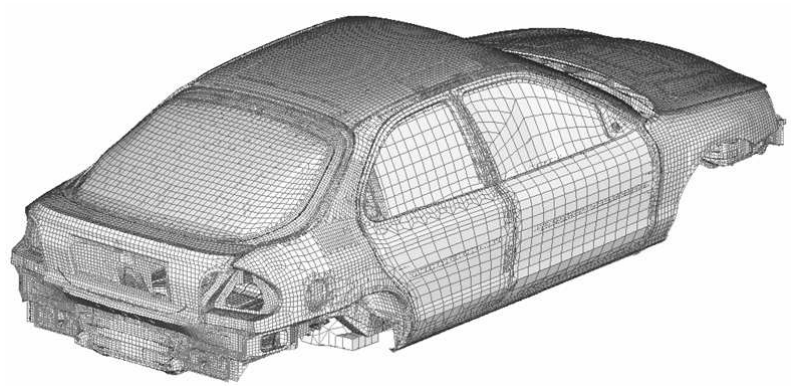
- $N = 65730$
- 4-level AS, $n_{\text{proj}} \approx 3000$
- Many eigenvalues are wanted (up to 8%)
- SIL reqiures multiple shifts (factorizations)



An industrial case study: NVH analysis

Modal analysis for noise, vibration and harshness (NVH)

- DOFs = 1,584,622
- 1073 eigenmodes
- SI Lanczos method
- NX NASTRAN



		CPU(min)	I/O(GB)	Wallclock(min)
sequential	SIL	283	2476	721
	AS	103	1402	167
parallel	SIL	45	325	85
	AS	9	98	47

16-node AMD Opteron (1.8GHz), 4GB RAM, 73GB per node

Further reading

The study of substructuring methods from an linear algebra point of view can be found in

- C. Yang, W. Gao, Z. Bai, X. Li, L. Lee, P. Husbands, E. Ng, *An algebraic sub-structuring method for large-scale eigenvalue calculation*, SIAM J. Sci. Comput. 27:873-892, 2005
- K. Elssel and H. Voss, *An a priori bound for Automated Multi-Level Substructuring*, SIAM J. Matrix Anal. Appl. 28:386-397, 2006

An implementation of the AS method is described in the following paper

- W. Gao, X. Li, C. Yang and Z. Bai, *An implementation and evaluation of the AMLS method for Sparse Eigenvalue Problems*, ACM Trans. Math. Software. Vol.34, 20:2–20:28, 2008