Block-Diagonal Decomposition of Matrices based on $*$-Algebra for Structural Optimization with Symmetry Property

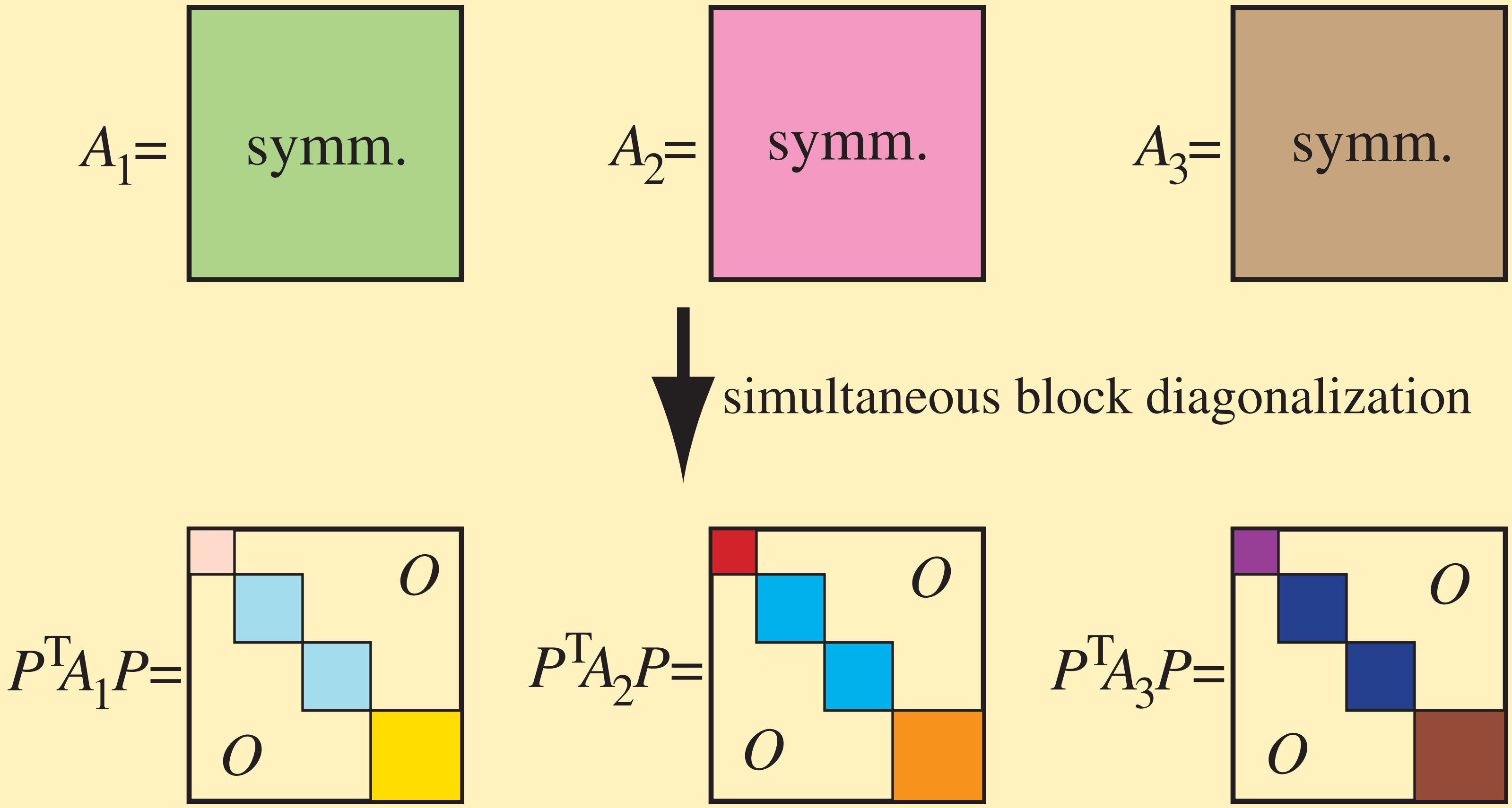
Yoshihiro Kanno[†] Kazuo Murota[†]
Masakazu Kojima[‡] Sadayoshi Kojima[‡]

[†]University of Tokyo (Japan)
[‡]Tokyo Institute of Technology (Japan)

June 16, 2008

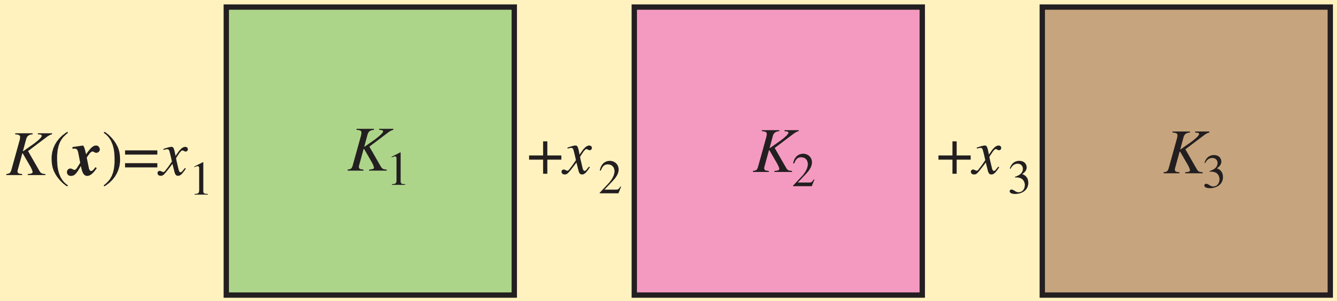
simultaneous block-diagonal decomposition

- given: $A_1, A_2, \ldots, A_m$ — symmetric $n \times n$
- find: $P \in \mathbb{R}^{n \times n}$ — orthogonal



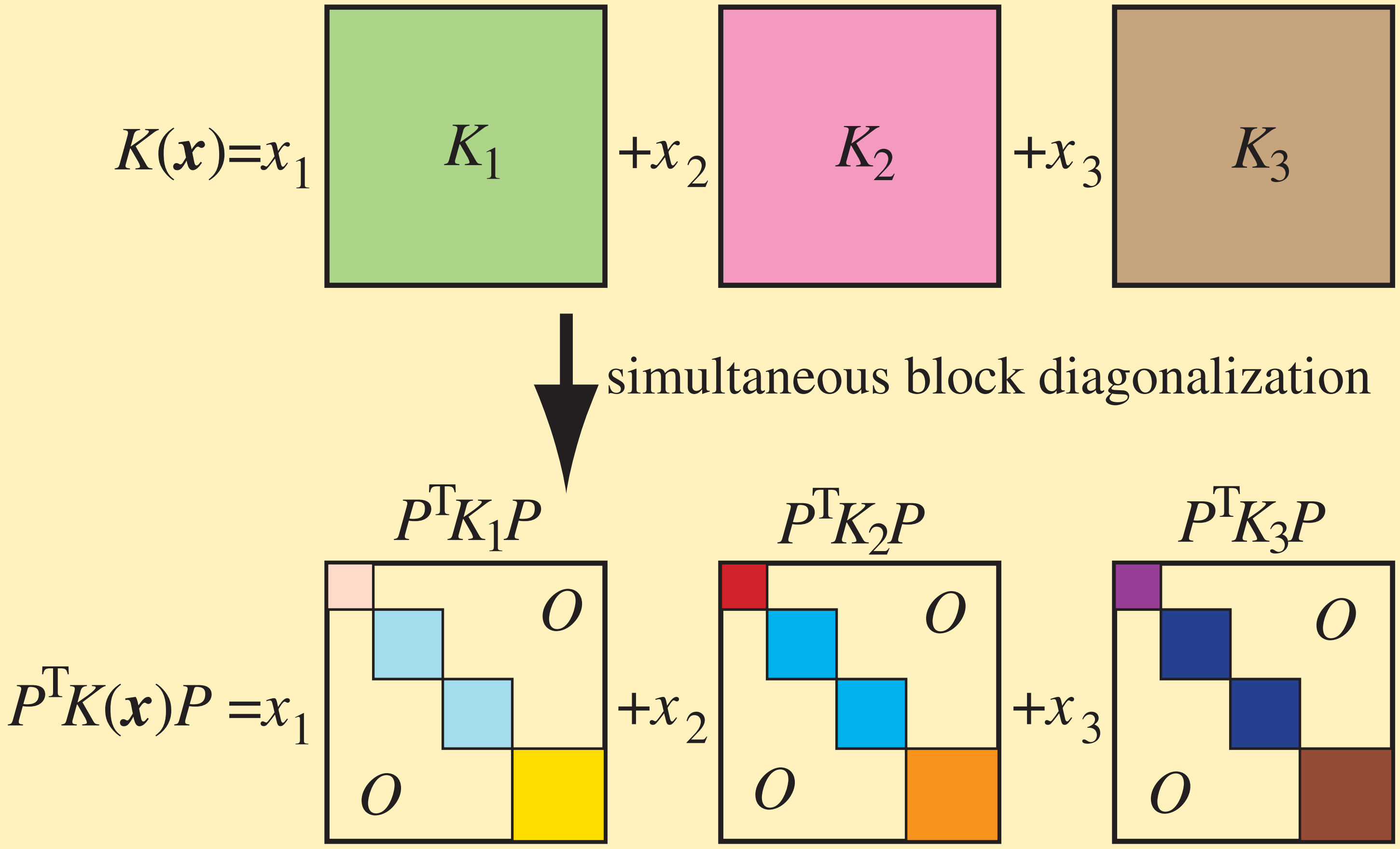
simultaneous block-diagonal decomposition: application

- $K(\boldsymbol{x})$: stiffness matrix $\boldsymbol{x}$: design variables
- given: $K_1, K_2, \ldots, K_m$
- find: $P \in \mathbb{R}^{n \times n}$ — orthogonal



simultaneous block-diagonal decomposition: application

- $K(\boldsymbol{x})$: stiffness matrix $\boldsymbol{x}$: design variables
- given: $K_1, K_2, \ldots, K_m$
- find: $P \in \mathbb{R}^{n \times n}$ — orthogonal



application: SDP

$$\begin{array}{ll} \min & \sum_{p=1}^{m} b_p y_p \\ \text{s.t.} & C - \sum_{p=1}^{m} A_p y_p \succeq O \end{array}$$

variables : $y_1, \ldots, y_m$

coefficients : $b_1, \ldots, b_m,$

$A_1, \ldots, A_m,\ C \in \mathcal{S}^n$ $\quad n \times n$ symmetric matrices

- $X \succeq \boldsymbol{O} \iff X$ is positive semidefinite

  $\leftarrow$ nonlinear but convex

application: SDP

$$\begin{array}{ll} \min & \sum_{p=1}^{m} b_p y_p \\ \text{s.t.} & C - \sum_{p=1}^{m} A_p y_p \succeq O \end{array}$$

- truss optimization (with eigenvalue constraints)

$$\begin{array}{ll} \min & vol(\boldsymbol{x}) \\ \text{s.t.} & \Omega_1(\boldsymbol{x}) \geq \bar{\Omega} \end{array}$$

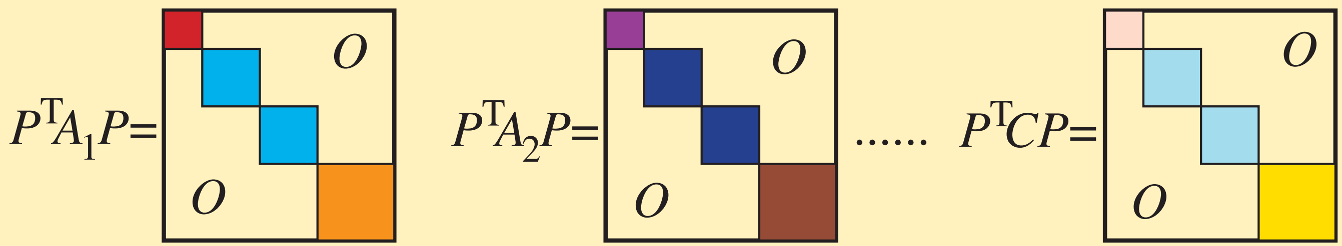
can be solved by SDP [Ohsaki *et al.* 99] as

$$\begin{array}{ll} \min & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} & \sum_{p=1}^{m}(K_p - \bar{\Omega} M_p) y_p \succeq O \end{array}$$

- optimization with buckling load factor constraints
- robust structural optimization

application: SDP

$$\begin{array}{ll} \min & \sum_{p=1}^{m} b_p y_p \\ \text{s.t.} & C - \sum_{p=1}^{m} A_p y_p \succeq O \end{array}$$



$$\Downarrow$$

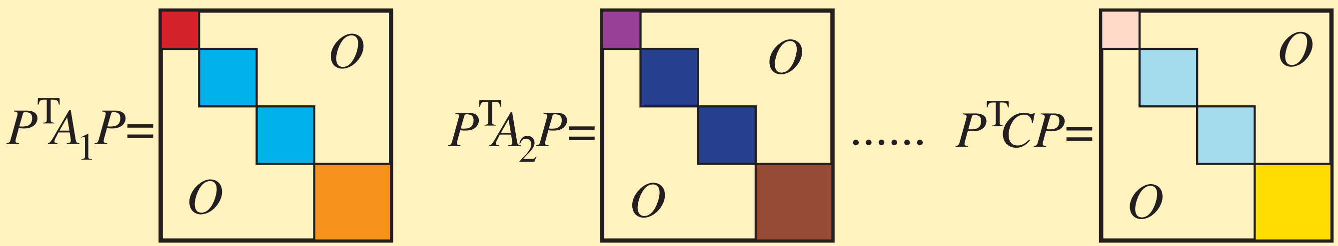
$$C^{(1)} - \sum_{p=1}^{m} A_p^{(1)} y_p \succeq O$$

$$\vdots$$

$$C^{(\ell)} - \sum_{p=1}^{m} A_p^{(\ell)} y_p \succeq O$$

application: SDP

$$\begin{array}{ll} \min & \sum_{p=1}^{m} b_p y_p \\ \text{s.t.} & C - \sum_{p=1}^{m} A_p y_p \succeq O \end{array}$$



- group theory
  [Gatermann & Parrilo 04], [Bai & de Klerk 07] [de Klerk & Sotirov 07]
- matrix $*$-algebra [de Klerk, Pasechnik & Scrijver 07]
- aim: numerical algorithm
  which can be executed without any knowledge
  of symmetry property in advance

matrix $*$-algebra

$\mathcal{T} \subseteq \mathbb{R}^{n \times n}$ (set of $n \times n$ matrices)

is a matrix $*$-algebra

$\Updownarrow$

- $I_n \in \mathcal{T}$
- If $A, B \in \mathcal{T}$, $\alpha \in \mathbb{R}$, then
  - $A + B \in \mathcal{T}$
  - $AB \in \mathcal{T}$
  - $\alpha A \in \mathcal{T}$
  - $A^{\mathrm{T}} \in \mathcal{T}$

structure theorem

1. $\exists Q$: orthogonal s.t.

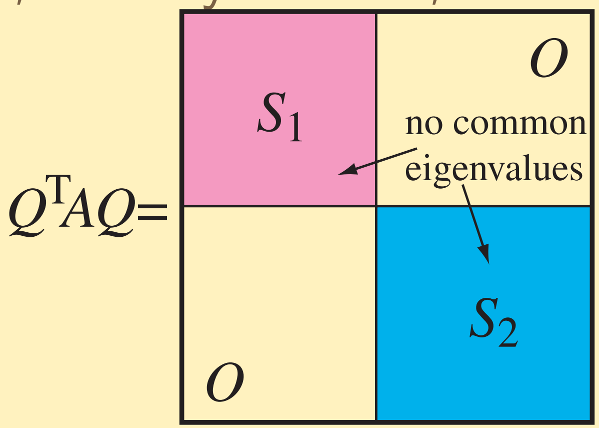
$$Q^{\mathrm{T}} \mathcal{T} Q = \left[\begin{array}{c|c|c|c} S_1 & O & \cdots & O \\ \hline O & S_2 & O & O \\ \hline \vdots & O & \ddots & O \\ \hline O & O & \cdots & S_l \end{array}\right], \quad S_j \in \mathcal{T}_j \text{ : simple}$$

2. If T_j is simple, then $\exists \hat{P}$: orthogonal s.t.

$$\hat{P}^{\mathrm{T}} \mathcal{T}_j \hat{P} = \left[\begin{array}{c|c|c|c} B_j & O & O & O \\ \hline O & B_j & O & O \\ \hline O & O & \ddots & O \\ \hline O & O & O & B_j \end{array}\right], \quad B_j \in \hat{\mathcal{T}}_j \text{ : irreducible}$$

structure theorem: transformations to

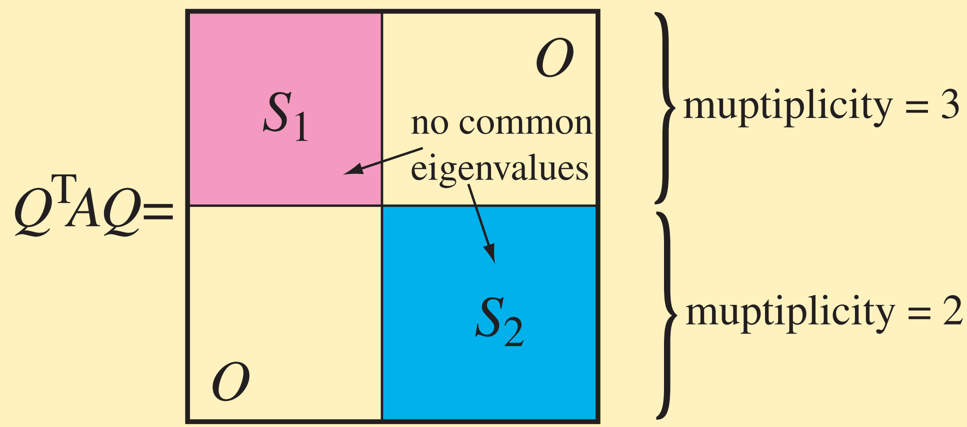
- simple components
  $\exists Q$: orthogonal, for any $A \in \mathcal{T}$,



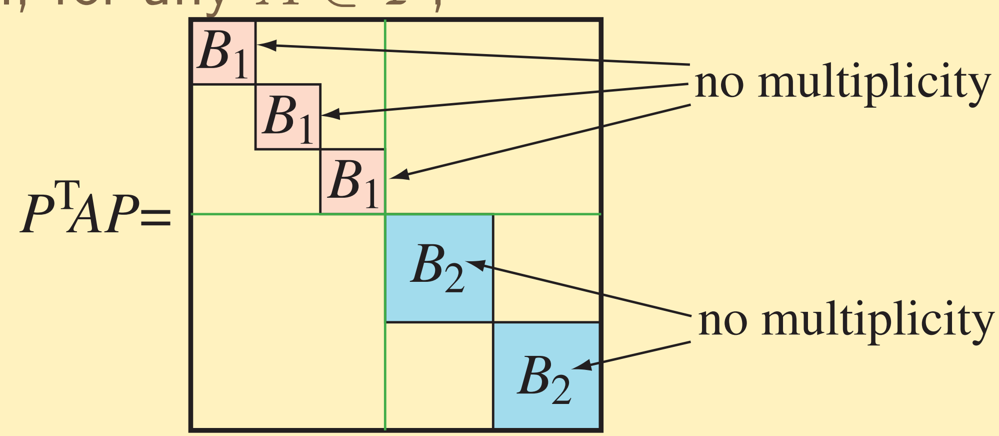
- & irreducible components
  $\exists \hat{P}$: orthogonal, for any $A \in \mathcal{T}$,

structure theorem: transformations to

- simple components
  $\exists Q$: orthogonal, for any $A \in \mathcal{T}$,



- & irreducible components
  $\exists \hat{P}$: orthogonal, for any $A \in \mathcal{T}$,



algorithm (1)
input: $A_1, A_2, \ldots, A_m$

1. Generate random $r_1, \ldots, r_m$. Put $X \equiv \sum_{i=1}^{m} r_i A_i$.

2. Eigenvalue decomposition of X:

$$Q^{\mathrm{T}} X Q = \left[\begin{array}{c|c|c|c} \alpha_1 I_{m_1} & O & O & O \\ \hline O & \alpha_2 I_{m_2} & O & O \\ \hline O & O & \ddots & O \\ \hline O & O & O & \alpha_k I_{m_k} \end{array}\right],$$

$$Q = \left[Q_1, Q_2, \ldots, Q_k \right].$$

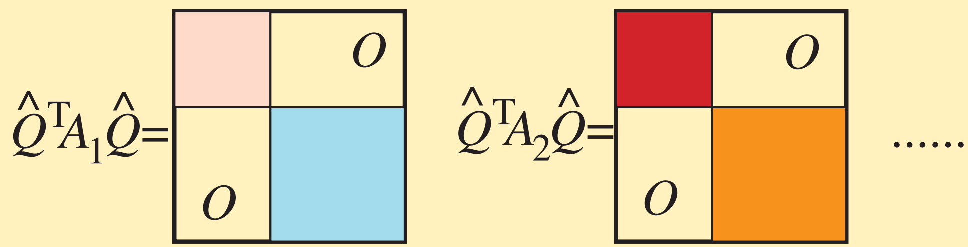
3. Compute $Q^{\mathrm{T}} A_1 Q, \ldots, Q^{\mathrm{T}} A_m Q$.

algorithm (1)
input: $A_1, A_2, \ldots, A_m$

1. Generate random $r_1, \ldots, r_m$. Put $X \equiv \sum_{i=1}^{m} r_i A_i$.
2. Eigenvalue decomposition of X:
3. Compute $Q^{\mathrm{T}} A_1 Q, \ldots, Q^{\mathrm{T}} A_m Q$.
4. Rearrange eigenvectors Q as

$$\hat{Q} = [Q[K_1], Q[K_2], \ldots, Q[K_l]]$$

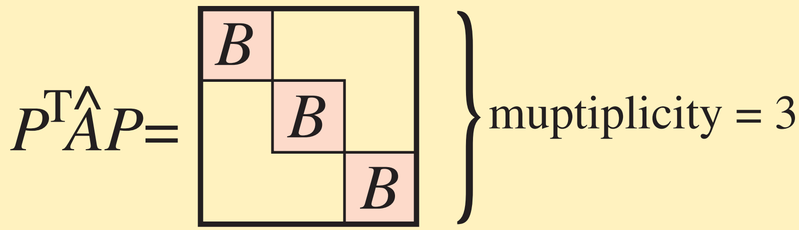
so that



which is the decomposition to simple components.

algorithm (2)
input: $\hat{A}_1, \hat{A}_2, \ldots, \hat{A}_m$ — simple

1. We know that the simple component is decomposed as



2. e.g. $B \in \mathbb{R}^{2\times 2}$, $P^{\mathrm{T}}\hat{A}P$ can be rearranged as

$$\tilde{P}^{\mathrm{T}}\hat{A}\tilde{P} = \left[\begin{array}{c|c|c} b_{11}I & b_{12}I & b_{13}I \\ \hline b_{21}I & b_{22}I & b_{23}I \\ \hline b_{31}I & b_{32}I & b_{33}I \end{array}\right], \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \qquad (\clubsuit)$$

We first find the transformation ($\clubsuit$).

algorithm (2)
input: $\hat{A}_1, \hat{A}_2, \ldots, \hat{A}_m$ — simple

1. In order to obtain

$$\tilde{P}^{\mathrm{T}} \hat{A} \tilde{P} = \left[\begin{array}{c|c|c} b_{11} I & b_{12} I & b_{13} I \\ \hline b_{21} I & b_{22} I & b_{23} I \\ \hline b_{31} I & b_{32} I & b_{33} I \end{array}\right], \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad (\clubsuit)$$

   we use the form of

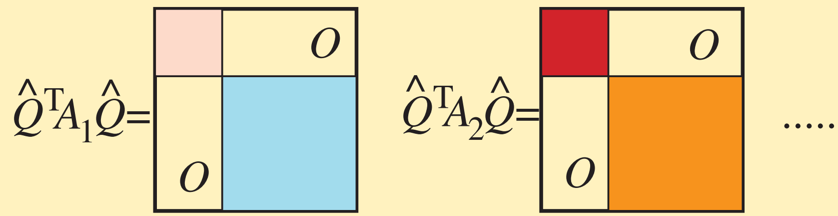
$$b_{ij} P_i P_j^{\mathrm{T}} = \hat{A}_{ij}.$$

2. Put $P_1 = I$. Then,

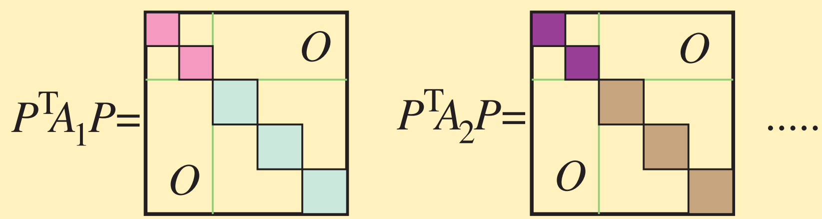
$$b_{31} P_3 P_1^{\mathrm{T}} = \hat{A}_{31} \implies P_3 = \hat{A}_{31} / b_{31}$$
$$b_{23} P_2 P_3^{\mathrm{T}} = \hat{A}_{23} \implies P_2 = \hat{A}_{23} P_3 / b_{23}$$

algorithm (1) & (2)

- (1) simple components

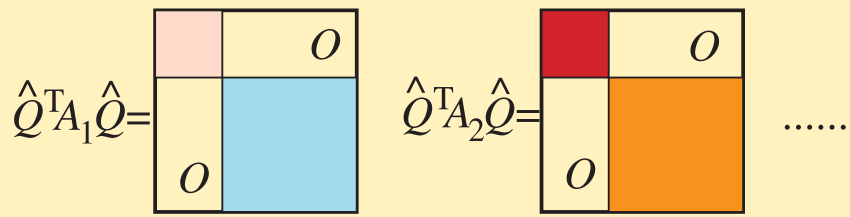


- (2) irreducible components

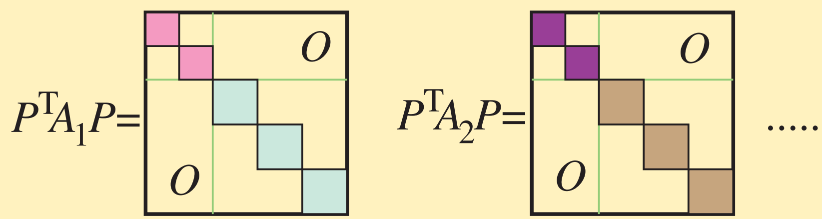


algorithm (1) & (2)

- (1) simple components



- (2) irreducible components

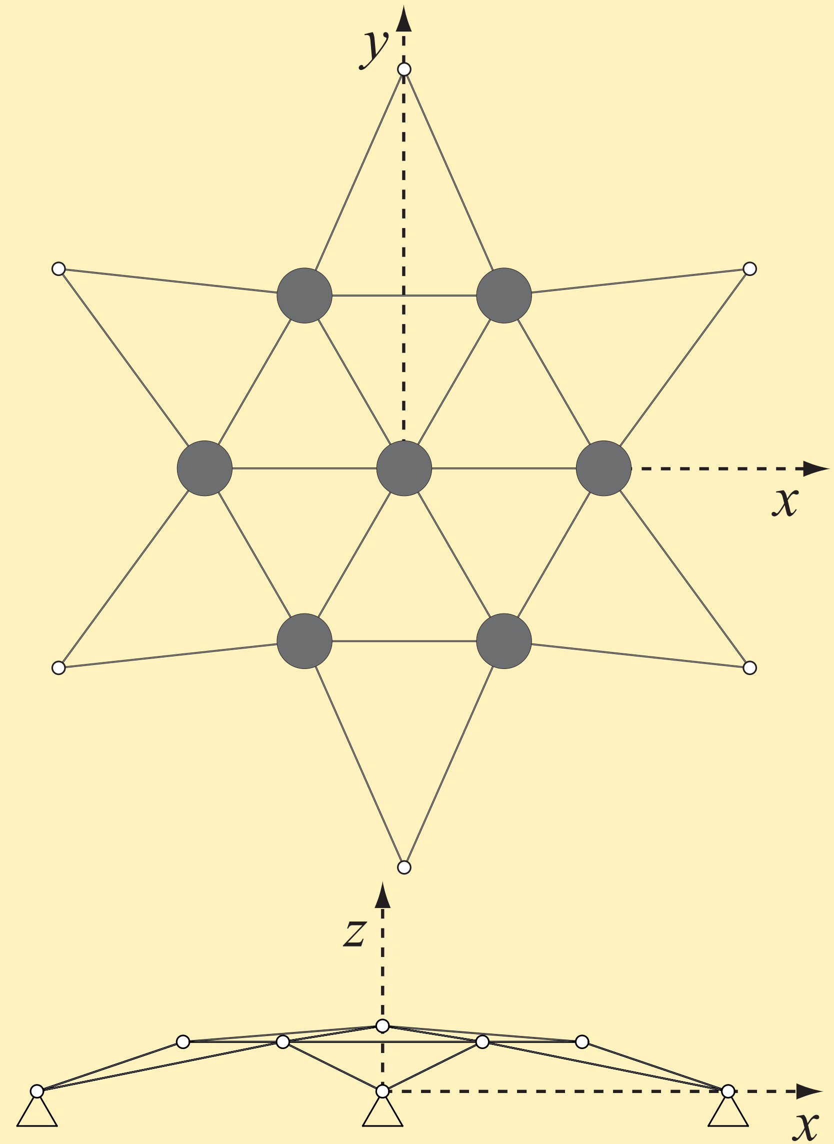


- advantages: algorithm uses
  - only linear algebraic computations (e.g. eigenvalue decomposition)
  - no knowledge of algebraic structure (symmetry property)

    execution is free from group representation theory or matrix $*$-algebra

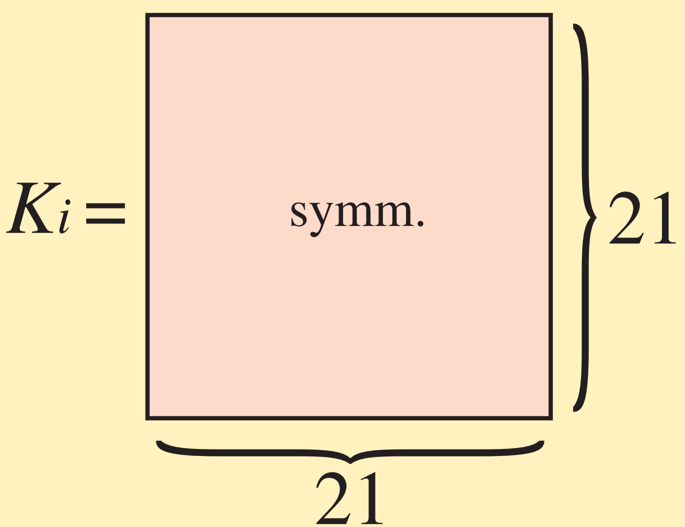
ex.) spherical dome

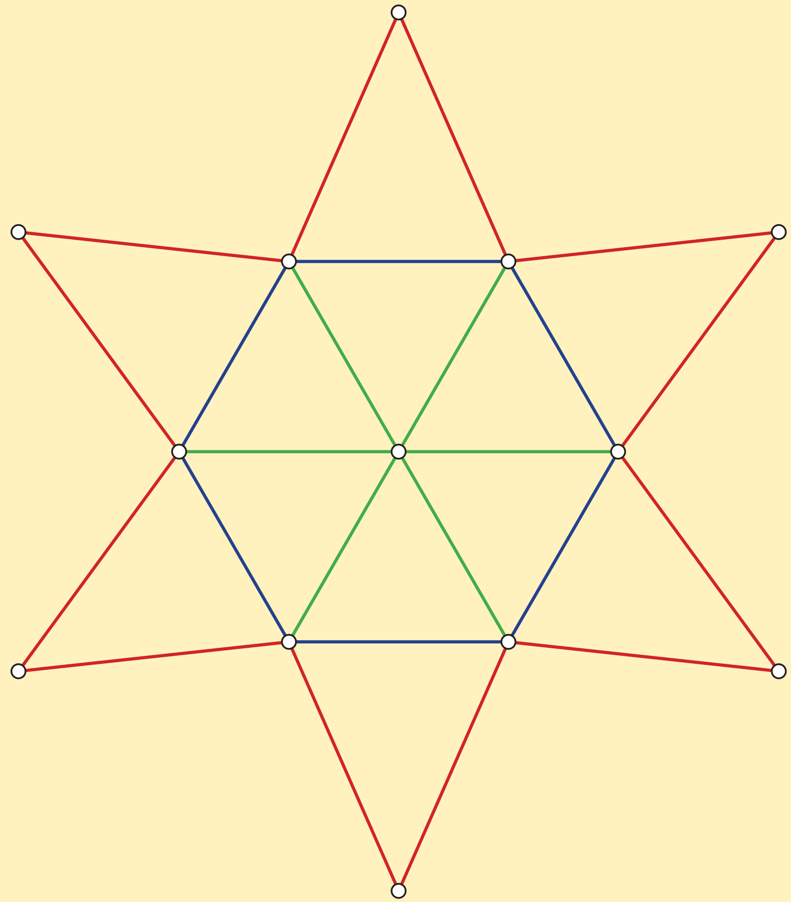
- spherical lattice dome
  D_6-symmetry



ex.) spherical dome

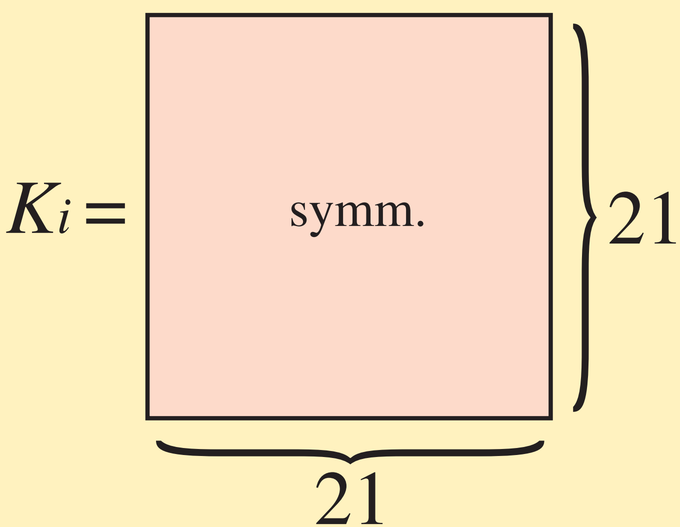
- spherical lattice dome
  D_6-symmetry
- variable linking
  3 kinds of members
  $K(\boldsymbol{x}) =$
  $x_1 K_1 + x_2 K_2 + x_3 K_3$
- 21 DOF

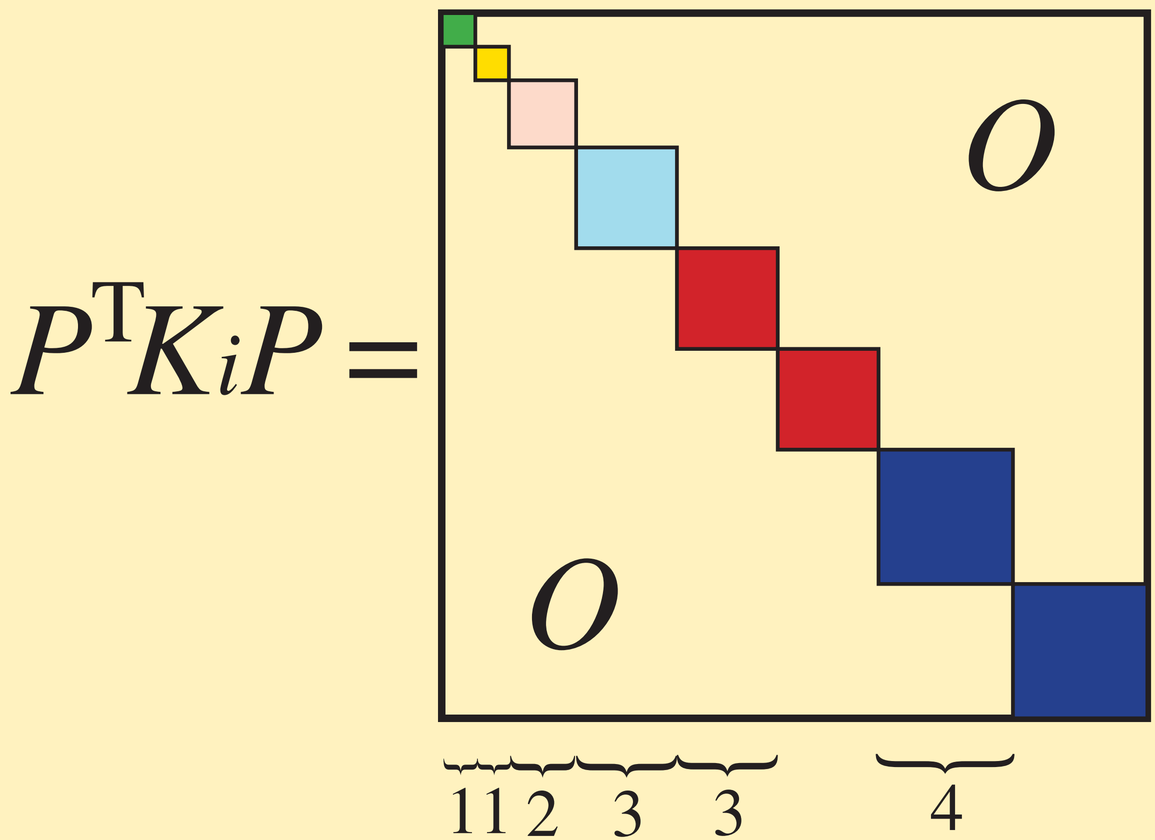




ex.) spherical dome

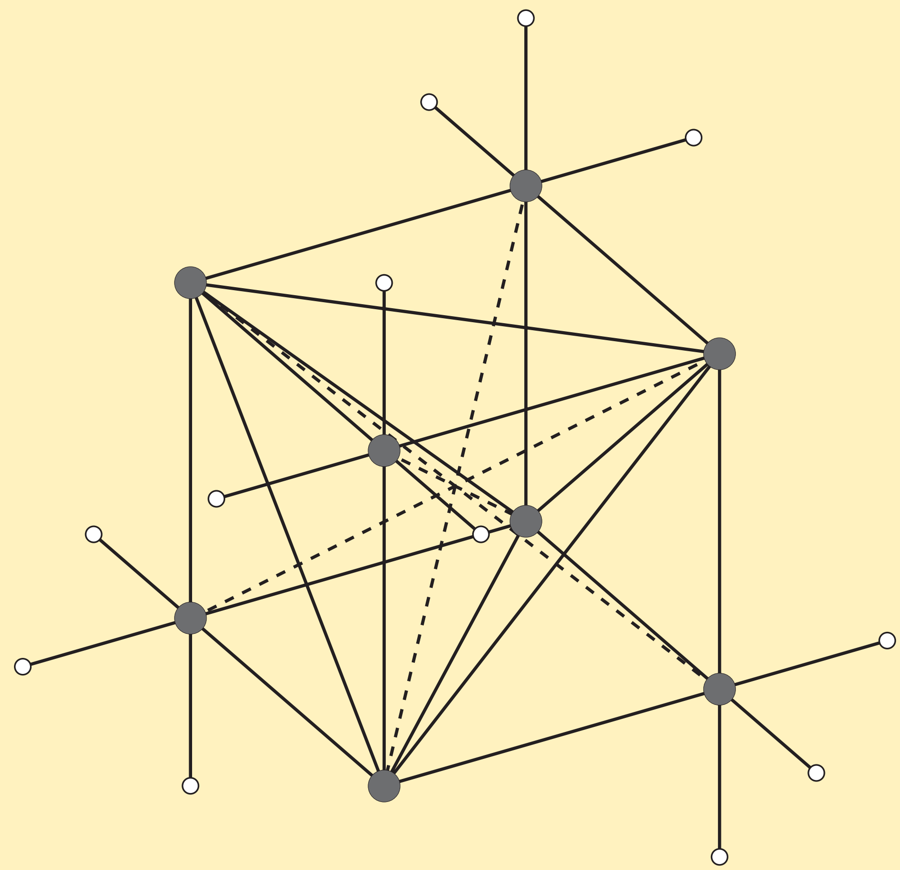
- spherical lattice dome
  D_6-symmetry
- variable linking
  3 kinds of members
  $K(\boldsymbol{x}) =$
  $x_1 K_1 + x_2 K_2 + x_3 K_3$
- 21 DOF





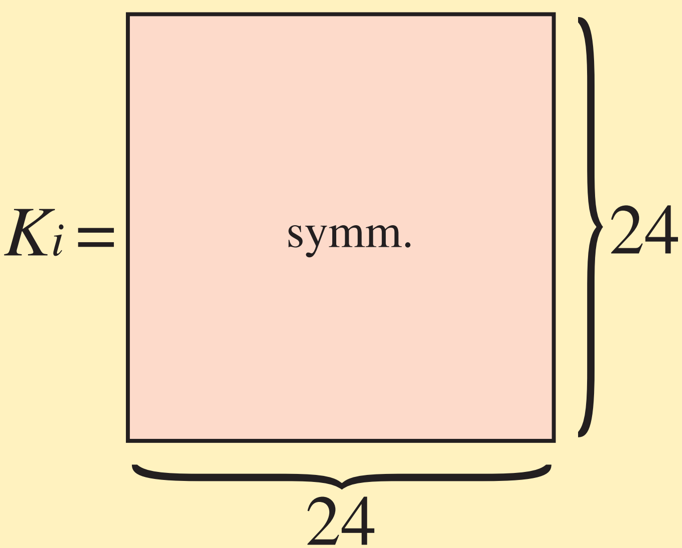
ex.) cubic truss (tetrahedral symm.)

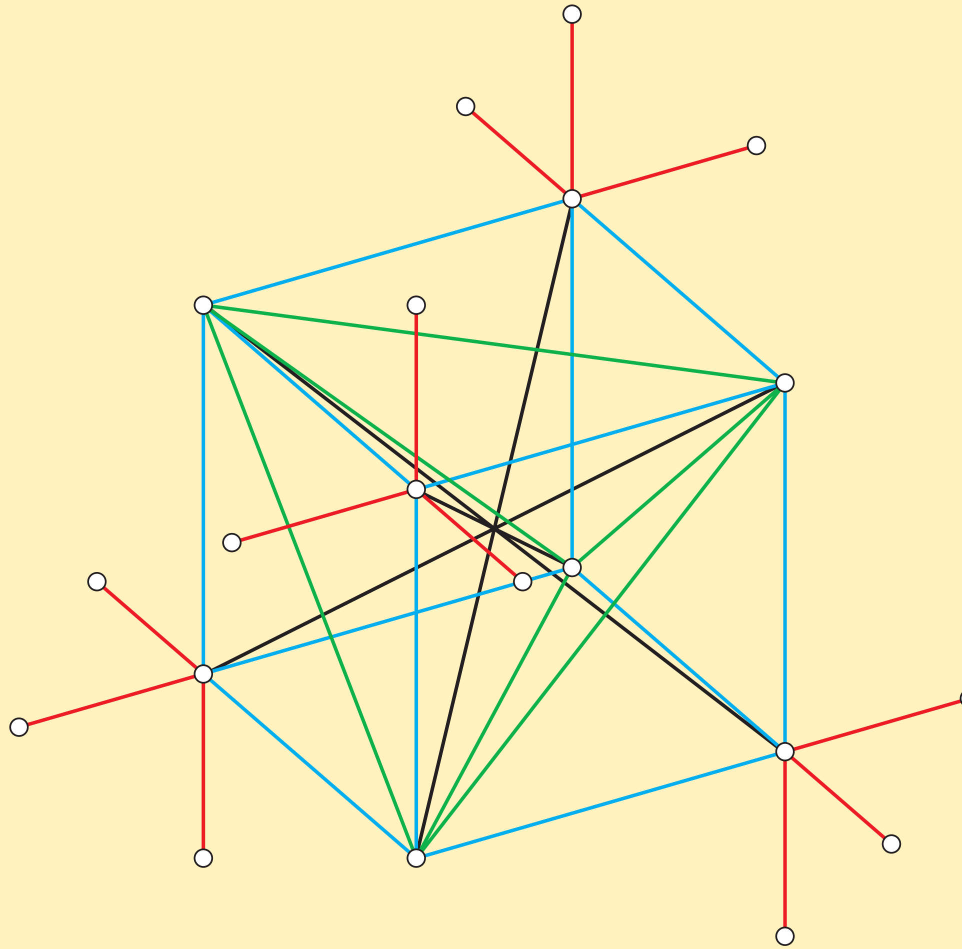
- $\mathrm{T_d}$-symmetry (tetrahedral)



- $\mathrm{T_d}$-symmetry (tetrahedral)
- variable linking
  4 kinds of members
  $K(\boldsymbol{x}) = x_1 K_1 + \cdots + x_4 K_4$

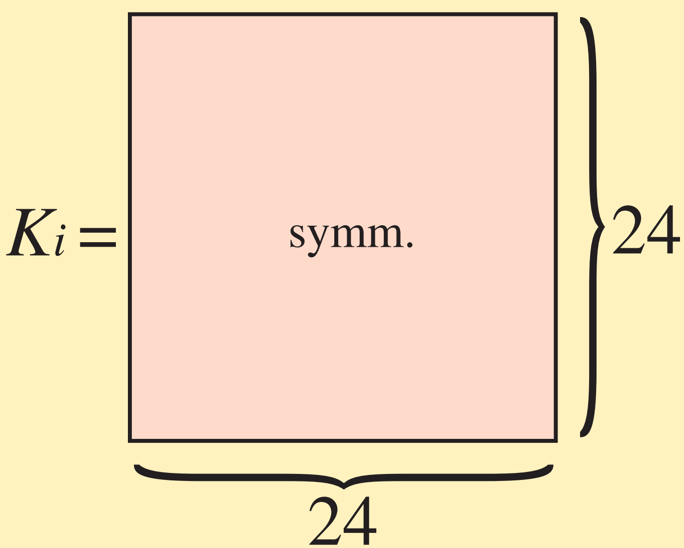
- 24 DOF

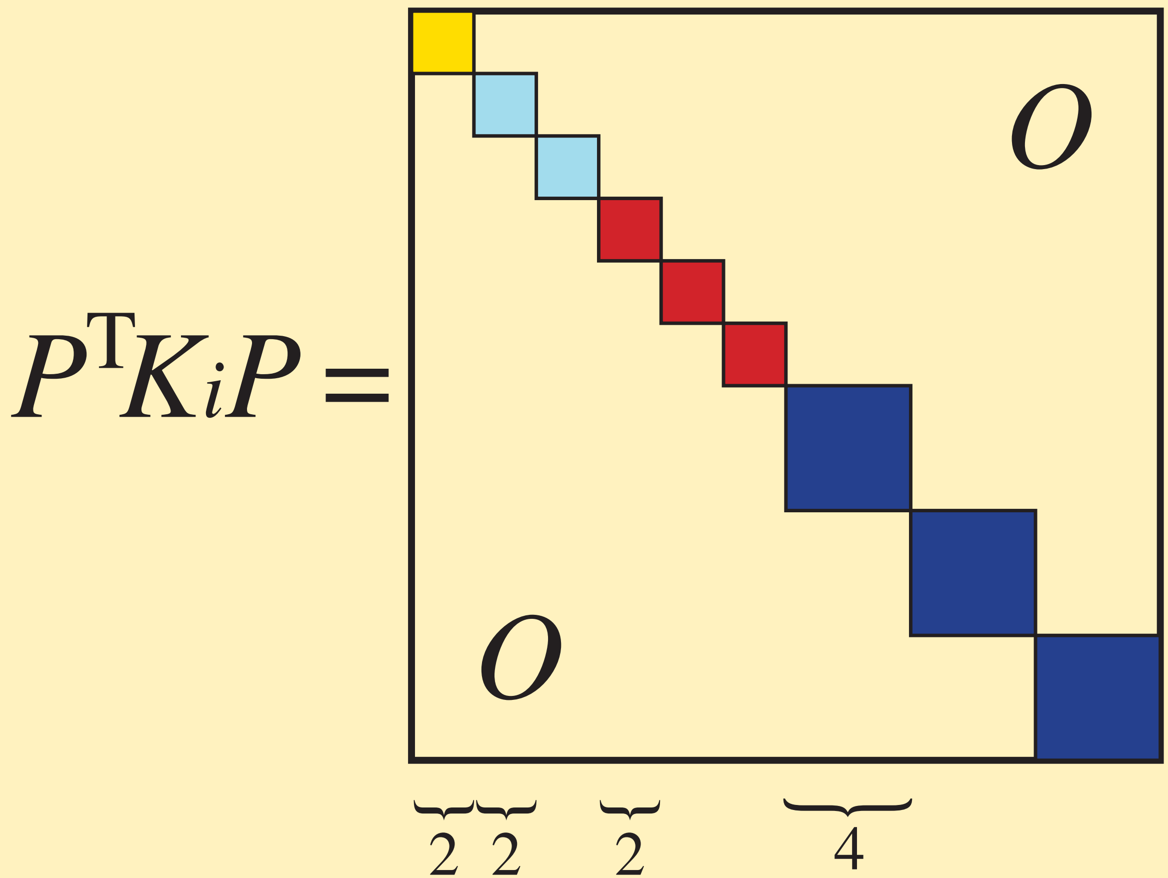




ex.) cubic truss (tetrahedral symm.)

- $\mathrm{T_d}$-symmetry (tetrahedral)
- variable linking
  4 kinds of members
  $K(\boldsymbol{x}) = x_1 K_1 + \cdots + x_4 K_4$
- 24 DOF



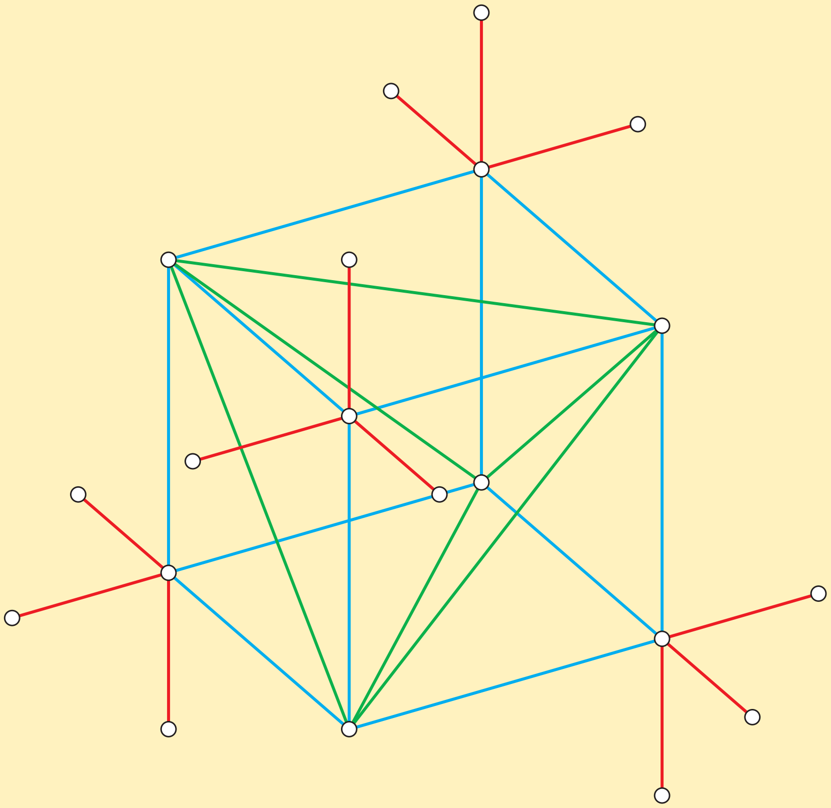


- remove 4 members
- variable linking

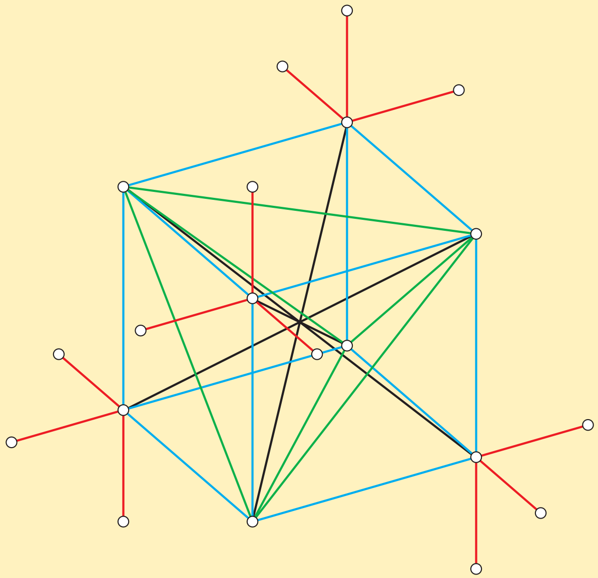
  3 kinds of members

  $K(\boldsymbol{x}) = x_1 K_1 + \cdots + x_3 K_3$

- 24 DOF



cf. original:



ex.) cubic truss (sparsity)

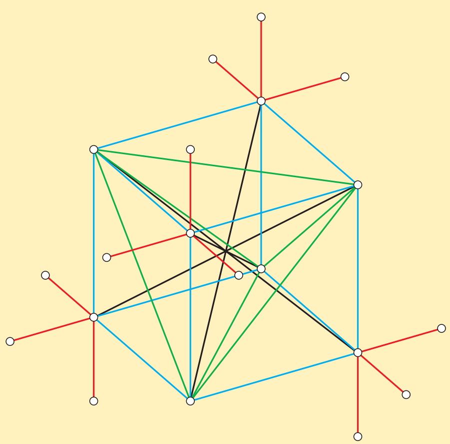
- remove 4 members
- variable linking

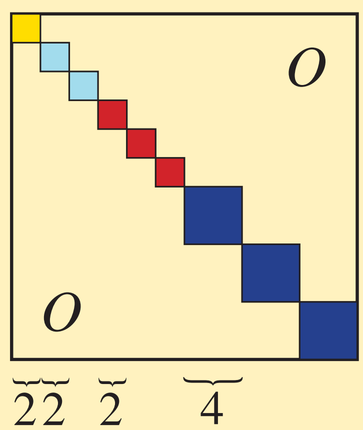
  3 kinds of members

  $K(\boldsymbol{x}) = x_1 K_1 + \cdots + x_3 K_3$

- 24 DOF

cf. original:





block-diagonal decomposition

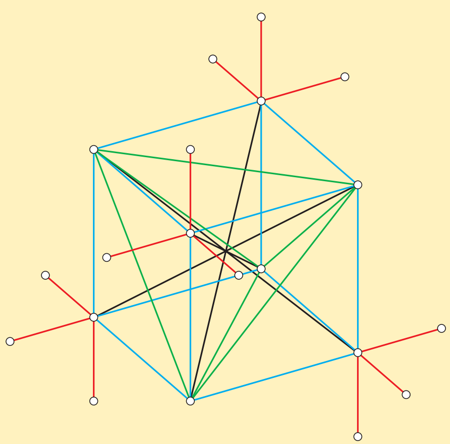
$=$ geometric symmetry

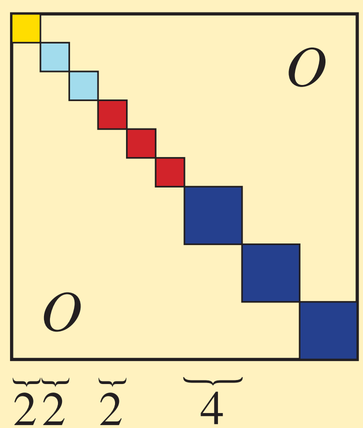
$+$ sparsity of matrices

ex.) cubic truss (sparsity)

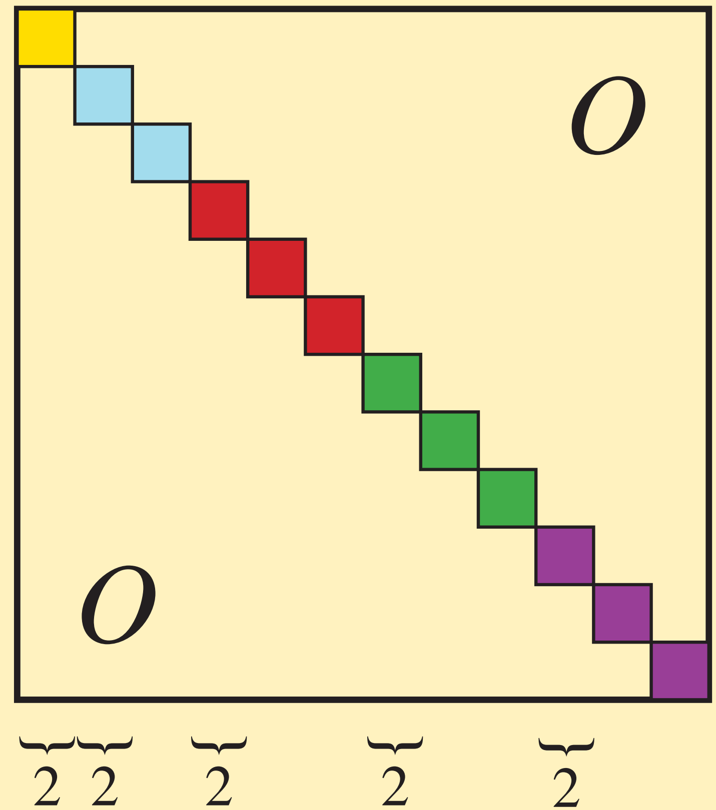
- remove 4 members
- variable linking
  3 kinds of members
  $K(\boldsymbol{x}) = x_1 K_1 + \cdots + x_3 K_3$
- 24 DOF

cf. original:





$P^{\mathrm{T}} K_i P =$



sparsity gives finer decomposition

conclusions

- simultaneous block-diagonalization
  - input: $A_1, A_2, \ldots, A_m$: symmetric matrices
  - output: block-diagonal decomposition (common block-structure)

  - validity: matrix $*$-algebra
  - application: member stiffness matrices of structure with symmetric configuration
- algorithm
  - uses only linear algebraic computation
  - does not use any knowledge of symmetry property
  - is free from

    group representation theory / matrix $*$-algebra