D'Alembert-Lagrange’s Principal Equations, Their Origin and Applications

K. C. Park, Carlos A. Felippa
Center for Aerospace Structures
University of Colorado at Boulder
Boulder, CO 80309

Roger Ohayon
Structural Mechanics and Coupled Systems Laboratory
Conservatoire National des Arts et Metiers (CNAM)
Paris, France.

USNCCM9
San Francisco, 23-26 July 2007
Principal Figures for Today’s Talk

D’Alembert’s Principle was reported to Académie des Sciences in 1742.

Lagrange came to Paris from Turin in 1787 and published his book in 1788.

Traité de Dynamique (1758)  Mécanique Analytique (1788)
Developments in the Formulation of Dynamical Systems

- It began with a single, free and rigid body with a point mass;
- Then, a rigid body with the rotational inertia properties;
- A rigid-link with joints;
- Lumped mass-spring models
- Continuum (flexible) models
- FEM/Rigid models with joints and constraints;
- And, the race for complex models is on!
Theme of Today’s Talk

As a result of the race for ever more complex modeling and analysis, often physical insight with simplicity in modeling is lost or abandoned on the wayside? And, we are left in the thick flood of numbers and gigabit analysis data.

“One always returns to where one has started.”
--Denise Levertov(1923-1997)

Hence, in dynamics, we may return to the simplicity and fundamental form of d’Alembert and Lagrange’s Principal Equations.
What are D’Alembert-Lagrange Principal Equations?

**Definition:**
For a N-degree of freedom system, regardless whether it is rigid or flexible, d’Alembert-Lagrange’s principal equations are obtained by summing all the forces and all the moments (with respect to a point) in the system.

**Symbolic Expressions:**

**Sum of forces (3 equations at most):**
\[ \Sigma (f_i - m_i a_i) = 0 \]

**Sum of moments (3 equations at most):**
\[ \Sigma \{M_i + r_i \times (f_i - m_i a_i)\} = 0 \]
Semantic Questions related to the d’Alembert-Lagrange Principal Equations:

Is a floating structure in an equilibrium condition?

Or

Is a free-free substructure, partitioned from an assembled system, in its equilibrium state?

The answer is not in the blowing wind, but in the mathematical expressions of the d’Alembert-Lagrange Principal Equations and their physical meaning.
A bottom-up approach to
D’Alembert-Lagrange’s Principal Equations

***Warning: This model does not capture the boomerang motions!
Equations of Motion for 6-DOF Boomerang Model

\[
\begin{bmatrix}
  m & 0 \\
  0 & m
\end{bmatrix}
\begin{bmatrix}
  \ddot{u} \\
  \ddot{v}
\end{bmatrix}
+ \begin{bmatrix}
  k_x & 0 \\
  0 & k_y
\end{bmatrix}
\begin{bmatrix}
  u \\
  v
\end{bmatrix}
= \begin{bmatrix}
  f_x \\
  f_y
\end{bmatrix}
\]

\[
m = \begin{bmatrix}
  m_1 & 0 & 0 \\
  0 & m_2 & 0 \\
  0 & 0 & m_3
\end{bmatrix}, \quad k_x = \begin{bmatrix}
  k_1 & -k_1 & 0 \\
  -k_1 & k_1 & 0 \\
  0 & 0 & 0
\end{bmatrix}, \quad k_y = \begin{bmatrix}
  0 & 0 & 0 \\
  0 & k_2 & -k_2 \\
  0 & -k_2 & k_2
\end{bmatrix}
\]

\[
u = \begin{bmatrix}
  u_1 \\
  u_2 \\
  u_3
\end{bmatrix}, \quad v = \begin{bmatrix}
  v_1 \\
  v_2 \\
  v_3
\end{bmatrix}, \quad f_x = \begin{bmatrix}
  f_{x_1} \\
  f_{x_2} \\
  f_{x_3}
\end{bmatrix}, \quad f_y = \begin{bmatrix}
  f_{y_1} \\
  f_{y_2} \\
  f_{y_3}
\end{bmatrix}.
\]
Definition of d’Alembert’s Forces

\[
\mathbf{f}^D = \begin{cases} 
\begin{align*}
\mathbf{f}_x^D &= f_x - m\ddot{u} - k_x u \\
\mathbf{f}_y^D &= f_y - m\ddot{v} - k_y v
\end{align*}
\end{cases}
\]

Note that the inertia forces are associated with the negative sign \((-\) which means: they are resisting forces in equilibrium equations.
Sum of x-direction forces:

\[ S_x^T f_x^D = f_x^{ext} - M \ddot{u}_c = 0 \]

\[ \bar{S}_x^T = [1 \ 1 \ 1], \]

\[ f_x^{ext} = (f_{x1} + f_{x2} + f_{x3}), \]

\[ \ddot{u}_c = (m_1 \ddot{u}_1 + m_2 \ddot{u}_2 + m_3 \ddot{u}_3)/M \]

\[ M = (m_1 + m_2 + m_3) \]

Note that \( u_c \) changes with time in general for flexible structures.
Sum of y-direction forces

\[ S_y^T f_y^D = f_y^{ext} - M \ddot{v}_c = 0 \]

\[ S_y^T = [1 \ 1 \ 1], \]

\[ f_y^{ext} = (f_{y1} + f_{y2} + f_{y3}) \]

\[ \ddot{v}_c = (m_1 \ddot{v}_1 + m_2 \ddot{v}_2 + m_3 \ddot{v}_3)/M \]

\[ M = (m_1 + m_2 + m_3) \]
Sum of moments around \((x_1, y_1)\)

\[
M_c^D = S^T_\theta \left\{ \begin{array}{c} f^D_x \\ f^D_y \end{array} \right\} = M^{ext} - J\ddot{\theta} = 0,
\]

\[
S^T_\theta = [-y_{1c} - y_{2c} - y_{3c} x_{1c} x_{2c} x_{3c}], \quad x_{ic} = x_i - x_c, \quad \text{etc.}
\]

\[
M^{ext} = S^T_\theta \left\{ \begin{array}{c} f^{ext}_x \\ f^{ext}_y \end{array} \right\}, \quad J = \sum_{i=1}^{3} m_i(x_{ic}^2 + y_{ic}^2)
\]

\[\left\{ \begin{array}{c} \ddot{u}_i \\ \ddot{v}_i \end{array} \right\} = \left\{ \begin{array}{c} \ddot{u}_c \\ \ddot{v}_c \end{array} \right\} + \left\{ \begin{array}{c} -y_{ic} \\ x_{ic} \end{array} \right\} \ddot{\theta}, \quad \sum_{i=1}^{3} m_i y_{ic} = 0, \quad \sum_{i=1}^{3} m_i x_{ic} = 0.
\]

Come on, K. C.! Every undergraduate knows this stuff . . .
And don’t waste my time here in a nice place, please.
A top-down generalization via Operational Form of Previous Equations
(We do not teach undergraduates this way!)

Sum of x-directional forces:

\[
S_x^T f^D = 0,
\]

\[
S_x = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, 
\quad f^D = \begin{bmatrix} f_{x_1}^D \\ f_{y_1}^D \\ f_{x_2}^D \\ f_{y_2}^D \\ f_{x_3}^D \\ f_{y_3}^D \end{bmatrix}
\]

Sum of y-directional forces:

\[
S_y^T f^D = 0,
\]

\[
S_y^T = [0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1]
\]

Sum of moment at \( x_1 \):

\[
S_{\theta_z} f^D = 0,
\]

\[
S_{\theta_z}^T = [-y_1 \quad x_1 \quad -y_2 \quad x_2 \quad -y_3 \quad x_3]
\]
D’Alembert-Lagrange Principal Equations for 6-DOF Boomerang Model

\[ S^T \ f^d = 0 \]

\[ S = [ S_x \ S_y \ S_{\theta_z} ] = \begin{bmatrix} 1 & 0 & -y_1 \\ 0 & 1 & x_1 \\ 1 & 0 & -y_2 \\ 0 & 1 & x_2 \\ 1 & 0 & -y_3 \\ 0 & 1 & x_3 \end{bmatrix} \]

What are they?  
Where does one utilize them for?  
What added value do they offer, if any?
What are they?

The transposition of the summation operator below consists of three rigid-body modes.

$$S = [S_x \ S_y \ S_{\theta_z}] = \begin{bmatrix}
1 & 0 & -y_1 \\
0 & 1 & x_1 \\
1 & 0 & -y_2 \\
0 & 1 & x_2 \\
1 & 0 & -y_3 \\
0 & 1 & x_3
\end{bmatrix}$$

Mathematically, $S^T f^D = 0$ is a projection of d’Alembert’s forces onto the rigid-body modes.
The d’Alembert-Lagrange principal equations,

\[ S^T f^D = 0, \]

represent the mean motions of flexible dynamical systems for which the instantaneous mass center of the total system is given, for the example problem, by

\[ u_c = \frac{(m_1 u_1 + m_2 u_2 + m_3 u_3)}{M}, \quad M = (m_1 + m_2 + m_3) \]

so that the corresponding \( x \)-direction equation is given by

\[ S^T_x f^D_x = f_x - M \frac{d^2}{dt^2} u_c = 0 \]
Physically, the three D’Alembert-Lagrange Principal Equations given by

\[ S^T f^D = 0 \]

are self-equilibrium equations, and 
\( S^T \) is the self-equilibrium operator whose transposition, \( S \), in turn consists of the rigid-body modes, \( R \), of the system.

\[ R = S \]
Variational Representation of D’Alembert-Lagrange Principal Equations

Introduce the self-equilibrium displacement

\[ \delta u_r = S \delta \alpha \]

so that one obtains the following variational statement:

\[ \delta u_r^T f^D = \delta \alpha^T S^T f^D = 0 \]

which clearly reveals that \( S^T \), in fact, is the self-equilibrium operator of the system.
What are they in partitioned mechanical systems?

Localization of Classical Lagrange Multipliers

Assembled

Classical Partitioning

Localized Partitioning

Localization is achieved by introducing a frame node, $f$
What are they in partitioned systems?
What are they in partitioned systems? - Concluded.

For the partitioned modeling of flexible mechanical systems, $S^T f^D = 0$ provides self-equilibrium condition for each partition.

Self-equilibrium equations for the two partitioned systems:

For partition 1:

$$S_1^T (f_1 - M_1 \ddot{u}_1 - K_1 u_1 - B_1 \lambda_1) = 0$$

For partition 2:

$$S_2^T (f_2 - M_2 \ddot{u}_2 - K_2 u_2 - B_2 \lambda_2) = 0$$

(a) Assembled structure  

(b) Partition into two substructures
Let’s examine the self-equilibrium equation further, that is, assume we can seek the solution of $u_1$ and $u_2$ in the form

$$u_1 = \Phi_1 q_1 + S_1 \alpha_1, \quad \text{with} \quad \Phi_1^T M_1 S_1 = 0, \quad S_1^T K_1 = 0$$

$$u_2 = \Phi_2 q_2 + S_2 \alpha_2, \quad \text{with} \quad \Phi_2^T M_2 S_2 = 0, \quad S_2^T K_2 = 0$$

Substituting the above expressions into the two self-equilibrium equations yields

For partition 1:  \[ S_1^T (f_1 - M_1 (\Phi_1 \ddot{q}_1 + S_1 \ddot{\alpha}_1) - K_1 (\Phi_1 q_1 + S_1 \alpha_1) - B_1 \lambda_1) = 0 \]

\[ \downarrow \]

\[ S_1^T B_1 \lambda_1 = S_1^T f_1 - M_{\alpha 1} \ddot{\alpha}_1, \quad M_{\alpha 1} = S_1^T M_1 S_1 \]

For partition 2:  \[ S_2^T (f_2 - M_2 (\Phi_2 \ddot{q}_2 + S_2 \ddot{\alpha}_2) - K_2 (\Phi_2 q_2 + S_2 \alpha_2) - B_2 \lambda_2) = 0 \]

\[ \downarrow \]

\[ S_2^T B_2 \lambda_2 = S_2^T f_2 - M_{\alpha 2} \ddot{\alpha}_2, \quad M_{\alpha 2} = S_2^T M_2 S_2 \]

The above two boxed equations are called the d’Alembert-Lagrange principal equations for two partitioned systems.
Examples for which self-equilibrium conditions are important:
Where does one utilize them for?

1. If the applied forces are known, they provide the mean motions of the system.

2. If part or all of the forcing functions are unknown and the mean motions are measured, they provide a least-squares solution of the applied forces and moments. For the example problem, they can provide a least-squares estimate of aerodynamic forces acting on the boomerang.

3. From the theoretical point of view, the d’Alembert-Lagrange principal equations provide the solvability conditions for completely free or partially constrained flexible systems, either quasi-static or dynamic. We will examine this aspect later in the talk.
What added value do they offer, if any?

1. In the modeling of multi-physics problems, they provide the principal (rigid-body modes) interface forces and moments, viz., average interface forces and moments.

2. In multi-body dynamics, if properly formulated and implemented, they provide the fundamental rigid-body motions and the corresponding joint forces that can aid first-hand physical insight for subsequent optimization, control and baseline solutions for detailed analysis.

3. They facilitate the divide-and-conquer paradigm for the modeling and solution of complex systems: a key property for partitioned modeling and analysis.

4. In the iterative solution of large-scale problems, they provide a crucial starting vector and subsequent filtering of residuals for faster iterative convergence.
Hey, K. C.! They are too abstract. Tell us their usage in plain English or by specific examples

All right, then we must examine the three terms in the d’Alembert-Lagrange principal equation:

\[ M_\alpha \ddot{\alpha} = S^T f - S^T B \lambda, \quad M_\alpha = S^T MS \]

1. \( M_\alpha \) provides the total mass properties of each subsystem, including the rotatory inertia and the mass center from the finite element consistent mass matrix.

2. **When the forcing function \( f \) is known, either for the quasistatic or dynamic problems, a least-squares solution of the interface forces (and moments) \( \lambda_\alpha \) provides the forces and moments acting on each subsystem by treating each as if it were a rigid-body system:**

\[ \lambda_\alpha = B^T S [S^T B B^T S]^{-1} (S^T f - M_\alpha \ddot{\alpha}) \]

Additional usages are to follow later in the talk.
D’Alembert-Lagrange Principal Equations for General 3-D Systems  *(Let’s get serious on their usage)*

Step 1: Variational Statement of the d’Alambert-Lagrange Principal Equations:

\[
S^T f^D = 0 \quad \leftrightarrow \quad \delta \alpha^T S^T f^D = \delta u_r^T f^D = 0
\]

\[
\delta u_r = S \delta \alpha
\]

where \(\delta u_r\) is clearly the virtual displacement.

*When the force and moment summation operator, \(S^T\), for a floating flexible body is transposed, it becomes the rigid-body modes designated as \(R\) in the present paper.*

The preceding statement is now mathematically expressed as

\[
R \leftrightarrow S \\
\downarrow \\
\delta u_r = R \delta \alpha
\]
Let’s get serious on their usage - cont’d

The preceding variational observation allows us to decompose the displacement as

\[ u = u_r + u_d, \quad u_d = \Phi q \]
\[ u_r = R\alpha, \quad R = S \]

where \( R \) and \( \Phi \) are the rigid body and deformational modes, respectively, and \( \alpha, q \) the corresponding amplitudes.

**Step 2: The summation operator for 3-dimensional problem:**

For each discrete nodal point we have

\[
S^T = \begin{bmatrix}
S_{trans} & S_{rot}
\end{bmatrix}^T = \begin{bmatrix}
S_1^T & S_2^T & \cdots & S_n^T
\end{bmatrix}
\]

\[
S_i^T = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & -(z_i - z_0) & (y_i - y_0) & 1 & 0 & 0 \\
(z_i - z_0) & 0 & -(x_i - x_0) & 0 & 1 & 0 \\
-(y_i - y_0) & (x_i - x_0) & 0 & 0 & 0 & 1
\end{bmatrix}
\]
Variational formulation of partitioned system

\[ \delta \Pi(u_g) = \delta u_g^T \cdot f_g^D \]

\[ f_g^D = f_g - K_g u_g - M_g \ddot{u}_g \]

\[ K_g = L^T K L, \quad M_g = L^T M L \]

where \( u_g \) is the discrete global displacement vector, \( f_g^D \) is d’Alembert’s force, \( K_g \) and \( M_g \) are the assembled global stiffness and mass matrices, \( K \) and \( M \) are the block diagonal substructure-by-substructure stiffness and mass matrices, \( L \) is the assembly Boolean matrix, and the superscript dot (\( ^{\cdot} \)) denotes time differentiation, respectively.

\[
\begin{pmatrix}
  u_1 \\
  u_2 \\
  \vdots \\
  u_n
\end{pmatrix}
= \begin{bmatrix}
  L_1 \\
  L_2 \\
  \vdots \\
  L_n
\end{bmatrix}
\begin{bmatrix}
  u_g
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
  u
\end{bmatrix} = L \begin{bmatrix}
  u_g
\end{bmatrix}
\]
Example of Assembly Boolean Matrix

\[ u_1 = L_1 u_g, \quad L_1 = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\end{bmatrix} \]

(a) Assembled structure

(b) Partition into two substructures
Partition Interface constraint functional

\[ B^T (u - L u_g) = 0 \quad \Rightarrow \quad B^T u - L_f u_f = 0, \quad L_f = B^T L \]

\[ \delta \pi_\ell = \lambda^T_\ell \left( B^T u - L_f u_f \right) \]

\[
B = \begin{bmatrix}
B_1 & 0 & \ldots & 0 \\
0 & B_2 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & B_n \\
\end{bmatrix}
\]

where \( u_f \) consists of those global degrees freedom pertaining to the partition interfaces. And \( B \) extracts the interface degrees of freedom for each partition.
Variational Statement for Partitioned Systems

\[
\delta \Pi(u, u_f, \lambda_\ell) = \delta u^T \cdot f^D \\
+ \delta \{ \lambda_\ell^T (B^T u - L_f u_f) \}
\]

\[f^D = f - Ku - M \ddot{u}\]

Introduce the decomposition of the displacement into deformation and rigid-body (D’Alembert-Lagrange principal) modes:

\[u = u_r + u_d, \quad u_d = \Phi q\]
\[u_r = R\alpha, \quad R = S\]
Four-Variable Variational Formulation for Partitioned Systems

\[
\delta \Pi(u_\alpha, u_d, u_f, \lambda_\ell) = \delta (u_\alpha + u_d)^T \cdot f^D(u_\alpha + u_d) \\
+ \delta [\lambda^T_\ell (B^T (u_\alpha + u_d) - L_f u_f)] \\
f^D(u_\alpha + u_d) = f - K (u_\alpha + u_d) - M (\ddot{u}_\alpha + \ddot{u}_d)
\]

\[
\begin{bmatrix}
\bar{K}_d & 0 & \Phi^T B & 0 \\
0 & \bar{K}_\alpha & R_b^T & 0 \\
B^T \Phi & R_b & 0 & -L_f \\
0 & 0 & -L_f^T & 0
\end{bmatrix}
\begin{bmatrix}
q \\
\alpha \\
\lambda_\ell \\
u_f
\end{bmatrix} =
\begin{bmatrix}
\Phi^T f \\
R^T f \\
0 \\
0
\end{bmatrix}
\]

\[
\bar{K}_d = \Phi^T M \Phi D^2 + \Phi^T K \Phi, \quad D = \frac{d}{dt}
\]

\[
\bar{K}_\alpha = R^T M R \ D^2
\]

\[
R_b = B^T R
\]
D’Alembert-Lagrange Principal Equations for Partitioned Systems

\[ M_\alpha \frac{d^2 \alpha}{dt^2} = R^T (f - B \lambda) \]

\[ L_f^T \lambda_\ell = 0 \]

\[ M_\alpha = R^T M R = \begin{bmatrix} M_{tt} & M_{tr} \\ M_{tr}^T & M_{rr} \end{bmatrix}, \quad R \leftrightarrow S \]

\[ \alpha^T = \begin{bmatrix} \alpha_x & \alpha_y & \alpha_z & \alpha_{\theta_x} & \alpha_{\theta_y} & \alpha_{\theta_z} \end{bmatrix} \]

In the above equation, \( M_{tt} \) is in general a (3x3)-diagonal matrix that represents the sum of the elemental translational nodal masses for each partition, \( M_{rr} \) represents a (3x3)-rotatory inertia matrix that is usually fully coupled unless the coordinates chosen for generating \( R \) coincide with the principal axes of the partitioned substructures, and \( M_{tr} \) represents the coupling between the translational mass and the rotatory mass matrix, respectively.

This equation provides the mean gross motions of the total system!
Application 1: mass properties of complex structures
(Example - Bar element)

It is not trivial at all for computing the rotatory inertia and its coupling matrix with the translational mass properties!

Translational mass matrix of a bar element:

$$m_{bar} = m \begin{bmatrix} 1/3 & 0 & 1/6 & 0 \\ 0 & 1/3 & 0 & 1/6 \\ 1/6 & 0 & 1/3 & 0 \\ 0 & 1/6 & 0 & 1/3 \end{bmatrix}$$

Summation operator for the case when the moment is computed around node 1

$$S^T_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & -\ell/\sqrt{2} & \ell/\sqrt{2} \end{bmatrix}$$
Application 1: mass properties of complex structures - Cont’d
(Bar element example)

\[ M_{\alpha_1} = S_1^T m_{bar} S_1 = m \begin{bmatrix} 1 & 0 & -\ell/2\sqrt{2} \\ 0 & 1 & \ell/2\sqrt{2} \\ -\ell/2\sqrt{2} & \ell/2\sqrt{2} & \ell^2/3 \end{bmatrix} \]

The above matrix correctly captures the rotatory inertia when the rotational motion is measured with respect to node 1 and the corresponding coupling matrix with the Translational mass. Complex cases yield the correct mass properties. Computations of the rotatory inertia for complex systems, e.g., automobile, satellites, etc., with the present procedure is straightforward and consistent.

All one needs for computing the inertia properties of complex systems are the translational mass matrix and the moment summation operator!
Small application: Generation of Diagonal Mass Matrix

Step 1: Generate the consistent mass matrix

\[
m = \rho A\ell \begin{bmatrix}
1/3 & 0 & 0 & 1/6 & 0 & 0 \\
0 & 13/35 & 11\ell/210 & 0 & 9/70 & -13\ell/420 \\
0 & 11\ell/210 & \ell^2/105 & 0 & 13\ell/420 & -\ell^2/140 \\
1/6 & 0 & 0 & 1/3 & 0 & 0 \\
0 & 9/70 & 13\ell/420 & 0 & 13/35 & -11\ell/210 \\
0 & -13\ell/420 & -\ell^2/140 & 0 & -11\ell/210 & \ell^2/105
\end{bmatrix}
\]

Step 2: Construct the summation matrix (the rigid-body mode shapes)

\[
R^T = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & -\ell/2 & 1 & 0 & \ell/2 & 1
\end{bmatrix}
\]

Step 3: Obtain \( m_\omega = R^T m R = \rho A\ell \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & \ell^2/12
\end{bmatrix} \)

Step 4: Obtain the diagonal mass matrix \( m_{diag} \) by allocating half of the mass and rotatory inertia for the two nodes

\[
M_{diag} = \frac{\rho A\ell}{2} (\text{diag})[1, 1, \ell^2/12, 1, 1, \ell^2/12]
\]
Application 2: Solvability for unconstrained systems under quasi-static equilibrium states

Partitioned equations of motion for structures:

\[
\begin{bmatrix}
\Phi^T K \Phi & 0 & \Phi^T B & 0 \\
0 & 0 & R^T B & 0 \\
B^T \Phi & B^T R & 0 & -L_f \\
0 & 0 & -L_f^T & 0
\end{bmatrix}
\begin{bmatrix}
q \\
\alpha \\
\lambda_{\ell} \\
u_f
\end{bmatrix}
=
\begin{bmatrix}
\Phi^T f \\
R^T f \\
0 \\
0
\end{bmatrix}
\]

This equation is indefinite and consequently requires a delicate care for its solution!
Application 2: Solvability for unconstrained systems under quasi-static equilibrium states

Eliminate $q$ to obtain:

\[
\begin{bmatrix}
F_{bb} & -R_b & L_f \\
-R_b^T & 0 & 0 \\
L_f^T & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\lambda_f \\
\alpha \\
u_f
\end{bmatrix}
=
\begin{bmatrix}
b_\lambda \\
b_\alpha \\
0
\end{bmatrix}
=
\begin{bmatrix}
B^T K^+ f \\
-R^T f \\
0
\end{bmatrix}
\]

\[
F_{bb} = B^T P_\alpha K^+ P_\alpha B = B^T K^+ B
\]

\[
u_d = \Phi q = P_\alpha u
\]

\[
P_\alpha = I - R (R^T R)^{-1} R^T
\]

This equation is regular and consequently can be solved in a routine manner!
Application 3: Consistent starting vector for an iterative solution of partitioned system equations

Step 1: Solve for a least-square value of the interface forces:

\[
\begin{bmatrix}
R_b^T \\
L_f
\end{bmatrix} \lambda_\alpha = \begin{bmatrix}
\bar{b}_\alpha \\
0
\end{bmatrix}
\]

\[
\downarrow
\]

\[
\lambda_\alpha = P_L R_b (R_b^T P_L R_b)^{-1} \bar{b}_\alpha
\]

\[
P_L = I - L_f (L_f L_f^T)^{-1} L_f
\]
Application 3: Consistent starting vector for an iterative solution of partitioned system equations

Step 2: Project the new iterate to be orthogonal to the interface rigid-body modes:

\[ \lambda_{\ell} = \lambda_\alpha + P \Delta \lambda_{\ell} \]
\[ P = P_L - P_L R_b (R_b^T P_L R_b)^{-1} R_b^T P_L \]
\[ P_L = I - L_f (L_f L_f^T)^{-1} L_f \]

in order to minimize the residual:

\[ r = P (\hat{b}_\lambda - F_{bb} P \Delta \lambda_{\ell}) \]
\[ \hat{b}_\lambda = \bar{b}_\lambda - F_{bb} \lambda_\alpha \]
Future Potential Applications

A. Least-squares nominal interface forces that may provide a preliminary design modification or control strategy for systems with constraints.

B. Augmented solution of \((q, \alpha)\) for dual control strategy development, i.e., for principal motions and deformational motions in tandem.

C. Filtering of mean motion signals from output signals.

D. Advanced multi-physics modeling
Discussions

1. The d’Alembert-Lagrange principal equations consists of 6 rigid-body motions regardless how large the flexible mechanical structural systems may be, and they provide the mean motions of the overall system dynamics.

2. The rotatory inertia and its coupling terms with the translational mass properties are obtained as part of the derivational process of the d’Alembert-Lagrange principal equations presented herein.

3. The d’Alembert-Lagrange principal equations constitute the key solvability condition for systems partially constrained or in completely free-free state.

4. For an iterative solution of coupled multi-physics problems, the solution of the d’Alembert-Lagrange principal equations provides a consistent starting vector, thus accelerating the iterative process.

5. There remains a challenge to expand the usage of the d’Alembert-Lagrange principal equations, some of which have been outlined herein.
Viva la dynamique élémentaire!

Fin!
Question:
Can you readily get the necessary physical insight from $10^8$-DOF simulation results?

Wisdom from Native American Culture:
I can count five from the right fingers, and I can count another five from the left fingers. And, I don’t know how to count further!