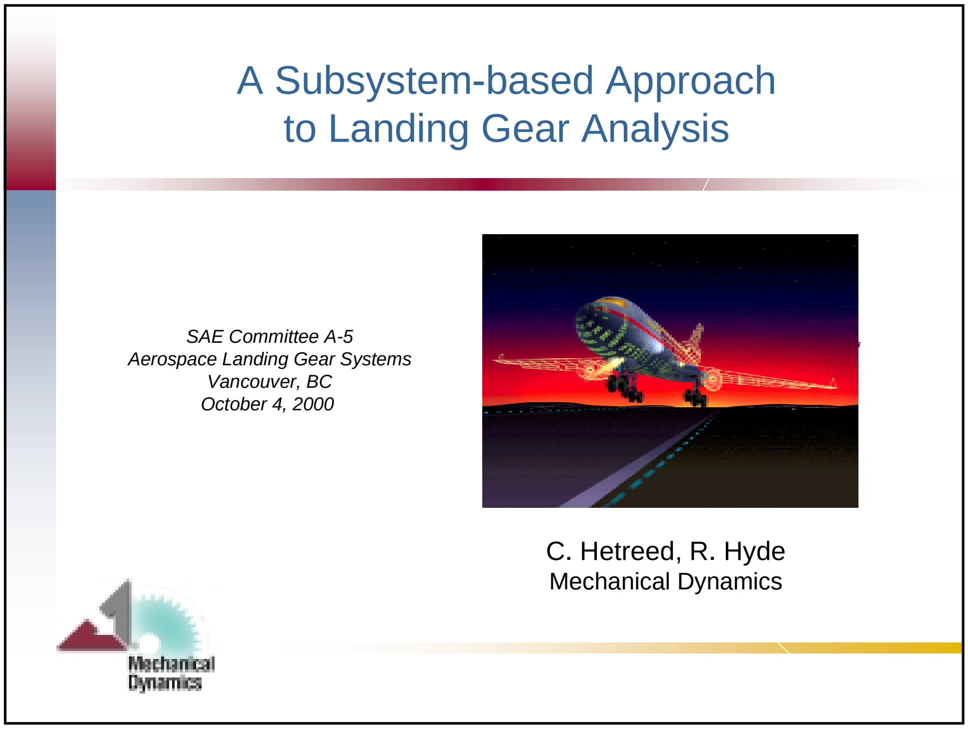
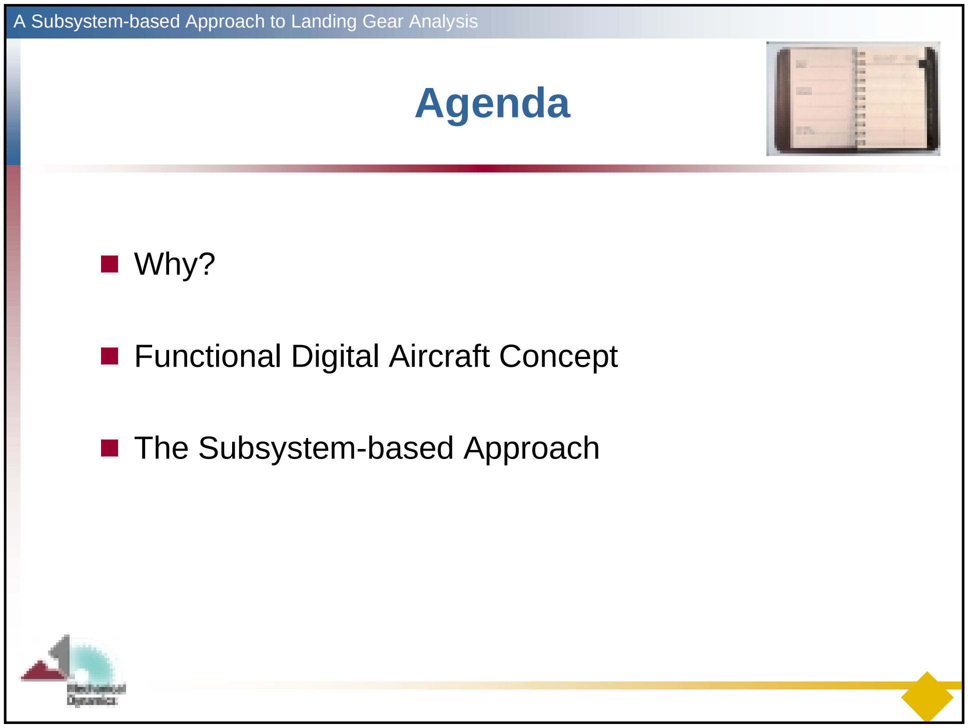


A Subsystem-based Approach to Landing Gear Analysis

SAE Committee A-5
Aerospace Landing Gear Systems
Vancouver, BC
October 4, 2000

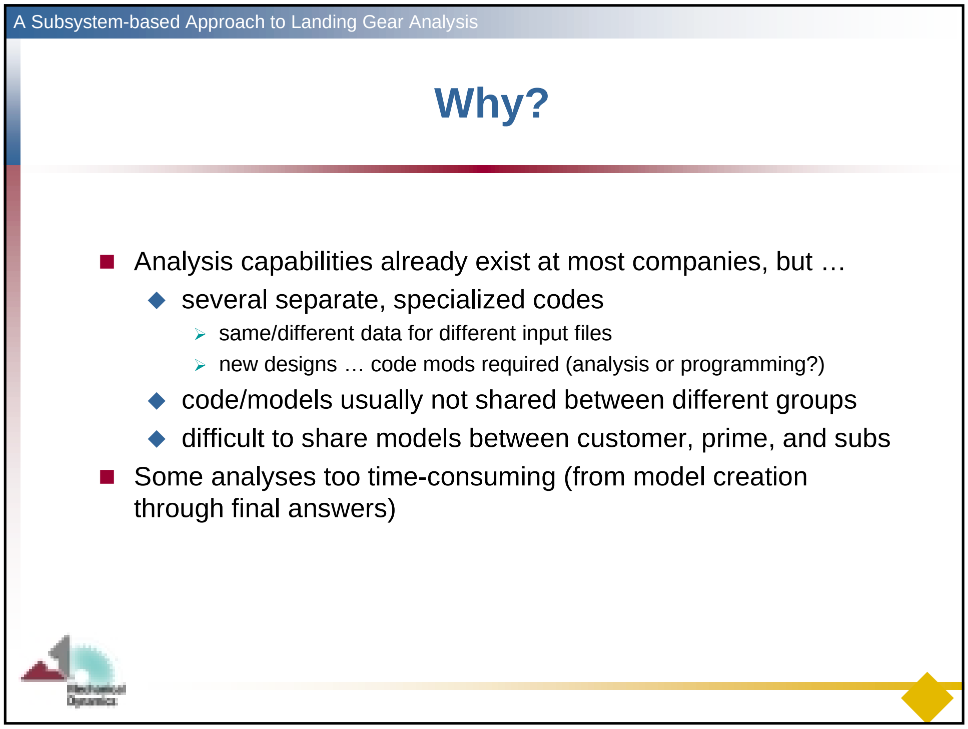
C. Hetreed, R. Hyde
Mechanical Dynamics



A Subsystem-based Approach to Landing Gear Analysis

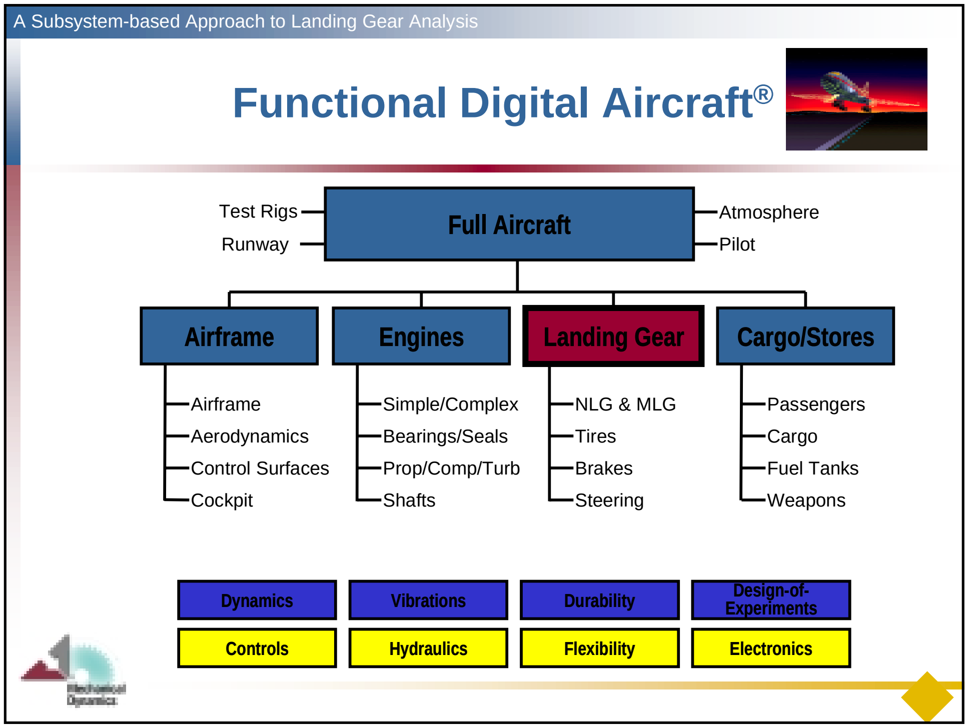
Agenda

- Why?
- Functional Digital Aircraft Concept
- The Subsystem-based Approach



Why?

- Analysis capabilities already exist at most companies, but …
  - several separate, specialized codes
    - same/different data for different input files
    - new designs … code mods required (analysis or programming?)
  - code/models usually not shared between different groups
  - difficult to share models between customer, prime, and subs
- Some analyses too time-consuming (from model creation through final answers)



Functional Digital Aircraft®

Full Aircraft

Test Rigs

Runway

Atmosphere

Pilot

Airframe
- Airframe
- Aerodynamics
- Control Surfaces
- Cockpit

Engines
- Simple/Complex
- Bearings/Seals
- Prop/Comp/Turb
- Shafts

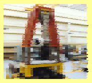
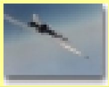
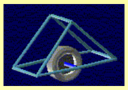
Landing Gear
- NLG & MLG
- Tires
- Brakes
- Steering

Cargo/Stores
- Passengers
- Cargo
- Fuel Tanks
- Weapons

Dynamics	Vibrations	Durability	Design-of-Experiments
Controls	Hydraulics	Flexibility	Electronics

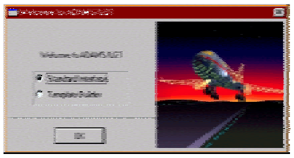
A Subsystem-based Approach to Landing Gear Analysis

Test & Analysis

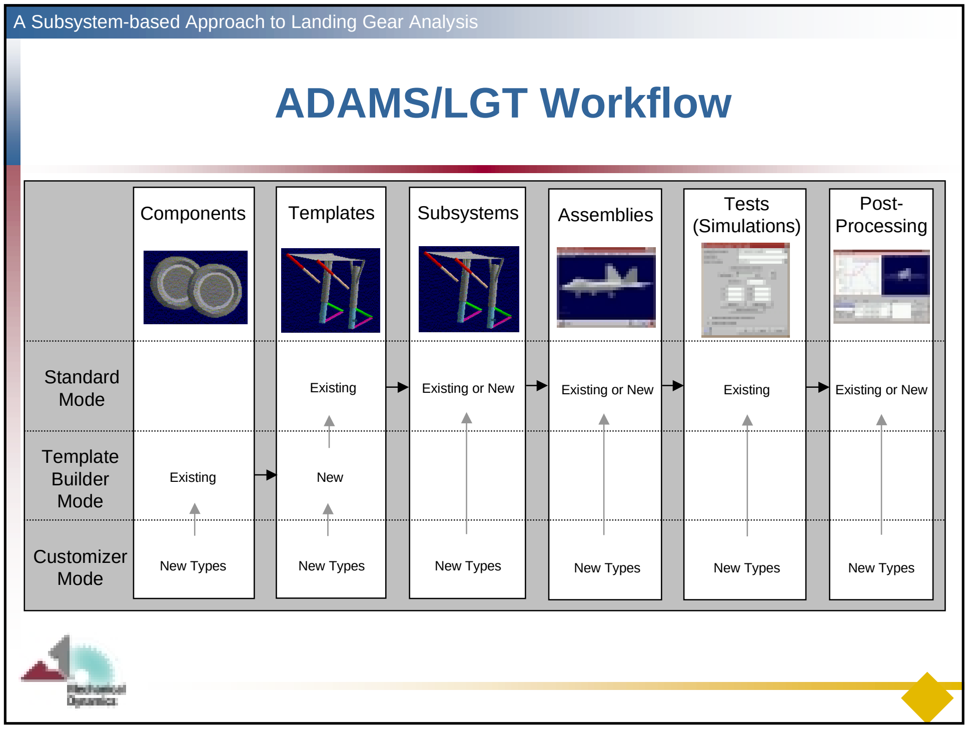
	Component Tests	Subsystem Tests	Actual Event Tests (Full Aircraft)
Actual Hardware & Software	$	$ 	$ 
Hybrid (Hardware In The Loop)	$	$	$
Complete Digital Hardware & Software	√ 	√	Functional Digital Aircraft √
Simplified Digital Hardware & Software	X	Legacy Fortran Code	Legacy Fortran Code

A Subsystem-based Approach to Landing Gear Analysis

The Subsystem-based Approach



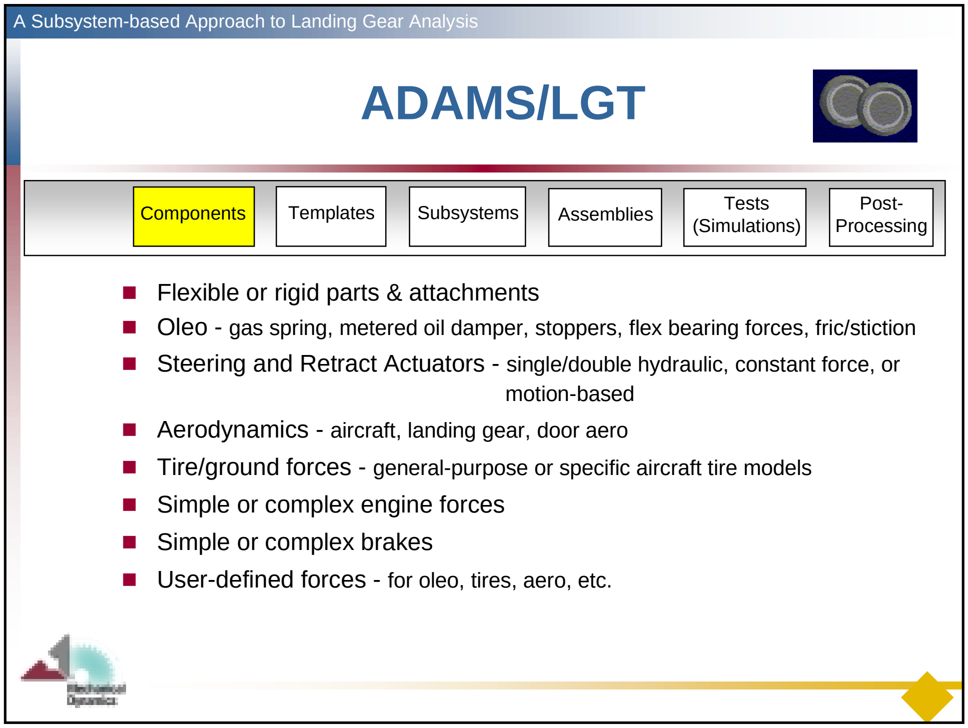
- ADAMS/LGT … customized ADAMS environment
- Allows one to:
  - quickly build subsystems from standard components
  - automatically combine subsystems into assemblies
  - run assemblies through standard, controlled tests/simulations



A Subsystem-based Approach to Landing Gear Analysis

ADAMS/LGT Workflow

	Components	Templates	Subsystems	Assemblies	Tests (Simulations)	Post-Processing
Standard Mode		Existing	Existing or New	Existing or New	Existing	Existing or New
Template Builder Mode	Existing	New				
Customizer Mode	New Types	New Types	New Types	New Types	New Types	New Types

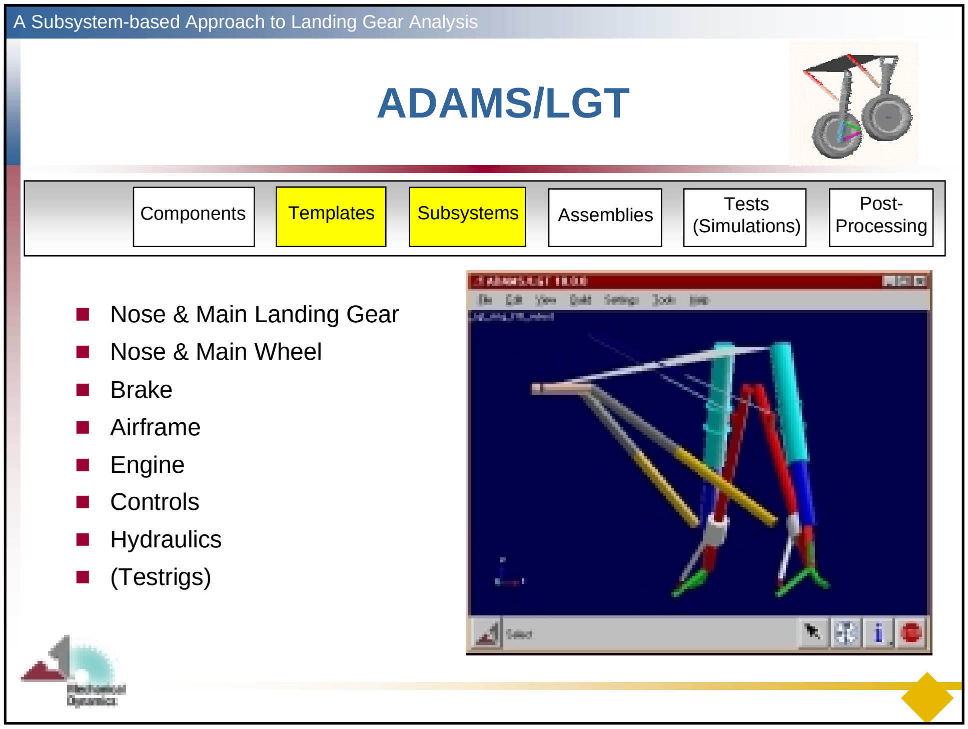


A Subsystem-based Approach to Landing Gear Analysis

ADAMS/LGT

Components	Templates	Subsystems	Assemblies	Tests (Simulations)	Post-Processing

- Flexible or rigid parts & attachments
- Oleo - gas spring, metered oil damper, stoppers, flex bearing forces, fric/stiction
- Steering and Retract Actuators - single/double hydraulic, constant force, or motion-based
- Aerodynamics - aircraft, landing gear, door aero
- Tire/ground forces - general-purpose or specific aircraft tire models
- Simple or complex engine forces
- Simple or complex brakes
- User-defined forces - for oleo, tires, aero, etc.



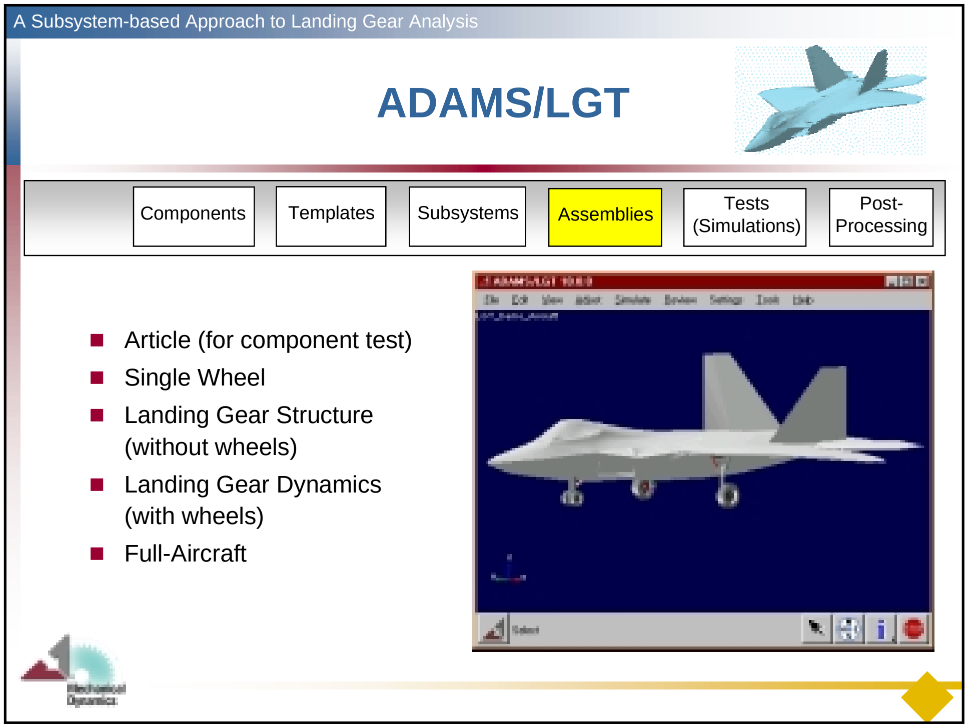
A Subsystem-based Approach to Landing Gear Analysis

ADAMS/LGT

Components | Templates | Subsystems | Assemblies | Tests (Simulations) | Post-Processing

- Nose & Main Landing Gear
- Nose & Main Wheel
- Brake
- Airframe
- Engine
- Controls
- Hydraulics
- (Testrigs)

ADAMS/LGT 10.0.0
File Edit View Build Settings Tools Help
Select
Mechanical Dynamics



A Subsystem-based Approach to Landing Gear Analysis

ADAMS/LGT

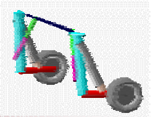
Components | Templates | Subsystems | Assemblies | Tests (Simulations) | Post-Processing

- Article (for component test)
- Single Wheel
- Landing Gear Structure (without wheels)
- Landing Gear Dynamics (with wheels)
- Full-Aircraft

ADAMS/LGT 10.0.0
File Edit View Adjust Simulate Review Settings Tools Help
Select
Mechanical Dynamics

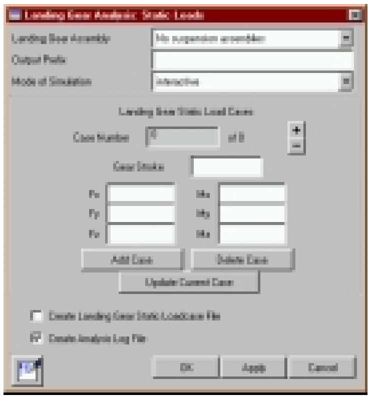
A Subsystem-based Approach to Landing Gear Analysis

ADAMS/LGT



Components | Templates | Subsystems | Assemblies | **Tests (Simulations)** | Post-Processing

- Article (for component test)
  - Article test (dyno)
- Single Wheel
  - Tire tests
- Landing Gear Structure (without wheels)
  - Steady Axle Loads
- Landing Gear Dynamics (with wheels)
  - Retract/Extend, Drop



Landing Gear Analysis: Static Loads

Landing Gear Assembly: My suspension assembly

Output Prefix:

Mode of Simulation: interactive

Landing Gear Static Load Cases

Case Number: 0 of 0

Gear Stroke:

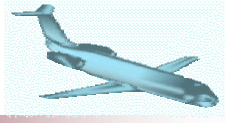
Fx: Mx:
Fy: My:
Fz: Mz:

☐ Create Landing Gear Static LoadCase File

☑ Create Analysis Log File

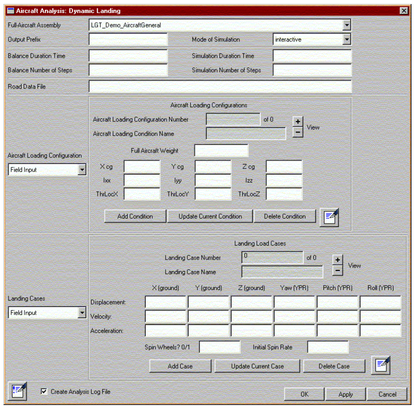
A Subsystem-based Approach to Landing Gear Analysis

ADAMS/LGT



Components | Templates | Subsystems | Assemblies | **Tests (Simulations)** | Post-Processing

- Full-Aircraft
  - Ground
    - Ground Attitude, Carrier Launch, Dynamic Tipback, Tiedown, Towing
  - Taxi
    - Dynamic Taxi, Turning, Braking, General Pilot Maneuvering
  - Landing
    - General Landing, General Pilot Maneuvering
  - In-Flight



Aircraft Analysis: Dynamic Landing

Full-Aircraft Assembly: LGT_Demo_AircraftGeneral

Output Prefix:

Mode of Simulation: interactive

Balance Duration Time:

Simulation Duration Time:

Balance Number of Steps:

Simulation Number of Steps:

Road Data File:

Aircraft Loading Configuration: Field Input

Aircraft Loading Configurations

Aircraft Loading Configuration Number: of 0

Aircraft Loading Condition Name:

Full Aircraft Weight:

X cg: Y cg: Z cg:
Ixx: Iyy: Izz:
ThrLocX: ThrLocY: ThrLocZ:

Landing Cases: Field Input

Landing Load Cases

Landing Case Number: 0 of 0

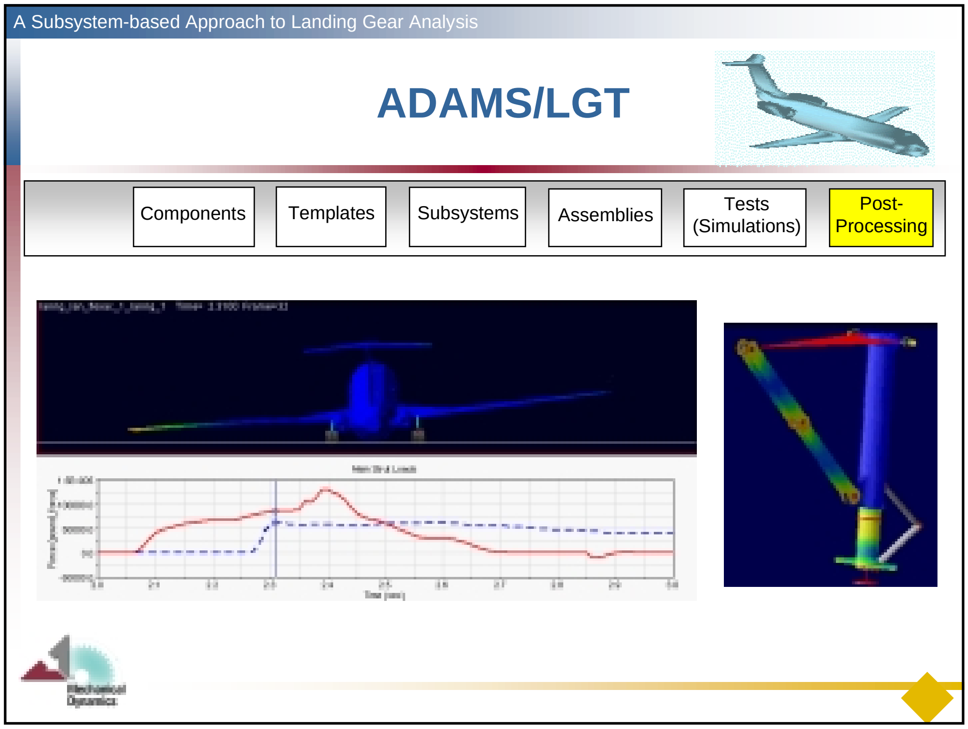
Landing Case Name:

	X (ground)	Y (ground)	Z (ground)	Yaw (YPR)	Pitch (YPR)	Roll (YPR)
Displacement:						
Velocity:						
Acceleration:						

Spin Wheels? 0/1:

Initial Spin Rate:

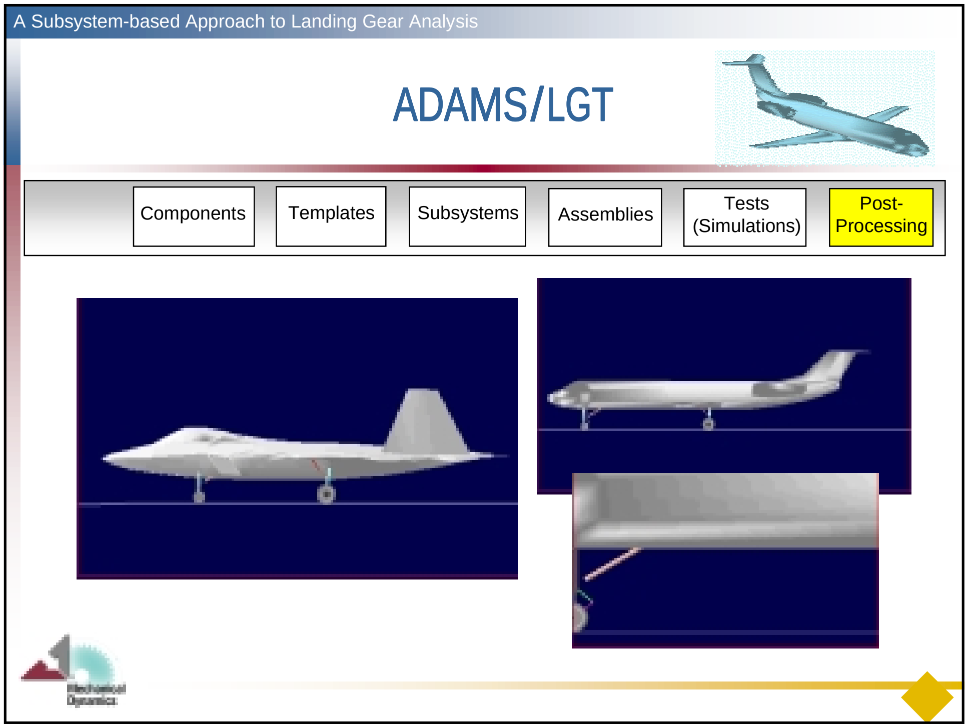
☑ Create Analysis Log File



A Subsystem-based Approach to Landing Gear Analysis

ADAMS/LGT

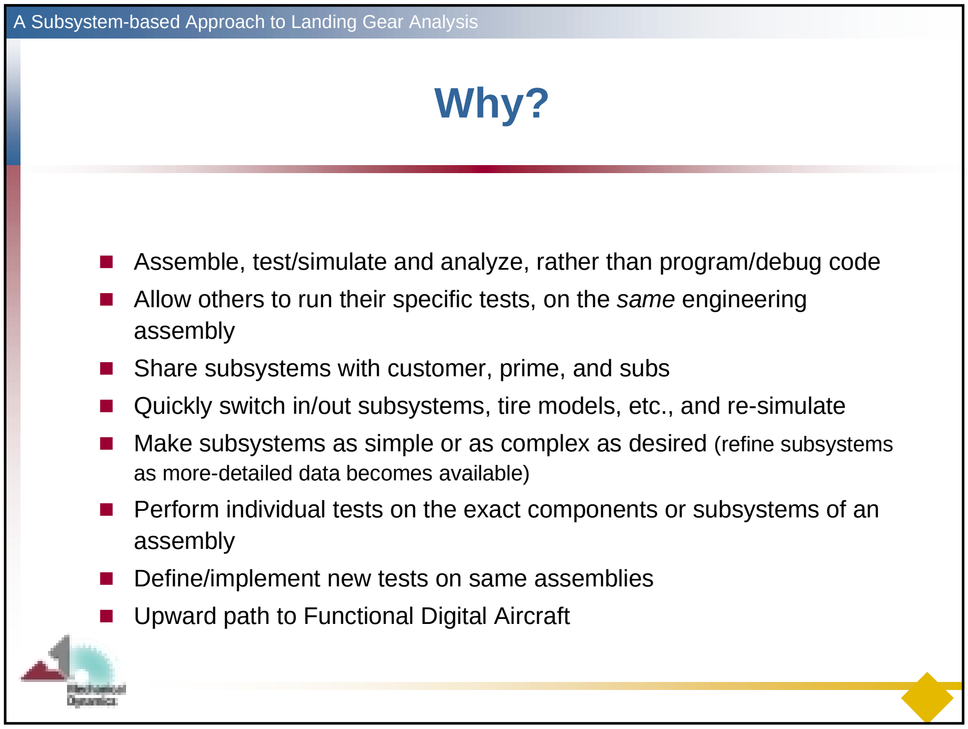
Components | Templates | Subsystems | Assemblies | Tests (Simulations) | Post-Processing



A Subsystem-based Approach to Landing Gear Analysis

ADAMS/LGT

Components | Templates | Subsystems | Assemblies | Tests (Simulations) | Post-Processing



A Subsystem-based Approach to Landing Gear Analysis

Why?

- Assemble, test/simulate and analyze, rather than program/debug code
- Allow others to run their specific tests, on the *same* engineering assembly
- Share subsystems with customer, prime, and subs
- Quickly switch in/out subsystems, tire models, etc., and re-simulate
- Make subsystems as simple or as complex as desired (refine subsystems as more-detailed data becomes available)
- Perform individual tests on the exact components or subsystems of an assembly
- Define/implement new tests on same assemblies
- Upward path to Functional Digital Aircraft