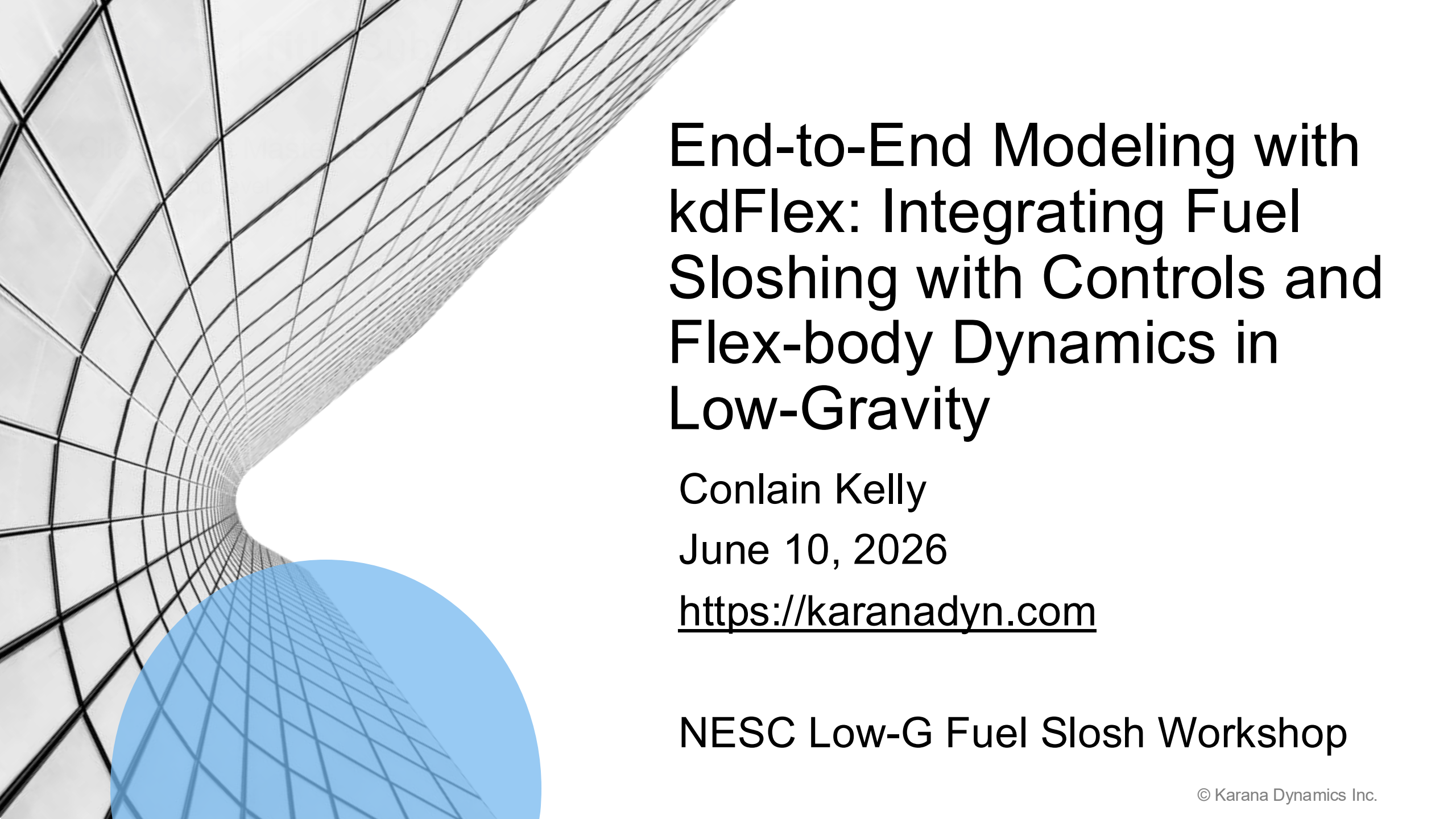




End-to-End Modeling with kdFlex: Integrating Fuel Sloshing with Controls and Flex-body Dynamics in Low-Gravity

Conlain Kelly

June 10, 2026

<https://karanadyn.com>

NESC Low-G Fuel Slosh Workshop

Motivation



- Explore interaction between flex + rigid motions, control structures, and fuel slosh
 - *Support in-the-loop coupling*
- Run what-if scenarios and Monte Carlo
- Once available, calibrate using experimental data

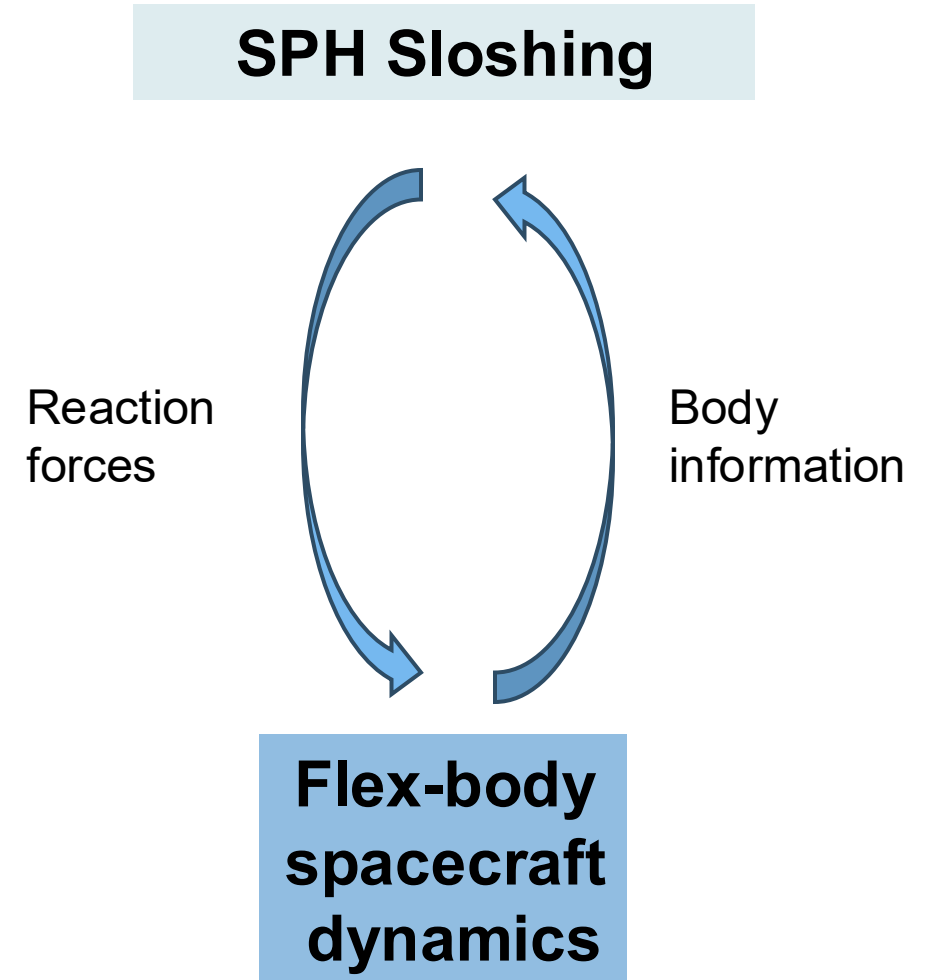
➤ This talk uses kdFlex to integrate sloshing with flex-body spacecraft



High-level overview



- Use Smoothed Particle Hydrodynamics (SPH)
 - Natural Lagrangian treatment of moving boundaries
 - Highly flexible, tunable, parallelizable
 - Slower than reduced-order pendulum models
- Interface SPH model with multibody dynamics
 - Input: body geometries, poses, velocities
 - Output: Per-body reaction forces
- Run integrated closed-loop simulations
 - SPH can always take the biggest safe timestep and always has latest pose information
 - Requires soft coupling between multibody integrator and SPH integrator





Smoothed Particle Hydrodynamics Methods

SPH quick summary

- *Meshless Lagrangian Discretization*

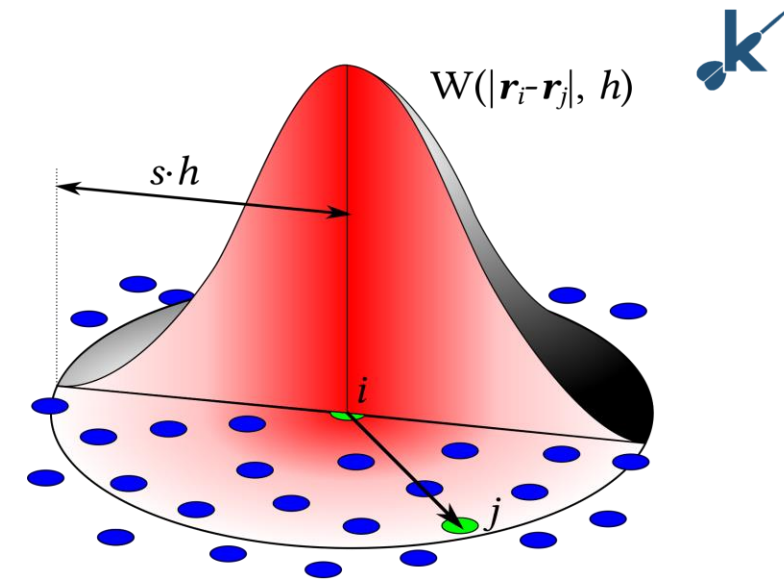
- Fluid mass is discretized into moving interpolation points (particles)
- Each particle carries discrete hydrodynamic properties (e.g., mass, velocity)

- *The Smoothing Kernel (W)*

- Field variables and their gradients are evaluated analytically over a localized support radius (h)
- No spatial grid; particles only interact with their immediate neighbors

- *Approximating Navier-Stokes*

- **Mass Conservation:** Track density variations by comparing against neighbor particles
- **Momentum Conservation:** Pressure gradients and viscous forces are transformed into pairwise particle forces

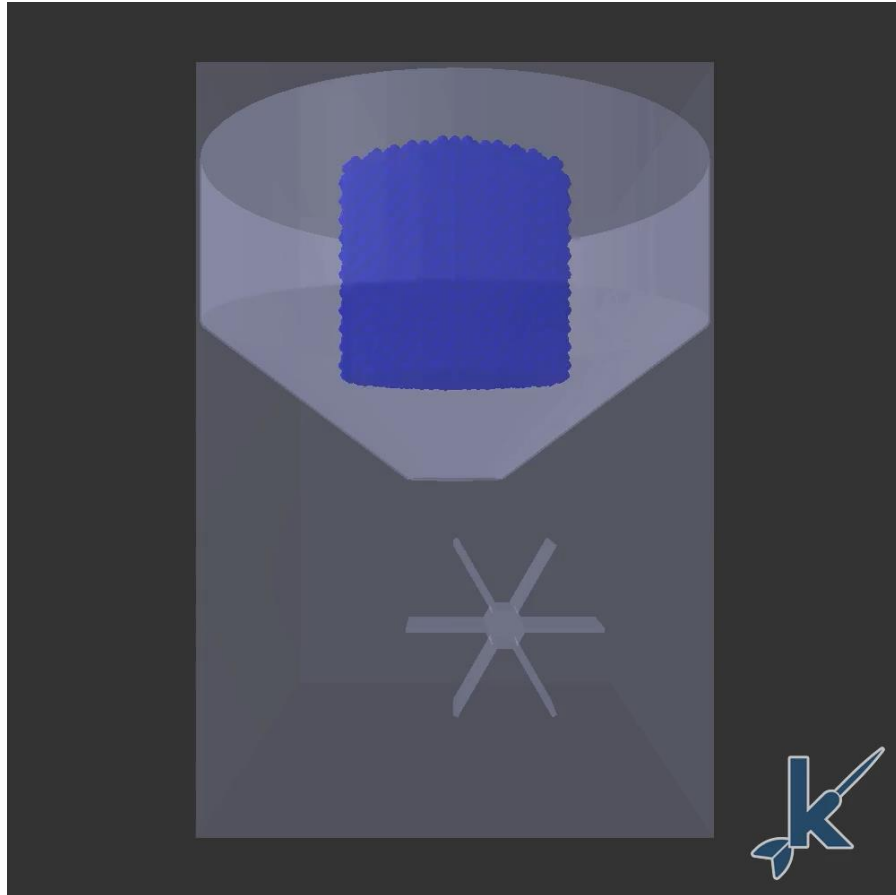


$$\rho_i = \rho(\mathbf{r}_i) = \sum_j m_j W_{ij}$$

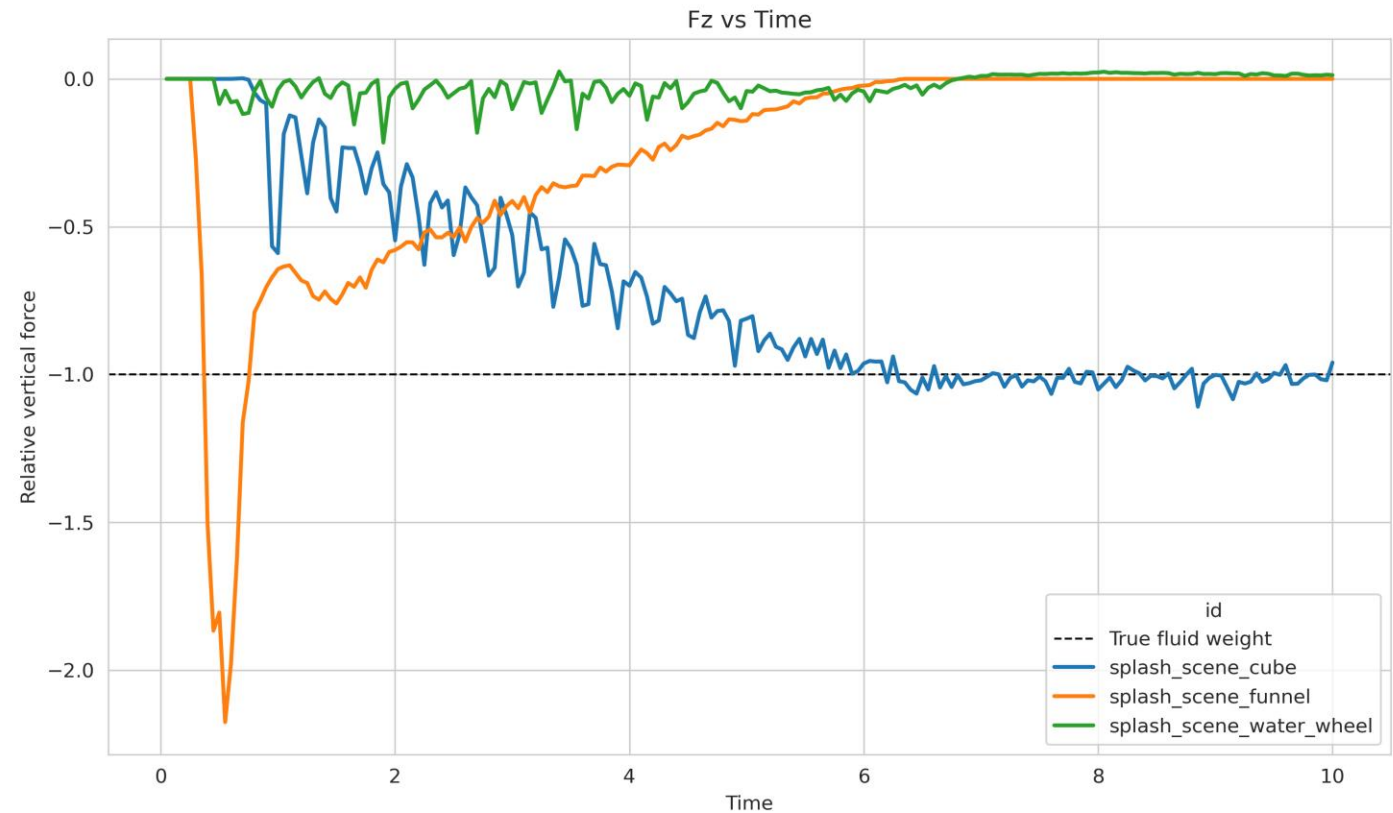
$$\frac{d\rho_i}{dt} = \sum_j m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla W_{ij}$$

Image source: Wikipedia, CC BY-SA 4.0
<https://commons.wikimedia.org/w/index.php?curid=70225405>

SPH funnel flow onto water wheel



6336 SPH particles (DFSPH)
colored by velocity



Reaction forces on each
body

Using SPH, how well can we model low-bond-# systems?

Modeling Challenges:

1. Physical realism
(e.g., continuity, momentum conservation)
2. Fluid-container coupling
3. Surface Tension & Viscosity

Our approach:

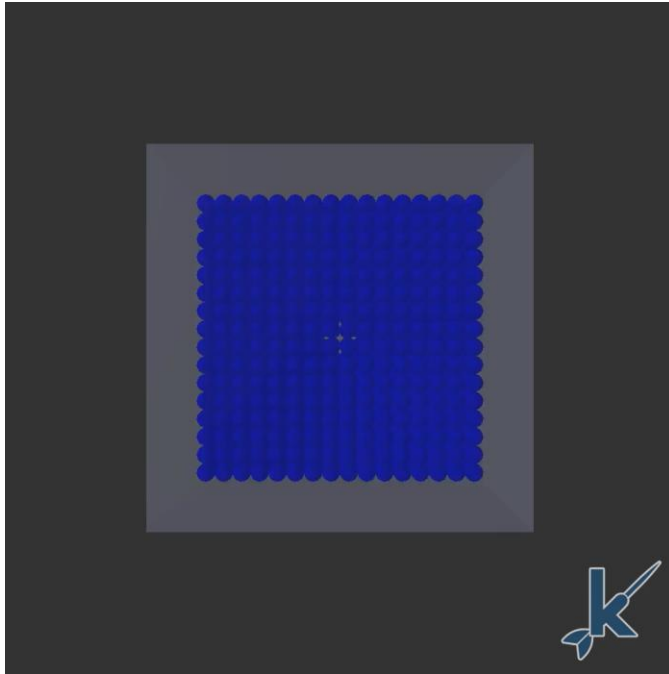
1. Divergence-free SPH (Bender 2016)
2. Multibody dynamics + boundary particles (Akinci 2012)
3. Implicit schemes
 - Zorilla (2020), Weiler (2018)

Future work: calibrate and evaluate models against data

Settling unit test

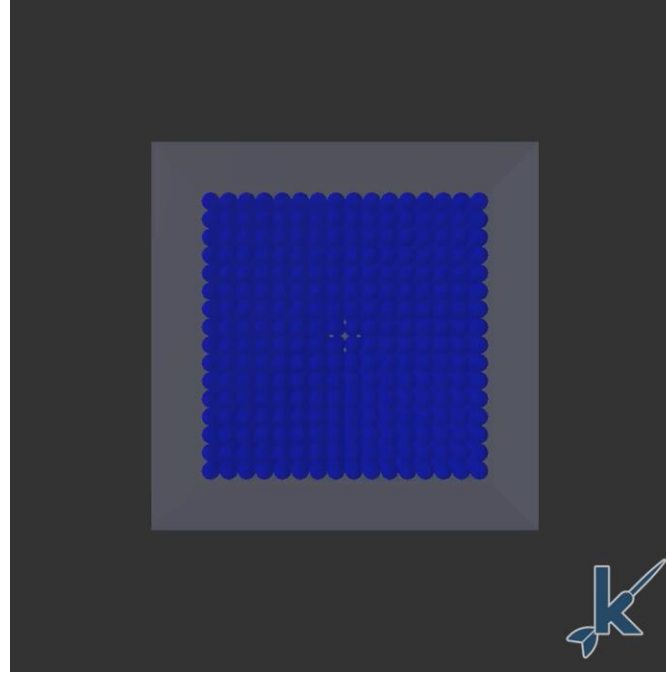


Simulate: 4096 water particles under 1g gravity, colored by density



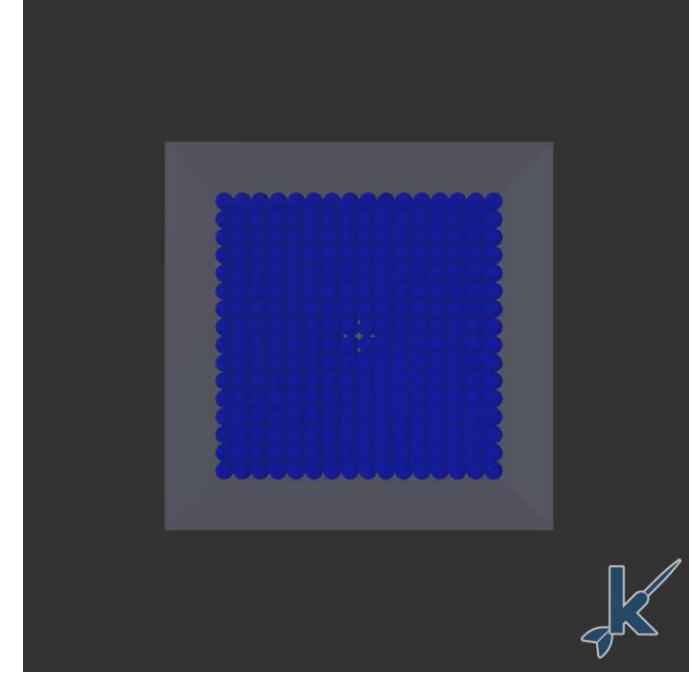
Weakly-Compressible SPH

- Lowest cost per step
- Requires small timesteps
- Density fluctuations



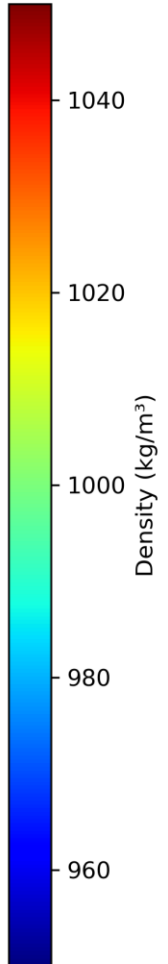
Implicit Incompressible SPH

- Strong incompressibility
- Moderate timesteps
- Slower convergence

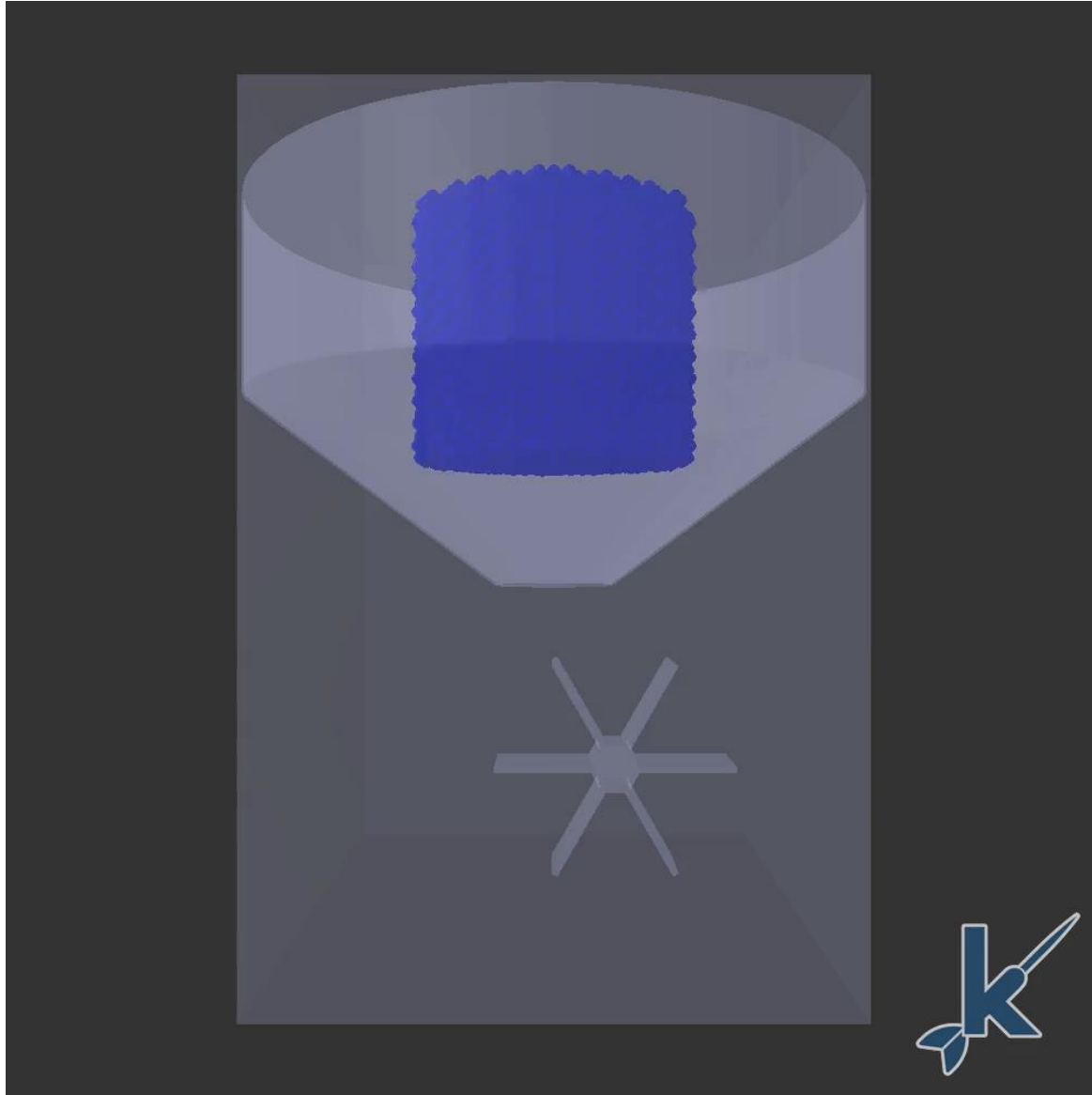


Divergence-Free SPH

- Most robust
- Larger timesteps
- Moderate compute cost



Modeling boundaries

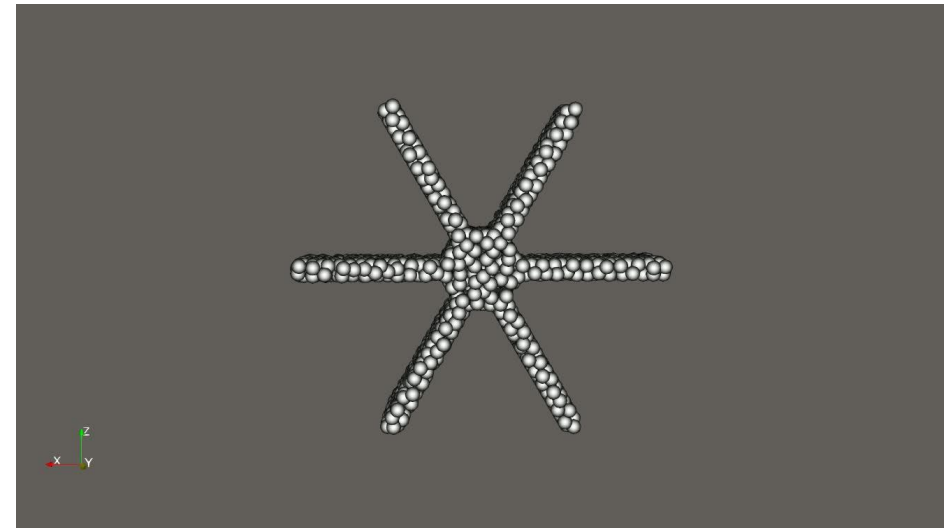


Decompose meshes into boundary markers

- Move them with associated physical body
- Accumulate reaction forces over all attached markers

Finer geometries: use volume maps

- Interpolate mesh using cubic spline grid

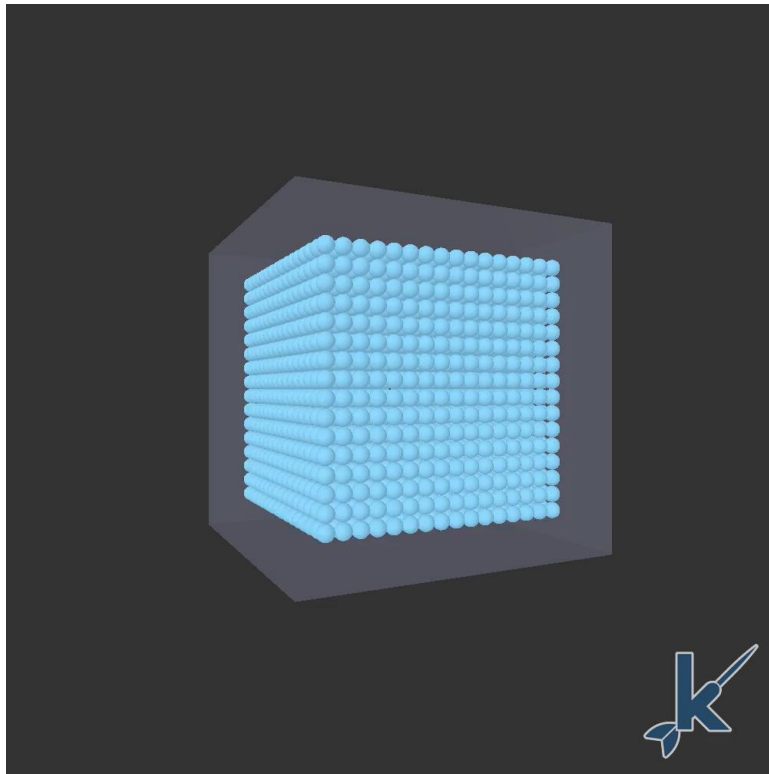


Modeling viscosity and surface tension

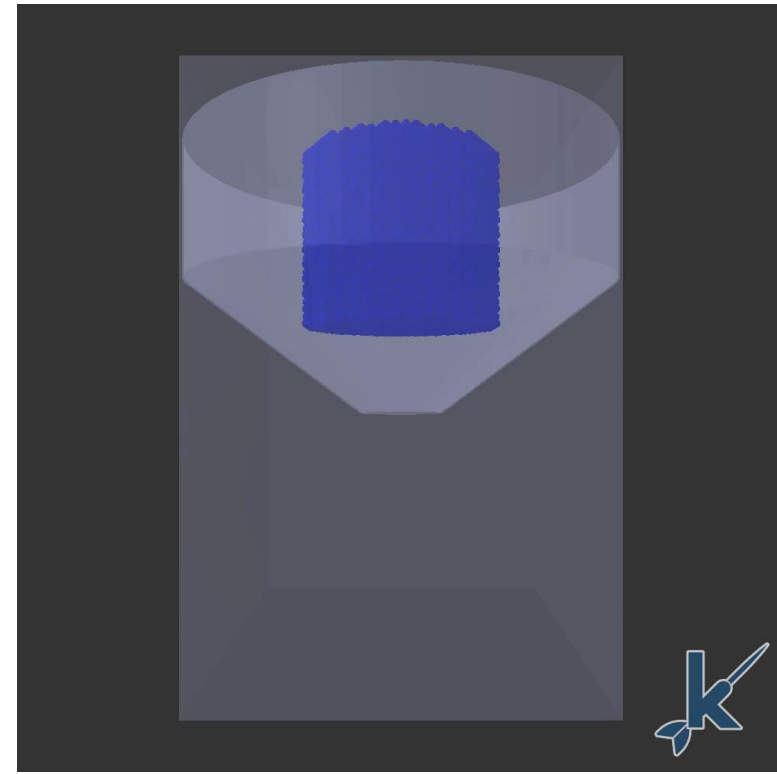


Low-G sloshing: surface tension forces dominate

1. Zorilla, et al. (2020): penalize surface area; classify surface using # nearest neighbors
2. Weiler, et al. (2018): conjugate-gradient solve for viscous forces



7441 adhesive particles with surface tension parameter = 0.5 (zero-g)

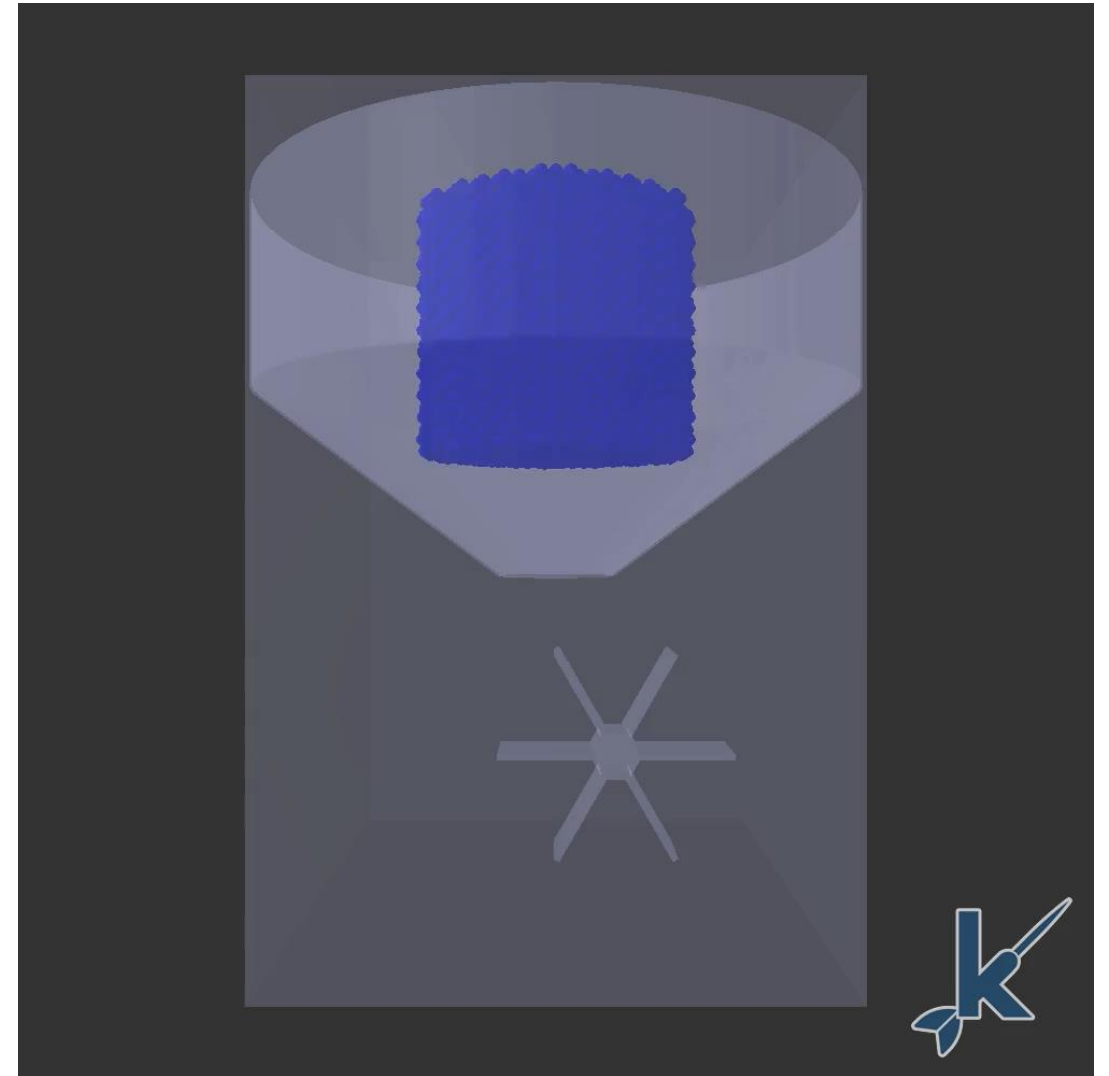


7441 highly-viscous particles flowing through a funnel, colored by velocity

Hyperparameter choices

- DFSPH / IISPH / WCSPH
- Particle size
- Kernel
- Boundary method
- Fluid density
- Viscosity (Fluid + Tank)
- Surface Tension (Fluid + Tank)
- CFL factor

Plus: initial conditions



Depending on particle count and timestep,
2-20X **slower** than realtime (16-core CPU)

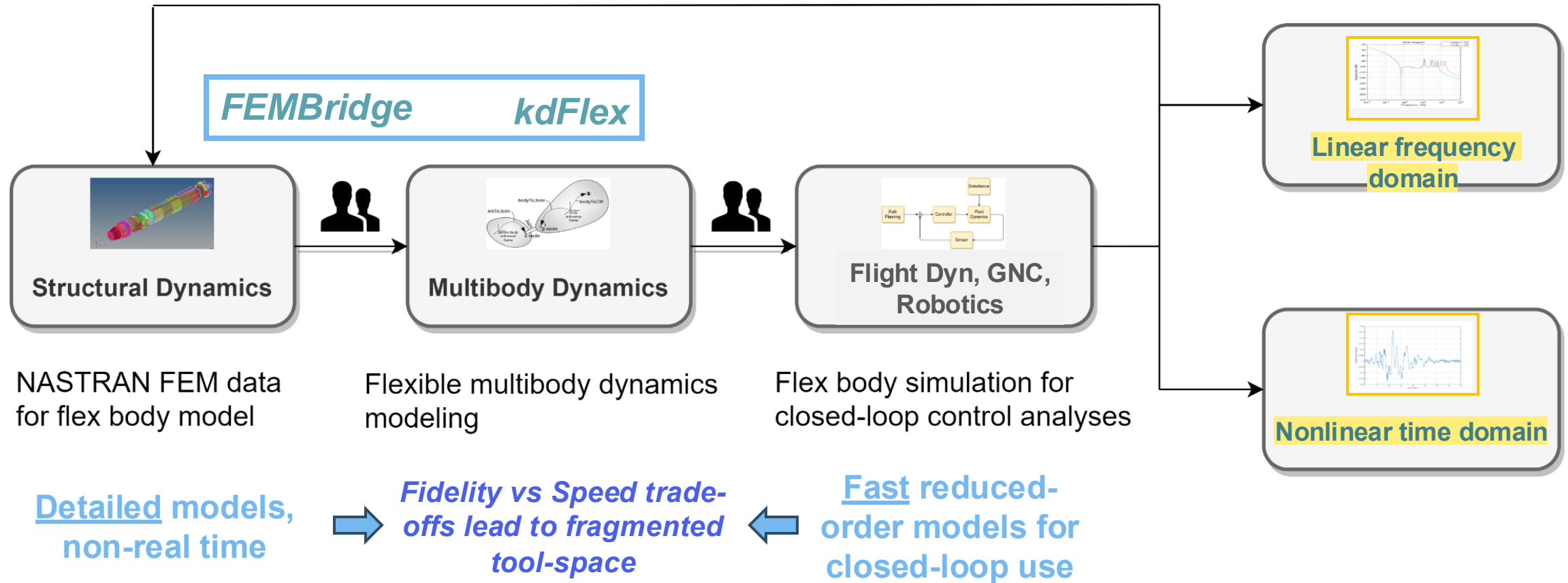


Integrated Simulations

Eliminate the Speed-Fidelity Trade-Off in Flexible Multibody Dynamics

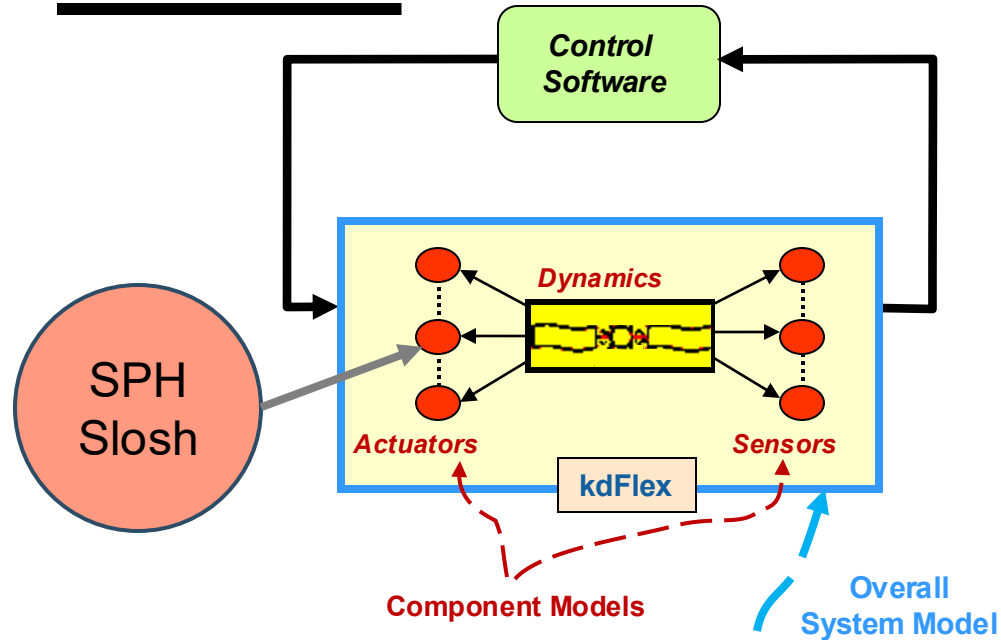


Design Iteration



The Karana team is led by ex-NASA/JPL inventors & developers of SOA dynamics methods and the JPL DARTS simulator. kdFlex is a next-generation capability based on this rich experience heritage.

kdFlex Features Overview



Closed-loop, G&C in the loop flexible body spacecraft simulation

- Extensive API
- C++, Python
- Linux, MacOS, Windows
- Chatbot, documentation, tutorials

Device, environment interaction models

Parameterized models; dynamics aware
Scratch, state variable structs; **pre-fabs**
Multi-rate execution; time delays
Run-time enable/disable
Time stamped outputs

State Propagator

Integrators – fixed & **variable step**
Multi-rate scheduler, state propagation
Step rollback; **zero-crossing** detection
Kinematics and dynamics simulation
Timed events; Hard and soft resets

Model Manager

Multibody and component models
Callback registries for methods
Integrated state management
System **equilibration, linearization**
Multiple solver modes

Multibody dynamics

O(N) SOA fast recursive dynamics
Tree & **graph, rigid/flex** bodies; subgraphs
Collision & contact dynamics
Variable topology & constraints, loads
FEM pipeline; constraint embedding

Simulation services

Frames layer, **ephemerides**
Data logging, message verbosity control
Web based 3D graphics, stripcharts, gui
Data import/export - YAML, JSON, HDF5
Units/quantities support, Monte Carlo

Coupling flexible spacecraft with fuel sloshing

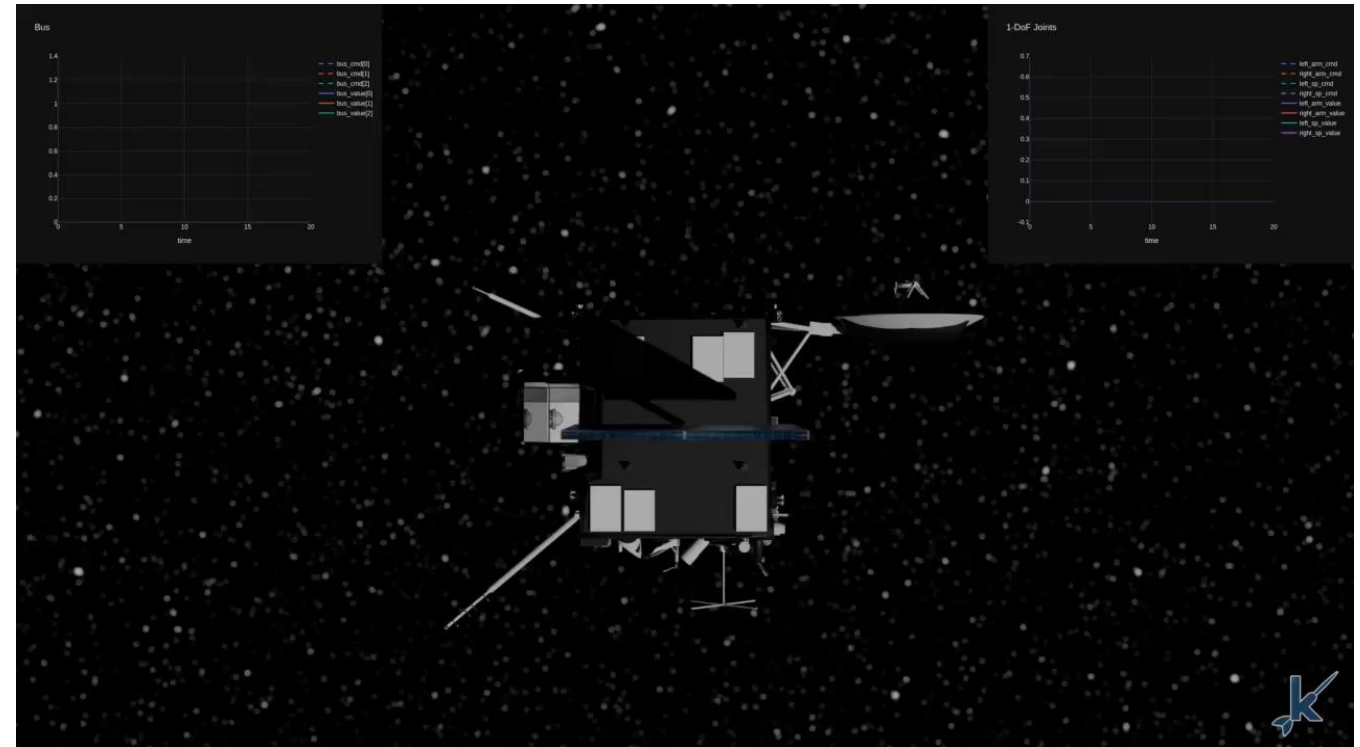


- **Starting point: spacecraft with flexible solar panels and bus controller**
 - Model interaction between fuel slosh and controller
 - “Glue” baffled tank to existing spacecraft assembly

Use reaction wheels to change pose

- Add capsule tank with baffles
- Control profile ramps over 8 seconds
- Rendered slower than realtime for clarity
- Initial fluid unsettled, fill level to target mass (~30%)

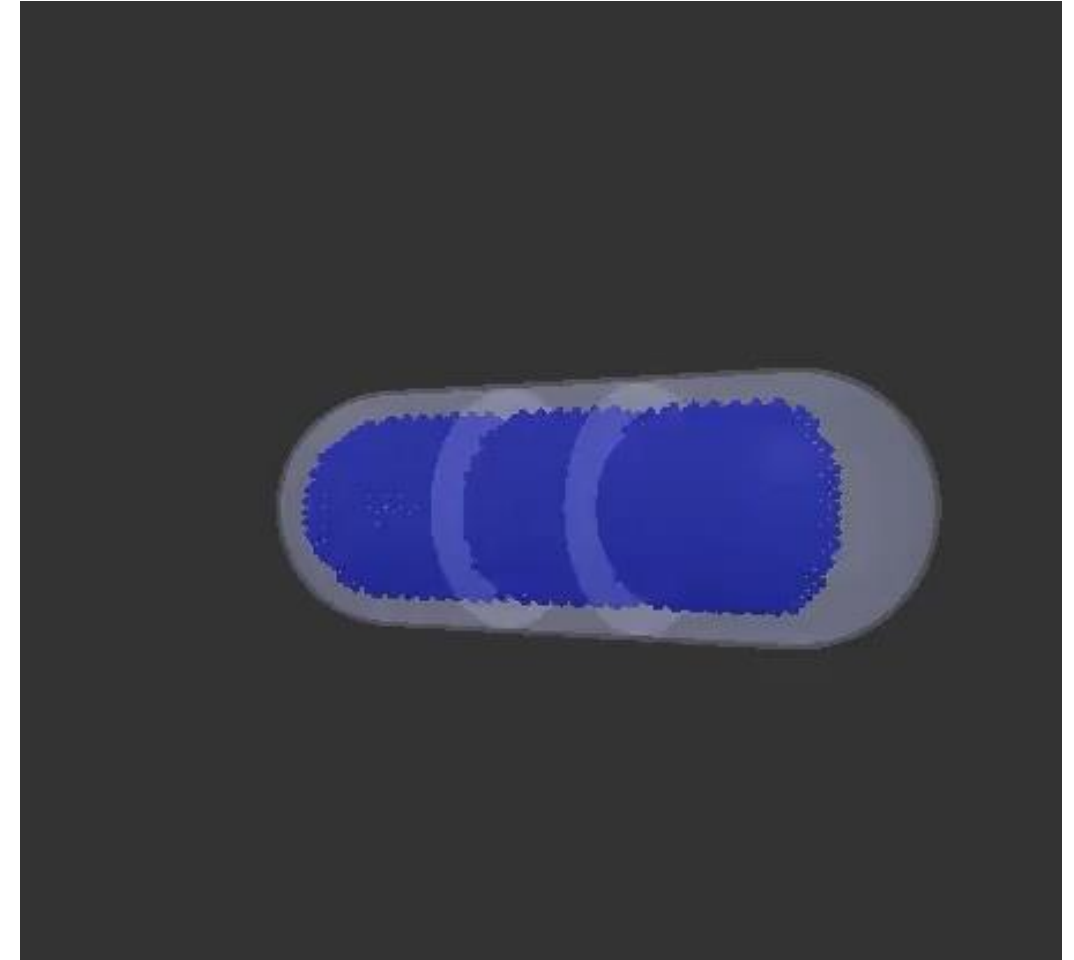
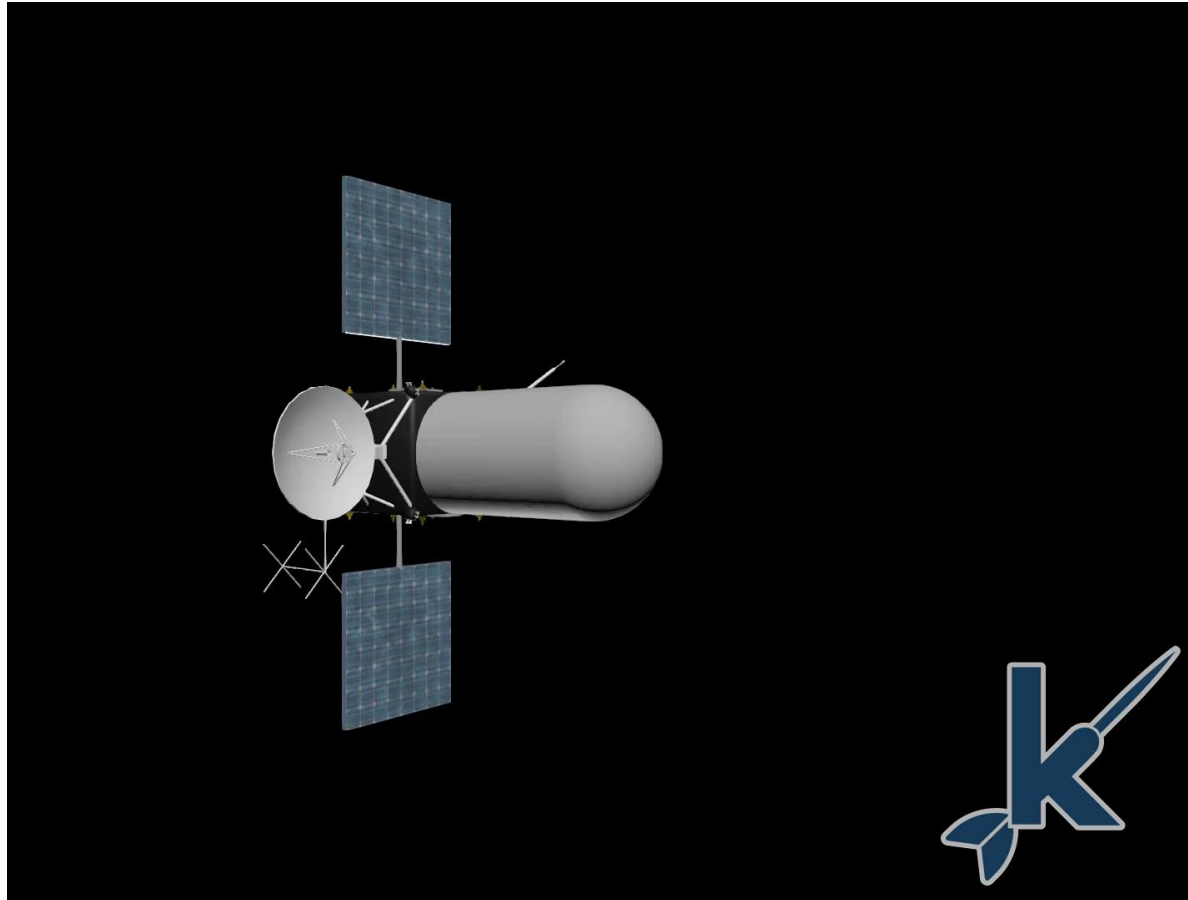
Result: ~120s wall time for 20s sim time



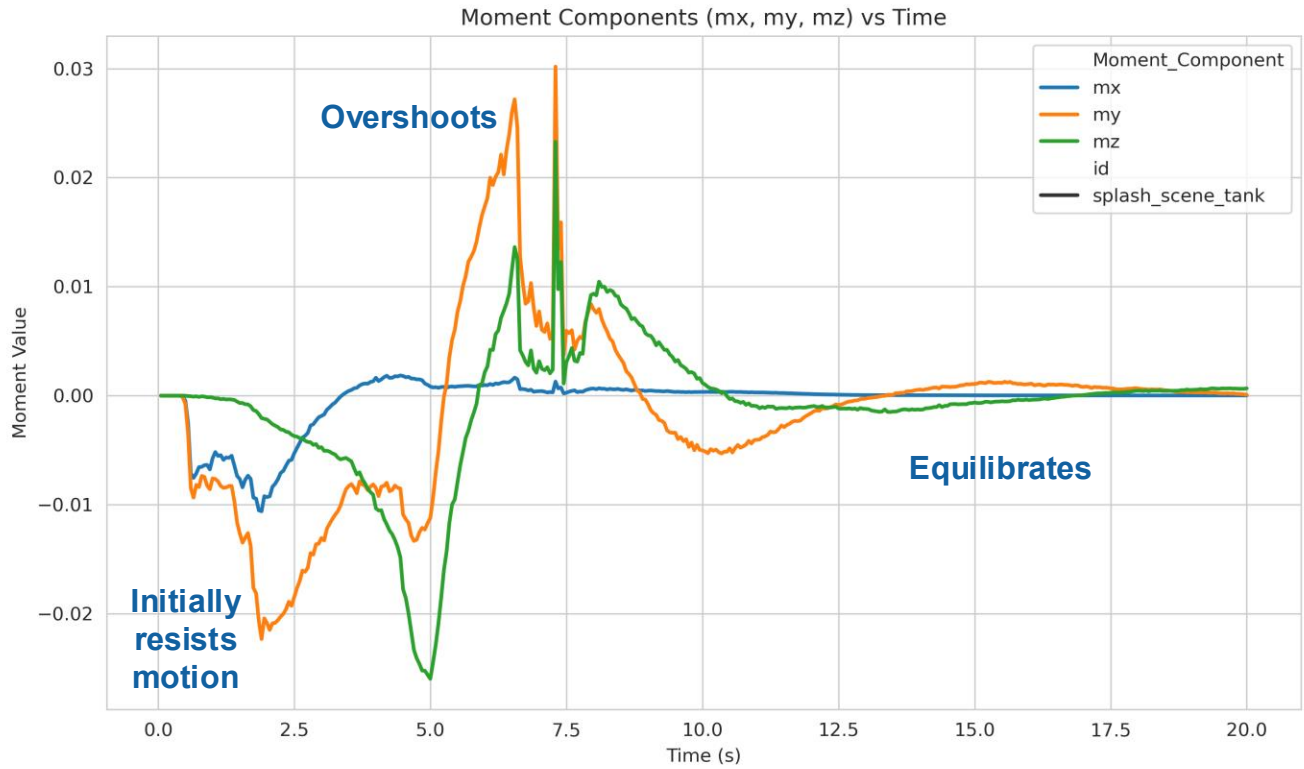
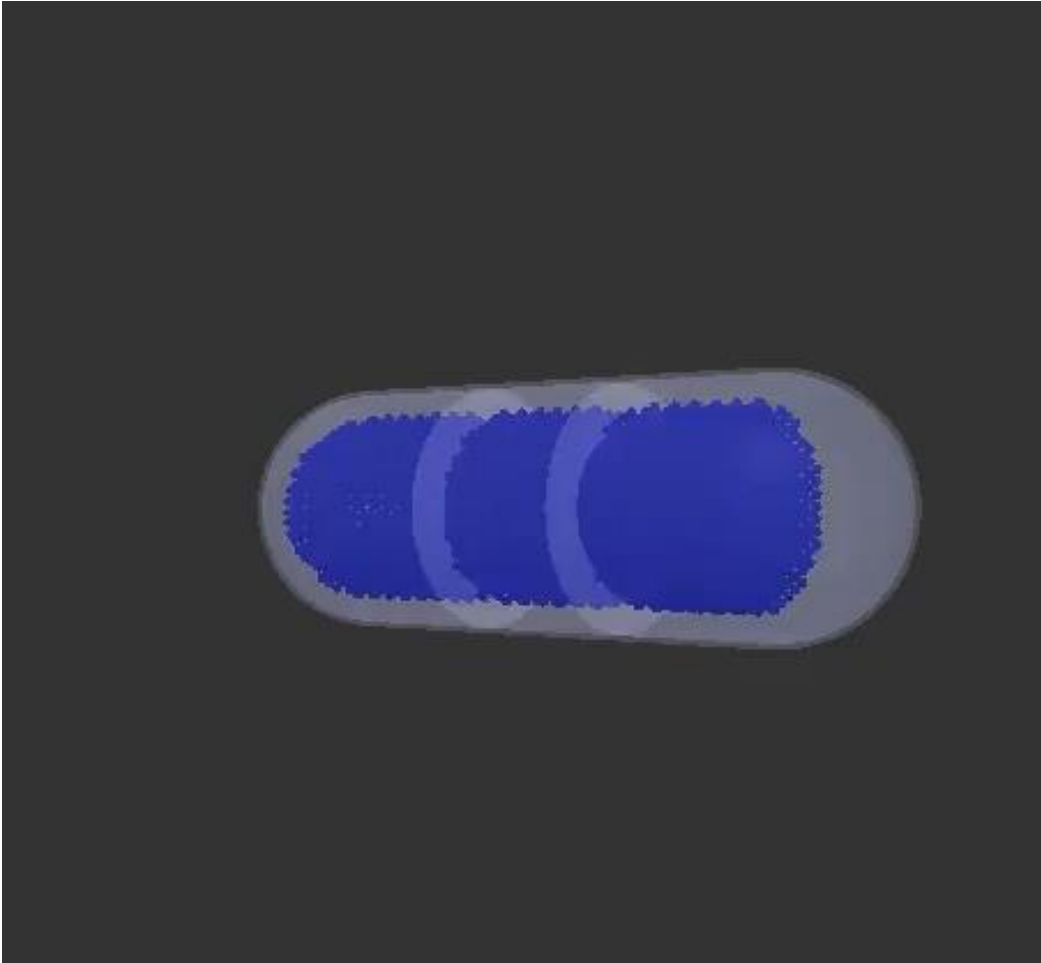
Coupling flexible spacecraft with fuel sloshing



- **Use feedforward controller to set target pose**
 - 8300 SPH Particles in tank, **no boundary adhesion**



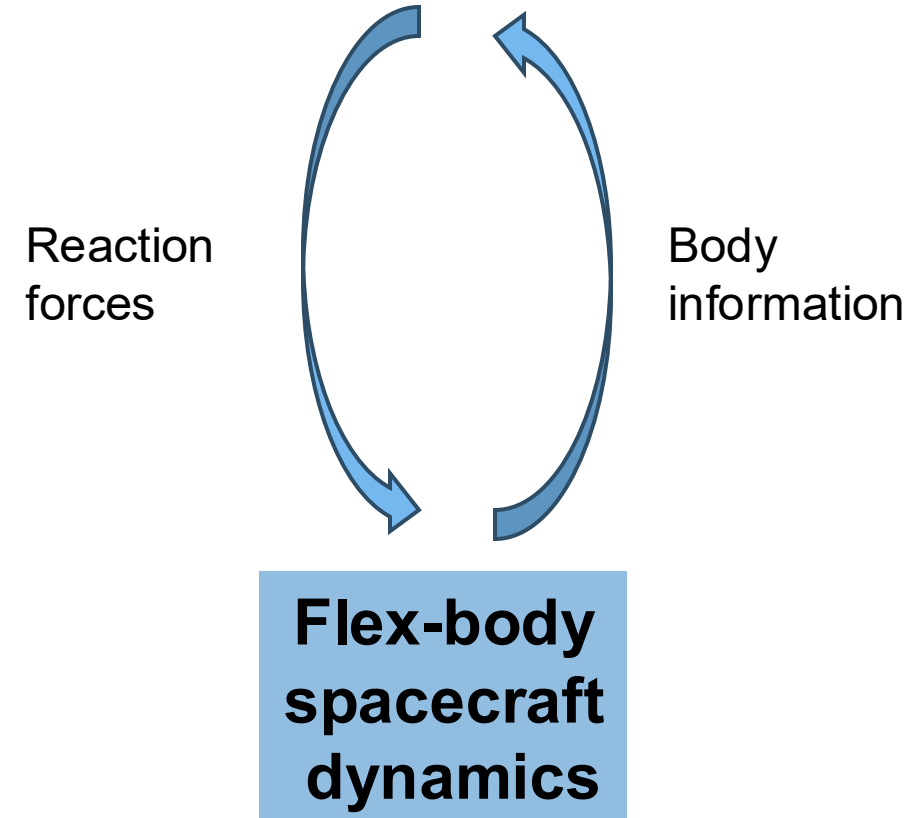
Reaction moments vs time



Moments exerted by fuel in **origin** frame

- Train surrogates from combination of experimental + sim data (multifidelity)
 - Bayesian Calibration (c.f. materials science)
- ML surrogate directions:
 - Neural operator (approximately solve N-S)
 - Autoregressive surrogate for reaction forces
 - Implicit learning (lower data overhead)
 - Flow matching generative models to handle uncertainty in initial conditions, fluid parameters

ML Sloshing



Future work

- Further ablations on numerical methods
 - Understand multicore parallelization performance
- Validation against other libraries and experimental data
- Use to calibrate reduced-order models (pendulums/few-particles)
 - ML real-time surrogates
- Improve timestep coupling
(e.g., parasitic slowdown between variable-step integrators)
- Further speedups available (single-precision, vectorization, GPU)



Thank you!

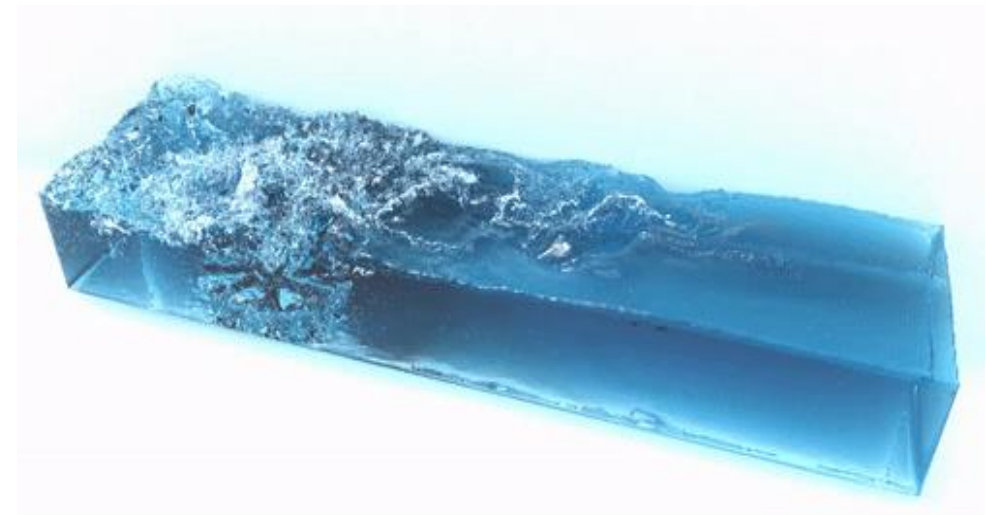
conlain.kelly@karanadyn.com



Backup

- Open-Source (MIT License)
- SOTA pressure solvers
- OpenMP parallelization
- Adaptive verlet timestepping
- C++ API

SPlisHSPlasH



Images source:

<https://github.com/InteractiveComputerGraphics/SPlisHSPlasH>

More about kdFlex



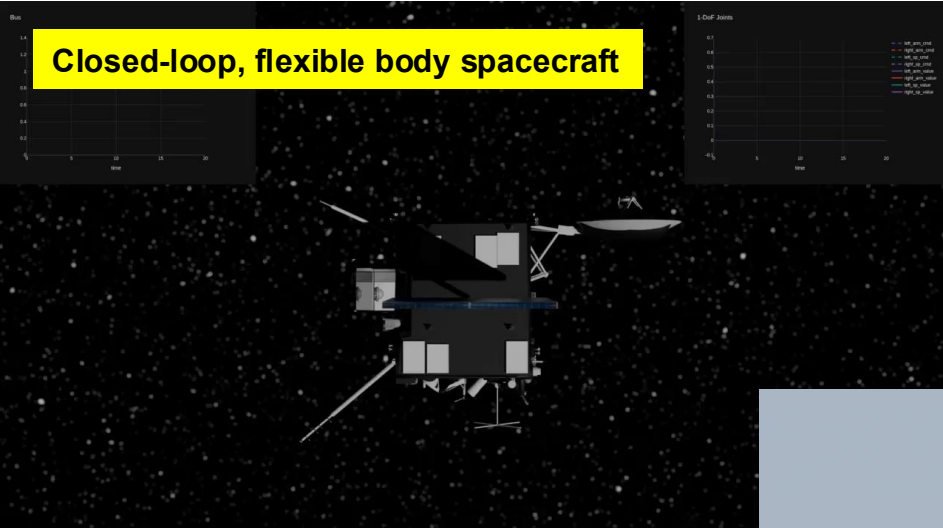
- **kdFlex** is a unique and comprehensive **cutting-edge** rigid/flexible dynamics modeling capabilities.
- The **speed, fidelity, and rigor** combination is unparalleled.
- Can avoid **toolset fragmentation** across programs, orgs, and life-cycles, which improves workflows.
- Rich and **mature heritage** from NASA missions use.

- **kdFlex** is available for **evaluation** at <http://karanadyn.com>
 - Free, no commitment or obligations
- **Resources:** Online documentation, chatbot, several examples and tutorials.
- **Collaborations welcome:** from near-term pilot studies to long-term partnering.

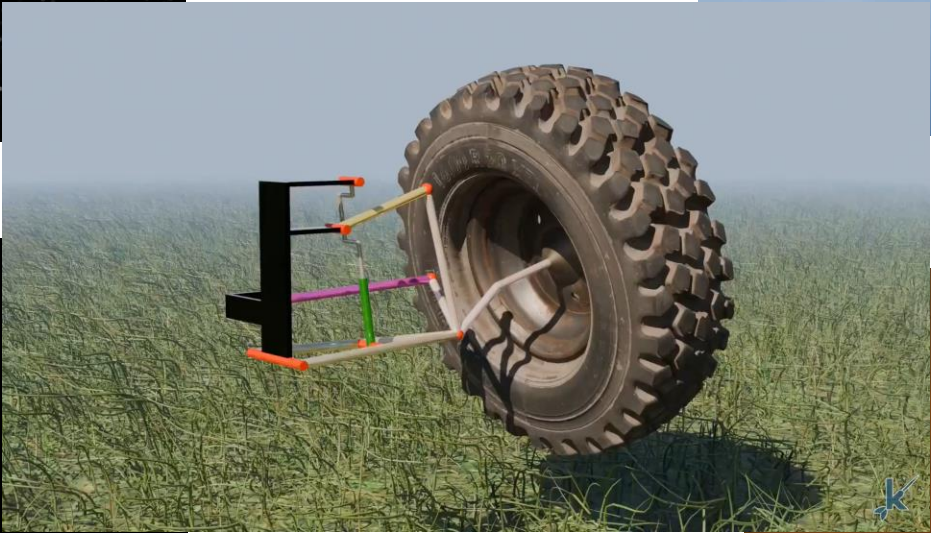
Additional Recent Simulations



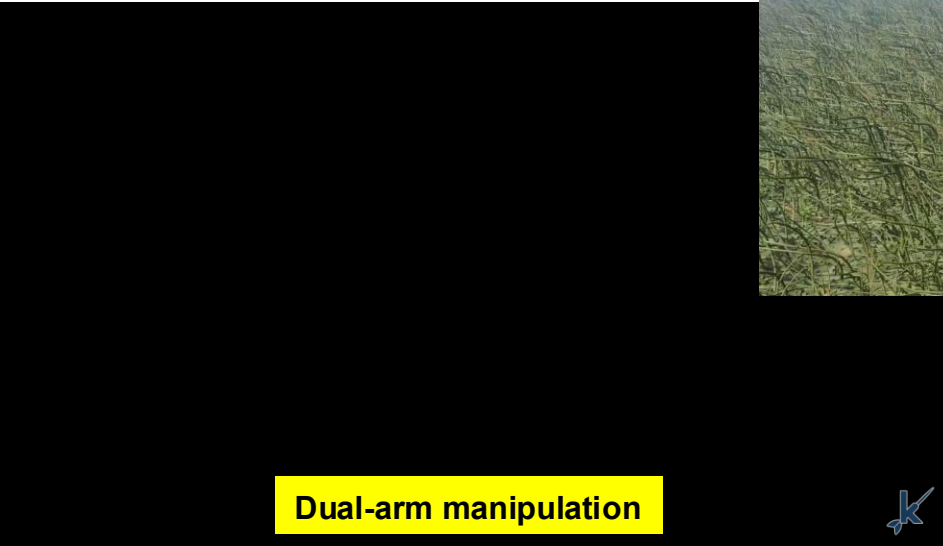
Closed-loop, flexible body spacecraft



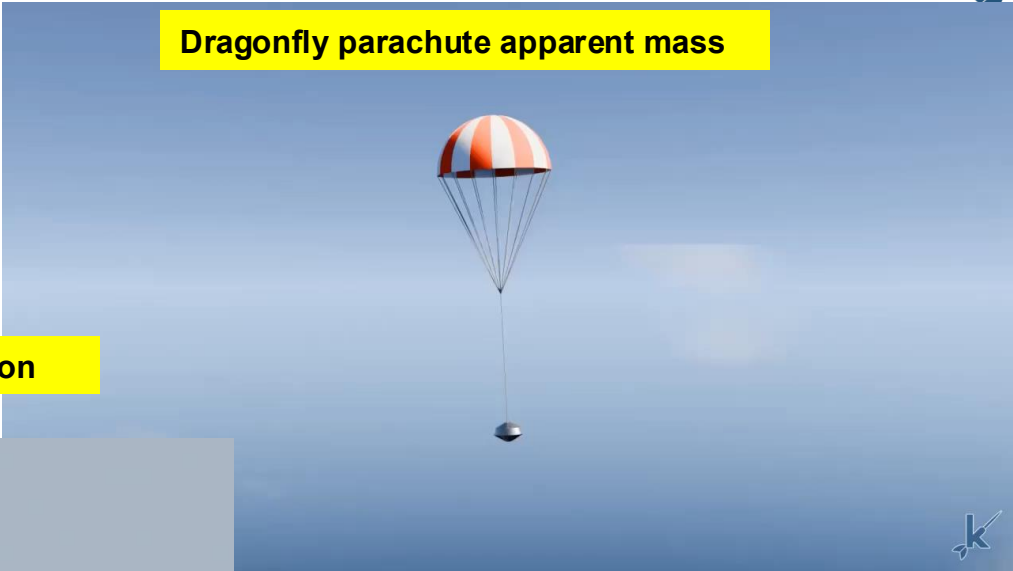
Stress recovery in flex suspension



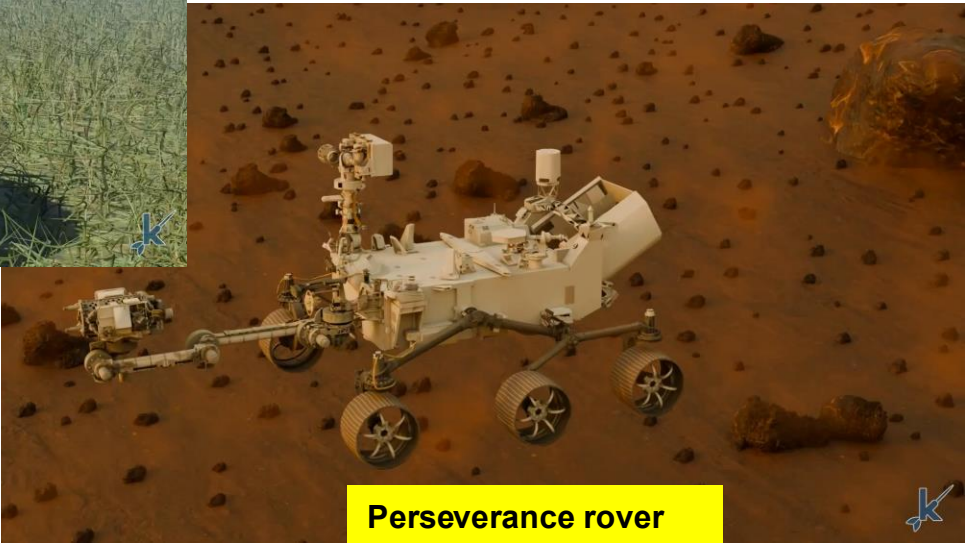
Dual-arm manipulation



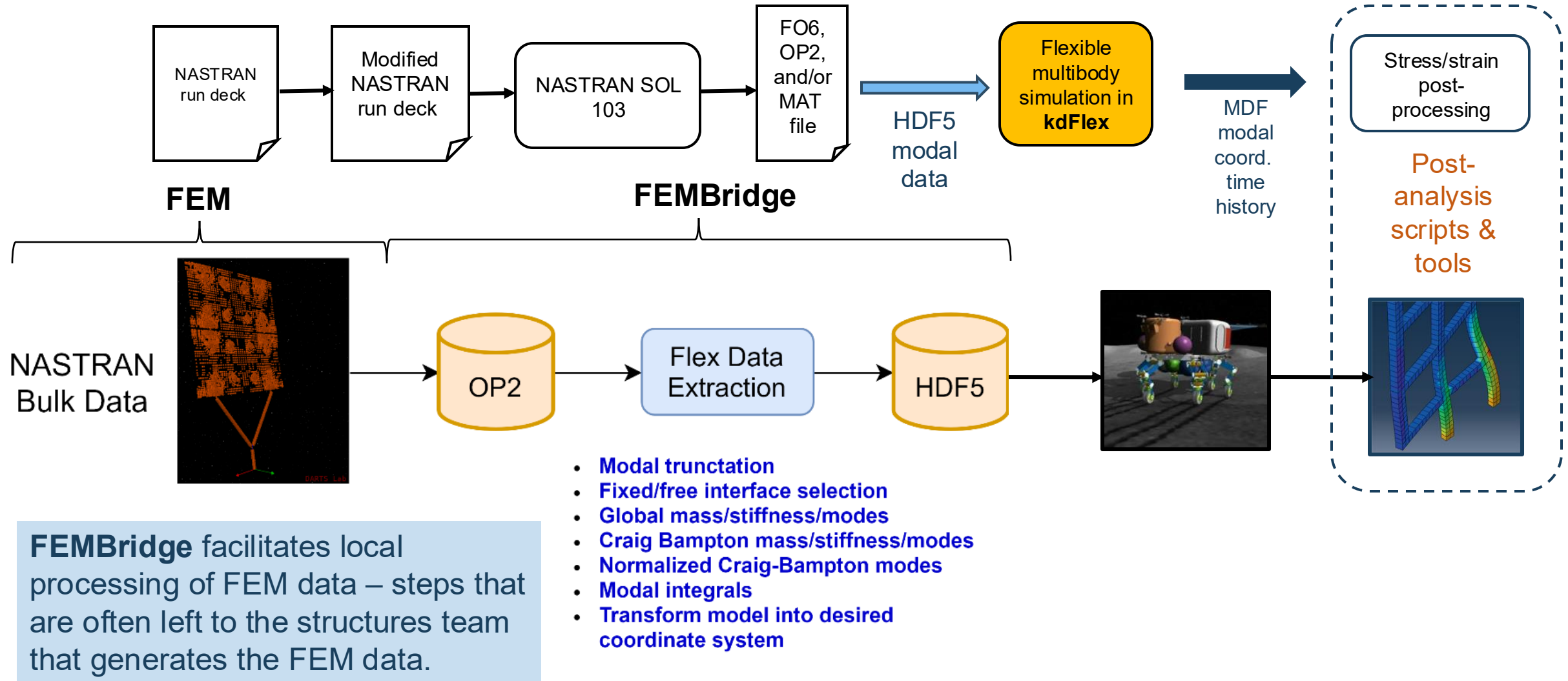
Dragonfly parachute apparent mass



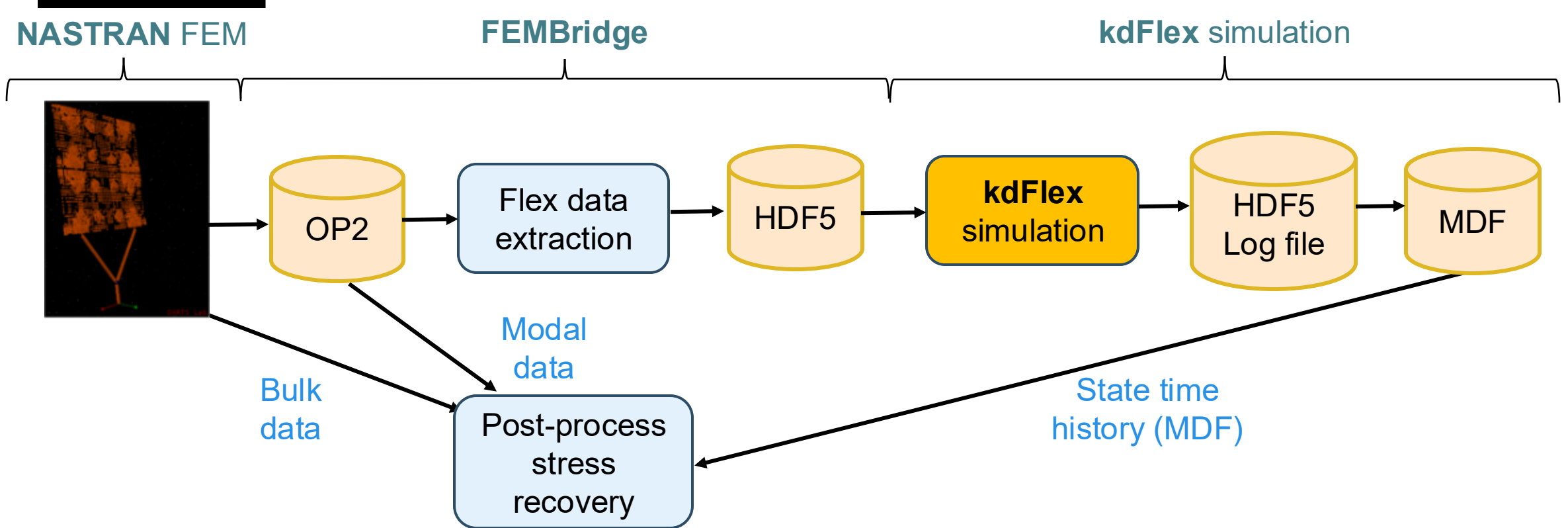
Perseverance rover



FEMBridge: Processing Single Body FEM data



FEMBridge-based Stress Recovery

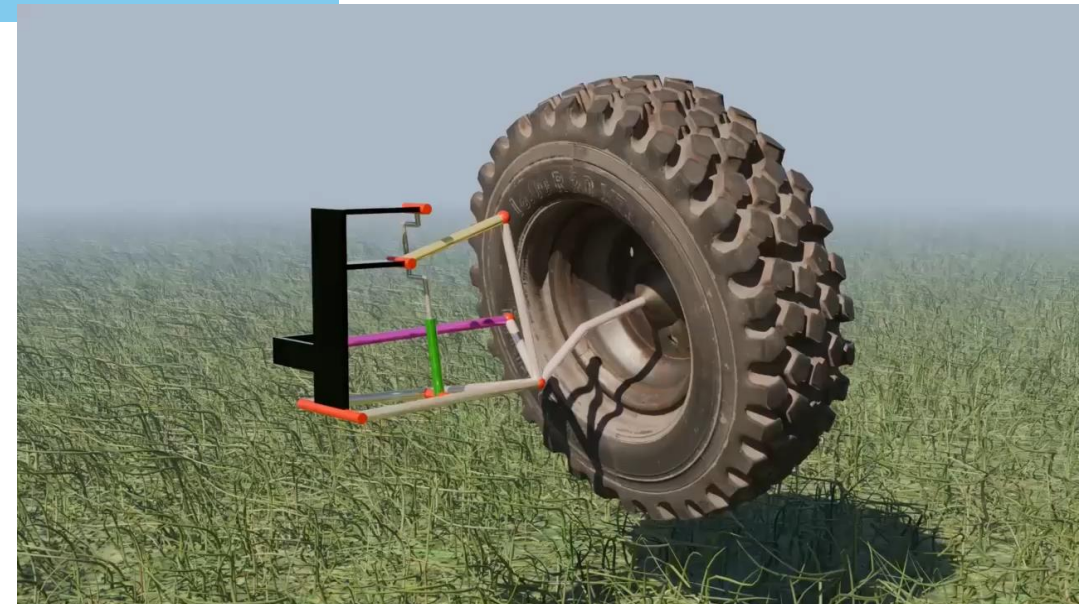
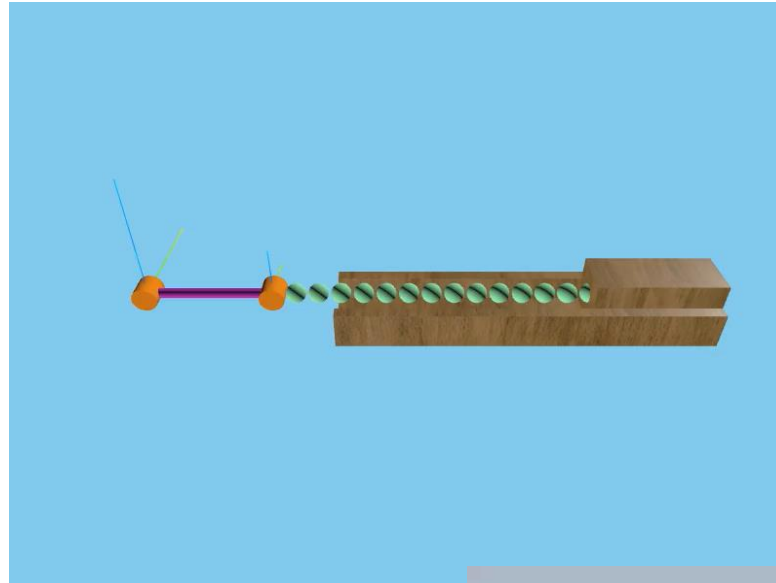


- Initial pre-processing uses NASTRAN to process FEM bulk data for **kdFlex** flexible body data
- During the simulation, extra logging is added to record the modal coordinates, velocities, and accelerations over time
- This data is used to write an MDF file
- The MDF file, FEM, and OP2 file are used as inputs to run NASTRAN for post-processing to recover stress, strain etc. for the flexible bodies

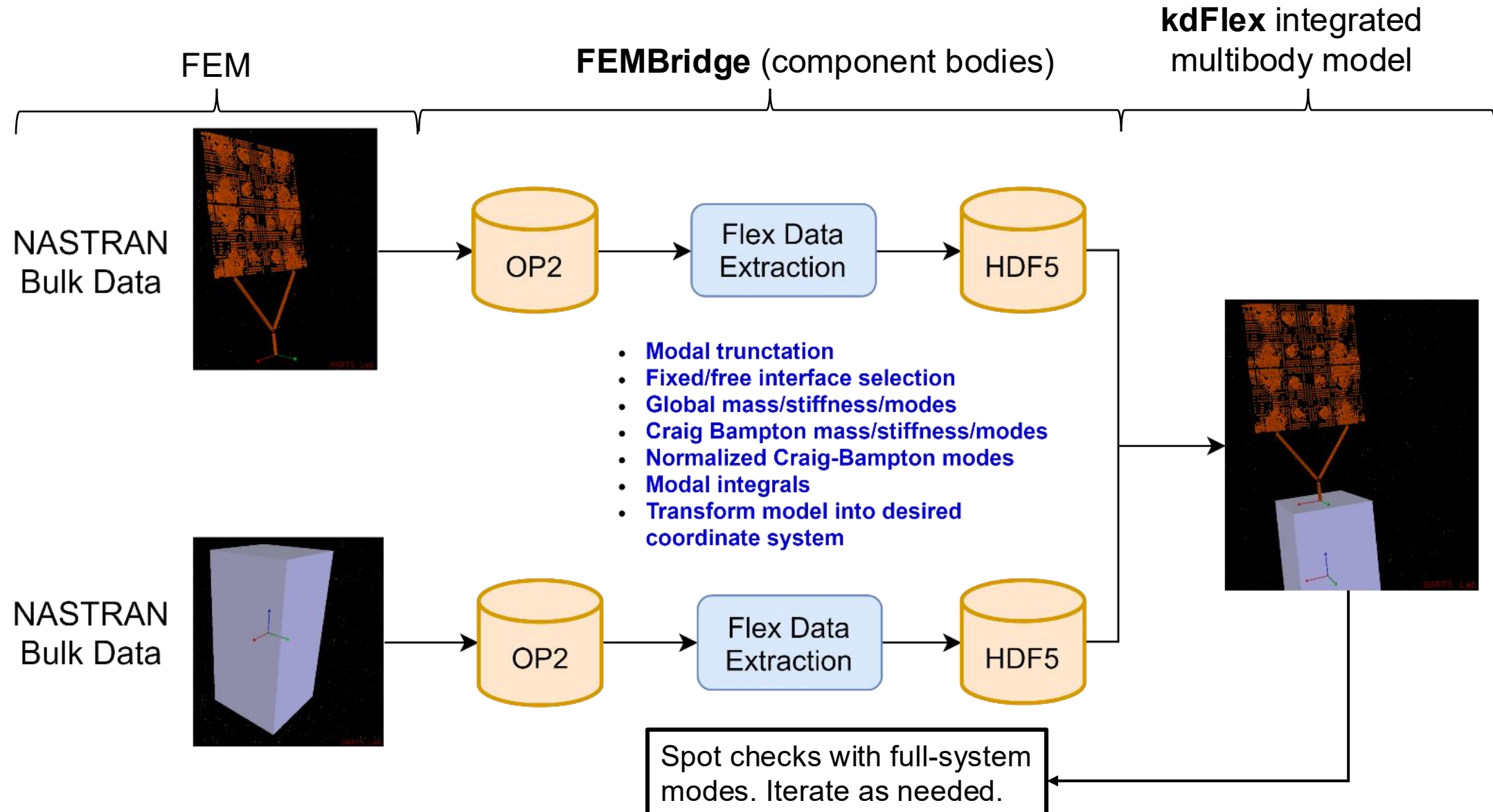
Slider Crank Example: Stress Recovery



- Closed-chain, multibody slider-crank system with a single flexible body
- The time-varying stresses are recovered via post-analysis using FEMBridge
- The colors across the flexible body illustrates the stress variation across the body over time
- Allows stress recovery with realistic G&C scenarios



FEMBridge: Processing multiple body FEM data



Minimal Coordinate Approaches

