GMAT-Monte Model Sharing

18 September 2020

Darrel J. Conway, Thinking Systems, Inc.
John McGreevy, Emergent Space Technologies, Inc.

Overview

- GMAT
  - Project Started in 2002
  - Open Source Releases, Apache 2.0
    - Initial Release, August 2007
    - Roughly annual release on SourceForge
    - Developed behind the GSFC Firewall
  - Current Release, R2020a, July 2020
  - Widely Used in Academia, Industry, Government (US and International)
  - Controlled Components managed as unreleased plugins
- Operational Use, GSFC FDF
  - Mission Design
  - Maneuver Planning
  - Orbit Determination
- Windows, Linux, Mac
- Part of NESC Tools Interoperability Effort

- GMAT API
  - Beta Functionality in R2020a
    - Beta → Release via testing
    - Release includes Python, MATLAB, and Jupyter examples
  - Access to
    - GMAT Classes
    - GMAT Scripting
- **Tools Interoperability Work**
  - OSIRIS-REx Based Use Cases
  - Data Sharing with Monte
    - Monte 140.1, GMAT R2020a
    - Ephemeris Round trip Sharing, April 2020
    - Maneuver Plan Sharing, In Development
  - Model Sharing with Monte
    - Matched Dynamics in GMAT and Monte
    - Enabling Technology: GMAT Collocation and Monte Orbit Determination

Use Case Progression

Data Sharing

- Ephemeris Sharing (Complete)
  - Monte writes ephem
  - GMAT ingests, and plans a maneuver
  - GMAT Writes ephem
  - Monte ingests, and works with planned data
- Maneuver Sharing (Underway)
  - Monte writes ephem
  - GMAT ingests, and plans a maneuver
  - GMAT Passes Mnvr Data to Monte
  - Monte ingests Mnvr and works with it

Model Sharing

- Issue: GMAT and Monte Models may be inconsistent
- Goal: Use Monte Dynamics during GMAT maneuver planning
- Next Slide: NESC Plan Update

  Flight Mechanics Analysis Tools Interoperability and Component Sharing (TI-18-01313)

  October 17, 2019

NESC logo inserted *after* NRB approval

GMAT API Updated Subtask Details

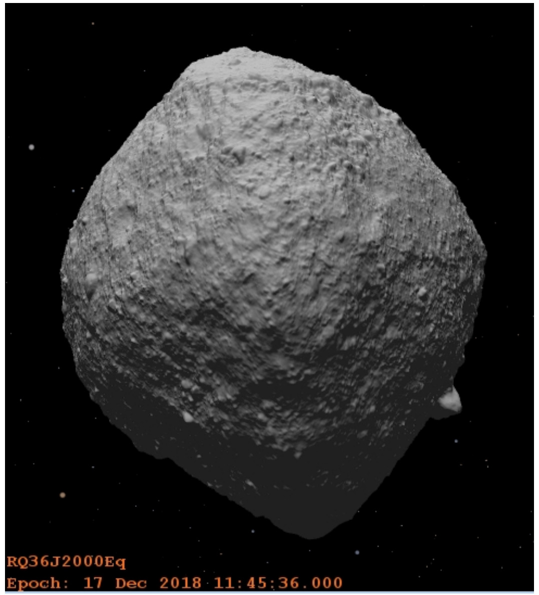
Phase 1: Subtask 1.2 Data Interoperability

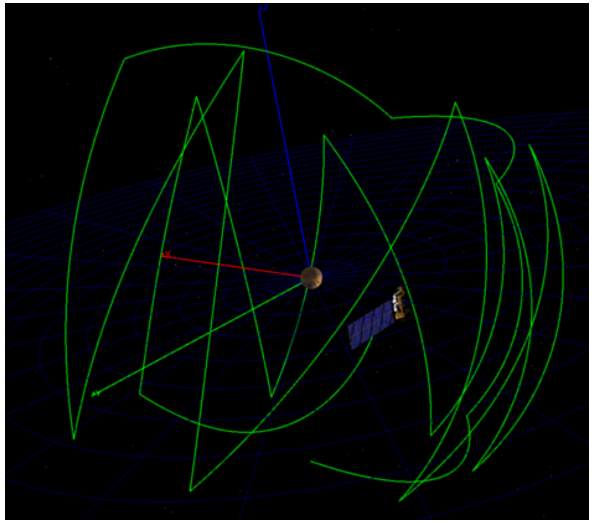
- O-REx use case definition
- MONTE setup and GSFC training
- Ephemeris sharing
- Maneuver plan sharing
- Re-targeting

Phase 2: Subtask 1.3 Component Integration

- C++ interface definition
- Dynamics interface coding
- GMAT Monte dynamics model integration
- O-REx demonstration configuration
- Demonstrate OD in Monte and Optimization in GMAT using shared dynamics

Note that subtask 1.3 is a tighter integration than originally planned as it will use C++ rather than Python APIs.

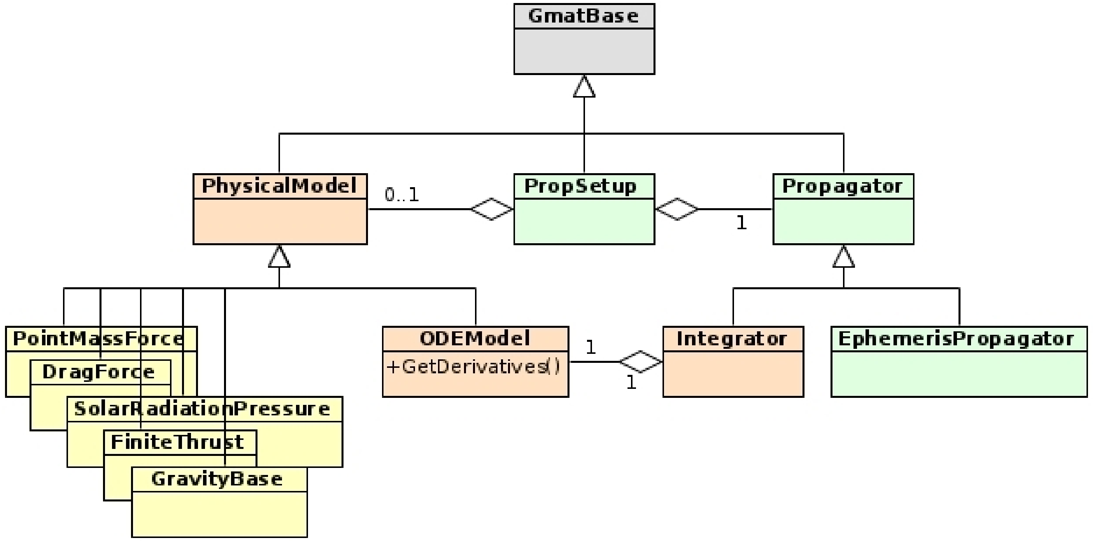




Restricted distribution to NESC and designated team members until reviewed by the NRB.
This is for status only and does not represent complete engineering analysis.

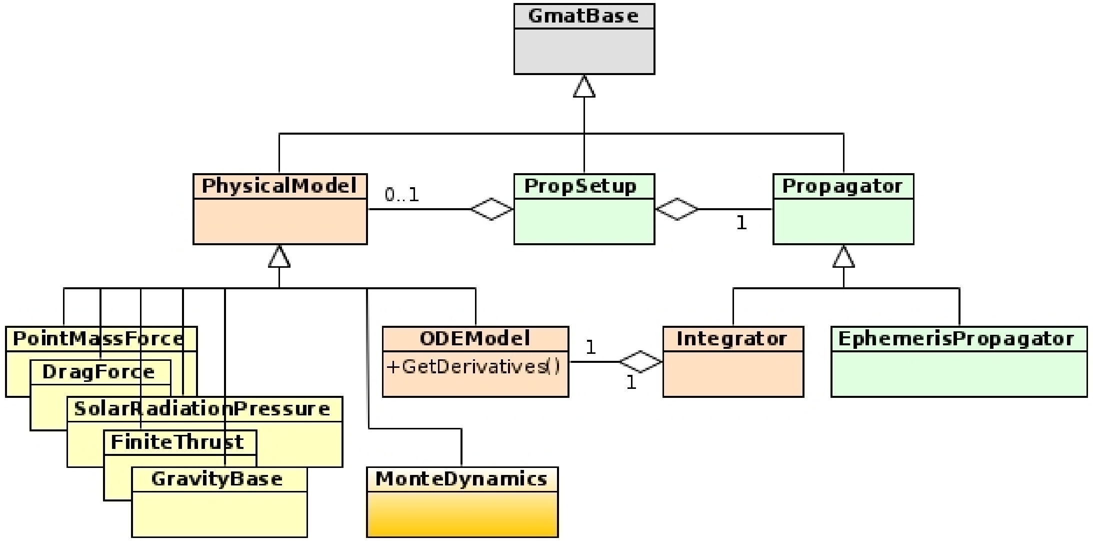
Model Sharing: GMAT Approach

- GMAT Dynamics
  - Part of Propagation Subsystem
  - Consolidated in ODEModel class
  - Derivatives Calculated in GetDerivatives()
- Proposed Approach for Monte Dynamics



Model Sharing: Adding Monte Dynamics

- GMAT Dynamics
  - Part of Propagation Subsystem
  - Consolidated in ODEModel class
  - Derivatives Calculated in GetDerivatives()
- Proposed Approach for Monte Dynamics
  - Add a class that wraps Monte Dynamics
  - Implement GetDerivatives() in Wrapper, compatible with GMAT
- Requires
  - Monte C++ headers
  - Libraries compatible with the headers
  - Help figuring it all out



Discussion

- Compiler Compatibility
  - GMAT uses
    - GCC 8.3.1 on Red Hat
    - Visual Studio 2017 on Windows
  - Monte?
- Access Help
  - JM and DJC need help understanding how Monte Dynamics work
- Other Items/Issues/Concerns?