Finanziato dall'Unione europea NextGenerationEU | Ministero dell'Università e della Ricerca | Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA | ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

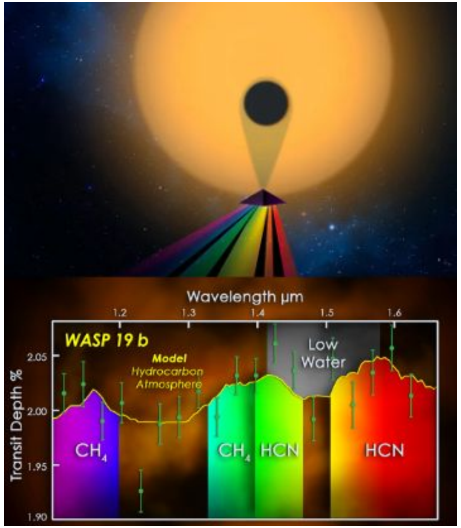
Parallelizing the Mercury-Arxes code using OpenACC

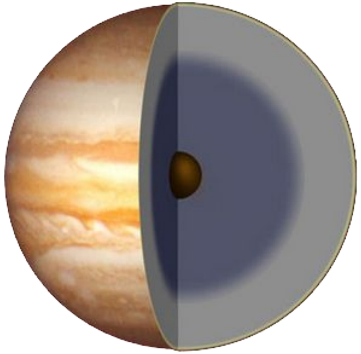
P. Simonetti, D. Polychroni, D. Turrini, R. Politi, S. Ivanovski and the OPAL team

Spoke 3 II Technical Workshop, Bologna Dec 17 -19, 2024

Scientific rationale

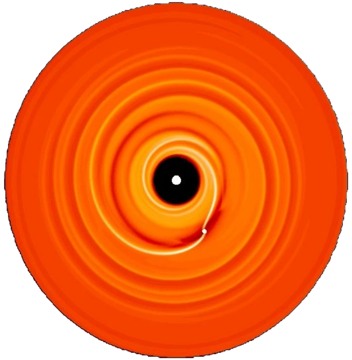
1. Chemical composition of the atmosphere from observations

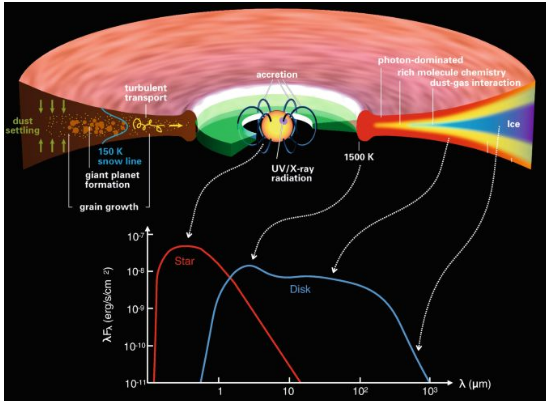




2. Chemical composition of the whole body from the atmosphere (gas giants)

3. Formation & evolution from chemical composition



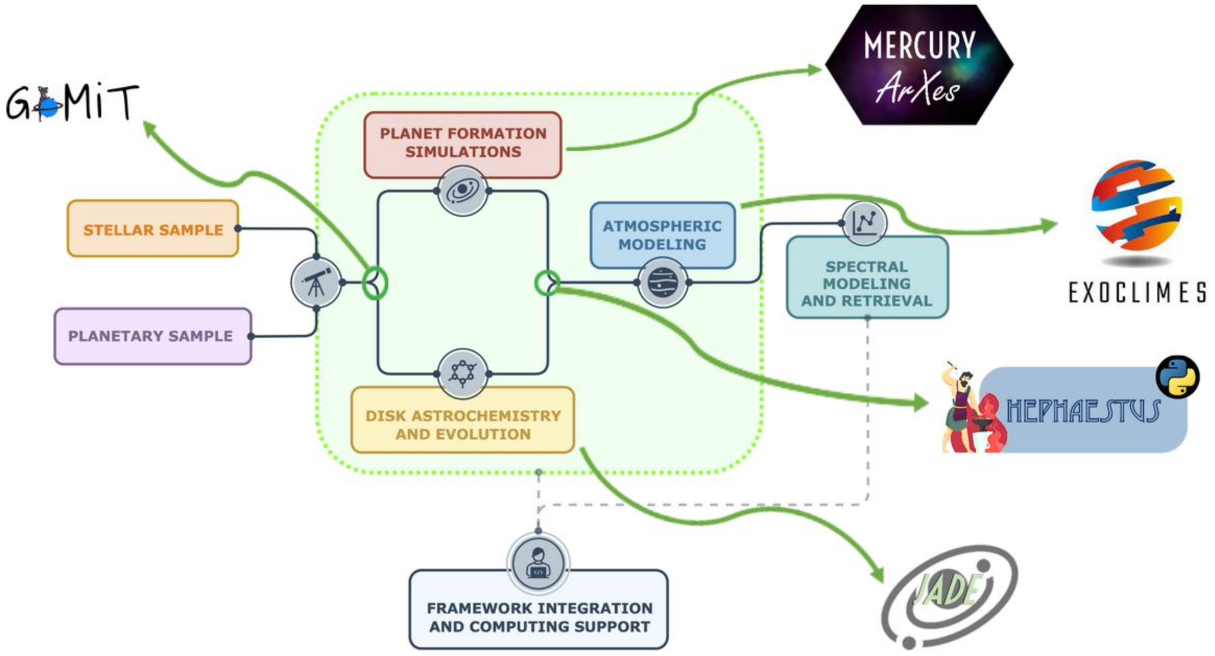


4. Understand planetary formation in a cosmic context

Scientific rationale

OPAL KSP: a pipeline of codes to self-consistently model how a planet looks like from its history:

- **Chemical composition of the protoplanetary disk**
- **Planetary formation & migration**
- **Atmospheric chemistry**
- **Synthetic spectra**



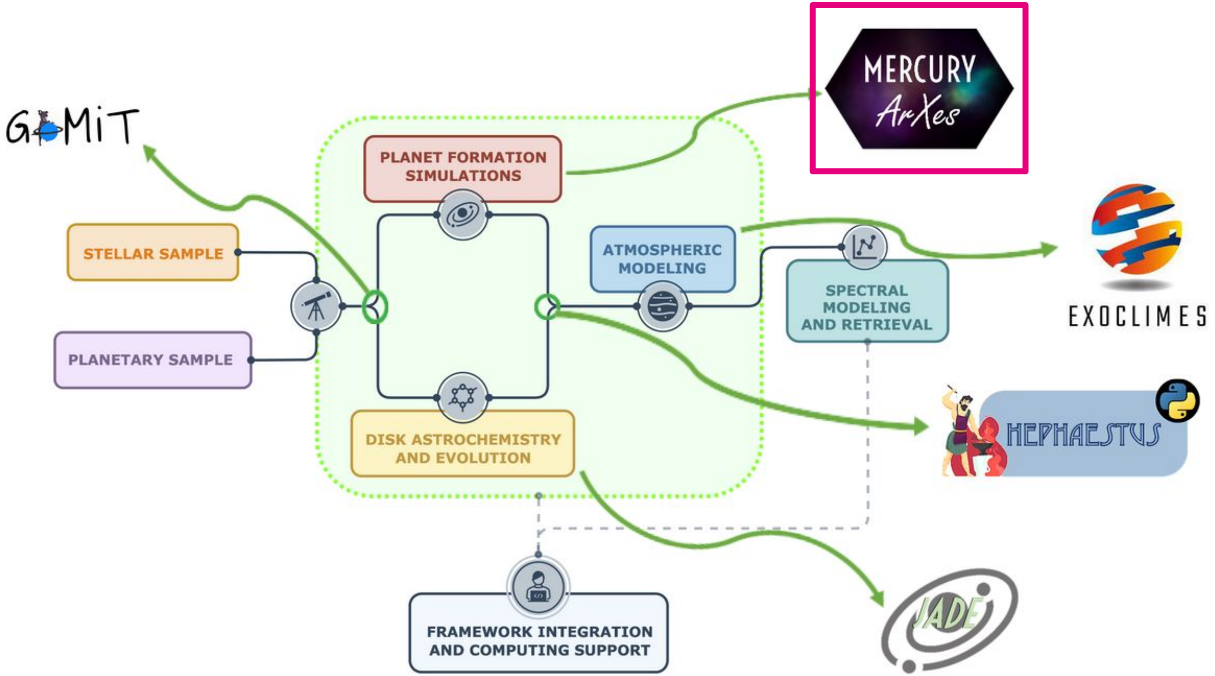
In the context of the italian contribution to the Ariel Dry Run

The OPAL Project

OPAL KSP: a pipeline of codes to self-consistently model how a planet looks like from its history:

- **Chemical composition of the protoplanetary disk**
- **Planetary formation & migration**
- **Atmospheric chemistry**
- **Synthetic spectra**

In the context of the italian contribution to the Ariel Dry Run



Starting point and objectives

MERCURY ArXes

- Mercury (Chambers+99): multipurpose n-body integrator
- Mercury-Arxes (Turrini+21): branched out version tailored to study planetary formation:
    - based on Mercury 6
    - viscous interaction with a time-evolving disk
    - management of merging trees
    - hybrid symplectic integrator to conserve energy and momentum over ~Myr timescales
- Current typical simulation: <10 gravity-source *planets*, >10^4-10^5 non gravity-source *particles*

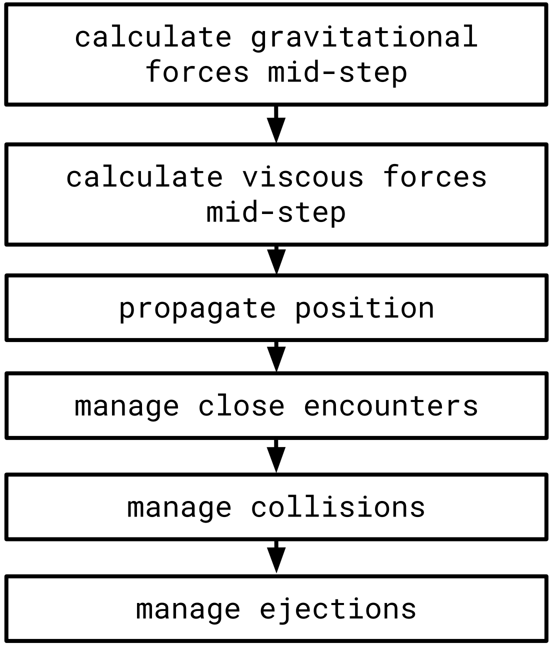
GOAL: have all the *particles* as gravity-sources

- especially important at the beginning of the simulation, when most of the mass is in *particles*

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Methodology

Mercury-Arxes main loop:

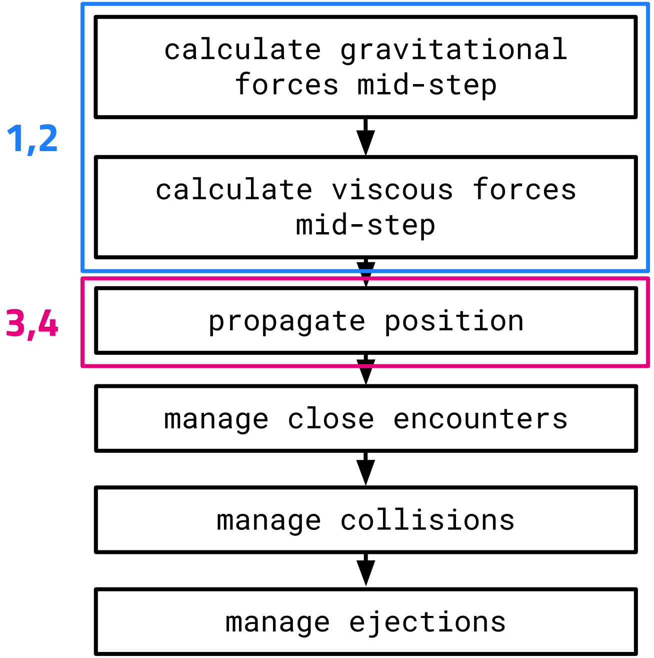


To achieve our goal, high level of parallelism is needed. In practice we:

- **parallelize the main calculations on GPU using OpenACC**
- **parallelize the rest of the code on CPU using OpenMP**

Why? Minimal need for code refactoring with respect to the potential gain

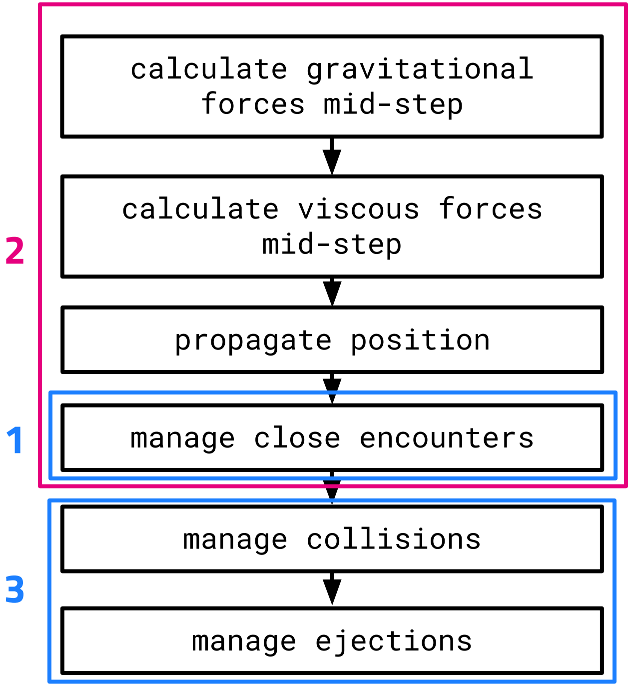
Current status: interventions done



1. merged gravitational and non-gravitational forces in one routine
2. ported force calculation on GPU
3. translated the position propagation routines from C to Fortran
4. ported position propagation on GPU

(3) needed because of issues in the interaction between OpenACC and the external C routines - solution suggested by G. Lacopo (Univ. of Trieste, INAF-OATs)

Current status: missing work



Parallelization still incomplete:

1. **data management has not been optimized**
2. **still managed sequentially → this will be especially evident in the next few slides!**
3. **again, currently managed sequentially (but it's mostly array manipulation)**

Main results

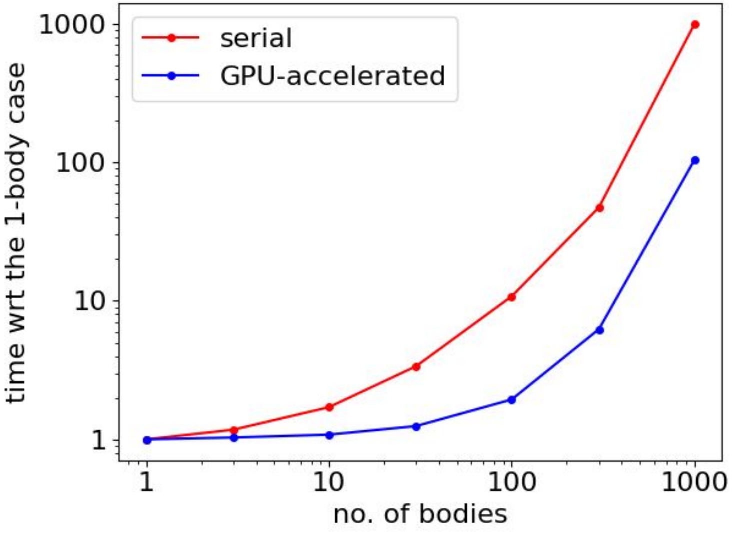
- marginal speedup from (1) and (3): few % decrease in runtime
- large speedup from (2) and (4): up to a factor of 10 decrease in runtime
- substantially improved serviceability of the code: now it is completely written in fortran

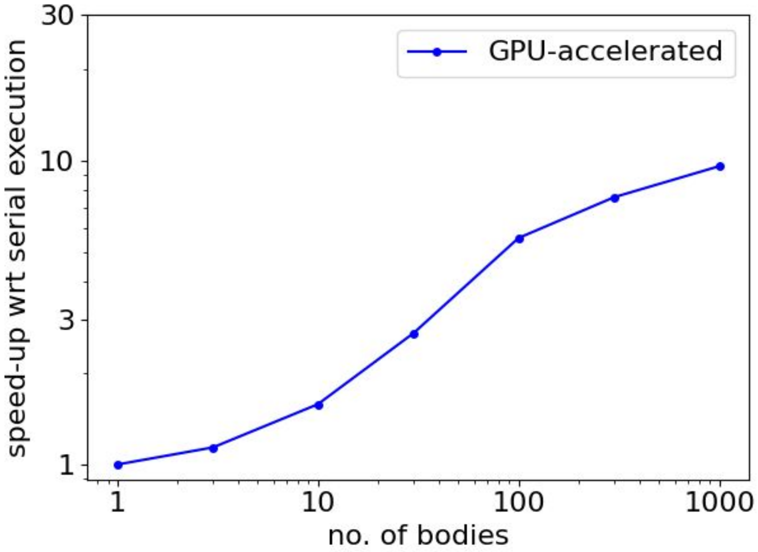
Results affected by not having completed the parallelization process:

- inefficient data management (30-35% of total time are memory transfers)
- inefficient execution of the serial part when compiled with nvfortran (~6-7 times slower wrt the ifort/ifx compiled code)

Main results

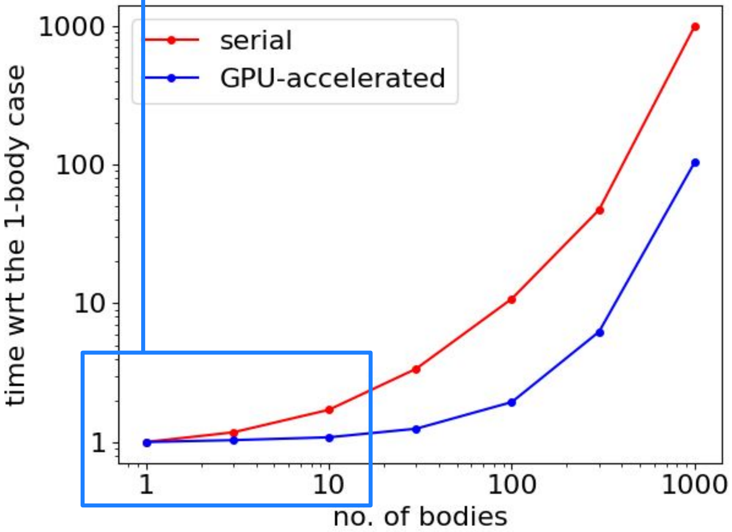
Competition btw overhead times (at low no. of bodies) and incomplete parellelization (at high no. of bodies)



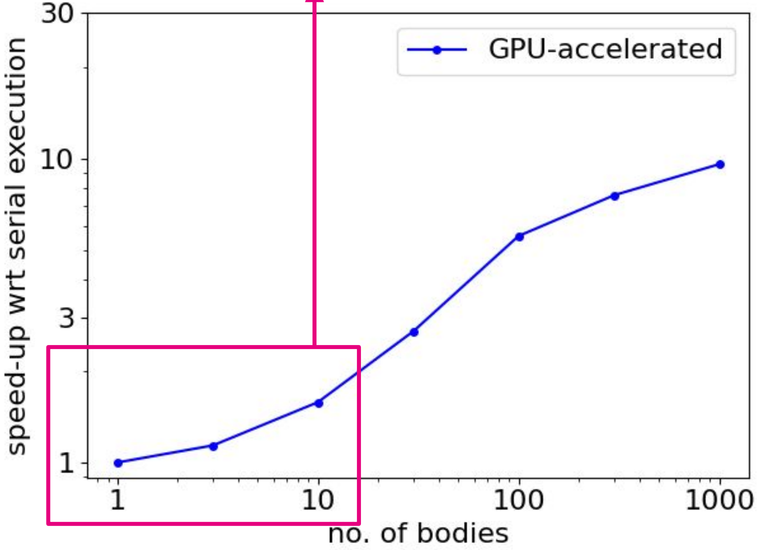


Main results

Few close-encounters: parallelization keeps execution time mostly unvaried



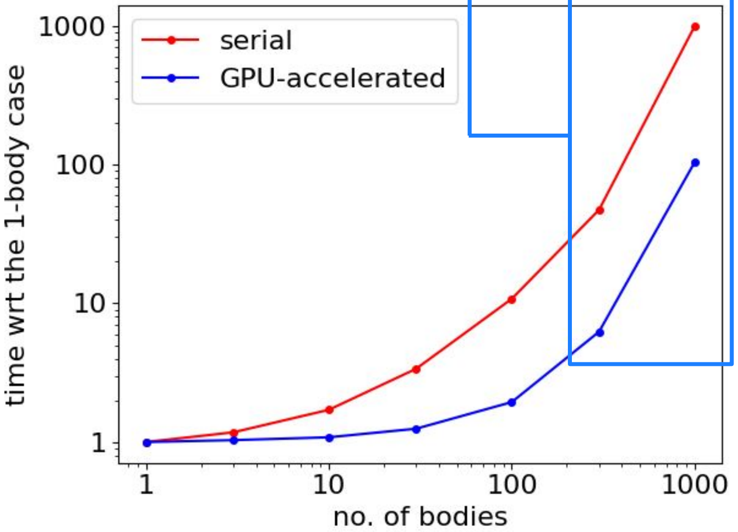
Large overhead keeps the speed-up relative to the serial execution low

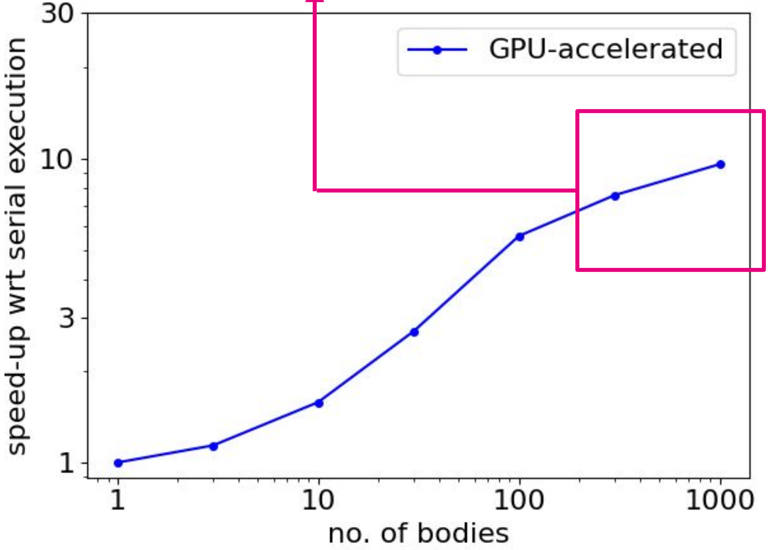


Main results

Close-encounters propto bodies^2: curve slopes becomes similar

Higher proportion of parallel calculations gives an higher edge over serial execution





Targets

M9 (oct. '24)
- analysis of interfaces & capabilities of different OPAL models
- identification of final design strategies for the porting of Mercury-Arxes to GPU

M10 (apr. '25)
- completion of the first part of the OPAL simulation campaign
- full porting of Mercury-Arxes to GPU

M11 (ago. '25)
- completion of the OPAL simulation campaign
- final release of the parallelized Mercury-Arxes

KPIs

M9 (oct. '24)
- initial report on Mercury-Arxes performance
- release of the alpha version of the GPU-parallelized Mercury-Arxes

M10 (apr. '25)
- first release of OPAL atmospheric models (with the current version of the code)
- release of the beta version of the GPU-parallelized Mercury-Arxes

M11 (ago. '25)
- full release of OPAL atmospheric models
- final report on Mercury-Arxes performance

KPIs

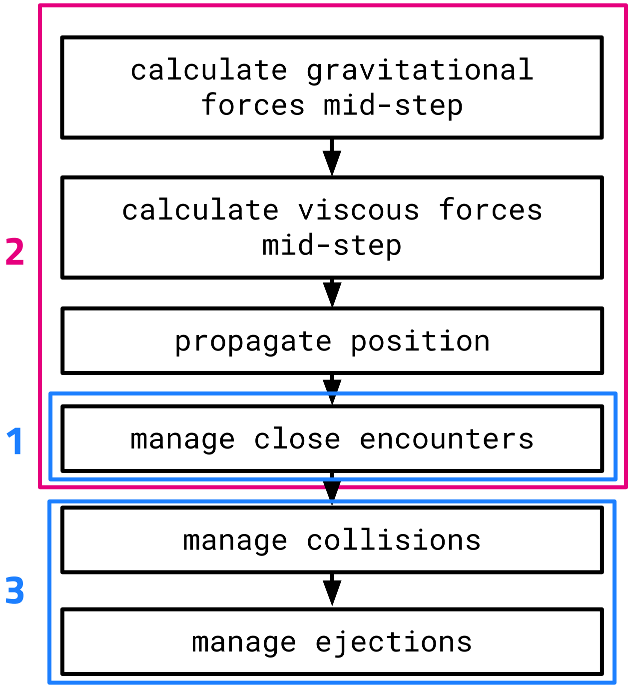
M9 (oct. '24)
- initial report on Mercury-Arxes performance

M10 (apr. '25)
- first release of OPAL atmospheric models (with the current version of the code)

M11 (ago. '25)
- full release of OPAL atmospheric models
- final report on Mercury-Arxes
ance

Rescheduling of Mercury-Arxes targets & KPI due to unforeseen need for code refactoring
75% completion towards full parallelization
10% completion towards OPAL simulations

Final steps



1. try to parallelize close encounter management - trickier than the rest of the loop, will definitely require refactoring!
2. define a data region in order to cut down transfer times btw. host and device
3. parallelize collision and ejection management via OpenMP (minor)

ENDPOINT: comparative performance assessment and deployment of the fully parallelized version of the code