



Contribution ID: 22

Type: **Poster**

## GPU offloads for gravity calculations in the SWIFT cosmology code

Tuesday 4 November 2025 13:20 (5 minutes)

To be compliant with modern heterogeneous HPC systems, large astronomy codes are needing to move towards GPU compatibility. SWIFT (SPH With Inter-dependent Fine-grained Tasking) is a versatile, open-source astronomy code used for a range of research areas in astronomy including galaxy formation, planetary impacts, and cosmology. A significant portion of SWIFT's runtime is dedicated to gravity calculations. In gravity n-body codes, each particle (representing a celestial object) interacts with every other particle based on gravitational forces, making the calculations computationally intensive. However, the repetitive and non-interdependent nature of these n-body interactions makes them ideal candidates for GPU acceleration.

In this work, we build on the existing SWIFT code by replacing specific CPU-based gravity calculation functions with new GPU kernels, minimizing disruption to the rest of the code while preserving the task-based parallelism. This creates a new hybrid C and CUDA version of the code which transfers gravity calculations to the GPU, freeing up the CPU to carry out the other tasks. Our GPU-accelerated gravity kernels achieve high accuracy, with less than 1% deviation from CPU results below the Nyquist frequency. Furthermore, the utilisation of GPUs allows for a redistribution of the gravity calculations making the results more accurate.

Although we currently face a memory transfer bottleneck, optimization efforts using CUDA atomics and streams have shown promising improvements. Future work will focus on eliminating this bottleneck, further integrating GPU offloading into SWIFT's task system, and leveraging additional GPU features to achieve an overall performance boost for the SWIFT code.

### Confirm eligibility

**Author:** JOHNSTON, Sarah (Durham University)

**Session Classification:** Poster Session

**Track Classification:** Poster