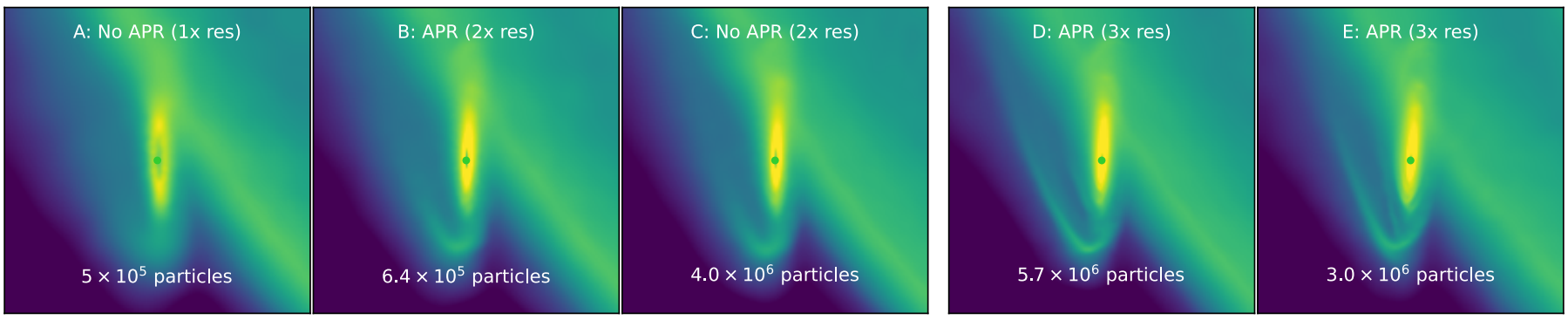
Adaptive particle refinement for compressible smoothed particle hydrodynamics

Rebecca Nealon[1,2] and Daniel Price[3]

Smoothed particle hydrodynamics has the natural advantage that resolution follows mass, but this is not always optimal. We introduce an adaptive particle refinement scheme that allows the user to locally alter the resolution. We test the effectiveness of our method with several problems. Here we show an example of a disc formed around a perturber star from a flyby using Phantom.

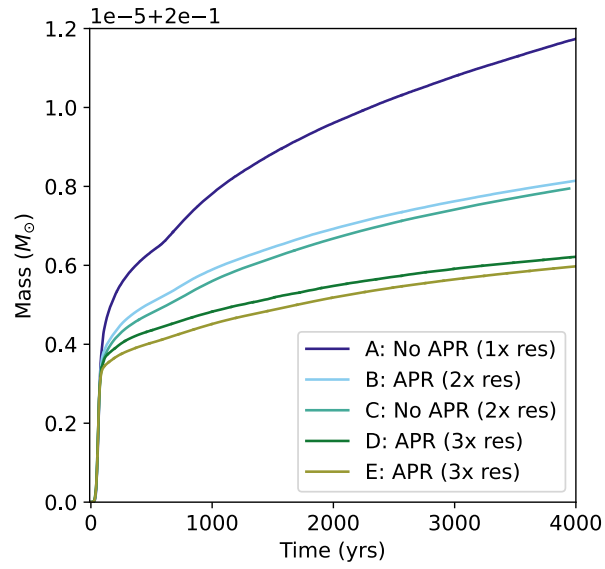
We apply an APR zone around the perturber star, increasing the local resolution of the disc that is formed:



Our APR method (B) shows the same disc structure as the reference calculation with matching global resolution (C) but runs ~6x faster.

The mass of the perturber acts as a proxy for the accuracy of our method, with differences between simulations (B) and (C) of 10^{-4}%.

We can then increase the resolution even more: simulations (D) and (E) have local resolutions equivalent to a global 32 million particle simulation, but only use 5.7 and 3 million particles, respectively.



Our new implementation is …

1. **Accurate**: We measure the mass accreted onto sink particles as a proxy of the disc structure in our simulations, with a difference between the APR and high resolution reference cases of ~10^{-4} - 9% depending on the problem.
2. **Fast**: speed up is problem dependent, offering between 1.07 – 6.62 times faster than a global high resolution simulation.
3. **Uses less storage**: fewer particles means smaller files, using between 15-27% of the original storage.

Our method is also highly **adaptable** with both refinement and derefinement, dynamic zones and tracking on either sink particles or gas particle properties like density.

1: Centre for Exoplanets and Habitability, University of Warwick, Coventry CV47AL, UK. 2: Department of Physics, University of Warwick, Coventry CV47AL, UK. [3] Monash University, Clayton 3800, Australia. Figures made using matplotlib [Hunter 2007, Computer Soc, 9, 3, 90] and sarracen [Harris and Tricco 2023, Journal of Open Source Software, 8(86), 5263, 2023].