



## **Accelerating activity coefficient calculations using multicore platforms, and profiling the energy use resulting from such calculations.**

David Topping (1), Irfan Alibay (1,2), and Michael Bane (3)

(1) University of Manchester, Centre for Atmospheric Science, Manchester, United Kingdom (david.topping@manchester.ac.uk), (2) University of Manchester, School of Pharmacy, Manchester, United Kingdom, (3) High End Compute, Manchester, United Kingdom (<http://highendcompute.co.uk>)

To predict the evolving concentration, chemical composition and ability of aerosol particles to act as cloud droplets, we rely on numerical modeling. Mechanistic models attempt to account for the movement of compounds between the gaseous and condensed phases at a molecular level. This ‘bottom up’ approach is designed to increase our fundamental understanding. However, such models rely on predicting the properties of molecules and subsequent mixtures. For partitioning between the gaseous and condensed phases this includes: saturation vapour pressures; Henry's law coefficients; activity coefficients; diffusion coefficients and reaction rates.

Current gas phase chemical mechanisms predict the existence of potentially millions of individual species. Within a dynamic ensemble model, this can often be used as justification for neglecting computationally expensive process descriptions. Indeed, on whether we can quantify the true sensitivity to uncertainties in molecular properties, even at the single aerosol particle level it has been impossible to embed fully coupled representations of process level knowledge with all possible compounds, typically relying on heavily parameterised descriptions.

Relying on emerging numerical frameworks, and designed for the changing landscape of high-performance computing (HPC), in this study we focus specifically on the ability to capture activity coefficients in liquid solutions using the UNIFAC method. Activity coefficients are often neglected with the largely untested hypothesis that they are simply too computationally expensive to include in dynamic frameworks. We present results demonstrating increased computational efficiency for a range of typical scenarios, including a profiling of the energy use resulting from reliance on such computations. As the landscape of HPC changes, the latter aspect is important to consider in future applications.