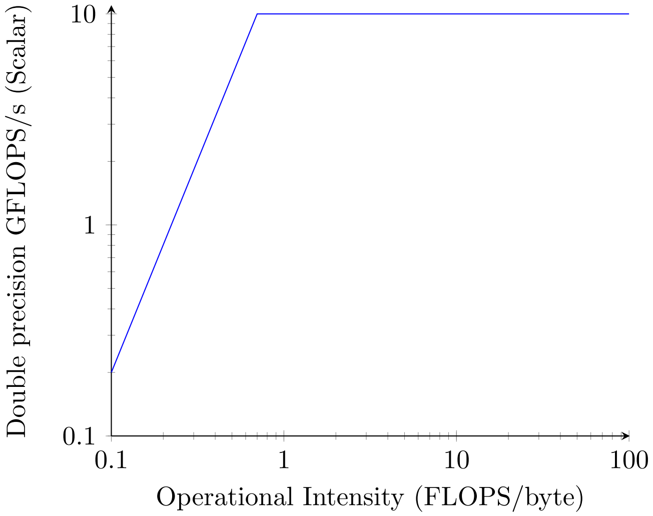
1 Introduction

2 Serial Optimisation

The first step to improving the was to reduce the number of memory fetched which took place. In the original implementation 3 parses took place over the cells dataset which required multiple fetches of the same sections of memory. To reduce this the 4 parses were fuzed into a single loop. The number of copies from `cells` to `tmp_cells` was reduced. This was achieved by using `tmp_cells` as the final answer space for a given timestep (ensuring cells was never changed) and then swapping `tmp_cells` and cells at the end of the timestep.



3 Vecotrisation

The first step to vectorizing the code is to ensure all the array used in timestep are aligned. This can be achieved by using `__mm_malloc` and then adding compiler directives which covey this alignment.