Acceleration of Electronic Structure Codes on Heterogeneous Hardware

1st Marcello Puligheddu
Scientific Computing Department
Science and Technology Facilities Council
Rutherford Appleton Laboratory, Harwell, United Kingdom
marcello.puligheddu@stfc.ac.uk

2nd Ian J. Bush
Scientific Computing Department
Science and Technology Facilities Council
Rutherford Appleton Laboratory, Harwell, United Kingdom
ian.bush@stfc.ac.uk

***Abstract*—We describe the development of a new library, libGint [1], for the calculation of the four-centre, two electron integrals required by local basis set ab initio electronic structure codes such as CP2K [2] and CRYSTAL [3]. The focus of this new work is the acceleration of the calculation via Graphical Processing Units (GPUs), particularly for codes that employ periodic boundary conditions. The core kernels for general contracted gaussian basis set have been rewritten allowing efficient calculation on a GPU. Initial integration into CP2K has been completed and correctness demonstrated. The current challenges involve optimizing the existing code, particularly batching the integrals so they can be calculated effectively on one or more GPUs, and development of a flexible application programming interface.**

***Index Terms*—Graphics processors, Chemistry, Physics**

I. Introduction

Electronic structure codes compute properties of a system from first principles. This gives us insight into physical and chemical properties of molecules and solids and are used for modelling and rational design of materials, e.g. batteries, thermoelectric, solar cells, pharmaceuticals.

These calculations are a large fraction of the computational time on research HPC clusters; in December 2021-January 2022 they accounted for $\approx 40\%$ of Archer2 [4] [5]. As such, there is a real need to make these calculations more affordable. One way is through the computational power provided by GPUs.

As such, we are developing a stand-alone library, **libGint** [1], for the calculation of the four-centre, two electron integrals required by local basis set ab initio electronic structure codes, and an interface layer to CP2K [2] and CRYSTAL [3].

II. Description

The initial aim for libGint is to accelerate calculation of the exact Hartree-Fock Exchange (HFX) contribution to the Kohn-Sham matrix F in CP2K:

$$F_{ik} = -D_{jl}\langle ij|kl\rangle \tag{1}$$

where D is the density matrix and $\langle ij|kl\rangle$ are four-centre, two-electron integrals between Gaussian-type basis functions ϕ:

$$\langle ij|kl\rangle = \int d^3x \int d^3x' \frac{\phi_i(x)\phi_j(x)\phi_k(x')\phi_l(x')}{|x-x'|} \tag{2}$$

Simple (ssss) integrals can be evaluated analytically. Integrals requiring higher angular momenta are iteratively built from the (ssss) terms and their derivative. In the current implementation we use the Obara-Saika [6] method as refined by Head-Gordon and Pople [7]. We apply these to compute the integrals, which we then contract them with the density matrix to obtain the desired result.

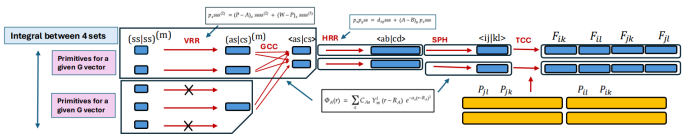

Fig. 1. Schematic picture of the calculation of the 4 centres - 2 electron integral and their contraction with the density matrix P to obtain the HFX contribution to the KS matrix F. The bounding boxes represent one of the possible splits of the computational work between different SM on the same GPU.

The main challenge in the work is the huge yet sparse nature of the computational space. In principle the number of integrals scales as the fourth power of the system size, yet physics dictates that the vast majority are sufficiently small to be neglected. Thus, to efficiently calculate the HFX on GPUs, one must identify the SIMD type parallelism available across sets of related integrals whilst also avoiding the calculation of those that can be screened.

To achieve this, we precompute a plan of execution then refine it to fuse similar operations to reduce the number of steps, precompute memory offsets and array sizes and contract primitives to reduce overall time to solution. Finally, we identify operations that can be done concurrently, minimising the number of synchronisation operations. We prepare this plan on the CPU, then run it on the GPU.

During both the preparatory and computation steps, we account for symmetry within the integrals, periodic boundary conditions, and screening of the integrals. This significantly increases the code's complexity but provides considerable time savings. For example, the use of screening reduces the

computational cost from $O(N^4)$ to $O(N)$, where N is the number of atoms simulated.

A challenge is that each integral takes a small amount of computation; on a modern GPU it can only take nanoseconds for the simplest integrals to microseconds for the more complex. As such how we pack the huge number of required integrals into batches to expose parallelism on the GPU may be just as important as how we compute them.

Finally, we present preliminary results and potential future directions, highlighting the project's ongoing progress and future possibilities.

ACKNOWLEDGMENT

ExCALIBUR [8] is a UK research programme to deliver the next generation of high-performance simulation software for the highest-priority fields in UK research. It started in October 2019 and runs to March 2025, redesigning high priority computer codes and algorithms to meet the demands of advancing technology and UK research. PAX-HPC [9] is a £3M project within ExCALIBUR, to port and optimise high impact particle-based scientific codes to modern HPC architectures, and it is this strand that has funded this work.

REFERENCES

[1] https://github.com/marcelloPuligheddu/libGint
[2] https://www.cp2k.org/
[3] https://www.crystal.unito.it/
[4] https://www.archer2.ac.uk
[5] https://www.archer2.ac.uk/news/2022/02/07/software-usage-data.html
[6] https://doi.org/10.1063/1.450106
[7] https://doi.org/10.1063/1.455553
[8] https://excalibur.ac.uk/
[9] https://excalibur.ac.uk/projects/pax-hpc-particles-at-exascale/