Implementation of high-performance SIMD kernels in *Nektar++*

Supporting: Dr. Giacomo Castiglioni (3 months)
Collaborators: Prof. Spencer J. Sherwin (Aeronautics, Imperial College London), Dr. David Moxey, (Engineering, University of Exeter)

1 Outline

In modern computer architectures, the gap between processor clock speed and memory bandwidth is constantly increasing. In this context, high-order finite element methods are particularly attractive due to their high (and tunable) arithmetic intensity. Matrix-free formulations of operators, on tensor product elements, in combination with single instruction multiple data (SIMD) vectorization is a well studied solution for efficient implementations. Recently the above strategy has been extended by Moxey *et al.* [1] to tensor product simplicial elements on a standalone app, which is partially based on the *Nektar++* library, for the solution of the Helmholtz equation. *Nektar++* is a tensor product based finite element package designed to allow one to construct efficient classical low polynomial order h-type solvers (where h is the size of the finite element)as well as higher p-order piecewise polynomial order solvers. The *Nektar++* library comes with a number of solvers and also allows one to construct a variety of new solvers. The main solvers are a continuous Galerkin incompressible Navier-Stokes solver and a discontinuous Galerkin compressible Navier-Stokes solver. In this project we plan to port the core operators of the above mentioned app as a new *Nektar++* library. The intent is to improve the efficiency of some operators of the *Nektar++* library with the specific end goal of accelerating the compressible flow solver. In particular, we anticipate this work will lay the foundation for operator porting of the *Nektar++* library to GPU. The purpose of this project is to apply techniques to accelerate high-order finite element operators which will not only benefit the *Nektar++* user base, but also both other PRISM partners and the wider academic community.

2 Project objectives

The end goal of this project is to improve the computational efficiency of key operators within the *Nektar++* library with particular focus on the operators needed for the Euler equation solved with a discontinuous Galerkin method.

Porting kernels: As first step towards the end goal, I will port kernels for matrix-free operators which exploit SIMD vectorization from an existing app[1] to *Nektar++* as a new core library. The app was developed to showcase the advantages of matrix-free operators for tensor product simplicial elements in the context of a Helmholtz solver; most of those operators are also needed for a Euler solver. The app was developed from the beginning with the idea of back porting these kernels to the *Nektar++* library, in fact the app utilizes the *Nektar++* library for the construction of basis functions, derivatives, quadrature points, weights, and other ancillary functions.

Developing missing kernels: Not all kernels that are needed for the Euler solver were developed for the app: the second step will consist in developing the missing kernels.

Benchmarking kernels: The third step of this project will consist of benchmarking the newly ported kernels against the currently used ones. The benchmarking will cover 2D and 3D elements such as quadrilaterals and triangles in 2D hexahedrals, prisms, and tetrahedrals in 3D. All elements will be tested in in regular and deformed configurations.

3 Alignment with PRISM strategy

Retention and development of key staff: I am currently the only active senior developer in the *Nektar++* framework and together with Dr. David Moxey I am supervising the developments of the compressible solver. This funding will bridge the gap until the next funding round from Rolls Royce plc starts thus allowing for the retention of a key developer. The developments that will be done in this project will be of particular benefit to the above industrial application where the efficiency of the compressible flow solver is a key component. Additionally this fund will enable me to focus on SIMD vectorization therefore strengthening my programming skill set and academic profile.

Supporting long-term research: This project will allow me to spend time on restructuring and improving the efficiency of the compressible solver in therefore directly benefit the *Nektar++* group and the wider *Nektar++* community. The solver restructuring will also create the foundation for the implementation of asynchronous communication that will enable the overlap of computation and mpi communication. Furthermore this project allows me to gain programming experience and skills on SIMD vectorization that will form the base for my future research efforts on GPUs.

Collaboration with other PRISM members: Dr. Peter Vincent is another PRISM investigator and he is a project leader of PyFR. PyFR is a solver based on flux reconstruction which can be seen as another flavor of high-order finite element methods. The techniques developed in this work could potentially be applied to PyFR.

4 Brief work plan

- Port kernels developed by Moxey *et al.* [1] as new core library in *Nektar++* , specifically kernels for the following operators:
  - `FwdTrans`: performs forwards transformation (polynomial interpolation) onto the modal space;
  - `BwdTrans`: performs backwards transformation onto the physical space;
  - `IProductWRTBase` calculates the L^2 inner product with respect to the basis
- Benchmark newly ported kernels against currently deployed kernels with the compressible flow solver;
- Develop missing kernel for the compressible flow solver for the following operators:
  - `IProductWRTDerivBase` calculates the L^2 inner product with respect to the base derivatives;
- Benchmark newly developed kernel against currently deployed kernels with the compressible flow solver;
- Write and submit a journal publication to disseminate the results to the wider academic community.

References

[1] D. Moxey, R. Amici, and R.M. Kirby. Efficient matrix-free high-order finite element evaluation for simplicial elements. *submitted to SIAM Journal on Scientific Computing*, 2019.