Available online at www.prace-ri.eu

Partnership for Advanced Computing in Europe

SHAPE Project: Renuda

Optimising 2D flow for faster, better steam turbine design

N. Tonello[a], P. Asproulis[a], K. Stratford[b], N. Banglawala[b], P.Graham[b *]

[a]*Renuda, UK Ltd*
[b]*EPCC, University of Edinburgh, UK*

Abstract

This project was a collaboration between Renuda UK Ltd and EPCC, the High Performance Computing centre at the University of Edinburgh. The goal was to improve the performance of Renuda's industrial and power generation steam turbine modelling code, referenced as CodeX for this project. This report outlines the work that has taken place towards this goal, including the technical approach, code analysis and final results.

The code has been investigated, with initial benchmarks and analysis used to identify areas for improvement and to identify the most suitable optimisation approach to take. CodeX has been refactored and restructured to allow for the parallelisation of the code, but also to make improvements in the serial performance. The code has been parallelised using OpenMP directives, ensuring portability across platforms.

Benchmarks performed during the project indicate that for runs of typical operational models, the optimised serial code is over twice as fast as the original. For parallel runs this enhancement is reflected further, with a headline figure of over 27x faster than the original code on 16 cores - this equates to a reduction in runtime from over 1.5 days to less than 90 minutes. The end result of this process is an optimised, parallelised version of CodeX, which can be used to perform simulations in a significantly shorter timescale, thus enabling Renuda to offer enhanced services to their customers.

1. Introduction

This white paper describes a collaborative project between Renuda UK Ltd, an independent engineering company offering software development, consultancy, training and support, specialising in Computational Fluid Dynamics, and EPCC, the UK national High Performance Computing centre at the University of Edinburgh.

Renuda offers several services, including consulting, software development and training and support. Renuda has recognised expertise in using and developing software for a large range of industrial thermal and fluid mechanics applications. Working mostly with blue chips clients in the UK and in Europe, such as Unilever and EDF, Renuda is involved in solving a wide range of problems: from how to simulate ventilation and contaminant dispersion in nuclear plants, the investigation and optimisation of mixing in chemical processing industries to improving the design of gear box oil cooling circuits and steam turbines. As part of the latter, Renuda, in a prior collaboration with EDF, has been developing CodeX, a throughflow steam turbine simulation code for fast optimisation of turbine designs and detailed investigations of turbine operation. Based on its core expertise in fluid mechanics, numerical methods, multi-language and multi-purpose software development of numerical solvers and Graphical User Interfaces (GUIs), as well as knowledge of the application markets, Renuda has been able to develop further capabilities into the software and is now looking to bring CodeX fully to market.

* Corresponding author. *E-mail address*: p.graham@epcc.ed.ac.uk

SHAPE (SME HPC Adoption Programme in Europe) is a pan-European initiative supported by the PRACE (Partnership for Advanced Computing in Europe) project. The aim of SHAPE is to help SMEs overcome barriers to adopting HPC in their business. SHAPE runs regular calls for applications, and successful bids get human effort from a PRACE HPC expert, plus machine time on a PRACE machine. In this case, the role of PRACE expert is taken on by EPCC, the national High Performance Computing centre based at the University of Edinburgh.

CodeX makes it possible to accurately model the performance of large industrial steam turbines with tens of stages. However, in its serial form, CodeX's marketability is restricted by its computing efficiency and unavailability on HPC systems. A parallelised and optimised version of CodeX, would transform the possibilities for the code, by opening market lines for Renuda as well as expanding the scope of application: from sale of software services, to consultancy and support for turbine design. Thus Renuda were interested in parallelising the code in order to speed up the calculations and make it possible to run on significantly denser computational meshes for increased accuracy, and to take advantage of different multi-core computer architectures.

2. Background

CodeX models the performance of industrial steam turbines in two dimensions using first-principles and sophisticated physics, from real gas thermodynamics to dedicated loss models for turbines. As a design and optimisation assist tool used in the power generation industry, CodeX must be robust, support quick decision making, deliver accurate and verifiable results fast, whilst remaining intuitive to use by non-specialists. Although faster than 3D software, CodeX's time-to-solution may be several days when running serially on desktop machines. For CodeX to be widely adopted by industry, a runtime of several hours is required. Validation tests against 3D simulations and experimental data show that, for better accuracy, simulations must run on larger computational meshes than currently possible. This project aimed to develop a parallelised version of CodeX for HPC platforms, reducing its runtime and improving its computational mesh capabilities and solution algorithms. HPC adoption will mean a step change in CodeX's capabilities, enabling further automation and improvement of the turbine design process.

As a throughflow code, CodeX may be used to produce detailed models of both perfect and real gas flows, in planar or axisymmetric geometries. For turbines, in addition to flow data such as velocity, Mach number or pressure fields, CodeX also provides detailed information about the global and stage-wise performance, such as efficiency, power, or the magnitude of the different losses, from leakage to moisture. Bleeds may also be taken into account.

Figure 1: Blades in steam turbine

The workflow to conduct an analysis with CodeX is straightforward and driven by a GUI which has been conceived for turbine engineers and does not require users to be Computational Fluid Dynamics (CFD) specialists. Users input the characteristics of the channel or turbine and, for the latter, specify the different stages, including the position of the rotor and stators, the seal characteristics, and rotational speed. The blades are specified as a series of radial profiles. All data may be input manually or read automatically from data files. Once the geometry has been fully specified, the computational domain is calculated automatically by the GUI and users may then specify the node densities and distribution in the radial and longitudinal directions to create the mesh. Physical and numerical model specification concludes the pre-processing and CodeX may then be run on the case of interest. Once the run is completed, the output may then be analysed from the GUI to create pre-defined plots such as the power of the individual stages, the streamlines through the meridian plane of a turbine, or a Mollier diagram. Figure 2 below shows an example of pressure and Mach number fields in the meridian plane of a turbine.

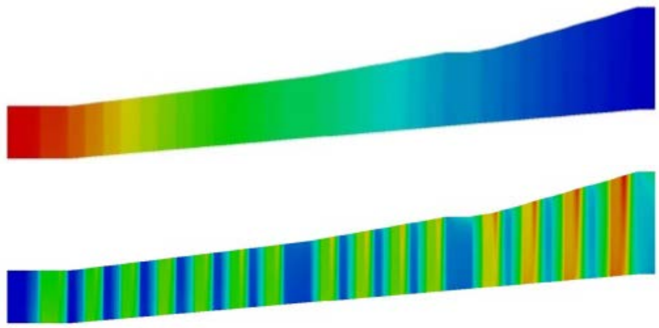

Figure 2: Example of flow fields computed by CodeX for a multi-stage turbine

In order to quantify and improve the computational performance of the code, the project used a set of test cases which are truly representative of the type of industrial modelling that the code is targeting. Three suites of test cases were then chosen, one of a single stage test turbine used as a verification case, and two others which represent industrial steam turbines. The modelling used for these cases was similar to a previous validation and comparison study which compared results from CodeX with results from 3D codes from research and commercial organisations. Tests were performed both with perfect gas and with real gas (steam) settings, the thermodynamics of the latter adding challenges to solution methodologies which are both significant and unusual for Computational Fluid Dynamics solvers.

3. Technical approach

CodeX models and simulates the performance of industrial steam turbines in a two-dimensional domain. Solution is based on a modified Roe scheme for the Euler equations[b] and includes options for sophisticated thermodynamic treatment of compressible non-ideal flow and turbine specific loss models. Fluid-structure interaction allows the inclusion of successive rotor/stator stages between the inlet and the outlet. It is a Fortran code, and each simulation is configured by options set in a provided input file.

For the parallelisation, it was decided to use OpenMP[c]: this would allow an incremental approach to the parallelisation of the code, maintain portability for non-parallel platforms, and was a suitable choice for the target user's hardware e.g. multicore desktops. In addition, it meant that a fundamental rewrite of the code was avoided, as would have most likely been required if a parallelisation paradigm such as MPI[d] was used. The parallelisation of CodeX took the approach of tackling the computation loops deemed to be contributing significantly to the runtime first, then more minor loops were tackled. Many of these loops could be parallelised straightforwardly without modification to the original code other than the addition of the OpenMP directive, but some required additional refactoring to maximise the speedup.

Perhaps the most important option influencing the computational effort needed for a system of given size is the level of approximation used in the thermodynamic expressions required to relate fluid density to pressure and so on. These operations are central to the calculation in the numerical scheme at each mesh point at every time step. CodeX allows a choice of three approaches:

---

[b] J.M. Masella, I. Faille, and T. Gallouêt, On a rough Godunov scheme, Int. J. for Computational Fluid Dynamics, 12 133–150 (1999).

[c] The OpenMP API specification for parallel programming, http://www.openmp.org

[d] The Message Passing Interface, https://en.wikipedia.org/wiki/Message_Passing_Interface

1. Use of “full polynomial” expressions which are based on empirical fits to the true situation for a real, non-ideal, gas (typically steam in a turbine). Some of these expressions are high-order polynomials, while others involve the iterative solution of a transcendental equation (which can be slowly converging). It would be expected that this option might place a relatively high stress on floating point operations.
2. Use of tabulated look ups based on values computed from the full polynomial expressions (often referred to as "steam tables”). There are some eight two-dimensional tables required, each consisting of appropriate values to cover the range of interest in the independent variables in question (e.g., pressure as a function of density and energy). In their standard form, each table consists of 5000x5000 (double precision) entries and so might be expected to place more stress on the memory system than the full polynomial calculation. However, using the tables is significantly faster and they are therefore the method of choice for most production simulations.
3. Use of a perfect, or ideal, gas approximation is available; these are closed expressions and are relatively simple to compute relative to the full polynomial expressions for a real gas. While useful as a reference, perfect gas approximation is generally not appropriate for production runs involving real systems.

Initial profiling of the code looked at these three principal options for thermodynamics calculations. The clear result was that the full polynomial was the most computationally expensive, then the look-up tables, then the perfect gas. As noted earlier, the look-up tables is the method of choice for production simulations, giving close to the accuracy of the full polynomial but in a much shorter time.

Further investigation into the performance of the “steam tables” option revealed that much of the time was spent in a binary search routine to find the appropriate table value for a given point. Inspection quickly suggested there is no parallelism associated with the operation in itself, so the preferred solution was to remove it completely. Fortunately this was possible, and an alternative schema based on direct table mappings was introduced which reduced the runtime by approximately 50%.

In addition to this, analysis revealed that the order of operations on two further parameters (the primitive and conservative fluid quantities) was such that re-ordering of their storage in memory would show benefits. As a further preliminary, some general structural modernisation was completed: the removal of common block storage in favour of module scope storage; the inclusion of all procedures at module level to render interfaces implicit, and the addition of intent specifications for dummy arguments. Much of the project effort was spent in this preparation and refactoring of the code in readiness for the parallelisation.

CodeX writes out various information and data sets to file both during the simulation and on completion of a run. Much of this is written as human readable text, or in a format suitable for third-party visualisation tools. During the project it became apparent that, as the execution time taken for individual timesteps in a simulation was reduced via the optimisation and parallelisation work, the percentage of time spent in performing (sequential) i/o became a more significant factor. To mitigate this, the Fortran "unformatted" output approach was enabled for use for the larger datasets during the run. This allows for a significant improvement in runtime.

4. Results

The system used to run benchmarks was Cirrus[e], EPCC’s compute platform dedicated for industry users. The basic unit of computation is a node, where one node has two 18-core Intel Xeon E5 2695 v4 (Broadwell) processors. Each core runs at 2.1 GHz and supports 2 threads in hardware. Each core can issue 4 double precision floating point multiply-add instructions per cycle (vector length 4 doubles), giving 16.8 Gflop/s per core, or about 0.60 TFlop/s per node. Two sockets have thermal design power rating of 290 W. The Intel (2017) compiler suite was used, with profiles provided by Intel VTUNE.

Parallel timings were obtained for the industrial case “HP1300” simulation: this is for a nuclear power plant high-pressure steam turbine with 15 stages (30 blade rows). The same problem is represented at four different mesh resolutions which will be referred to at Mesh1–4. The lowest mesh resolution, Mesh1, has 961×21 (21,164) points:

[e] https://www.epcc.ed.ac.uk/facilities/demand-computing/cirrus

Mesh2 has 1761×41 (74,004) points: Mesh3 has 2711×61 (168,144) points; and the highest resolution Mesh4 has 3801×81 (311,764) points. Both perfect gas and "steam tables" cases were run. Each run was set for a specified maximum number of iterations (10,000 for mesh1and 2, 20,000 for mesh 3, and 30,000 for mesh 4) or until the solution converged, with file output set to take place once only at the conclusion of the simulation.

Table 1 shows a comparison of the timings for the HP1300 perfect gas example, run on four mesh sizes. For the serial runs, the optimised code is marginally faster than the original for this case. For the parallel runs, the code scales reasonably well, and gives for example a speedup of over 16x faster for the optimised parallel version on 16 cores versus the original serial code for the mesh 4 example. This equates to a reduction in runtime from 16.5 hours to less than an hour.

Table 1: Performance timings for CodeX for Perfect Gas HP1300 test cases

		Original	Modified: 1 core		Modified: 8 core		Modified: 16 core	
Case	Iterations	Time (s)	Time (s)	Speedup v Original	Time (s)	Speedup v Original	Time (s)	Speedup v Original
Mesh1	10000	1142	1030	1.1	189	6.0	139	8.2
Mesh2	10000	4137	3696	1.1	526	7.9	345	12.0
Mesh3	14178	14670	12370	1.2	1768	8.3	973	15.1
Mesh4	30000	59620	48430	1.2	6546	9.1	3544	16.8

Table 2 shows a comparison of the timings for the HP1300 example run on four mesh sizes with the optimised steam tables. For the serial runs, the optimised code is over twice as fast than the original - due principally to the improved steam table interpolation algorithm introduced during the project. For the parallel runs this enhancement is reflected well, with a headline figure of over 27x faster on 16 cores for the mesh 4 example, equating to a reduction in runtime from over 1.5 days to less than 90 minutes.

Table 2: Performance timings for CodeX for Steam Tables HP1300 test cases

		Original	Modified: 1 core		Modified: 8 core		Modified: 16 core	
Case	Iterations	Time (s)	Time (s)	Speedup v Original	Time (s)	Speedup v Original	Time (s)	Speedup v Original
Mesh1	10000	3287	1328	2.5	304	10.8	216	15.2
Mesh2	10000	10850	4597	2.4	713	15.2	493	22.0
Mesh3	20000	49240	22320	2.2	3233	15.2	1919	25.7
Mesh4	30000	134000	60790	2.2	8822	15.2	4940	27.1

5. Conclusion

A new version of CodeX has been developed and validated. Starting from its original, serial implementation, CodeX has been optimised and then parallelised to make it possible to run on multiple threads and take full advantage of modern computer architectures available to turbine engineers. As a result, the speed of calculations has been increased considerably, and in proportion to the number of compute cores available on the systems where the code is run. The efficiency of the parallelisation and resulting solution algorithms has been measured on runs on up to 16 cores, showing close to ideal, linear scaling up. This opens up completely new modelling and commercial applications for the code.

Acknowledgements

The project was supported via SHAPE (SME HPC Adoption Programme in Europe), a pan-European programme that promotes High Performance Computing adoption by SMEs (small and medium sized enterprises), supported as part of the PRACE (Partnership for Advanced Computing in Europe) initiative.

This work was financially supported by the PRACE project funded in part by the EU's Horizon 2020 Research and Innovation programme (2014-2020) under grant agreement 730913. EDF's participation and support during this project is also gratefully acknowledged by Renuda.