Business and Technical Adaptivity in Marine CFD Simulations
Bridging the Gap

Thomas Hildebrandt, NUMECA Ingenieurbüro, Nürnberg/Germany,
thomas.hildebrandt@numeca.de
Matthias Reyer, cpu 24/7 GmbH, Potsdam/Germany, m.reyer@cpu247.com

Abstract

The paper describes progress in software and business solutions for advanced CFD applications. Adaptive grid algorithms are shown to be mature for industry applications, refining and coarsening grids in unsteady flow problems, e.g. for capturing free surfaces around ships, cavitation at propellers, etc. The "NUMECA on Demand" business model allows occasional users renting high-performance computing hardware along with parallel CFD licenses to compute resource-intensive applications. Examples from marine applications and related industries serve as illustrations.

1. Introduction

To adapt [ə`dæpt]: *vt* make (sth) suitable for a new use, need, situation (*Hornby (1974)*).
Adaptivity [ə`dæp`tıvıtı]: the ability to adapt.

Adaptivity is an expression which is meant in several ways and applied to numerous disciplines. In a broader sense, adaptivity is the key to all live. It forms the basis of evolution. Today, it is widely accepted, that all species and organisms are advancing by permanently undergoing natural selection. This selection process usually leaves the choice of failing or passing. If an organism passes selection, it will incorporate the lessons learned, hence leading to a fitter and/or more intelligent organism. The ability to adapt to new situations, changing needs or altering conditions, is crucial for survival. Along these lines, new conditions and needs can also mean new flow conditions in a CFD simulation or changing customer demands when it comes to the commercial use of massively parallel hard- and software. Wherever the challenge for organisms, individuals, organisations or software codes comes from – adaptivity is the key!

In the context of this paper, adaptivity is considered in two meanings and as a reaction to changes coming from two different disciplines: First, a technical or numerical one, and second a business one. Both areas cover techniques, procedures and business models of the most efficient usage of computational resources for CFD applications.

Here, technical (numerical) adaptation means techniques and procedures of a most efficient usage of computational resources for CFD applications. Technical (numerical) adaptivity can be accomplished by adapting the numerical accuracy to the changing conditions of a converging (or non-converging) flow solution. Accuracy is expensive to achieve, let it be the scheme or the grid resolution. Therefore it makes a lot of sense to be as accurate as needed – but not more. In other words, being accurate only where it really matters. Good is good enough. Numerical accuracy is driven by a number of factors of influence. The grid resolution and the numerical scheme are two of these factors which are well suited for adaptivity.

The commercial viewpoint is similar to the numerical one. Only use, provide and pay for what is really needed. The commercial aspect covers an adaptive business model where the user is no longer stuck with the traditional commercial license models. Software licenses are expensive resources, which are purchased or rented scarcely. In times of high workloads maxed out licenses are quite common, while in other times (or even in other departments within the same company) expensive software licenses are idling. The same holds true for the massively parallel hardware needed for high level CFD simulations.

The paper is structured in two parts. The first part covers the CFD flow solver incorporated in FINE™/Marine and the adaptive grid refinement algorithm illustrated on a number of practical examples. The second part describes the commercial adaptivity in terms of intelligent hardware allocation and license management.

2. Adaptivity – The Technical One

General Considerations: Adaptivity always works via two pivot elements. First, there is a driver or trigger, which might be an event, circumstance or change initiating the adaptivity. The trigger is secondly followed by a procedure or scheme reacting on the trigger and adapting the organism, software code or business model to the changed boundary conditions.

How to apply adaptivity to CFD? The global objective of each numerical scheme is to obtain a sufficiently good overall result with the least computational effort. Numerical adaptivity in CFD describes techniques and procedures to adapt the local accuracy and/or resolution of a flow solution only in regions where it matters. In simple words, this means for mesh adaptation that the mesh should only be refined where high flow gradients call for a high mesh resolution. Consequently, the mesh should be coarsened in flow regions without serious gradients. Therefore the most efficient use of a given number of – computationally expensive – mesh cells can be ensured. But how does the CFD code know, how accurate it has to be at which location? It is the gradients that matter. For one, gradients can be spatial gradients, e.g. regions in the flow field where severe changes in the flow quantities take place within a very limited spatial region. Examples for high spatial velocity gradients are the boundary layer around the ship hull or on the propeller blades, where the flow velocity changes from zero on the wall to full speed within a very small distance. Another quite dominant flow gradient in free surface flow is the density gradient at the position of the free surface. Between water and air the density varies approximately by a factor of thousand. Pressure gradients linked to wave patterns are another prominent example of spatial flow gradients. Similar to spatial gradients, temporal gradients exist in unsteady flow fields. Examples are time dependent events such as impact investigations, or sliding grid propeller simulations.

In general, when talking about numerical adaptation in CFD there are two different techniques, which can be employed either separately, combined or in different space and time combinations within the computational domain. The most influencing factors of numerical accuracy are the mesh resolution and quality and the order of the numerical scheme. Consequently, the numerical adaptivity uses these two key factors in order to improve or reduce the local accuracy. In most people's mind, adaptation mainly refers to refinement. But it is worthwhile mentioning that adaptation also means de-refinement in regions or in times where a high resolution is not necessary.

Grid Adaptation: Grid adaptation works by modifying the numerical grid, hence the local resolution of the discretization scheme according to the needs – the gradients of the flow as discussed above. There are two different methods for grid adaptation:

1. Adaptation of the mesh refinement by redistribution, also called r-adaptation (Fig.1). Here, the mesh distribution is modified in accordance with the flow gradients. The number of mesh nodes as well as the mesh topology and the connectivity remain unchanged. The redistribution is achieved by moving grid points or lines around, following the flow gradients. This method is in general somewhat limited, since a refinement in one area may result in excessive coarsening in another. An advantage is that r-adaptation can in principle be applied for structured grids also, while most other methods only work for unstructured grids. Grid quality however, might be difficult to control or maintain.
2. H-adaptation adds or removes mesh points. The overall number of nodes usually (although not mandatorily) changes. The node distribution and the mesh topology are altered in any case. Various strategies exist for h-adaptation. Simple procedures subdivide cells, more complex strategies insert or remove or cells (Fig.2). In general, h-adaptation only works on an un-structured level, let it be the overall mesh, or – in case of structured meshes – the block arrangement.

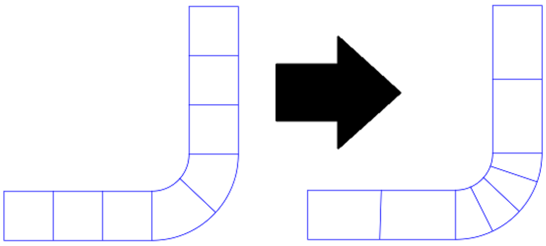

Fig.1: Example of simple r-adaptation, *CFD Online (2012)*

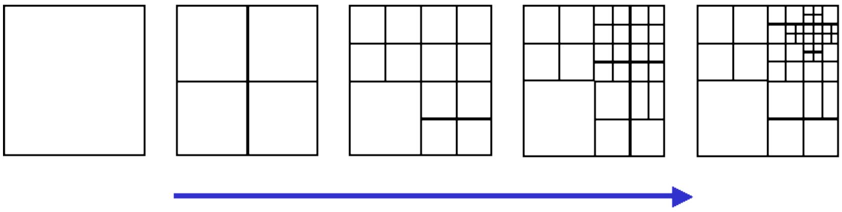

Fig 2: Example of h-adaptation, *MEVIS RESEARCH*

Scheme Adaptation: Adapting the resolution of the numerical scheme to the flow gradients can also be achieved by changing the polynomial order of the numerical scheme. Most CFD solvers available today are (theoretically) second order in space. However, hardly anyone reads the small print in the ads, saying that this only holds true for smoothly varying grids. And most practical grids are definitely not smoothly varying. The adaptation of the numerical scheme is also called p-adaptation. When p-adaptation kicks in, the resolution of the numerical scheme is increased from second order to higher orders. Consequently scheme "coarsening" can be accomplished by reducing the order to one.

Combined Adaptation: Mesh and scheme adaptation procedures can also be combined. These techniques are then called e.g. hp-adaptation.

2.1. The CFD System FINE™/Marine

All applications of technical adaptivity presented in the frame of this paper are performed using FINE™/Marine, a marine specific CFD system by NUMECA International S.A. FINE™/Marine is a complete CFD tool chain described for example in *Visonneau et al. (2012)* and incorporates the following modules:

1. **The mesh generator HEXPRESS™:** A full hexahedral unstructured mesh generator, which is capable of solver driven mesh refinement and coarsening (h-adaptation). It features body fitted meshes with a high quality boundary layer resolution. Grid refinement and coarsening, during the initial generation process as well as in later adaptations is achieved by means of hanging nodes (Fig.3).

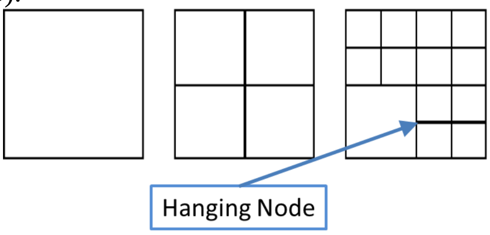

Fig 3: Hanging Node, *MEVIS RESEARCH*

An example of a HEXPRESS™ mesh of ship hull appendages system is shown in Fig.4.

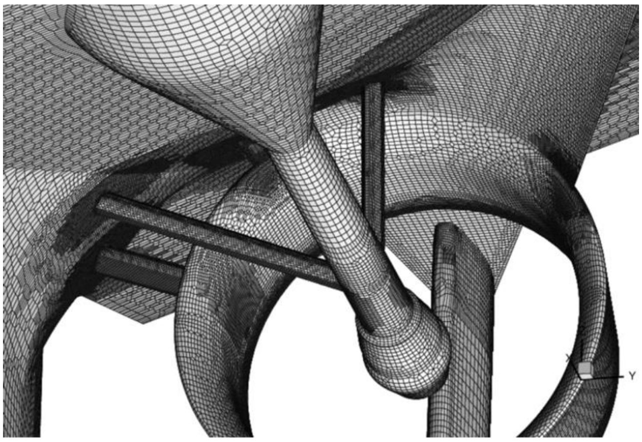

Fig.4: HEXPRESS™ mesh

2. **The flow solver ISIS:** The flow solver inside FINE™/Marine is a steady and unsteady incompressible free surface RANS-Code (Reynolds-Averaged-Navier-Stokes) presented in detail by *Duvigneau et al. (2003)*, and *Queutey and Visonneau (2007)*. The spatial discretisation of the transport equations is accomplished by a finite volume method. The velocity field is obtained from the momentum conservation equations and the pressure field is extracted from the mass conservation constraint, or continuity equation, transformed into a pressure equation, *Schrooyen et al. (2014)*. Pressure-velocity coupling is obtained through a Rhie & Chow SIMPLE type method. No specific requirements for the topology of the cells are imposed. The grid can be completely unstructured and cells with an arbitrary number of arbitrarily-shaped faces are accepted. In the usual case of turbulent flows, additional transport equations for turbulent entities are introduced. Several turbulence models ranging from relatively simple one-equation Spalart-Almaras to advanced EARSM (Extended Algebraic Reynolds Stress) Models *((Duvigneau et al. (2003))* are implemented. Free-surface flow is represented by a VOF (Volume of Fluid) technique with an interface capturing approach. Both non-miscible flow phases (air and water) are modelled through the use of a conservation equation for a volume fraction of phase. The free-surface location corresponds to the iso-surface with a volume fraction of 0.5. To avoid smearing of the interface, the volume fraction transport equations are discretized with a specific discretization scheme, which ensures the accuracy and sharpness of the interface, *Queutey and Visonneau (2007)*. Furthermore, the flow solver features 6 DOF motion for the simulation of freely moving ships, *Leroyer and Visonneau (2005)*. Parallelisation is based on domain decomposition.

3. **The flow visualisation system CFView™** also incorporates marine specific plug-ins. The visualisation of characteristic features such as wave patterns, the free surface, wetted surface as well as the calculation of forces, momentum and angles is done by a mouse-click.

2.2. Adaptive Grid Refinement

FINE™/Marine incorporates adaptive grid refinement (AGR). When a flow simulation with adaptive grid refinement is launched, the refinement procedure is called every *n* time steps in order to keep the grid adapted to the evolving flow solution. Usually, the flow solver is first run on the initial mesh for a given number of times steps, after which the adaptation algorithm is activated. The existing flow solution is then evaluated and in case one or several adaptation criteria indicate the mesh is too coarse at certain locations, the cells in question are then refined, or cut. The flow solution of the previous step is then interpolated on the refined (=adapted) grid and the flow solver continues for a given number of iterations. Thereafter, the adaptation procedure is called again, and the adaptation criteria are applied. In addition to the first step, all further adaptation steps do not only have the option to refine, but also to de-refine, meaning that earlier refinement steps can be undone. This cycle is then repeated a number of user defined time steps. This technique is described in more detail for example in *Wackers et al. (2010a), Wackers et al. (2010b), Wackers et al. (2011)*. It is designed with a broad range of applications in mind and written in an as general way as possible, *Visonneau et al. (2012).* To ensure an equal processor load even with adapting meshes, the newly created cells are distributed automatically between the partitions by the flow solver. Hence the total number of cells on each processor is comparable and an efficient usage of all processors is achieved. Several refinement criteria are available and can be selected separately, combined or in succession of each other according to the task at hand. Examples are:

- **Free surface criterion (Fig.5):** This criterion refines close to the free surface. Since the free surface is clearly characterised by the gradient of the volume fraction normal to the surface, the refinement is employed to refine the grid in the direction normal to the surface only. In large regions of the flow domain this directional (= anisotropic) refinement will be applied in order to keep the number of additional grid points as low as possible. The resulting zone of directional refinement includes the undisturbed water surface, as well as smooth wave patterns. Only in cases where the free surface seriously deviates from the main grid directions, such as breaking waves, isotropic refinement is used (Wackers et al. (2010a)).

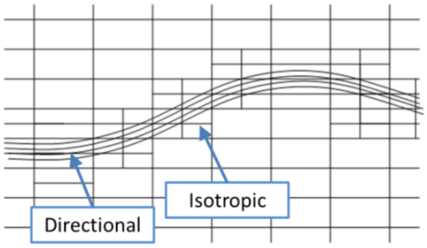

Fig.5: Directional and isotropic refinement at the free surface (*Wackers et al., 2010a*).

- **Gradient criteria**: A second group of refinement criteria is based on the absolute values of the gradients of solution quantities in each cell. These criteria detect the regions where the flow field changes rapidly; they react to most features of a flow and are thus more general than the free-surface criterion. Also, they are obviously not restricted to the vicinity of the free surface and can refine in the whole computational domain. Three gradient criteria are available in FINE™/Marine.
    1. **Pressure gradient**
    2. **Velocity gradient**
    3. **Vorticity gradient**
- **Hessian based criteria**: This criterion works with the second spatial derivative of the pressure. It is a very robust refinement criterion, yielding good results for a variety of applications. In contrast to velocity based gradients, it does not introduce any unnecessary refinement into the already refined boundary layer, *Wackers et al. (2014).*

The effect of mesh adaptation is nicely visible in Fig.6 showing an impacting cone probe, the splashing scheme and the adapted mesh.

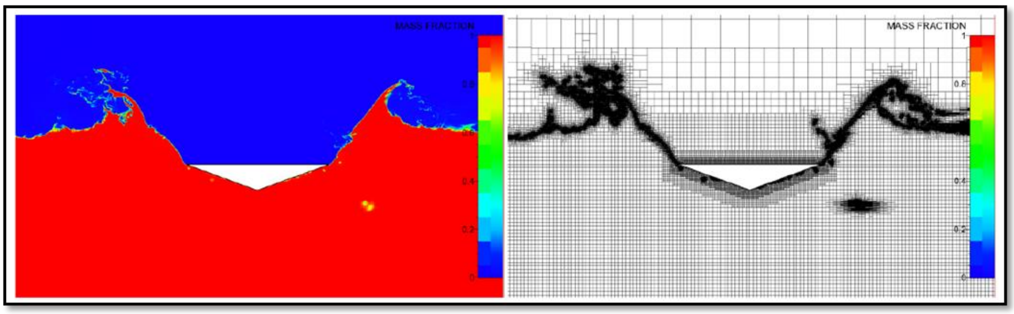

Fig.6: Impacting probe and adapted mesh

The efficiency of adaptive mesh refinement is highlighted in Fig.7. In this case, the adapted mesh has only about 40% more cells than the initial mesh, still the surface wave pattern is considerably more detailed showing features which do not appear in the original mesh.

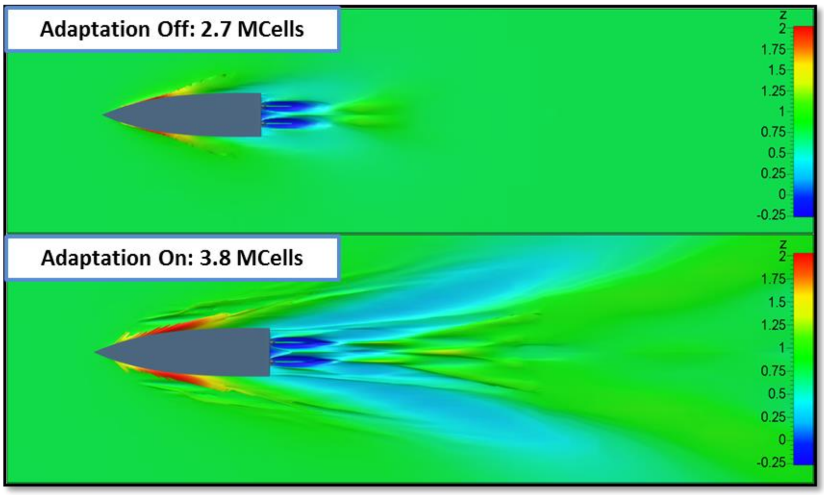

Fig.7: Wave pattern without (top) and with (bottom) mesh adaptation

Numerical experiments have shown, that similar or even better resolution of flow features can be obtained on properly adapted grids, having not even a fifth of the mesh count compared to a fine mesh without adaptation.

2.3. HPC – and Parallelisation

High level CFD simulations in the maritime field are challenging due to numerous computationally very expensive features. Examples are:

- 6 Degree of Freedom (DOF) simulation for arbitrary movements of multiple bodies.
- Wave breaking and splashing.
- Full unsteady simulations and sliding grid computations of rotating propellers.
- Acceleration and dynamic positioning.
- Seakeeping and self-propulsion.
- Optimisation.

It is very obvious that these tasks, despite all the artful programming of intelligent features which increase the efficiency of computations, call for extreme computing power. Even relatively simple resistance calculations – the bread and butter business of maritime CFD – should be finished in less than one hour. At least, this is what is on the naval architect's wish list. So, High Performance Computing (HPC), and along with it, parallelisation is key.

Full (or more realistically nearly full) and automatic parallelisation is therefore a highly important property of a CFD code. But parallelisation of adaptive grids is not obvious. In non-adaptive calculations, usually a domain-decomposition is performed before the computation is initialised. The decomposed regions (partitions) of the grid are then distributed over the available cores, which may even have different individual performances. The number and size of the decomposed regions depend on the number and performance of the computing nodes. Each domain is then put on one core. The communication is ensured by parallel libraries, which in the case of FINE™/Marine is MPI (Message Passing Interface). When it comes to adaptive grids, the initially decomposed regions will not remain constant in size. The refinement (or coarsening) algorithm will alter the number of nodes in one partition. Keeping the original decomposition constant, would mean a severe degradation of computational performance due to load imbalance between the processors. Image one region having five times the number of nodes would still run on the same type or number of cores as smaller regions. So, a new decomposition and redistribution over the parallel hardware will become necessary after each adaptation step. In FINE™/Marine the grid refinement is capable of dealing with massively parallel hardware. It includes an automatic dynamic load balancing which redistributes the refined grid over the already allocated cores when some regions have been refined or coarsened.

3. Adaptivity – The Business Perspective

3.1. General Considerations on Speeding Up CFD Simulations

CFD is certainly one of the CAE disciplines which require the highest level of numerical effort. This holds true despite of all the intellectual effort put into the coding of highly efficient numerical schemes, such as adaptive grids, pseudo time stepping, predictor-corrector algorithms and the more. Therefore high hardware requirements are quite common in the maritime CFD environment as shown in section 2.3. It is in this region of the highest hardware demands where Moore's law, predicting a doubling of processor performance every 18 months started to lose momentum around 2003. The loophole out of this dilemma is parallelisation. The computational problem is no longer tackled by one single processor, but by many instead. Many can mean several hundreds or even thousands. In exceptional cases Large Eddy Simulations (LES) of a towing tank model ship on 200.000 cores have been successfully demonstrated. (*Nishikawa et al., 2013*)

Parallelisation, also called High Performance Computing (HPC), is based on two pillars:

1. A hardware architecture providing a sufficient number of processors.
2. Software which is capable of running efficiently on these many processors.

In this context efficient means scalable. No software ever will run ten times as fast on ten processors as it does on one, although this should be the objective. In this case one would already talk of good scalability, if it would run something like eight times faster than on a single processor, ideally maintaining this ratio independently of an increasing number of cores. One of the reasons for this non-perfect speed-up is that some parts of the software cannot be parallelised while other parts need to do the communication between the processors. By using Amdahl's law (*Amdahl, 1967*), it can easily be demonstrated that even small portions of the software code, which are not running in parallel, will lead to a seriously limited maximum speed-up, which will be asymptotically reached with increasing processor number, but never exceeded (Fig.8).

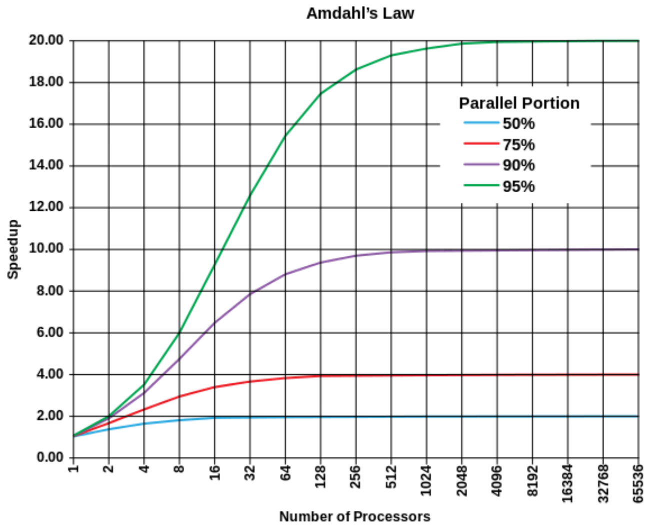

Fig.8: Amdahl's Law – Maximum speed-up as function the parallel portion of a software code

To makes things worse, usually the scalability will go further down when the processor count increases, due to the decreasing volume-to-surface ratio of the single partitions running on each core.

More cores mean smaller partitions. The volume of the partitions (= problem size) drops by one over the power of three, while the surface (= communication size) only goes down by one over square.

In any case, if future CFD requirements are to be met, users will find themselves in a situation where they have to invest in large hardware and, in case of commercial CFD codes, also in licenses. How can be guaranteed that these investments will pay off?

3.2. Adaptivity as a Business Model

As for the numerical adaptivity, the business adaptivity works in a very similar way. Again, there is a driver or trigger, which initiates the process of adaptation in order to match the resources to the expected outcome. When the trigger kicks in, a procedure or scheme reacts on the trigger and adapts the business model to the changed boundary conditions. Now, what is the trigger, or changed boundary conditions and how does the adaptation mechanism look like?

Nowadays the use of especially powerful, but also very costly computational resources in industry and academics is standard. In a traditional business model these resources will be bought once, are completely to be maintained during their useful life span and disposed of thereafter. Apart from the costly disposal, the business model for licenses of commercial software is not much different. This traditional business model makes perfect sense, if a constantly high usage of both, licenses and hardware, can be ensured. But if this is not the case, maintaining huge computers and a large license pool becomes uneconomic. Buying computer hardware and licenses will also tie up huge amounts of capital and human resources. Again, as in numerical problems, adaptivity is the solution. The hardware as well as the licenses should be provided only in times when they are needed. Although this option seems obvious and not very new at first it has hardly been applied before.

What do you do with a 2000 core cluster when you do not need it? Shutting it down and mothballing it is probably not an option, if you might need it unexpectedly tomorrow, when model tests have shown that you need to optimise your hull design. Also, the 2000 cores and other hardware architecture are perishable goods. They have a “best before” date stamped on them. So, better use them now. Also scalability is an issue. You may use 2000 cores occasionally, but only 200 permanently.

In terms of software things are quite similar. You cannot stop paying for annual software licenses just because your CFD team goes to summer holidays. On the opposite side, shortly before the year ends there is usually a peak load, where you might need three times the normal license volume. Usage or non-usage is clearly the trigger of business adaptation. Now, how to realise it?

One viable avenue is to employ an external provider. The self-evident question then is, why should an external provider be capable of dealing with fluctuating demands better than oneself? The high volume of these requests with which an external provider has to deal is the key. One strongly fluctuating demand might be a problem for a company, but hundred fluctuating request are standing a good chance of leading to a solid average load with only small seasonal deviations. One single company (or individual, or academic institute) may need 2000 cores for a couple of weeks, which are then on idle for next few months, ready to be used by another client. An external provider who deals with several dozens or hundreds of clients then profits from an averaging effect. It is statistically highly unlikely that all his clients have their up- and downtimes at the same time. While one company’s cores might go idle, they will soon be picked up by another one, who has a peak demand. Clearly, the provider will need a certain overhead which increases the cost slightly, but in total it will be less expensive for his clients than maintaining their own hardware which runs with a much smaller load factor than the provider’s. What has been said for the hardware works out just the same for the software. Ideally, hard- and software will be coupled. They are only employed together and there is no need to have a 2000 parallel license sitting around, when your cluster only features 200 cores.

Key issues of this concept are of course security and availability on very high levels, but this can be accomplished. Equally important are high computational performance and a transparent scheduling and workload sharing. CPU 24/7 is one such provider who offers hard- and software as a bundle. Due to many years of experience and expertise, CPU 24/7 is very conscious about the special requirements of on-demand licensed computing power resources (HPC On Demand), in particular for CAE applications for industrial or high level academic purposes. In parallel to the computer hardware, an intelligent and adaptive license management, which enables a tailored license usage, is desired. The so-called *"NUMECA On Demand"* concept (Fig.9) combines arbitrarily scalable HPC resources with an on-demand license management for FINE™/Marine.

3.3. Putting Adaptive HPC Resources into Practice

Since 2006 CPU 24/7 has been offering on-demand computing power resources and provides custom-made and dedicated HPC systems and capabilities with a completely pre-configured, secure and easy-to-use remote simulation environment. This is realised as an all-inclusive package taking in the HPC instance on bare metal servers with related CAE applications such as FINE™/Marine. A competent hot-line technical support, sufficient and flexible storage and broad band data traffic are the ingredients to make this concept viable for CAE users.

HPC On Demand has become primarily of interest to those companies, struggling to cope with budget restrictions and a fluctuating demand for computing and licensing capacity. Investing in their own HPC clusters is very cost-intensive, binds valuable capital and expertise as well as requires the services of staff whose skills and capabilities are not central to a company's core business activities. For such investments in order to remain profitable, it is essential that all available capacity is constantly being utilised. If this is not the case, renting the access to High Performance Computing capacity instead of binding one's capital in a complex IT infrastructure and a large license pool might be the smarter option.

It is the temporary, dynamic aspect that makes HPC On Demand so attractive. One advantage of acquiring access to HPC resources for a specified period, for example for a particular project, is that it involves paying only for the capacity and for the duration of the computing services that one actually requires. HPC remote resources are available either as flexibly and online bookable (Fig.9) computing capacities (Resource Area) or also as continuous state-of-the-art HPC systems geared to one's individual needs (Tailored Configurations). In both cases, it is possible to handle the resources due to a cluster management system respectively a job queueing system. This ensures an efficient use considering a dynamic change of hardware requests like number of cores, main memory etc. due to an adaptive resource management. An important aspect is the right setting of several system parameters in the queuing system which is a service from the resource provider. This basic concept fulfils the requirements regarding dynamically changing job characteristics as for example during the adaptation steps of a CFD code.

Fig.9: *"NUMECA on Demand"* @ CPU 24/7

3.4. License Management

The HPC power is just one part of the equation which is necessary for up to date numerical simulations. The second pillar is the CAE software, which is usually licensed, let it be a commercial, in-house, or even your personal version of an open source code. The standard business model during the last years was fixed license contracts over a period of several months or years. Such a static license model does not follow a dynamic and "On-Demand" use of HPC resources and can also hinder the handling of adaptive simulation procedures. Imagine your license features 64 parallel cores, which are fully used at the initialisation of a CFD simulation. Soon, the first adaptation step kicks in and calls for a refinement of certain regions. The job is then automatically re-launched, starting with a fresh decomposition, which would be optimal only for a higher core count, if – your parallel license feature would not be exhausted!

As a response, NUMECA now provides a dynamic and adaptive licensing model which enables the user to get the licenses he needs spontaneously and tailored to the real demand. If more features, such as parallel cores are needed, they will be automatically provided, respecting a user-defined upper (or lower) threshold.

3.5. Commercial Considerations

Also commercially some fresh thinking is needed. The traditional steps of a commercial sales process starting with a quotation, followed by a purchase order, the delivery and finally an invoice do not fit anymore to a dynamic allocation of hard- and software. Instead, the user books the required resources of hard- and software on-line, at completely transparent costs. The entire allocation, technically as well as commercially is completed with a couple of mouse-clicks within 10 minutes (Fig.10).

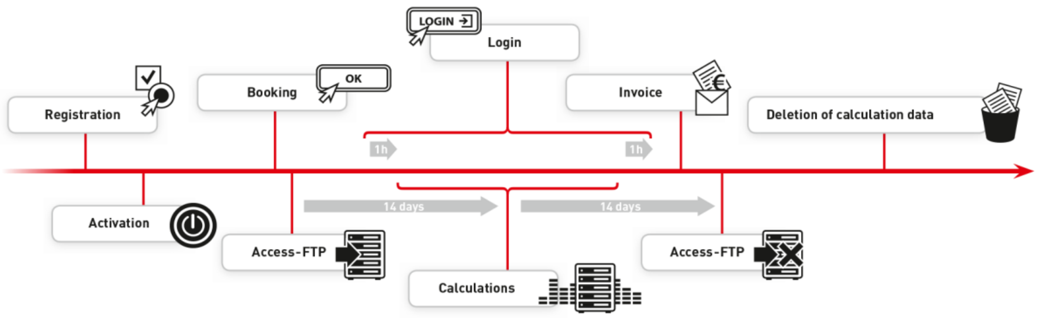

Fig.10: Time sequence of HPC allocation

4. Conclusion

The paper describes the benefits of adaptation for maritime CFD applications. In particular the mesh h-adaptation driven by various gradient based criteria is discussed. A showcase demonstrates that with only a moderate increase in mesh cell count, a substantial improvement in the quality of the results can be achieved. However, adaptivity can not only be applied to numerical problems, but also is a viable concept if utilized in a business perspective. Hardware as well as software licenses are traditionally purchased in a static business model, which is non-reactive to a fluctuating demand. The "*NUMECA on Demand*" approach offers a flexible and demand-driven way to allocate computational resources in a tailored way, according to the task at hand. The challenges of aligning adaptivity concepts, the numerical and the business ones, have been addressed. In combination, numerical and business adaptivity offer the potential to achieve the high requirements of full featured maritime CFD at a fraction of the cost of the traditional approach.

5. Acknowledements

The authors are very grateful to Prof. Dr. Volker Bertram, DNV-GL, who constantly motivated and challenged the NUMECA and CPU 24/7 teams to pursue the goal of this work.

References

AMDAHL, G.M.; (1967) *Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities*, *AFIPS Conference Proceedings* (30), pp. 483–485

CFD Online (2012), http://www.cfd-online.com/Wiki/Mesh_adaptation

DUVIGNEAU, R.; VISONNEAU, M.; DENG, G.B. (2003), *On the role played by turbulence closures in hull shape optimization at model and full scale*, J. Marine Science and Technology, 8/1, pp.1–25

HORNBY, A.S.; (1974) *Oxford Advanced Learners Dictionary of Current English*, Oxford University Press, p.10

LEROYER, A.; VISONNEAU, M. (2005), *Numerical methods for RANSE simulations of a self-propelled fish-like body*, J. Fluid & Structures, 20/3, pp.975–991

MEVIS RESEARCH, http://www.mevis-research.de/~tp/LargeScaleFE

NISHIKAWA, T.; YAMADE, Y.; KATO, C. (2013), *Fully Resolved Large Eddy Simulation as an Alternative to Towing Tank Resistance Tests – 32 Billion Cells Computation on K computer*, In Proceedings of the 16th Numerical Towing Tank Symposium (NuTTS '13), Mülheim, Germany

QUEUTEY, P.; VISONNEAU, M. (2007), *An interface capturing method for free-surface hydrodynamic flows*, Computers & Fluids, 36/9, pp.1481–1510

SCHROOYEN, I.; RANDLE, K.; HERRY, B.; MALLOL, B. (2014), *A Methodology for quick and accurate CFD calculations of self-propelled ships using sliding grids and dynamic RPM control* Conf. Computer and IT Appl. Maritime Ind., Redworth, UK

VISONNEAU, M; QUEUTEY, P; DENG, G.B; WACKERS, J; GUILMINEAU, E; LEROYER, A; MALLOL, B, (2012), *Computation of Free-Surface Viscous Flows around Self-propelled Ships with the Help of Sliding Grids*, Conf. Computer and IT Appl. Maritime Ind., Liege http://www.numeca.com/sites/numeca/files/computation_of_free_surface_viscous_flows_around_self-propeller_ships_with_the_help_of_sliding_grids_-_compit_2012.pdf

WACKERS, J.; AIT SAID, K.; DENG, G.B.; MIZINE, I.; QUEUTEY, P.; VISONNEAU, M. (2010a), *Adaptive grid refinement applied to RANS ship flow computation*, In Proceedings of the 28th ONR Workshop on Naval Hydrodynamics, Pasadena, California

WACKERS, J.; DENG, G.B.; VISONNEAU, M. (2010b), *Tensor-based grid refinement criteria for ship flow simulation*, In Proceedings of the 12th Numerical Towing Tank Symposium (NuTTS '10), Duisburg, Germany

WACKERS, J.; DENG, G.B.; VISONNEAU, M. (2011), *Combined tensor-based refinement criteria for anisotropic mesh adaptation in ship wave simulation*, In Proceedings of ADMOS 2011, Paris, France

WACKERS, J.; DENG, G.B.; GUILMINEAU, E.; LEROYER, A.; QUEUTEY, P.; VISONNEAU, M. (2014), *Combined refinement criteria for anisotropic grid refinement in free-surface flow simulation*, Computers & Fluids 92 pp. 209–222