The Naval Hydro Pack at Wikki

To The Partners of The Naval Hydro Pack:

It's been a very interesting and exciting year for the Naval Hydro Pack with some quite advanced functionalities being developed. In this letter, I intend to summarize the current status, latest developments and ongoing projects in order to share what we have learned with you, our Partners.

Before I start explaining the useful things related to CFD and what we have improved and learned during the past year, I think it is instructive to explain the difference between our *Partners* and *Clients*. Before I go on, I need to explain our business model.

Almost everything we do can be divided into three parts:

1. Specific **R&D projects** for our *Partners*. Basically, we develop functionality that our *Partners* need in order to perform reliable CFD simulations.
2. **Support contracts**, where we provide *"unlimited"* support along with the source code to our *Partners* (just in case someone missed it: the Naval Hydro Pack is based on `foam-extend`, which is a community driven fork of the `OpenFOAM` software).
3. Performing specific **CFD simulations** for our *Clients*.

You might have already guessed the difference between our *Partners* and our *Clients*. The reason for this distinction is rather simple: it often happens that we *learn when working intimately with our Partners*, while we provide solid knowledge and vast experience to our *Clients*, usually working intimately with them as well and aiding them in their design process. Based on our experience, here are the two common traits of our *Partners*:

1. He/she is often highly experienced and knowledgable of the physics they are after,
2. He/she has strong knowledge in CFD and is not scared to go and look through the code and challenge us if something *doesn't look right*.

A question: would you rather work with people who constantly challenge you or with someone who just goes along with the inertia? Now you hopefully understand why we refer to you as *Partners* instead of *Clients* and why we enjoy working with you.

This business model has a lot of advantages, but also some serious limitations. Let's first get the limitations out of the way. The model based on providing *"unlimited"* support is rather unique. You might have noticed that *"unlimited"* is surrounded by quotation marks, mainly because we also like spending time with our families. But rest assured, if the code we develop for you doesn't do what you would expect it to: we will improve it, regardless of how long it takes. I think that this model is *more than fair* to you, while forcing us to stay ahead of the game and constantly improve. Moreover, it is unreasonable to think that we know everything about every aspect of CFD and physics, although I sometimes think that my boss, Hrvoje Jasak, is fairly close. My assumption is that because of the *"unlimited"* support, our business model will not be copied by others. In that sense, we sleep well at night.

Why in the world do you need *"unlimited"* support? Well, it is no secret that `OpenFOAM` is thought of as *hard and challenging to learn and use*, mostly because it does not have a GUI and you can control *almost everything*. We currently do not have a GUI and we don't have any *strict* plans to develop one in the near future. Although, if our Partners start insisting on it, we will probably invest in it in future. If I had to take a bet, I don't think that our existing Partners will insist on a GUI. But then again, I've been wrong before. Regarding the control, some people feel more comfortable with less control and more default options and some prefer having full control. We provide both options to our Partners, based on *how much they are willing to learn*. I will end this discussion with a comparison: I view our Partners more like experienced pilots and less like ordinary car–drivers. Now, let's look at the difference between us and our competitors for a moment.

One of the differences between us and our competitors is that we will *never* try to sell you something that we feel doesn't make sense to you. I know it sounds corny, but I truly believe that win–win situations are a foundation for long–term business relationships, the kind we are interested in. When you look at the major marine hydrodynamics software providers: they have a strict distinction between their *development*, *support* and *sales* teams. The sales–person is operating under a significant incentive to sell you the licence, because he/she gets a significant cut out of each sale: regardless how happy you are with the software. I would assume that the clear distinction between developer, support and sales teams makes them slow to respond to some problems. At Wikki, we are all developers and support engineers, making us significantly faster to respond. We also don't have a sales team. Actually, this letter is the highlight of our marketing strategy so far, and it will probably remain so. Because of this distinction between us and our competitors, my good friend and colleague, Inno Gatin thinks that they are not actually our competitors since we target a completely different type of CFD engineers (pilots vs. drivers). I came to agree with him during the last several years that we have been working together.

Another significant benefit for our Partners (and a drawback for us from a business point–of–view) is that we do not have *software licences*, which means that if you want to run dozens of jobs at the same time you will pay the same price as you would pay if you were running a single job. No commercial software can compete with `OpenFOAM` on that front. Now, imagine that you need to run 10 seakeeping simulations in a day side–by–side as one of our Partners does. Would you rather invest in 10 licences that will be used by a single CFD engineer or a single support contract, training and education for several of your CFD engineers? To me, that's a no–brainer. But we live in the real world and I assume that if you have an engineer that has been working with GUI and less control for years, he/she is less likely to accept the change. If you are a principal CFD engineer in charge of software procurement in the company and working with people that don't mind challenges and like learning: You should carefully consider working with us and if it makes sense for you, go to your boss with this wonderful plan on how to save some money and add value to the business. And don't forget to demand a raise: you deserved it.

The serious drawback of our model from a business point of view is that it *does not scale*. Our response: we don't care. I work with wonderful people and wonderful Partners on challenging and interesting projects and I would be happy to continue like this until retirement. Long–term, I think that it is far more rewarding to invest in building profound relationships and mutual trust between people you like (both in personal and professional world) than to try and scale–up the business in order to make more money by *networking* (which is usually measured by how many business cards you exchange on an event). In fact, although I like the free market and capitalism, I don't think that they require you to *earn as much money as you can*, which seems to be a common interpretation these days. Here, I will follow the business advice I heard once: "Be sure to sell something that you would be happy to buy if you were on the other side". We all share this opinion and ethos.

Finally, let's list the advantages of our business model. Our experience so far shows that our Partners *like working with us as much as we like working with them.* Beware new Partners: you are most likely to get hooked after the first year. Call me old fashioned, but I will compare the relationships between us and our Partners as a marriage. Therefore, it is not uncommon for us to continue to support you even if your business is going through a rough patch. It can be a tough world out there, and we are always happy to help if we have the capacity to do so.

Another advantage comes from the fact that by working closely with you, we are constantly challenged to develop new cutting edge functionalities and improve. In just a few years with effectively three people, we not only managed to catch–up with the big guys, but we went one–step further with some unique features: Immersed Boundary Method, Ghost Fluid Method, Level Set Method, Enhanced and Monolithic Hydro–Structural Coupling, Adaptive Mesh Refinement for arbitrary polyhedral cells, to briefly name a few. Even though we are smaller in size, I assume that this trend of advanced development will continue, especially because we love to do it.

Since we are quite fast and experienced in certain marine hydrodynamics simulations, we started expanding the third part of the business: Performing CFD simulations for Clients. This is the part of the business that we expect to scale well since we have automated the whole procedure for most of the simulations: from meshing and pre–processing to post–processing and report generation. In just one day, we can calculate steady resistance for your ship design. Currently, we have also automated the added resistance simulations and I expect that we will be able to calculate added resistance of a ship for ten wave lengths (usually enough for engineering purposes) in a single day. In fact, one of our Partners is already doing this within roughly 8 hours (no experimental facility can do this to the best of my knowledge). You can take a more detailed look at this part of the business on our new website: `www.navalhydro.com`. Now, allow me to make a comment on the accuracy of CFD simulations for marine hydrodynamics.

Here's an example on our added resistance simulations. I think that the accuracy we achieve with latest improvements (usually within 5-10% for the *added resistance*) is good enough.

Now, the scientific community might scream at me saying that we should be within 2% of the experiment, and I have a strong opinion that this is completely nuts. Let me try and explain why:

- First of all, by speaking to a lot of people from different experimental facilities and reading papers, it turns out that it is not at all easy to measure added resistance in experiments and the uncertainty can be quite high: around 10% for resonant condition and up to 65% for long waves where the added resistance is very small. And this happens when performing the *same* experiment, with the *same* model in the *same* facility. Imagine what would happen if they build the same model multiple times, much like we do when we perform grid refinement studies. That would be interesting to investigate.

- Secondly, although almost all CFD codes are extremely good and provide accurate representation of the physics, there is still some modelling involved (*e.g.* turbulence, waves *etc.*). Now, even for an experienced CFD engineer, mistakes happen, and this is why Inno is strongly against using the CFD code as a black–box, and I agree with him 100%. At the end, an intelligent engineer should always look at the results with a grain of salt.

- To make a third point, let's make a mental exercise. Let's say that you work at a ship design office and have been using our code for quite some time now. You are mostly interested in calm water resistance for your designs and you have gotten quite some experience and confidence in your CFD predictions. For ten ships you investigated so far, the prediction of resistance was within 2% and you feel confident to use the results for future design choices. You now have a new design, you perform your calculations and you predict the required effective power P_e needed to obtain the design speed. Further assume that the design speed is the most important design parameter within your contract. Now you need to choose the engine for your ship and let's say two of them are available: one with exactly the same effective power P_e that you need to achieve the speed and one with $1.1P_e$. Which one do you choose? What was that first lesson in engineering? **Margin of safety**. I would sleep better if I pick the $1.1P_e$ engine, especially since there are lots of assumptions (both in CFD and in other engineering predictions) that you at least need to be aware of. It's not the thing you don't know that does you harm, it's the thing that *you think you know.*

Just last year, after I finished presenting a paper at a conference saying that our newly developed linearised free surface CFD model (`singlePhaseNavalFoam`) predicted the resistance within 2%, I got a strong comment saying that this poor accuracy is a serious limitation of the model and that the *best* CFD codes can do better: being within 0.5%. At that time, that person worked for a major CFD provider as a support engineer. Go figure. I think that was enough of my thoughts for a year, let's now talk about the new stuff that we have been working on and what we have learned during the last year.

Current Status and Capability

Since this is the first letter of its kind (yup, we are making history here!), I will start by providing an overview of the Naval Hydro Pack, a bit of philosophy that we pursue while developing it and some challenges. The part of this text is submitted as a short abstract for the 13th `OpenFOAM` Workshop, but I think it's instructive to repeat myself here. The goal is to provide you with an overview of numerical models that we use for free surface flows in marine hydrodynamics, with open discussion on drawbacks and advantages of each of our methods.

The final goal of the Naval Hydro Pack is rather simple: I would like us to be able to reliably calculate the most complicated problem in marine hydrodynamics with reasonable computational resources and sufficient accuracy for engineering purposes. The most difficult problem I can currently think of is a self–propelled ship performing some kind of manoeuvre in severe weather. Now let's add a layer of complexity and consider that the ship is elastic and the structural response and hydrodynamic excitation are dependent on each other. Let us take a "divide–and–conquer" approach to tackle this complicated problem and identify the underlying challenges:

- The first obvious challenge is the modelling of the two–phase flow (water and air), where we need to accurately take into account free surface kinematics and dynamics.
- The second challenge is the efficient modelling of gravity waves, their propagation and prevention of reflection.
- The third challenge is associated with the hydro–structural coupling where the ship naturally responds to hydrodynamic forces acting on it. Even if we consider the ship as a rigid body, the hydro–structural coupling is highly nonlinear and therefore efficient strategies for resolving this coupling need to be considered.

I will now briefly explain our solution to these challenges without going into exhaustive details. In case you would like to know more about a certain topic, make sure to contact me via E-mail (`v.vukcevic@wikki.co.uk`) and I will provide references (tips, papers and slides).

Free Surface Modelling

As I see it, the free surface problem in numerical marine hydrodynamics can be divided into two parts: i) How to represent and transport the free surface and ii) Once the free surface location is known, how to obey the jump conditions due to different physical properties of the two fluids (air and water). Let's first tackle the transport and representation of the free surface.

Interface Capturing Schemes

In the Naval Hydro Pack, you can choose between three different interface capturing schemes:

1. Algebraic Volume–of–Fluid Method (A–VOF),
2. Implicitly Redistanced Level Set Method (IR–LS),
3. Geometric Volume–of–Fluid Method (G–VOF).

Each of these methods has certain advantages and drawbacks that you should be aware of. They are summarized in Table 1, where you can easily deduce that there is no *perfect* tool and some trade–off is always necessary (although you could open a bottle of wine with a knife, you should really consider investing in a cork–screw). Here are our thoughts on how to effectively use the

Naval Hydro Pack's interface capturing schemes based on the problems you are after. The A–VOF scheme is suitable for steady resistance and seakeeping simulations where large time–steps are required and the perfect mass conservation is desirable. In cases where unphysical numerical smearing of the interface is present with the A-VOF scheme due to complex flow field or poor–quality mesh, we tend to use the IR–LS method. If you have been running CFD simulations for planning hulls and experiencing that pesky *numerical ventilation*, this is a perfect solution. IR-LS ensures an interface as sharp as you wish, controlled by a parameter. Since there's no such thing as free lunch, the method is not strictly mass conservative. Still, this does not seem to affect the results we are mostly interested in: forces and motions. The G–VOF scheme is by far the most accurate scheme and therefore we often use it for violent free surface flows where we seek accurate interface resolution, *e.g.* green water simulations and atomization. The obvious drawback is that it requires a Courant number lower than 1.

Table 1: Comparison of different interface capturing schemes as implemented in the Naval Hydro Pack.

	A-VOF	**IR-LS**	**G-VOF**
Mass conservation	Machine tolerance	Discretisation error	Machine tolerance
Courant number limit	No	No	Yes, $C_o < 1$
Control of interface smearing	Poor	Excellent	Excellent

Ghost Fluid Method for Interface Jump Conditions

Now that we know the location of the free surface, we need to do something about the discontinuity in density (and consequently pressure gradient) in the best way we can. If a CFD code uses standard discretistaion schemes where the pressure gradient and density gradient are coupled within the momentum equation, this will cause spurious velocities near the interface in air, even without surface tension. This can cause all kinds of trouble: artificially increasing the Courant number of your simulation and/or affecting the transport of the interface (has anyone seen some weird *ripples* at the free surface lately?). Our solution to this is quite unique compared to other CFD codes: we use something called the Ghost Fluid Method (GFM), where we take into account the kinematic and dynamic boundary conditions when we *discretise the discontinuous fields* (no need to worry, no actual ghosts are involved, it's just a weird name of the method that has been around for 20 years, but not in industrial polyhedral Finite Volume codes). One of the commercial software providers actually uses a similar approach (I avoid using names here, in case you are interested, let me know and I'll tell you). Currently, the GFM takes into account the pressure gradient and density jumps across the interface, while the tangential stress balance is still approximated by calculating the velocity gradient using standard FV discretisation. This is something we should improve, but it doesn't affect the results for marine hydrodynamics length–scales. All interface capturing schemes may be used alongside the GFM in the Naval Hydro Pack. The old formulation without GFM is still available, although we rarely use it.

Since I promised to be fair and discuss the challenges as well, there is one difficulty with the GFM that we have recently observed. Since the GFM assumes perfectly discontinuous density and pressure gradient fields across the interface, the simulation becomes unstable if water entrains air in a way that the air should be compressed. This is actually the drawback of the incompressibility assumption and not the GFM method. Currently, Inno is working on extending the GFM for two phase flows where one phase is fully incompressible (water) and the other is compressible with isentropic equation of state (air).

It's time to recognize the indirect benefits of the GFM as well. As the velocity field across the free surface is continuous and there are no spurious air velocities in air, we observe two advantages: i) The advection of the free surface with all methods is much more stable since the flux is well defined across the interface and ii) The maximum Courant number is significantly lower, enabling faster simulations.

Wave Modelling

In the wave modelling arena and capabilities, I'd say that we are at the top as far as the commercial CFD software is considered. We use the relaxation zone approach where a part of the domain near the farfield boundaries is dedicated to forcing the CFD solution towards the desired potential flow solution. This way, we are reusing vast knowledge and decades of experience from the potential flow world. After all, potential flow methods for marine hydrodynamics are like an older brother to CFD methods and I think it would be foolish not to reuse this knowledge. Since the potential flow solution has negligible cost compared to CFD, we prefer to use either:

- Fully nonlinear stream function wave theory for monochromatic waves or
- Higher Order Spectrum (HOS) method for irregular sea states. The HOS is my favourite pick since it accounts for nonlinear wave modulation and wave–wave interaction, so there's no need to resolve these phenomena within CFD. Inno is the culprit who developed this for his Master thesis, so if you are using it, make sure to say thanks when you see him.

In the Spectral Wave Explicit Navier–Stokes Equations (SWENSE) method as implemented in the Naval Hydro Pack, the waves are introduced in the whole CFD domain. This approach gives us the possibility of calculating only the nonlinear perturbation around the explicit incident wave, instead of calculating the total fields using the standard approach and introducing waves only within the relaxation zones. Since the incident wave field is explicit in the SWENSE method, it *slightly* facilitates wave propagation in CFD domain. However, for violent free surface effects such as green water, it is reasonable to expect that the flow solution will be significantly different from the incident wave, making the SWENSE method unsuitable for this type of problems. I often use the SWENSE method for certain wave loads and almost all seakeeping calculations and for nothing else to be honest.

Hydro–Structural Coupling

Now that the numerics associated with the free surface flow modelling are behind us, we can move forward to the missing bit: the coupling of the flow field with the motion of a ship or an offshore object. Two distinct challenges can be identified when considering the hydro–structural coupling: i) Coupling strategy between the fluid–flow and 6 DOF equations of motion and ii) Efficient handling of dynamic mesh.

Algorithms for Hydro–Structural Coupling

In Finite Volume CFD, almost all hydro–structural coupling algorithms are partitioned, meaning that the flow field and 6 DOF equations of motion are solved one after another in an iterative manner. In the Naval Hydro Pack, the 6 DOF equations are strongly coupled to the fluid flow within the nonlinear (PIMPLE) loop, where a sufficient number (usually six or more for seakeeping) of nonlinear corrector steps is necessary to converge the solution within a time–step. Inno has implemented an enhanced strategy where the 6 DOF equations are additionally integrated after each pressure correction step. The approach provides significantly improved convergence of 6 DOF and flow field at a negligible CPU time cost. The benefit of this approach is that the same motions and forces are obtained by using 2, 4, 6 or 8 nonlinear correctors. Take a look at Figure 1, where the heave motion for seakeeping of the KCS model is shown. This allowed us to speed-up our seakeeping simulations up to three times with often insignificant loss of accuracy.

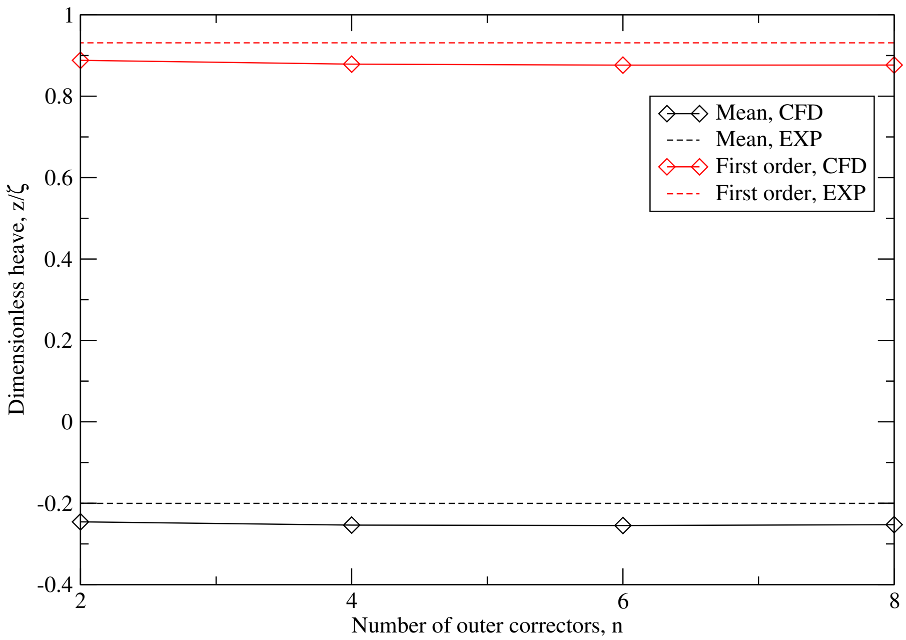

Figure 1: Heave amplitude and phase for different number of nonlinear correctors, KCS model, C5 case, Tokyo Workshop.

Recently, we (well, Inno again) have also developed a monolithic coupling approach, where the 6 DOF equations are solved as a constraint within the pressure equation. Similar to the enhanced coupling, the monolithic coupling improves the convergence with lower number of nonlinear correctors. The monolithic approach proved to be quite favourable for motions with high acceleration (*e.g.* seakeeping of high speed planning hulls) and high added mass (relative to the mass of the ship). However, you should know that for the seakeeping of displacement ships, this approach is a bit of an overkill: although it can be used confidently, the benefit will be minor compared to the enhanced approach. Give us a shout if you are interested in details.

Dynamic Mesh Handling

Once the 6 DOF equations are solved and the new position of the ship is obtained, one needs to efficiently handle the dynamic mesh motion. Different approaches exist in the Naval Hydro

Pack and `foam-extend`, which are summarized in Table 2. The first option is the simplest: the domain is moved as a rigid body. The method is simple to implement and robust, while it does not allow multiple bodies and is not suitable for fairly large motions. The second method is mesh deformation, where we prefer to use an algebraic approach where the mesh is moved rigidly in the vicinity of the body and the deformation slowly decays towards farfield boundaries. The approach is as efficient as the domain motion. Although it can also have difficulties with large amplitude motions, the method is suitable for most of the standard marine hydrodynamic problems. Both domain motion and mesh deformation strategies cannot handle appendages that should move relative to the ship. Recently, we have been working quite a lot on improving the Immersed Boundary library and Overset Mesh library in `foam-extend`, making these choices more suitable for marine hydrodynamic flows. Both methods offer extreme versatility in a sense that they can handle complex relative motions. The Overset Mesh method is more suitable for flows where viscous effects are important, while the Immersed Boundary method is favourable where the pressure effects are dominant and the viscous effects are of secondary interest. In current state, both methods are efficiently parallelised and work well, although the pre–processing stage is more demanding for the Overset Mesh since it requires careful consideration of the overlapping region. I'd say that both methods are in their intermediate stages of development, which practically means that the tools are working and we'd love to improve them with some latest ideas.

Table 2: Comparison of different dynamic mesh strategies as implemented in the Naval Hydro Pack and `foam-extend`.

	Rigid Motion	**Deformation**	**Overset Mesh**	**Immersed Boundary**
Complexity	Low	Low	High	High
Status	Mature	Mature	Intermediate	Intermediate
Robustness	High	High	Intermediate	Intermediate
Versatility	Low	Low	High	High

The Year Behind Us

Let's now discuss the interesting part of the work that we've done during the last year. I will divide the recent developments into three categories:

1. New Numerical/Mathematical Models,
2. Improvements to Existing Models,
3. Helpful Post–Processing Functionalities.

New Numerical/Mathematical Models

Let me list some new numerical/mathematical models that we have been developing for specific applications.

Linearised Free Surface Solver: `singlePhaseNavalFoam`

You know how two–phase CFD simulations can take a long time? Well, one of our Partners had a good idea to linearise the free surface boundary condition as they often do in potential flow. What does this mean? It means that you don't need to consider air in your domain and that you need to mesh only up to calm free surface. There's also no need to aggressively refine the mesh towards the free surface, saving quite a lot of cells and therefore CPU time. Now the limitations. The first assumption of the model is that, by definition, it cannot handle wave breaking and overturning waves. Furthermore, the modelling error is proportional to the square of wave elevation, η^2, meaning that the model is formally valid only for small, mild waves. For a general case, it's difficult to say in advance how big of an error you actually end up with, and again, it's up to an engineer to look at the results with a dash of common sense. Here, a lot of testing, validation and experience will be paramount to effectively use the method. So far, we have tested this on:

- Calm water resistance cases (Froude numbers of 0.14 and 0.28): with very good results compared to experiments,
- Wave diffraction on full scale ship, with good results compared to linear potential flow code,
- Seakeeping of ship at design speed (Froude number 0.14), where motions compare well with experiments, but the added resistance, being a second order effect, is not well predicted.

For these comparisons, we also compared the results to our fully nonlinear two–phase solver and apart from added resistance, they were quite good. One important thing to note is that we have used roughly 300 000 cells with linearised free surface solver and 3 000 000 with two–phase solver. Now that's a saving of an order of magnitude! The name of the new solver is `singlePhaseNavalFoam` and if I have managed to intrigue your interest, feel free to ask for additional information (papers, slides) and instructions on how to run the code.

Propeller Controllers

We recently invested substantial amount of time and energy to validate the Naval Hydro Pack for *full scale sea trials*. The first problem anyone encounters is: How am I supposed to find freely available data to compare my CFD results with? Luckily for us, Lloyd's Register organized a Workshop on Ship Scale Hydrodynamic Computer Simulation, where they took a 20 year–old general cargo carrier, dry–docked it, cleaned the hull and propeller, laser–scanned the geometry and made it *freely available*. They then took the ship to sea trials and measured the achieved speed for different propeller rotations. I personally think that they did a great thing since the CFD *should and will be* used for direct full scale calculations. A big thanks to them. Although it's quite impossible to get the shipyard to share the data from sea trials, we managed to do it. Thanks to Uljanik shipyard from Croatia, we got another set of sea trial measurements for a car carrier. This time, we were asked to report the achieved speed with specified 80%MCR.

Given two sets of results, we felt confident to perform CFD simulations and compare the results. However, we were missing propeller controllers for our actuator disc model of the propeller for the car carrier. Inno went on to implement them: `fixedRotationRate` and `fixedShaftPower`. Now, if you'd like to run self–propulsion simulations with the actuator disc model, you only need to provide open water propeller curves (K_T and K_Q) and specify the desired controller. The code will do the rest.

The first set of results for general cargo carrier was not quite good I must admit: we have under–estimated the achieved speed by roughly 6%. After fixing a bug in our actuator disc model, we got significantly better results. For both ships, the achieved speed was within 0.3% of the sea trial measurement. We have also performed grid sensitivity studies with quite satisfactory results. Take a look at the paper (`http://navalhydro.wikkiltd.co.uk/references/` under "Self propulsion simulations") for additional details.

Improvements to Existing Models

Here, I will talk only about *relevant* improvements and I'll skip a lot of minor details related to software maintenance and numerical consistency that we did during the last year. If you really want to know about those, take a look at the output of the `git log`.

Immersed Boundary Method

Previously, the Immersed Boundary method worked by identifying the cells in the vicinity of the .stl surface and tempering with the matrix coefficients to enforce Dirichlet or von Neumann conditions for these cells by doing a least squares fit. Then, Hrvoje had a happy thought that we should actually *cut* the mesh according to the .stl, but without performing any topology changes. The *cutting* happens behind the scenes and it only tempers with the Finite Volume mesh data (surface area vectors, delta coefficients, *etc.*). Now, the very cool thing about it is that it is much easier to enforce boundary conditions if you really do have a nicely defined boundary. Plus, the parallelisation is much easier and more efficient. This is now available in `foam-extend` and the Naval Hydro Pack, although we need to do additional testing. I'm quite optimistic about this improvement and eager to try it out on self–propelled course–keeping simulations where the rudder needs to move in order to keep the ship on the right course.

Free Surface Sensitised Turbulence Models

Another project with one of our long–term Partners was to improve the added resistance predictions with the two–phase solvers. We decided on a test case with quite good experimental results and ten wave lengths (ranging from $\lambda/L_{PP} = 0.3$ to $\lambda/L_{PP} = 2.0$). I happily ran 9 out of 10 cases without issues, but the shortest wave length kept crashing after several encounter periods for some reason. As we post–processed the remaining 9 simulations, we noticed that the added resistance was significantly over-predicted: up to 30% for some items! Then we started digging and found out that the turbulent viscosity was going crazy near the interface. Not only that the waves were damped, but they significantly changed the phase speed, causing a major phase shift between undisturbed solution in relaxation zones and CFD solution. This was especially apparent for the shortest wave length, which kept crashing for this particular reason. We went on to investigate what is happening at the interface with the turbulence model and it was quite straightforward to figure out. The velocity gradient at the interface is quite large when you have waves, and this causes the production term in the turbulence kinetic energy equation to grow significantly over time. When you take step back and think how these models have been developed and tuned for wall–bounded, **single–phase flows**, you immediately understand that we have to do something about this. At that point, we had two options:

1. Devise the turbulence model that is somehow sensitive to the presence of the free surface,
2. Properly implement the tangential stress balance jump condition at the interface with the Ghost Fluid Method.

Although I'd be happier with Option 2, real life circumstances (*i.e.* deadlines) made us pursue Option 1 first. Thankfully, I had the opportunity to speak to Mr. Devolder at a conference and he told me about his buoyant $k - \omega \; SST$ model, which practically forces a laminar flow near the free surface by an additional sink term in the k equation. I've implemented the model, but it proved to be unsuitable for long simulations (more than several wave periods) since the artificial decrease of turbulent viscosity spreads far from the interface over time. Still, if you'd like to try it: set `buoyantKOmegaSST` instead of ordinary `kOmegaSST`.

The solution we ended up being happy with is the `freeSurfaceKOmegaSST` model. It's quite simple to explain, but first allow me to introduce our reasoning for it. After reading quite a lot of papers regarding turbulence in waves close to the free surface, it seems quite difficult to draw any general conclusions. The thing that I'm confident though, is that the ordinary turbulence models tuned for **single–phase flows** are not suitable. In the absence of a better understanding of free surface turbulence, we decided to force a laminar flow only in a *narrow band* near the free surface. The only parameter you need to specify is the `blendingLength`, which indicate how far from the interface you want your turbulent viscosity to disappear. Of course, the blending is done in an intelligent way to ensure smooth transition from forced laminar flow near the free surface towards turbulent flow far from the interface. With this model, our *added resistance* prediction improved significantly, lowering the errors from $\approx 30\%$ to within $\approx 5\%$, and I warmly recommend it.

We have an on–going project on this topic and I hope to improve the turbulence models further in order to account for more physics near the free surface.

Additional Explicit Level Set Redistancing

You remember that I mentioned that we have *Implicitly Redistanced* Level Set method? Although the implicit redistancing worked quite well so far, it seems that the Level Set does not remain a signed distance function everywhere, especially in the stern region. In order to improve this, we have implemented an additional, *explicit* redistancing step that actually re–initializes the Level Set to signed distance function: no questions asked. This step takes some time (2–5% increase in CPU time), but it improves the conservation properties of Level Set. By default, this additional step is switched off. You can switch it on by adding `redistanceLevelSet on;` in `fvSolution::levelSet` dictionary.

Helpful Post–Processing Functionalities

To conclude the latest developments, I will simply list some new post–processing functionalities that you might find useful. These are all implemented as `OpenFOAM`'s function objects so you can use them for on–the–fly processing.

- `shipHydroCharacteristics`: This function objects outputs everything you'll probably need when running calm water resistance simulations. It writes temporal data of: resistance (pressure, viscous, total), resistance coefficients (pressure, resistance, total), wetted surface and sinkage and trim. In addition, it writes average y^+ on the wetted part of the hull to let you know whether you ended up in the buffer layer, and also viscous resistance and its coefficient according to ITTC 1957 correlation line. It can't hurt to check your results against the ITTC.
- `smartPressureProbe`: This is used to keep track of large pressure spikes on your patch in a memory–friendly way. Pressure spikes are constantly screened for on a given patch and when the threshold for pressure or pressure change rate is exceeded, the output is activated. But first, last n (10 by default) time–steps are retrieved from memory and the pressure trace is recovered. The output automatically deactivates when the pressure goes below the lower threshold. This is useful for screening for slamming or sloshing events.
- `maximumPressureIndicator`: This function object is used to keep track of the maximum pressure on a given patch. At the end of the simulation, maximum pressure values are reported for each face of the patch, along with the time at which the pressure peak has occurred and the face centre location. Now you can easily find maximum pressure exerted on your structure and at which time it occurred.

The Year Ahead of Us

Before I leave you to your interesting work, I'd like to tell you what will be our focus during the year ahead of us. Regarding development, as it currently stands, we will be working on:

1. Extending the Ghost Fluid Method for flows with compressible air, while the water remains incompressible. This will be interesting for those of you interested in slamming, sloshing and probably some other application areas I currently can't think of. The nice thing about this formulation is that we are able to run high Courant numbers as we did so far with the incompressible solver.
2. Including the tangential stress balance jump condition for our incompressible Ghost Fluid Method solvers. To be honest, this won't probably make a slightest difference for marine hydrodynamics applications from the practical point of view, but the high–fidelity two–phase atomization simulations will be more accurate. I also expect this to fix the issue with turbulence models near the free surface in a natural way, since the discontinuity of the velocity gradient will be taken into account properly.
3. Improving the turbulence models for free surface flows. Now this is going to be an interesting project. I'm currently reading up a lot on turbulence measurements in free surface flows, and it's going to be interesting to see whether we can significantly improve the existing free–surface–sensitised turbulence models to account for observed phenomena.

4. Extending the Immersed Boundary method with Adaptive Grid Refinement and Dynamic Load Balancing. Yup, you read that correctly. Yes, we have improved Immersed Boundary now, and yes, we have Adaptive Grid Refinement that works on arbitrary polyhedral cells. The missing bit is the Dynamic Load Balancing, which is something Hrvoje is working on at the moment. I am very eager and optimistic to get my hands on this and test it for rudders and propellers. Next year, I'll let you know how it went.

We have also applied for a project related to automatic hull form optimization using open source tools. Apart from that, we would love to work on improving the automatic fringe assembly in Overset Mesh and also start creating a framework for complete hydro–structural analysis of a ship as a flexible body. This is probably where we will spend some free time.

Of course, testing, verification and validation of the Naval Hydro Pack never stops. We will also spend some time on automating our workflow regarding different marine hydrodynamics simulations in order to further speed–up the delivery of results to our Clients. By doing this, we hope to moderately scale that third part of the business.

As usual, I'm sure that you, our Partners, will let us know if you find some things that can be improved to make your life with the Naval Hydro Pack better. It's been fantastic working with you and I'm looking forward to the next year.

Sincerely,

Vuko Vukčević, April 16, 2018