Computer Aided Ship Design 2030 - I Can See Clearly Now

Herbert Koelman, SARC, Bussum/The Netherlands, H.J.Koelman@sarc.nl

Abstract

This paper sketches an image of the near future of Computer Aided Ship Design (CASD). It starts with a reflection on the model of ship design methodology, followed by several technical CAD-related individual subjects, such as advances in virtual and tactile modelling methods of the ship hull, tools for data representation and processing and the organization of heterogeneous software components into one (virtual) CAD system. Other subjects are of more general nature, such as a plea for the return of empirical methods in ship design, and hardware support for computation-intensive tasks. Finally, three possible future scenarios are elaborated, which reflect general trends in programming and the daily use of computers, and their impact on CASD.

1. Introduction

This paper is an update of *Koelman (2013)*, and contains an outlook on Computer Aided Ship Design (CASD). I restrict myself to the mid-term — say the next decade — because a longer period cannot be foreseen. The basis for the outlook is threefold: (1) our experience with software development and contributions to research projects, (2) some notions from literature and (3) personal observations and projections. Addressed will be six, more or less disconnected subjects:

- The model of the ship design process.
- Contemporary methods for design and representation of the shape of the ship hull.
- Product Data Technology; data exchange and inter-program communication.
- A plea for the revival of empirical prediction methods in ship design.
- PC hardware support for computation-intensive tasks.
- Scenarios on the use and advancement of computers and their impact on CASD.

2. Ship design methodology

When it comes to design methodology, the design spiral always surfaces. This familiar model of ship design activities, which is attributed to *Evans (1959)*, shows the distinct design phases (initial design, embodiment design, contract design, detailed design) and suggests a fixed sequence of activities. It is a bit funny that this model is still so often referred to in these years, because its reality level is low. After all, in practice we don't see fixed sequences of design activities, while design phases may overlap or merge. Also in literature, the concept of the design spiral has been criticized, see e.g. *Nowacki (2009)*, *Harries et al. (2011)* and *Koelman (1999)*. In the latter, a toolbox is proposed as model for ship design activities, Fig.1, although this can hardly be considered to be a structured model, because summarized it expresses that "every activity can be performed an undetermined number of times in an arbitrary sequence".

In this respect, we can question the use of these kinds of models anyway. See e.g. Fig.2, where the model of a company is represented (in Dutch). What is its use? It might be that it has a certain descriptive value, e.g. for a novice in the company, or an external party. On the other hand, such a model is also conservative, in the sense that it conserves the present state of the company. And, as a consequence, suggests that the company should be organized according to this scheme. Or that the implementation of company automation is done according to this model, while the automation could enable a new, better, organization structure.

Fortunately, making these models is a bit of a hobby for academics, without much consequences in practice. However, in the ship design practice some other concepts or abstractions exist, which are not holistic models, but which are guiding decisions nevertheless, such as:

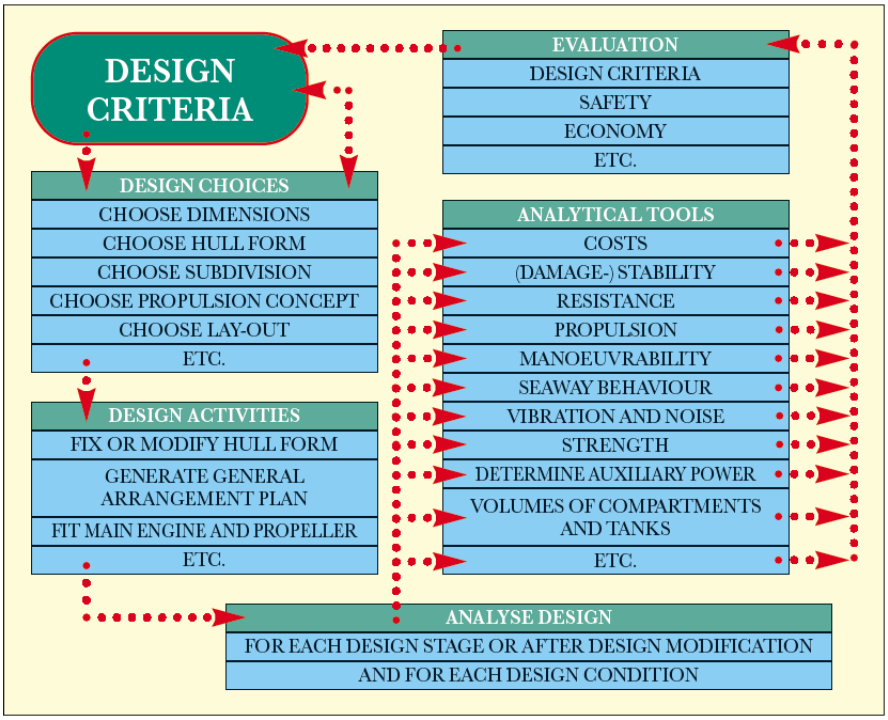

Fig.1: Toolbox metaphor of the ship design process

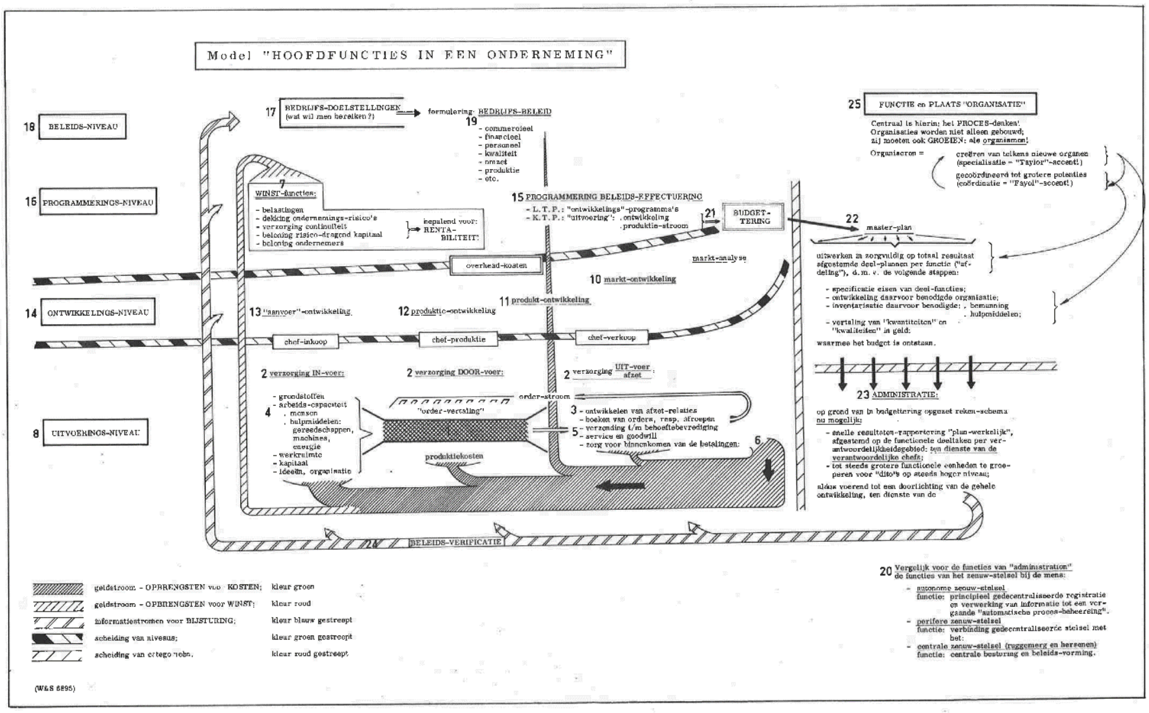

Fig.2: Main functions in a company (TU-Delft lecture notes, 197X)

- The question whether to use 2D or 3D (computer) representations for the ship design. This is really a question from the 1980s, every decent CAD or dedicated ship design program from today has an underlying 3D coherence, although the workbench or output (the "drawing") may just show a 2D intersection or projection. An occasional genuine 2D software program that has escaped the author's attention can be considered to be ripe for the museum.
- The concept of "design phases", which we also encountered in the ship design spiral. There might have been days that the only way to manage the complex task of ship design was to

divide it into phases (and correspondingly divide the design office into departments), but improved CAD modelling methods and communication facilities make such a strict discrimination between design phases a bit obsolete. E.g. moving a pipe line a short distance in the "detailed design phase" may bring it − unnoticed and undeliberate − into another damage stability zone, and consequently may have a large impact on damage stability characteristics. While damage stability is an aspect of "embodiment design" as well as the final trim and stability booklet, which is in its turn part of the "post-detailed design phase". So, this simple action affects all design phases.

- The idea that the major purpose of a piece of software is its only role. Obviously, in some cases this statement is correct, after all nobody would like to make a lines plan with MS-Paint. However, in other cases this is a simplification. E.g. the main purpose of our ship design software is to make all kinds of ship design computations, so it can be labelled as a "calculation package". We have seen users defining hull form and compartments with a general-purpose CAD-system − because of its label of being a "design package" − and subsequently typing over or exporting coordinates and bulkhead locations to the "calculation package". While the other way round − defining in the "calculation package", with its dedicated and optimized definition methods, and then exporting to the CAD-package for the fringes and details − is faster, more pleasant and more robust.
- The notion that information which is available electronically is generally usable. This idea is stimulated by the nice flow diagrams in leaflets of software suppliers, which often put their products in the core of a network of arrows and circles. We will revert to this subject later on…

The conclusion is a bit that all models and abstractions might on the one hand add guidance and structure to a confusing reality, but on the other hand lead to suboptimal decisions because people might think that the model <u>is</u> reality.

3. Hull shape

The prevailing hull shape representation method is by B-spline or NURBS surface. That method has its merits but certainly also its disadvantages, on which we will not re-iterate because it has been duly reported in literature, e.g. *Koelman (1997), Sharma et al. (2012), Koelman and Veelo (2013)*. In the latter, also an alternative method is presented, which is implemented in the ship design suite from my company, already quite some years ago. So, there is sufficient material to reflect on the reception of this method, and the conclusion is that in particular by the users the method is praised for its flexibility and preciseness, but that recognition in general goes slow. For which a reason will be that many people have become so accustomed to the prevailing NURBS method that they don't see their drawbacks; we have seen nicely rendered NURBS-based hull shapes declared to be fit for production with gaps of a decimetre, with waterlines containing multiple deflection points, or with buttocks in the parallel body that where just not straight but oscillating up and down. Another reason can be that users, without realizing, work around the limitations of the NURBS method. For example, by introducing chines, which are not present out of naval architectural necessity, but because in that way the program limitations can be circumvented.

Concerning export of surfaces to IGES, our company has experienced a bit of a drama. After a ship hull has been designed with our ship design program, the result has to be exported to other tools, e.g. for engineering, CFD, FEM. And many of those accept import preferably in NURBS format. So, through a backdoor the NURBS re-enter! In principle, our modeller can readily export a NURBS model through the IGES-NURBS standard, that function is already available for many years. However, it also results in a vast number of small NURBS surfaces, and quite some software packages have problems with that amount, which hampers a fluent data transfer. In order to smoothen this process, our modeller is at this moment being equipped with a postprocessor which provides two functions:

- Recognition of larger four-sided regions (which is one of the limitations of NURBS, to be four-sided). Take the example of Fig.3, where without special provisions each face between the curves would be converted to an independent NURBS surface, where the nonfour-sided faces would be split into multiple four-sided ones. For the relatively simple example of this figure this process would lead to some 80 NURBS surfaces in the IGES file. With the four-sided recognition feature the number of surfaces will be decreased to some 11, as depicted in Fig.4.
- These regions are being converted to NURBS by means of a special algorithm, which guarantees a set of NURBS surfaces which are guaranteed to be gap-free (which is a prerequisite for some CFD programs).

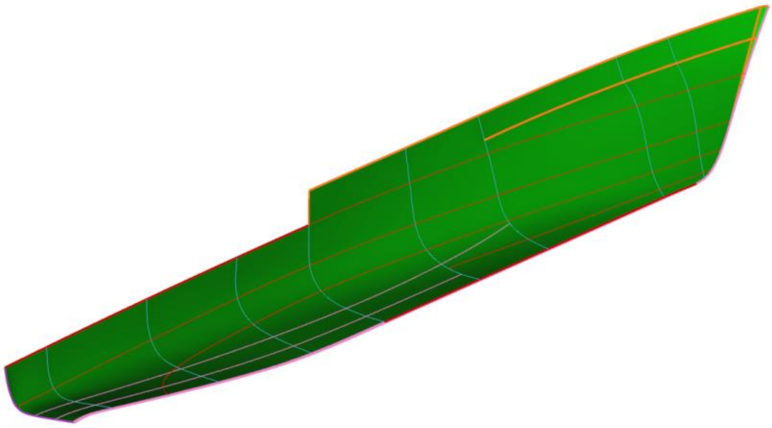

Fig.3: Some 80 surface patches between the curves of this ship hull model

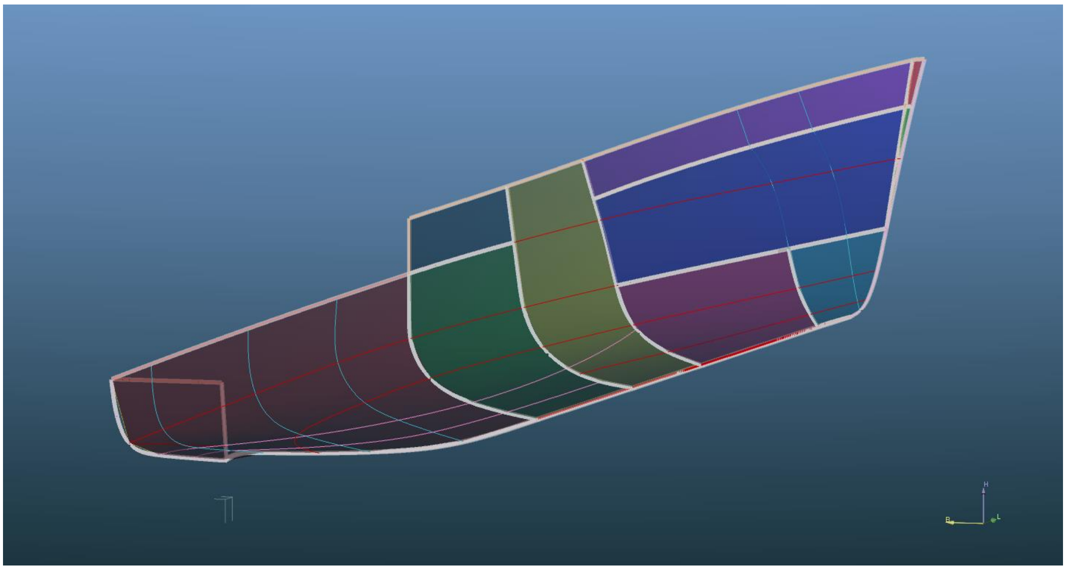

Fig.4: Some 11 four-sided patches covering the same surface as depicted in Fig.3

It will be a bit of a sour remark, but these features may repair some of the difficulties caused by the employment of methods of limited capabilities in other software.

The final question is whether the ship design community will stick to the NURBS method, or will eventually switch to something better. That is not easy to predict. On the one hand, it could be expected that ship designers with an independent state of mind critically evaluate their present tools and methods. On the other hand it could be that a generation has already become so custom with the NURBS method, that they do not realize that alternatives do exist. A phenomenon known in psychology as imprinting, https://en.wikipedia.org/wiki/Imprinting_(psychology).

4. Don't Let Me Be Misunderstood (On PDT)

PDT, Product Data technology, is the common denominator used for data exchange, data sharing and collaboration between computer programs. This subject is perhaps the determining factor for CASD development in the years to come. However, littered with some major misunderstandings. When it comes to communication, three layers can be distinguished, ordered top-down in Table I.

Table I: Layers of communication

	Inter-human		PDT for CASD	
Layer	Embodiment	Applicable standard	Embodiment	Applicable standard
Semantics	Knowledge of the world	A system of values and believes	Knowledge of the ship and her parts	High-level agreement on the ship and her components
Representation language	E.g. English	E.g. the Oxford dictionary	E.g. STEP, XML, DXF or IGES	Dictionary
Communication channel	E.g. radio waves	E.g. modulation and frequency	E.g. discfile, USB-stick, Tcp/ip or Internet	E.g. volt, tesla, bit-coding and tcp/ip protocol

The lowest level, which is the hardware level of the communication channel, can easily be established; it appeared to be easy for mankind to set standards on frequencies, disc file formats, volts and bit coding standards. However, one level higher, on the language level, this become more difficult; you cannot understand the Korean TV news if you don't master the language. But even if you understand the language, inter-human communication and understanding fails without a common set of values and beliefs; I understand the words of the US press officer Kellyanne Conway if she comes up with the concept of "alternative facts" in January 2017, but lacking a shared system of values I fail to understand the point she is trying to make.

Applied to PDT similar conclusions can be drawn. A shared communication channel implies that the raw data are present "electronically", however, as such that does not imply their usability. One level higher, by using the same representation language, is a prerequisite for successful data exchange, but not sufficient. E.g. STEP and IGES contain many representation variants, and only with agreement on all variants the exchanged data is usable. An example is that an IGES file can contain a faired lines plan, represented by construction frames and many waterlines and buttocks. However, if the importing program's internal representation is a NURBS surface, then the received data are not readily usable. They will have to be converted, which is not a trivial task, in this example.

Why is this so surprising? After all we already know that computer communication often works faulty. If you try to import a nicely formatted Word document in OpenOffice the whole document format will be destroyed; pictures are shifted to the edges of the pages, often illegible, font features are gone etc. Or take the title of this section, it is a song title, and for convenience I copy/pasted it from Google, with the result you see (the reason is that decimal 39 is ASCII coding for the single quote ' character, but that is not the question). Apparently, mankind is not able to transfer a line of text or a flat A4 paper from one program to another, but we still expect that 3D product models can be communicated meaningfully? The remarkable thing here is not that shit happens, the funny thing is that people persist in expecting flawless program interoperability "because the data are available electronically" while the practice shows so many examples of the opposite.

What can we do now after this gloomy conclusion? Try to find an alternative interoperability solution. Standard data exchange formats, such as STEP, DXF or IGES, drop out, firstly due to the representation variation issue (the fact that so many alternative representations exist), and secondly because they carry only geometry, and no semantics (e.g. a file can contain a plane. What is that plane, a bulkhead? Watertight or not? Or fire-resistant? Is the bulkhead from the type "collision bulkhead"? Does it require corrosion allowance? Etc. etc.). Emerging standards are 3D PDF, X3D and JT (Jupiter Tessellation), however these are just viewer standards, and do not contain a complete geometric model (let alone semantic information). So, these formats also drop out. Instead, for the past years we have realized a collaborative system based on:

- Direct communication between multiple applications, over TCP/IP.
- No common product model.
- Exchange of higher-level entities, such as “a transverse bulkhead”, or “sewage piping system”.
- Coding in XML, on the basis of a dictionary, where all semantics are written down. For the sake of convenience these definitions are as much as possible extracted from STEP (notably AP215 and AP216), extended where required.
- Communicate not only product model data, but also request/replies (of derived, volatile data).
- Exchanging bulkheads and decks, restricted to two systems.

While, at present, the system is being extended with:

- N-system communication.
- System management.
- Compartments.
- Piping details, to be used for e.g. damage stability, progressive flooding, time-domain analyses (such as time-to-evacuate) and engineering, Fig.5.

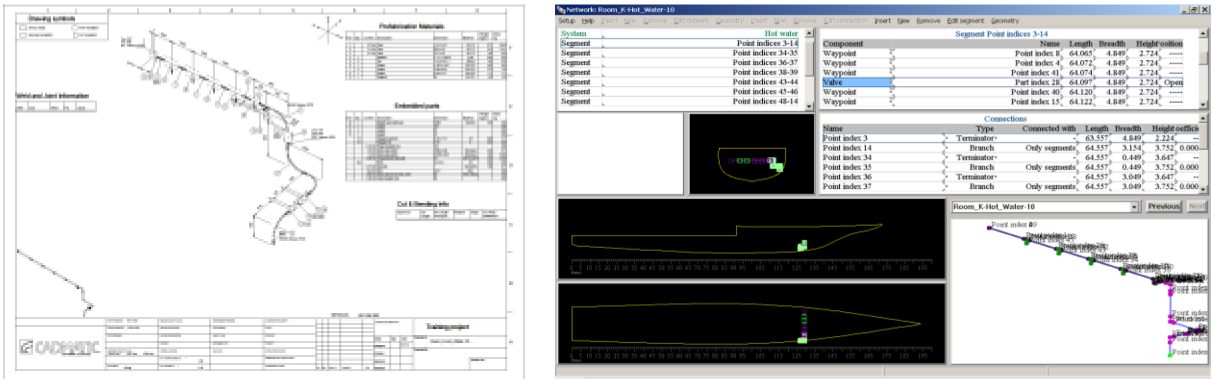

Fig.5: Experiment with piping system shape and meta-information in CADMATIC (left), shared by XML with PIAS (right)

A screenshot of the two GUI’s of the system is depicted in Fig.6. According to pilot users, the present development version of this system works quite satisfactorily. At this moment suggestions are being compiled for auxiliary user functions, in order to be able to offer a comprehensive product to the market. The conclusions on this development are:

- This initiative is different from earlier central product model concepts.
- A strong aspect is not only data sharing, but also commands.
- It relies on high-level semantics (e.g. the concepts “transverse bulkhead” or “sounding pipe”).
- So, it requires willing partners. Computer communication relies to a large extent on human communication.
- It is a powerful approach, but won’t conquer the whole world. The roughly estimated maximum number of connected computer programs will be some 10, perhaps 20.

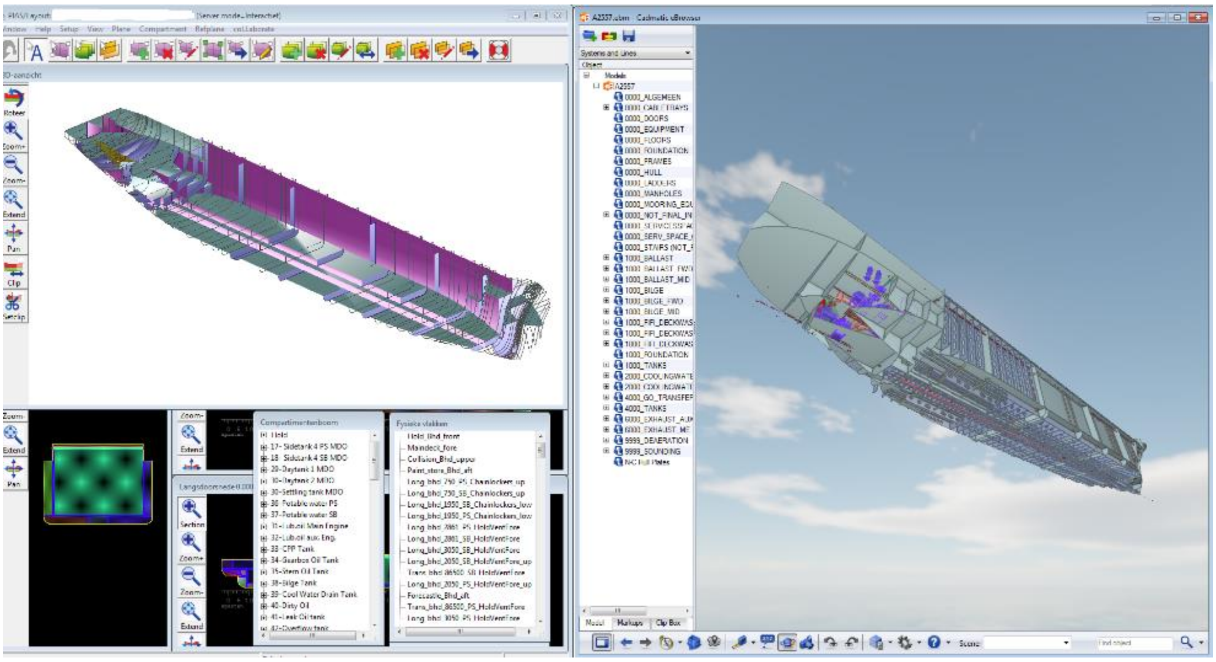

Fig.6: Screenshot of two systems, simultaneously working on the same ship design, left PIAS and right CADMATIC

5. 3D printing

Koelman (2013) reported some experiments on ship design support with 3D printing of cheap desktop printers. It was anticipated at the time that desktop printers would emerge for design support applications, but in our direct environment we see little sign of that. Although the technology has become much more mature and reliable. E.g. the printer we used was a Do It Yourself Ultimaker Original, which has initially a fail rate of some tens of percents. Newer models are version 2, which comes assembled, and 3, see ultimaker.com/en/products/ultimaker-3. The applied Ultimaker technology is Fusion Deposit Modelling (FDM), which is essentially layered manufacturing with one source of material. This method requires a flat base layer of sufficient size, for example, it will not be possible to print an Eiffel Tower model upside down, because due to the lack of a base layer the object will turn over. This might be solved by segmenting the model, but that requires additional processing as well as post-printing assembly. Version 3 has two print heads, one with the conventional construction material, and the other with water soluble support material. The latter can be used to support the artefact during printing, and washed away later on, thus allowing many more shapes to be printed in a single run. Also, larger manufacturers enter this market, for example Hewlett-Packard, although their Jet Fusion 3D printer, www8.hp.com/us/en/printers/3d-printers.html, is with a price tag of $200,000 a bit expensive for common desktop use.

Although penetration of 3D printing for design support may go a bit slow, its application in manufacturing is presently boosting, *Economist (2017)*. A typical marine manufacturing application is printing a propeller by Wire Arc welding, www.imarest.org/themarineprofessional/item/3364-3d-printing-takes-the-next-step-in-the-maritime-industry.

6. The return of empiricism

The advance of the computer has opened up new possibilities for the ship designer and brought many handy tools, such as CFD, advanced optimization algorithms and product model sharing. However, with the emphasis on these new possibilities an elder class of tools, the empirical prediction methods, is being neglected. In particular, for concept studies and the early design stage, such methods have proven to be extremely useful. Take for instance resistance prediction methods such as from Savitsky and Holtrop & Mennen, or steel weight approximations by Schneekluth or Westers. Unfortunately, these methods have not been updated for modern designs or construction methods. The most recent Holtrop & Mennen publication is from 1984, and the steel weight approximations date back to the early 1960s. This is peculiar, not only because the need for such methods is compelling, but also because now empirical methods could be built with today's possibilities, such as:

- Massive statistical analyses, such as regression models with a large degree of freedom, or response surface models.
- Collecting empirical material used to be tedious, for example doing model experiments. However, numerical experiments based on FEA or CFD could generate "numerical series".
- In 'those' days a prime requirement was to communicate the method in a condensed way, by using equations with only a few coefficients or graphs. But nowadays things are much easier. Large amounts of numerical data can easily be distributed and just as easy being read into a computer program or a spreadsheet.
- The increased processing power brings additional features within reach. For example, by extending the prediction method with confidence limits. In this fashion a probabilistic steel weight method might be possible, where not only the steel weight is predicted, but also its probability distribution.

In this respect, the occasional publication with a fully elaborated empirical method is a delight, e.g. the compact overview of empirical prediction methods for catamarans in *Haase et al. (2013)*. And notably *Hekkenberg (2013)* and *Rotteveel et al. (2014)*, which contain quite some useful empirical material for steel weight and hydrodynamics of inland waterway vessels. Another example of recent application of empirical methods, in the field of seakeeping and stability, can be found in the contemporary development of second generation intact stability criteria, *Umeda and Francescutto (2016)*. I hope that these efforts will be an inspiration to other scholars to work in this area. And remember: Eternal fame will be yours. For what is the reason we are still familiar with the names of e.g. Puchstein or Lap?

7. Hardware-supported accelerators

Distributed processor power supply has become abundant for the past decades, but processor power demand has increased in more or less the same pace. Reasons for the latter are for instance the increasing computation load as required by the legislation – probabilistic damage stability, 2nd generation intact stability criteria – and the application of optimization methods; genetics algorithms are powerful, but require many generations to find the optimum. Until some ten years ago that was somewhat automatically compensated by the autonomous increase of PC processor power, because every two or three years a new generation of processors was released with more or less double speed. Unfortunately, that effect has grinded to a halt, processors running single-core tasks are not significantly faster today than ten years ago.

For this problem, a solution may be found in High Performance Computing or Cloud Computing, *Mallol et al. (2016)*, an approach which brings huge time savings for tasks that can be massively parallelized, such as CFD, FEM and, potentially, probabilistic damage stability. However, there are also tasks that play predominantly on the desktop, in an interactive fashion, with a short burst of computing power demand. For example, a surface rendering task, dozens of stability calculations in a series of waves (such as required for second-generation stability criteria) or a series of some 100 damage stability calculations, as may be required to assess compliance with damage stability requirements on an on-board loading computer. So, demand for high-performance desktop processing power remains. Modern processors offer three facilities for that:

- Spreading the workload over multiple processors (multi-threading).
- Using Advanced Vector Instructions (AVX) on the Intel processor family, en.wikipedia.org-/wiki/Advanced_Vector_Extensions. With this feature, arithmetic operations on Floating Points (FP's) can be done in parallel, as depicted in Fig.7. Depending of the AVX-version a

maximum of four or eight simultaneous operations can be performed. The reason that AVX is dedicated to FP's multiplications, divisions etc., is that those are relatively time-consuming operations.

- Using Graphics Processing Units (GPUs). This facility may rely on the availability of a GPU of a certain brand or family, although standards are emerging, such as CUDA, en.wikipedia.org/wiki/CUDA, and OpenCL, en.wikipedia.org/wiki/OpenCL.

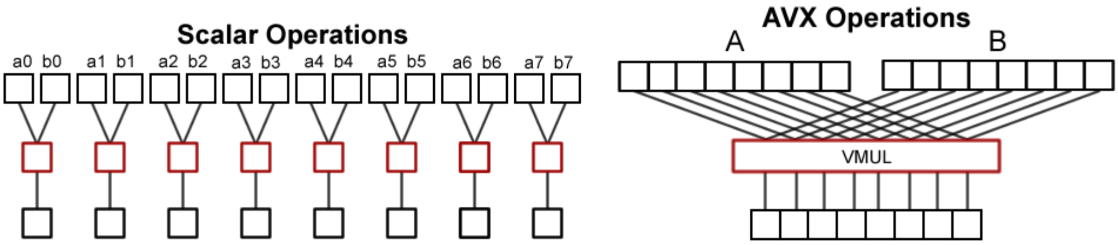

Fig.7: Eight multiplications sequentially (left), and parallel with AVX (right).

The GPU solution is reported to deliver quite some performance. However, two factors make it less suitable for standard application. The first is the requirement of a powerful GPU to be installed, which may not always be available on e.g. a laptop computer, and the second the lack of a uniform standard for the communication with the GPU. So, for our software family we have only investigated the first two options, which led to the following conclusions:

- Contemporary standard Windows multi-threading facilities perform reasonably well. The original Win95 CreateThread had quite some overhead, hampering its use for short tasks. In the best Microsoft tradition, this phenomenon was not solved by improving the multi-threading functions as such, but by adding a complete new design and function set, called Thread Pools. With the latter, also shorter tasks can be efficiently parallelized, although finding the balance between speed gain and overhead still requires quite some experimenting.
- Also in an application that was originally not design for multi-threading, sufficient tasks can be identified for and adapted to parallel execution.
- Because some practical implementation choices have been made, based on experiments, the maximum number of supported cores is eight. A higher maximum will be feasible, but will also require re-tuning.
- On a computer with N cores, speed gain is ≈ 0.75-0.90 N, where a core is a real, independent core, and not the kind of pseudo-core as Intel provides with its 'hyperthreading' technology.
- AVX may in theory provide a performance gain of a factor 8 on FP operations, however its application in practice is slowed down anyway because the numbers have to be shuffled and rearranged before presenting it to the processor. However, a major governing factor is that even programs that do a lot of arithmetic also spend quite some processor cycles to administrative tasks, such as the transfer of data between CPU and memory. In our application, some hyper-frequently executed, FP intensive, tasks have been identified and adapted to explicit AVX processing. The AVX speed improvement of the entire program is with 5-10% a bit disappointing, on the other hand, every contribution counts, and if you are working against a deadline…

8. Scenarios on the use and advancement of computers and their impact on CASD

Koelman (2013) postulated three mid-term scenarios on the use and advancement of computers, and their impact on CASD. The world has not changed much since 2013, so these scenarios, updated to the 2017 insights, are still deemed to be realistic:

- **Fragmentation, symbolized by the collapse of MS-Windows**. Time was that Microsoft (MS) ruled the world. However, in the technology battle of today its role seems to be over; for example a briefing on future platforms, *Economist (2012)*, reports in depth on Apple, Amazon, Google and Facebook, but spends only a single word on MS. A possible reason for the silence around MS was given in the *Financial Times (2017)*: "The Microsoft future epitomises the economist John Hicks's quip: 'the best of all monopoly profits is a quiet life'. Microsoft in the 1990s became famous as a once-brilliant company that decided to pull up the drawbridge, locking in consumers and locking out competitors". Although as such the downturn of MS might be a relief for many, the annoying undertone is that no prevailing platform will arise that can mature to mainstream. As such, the lack of a clear winner is not a big disadvantage, it might even be healthy for innovation. But consequences might be that PDT and collaborative design will become tedious.

- **The standstill era**. In the Netherlands, the 18th century is known as the 'wig age', the 'pruikentijd'. This is generally considered to have been a standstill era, where the country still floated on its successes of the Dutch glorious Golden Age, however, with little innovation. The same might be the future of computing: overwhelmed by the magic of the computing and connectivity power the strive for innovation is lost for the moment. Or to condense this scenario in a rhetorical question: Do you think that our common desktop tools, such as spreadsheets, word processors and databases have significantly improved since 1986? I don't. Admittedly, we have gained e-mail and Internet, but for the rest my office productivity was higher in 1990 than it is now. A second example of standstill is the unconnectiveness of apps. It is often said that an iPhone or Android device makes you a member of the connected world, and that is true, but at the same time you are the endpoint, the terminal. The apps themselves are not interconnected, all connection must pass through and be interpreted by the human being. In some sense, it is a reminiscence of the mainframe model of the 1960s and 1970s; a powerful computer at the center and many dumb terminals at the spokes of the wheel.
- **A bright young world**. In this scenario, the focus lies not so much on computer programs, but more on methodological advancement; the development of methods that assist the ship designer in the daily practice, but also at a deeper level help to make our industry more competitive and more pleasant. Obviously, the results will become available as computer applications, however, with more attention to the methodological user-friendliness than to appearance and user-interface wizardry. This kind of user-friendliness also stimulates a common ground for interoperability technologies.

The advantage of scenarios is that we do not have to choose. It is likely that a mixture will become reality. Obviously, everybody will favor the third scenario. But please see that such a choice is not without engagement. This scenario can only become reality if we actively contribute to it and stimulate others to do so, too.

9. Conclusion

Looking back, the outlook is sometimes a bit grim, summarized:

a) all ship design model disposed,
b) an awkward but popular ship hull representation method might continue to haunt us,
c) even in simple 2D cases data exchange may fail and
d) office PC's have lost their autonomous performance increase.

On the other hand, there is progress:

1) a better hull form method does exist,
2) for collaborative design a workable framework has been sketched,
3) PC processors offer alternative performance increasing facilities, and
4) 3D printers gradually become more reliable and more abundant.

So, a bright future may lie ahead, but will not arrive without us pursuing it. In the words of *Popper (2001)*: "When I say, 'Optimism is a duty', this means not only that the future is open but that we all help to decide it through what we do. We are all jointly responsible for what is to come. So, we all have a duty, instead of predicting something bad, to support the things that may lead to a better future".

References

ECONOMIST (2012), *Technology giants at war*, Dec. 1, pp.25-28

ECONOMIST (2017), *3D printers will change manufacturing*, June 29

EVANS, J.H. (1959), *Basic design concepts*, J. American Society for Naval Eng. 71/4, pp.671-678

FINANCIAL TIMES (2017), *This is the age of the Microsoft and Amazon economy*, May 18

HAASE, M.; DAVIDSON, G.; THOMAS, G.; BINNS, J.; BOSE, N. (2013), *A practical design approach including resistance predictions for medium-speed catamarans*, Ship Technology Research 60/1, pp.4-12

HARRIES, S.; TILLIG, F.; WILKEN, M.; ZARAPHONITIS, G. (2011), *An integrated approach for simulation in the early ship design of a tanker*, 10th Int. Conf. Computer Applications and Information Technology in the Maritime Industries (COMPIT), Berlin, pp.411-425

HEKKENBERG, R. (2013), *Inland Ships for Efficient Transport Chains*, PhD thesis, TU Delft

KOELMAN, H.J. (1997), *Hull form design and fairing: Tradition restored*, 6th Int. Marine Design Conf., Newcastle, pp.421-429

KOELMAN, H.J. (1999), *Computer Support for Design, Engineering and Prototyping of the Shape of Ship Hulls*, PhD thesis, SARC, Bussum

KOELMAN, H.J. (2013), *A Mid-Term Outlook on Computer Aided Ship Design*, 12th Int. Conf. Computer Applications and Information Technology in the Maritime Industries (COMPIT), Cortona, pp.110-119

KOELMAN, H.J.; VEELO, B.N. (2013), *A technical note on the geometric representation of a ship hull form*, Computer-Aided Design 45/11, pp.1378-1381

MALLOL, B.; DEL TORO LLORENS, A.; BERTRAM, V. (2016). *Trends in CFD illustrated exemplarily for competitive sailing yachts*, 15th 10th Int. Conf. Computer Applications and Information Technology in the Maritime Industries (COMPIT), Lecce, pp.33-41

NOWACKI, H. (2009), *Developments in marine design methodology: Roots, results and future trends*, 10th Int. Marine Design Conf. (IMDC), Trondheim, Vol 1, pp.47-80

POPPER, K.R. (2001). *All Life is Problem Solving*, p.143

ROTTEVEEL, E.; HEKKENBERG, R.; LIU, J.L. (2014), *Design guidelines and empirical evaluation tools for inland ships*, European Inland Waterway Navigation Conf., Budapest

SHARMA, R.; KIM, T.W.; STORCH, R.L.; HOPMAN, H.; ERIKSTAD, S.O. (2012), *Challenges in computer applications for ship and floating structure design and analysis*, Computer-Aided Design 44/3, pp.166-185

UMEDA, N.; FRANCESCUTTO, A. (2016), *Current state of the second generation intact stability criteria - achievements and remaining issues*, 15th Int. Ship Stability Workshop, Stockholm, pp.1-7