Proceedings of 15th International Marine Design Conference (IMDC-2024)
June 2-6, 2024
Amsterdam, the Netherlands

Piping Layout Integrated in Ship Design and Stability Assessment

Herbert J. Koelman[1,*]

ABSTRACT

Damage stability assessment in ship design is a well-established area of our trade. However, where originally only a limited number of aspects were involved, gradually more details are included. Notably compartment connections by pipes and ducts etcetera. Combined with a high number of damage cases, in practice this results in a set of computations which is not complex as such, yet complicated by its sheer size. Although in the PIAS ship design software suite quite some dedicated tools are available, those have never been designed to support the requirements from today. In this light the software has been extended with a new system to fully define shape and topology of compartments and their connections. This paper reports on the system design, its application in damage scenarios, and on complications.

KEY WORDS

Damage stability; Piping; Time-domain; Ship design.

INTRODUCTION

If a compartment is damaged in such a way that it is open to the sea, then it will obviously be flooded, which can also extend further into the vessel through all kinds of openings or connections between compartments. In stability regulations, the word *progressive flooding* is mostly used for this, but we rather avoid that because it suggests that the flooding continues until it is fatal, which is not necessarily the case. The ship design and stability suite PIAS accommodates this phenomenon, by a sub-system *complex intermediate stages of flooding*, which dates back to the nineties. This works on the basis of non-uniform filling percentages per compartment, where necessary supplemented by virtual compartment connections. Although this has served the program users well over the past decades, it has some limitations:

- It supports only *virtual* connections between compartments. That means that a point in 3D space can be assigned as being 'the connection location', without any physical object connected to that point.
- Only one single location is supported as connection point between two tanks.
- Data input and output only in text.
- Supports computation of time to equalization. Yet only a) for a single compartment connected to the sea, and b) as disconnected calculation, not integrated in the stability computation and assessment.

Although as such it works well and is widely used, this system has never been conceived for intensive usage. While on the other hand we can witness an ever-increasing attention to effects of filling and flooding through holes, openings, ducts and pipes. Which is caused by stringent rules, increasing scrutiny by authorities and classification societies, hence increasing awareness of ship designers and shipyards, and finally increasingly complex vessels (from the layout design point-of-view). So, in order to satisfy this market demand, it was about time for a new PIAS subsystem for next-level damage stability assessments, including major effects of internal flooding and its progression.

In the set of rules and regulations on ships and shipping literally dozens of places can be identified where aspects of openings, internal connections and damage to piping systems are addressed. Although it is not the intention to provide a full overview here, some examples will be presented. For example, in IACS Unified Interpretations, notably No. 110, on tankers, chemical

[1] SARC, Bussum, The Netherlands
* H.J.Koelman@sarc.nl

Submitted: 7 February 2024, Revised: 9 April 2024, Accepted: 1 May 2024, Published: 23 May 2024

Conference paper, DOI: https://doi.org/10.59490/imdc.2024.886 e-ISSN: 3050-486

tankers and gas carriers: "Progressive flooding through internal pipes: In case of damage of an internal pipe which is connected to an undamaged compartment, the undamaged compartment should also be flooded, unless arrangements are fitted (e.g. check valves or valves with remote means of control), which can prevent further flooding of the undamaged compartment". Another example, taken from IMO (2020), on Probabilistic Damage Stability: "The factor s_i is to be taken as zero in those cases where the final waterline immerses the lower edge of openings through which progressive flooding may take place and such flooding is not accounted for in the calculation of factor s_i. Such openings shall include air pipes, ventilations and openings which are closed by means of weathertight doors or hatch covers". Finally, MSC (2020) links consequences to equalization times by cross-flooding arrangements (which are constructed to reduce the heel in the final equilibrium condition), with thresholds of one and ten minutes.

Until now, we have discussed the realm of requirements. Now we will briefly revisit recent related advances in modelling of algorithms on internal flooding after damage. Both Vermeer et al. (1994) and Ruponen (2007) researched in detail flooding scenarios after collision damage. Although this has created valuable insights, the applications were quite detailed and focused on specific specimens. Ruponen et al. (2012) have compared cross-flooding times as derived with different methods, and concluded amongst others that the simplified formula from IMO resolution MSC.245(83) satisfies well in case of small pressure variations, however, this conclusion may not be extended to different flooding cases. In Kariolus et al. (2019) a probabilistic assessment of progressive flooding has been proposed which takes into account the probability of opening internal watertight doors etcetera. In Braidotti & Mauro (2019), a fast method for the calculation of progressive flooding is presented, based on an analytical solution of the linearized system of differential equations.

THE NEW PIPING-BASED SYSTEM

Although from the viewpoints of regulations and science on the models and scenarios of flooding and related residual stability the last word has not been said, our aim in the development of the new system was not on advancing the latest theoretical insights. Instead, our contribution lies in helping our software users by taking one step forward from the conventional, while still maintaining an understandable and traceable modelling and computation method. With this background a design specification has been drafted, which was circulated under PIAS' customer base. Some replied with valuable remarks or ideas, which have been processed in the specification and the subsequent implementation.

In the remainder of this section, we will discuss the modelling methods, followed by an overview of the computational approach, and finish with a few core computational details.

Topology And Geometry of Openings, Pipes and Ducts

A foundational choice for the new system was to build the data entities and mutual relationships onto a specific standard for a sector which is — as a matter of speaking — littered by pipes and their purpose: the process plant industry, ISO (2019). However, this ISO 15926 standard is conceived for a wide range of industries and applications, so unavoidably quite broad, and yet unfocussed. Fortunately, for the process industry this has been further detailed by the DEXPI association, DEXPI (2019). From this extended standard five entities have been used which are sufficient to model a shipboard piping system for our purposes:

- **Equipment**, which is a thing, not being a compartment, connected to a pipe line but not part of it. Such as an engine or a chiller. Equipment play no role in computations, it is just defined for the completeness of definition and drawings.
- A piping **system**, which is an administrative collection of pipes of the same type, for example 'ballast' or 'Methanol'.
- A piping **network**, which is one closed system of connected pipes, which belongs to a piping system. This is the core of the piping data structure.
- A piping **segment**, which is one branch of a piping network, and extends between two points without sub-branching in between.
- A piping **connection**, which is a part located at the extremities of a piping segment. These come in six types:
    - Branch, where multiple piping segments meet.
    - Unprotected opening, external (so, to the outside of the ship).
    - Opening + Weathertight Air Pipe Closing Device (WAPCD) a.k.a. vent check valve.
    - Terminator, which closes a dead-end segment.
    - Compartment, or, more precisely, a connection to a tank or compartment at a certain location. So, a compartment may have multiple connections. The representation method of compartments will not further be discussed here, that has been elaborated in De Koningh et al. (2011).
    - Equipment, or, more precisely, a connection to a piece of equipment at a certain location.

- A piping **component**, which is a part located in a piping segment. This can be a waypoint, elbow, valve, pressure relief valve, reducer, check valve, internal WAPCD or a straight pipe section.

A schematic example of a piping system, including these entities and their relationships, is sketched in Figure 1, while an(other) example of the PIAS GUI implementation is reproduced in Figure 2. Space does not allow here to discuss all tools and features of this GUI, so we suffice here with this link to the manual.

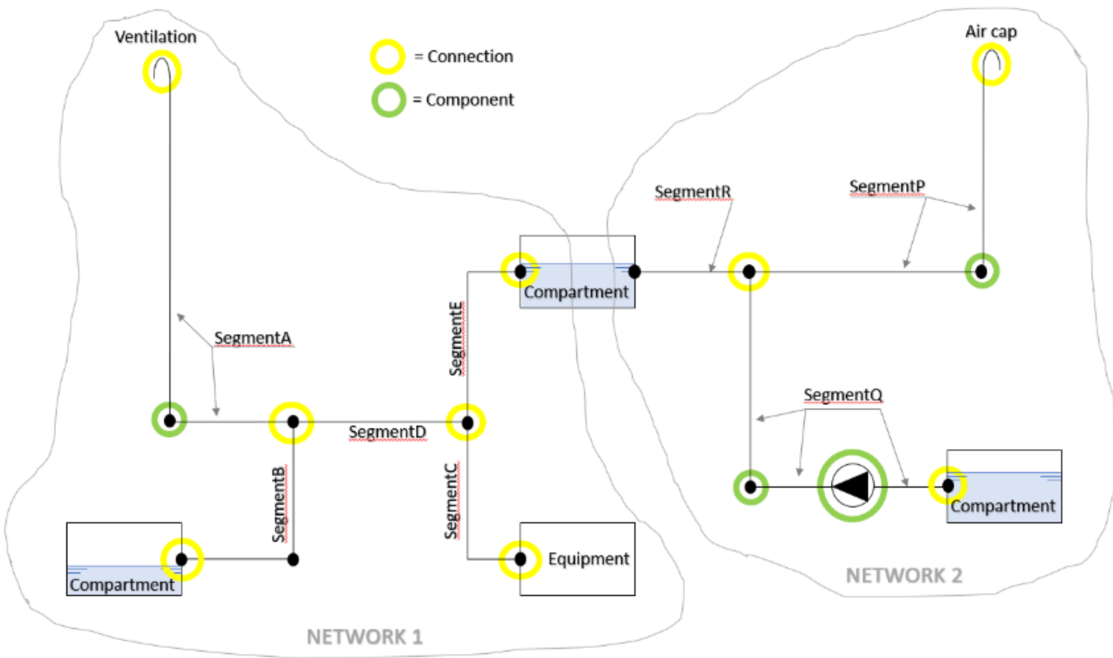


Figure 1: Schematic example of a two piping networks

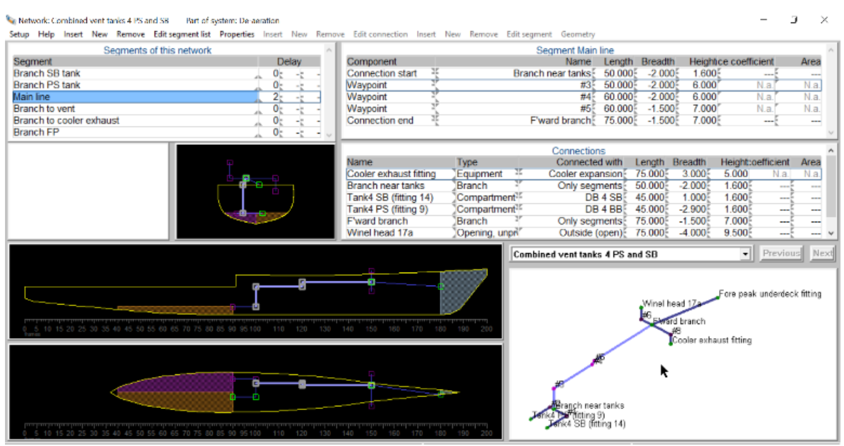


Figure 2: GUI implementation, showing a single network including connected compartments

Two Categories of (Damage) Stability Computations Including Pipes and Ducts

As elaborated in the introduction, the aim of our developments has been to create a framework for next-level damage stability assessments, which is a) suitable for large-scale application, b) efficient from the viewpoint of human effort and c) understandable by the commonly trained naval architect. The presented piping networks, combined with existing representations of loading conditions, hull shape and compartments, open the way to a twofold of such applications:

- A time-domain analysis. For the reasons that a) this has a clear physical meaning, so b) it represents reality an order better than conventional 'intermediate stages of flooding', while c) time as such plays a role in some stability criteria and time-to-capsize assessments.

- Something that resembles 'stages of flooding'. For the mere fact that such stages are anchored in tradition, and hence in textbooks and regulations.

Combined, the new system, which embraces these two calculation types, is called *Consecutive Flooding*.

We start the discussion with **time domain** because its physical background makes it easy to grab. In line with the premise to create a practically useful system, the simplifications which are common in stability assessment have been applied here as well. These concern the omission of inertia of ship and fluids, temperature effects and the impulse of the inflowing water. Furthermore, the following assumptions apply:

- Ultimately, in fully damaged condition all initial damaged tank content is lost and fully replaced by sea water.
- Fluids mix instantaneously and fully, so there is no density gradient. For a damaged tank with intact fluid this implies that a stage at 99% of total flooding time will contain a mixture of intact and ingressed fluid, which introduces a discontinuity with the previous bullet. This is addressed by the computation of two variants at the moment of total flooding, the first with a mixture of fluids and the second with sea water. The first represents the condition directly after full flooding, the second at a long time after that.
- Non-damaged tanks, which are filled through connections with other tanks, receive a new density which is the proportional mix between original and ingressed densities. So, the intact content is never fully replaced (unless it flows out through another pipe connection), regardless the size of the connection.
- The volumes of the connecting pipes are neglected.
- Components have a single resistance coefficient, so there is no transition between laminar and turbulent flow.

With the latter simplification, the effort comes down to applying Bernoulli's equation in discrete time steps. In Ruponen (2006) this numerical differential equation was solved by a second-order scheme, which is more efficient than linear. However, it also requires that discontinuities (in flow vs. time) are identified, which is easy to do with time-based events (e.g. opening a valve) but less easy to identify in case of geometric effects (e.g. a large discontinuity in the shape of a tank). For that reason we have chosen first-order Euler integration. In each integration step the following actions are performed:

- Set the positions of pressure relief valves.
- Extending pipe openings with an outflow above a fluid surface with an additional, artificial pipe perpendicular to that surface. For the reason that pipe flow theories are based on pipes and reservoirs, and don't support waterfalls.
- Converting this piping network into a (mathematical) graph.
- Identifying closed loops in this graph, with a method for finding Fundamental Cycles.
- Applying Kirchhoff's laws around these loops and all nodes, as illustrated in Figure 6.25 from White & Xue (2021). This leads to a system with N equations and N unknowns.
- The resulting flow multiplied by the time step gives the fluid quantities per tank. These are capped if that leads to overfilling (>100%) or underfilling (<0%) of a tank in that time step.

With the second system the flooding is modelled with **intermediate stages of flooding**, so without explicit time. This implementation was conceptualized given two facts:

- Standard stability regulations apply the concept of 'intermediates stages of floodings' of fixed percentages, i.e. 25%, 50%, 75% and 100%.
- Not all compartments are always flooded with the same percentages, i.e. with small connections between compartments the flooding of the connected compartments may lag behind the flooding of the ruptured compartment.

Our system maintains the notion of 'percentual stage of flooding', because a) this is a fundamental concept in present damage stability regulations, b) therefore this concept is familiar to authorities and classification societies and c) the concept is easy to understand. To have a shorthand word for this concept it was labelled *fractional*, because essentially it fills compartments by 'fractions' of the final volume. Additionally, an integer *delay* is available to specify a lagging in time. All in all, this is a simple system, supporting conventional intermediate stages and a bit beyond. Those interested in examples are directed to this section in the manual.

Core Computational Details

Besides the core of the computation, some additional facilities and choices are of interest. The first is the **WAPCD** (Weathertight Air Pipe Closing Device), which besides its common modus operandi according to IMO/SOLAS interpretation, in our implementation also supports an additional *safety distance*, hence accommodating European Inland Waterway tanker regulations ES-TRIN, CESNI (2019).

Frictional resistance by pipes is in the essence a complex issue. In practice, however, a number of practical methods and parameters exist, some of which have been implemented:

- The (cross-sectional) shape. Choice of round or square, the most common shapes.
- The cross-sectional dimension. If round then diameter, if square then edge length.
- The (dimensionless) resistance coefficient, for which three common methods are available:
    - According to IMO resolution MSC.362(92), where the frictional resistance per meter length is 0.02 ÷ hydraulic diameter.
    - With a user-specified Darcy-Weisbach coefficient, where the frictional resistance per meter of length is that coefficient ÷ hydraulic diameter.
    - With a user-specified resistance coefficient per meter of length.

A second issue is the energy loss due to **fluid outflow from a pipe**. For flow resistance of pipes and components, IMO has adopted a family of resolutions, namely A.266 1973, MSC.245(83) 2007 and MSC.362(92) 2013. To support the stability assessment of elder vessels, our system supports all three. A central element in these resolutions is the so-called 'reduction of speed', F, which is directly related to component resistance coefficients. A confusing factor is that in the three resolutions the equations for F are similar, but not equal, because two of the three have a factor *one* in the denominator, which the other lacks. This factor is to account for the energy loss due to outflow from the end of a pipe, Söding (2002). In order to harmonize these regulations, as well as to accommodate more than a single pipe outlet, the user can choose between implicit (= with the *one* in the denominator) or explicit (= user-defined) outflow energy loss. By the way, in the mentioned IMO resolutions the underlying outflow differential equation is (after some simplifications) solved analytically, which leads to a closed-form expression of equalization time. Instead of that expression, in our approach the flow resistance factors from these resolutions are used directly during the numerical integration of the differential equation.

Finally, the issue of **larger angle stability**, the computation of the GZ curve. From the viewpoint of consistency this is a bit of a strange phenomenon, because it is artificial. In the physically based time domain computation, no force or moment exists that forces the ship to heel. And there is not sufficient time to heel, because the next unheeled time step is ready to commence. This is solved by a) computing GZ without time effects, while b) omit possibly ingressed or shifted fluid *during heel* in the next time step. Still, there is an open question on how to determine whether fluid is transferred through pipes or openings during heel. In case of a small diameter pipe it is obvious that the roll period is not sufficiently large to allow for significant fluid transfer during heel. While a half-height bulkhead allows a waterfall over the bulkhead edge during heel (possibly without backflow when heeling to the other side). To differentiate between the two cases, a user can specify a 'minimum cross-sectional area for instantaneous fluid passage', which gives the program a handle to either allow flow through pipes and openings (if the actual area is larger than this minimum) or block the flow (if the actual area is less).
For all intermediate stages a GZ curve will be computed and assessed. In time domain for each time step full stability could be computed as well, however, this would lead to a large number of computations without much added values. For this reason, the program has a few switches to balance the amount of GZ evaluations. Anyway, for each time step draft, trim, heel and fluid quantities are available.

CHOICES AND OBSTACLES

Although the basic functionality and procedures are rather straightforward, completing it into a computer program usable in day-to-day ship design practice requires quite some deliberations and choices, in which we also encountered an unexpected obstacle here and there, which will be discussed in this section.

Permeabilities

There is this famous Monty Python sketch about an accountant, Herbert Anchovy, who gets bored of the dullness of his work. Well, our profession equally knows a Herbert who also has that feeling, about a small detail that has haunted him for more than four decades; the permeability of a tank. We all know that for the computation of tank tables a 2% reduction on volume is a common estimation to account for internal construction of steel vessels. Hence, a permeability of 98%, while all damage stability regulations impose a 95% permeability for tanks containing liquids. As such it is silly to apply different permeabilities for the same space; suppose a double bottom (DB) is fully filled with sea water. When it becomes damaged, due to some magical intervention from above suddenly 3% of the content is assumed to vaporize. And suppose the ship has separated SB and PS DB tanks, with the SB tank damaged and PS intact. Then, due to this loss of weight at SB, the vessel will slightly heel over to PS. While in reality nothing happens.

In the 1980s, for PIAS the choice has been made to interpolate permeabilities based on the stage of flooding, in order to achieve continuity at 0% stage = intact, and 100% stage= fully damaged. But now, with pipe connections, the situation is even more confusing, Take, for instance, a tank which is not ruptured but nevertheless flooded by seawater through a pipe, which is on its turn connected to a second, damaged, tank. What should then be the permeability of the first tank, damaged or intact? Consulting the regulations does not provide a unison answer, for IMO (2016) provides a different choice than IMO (2020). So, a program setting was created for this, including a paragraph in the manual with some more details.

Later in this paper the verification process will be discussed, and it can already be revealed that on many occasions we thought to be looking at wrong results, which later turned out to be correct. Had permeability again played tricks on us.

Intermediate stages of flooding

The whole assumption behind the idea of a *fraction* (a generalization of an intermediate stage of flooding) is that the immediately affected compartments will be flooded through a small damage. After all, if the damage were large, the ingressed water would spread rapidly, and the intermediate stage would be so short that it would have no effect on ship's position and stability. So, then the intermediate stage would actually not exist. Based on this physics-based reasoning, a distinction is made between large and small damages.

To assess stability in damaged condition, the worst-case scenario will have to be considered and since it is not known in advance how large the damage will be, cases with both a large and small damage are calculated. In the event of large damage, seawater can flow freely in and out of the damaged compartments, so that even during roll the water level in those compartments is equal to the sea water level. Because this all happens so quickly, intermediate stages do not actually emerge. In the case of a small damage, on the other hand, the water flows through the hole so slowly that the intermediate stages can take a long time, and thus should be considered in separate consideration. However, if the water flows slowly, then during rolling it does not have time to flow in or out significantly. So, in this case the volume of water in a compartment can be assumed to be constant for all heeling angles. Suppose intermediate stages of 25, 50 and 75% are specified by the user, the complete damage stability evaluation will consist of:

Table 1: Damage stability assessment stages.

Damage	Stage	Water in compartment	Verification against stability criteria for
Large	Final	Freely flowing in and out	Final stage
Small	Final	Constant, as at equilibrium heel (call that W)	Final stage
Small	Intermediate	75% of W	Intermediate stages
Small	Intermediate	50% of W	Intermediate stages
Small	Intermediate	25% of W	Intermediate stages

Damage stability criteria

Table 1 already shows which criteria for final or intermediate stages of flooding are applied. For the time domain computation, for (user-configurable) time intervals a full stability calculation can be made and verified against criteria. The question whether to apply criteria for final or for intermediate stages can be derived if a time limit is given, for example the 10 minutes from MSC (2020).

Another dilemma with stability assessment is the displacement through which the uprighting moment is divided, in order to obtain uprighting lever. Intact displacement or damaged displacement? Quite often, in literature this aspect is related to the lost buoyancy vs. added weight issue, but that is utterly confusing, for this issue is related to two separate *computation methods*, while our dilemma is just the choice of a denominator in an arithmetic division. Even in my favorite textbook on stability, Biran & Lopez-Pulido (2014), the method (lost buoyancy) is confused with the denominator equaling the intact displacement (which is called *constant displacement* in some stability regulations). Because different stability regulations dictate different denominators, our program has, reluctantly, been equipped with a setting where the user can specify the applicable choice. *Constant displacement* has been the program's default since the 1980s, a choice that has recently conveniently been formulated by eq. 3 of Ruponen et al. (2018).

Program settings

In the ideal world, where everything is equal, a computer program needs no settings. In reality, preferences, traditions, ships and regulations differ, so there is a compelling need for settings in a program or app. Quite some settings have already been discussed in this paper, while some additional remarkable settings will be elaborated in this section.

The whole idea of a time domain calculation is to calculate the **time in which the fluids equalize**. The flooding process is finished when the whole system of vessel and fluids have come to rest. However, towards the end of the process the fluids start to flow slower and slower; after all, the level differences get smaller, and hence the pressures and the flow velocities and flow rates. In essence, it is an asymptotic process (in particular exponential, see Braidotti & Mauro (2019)) where after, so to speak, many hours, milliliters are still flowing through the pipelines and openings. Indeed, in theory, equalization time will always be infinite. In practice, a certain tolerance can be set, so that if e.g. the difference in draft between two consecutive time steps is less than a mm or 1/10 mm, that is considered as 'rest'. However, that is an arbitrary tolerance that unintentionally has a large outcome on the final answer; at 1/10 mm, the equalization time can easily be twice as long as at 1 mm.
One might think of implementing a practical limit; after all, we are interested in the tons flowing through the system in the early time, and not so much in the milliliters in the last seconds. With that idea, a criterion can be formulated that is related to the transferred weight. For example 'if 98% of the total, final, fluid weight has flown through then I consider the system at rest'. The default percentage used in PIAS is indeed 98%, but this setting allows the users to adjust it as they see fit.

Another aspect is the concept of **equalization time** itself. In MSC (2020) this appears to be a strong issue, although the concept is not crystal clear. We assume that the idea is that an asymmetric damage is equalized by fluid flowing through a cross-flooding device, to the other side of the ship. And that, in the ideal world, this process comes to an end when symmetry has been achieved. But what when the geometry as such is not purely symmetrical, when, for instance, the tank at the other side is slightly smaller than the ruptured tank? Then a symmetrical condition will never be reached, and hence it will never be equalized? Unless MSC (2020) is augmented with explanatory notes, this will be an issue for subjective preferences by ship designer or inspection body. Hence, a cause for a setting.

Finally, **conflicting settings** may exist in our program. As such, this does not raise an alarm, it is just a matter of choices, some of which were made decades ago. Quite some of these conflicts are not fundamental, it is just that we don't consider it to be economical to allow for a certain combination of features, such as *heeling angles larger dan 90 degrees* (which exists in PIAS since 1982, and has been used a few dozen times) combined with this new *Consecutive Flooding* system. Others are intrinsic, such as that this advanced *Consecutive Flooding* cannot be combined with the approximative method of accounting free liquid surfaces by only their transverse moment of inertia. That is fundamentally impossible. Anyway, when conflicting options have been set, the conflicts are reported in a matrix in a popup box to the user, so he or she can make an appropriate choice. We realize that such messages are always a bit annoying when popping up, but they are inevitable given the sheer size of computation options in PIAS. An alternative would be to discard elder options, but we never know who in the customer base will still want to use such a legacy feature, for whatever reason, one day.

Effects of internal openings on the GZ curve

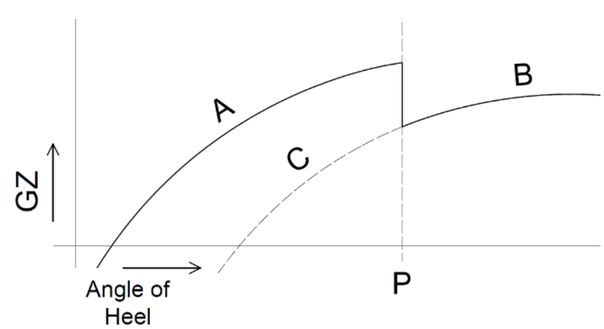


Figure 3: GZ curve with internal opening submerged at angle P

How does our program deal with an internal threshold or pipe which is submerged, and consequently allows transfer of water, at an angle of heel which is beyond the static equilibrium? Take for example the GZ curve from Figure 3, where at angle P the upper edge of a partial bulkhead overflows, leading to the filling of an adjacent compartment and therefore a deteriorated stability. It will be undebatable that the GZ will initially follow curve A, until angle P is reached, where a greater amount of ingressed water will lead to reduced curve B. However, the question is what happens on the “way back”, i.e. with decreasing angle of heel? The water will not fully flow back over the bulkhead, so a curve as indicated by C can be expected. And the subsequent question is which curve to use for the verification of GZ against stability criteria, A+B or C+B?

In PIAS, the past decades A+B has always been used — numerous calculations have been issued at classification societies and shipping inspections, and approved — based on the reasoning that the notion “way back” is never properly addressed, neither in literature nor in regulations. A few more arguments can be made in favor of this choice:

- The example above is expressive, but counter examples also exist. Take the GZ curve from Figure 4, with the partial bulkhead now immersed at an angle P which is much larger now. If the vessel is subject to IMO's Intact Stability Code then the maximum heel for criteria evaluation is 50° — the weather criterion — while angle P is much larger than 50°. So, this loading condition meets all stability criteria long before P is reached, and a reduced C-branch will not be applicable.
- Will 50° then be the determining angle? In many cases not, because dynamic stability equality (area A=B from the weather criterion) may have been reached at a much smaller angle. So, the possible branching of the GZ curve should be related to the applicable stability criteria, one way or another.
- Assume now that at the same large angle P not an internal opening spills over, but an external opening (e.g. a ventilation inlet), which sinks the ship. Then beyond P, the GZ curve will vanish, so also branch B. If one would argue that with an internal opening branch C should be taken, then the same reasoning should be applied to external ones. However, with branch B also C has vanished, so using this branch will render the whole GZ curve non-existent. Nobody — user, researcher, authority nor classification society — has ever suggested such a ‘solution’, because it would be unrealistic.

Supported by these arguments, it was chosen to keep the computation method for this subject in PIAS as it always has been. This is an implementation choice, not the irrevocable result of the modelling method. So, alternative choices could be made, if there would be a generic reason, such as clear and unambiguous guidance by rules or regulations, or unified regulations from institutions, such as IMO, IACS or national authorities.

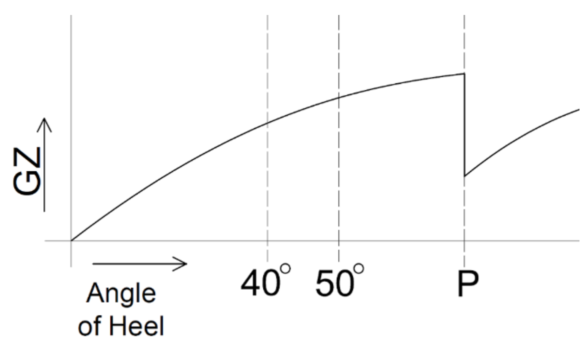


Figure 4: GZ curve with internal opening submerged at a large angle

VERIFICATION AND ACCEPTANCE

The whole idea behind this software is that it satisfies user demands, for which it needs to be practically usable, efficient, and reliable. The first two aspects have permanently been in the background, in the discussion of the software design so far. Anyway, all three issues will be addressed in this section.

Manufacturer’s verification

For verification of the proper functioning of software, the first impulse is mostly to maintain a few test cases on file, which can be used to verify the outcome at regular intervals. Although this is as such a feasible approach, it has a few drawbacks:

- In practice it is not easy to keep the test cases stable. If, for example, for a specific test or demonstration somebody changes something in the ship data, or even in a setting, some outcomes may change. But not necessarily all outcomes, which makes the change sometimes hard to notice.
- If multiple settings should be tested, then multiple instances of the same test cases are required. If, subsequently, something changes in the test case then that change should be (manually?) copies to the other instances.
- Automated comparisons are not feasible without special provisions. Each and every change in a piece of static text in the output (a word, a unit) leads to a difference. Indeed, all outputs are different anyway, because of different times and dates!

So, for the piping-based computations a different approach is used. The basis is still some pre-defined data in PIAS format, although only from shape of hull and compartments. All other elements are added by a specific test program, which generates variations in openings, pipes, connections, loading and program settings. In that way the consistency of the test cases is guaranteed because the variations are not stored, but re-generated when necessary. This test program — which is actually very short, and relies for all computations on the same libraries as the regular program — redirects the computational results to a plain text file, without pages, dates or much text. These text files are fit for automatic comparison on a regular basis. Obviously, the output of the end-user's program should still correspond with these text files, which requires a manual verification. However, if later undeliberate program modification destroys some output then it will be quickly apparent. The consequences of an unintended change in computation is much more dangerous, because it is not always visible. And that has been covered by the test program.

The test program generates some 50 independent test cases, with varying levels of complexity. All these cases are in separate documents compared with independent verifications, which are as such quite basic, for example by spreadsheet or by comparing with other PIAS output (such as that the 0% intermediate stage of flooding should equal the intact loading condition). Obviously, verifying all computation steps would be too labor intensive, so e.g. in a time domain computation of 800 time steps, a few samples have been verified manually.

These verification documents form a part of our IP, and will not reach the public domain. Although one or two of the test cases will be commented and released, to serve as example for others to do similar exercises. Because the underlying calculation steps are so elementary, with a simple test ship, tables of (inclined) hydrostatics & tank volumes and workable knowledge of Bernoulli's equation, every naval architect should be able to verify (samples of) these calculations with spreadsheet or calculator. That is not the prerogative of SARC.

Anticipated embracement by users

Inspired by literature on engineering design, in particular Horváth (2007) and Gero & Kannengiesser (2004), for new software functionality we apply a stepwise process, including a number of feedback loops. The steps comprise 1) Establishing the need, 2) Analysis of task, 3) Conceptual design, 4) Embodiment design, 5) Detailed design and 6) Implementation and dispatchment. Usually, a written specification of the first four steps is drawn, while the last two steps are set during implementation or short pre-implementation tests. For the software under consideration, such a specification of the first four steps, with an outlook to the fifth, has been reviewed by (some) program users, so a thorough design document was available at the start of the implementation.

Although this specification was very much aimed at the *solution* — the data structure and visual appearance of the new system — the *fundamental reason* for change was also addressed, and recognized by the reviewers. So, in principle we have no reason to doubt a warm reception by the community of PIAS users, although we always have to wait and see how the market reacts when such a plan is actually put into software form, including a price tag (no matter how modest). A possible reluctance could be caused by the concern that calculation times may increase, because where previously a few intermediate stages of flooding were considered to be sufficient, today, in time domain mode, some hundreds of time steps will be calculated. Indeed, here and there processing time may increase, although some counterarguments can be made as well:

- Determining (temporary) equilibrium in each time step requires iterations, and hence computing time. However, each iteration starts with as initial condition the previous time step, which in general will not differ much from the final condition in that step. So, the number of iterations *per step* will be less than with conventional stages of flooding.
- For a ship with a device that acts for cross-flooding, the time to equalization should be determined anyhow. That used to be done in a separate assessment, with the outcome to be processed manually. So we can ask the user what he or she prefers; spending manual time or let the computer do the work?
- In PIAS many intensive computation tasks are spread over multiple cores or threads. That will also become available for these *Consecutive Flooding* computations, so up to a maximum of (today) twenty damage stability computations can be performed simultaneously.
- Experiments have indicated that total computation time on one core of a single time domain computation consisting of roughly 1000 time steps takes some 5-20 seconds, depending on level of detail of hull form and compartments. Concentrating on Probabilistic Damage Stability, in combination with an efficient method for determining the probability of damage p_i —such as by *Numerical Integration*, Koelman (2006), a variant of what is nowadays called the *Monte Carlo Method* — this delivers a good balance between accuracy and processing time.
- Indeed, Braidotti & Mauro (2019) proposed a computational more efficient method for progressive flooding, at the cost of increased algorithmic complexity. If computational costs of the internal flooding in time domain would become an issue, such a method could be investigated. If..would..could, because a sensible adage in software development is to only solve real problems.

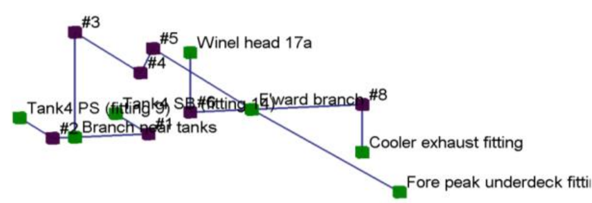
Appraisal by national authorities and classification societies

It could have been an option to distribute mentioned specification to inspection bodies as well, for their comments or advices. However, the question is what we would have gained by this. Because it's been quite a few years now, when SARC developed another new piece of software, on heeling and stability around an arbitrary axis. Similarly, we had drawn up a rather detailed elucidation of the proposed computational approach, including algorithms and references to specific and high-level literature, such as Pawlowski (2016). For comments we had sent that to a few inspection bodies, from whom most did not reply, while one wrote a short email reading “we use computer program ABC, and if your software provides the same results as ABC then we agree”. Although multiple normative, administrative and practical comments can be made regarding this statement, this particular paper is not the place for that. Let it suffice that we trust the reader appreciates that we lost appetite for this path.

Fortunately, the design of our piping software, and its underlying algorithms, data structures and choices are rather well documented, by its specification document, this paper and a bit more extensively in the manual. Additionally, on request some more design considerations or sketches from the implementation records can be made available. That should enable inspection bodies to perform their independent verification of the proper functioning of the software. A reassuring idea, incidentally, is that a time domain calculation is explicitly supported by regulations such as MSC (2020).

Concerning the ship design verification (i.e. the appraisal of damage stability computations), the obviousness of the ship model will only improve. Where with elder PIAS’ compartment connection facility the physical reality was to be translated into tables with numbers from virtual ‘connection’, now the basis is a topological and geometrical model which is easy to verify, as demonstrated by the report of the input from Figure 5.

Network: Combined vent tanks 4 PS and SB



Part of system: De-aeration

Connections in this network

Name	Type	Connected with	Cf	Area	Length	Breadth	Height
Cooler exhaust fitting	Equipment	Cooler expansion	0.350	400.00	75.000	3.000	5.000
Branch near tanks	Branch	-	0.600	400.00	50.000	-2.000	1.600
Tank4 SB (fitting 14)	Compartment	DB 4 SB	0.350	400.00	45.000	1.000	1.600
Tank4 PS (fitting 9)	Compartment	DB 4 BB	0.350	400.00	45.000	-2.900	1.600
F'ward branch	Branch	-	0.750	400.00	75.000	-1.500	7.000
Winel head 17a	Open opening	-	0.900	400.00	75.000	-4.000	9.500
Fore peak underdeck fitting	Compartment	Voorpiek	0.350	400.00	90.000	0.500	6.000

Segment: Branch SB tank

Component	Name	Cf	Area	Length	Breadth	Height
Connection start	Tank4 SB (fitting 14)	0.350	400.00	45.000	1.000	1.600
Waypoint	#1	0.350	400.00	50.000	1.000	1.600
Connection end	Branch near tanks	0.600	400.00	50.000	-2.000	1.600

Segment: Branch PS tank

Component	Name	Cf	Area	Length	Breadth	Height
Connection start	Tank4 PS (fitting 9)	0.350	400.00	45.000	-2.900	1.600
Waypoint	#2	0.350	400.00	50.000	-2.900	1.600

Figure 5: Small part from program output of a piping network

FURTHER DEVELOPMENTS

The software system as described offers these advantages:

- Physics-based computations on basis of a model which includes topology, geometry and true features of the piping system. So, no artificial notions or concepts; WYSIWYG (What You See Is What You Get).
- Intuitive definition of compartment connections, by just adding shape and types of pipes and their components.
- As a result, interfacing with other CAD/CAE software is in principle doable, as the same concepts (pipe shape and connections) form their basis.
- Both conventional (stages of flooding) and time domain damage stability computations.
- Easy generation of complicated stages of flooding.

The next developments of this system will comprise:

- Time domain computations *integrated* in Probabilistic Damage Stability. So, the effect of exceeding maximum allowable equalization time (for example 10 minutes, for SOLAS 2020) can automatically be determined, and incorporated in the table of results of damage cases and their *a* probabilities, and ultimately in the Attained Subdivision Index.
- With damage *to* pipes. For example, in PIAS' Probabilistic Damage Stability module the damage cases are presently generated on the basis of compartment boundaries. This will be extended to include piping edges and corners as well.

These are internal developments, which can be realized at SARC, independent of others. It is different when exchanging data with other CAD/CAE software. Ideally, this would be elaborated on the basis of a common data standard, which includes geometry, connectivity as well as functionality (e.g. the opening pressure of a pressure relief valve), while still being easy to manage and not too elaborate. Preliminary investigations have indicated that such a standard is not available, at least not widely used in the maritime domain. So, finding a solution is a challenge, which is at present being undertaken within a consortium where shipyards, universities and software suppliers are collaborating to develop a next-generation maritime software framework, SEUS (2023). In that context, lately we have been experimenting with a Graph Database, see Koelman et al. (2024), which most probably offers the flexibility to represent the piping data structure as described in this paper. This flexibility goes so far that, in addition to database capacity for pipe geometry, components and connections, there is also capacity for other facets of the pipeline system, such as its requirements and functions, as well as the design validation. This opens the door to a representation of the piping system which is extended with concepts of Systems Engineering — functional definition, physical definition, design validation (see Kossiakoff et al. (2011) — and hence support for a Model-Based Systems Engineering framework.

Where our computation method for the modelling of the flooding process will be adequate for many practical applications — and offers the advantage of traceability — it is not at the technological cutting edge. As SARC does not have the ambition to extend to more enhanced computation methods —such as a 6-DOF motion analysis, or the effects of waves, ship's inertia and the momentum and sloshing of the ingressed water — there lies an opportunity for interfacing with specialized third-party computational analysis software that covers this area.

CONCLUSION

In this paper a new system for modelling geometry and connectivity of pipes, ducts and openings is presented, including the application of these data in damage stability analyses. The status is that this is fully implemented in PIAS' deterministic and probabilistic damage stability modules, including the corresponding manual pages. After a small addition (consisting of some more intermediate results) it will be ready to be distributed for production use. That will be the first independently usable and useful release of PIAS' *Consecutive Flooding* system, including both the systems of delay factors (an extension of stages of flooding) and time domain. On this basis, more extensions will follow, either on the computational side, such as time domain integrated in Probabilistic Damage Stability, or on interfacing with external software. From the user's perspective this new software tool will come with some uncertainties —such as how to determine a crisp equalization time in the asymptotic flooding process, or how to cope with differing permeabilities for the same space — as elaborated in this paper. However, one must realize that these ambiguities have always existed and are now surfacing through the use of sophisticated software. The hope and expectation is that regulators will eventually make responsible choices here.

Occasionally we are asked to indicate the effect of this new software on a particular ship design, however, the prime motive for this new development lies in the process, not in optimizing the computational result. Having said that, it may be expected that by not considering all flooding as a priori progressive, the conclusion of some computations may switch from 'capsized or sunk' to 'survived'. In any case, combined with an efficient non-zonal computation method to determine the probability of damage p_i, results of the probabilistic damage stability assessment will be more in line with reality.

Perhaps, some day after the mentioned extensions have been implemented, we could make some comparative computations with and without Consecutive Flooding, with realistic specimen of ship designs. On the other hand, that would be mere examples, with little generic value. Just examples, while contemporary literature in our profession is already littered with examples of specific computations applied to specific ships. The new software presented in this paper is so easy to use that users will be able to create their own examples.

REFERENCES

Biran, A. & Lopez-Pulido, R. (2014). Ship Hydrostatics and Stability, 2nd ed. Butterworth-Heinemann.

Braidotti, L. & Mauro, F. (2019). A new calculation technique for onboard progressive flooding simulation, Ship Technology Research, 66(3), 150-162.

CESNI (2019). European Standard laying down Technical Requirements for Inland Navigation vessels (ES-TRIN). European Committee for drawing up Standards in the field of Inland Navigation (CESNI).

De Koningh, D., Koelman, H.J. & Hopman, J.J (2011). A Novel Ship Subdivision Method and its Application in Constraint Management of Ship Layout Design. Journal of Ship Production and Design, 27(3), 137-145.

Dexpi (2019). DEXPI P&ID Specification Version 1.3. https://dexpi.org/wp-content/uploads/2020/09/DEXPI-PID-Specification-1.3.pdf.

Gero, J. S., & Kannengiesser, U. (2004). The situated function–behaviour–structure framework. Design Studies, 25(4), 373–391.

Horváth, I. (2007). Comparison of Three Methodological Approaches of Design Research. International Conference on Engineering Design. ICED 07, (7), 28–31.

IMO (2016). International Code for the Construction and Equipment of Ships carrying Dangerous Chemicals in Bulk. International Maritime organization.

IMO (2020). SOLAS Consolidated Edition, chapter II-1, part B. International Maritime organization.

ISO (2019). Industrial automation systems and integration – Integration of lifecycle data for process plants including oil and gas production facilities – Part 4: Initial reference data, ISO/TS 15926-4.

Karolius, K.B., Cichowicz, J. & Vassalos, D. (2019). Modelling of compartment connectivity and probabilistic assessment of progressive flooding stages for a damaged ship. 17th International Ship Stability Workshop.

Koelman, H.J. (2006). A new method and Program for Probabilistic Damage Stability. Ship Technology Research 53:183–193.

Koelman, H.J., Veelo, B.N., Seppälä, L. & Filius, P. (2024). Closing the Gap between Early and Detailed Ship Design Models. 15th International Marine Design Conference (IMDC), Amsterdam, Netherlands, June 3-6.

Kossiakoff, A., Sweet, W.N., Seymour, S.J. & Biemer, S.M. (2011). Systems Engineering Principles and Practice. John Wiley & Sons, New Jersey, USA.

MSC (2020). Resolution MSC.429(98)/rev.1: revised explanatory notes to the SOLAS chapter II-1 subdivision and damage stability regulations.

Pawlowski, M. (2016). The Stability of a Freely Floating Ship. Polish Register of Shipping, Technical Report 72. www.prs.pl/uploads/tr_no_72.pdf.

Ruponen, P. (2006). Pressure-Correction Method for Simulation of Progressive Flooding and Internal Air Flows. Ship Technology Research, 53, No.2, 63-73.

Ruponen, P. (2007). Progressive flooding of a damaged passenger ship. Doctoral dissertation. Helsinki University of Technology.

Ruponen, P., Queutey, P., Kraskowski, M., Jalonen, R. & Guilmineau, E. (2012). On the calculation of cross-flooding time. Ocean Engineering, 40, 27-39.

Ruponen, P., Manderbacka, T. & Lindroth, D. (2018). On the calculation of the righting lever curve for a damaged ship. Ocean Engineering, 149, 313-324.

SEUS (2023). Smart European Shipbuilding. Horizon Europe Framework Programme. www.ntnu.edu/seus.

Söding, H. (2002). Flow computations for ship safety problems. Ocean Engineering (20), 7, 721-738.

Vermeer, H., Vredeveldt, A.W. & Journée, J.M.J. (1994). Mathematical modelling of motions and damaged stability of ro-ro ships in the intermediate stages of flooding, STAB'94 Conference, Melbourne, USA, Nov. 7-11.

White, F.M. & Xue, H. (2021). Fluid mechanics. 9th ed. Mcgraw-Hill.