Procedia Computer Science
Volume 29, 2014, Pages 1688–1700
ICCS 2014. 14th International Conference on Computational Science

On-board Decision Support System for Ship Flooding Emergency Response

J.M. Varela[1*], J.M. Rodrigues[1†] and C. Guedes Soares[1§‡]
[1] *Centre for Marine Technology and Engineering (CENTEC), Universidade de Lisboa, Lisbon, PORTUGAL*
[§] *Also Distinguished Research Professor ITMO University, Saint-Petersburg, Russia*
**varela@centec.tecnico.ulisboa.pt, †miguel.rodrigues@centec.tecnico.ulisboa.pt, ‡c.guedes.soares@centec.tecnico.ulisboa.pt*

Abstract
The paper describes a simulation system to support emergency planning decisions when ship flooding occurs. The events of grounding and collision are considered, where the risk of subsequent flooding of hull compartments is very high, and must be avoided or at least minimized. The system is based on a highly optimized algorithm that estimates, ahead in time, the progressive flooding of the compartments according to the current ship status and existing damages. Flooding times and stability parameters are measured, allowing for the crew to take the adequate measures, such as isolate or counter-flood compartments, before the flooding takes incontrollable proportions. The simulation is supported by a Virtual Environment in real-time, which provides all the functionalities to evaluate the seriousness and consequences of the situation, as well as to test, monitor and carry out emergency actions. Being a complex physical phenomena that occurs in an equally complex structure such as a ship, the real-time flooding simulation combined with the Virtual Environment requires large computational power to ensure the accuracy of the simulation results. Moreover, the distress normally experienced by the crew in such situations, and the urgent (and hopefully appropriate) required counter-measures, leave no room for inaccuracies or misinterpretations, caused by the lack of computational power, to become acceptable. For the events considered, the system is primarily used as a decision support tool to take urgent actions in order to avoid or at least minimize disastrous consequences such as oil spilling, sinking, or even loss of human lives.

Keywords: Decision Support Systems, Ship Damage Control, Virtual Environments, Flooding Simulation

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2014

doi:10.1016/j.procs.2014.05.154

1 Introduction

Security and safety on-board ships has always been one of the primary concerns in Maritime Industry. Ultimately, the design of a ship is itself a compromise between the safety and the capability of effectively fulfilling the mission to which the ship is built. Ship systems, structures, equipment and spaces are designed and produced in order for the ship to obtain the best possible performance in safe and secure conditions. However, history has proven that this is not always the case. Despite of all the modern safety systems and equipment on-board, the risk of accidents is always present, and many have occurred with serious consequences such as the sinking of ships, loss of human lives and irreversible costs to the natural environment (Guedes Soares and Teixeira, 2001). Even when dealing with the supposedly most advanced safety systems and modern ship designs, the accident can always occur as recently shown by the collision and grounding case of the cruise ship, *'MS Costa Concordia'*, (Schröder-Hinrichs *et al.*, 2012). Even when navigating in open seas, where there is no risk of collision or grounding, accidents still occur too frequently due to severe sea states, as described in Guedes Soares *et al.* (2001).

When the ship is at sea, it may be regarded as an independent isolated system that can only rely on its own means to maintain safety in what can be an extremely adverse environment, such as the ocean. In most of the cases, when accidents occur with consequent flooding of compartments, salvage is not possible within the next hours. Therefore, it is very important to keep stability as intact as possible to allow enough time for external help to arrive, or at least to evacuate the ship in the best possible safety conditions.

Decision-support systems for such situations, must anticipate the status of the ship in advance. This can be done through time-scaled simulations in real-time. Within this scope, the scale results in a fast-forward simulation, which implies that the time required to compute one simulation step, is smaller than the simulation time-step. As the increase of the simulation time-step normally decreases the accuracy of the results (which is not acceptable for this case), the solution to achieve reliable results is to reduce the calculation time through intensive and, most often, distributed computation.

Using Virtual Reality techniques to improve the efficiency of such systems has been recognized to be of great help by Beroggi *et al.* (1995). Since then, and largely due to the extraordinary evolution of Graphics Processors, mainly in the last twenty years, simulation and decision support systems in the maritime industry have gradually adopted 3D visualization and interaction (Varela *et al.*, 2011). The majority of these systems focus three main emergency situations: passenger evacuation, fire and flooding propagation. It is the case of th*e maritmeEXODUS* and the *SMARTFIRE* systems described in Galea *et al.* (2003) for simulating passenger evacuation under fire conditions. More recently, the VELOS system described in Ginnis *et al.* (2010), is also based on a VR platform to analyze the urgent evacuation of passengers. Tate *et al.* (1997) demonstrated that using Virtual Environments for training and mission rehearsal improves the efficiency of firefighters in fighting real on-board fires. Concerning the ship flooding simulation, Varela and Guedes Soares (2007) describe a VR based Decision Support System to assist the coordination of damage control teams and to take the appropriate counter-measures in case of flooding fire or contamination onboard military vessels.

For the specific case of progressive flooding, fast-forward simulation allows the operator to check which compartments will be flooded, the order by which they will flood, and how much time they will take to be flooded. Ideally, the simulation must complete in time for the operator to take the adequate counter measures before the ship enters into an irreversible capsize or sinking situation. Within this scope, the simulation ends when the stability is recovered or when the ship capsizes.

The current paper presents an on-board decision support system for ship flooding emergency response. The system runs a fast-forward simulation of the progressive flooding of ship compartments given an initial condition, which includes the current load and damage conditions of the ship. From these, the progressive flooding algorithm carries out the flooding sequence taking into account the position and shape of the damages, the inter-compartment openings and each compartment's particular geometry and internal arrangement. A quasi-static approach, as in Ruponen (2007), is followed, which means that the solution is based on hydrostatic theory and the ship reaches its instantaneous equilibrium position, corresponding to the addition of the weights, buoyancy and flood water induced forces, at each time step. All these forces are assessed using a generalized form of the pressure integration technique described firstly by Witz and Patel (1985), extend in Schalck and Baartrup (1990) and implemented on the studies of Santos and Guedes Soares (2001, 08, 09). The flow at each opening is governed by the balance of pressures on each side; full compartments are addressed by means of an improved version of the propagation-of-flow algorithm presented by Dankowski (2012). This algorithm balances to zero the flow on such compartments resulting in pressures higher than the atmospheric inside these. The updated condition of the ship and the levels of water inside each compartment constitute the output..

The software system is composed by two main modules: the Virtual Environment and the Progressive Flooding. Depending of the complexity of the ship, both modules can be very heavy from the computational point of view, and therefore, for this case, intensive distributed computation is used. The Virtual Environment pushes the Graphics Processing Units (GPUs) to their limits in order to provide a realistic interactive environment where the user is able to navigate, to modify and effectively obtain information about the ship status. The Progressive Flooding module contains the physical algorithm whose computation is distributed by the available Central Processing Units (CPUs). Currently the distributed architecture using powerful core units is the only way of obtaining reliable flooding simulations in fast-forward time

2 System architecture

The architecture of the system is presented in Figure 1.

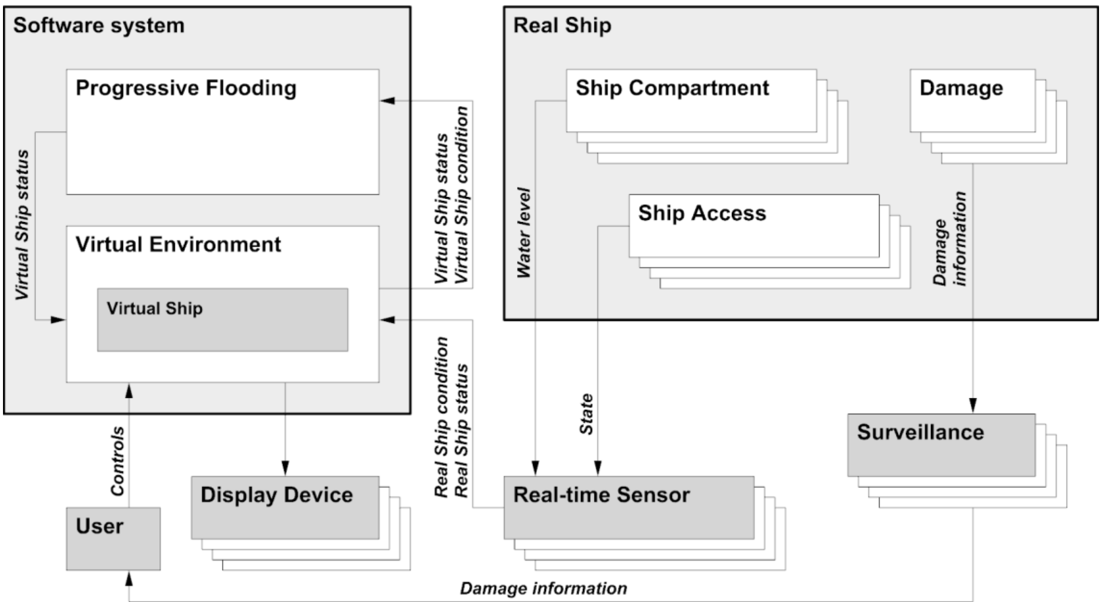

Figure 1: The software system is composed by the Virtual Environment and the Progressive Flooding modules, which access the real ship condition through real-time sensors and surveillance systems.

In order to fully understand the diagram, some concepts must be clarified within the scope of the presented system. Ship compartments are in this case watertight spaces in the interior of the hull connected by watertight accesses like doors and hatches. The state of an access may be opened or closed allowing or not the propagation of the water through the ship compartments. Damages in this case are typically holes in the hull caused by collisions or groundings that will trigger the flooding of a ship compartment. The damage information is the dimensions, shape and location of the holes, which will influence the flooding progression. The concept of ship condition in this scope defines the state of all accesses, levels of water in compartments and damages' characteristics, while the ship status refers to the stability condition and is defined by the draft, trim and heel. Finally, the virtual and real ship refer to the ship in the simulation system and to the ship in the real world respectively. Excluding the starting time of the simulation, the virtual and real ship conditions and status do not have to, and will not be coincident most of the time.

In a real situation, the system is typically installed in the Damage Control Room for the case of military vessels or in the navigation bridge for other types of ships. In order for the system to work with its full capabilities, a detection system composed by sensors at the compartments and watertight accesses such as doors and hatches must also be installed in the ship.

As depicted in Figure 1, three main elements are considered in the real ship: the compartments, the accesses between compartments and the damages. In a flooding situation, watertight compartments and accesses between them are the most significant elements for the progression of the flood. Sensors are installed to detect the water level inside each compartment and the current of state of watertight doors, hatches or valves. In order to run the simulation, ship damages like holes in the hull, must also be identified and inputted into the system. For this case, real-time sensors like the ones mentioned before, are not able to provide damage information such as dimension, shape or location of the holes. This can only be achieved by camera surveillance or by visual inspection. Sensors will send the current ship condition and status in real-time to the simulation system. Information collected by the sensors is sent to the simulation system and updated accordingly in real-time. However, damage information, either recorded by cameras or provided by emergency crew members must be inputted manually by the user. Therefore, the Virtual Environment must provide quick and efficient methods to input damage information into the simulation system. The ship status and condition, including the damage information, is the required/sufficient information to start the simulation.

Within the software system, two main modules must exist: the Virtual Environment and the Progressive Flooding. The exchange of data between the simulation system and external elements such as sensors, display devices or the user, is achieved uniquely through the Virtual Environment. It provides all the necessary features to setup, start, control and visualize in real-time the course of the simulation. A virtual model of the ship containing all the compartments with accesses and their underlying topology as defined in the flooding mathematical model, is built and connected to the progressive flooding algorithm. According to the input data provided by the Virtual Environment, which includes the ship status computed in the previous simulation step, plus any updates to the ship condition, the flooding algorithm computes and sends the new ship status to the Virtual Environment.

3 The Virtual Environment

As mentioned in the previous section, the Virtual Environment establishes the communication between the user, the simulation algorithm and the real ship. Therefore, the effectiveness of the system depends largely of an appropriate set of functionalities that this tool is able to provide. An effective and quick way of defining the initial conditions to test different counter-measures, or the recording of the tested procedures and actions are just an example of two crucial functionalities that the Virtual Environment must provide. Next, some of the most important functionalities are presented.

3.1 Visualization of ship compartments

The main focus of the Virtual Environment is a realistic 3D representation of the ship. All compartments, structures and equipment that may influence Damage Control operations, are modelled.

A flythrough camera navigation system is implemented allowing the user to visualize the ship from any point of view. As the scenario is only composed by the ship, some constrains are imposed to the navigation system, allowing only the camera movements specified in Figure 2. It should be noted that the system is to be used by operators which may not be familiarized with Computer Graphics or Virtual Reality applications, and therefore, a totally free flythrough camera could easily lead the observer to get lost in the scenario.

The camera target is allowed to move along a longitudinal line segment defined at the center plane on the top of the topmost visualized deck. The camera is allowed to move vertically along an imaginary transversal circumference, and horizontally along an ellipse with the longitudinal radius equal to half the length of the ship. A minimum value of the circumference radius defines an interdicted volume around the ship where the camera is not allowed to enter.

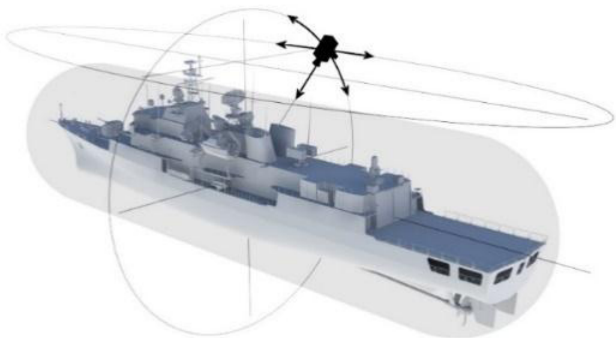

Figure 2: A flythrough camera navigation system with specific constrains to the movements of the camera is implemented. This allows the user to visualize the scenario from any point of view without losing the main focus of the simulation: the ship.

Figure 3: The deck-based visualization approach allows to observe lower decks by hiding the upper ones.

Due to the eventual complexity of the ship, a deck-based visualization approach was adopted. Upper decks may be hided in order to unhide the lower ones as presented in Figure 3. This requires that each object is associated to the deck where it is located.

3.2 Evaluation of risk areas

Using only the Virtual Environment and the ship condition provided by the real-time sensors or inputted manually, the operator may perform useful queries to assist her/him on the coordination of emergency teams. Three main functionalities are currently available: identification of risk zones, compartment isolation and identification of evacuation paths. These functionalities use the topology and the associations between virtual objects to achieve the final results.

The identification of risk zones is used to find which compartments are in risk of flooding for the current ship condition. From this, the operator has an initial estimation of the volume that may be flooded (with the consequent loss of buoyancy) even without running the flooding simulation. Moreover, the operator may also check if crucial compartments for ship operation are in risk. For military vessels, these may include communications, damage control, ammunition or power supply rooms. If this functionality is used in conjunction with the simulation algorithm, then additional information may be obtained such as the time before each compartment is partially or totally flooded. Figure 4 (left) presents the output of this functionality when a specific compartment (in green) is identified as a risk zone.

Figure 4: Zones with risk of flooding may be identified based on the current ship condition. Compartments are highlighted and the operator gets an initial estimation of the total zone that is in risk. Then, evacuation paths may be created between two compartments based on the current ship condition and concerning the risk zones that may exist along the way.

Compartment isolation allows the operator to isolate a specific zone to avoid the propagation of the flood through the compartments. This can be done automatically through the Virtual Environment by identifying the compartment to be isolated. The identification of evacuation paths allows creating safe paths between two compartments. The system takes in consideration the risk zones and tries to avoid them. However if there is no other alternative, then the path may cross these zones but a warning is triggered. Once more, if this functionality is connected to the simulation algorithm, then the system is able to compute how long a certain evacuation path will remain safe. This can be used to coordinate the evacuation of passenger on cruise ships or to guide emergency control teams in military vessels when for instance the visibility conditions are very low due to smoke. Figure 4 (right) presents a simple path between compartments with different tonalities depending of the risk level in each compartment.

3.3 Simulation setup and control

In order to obtain a reliable estimation of the ship status ahead in time, the initial setup for the simulation implies that the virtual ship condition and status are as close as possible to the real ship. In an emergency situation, the amount of input from the operator to setup the initial conditions for the simulation should be as low as possible. This reduces the time spent to setup the simulation and avoids human errors on the introduction of data. Therefore, most of the data required to run the simulation should be provided by the real-time sensors, which include the water level at each compartment, the state of accesses (opened/closed) like watertight doors and hatches, and the ship status (trim, heel and draft). It is supposed that hydrostatic and hydrodynamic parameters of the virtual ship, such as the Center of Gravity, Moments of Inertia, Added Masses, etc. are always updated in the system. However, if this is not the case, the system allows to change them manually before running the simulation.

The trickiest and error prone part of setting up the simulation is the input of the damage information. As mentioned in section 2, real-time sensors do not detect the location or characteristics of holes in the hull, and therefore, this information can only be provided by surveillance cameras or by visual inspection. The Virtual Environment provides functionalities to specify directly in the virtual ship the location and dimensions of the holes in the hull (Figure 5, top-left).

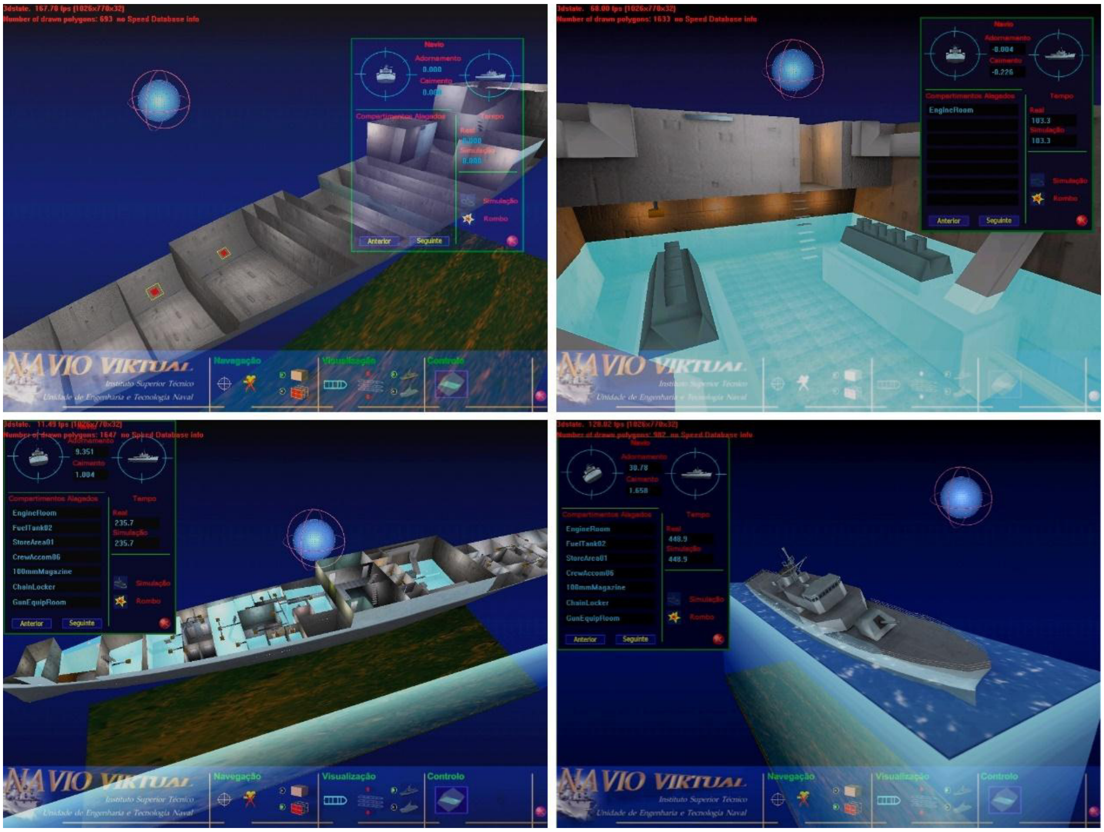

Figure 5: The Virtual Environment provides functionalities to specify new holes directly in the virtual ship before and during the simulation. Ship status and condition are updated both graphically and by numeric values in the interactive menus.

Graphically, water is represented in flooded compartments at the level provided by the sensors or computed by the flooding algorithm, and the position of the ship is updated according to its status, as presented in Figure 5.Interactive menus display the values of flooding percentage and volume in each compartment and the values of ship heel and trim.

3.4 Simulation reports

Beyond the display of the virtual scenario, the system also produces simulation reports containing the relevant events during the simulation. The main purpose of such reports is to provide a procedure guide to achieve the same result in the real ship as the one achieved in the fast-forward virtual simulation. Therefore, a time stamp is assigned to each event. Additionally to the events, the system also reports the ship status with a periodicity defined by the user. Table 1 lists the relevant events considered by the system with the corresponding significant information.

Reported Event	Significant Data
Change access state (open/close)	Access ID/Name Adjacent compartments
Ship status update	Trim, Heel, Draft
New compartment flooded	Compartment ID/Name

Table 1: Reported events during the simulation are recorded and reported by the system with a time stamp.

Simulation reports allow checking the correctness of the simulation before the final result is achieved, and evaluate if a new corrected simulation is required.

4 The progressive flooding algorithm

The design of a progressive flooding algorithm, intended for a fast forward prediction computation presented in this paper, is governed by the need to attend the required computational overhead related limitations, while still producing reliable and realistic results in light of its relative significance to the decision support system. In this section, several approaches to the solution of the problem of predicting the behavior of a damaged ship subject to progressive flooding are briefly depicted, the simplifying assumptions and their validity are discussed and the characteristics of the algorithm implemented in the present decision support systemare listed.

4.1 Field methods

The application of *Computational Fluid Dynamics (CFD)* codes which consider the full physics of the problem with relatively minor simplifications, such as *Reynolds Averaged Navier-Stokes (RANS)* turbulence modelling, to the entire domain of the problem, constitute a cumbersome solution. Such codes pollute the system with unnecessary detail, greatly increase the complexity of code development and typically require a prohibitive amount of processing power. Additionally, there is a necessity to introduce, *a priori*, a 3D mesh distributed through all the domain where the flow governing equations are to be evaluated (*CFD* mesh modeling is known to be the most time consuming task in a typical study). In practice, these methods are applied when the exact free-surface flow around a moving ship needs to be assessed (e.g.: resistance prediction and wake modeling), such as in Ciortan et al (2007), to particular details of a flow problem to be posteriorly used in simpler algorithms – see Wood et al. (2010) - or to serve as benchmark for less complex, and faster, codes being studied. A good example of such benchmark studies is the one carried out by Torres et al. (2008).

4.2 Simplifying approaches

A simplifying step, relative to full *RANS*, is to limit its domain of application to pertinent zones of the problem, namely: to a limited area around the hull as done by Greco et al. (2014), who considered simpler formulations to the remaining areas, or, when focusing on damaged ships analysis, to the interior of flooded compartments, as in Gao et al. (2010). Further on the simplification path, Santos and Guedes Soares (2008) implemented a *Shallow Water Equations (SWE)* solution to the problem of predicting the water flow inside flooded compartments. The *SWE* simplifies the problem by assuming potential flow – incompressible fluid without viscosity or eddy making – and a zero influence of the vertical acceleration of the fluid's particles on the equations which rule the flow. A 3D grid, inside the compartment, is necessary for the numerical scheme. This poses an additional complication to the usual *Volume of Fluid (VOF)* spatial discretization, used in *RANS* models, which is able to handle irregular tetrahedral elements' meshes, even though a significant increase in computational speed is accomplished in a grid type mesh.

Although of increasing simplicity and computational speed, all these methods are far from being able to comply with fast computation, in real time, and lag tremendously. However, if one decouples the effect of the ship's advancing speed and of the waves from the ship motion and flooding, a simpler physics problem is set to be addressed. This approach is not to be seen as unrealistic, once one realizes that a damaged vessel will immediately stop and that the average position of the ship, about which it oscillates, is governed by the flooding in still waters. Although flooding may occur solely due to a wave, which only surpasses at its crests an existing damage opening, the following flood will soon bring down the exposed breach, if the process is not interrupted.

Despite having simpler physics, the still water approach, on its own, does not suffice for fast computations; a *RANS* method will still lead to intensive computational time – see Gao et al. (2010). A process know to work fast is to implement a quasi-static motion algorithm which reacts instantly to the progression of the flow computed at the openings, in each time-step. This approach has been successfully implement by Ruponen (2007) and Dankowski (2012), where both have made use of commercial software to carry out the flooding of the compartments, focusing on the computation of the flow between compartments. The flow is addressed by evaluating the pressure differential between both sides of the opening.

The quasi-static approach is realistic when applied to ships of significant length, which do not oscillate heavily due to a sudden ingress of water in one of its several compartments. The absence of dynamic effects which convey inertial effects between time-steps, gives way to the disposal of the need to assess the added masses, a set of coefficients which introduce the inertial effect of the ship's surrounding water disturbance in its motion, the damping coefficients, a set of dissipative constants responsible for considering the effect of the radiated waves originated by the motion of the ship in an, otherwise, still water surface, and the impulse originating from the sudden change of mass at a flooding compartment.

4.3 Present algorithm

The progressive flooding algorithm which has been implemented in the present system, follows the quasi-static approach and the pressure differential calculation at the inter-compartment openings, described above. Floodwater is considered as a static, time variant, force, i.e., no added inertia is present due to the flood water variation per time step and all waterlines, inside and outside of the ship, remain horizontal at all times. The effect of trapped air being compressed due to rising water inside a closed compartment is neglected. The trapped air would induce a damping effect to the inflow of water, but only if the velocity of the rising waterline is considerable, and the existing ventilation openings are not enough to properly perform their task of allowing air to escape. Inter-compartment air flow is also neglected, assuming that if the previous assumption is valid for a single compartment, the same is true for a network of connected compartments.

The flow calculation formulation is segregated in two cases, as is shown in Figure 6 (left). The first case relates to when both sides of an opening are immersed and both sides' waterline height induce a hydrostatic pressure; the second case is when one of the sides exhibits a waterline height higher than that of the opening, while the other does not. Analyzing the second case in Figure 6 (left), it becomes evident that the height of the latter is of no importance for the pressure differential computation, as at such side the pressure is kept equal to the atmospheric, due to the, previously stated, neglecting of any air pressure change inside the compartments.

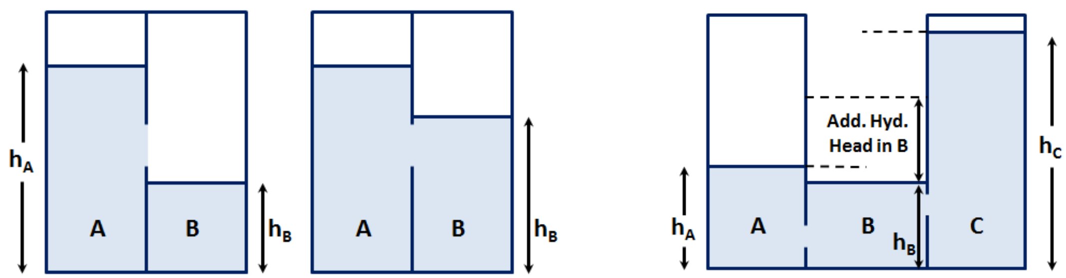

Figure 6: The status of the opening determines which formulation is applied for the flow calculation (left); full compartments must be specially addressed (right).

When one or several compartments reach a full condition, the straightforward application of the pressure differential at each opening is not enough for a consistent progressive flow scheme. Looking at Figure 6 (right), the hydrostatic pressure differential would result in water flowing from A and C compartments to the, already full, compartment B, which is nonsense. To work around this issue, an intermediate step, which introduces a balancing of all connected compartments' pressures and governed by the imposition that the balance of inflow and outflow at already full compartments must be equal to zero, must be introduced – the output is an additional hydrostatic pressure head, which is applied to B. Details on the formulation of this intermediate step may be found in Dankowski (2012).

Finally in what relates to the calculation of water levels and of hydrostatic forces on the structure, the algorithm performs an iterative scheme converging to zero residual force/moment, at each time step, by using an adaptive mesh cutting procedure as presented in Figure 7.

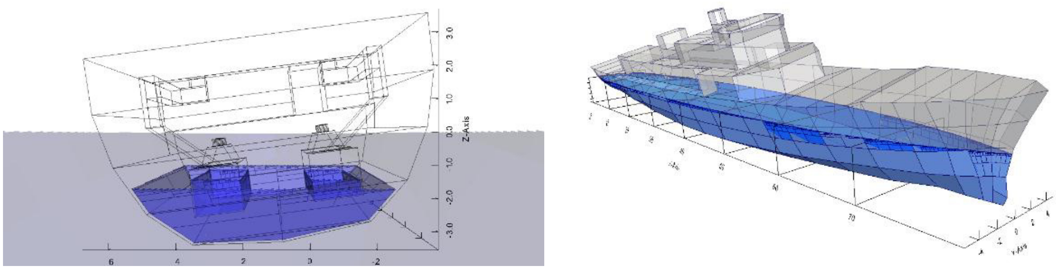

Figure 7: The adaptive mesh iterates to the corresponding flooding water level inside the breached Engine room (left); the hull's wet surface corresponding to the vessel's instantaneous attitude is covered by the adaptive mesh (right).

The resulting adapted meshes cover all wet areas, including the immersed sides of the openings. A set of analytical, exact, expressions allow for the computation of the pressure at each mesh element. Integration of these pressures through the entire surface gives the resulting forces which are then applied to the vessel. The vessel responds, by changing its attitude and position, in order to achieve the equilibrium with the buoyancy force and the action of the mass weights on board. This element-wise pressure integration process is commonly known as the *Pressure Integration Technique*, described firstly by Witz and Patel (1985), extended by Schalck and Baartrup (1990) and implemented in the studies of Santos and Guedes Soares (2001,2008,2009).

In Figure 7 (left) the adaptive mesh generated on the partially flooded Engine Room of a listed corvette is shown; in Figure 7 (right) the same adaptive procedure is illustrated concerning the wet area of the same corvette subjected to significant trim and list.

5 Discussion

The study clearly revealed that an emergency decision support tool such as the one presented, benefits enormously with the use of a well-designed Virtual Environment. One can even say the without this Virtual Environment it is not possible to setup, control and visualize the simulation properly. The study also highlighted the difference, but at the same time the mutual support, between the Virtual Environment and the Simulation Engine. They were independent of each other and may run independently. However, by linking both, an improvement of both programs is achieved. The Virtual Environment becomes more real because it simulates a real phenomenon even if simplified, and the results of the simulation engine become more clear, understandable and easily manipulated. For this specific case, the connection of the Virtual Environment to the simulation algorithm proved to be fundamental for setting up, controlling and visualizing the results provided by the simulation: the user is able to analyse the flow of water between the compartments in real time and the effect on ship stability.

The use of advanced real-time visualization techniques and Graphical User Interfaces, consumes many computational resources not only from the CPUs but also from the GPUs. The number of virtual objects considered in the ship may easily reach some hundreds, and these must be continuously detected for possible selection by the user. On the other hand, for the physical algorithm to run in real-time, and in this case in fast-forward time, intensive computation is also required. For these two reasons, the system requires a robust and powerful hardware specification preferably with distributed computation that allows ultra-fast calculation both for the physical and graphical components.

Although the system was primarily conceived to be a decision support tool, it is planned in short term future to use it as a training tool. For this purpose, some modifications and some new features need to be added. A distributed virtual environment needs to be implemented with two slightly different versions of the application: one for the monitor and one for the trainee. Both will have the same features for changing accesses' states (opening/closing doors, valves, etc.), however, the monitor's version will have the ability to generate casualties and to start the physical simulation. The monitor version will also simulate the actions performed by damage control teams' according to the trainee's instructions.

Acknowledgments

The second author was funded by the Portuguese Foundation for Science and Technology (FCT), under the grant nr. SFRH/BD/64242/2009. The paper presentation is supported by the project Holistic Risks Studies for Complex Systems Sustainability funded by the Government of the Russian Federation, Grant 074-U01.

References

Beroggi, G., Waisel, L., Wallace, W., 1995, Employing virtual reality to support decision making in emergency management, *Safety Science,* 20:79-88.

Ciortan, C., Wanderley, J., Guedes Soares, C., 2007, Turbulent Freee-Surface Flow Around a Wigley Hull using the Slightly Compressible Flow Formulation, *Ocean Engineering*, 34 (2007): 1383-1392.

Dankowski, H., 2012, An Explicit Progressive Flooding Simulation Method, *11th International Conference on the Stability of Ships and Ocean Vehicles*, Athens.

Galea, E., Lawrence, P., Gwynne, S., Fillippidis, L., Blackshields, D., Sharp, G., Hurst, N., Wang, Z., Ewer, J., 2003, Simulating ship evacuation under fire conditions, *Proc. 2nd Int. Pedestrian and Evacuation Dynamics Conference,* Greenwich.

Gao, Z. Vassalos, D., Gao, Q., 2010, Numerical Simulation of Water Flooding into a Damaged Vessel's Compartment by the Volume of Fluid Method, *Ocean Engineering*, 37(2010): 1428-1442.

Ginnis, A., Kostas, K., Politis, C., Kaklis, P., 2010, VELOS: A VR platform for ship-evacuation analysis, *Computer-Added Design*, 42:1045-1058.

Greco, M., Colicchio, G., Lugni, C., Faltinsen, O. M., 2013, 3D Domain Decomposition for Violent Wave-Ship Interactions, *International Journal for Numerical Methods in Engineering*, 95: 661-684.

Guedes Soares, C., Bitner-Gregersen, E., Antão, P., 2001, Analysis of the Frequency of Ship Accidents Under Severe North Atlantic Weather Conditions, *Proc. of the Conference on Design and Operation for Abnormal Conditions II*, 6-7 November, 2001, RINA London, UK, pp. 221-230.

Guedes Soares, C., Teixeira, A.P., 2001, Risk assessment in maritime transportation, *Reliability Engineering and System Safety*, 74:299-309.

Ruponen, P., 2007, Progressive Flooding of a Damaged Passenger Ship, *Doctoral Dissertation*, Department of Mechanical Engineering, Helsinki University of Technology.

Santos, T.A. and Guedes Soares, C., 2001, Ro-Ro Ship Damage Stability Calculations Using the Pressure Integration Technique, *International Shipbuilding Progress*, 48(2/2001):169-188.

Santos, T.A. and Guedes Soares, C., 2008, Study of Damaged Ship Motions Taking into Account Floodwater Dynamics, *Journal of Marine Science and Technology*, 13:291-307.

Santos, T.A. and Guedes Soares, C., 2009, Numerical Assessment of Factors Affecting the Survivability of Damaged Ro–Ro Ships in Waves, *Ocean Engineering*, 36(11):797-809.

Shalck S. and Baatrup, J., 1990, Hydrostatic Stability Calculations by Pressure Integration, *Ocean Engineering*, 17(1/2):155-169.

Schröder-Hinrichs, J., Hollnagel, E., Baldauf, M., 2012, From Titanic to Costa Concordia – a century of lessons not learned, *WMU J Marit Affairs*, 11:151-167.

Tate, D., Sibert, L., King, T., 1997, Using virtual environments to train firefighters, *IEEE Computer Graphics and Applications*, 17(6):23-29.

Torres, F., Alho, A., Sales-Jr, J., Sphaier , S., Nishimoto, K., 2008, Experimental and Numerical Analysis of the Behavior of a Monocolumn with a Moonpool, *OMAE 2008 – Estoril, Portugal*, 57657: 291-300.

Varela, J.M and Guedes Soares C., 2007, A Virtual Environment for Decision Support in Ship Damage Control, *IEEE Computer Graphics and Applications*, 27(4):58-69.

Varela, J.M., Cacho, A.J., Guedes Soares, C., 2011, Virtual Environments for simulation and study of maritime scenarios, *Marine Technology and Engineering*, Guedes Soares *et al.* (Eds.), Taylor & Francis Group, London, ISBN 978-0-415-69808-5, pp. 719-740.

Witz, J.A. and Patel, M.H., 1985, A Pressure Integration Technique for Hydrostatic Analysis, *The Royal Institution of Naval Architects – Supplementary Papers*, 127:285-294.

Wood, C. D., Hudson, D. A., Tan, M., 2010, CFD Simulation of Orifice Flow for the Flooding of Damaged Ships, *The 13th Numerical Towing Tank Symposium – Duisburg - 2010*, 144-149.