Artificial Intelligence Research Issues in Computational Simulation of Physical System Behavior

Andrew Gelsey
gelsey@cs.rutgers.edu

CAP-TR-8
March 1992

Abstract

Computational simulation is an important tool for predicting the behavior of physical systems. Many powerful simulation programs exist today. However, using these programs to reliably analyze a physical situation requires considerable human effort and expertise to set up a simulation, determine whether the output makes sense, and repeatedly run the simulation with different inputs until a satisfactory result is achieved. Automating this process is not only of considerable practical importance but also raises significant AI research issues in the areas of spatial reasoning and deep models of expert reasoning about physics and numerical analysis.

Computer Science Department
Rutgers University
New Brunswick, NJ 08903

Contents

1 Introduction **2**

2 Domains **3**
2.1 Design of Racing Yachts 4
2.2 Clockwork Mechanisms 5

3 Processes to be Automated **6**
3.1 Setting up a Computational Simulation 6
3.2 Quality Assurance 7
3.3 Feedback 9

4 Artificial Intelligence Research Issues **9**
4.1 Spatial Reasoning 10
4.2 Deep Models of Expert Reasoning about Physics and Numerical Analysis 12

5 Experimental Results **13**

6 Related Work **17**

7 Conclusion **18**

8 Acknowledgments **19**

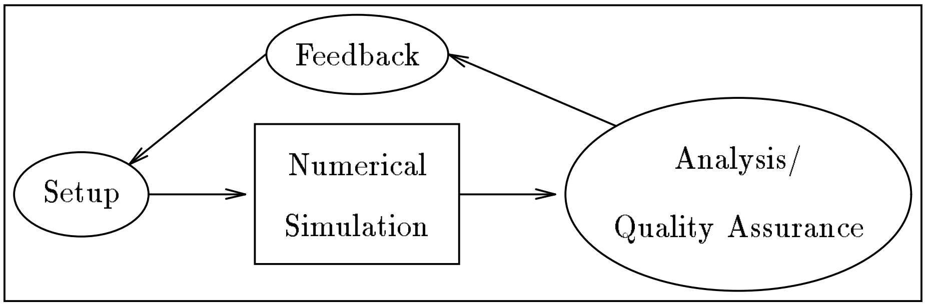


Figure 1: Analyzing a physical situation

1 Introduction

Computational simulation is an important tool for predicting the behavior of physical systems. Many powerful simulation programs exist today. However, as illustrated in Figure 1, using these programs to reliably analyze a physical situation requires considerable human effort and expertise to

- set up the simulation by transforming a description of the physical situation into a representation the simulation program can successfully process,
- analyze the output of the simulation program to extract desired information and in particular to
- determine whether the output makes sense and how accurate it is likely to be, and if the output is not acceptable, to
- determine how to change the simulation program's input so that it will more reliably predict the behavior of the physical system being analyzed

As a result, these simulation programs typically can't be run successfully by inexperienced users. Perhaps more importantly, these simulation programs can't be reliably invoked by other programs. For example, human designers of complex objects like ships and airplanes typically run sophisticated simulation programs to analyze the object's physical behavior, but an automated system for designing complex objects could not easily include such a computational simulation as part the process it uses to evaluate new designs.

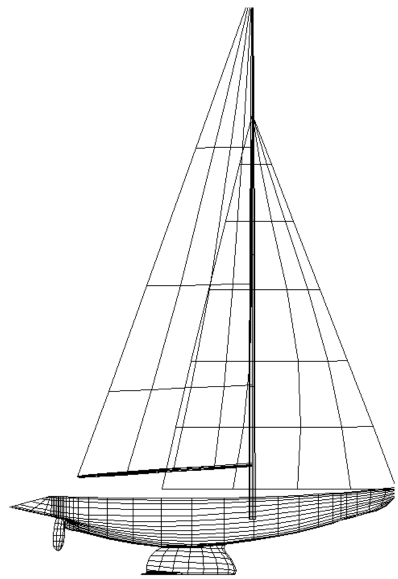


Figure 2: *Stars & Stripes*, winner of the 1987 America's Cup competition

Artificial intelligence techniques seem essential in order to automate the processes of setup, analysis, quality assurance, and feedback for computational simulation. However, simple application of known AI technology appears inadequate for the task of automating these processes. Basic AI research is needed, particularly in the areas of spatial reasoning and deep models of expert reasoning about physics and numerical analysis.

Section 2 of this report describes the domains in which we are conducting our research. Section 3 describes in more detail the processes shown in Figure 1 which need to be automated. Section 4 describes the AI issues on which we are concentrating our research effort. Section 5 describes our current implementation and experimental results. Section 6 describes related work, and Section 7 concludes the report.

2 Domains

This section describes our two domains.

2.1 Design of Racing Yachts

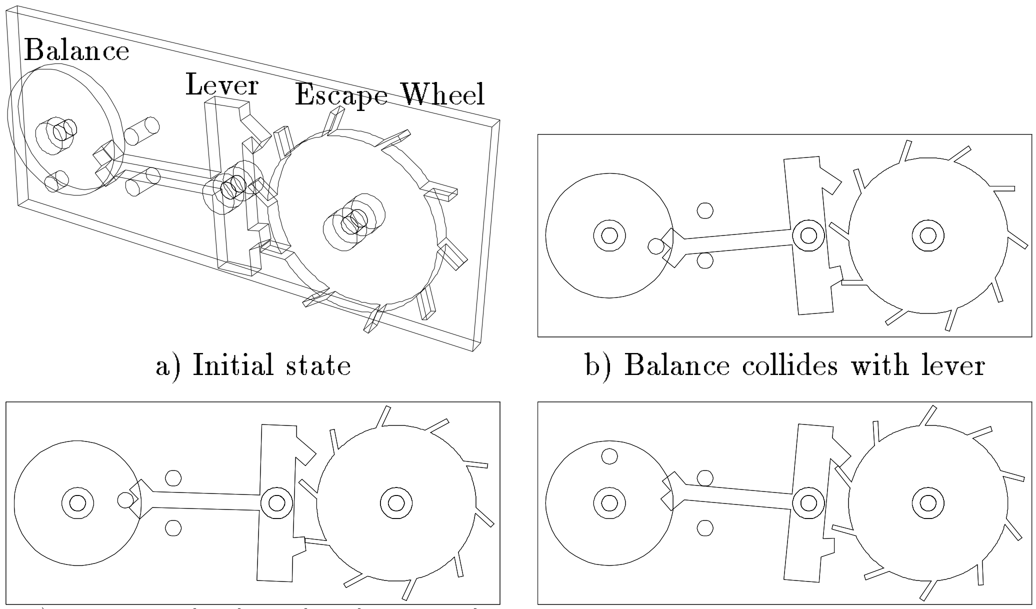
Much of the research discussed in this report has been done in the context of a larger system called the *Design Associate*, which partially automates the design of racing yachts like the one in Figure 2. In order to evaluate a possible yacht design, the Design Associate must determine how much time the yacht will need to traverse a specified race course, which will depend on the various forces acting on the yacht. Some of these forces can be computed quite accurately with simple formulas: for example, much of the drag on the yacht is due to an effect known as skin friction, which is directly proportional to the surface area of the yacht. Other forces, however, can only be computed with sufficient accuracy by using powerful numerical simulators. One force in this category is *lift-induced drag*, the main example discussed in this report.

The force the wind exerts on a yacht can be decomposed into two components, one oriented in the direction the yacht is moving and the other perpendicular to the direction of motion. The perpendicular force must be balanced by a force from the yacht's keel, which acts as a lifting surface just like an airplane's wing does, except that for the yacht the lift force is horizontal instead of vertical. The physical effect which generates this lift force also necessarily generates a corresponding drag force, lift-induced drag, which can significantly affect a yacht's performance.

In order to tractably compute this force, a number of modeling assumptions must be made. The key assumption is that viscosity may be neglected when computing lift-induced drag, because viscous contributions to drag will be modeled in other ways, for example as skin friction. If viscosity is neglected, the general nonlinear Navier-Stokes equations of fluid flow may be reduced to a linear partial differential equation, Laplace's equation for velocity potential

$$\nabla^2 \phi = 0$$

The velocity vector of the fluid at any point is given by the gradient of the scalar velocity potential $\phi(x, y, z)$. Laplace's equation may then be reformulated as an integral equation describing the velocity potential on the surface of the racing yacht.



a) Initial state

b) Balance collides with lever

c) Escape wheel pushes lever and balance

d) Halfway through a full cycle

Figure 3: Clock or watch escapement mechanism

2.2 Clockwork Mechanisms

The escapement mechanism in Figure 3 keeps the average speed of a clock or watch constant by allowing the escape wheel, which is pushed clockwise by a strong spring, to advance by only one tooth for each oscillation of the balance. I've discussed this mechanism in prior publications [Gelsey 1987, Gelsey 1989, Gelsey 1990, Gelsey 1991]. Modeling the behavior of this mechanism requires a series of assumptions. The first assumption is that Newton's laws of motion are adequate for this domain. Newton's laws relate motions to forces, and must be supplemented by models of the forces in a system in order to predict its behavior, so the remaining modeling assumptions are those supporting models for the various forces. In particular, I've explored two alternative ways of modeling collision forces: a microscopic model in which solid objects are allowed to overlap slightly in space, generating a repulsive force proportional to their depth of overlap (the depth of overlap approximates the slight distortion of the part that is the actual source of the contact force), and a macroscopic model in which moving parts are assumed to collide inelastically and move under geometric constraints from then on. Both models predict similar behaviors for the escapement mechanism.

3 Processes to be Automated

This section describes in more detail the processes shown in Figure 1 which need to be automated.

3.1 Setting up a Computational Simulation

A model of a physical situation suitable for computational simulation must include a set of state variables, quantities whose values represent the state of the physical system. The model must also include a set of differential equations describing the relations between the values of the various state variables. The particular equations for a specific situation must be instantiated from general template equations like Newton's laws or Laplace's equation. The instantiated equations will refer to a specific set of state variables, for example the positions and velocities of the three moving parts of the escapement mechanism in Figure 3, and therefore may be solved by a numerical method which will then compute the values of the state variables for the specific situation being modeled.

I've discussed the problem of identifying state variables and differential equations for clockwork mechanisms extensively in prior publications [Gelsey 1987, Gelsey 1989, Gelsey 1990], so I'll focus here on the fluid flow domain. The state of the fluid is represented by the value of the velocity potential $\phi(x, y, z)$ at every point. For a computer simulation, however, ϕ must be represented by a finite set of discrete values. This discretization is the principal problem of setting up a simulation of fluid flow. PMARC[1], the program we use to compute lift-induced drag, is a "panel method" - the discretization it uses to analyze flow around a solid body moving through a fluid is based on dividing the surface of the body into small pieces called panels. Using PMARC to analyze fluid flow requires

1. Partitioning the surface of the body into panels

2. Modeling the wake — the set of streamlines leaving the body

[1]Panel Method Ames Research Center

3. Partitioning the surface of the wake into panels

It is important to emphasize that the initial description of a physical situation to be analyzed typically does not identify state variables. In the case both of mechanical devices and fluid flow, the input consists primarily of a purely geometrical description supplemented by information about various other physical properties like masses of parts in a mechanical device or density of fluid in a flow problem. Currently we represent mechanical device geometry using CAD/CAM Constructive Solid Geometry techniques [Requicha 1980], while ship geometry is represented using B-spline surfaces [Rogers and Adams 1990]. These geometrical descriptions must then be partitioned into pieces corresponding to different state variables.

3.2 Quality Assurance

A major difficulty in using computational simulation, especially in an automated manner, is that it is generally difficult to tell whether or not to trust the output of the program. A human expert building a model of physics on which to base a simulation program makes use of a body of knowledge and assumptions which does not end up in the simulation program itself. For example, PMARC is based on the assumption that viscosity safely can be neglected. This assumption tends to hold much better for long, thin shapes than for short, fat ones. However, PMARC can be run to predict flow around a short, fat shape, and to an inexperienced user the results will look perfectly reasonable, until compared with experimental data, if any is available.

An automated solution to this "quality assurance" problem requires representing and using this body of knowledge and assumptions, which can be broadly classified into the following categories

- Modeling assumptions, e.g. PMARC assumes zero viscosity — these assumptions may or may not be directly testable. An example of a testable assumption from my microscopic model of collision forces in mechanisms is the assumption that the volumes of moving parts will overlap only slightly. If a large overlap occurred during a simulation that would indicate that the simulation output was not trustworthy.

- Expectations about input, e.g. long, thin body — more directly testable properties of the input typically derived from the simplifying assumptions, often by a fairly complex chain of reasoning.

- Expectations about output
  - ranges of plausible output values, e.g. velocities within fluid flowing around a body shouldn't be greater than several times the body's velocity with respect to the surrounding fluid.
  - consistency requirements for output values, e.g. the sum of the kinetic and potential energy in a clockwork mechanism should not increase as time passes
  - qualitative relationships among output values, e.g. fluid flow around a solid body should have a stagnation point where the flow hits the body, then should speed up as the fluid passes around the body again and then slow down again as the fluid merges together and leaves the body.
  - convergence behavior — as space and time discretizations are refined, differential equation solutions should settle down to asymptotic values
  - sensitivity to minor input perturbations — typically, small perturbations should generate small responses
  - response to extreme input values — may be computable from overall trends
  - trusted output results from similar physical configurations
- Test cases with known solutions
- Associated simpler models useful for coarse checking of output
- Associated deeper models to use if satisfactory output cannot be achieved
- Alternative numerical methods that might be appropriate

Some of these criteria could be better applied given a database of past experience. For example, values of lift and drag forces for similar sailing yachts could be used to reject implausible values resulting from a bad simulation.

3.3 Feedback

Successful analysis of a physical situation will typically require running a numerical simulator several times and intelligently modifying the simulator's input between each run. In the case of fluid flow, the heuristic nature of the guidelines for producing a good panelization make it likely that the first attempt will yield an unacceptable solution. Quality assurance knowledge must then be applied to recognize problems in the solution and attempt to diagnose and correct flaws in the initial panelization that led to these problems. In the mechanics domain, feedback between simulation runs is needed not so much to change state variables as to change initial simulation conditions in order to more fully understand the physical situation (see [Gelsey 1991]).

4 Artificial Intelligence Research Issues

This section describes the AI issues on which we are concentrating our research effort.

The three central subtasks for setting up a PMARC computation are

1. Partitioning the surface of the body into panels
2. Modeling the wake — the set of streamlines leaving the body
3. Partitioning the surface of the wake into panels

These tasks require a complex blend of spatial reasoning and reasoning about physics and numerical analysis. The goal of these tasks is to set up a PMARC computation by providing geometrical input that will allow PMARC to reliably predict fluid flow around a body. Spatial reasoning is essential to find geometrically plausible input for PMARC. For example, panels must lie on the surface of the body and cover the entire surface. However, many spatially plausible panelizations may produce wrong or unreliable fluid flow prediction. Choosing good configuration from the set of possible configurations requires significant reasoning about both physics and numerical analysis.

Both sorts of reasoning are also needed for quality assurance and feedback. For example, spatial reasoning might be used in quality assurance to recognize a body which is symmetric with respect to the flow direction could not generate unsymmetrical forces. If PMARC predicts unsymmetrical forces in such a case, perhaps because of a bad panelization, the output could be classified as unreliable. Feedback could then modify the input to PMARC in order to try to produce more reliable flow predictions.

4.1 Spatial Reasoning

Though spatial reasoning is of increasing interest to AI researchers [Chen 1990], this subfield is not yet sufficiently developed for it to be clear whether a set of generic spatial reasoning tasks can be defined, or what these tasks might be. As we attempt to automate the spatial reasoning needed to set up a PMARC computation, we will attempt to identify pieces of the spatial reasoning process that appear generic in the sense of being likely to occur in a wide variety of spatial reasoning contexts.

Reasoning about geometrical symmetry (or partial symmetry) seems to be one fairly generic form of spatial reasoning, in the sense that it appears in many varied contexts, for example in reasoning about the kinematic behavior of mechanical devices [Gelsey 1987]. Symmetry is quite an important consideration in choosing panelizations for PMARC. For example, one basic panelization method is to chose panel corner points by intersecting a sequence of parallel planes with the body around which fluid is flowing, choosing an equal number of panel corner points on each intersection line. Very different panelizations will be produced by this method depending on the direction in which the sequence of parallel planes lies. Finding approximate symmetries in the shape of the body can be very helpful in choosing an appropriate direction for these planes.

We have received considerable advice about how to create "good" panelizations of a surface from the domain experts with whom we are working, Drs. Fritts, Salvesen, and Letcher, all members of the design team for the *Stars & Stripes* yacht shown in Figure 2. The principal guidelines we have been given include

- Resolve stagnation points and other areas of rapidly changing potential. The fluid flow hitting a solid body must change its direction in order to pass around one side of the body or the other side. As a result, there are *stagnation points* on the surface of the body where fluid velocity is zero. Since the velocity potential changes rapidly near points like these, small panels must be used in those areas to properly represent the state of the fluid flow.

- Attempt to maintain a nearly orthogonal mesh of panels. Panels should resemble rectangles rather than long, thin diamonds.

- Attempt to keep panel aspect ratio low. Panels should not be far longer than they are wide.

- Control panel expansion ratio. Bound the ratio of the areas of adjacent panels so that very small panels won't have very large neighbors.

- Attempt to orient panel boundaries along streamlines of fluid flow.

Managing the tradeoffs between these guidelines can be a complex reasoning problem. For example, if velocity potential near stagnation points changes rapidly in one direction but remains fairly constant in the other, resolution of the stagnation point may best be achieved with long, thin panels which tend to violate aspect ratio constraints. In some cases constraints need to be applied more and less stringently in different areas of the body's surface. For example, a rounded body can't have an extremely orthogonal mesh everywhere, but orthogonality should be maintained as widely as possible.

Modeling the wake also involves complex spatial reasoning. The advice we've been given includes

- Base the wake model on streamlines generated by a computation of fluid flow when the body is oriented so as to generate no lift or lift-induced drag — the "zero angle of attack" situation for airplanes or "zero yaw" situation for yachts.

- Attach wakes to all sharp trailing edges of the body.

- The upper edge of the wake from a yacht's keel should follow its hull to the waterline, and then follow the water's surface — "wake contraction".

- Wakes should not pass through solid bodies — for example, the wake from the keel must miss the rudder.

Figure 4 shows a panelization of hull, keel, and wake for *Stars & Stripes*. The panelization algorithm used is described in Section 5.

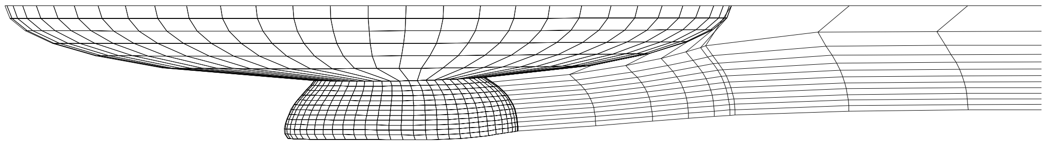


Figure 4: Panelization of hull, keel, and wake for *Stars & Stripes*

4.2 Deep Models of Expert Reasoning about Physics and Numerical Analysis

While spatial reasoning can produce "plausible" panelizations, it is unlikely to produce "good" ones, because spatial reasoning is not adequate for making certain basic decisions like determining an appropriate ratio between panel density parallel to the flow direction and panel density perpendicular to the flow direction. Such a decision requires reasoning about error models for numerical approximations. For example, PMARC treats velocity potential as being constant over each panel. Clearly, such an approximation cannot represent the potential with an error less than the largest jump in potential between neighboring panels. If the result of a PMARC flow computation has jumps in potential that are considerably larger in one direction than the other, that would tend to indicate that the ratio between panel density parallel to the flow direction and panel density perpendicular to the flow direction should be modified. Feedback based on this analysis would be likely to result in a more reliable flow computation.

These models of expert reasoning about physics and numerical analysis are also necessary to judge tradeoffs between the various panelization guidelines listed earlier. For example, our experiments with PMARC indicate that superficial rules like "limit panel aspect ratio to 10" are not adequate, even if the particular numerical parameters are indexed by a set of cases each having a different value. A better approach is to reason about error models for numerical approximations, as discussed above. A panel with high aspect ratio may be quite acceptable if the velocity potential is not much different between its neighbors in its long direction, while a panel with much lower aspect ratio may be unacceptable due to the behavior of the potential around it.

5 Experimental Results

Our initial implementation effort has focused on extending the standard version of the PMARC simulator so that it can automatically set up PMARC computations for a limited range of physical situations. This first step is important for the following reasons:

1. By doing this process ourselves, we are learning more about the kinds of physical, spatial and numerical reasoning that we will need to automate for the system to handle a more general range of physical situations.

2. This implementation provides an interface allowing automated control of PMARC by another program, which will be essential for exploring our ideas about spatial reasoning and modeling physical systems

3. This implementation allows the Design Associate to automatically invoke PMARC to compute a yacht's effective draft, which is essential for testing ideas being developed in that work for choosing between PMARC and cheaper but less accurate models.

This extension involved adding the following capabilities to PMARC:

1. automatic panelization

2. automatic generation of a wake which follows streamlines computed by PMARC

3. computation of effective draft by integrating momentum and kinetic energy in the wake at a point far downstream from the yacht

We have successfully implemented these capabilities, as described below.

Our panelizer accesses the geometry of a ship by calling a "black box" function whose inputs are two parametric coordinates u and v and an index indicating which surface is being panelized, and whose output is the location of a point (x, y, z). Internally, a black box will typically use b-spline surfaces to represent parts of a yacht like the hull and the keel, but the panelizer depends only on the well-defined black box interface and makes no assumptions about the black box internals. At present, the panelizer assumes that the ship consists of two pieces: a hull and a keel. The surfaces of these two pieces (as described by the black box function) are allowed to cross each other, and

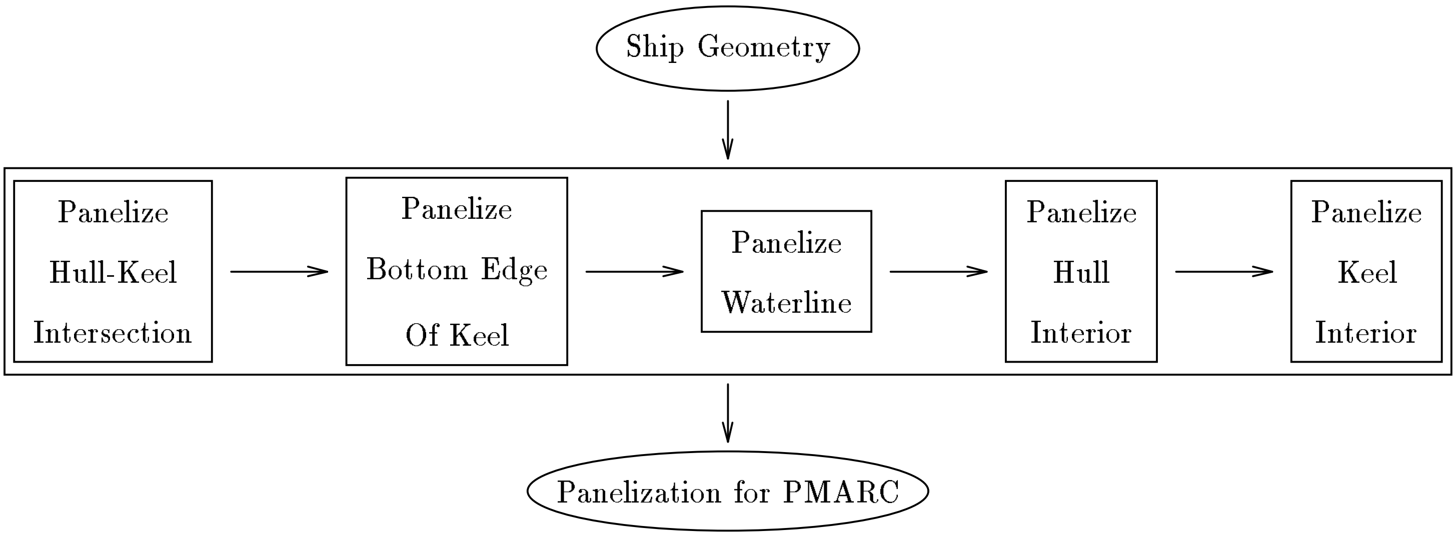


Figure 5: Panelization process

the panelizer throws away any fictitious parts of the surfaces that lie inside the other body. The panelization algorithm is the following (see Figure 5):

- Find panel corner points on the boundary between the hull and the keel by finding the intersection of the hull, the keel, and a sequence of planes perpendicular to the flow direction. Each intersection is computed by solving a four-dimensional nonlinear equation.

- Find panel corner points on the bottom edge of the keel by finding the intersection of the keel and a sequence of planes perpendicular to the flow direction. Each intersection is computed by solving a one-dimensional nonlinear equation.

- Find panel corner points on the waterline of the hull by finding the intersection of the hull, a plane representing the water's surface, and a sequence of planes perpendicular to the flow direction. Each intersection is computed by solving a two-dimensional nonlinear equation.

- Find all other panel corner points on the hull and the keel by

  - choosing a depth for the point by interpolating between the corresponding depths on the upper and lower edges of the surface,

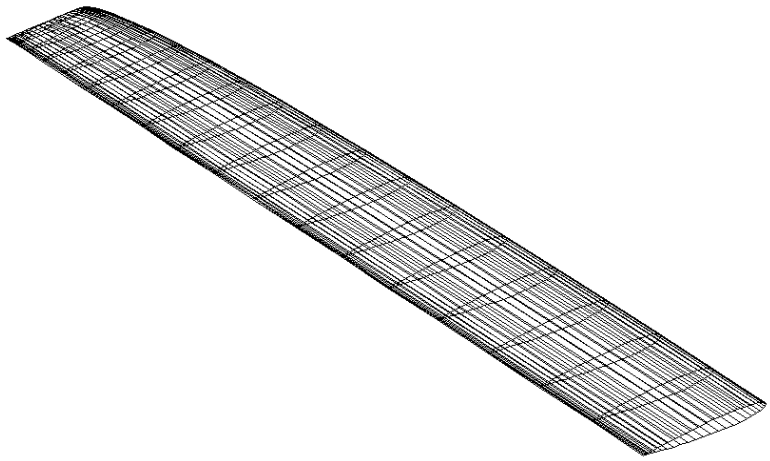


Figure 6: Wing with NACA 0012 cross-section

- choosing a position for the point with respect to the flow direction by interpolating between the leading an trailing edges of the surface, and finally
- computing the third coordinate of the panel corner point by solving a two-dimensional nonlinear equation.

We have also implemented automatic generation of a wake which follows streamlines computed by PMARC and computation of effective draft by wake integration. Effective draft is a function of lift and drag, which PMARC normally computes by summing the pressures on the surface panels. However, our domain experts found this technique only gave reliable values for lift, not for drag, so they instead computed effective draft by integrating momentum and kinetic energy in the wake at a point far downstream from the yacht [Letcher *et al.* 1987]. Our experiments have also confirmed that this method is preferable.

We have successfully implemented the capabilities described above, and Figure 7 and Figure 8 show some data produced by the resulting program. The program was set up to automatically run a series of computations for the same physical situation using different panelizations. Figure 7 shows

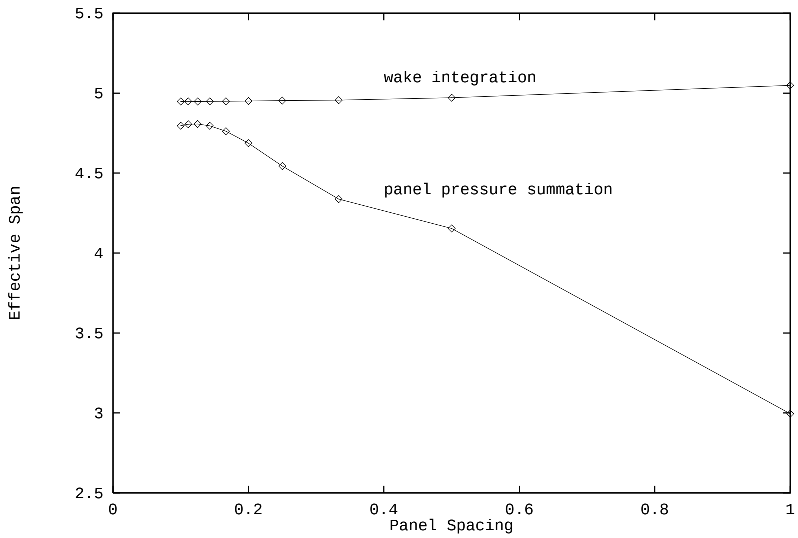


Figure 7: PMARC output for various panelizations of the wing

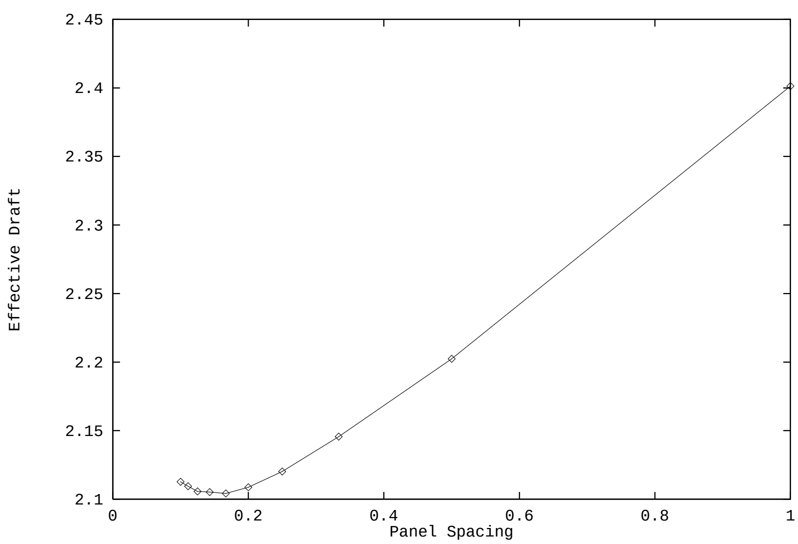


Figure 8: PMARC output for various panelizations of *Stars & Stripes*

various computations of effective span for the wing shown in Figure 6, and Figure 8 shows computations of effective draft for *Stars & Stripes*. Figure 4 shows one of the panelizations in the series. For both physical situations, ten different panelizations were used, with grids ranging from 3 by 10 to 30 by 100 panels. The horizontal axis of the plots shows normalized panel spacing — the smallest spacing corresponds to the largest number of panels.

Figure 7 includes computations of effective wing span using two different methods: panel pressure summation and wake integration. The actual span is 5, and in both cases the effective span computed using a sufficiently large number of panels is slightly lower, as would be expected due to efficiency losses at the wing tip. Our domain experts found wake integration to be the more reliable method [Letcher *et al.* 1987]. Figure 8 only shows results of computations using wake integration: in this case panel pressure summation gave physically meaningless results. Again the effective draft computed using a sufficiently large number of panels seems consistent with the dimensions of the physical situation: *Stars & Stripes* has a maximum keel draft of 2.8 meters and a maximum hull draft of 1.6 meters. For a simple keel without winglets, theory predicts that effective draft will be smaller than physical draft.

For our analysis of *Stars & Stripes* we replaced the actual hull with an ellipsoid having the same draft, waterline length, and waterline beam. This change simplified the setup process, and our domain experts indicated that effective draft is much more sensitive to keel shape than to hull shape. The spatial reasoning research described earlier in this report should eventually allow up to automatically set up computations for a wide variety of shapes including the full *Stars & Stripes*.

In the mechanical device domain I've implemented programs to automatically set up and intelligently control computational simulations of clockwork mechanisms. [Gelsey 1987, Gelsey 1989, Gelsey 1990, Gelsey 1991]. I am currently considering the problem of quality assurance for this domain.

6 Related Work

Jambunathan *et al.*[1991] and Andrews[1988] discuss the use of expert systems technology to augment more traditional computational fluid dynamics programs. Most other artificial intelligence research concerning reasoning about physical systems has focused on qualitative rather than numerical simulation. [Weld and de Kleer 1990] Exceptions are the work of Sacks[1991] and Yip[1991]; however, they have focused on numerical simulators for ordinary differential equations and have not addressed the issue of quality assurance. Forbus and Falkenhainer[1990] discuss the use of qualitative simulation to check the quality of numerical simulation results; however, the approach described appears limited to physical situations modeled by ordinary differential equations.

7 Conclusion

While the problems of setup, analysis, quality assurance, and feedback for computational simulation are clearly of considerable practical importance, their solution should also lead to significant insights into some basic AI problems. Successful computational simulation of complex physical systems appears to require a combination of spatial reasoning and reasoning about physics which has so far received little attention from AI researchers. The first step in classifying and understanding these reasoning methods is to collect a set of concrete examples of them, and the only way to be sure the examples fully capture the reasoning used is to incorporate them in programs which can successfully apply the reasoning to the control of computational simulations. This is the goal of the research discussed in this report.

So far, we have successfully implemented extensions to the standard version of the PMARC simulator so that it can automatically set up PMARC computations for a limited range of physical situations. This first step is important both to clarify the reasoning and representations needed to handle a wider range of problems, and to make automated control of PMARC possible, which is an essential prerequisite both for our research on spatial reasoning and modeling physical systems and in order to allow the Design Associate to automatically invoke PMARC to compute a yacht's effective draft

During the coming year, we plan to pursue our ideas regarding spatial reasoning and deep models of expert reasoning about physics and numerical analysis. In the spatial reasoning area, we are attempting to formulate the PMARC panelization problem as a search for appropriate transformations within a hierarchy of spatial representations for a yacht (or other physical object). In the modeling area, we are attempting to formalize the space of approximate physical models on which PMARC is based, in order to make all simplifying assumptions explicit so that PMARC output can be tested for consistency with these underlying assumptions.

8 Acknowledgments

The research on automated use of PMARC was done with fellow Rutgers Computer Science Dept. faculty member Gerard Richter and graduate student Ke-Thia Yao. We worked with hydrodynamicists Martin Fritts and Nils Salvesen of Science Applications International Corp., and John Letcher of Aero-Hydro Inc. Our research is part of the CAP (AI and Design) project, and benefited significantly from interaction with other members of the project. The CAP project is supported by the Defense Advanced Research Projects Agency and the National Aeronautics and Space Administration under NASA grant NAG2-645.

References

[Andrews 1988] Andrews, Alison E. 1988. Progress and challenges in the application of artificial intelligence to computational fluid dynamics. *AIAA Journal* 26(1):40–46.

[Chen 1990] Chen, Su-shing, editor 1990. *Advances in Spatial Reasoning*, volume 1. Ablex, Norwood, New Jersey.

[Forbus and Falkenhainer 1990] Forbus, Kenneth D. and Falkenhainer, Brian 1990. Self-explanatory simulations: An integration of qualitative and quantitative knowledge. In *Proceedings, Eighth National Conference on Artificial Intelligence*, Boston, MA. AAAI-90.

[Gelsey 1987] Gelsey, Andrew 1987. Automated reasoning about machine geometry and kinematics. In *Proceedings of the Third IEEE Conference on Artificial Intelligence Applications*, Orlando, Florida. Also appears in Daniel S. Weld and Johan de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems*, Morgan Kaufmann, 1990.

[Gelsey 1989] Gelsey, Andrew 1989. Automated physical modeling. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Detroit, Michigan USA.

[Gelsey 1990] Gelsey, Andrew 1990. *Automated Reasoning about Machines.* Ph.D. Dissertation, Yale University. YALEU/CSD/RR#785.

[Gelsey 1991] Gelsey, Andrew 1991. Using intelligently controlled simulation to predict a machine's long-term behavior. In *Proceedings, Ninth National Conference on Artificial Intelligence*, Cambridge, MA. AAAI Press/The MIT Press.

[Jambunathan *et al.* 1991] Jambunathan, K.; Lai, E.; Hartle, S. L.; and Button, B. L. 1991. Development of an intelligent front-end for a computational fluid dynamics package. *Artificial Intelligence in Engineering* 6(1):27–35.

[Letcher *et al.* 1987] Letcher, John S.; Cressy, Christopher P.; Olivier III, James C.; and Fritts, Martin J. 1987. Hydro-numeric design of winglet keels for *Stars & Stripes*. *Marine Technology* 24(4):265–285.

[Requicha 1980] Requicha, Aristides A. G. 1980. Representations for rigid solids: Theory, methods, and systems. *ACM Computing Surveys* 12:437–464.

[Rogers and Adams 1990] Rogers, David F. and Adams, J. Alan 1990. *Mathematical elements for computer graphics.* McGraw-Hill, 2nd edition.

[Sacks 1991] Sacks, Elisha P. 1991. Automatic analysis of one-parameter ordinary differential equations by intelligent numeric simulation. *Artificial Intelligence* 48(1).

[Weld and de Kleer 1990] Weld, Daniel S. and de Kleer, Johan, editors 1990. *Readings in Qualitative Reasoning about Physical Systems.* Morgan Kaufmann, San Mateo, California.

[Yip 1991] Yip, Kenneth 1991. Understanding complex dynamics by visual and symbolic reasoning. *Artificial Intelligence* 51(1–3).