Topics in Development of Naval Architecture Software Applications

Kevin McTaggart, David Heath, James Nickerson, Shawn Oakey, and James Van Spengen
Simulation of Naval Platform Group
Defence R&D Canada – Atlantic
Dartmouth, Nova Scotia, Canada

Society of Naval Architects and Marine Engineers
Atlantic Canada Section, Halifax, 16 April 2014
## Authors

<table>
<thead>
<tr>
<th>Kevin McTaggart</th>
<th>David Heath</th>
<th>James Nickerson</th>
<th>Shawn Oakey</th>
<th>Jim Van Spengen</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.jpg" alt="Kevin McTaggart" /></td>
<td><img src="image2.jpg" alt="David Heath" /></td>
<td><img src="image3.jpg" alt="James Nickerson" /></td>
<td><img src="image4.jpg" alt="Shawn Oakey" /></td>
<td><img src="image5.jpg" alt="Jim Van Spengen" /></td>
</tr>
</tbody>
</table>
Naval Architecture Software Applications - Outline

- Evolution of naval architecture software
- What’s happening today: great complexity
- How we work as a team to develop software
- Where are we going?
Early Days
Evolution of Naval Architecture Software Applications: Early software applications (1960s and 1970s)

- Hydrodynamic added mass and damping for arbitrary two-dimensional hull sections (Frank 1967)
- Ship hydrostatic analysis (Naval Ship Engineering Center 1976)
- Basic ship structural optimization (Hughes and Mistree 1976)
- Definition of hull lines using splines (Fuller, Aughey, Billingsley 1977)
- Estimation of ship performance properties using regression of experimental results:
  - Powering (Holtrop and Mennen 1978)
Evolution of Naval Architecture Software Applications: Contemporary applications

- Prediction of ship resistance with computational fluid dynamics, including detailed modelling of viscous effects (Thornhill 2008)
- Simulation of ship maneuvering in waves (McTaggart 2010)
- Simulation of large-magnitude ship structural deformation, including collisions (Haris and Amdahl 2013)
## Evolution of Naval Architecture Software Applications: Increasing Complexity

<table>
<thead>
<tr>
<th></th>
<th>1988</th>
<th>2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>Languages</td>
<td>Fortran</td>
<td>C++, C#, Java, Python, ...</td>
</tr>
<tr>
<td>Lines of code</td>
<td>5,000</td>
<td>5,000,000</td>
</tr>
<tr>
<td>Size of development team</td>
<td>1-2</td>
<td>5-10</td>
</tr>
<tr>
<td>User input method</td>
<td>Text file or console line</td>
<td>Graphical user interface</td>
</tr>
<tr>
<td>Seakeeping prediction</td>
<td>Strip theory, frequency domain</td>
<td>Strip theory or 3D, frequency domain or time domain</td>
</tr>
<tr>
<td>Resistance prediction</td>
<td>Potential flow</td>
<td>Viscous flow</td>
</tr>
<tr>
<td>Structural analysis</td>
<td>Hull girder</td>
<td>3D finite element</td>
</tr>
</tbody>
</table>
Example Contemporary Applications: Frequency Domain Seakeeping with Strip Theory

- Salvesen, Tuck, and Faltinsen (1970) give excellent overview
- Assumes ship geometry is slender:
  - Applicable for most naval vessels
- Surprisingly good results when compared with more sophisticated approaches
Ship Motions Prediction with 3D Models

- 3D theory gives advantages over strip theory:
  - Applicable to wider range of hull forms
  - More accurate sea load predictions
  - Modelling of interaction effects between vessels
- Small computational times on modern desktop computers
Simulation of Ship Motions in the Time Domain

- Time domain simulation presents new opportunities
  - Maneuvering
  - Nonlinear forces
  - Interoperability with other simulations
- Can often run in real-time or faster
Example Contemporary Applications: Simulation of Replenishment at Sea

- Seaway
- Supply ship, including helm and motions in seaway
- Receiving ship, including helm and motions in seaway
- Replenishment gear
Example Contemporary Applications:
Simulation of Launch and Recovery of Small Boats from Ships

- Navies have great interest in launch and recovery
  - Anti-piracy
  - Search and rescue
  - Autonomous vehicles
- Launch and recovery becomes increasingly challenging as sea state increases
Ship Operator Guidance

- Can provide real-time guidance
- Seaway measurement
- Rapid computation of ship motions
- Very enthusiastic response from ship operators
Where Are We Today?

- Software is very complex
- Huge range of skills required
- No single person can complete work
How Do We Get the Job Done?

- We use modern software technologies
- We work as a team
Relevant Technologies for Modern Software Development

- Spoilt for choice
- Must consider both initial development and long-term maintainability of software
Wide Availability of Programming Libraries

- Usage of existing software libraries can bring many advantages:
  - Reduction of development effort
  - Existing documentation
  - Existing expertise using software library

- Be aware of licensing terms:
  - No cost or restrictions
  - No cost but restrictions on developed software
  - Cost for developer license ($)
  - Cost for developer license and for runtime license ($$)
Object-Oriented Programming

- Dominant approach for developing modern software
- C++, C#, Java, etc.
- Example objects from ship motion library:
  - Seaway
    - RegularSeaway
    - MultiComponentSeaway
  - ShipHull
    - DryShipHull
    - WetShipHull
  - ShipAppendage
    - Rudder
    - Bilge keel
Graphical User Interfaces

- GUIs can greatly improve usability of software
- Modern software libraries make development easy
3D Modelling and Visualization

- Ship hulls and other 3D surfaces can be modelling highly accurately using parametric surfaces

\[ x, y, z = F(u, v) \]
3D Modelling and Visualization

- 3D visualization has become a vital tool for understanding results of complex simulations.
Distributed Simulation

- A complex simulation can run on an array of computers linked via a network
- Array of software programs must share data and run in a synchronized manner
- High Level Architecture (Kuhl, Weatherly and Dahmann 1999) is one approach that can be used
  - Specialized skill required
  - Large investment, but potential long-term payoff
Parallel Computing

- Computers with multiple processors are now the norm
  - Central processing units (CPUs)
  - Graphic processing units (GPUs)
- Development tools for parallel computing are widely available ...
- But, parallel programming can be difficult
- Examples well-suited to parallel processing:
  - Evaluation of Green functions for flow on a hull surface
  - Matrix multiplication
Programming Languages: Factors Influencing Selection

- Numerical computation, including complex numbers
- Ease of programming and maintenance
- Execution speed
- Platform portability
- Availability of libraries, including visualization and user interfaces
- Interoperability with other programming languages
- Availability of programmers and time required to train programmers
- Suitability to problem
# Programming Languages: Current Popularity

<table>
<thead>
<tr>
<th>Language</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>2</td>
</tr>
<tr>
<td>C++</td>
<td>4</td>
</tr>
<tr>
<td>C#</td>
<td>5</td>
</tr>
<tr>
<td>Python</td>
<td>8</td>
</tr>
<tr>
<td>Fortran</td>
<td>34</td>
</tr>
</tbody>
</table>

Source: TIOBE Index ([www.tiobe.com](http://www.tiobe.com)), April 2014
Programming Languages: Current Usage by Authors

- **C++**
  - Used by authors since early 1990s
  - Object-oriented, high execution speed
  - Requires very skilled programmer

- **C#**
  - Modern language developed using lessons from C++ and Java
  - Relatively easy to program
  - Runs in virtual machine, giving some performance penalty

- **Python**
  - Dynamically typed, so variables aren’t declared (\(\text{double } x\) not required)
  - Relatively easy to program and code is concise
  - Slower execution speed because code is interpreted during execution
Matching of Skills To Tasks: Two Types of Contributors

- **Domain expert**
  - Post-graduate degree in engineering, math, or physics
  - Specialized domain knowledge in hydrodynamics, structural mechanics, or multi-body dynamics
  - Writing papers and/or reports is part of job responsibilities
  - Competent with 1 or 2 higher level languages (e.g., C# and Python)

- **Computer scientist**
  - Degree(s) in computer science, with proficiency in math and physics
  - Knowledge of relevant naval architecture for writing software
  - Prefers to write software rather than papers or reports
  - Very competent with several computer languages
  - Skills in other areas, such as geometric modelling, visualization, graphical user interfaces, distributed simulation, and computer administration
Matching of Skills To Tasks: Assignment of Work

- Numerical modelling of physical phenomena for hydrodynamics and structural mechanics
  - Domain expert uses higher level language (e.g., C#, MATLAB, Python)
- User interfaces and visualization
  - Computer scientist applies expertise to range of software applications
  - Domain expert can contribute if high-level framework is available
- Interoperability using distributed simulation
  - Computer scientist applies expertise in range of computer languages, including C++
- Code optimization
  - Computer scientist applies expertise in range of computer languages, including re-writing portions of code in faster language
Software Documentation

- Types of documentation
  - Comments within source code
  - Theory reports describing what is being modelled by software
  - User manuals
- Provides many benefits:
  - Improved software maintainability
  - Confidence in software
  - Wider range of software users
Verification and Validation

- **Verification:**
  - Testing to ensure that software correctly solves equations as intended
  - Can include comparison of results with known solutions:
    - Analytical solutions
    - Other software that is known to be correct

- **Validation**
  - Testing to ensure that software gives results that compare favourably with “real world”
  - Can include comparison of results with:
    - Model experiments
    - Full-scale trials
Verification Example:
Volume and Added Mass for a Hemisphere at Free Surface

<table>
<thead>
<tr>
<th>Number of panels</th>
<th>Volume V</th>
<th>Computed/Exact Surge added mass $A_{11}$</th>
<th>Sway added mass $A_{22}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>308</td>
<td>0.974</td>
<td>0.978</td>
<td>0.982</td>
</tr>
<tr>
<td>704</td>
<td>0.988</td>
<td>0.990</td>
<td>0.990</td>
</tr>
<tr>
<td>1972</td>
<td>0.996</td>
<td>0.996</td>
<td>0.996</td>
</tr>
</tbody>
</table>

Agreement and convergence are required
Validation Example:
Motions of a Naval Destroyer from Seakeeping Trials

- Full-scale sea trials are routinely conducted to obtain seakeeping validation data
- Accurate measurements of ship motions and directional wave spectra are essential
Predicted Versus Measured RMS Pitch
Where Are We Going?

- Validation of complex simulations with model tests and full-scale trials
  - Replenishment at sea
  - Launch and recovery of small boats
- More timely and higher quality transfer of CAD data to analysis applications
  - Hydrodynamic models
  - Finite element models
- Routine application of computational fluid dynamics (CFD)
  - Prediction of hull maneuvering forces