AN ABSTRACT OF THE DISSERTATION OF

Seshu B. Nimmala for the degree of Doctor of Philosophy in Civil Engineering presented on Sept 7, 2010.

Title: An Efficient High-Performance Computing Based Three-Dimensional Numerical Wave Basin Model for the Design of Fluid-Structure Interaction Experiments

Abstract approved:

Solomon C. Yim

Abstract:

Fluid-structure interaction (FSI) is an interesting and challenging interdisciplinary area comprised of fields such as engineering- fluids/structures/solids, computational science, and mathematics. FSI has several practical engineering applications such as the design of coastal infrastructure (such as bridges, levees) subjected to harsh environments from natural forces such as tsunamis, storm surges, etc. Development of accurate input conditions to more detailed and complex models involving flexible structures in a fluid domain is an important requirement for the solution of such problems. FSI researchers often employ methods that use results from physical wave basin experiments to assess the wave forces on structures. These experiments, while closer to the physical phenomena, often tend to be time-consuming and expensive. Experiments are also not easily accessible for conducting parametric studies. Alternatively, numerical models

when developed with similar capabilities will complement the experiments very well because of the lower costs and the ability to study phenomena that are not feasible in the laboratory.

This dissertation is aimed at contributing to the solution of a significant component of the FSI problem with respect to engineering applications, covering accurate input to detailed models and a numerical wave basin to complement large-scale laboratory experiments. To this end, this work contains a description of a three-dimensional numerical wave tank (3D-NWT), its enhancements including the piston wavemaker for generation of waves such as solitary, periodic, and focused waves, and validation using large-scale experiments in the 3D wave basin at Oregon State University.

Performing simulations involving fluid dynamics is computational-intensive and the complexity is magnified by the presence of the flexible structure(s) in the fluid domain. The models are also required to take care of large-scale domains such as a wave basin in order to be applicable to practical problems. Therefore, undertaking these efforts requires access to high-performance computing (HPC) platforms and development of parallel codes. With these objectives in mind, parallelization of the 3D-NWT is carried out and discussed in this dissertation.

©Copyright by Seshu B. Nimmala

September 7, 2010

All Rights Reserved

An Efficient High-Performance Computing Based Three-Dimensional Numerical Wave Basin Model for the Design of Fluid-Structure Interaction Experiments

by

Seshu B. Nimmala

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented September 7, 2010
Commencement June 2011

Doctor of Philosophy dissertation of Seshu B. Nimmala
presented on September 7, 2010.

APPROVED:

Major Professor representing Civil Engineering

Head of the School of Civil and Construction Engineering

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Seshu B. Nimmala, Author

ACKNOWLEDGEMENTS

This dissertation is the culmination of the direct and indirect support and help of many individuals. I express my foremost gratitude to my Advisor Prof. Solomon Yim for his inspiring support, guidance and encouragement throughout the years of my research work and studies at Oregon State University. My thanks are due to Prof. Thomas Miller for encouraging me to start on the doctoral program and providing the necessary information and contacts in the early days in graduate school and also for showing interest in my progress. I am indebted to Prof. Malgorzata Peszynska for introducing me to the wonderful numerical methods and high-performance computing methods through the challenging course work in Mathematics and also for agreeing to be a member on my graduate committee. This course work has proven to be invaluable in my research and undoubtedly will continue to be a major contributor in my future professional pathways. I would not be able to accomplish all this without the education I received in the past from many teachers and also the mentoring and training from the senior staff in the industry where I worked in the past. I wish to specifically mention the guidance and encouragement offered by my Masters dissertation Advisor Prof. Anantha Ramu at Indian Institute of Science, Bangalore, India in pursuing advanced studies. It was from him that I heard of the boundary element method for the first time and this seminal discussion had planted the seed of curiosity in me. My days of working under the guidance of Dr. Peter Woytowitz at Engineering Mechanics Technology, Inc. have helped strengthen my experience and interests in the finite element method and related technologies. Lastly but not least, I am very thankful for the blessings I received from my parents, relatives and good wishes of friends. I would also like to mention the hardship endured and patience demonstrated by my family for missing me on several occasions due to the research work. I found that it is an extremely difficult job to be able to balance both sides of life, but this experience has hopefully helped me become stronger.

This work would never have been possible without the ready support of the computing system administration (led by Mr. Paul Montagne), help desks of the high-performance computing platforms and other software tools offered by the College of Engineering. The software tools, especially in the open source, the scientific visualization tool “xd3d”, several utilities on the Linux platform and the commercial plotting tool “dplot” have helped me a lot in preparing the animations and plots for this work. The friendly support offered by the administrative staff of the School of Civil and Construction Engineering and the Valley Library is hereby appreciated.

Discussions held with Prof. Stephan Grilli of the University of Rhode Island and the numerical code shared by him are very helpful as a good basis. The LS-DYNA program offered by Livermore Software Technology Corporation (LSTC) as part of the agreement with Prof. Yim and the discussions with Dr. Grant Cook, senior staff and expert on computational fluid dynamics at LSTC, have been instrumental in helping with my research. In fact, it was the ideas of Dr. Cook that initiated me in the direction of the boundary element method with fast multipole algorithm. I am thankful to my fellow students who chipped in with some of the interesting ideas and questions that refreshed me, and at times enlightened me, when I was struggling with problems that seemed insurmountable. I hereby acknowledge many other contributions, that are of no less importance and may not get mentioned here due to spatial limitations, and those that significantly impacted my work and helped me reach the milestones effectively and efficiently. It has been my dream for long to understand the computational techniques in depth to be able to apply them to engineering problems in an effective fashion. This work and the doctoral program gave me a very good opportunity and provided a solid foundation to take-off in such a direction. I am thankful to the Office of Naval Research (ONR) and Oregon Department of Transportation (ODOT) for generously supporting this work.

CONTRIBUTION OF AUTHORS

The author expresses gratitude to Prof. Stephan Grilli of the Department of Ocean Engineering, University of Rhode Island, USA for generously allowing the use of the numerical wave basin code developed by him and under his supervision, which has assumed an important role in this research work. His advising and reviewing the progress of this work from time to time despite his busy schedule has helped shape up the direction of this dissertation. Thanks are also due to him for agreeing to be a co-author to some of the publications.

TABLE OF CONTENTS

Page

Introduction …………………………………… 1

- Introduction …………………………………… 1
- Literature survey …………………………………… 2
- Scientific contributions of the dissertation …………………………………… 9

An Efficient Parallelized 3-Dimensional Numerical Wave Tank to Model Large-Scale Wave Basin Experiments …………………………………… 12

- Abstract …………………………………… 12
- Introduction …………………………………… 13
- FNPF theory and modeling of free surface waves …………………………………… 16
- Mathematical model …………………………………… 25
- Numerical methods …………………………………… 27
- Salient features of the algorithms …………………………………… 29
- Piston motion input …………………………………… 30
- Wave generation …………………………………… 32
- Discussion of results …………………………………… 35
- Parallelization and performance studies …………………………………… 37
- Concluding remarks …………………………………… 46
- Figures and tables …………………………………… 48
- Acknowledgement …………………………………… 64
- References …………………………………… 64

TABLE OF CONTENTS (Continued)

Page

Comparison and Validation of Nonlinear Potential Flow (BEM) and Navier-Stokes (FEM) Free Surface Flow Models Using a 3D Wave Basin Experiment ………………………………………………………………….. 70

Abstract ……………………………………………………………….… 70

Introduction ……………………………………………………………... 72

Background ……………………………………………………………... 72

Description of the benchmark experiment …………………………… 76

Input motion to the wavemaker ……………………………………… 78

Simulation using the FNPF-BEM ……………………………………. 78

Simulation using the NS-FEM ……………………………………….. 79

Comparison of results of the 3D-NWT with the benchmark ………… 82

Comparison of results of the NS-FEM, the FNPF-BEM and the benchmark……………………………………………………………… 84

Results of parallel performance ……………………………………… 85

Insight and experience gained ……………………............................ 86

Concluding remarks …………………………………………………….. 89

Figures and tables …………………………………………………….. 92

Acknowledgement …………………………………………………… 107

References ……………………………………………………………. 107

Generation of Focused Waves Using an Efficient 3D Numerical Wave Tank and Validation with Large-Scale Experiments ……………………. 111

Abstract ………………………………………………………………. 111

Introduction ……………………………………………………………... 112

Mathematical formulations …………………………………………... 117

TABLE OF CONTENTS (Continued)

Page

High-performance computing (HPC)……………………………………. 119

Focused waves ……………………………………………………….. 123

Discussion of results ………………………………………………… 125

Concluding remarks ………………………………………………….. 128

Figures and tables …………………………………………………….. 130

Acknowledgement ………………………………………………….... 145

References ……………………………………………………………. 145

Performance Studies of a 3D Numerical Wave Tank Model for Large-Scale Wave Basin Experiments ……………………………………….. 149

Abstract ………………………………………………………………. 149

Introduction …………………………………………………………... 150

Background …………………………………………………………... 150

Parallelization options ………………………………………………... 154

Choice of an optimal strategy ………………………………………... 159

Implementation using OpenMP …………………….......................... 161

Examples using 3D-NWT ……………………………………………. 163

Performance studies ………………………………………………….. 163

Insight and experience gained ……………………………………….... 167

Concluding remarks ………………………………………………….. 170

Figures and tables …………………………………………………….. 173

Acknowledgement …………………………………………………… 186

References ……………………………………………………………. 186

TABLE OF CONTENTS (Continued)

Page

General Conclusions ………………………………………………………... 188

Bibliography ……………………………………………………................ 193

Appendices . …………………………………………………………… 201

Appendix-A: Algorithms of the 3D-NWT …………………………… 202

Appendix-B: Build Instructions and Visualization Capabilities……… 204

Appendix-C: Details of the I/O Files ………………………………… 206

Appendix-D: Details of the FE Code Input ………………………….. 208

Appendix-E: Application of FSI to an ODOT Bridge ……………….. 210

LIST OF FIGURES

Figure Page

2.1 Sketch of 3D-NWT geometry and parameters, for wave generation by a snake piston wavemaker …………………………………….. 48

2.2 Snapshot of 3D numerical wave tank propagating a solitary wave .. 49

2.3 Surface elevation (in m on vertical axis) versus time (in sec on horizontal axis), at the wavemaker, for a solitary wave of height *H* = 0.3 m, in depth *h* = 0.75 m ……………………………………. 50

2.4 Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of solitary wave elevations at three gages, at x=8.8 m, 14.9 m, and 18.7 m, at y=0 for wave height=0.3 m ……….. 50

2.5 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of solitary wave elevations at wave gages along the length of the NWT, at y=0 for wave height=0.3 m ……………. 51

2.6 Solitary wave pressure on the bottom (in N/m^2 on vertical axis) as a function of time (in sec on horizontal axis), along the length of the tank at *y* = 0 for wave height = 0.3 m ………………………… 51

2.7 Solitary wave particle velocity components (*u*, *w*) (in m/sec on vertical axis) as a function of time (in sec on horizontal axis) at *x* = 18.7 m, *z* = -0.61m, at *y* = 0 for wave height = 0.3 m ……………... 52

2.8 Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of solitary wave at wavemaker with wave height=0.45 m …………………………….. 52

2.9 Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of solitary wave at three wave gages at x=8.8 m, 14.9 m and 18.7 m at y=0 for wave height=0.45 m ………... 53

2.10 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of solitary wave at wave gages along the length of the tank at y=0 for wave height=0.45 m ………………………... 53

LIST OF FIGURES (Continued)

Figure Page

2.11 Solitary wave pressure on the bottom (in N/m^2 on vertical axis) as a function of time (in sec on horizontal axis), along the length of the tank at y = 0 for H=0.45 m …………………………………… 54

2.12 Solitary wave particle velocity components (u, w) (in m/sec on vertical axis) as a function of time (in sec on horizontal axis) at x = 18.7 m, z = -0.61m, at y = 0 for wave height=0.45 m …………….. 54

2.13 Surface elevation at the wavemaker (in m on vertical axis) as a function of time (in sec on horizontal axis) for a Cnoidal wave of height H = 0.3 m and period T = 3.5 s, in depth h = 0.75 m ……... 55

2.14 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of Cnoidal waves at three wave gages at x=8.8 m, 14.9 m and 18.7 m with y=0, height H = 0.3 m and period T = 3.5 s…………………………………………………… 55

2.15 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of Cnoidal waves at three wave gages at x=4m, 16m and 24.95m at y=0 for wave height=0.3 m …………... 56

2.16 Numerical bottom pressure (in N/m^2 on vertical axis) as a function of time (in sec on horizontal axis), for Cnoidal waves at two gages at x = 4 and 16 m with y = 0 and wave height=0.3 m …. 56

2.17 Numerical Cnoidal wave particle velocity components (u, w) (in m/sec on vertical axis) as a function of time (in sec on horizontal axis) at x = 18.7 m, z = -0.61m, at y = 0 and for wave height = 0.3 m …... 57

2.18 Elevation of an Airy wave at wavemaker (in m on vertical axis) as a function of time (in sec on horizontal axis) for a wave height H = 0.2 m and period T = 3.0 s …………………………………...... 57

2.19 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of Airy waves at three wave gages at x=8.8 m, 14.9 m and 18.7 m at y=0 for wave height = 0.2 m …………… 58

LIST OF FIGURES (Continued)

Figure Page

2.20 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of Airy waves at three wave gages at x=4m, 16m and 24.95m at y=0 for wave height=0.2 m ………………….. 58

2.21 Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of Airy waves at two pressure gages at x=4m and 16m located at the bottom surface with y=0 and wave height=0.2 m ………………………………………………………. 59

2.22 Time (in sec on horizontal axis) versus numerical velocity components u, w (in m/sec on vertical axis) of Airy waves at x=18.7 m, z=-0.61m with y=0 and wave height=0.2 m ……......... 59

2.23 Performance (clock time in sec versus # of CPU) for Application-1 60

2.24 Performance (clock time in sec versus # of CPU) for Application-2 61

2.25 Performance of the parallel 3D-NWT, tested using several applications with *n*-levels=5, 6, 7 in the FMA ……………………. 62

2.26 Performance of the parallel 3D-NWT, tested using several applications with *n*-levels=5, 6, 7, 8 in the FMA …………………. 63

3.1 3D view of the bathymetry used in the benchmark experiment …... 92

3.2 Cross-sectional view of the bathymetry used in the benchmark experiment ……………………………………………………… 92

3.3 3D view of the bathymetry, used in the benchmark experiment, plotted using a Matlab script ……………………………………. 93

3.4 Plan view of the bathymetry used in the benchmark experiment … 93

3.5 3D view of the grid of the bathymetry used in BE and FE models .. 94

3.6 View of the wave propagation in the 3D wave basin at OSU …….. 94

LIST OF FIGURES (Continued)

Figure Page

3.7 View of the wave propagation in the 3D wave basin at OSU; shows a breaking wave over the triangular sloping shelf …………. 95

3.8 Top view of the wave propagation in the 3D wave basin at OSU; wave breaking and progress ………………………………………. 95

3.9 Top view of the wave propagation in the 3D wave basin at OSU; wave breaking and progress ………………………………………. 96

3.10 Top view of the wave propagation in the 3D wave basin at OSU; wave breaking and progress ………………………………………. 96

3.11 Motion input to the wavemaker …………………………………... 97

3.12 3D view of the bathymetry used in the benchmark experiment …... 97

3.13 FE models (top and bottom views) ……………………………….. 98

3.14 Benchmark problem-propagation of solitary wave ……………….. 98

3.15 Propagation of solitary wave displayed at locations x=1.3m, 1.5m, 1.7m, 2.0m, 2.2m and 2.4m (all locations are at y=0) respectively 98

3.16 Free surface elevations at y=0 (time in sec versus free surface elevation in m) …………………………………………………….. 99

3.17 Free surface elevations at y=-5m (time in sec versus free surface elevation in m) …………………………………….................... 99

3.18 Velocities U, V, W (in m/sec) from 3D-NWT at x=13.0m, y=0m, z=0.75m (Z is w.r.t. datum used in the benchmark experiment) ….. 100

3.19 Velocities U, V, W (in m/sec) from the benchmark experiment at x=13.0m, y=0m, z=0.75m ………………………………………… 100

LIST OF FIGURES (Continued)

Figure Page

3.20 Velocities U, V, W (in m/sec) from 3D-NWT at x=9.0m, y=3m, z=0.52m …………………………………………………………… 101

3.21 Velocities U, V, W (in cm/sec) from the benchmark experiment at x=9.0m, y=3m, z=0.52m ……………………………………….. 101

3.22 Velocities U, V, W (in m/sec) from 3D-NWT at x=11.5m, y=3m, z=0.43m …………………………………………………………… 102

3.23 Velocities U, V, W (in cm/sec) from the benchmark experiment at X=11.5m, Y=3m, Z=0.43m ……………………………………….. 102

3.24 Screenshots of propagation of solitary wave using FEM ………… 103

3.25 Close-up screenshot of the commencement of breaking wave over the triangular shelf in FE model …………………………………... 103

3.26 Comparison of X-velocity (U in m/sec) …………………………... 104

3.27 Comparison of Y-velocity (V in m/sec) …………………………... 105

3.28 Comparison of Z-velocity (W in m/sec) ………………………….. 105

3.29 Performance using several models with nlevels=5, 6, 7, 8 ……….. 106

3.30 Performance (#of CPUs versus clock time in hours) using benchmark problem ……………………………………………….. 107

4.1 Performance (#of CPUs versus clock time in sec) using wave basin model …………………………………………………….. 131

4.2 Performance (number of nodes versus clock time) using several models with nlevels=8 …………………………….................... 131

4.3 Evolution and propagation of the focused solitary wave using model-1 ……………………………………………………......... 132

LIST OF FIGURES (Continued)

Figure Page

4.4 Evolution and propagation of the focused solitary wave using model-2 ……………………………………………………........ 132

4.5 Evolution and propagation of the focused solitary wave using model-3 ……………………………………………………........ 132

4.6 Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of focused solitary wave using model-1 at three locations (starting at the edge first and ending at the middle) along the wavemaker for wave height=0.3 m ……………………………………………………. 133

4.7 Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of focused solitary wave using model-1 at four wave gages at (x,y) m =(8.8,0), (14.9,0), (14.9,-3.99) and (18.7,0) for wave height=0.3 m ……….. 133

4.8 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-1 at wave gages along the length of the tank at y=0 for wave height=0.3 m ……………………………………………………. 134

4.9 Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of focused solitary wave using model-1 at pressure gages located at the bottom surface along the length of the tank at y=0 for wave height=0.3 m ……………………………. 134

4.10 Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of focused solitary wave using model-1 at three locations (starting at the edge first and ending at the middle) along the wavemaker for wave height=0.4 m ……………... 135

4.11 Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of focused solitary wave using model-1 at four wave gages at (x,y) m =(8.8,0), (14.9,0), (14.9,-3.99) and (18.7,0) for wave height=0.4 m ……….. 135

LIST OF FIGURES (Continued)

Figure Page

4.12 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-1 at wave gages along the length of the tank at y=0 for wave height=0.4 m … 136

4.13 Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of focused solitary wave using model-1 at pressure gages located at the bottom surface along the length of the tank at y=0 for wave height=0.4 m ……………………………. 136

4.14 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-2 at three locations (starting at the edge and ending at the middle) along the wavemaker for wave height=0.3 m ………………………………... 137

4.15 Time (in sec on horizontal axis) versus numerical wave profile (in m on vertical axis) of focused solitary wave using model-2 at a wave gage at (x,y) m =(8.8,0) for wave height=0.3 m …………….. 137

4.16 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-2 at wave gages along the length of the tank at y=0 for wave height=0.3 m … 138

4.17 Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of focused solitary wave using model-2 at pressure gages located at the bottom surface along the length of the tank at y=0 for wave height=0.3m …………………………….. 138

4.18 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-2 at three locations (starting at the edge and ending at the middle) along the wavemaker for wave height=0.4 m ………………………………... 139

LIST OF FIGURES (Continued)

Figure Page

4.19 Time (in sec on horizontal axis) versus numerical wave profile (in m on vertical axis) of focused solitary wave using model-2 at a wave gage at (x,y) m =(8.8,0) for wave height=0.4 m …………….. 139

4.20 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-2 at wave gages along the length of the tank at y=0 for wave height=0.4 m …. 140

4.21 Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of focused solitary wave using model-2 at pressure gages located at the bottom surface along the length of the tank at y=0 for wave height=0.4 m …………………………….. 140

4.22 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-3 at three locations (starting at the edge and ending at the middle) along the wavemaker for wave height=0.3 m ………………………………... 141

4.23 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-3 at three wave gages at (x,y) m =(8.8,0), (14.9,0), (14.9,-3.99) for wave height=0.3 m …………………………………………………….. 141

4.24 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-3 at wave gages along the length of the tank at y=0 for wave height=0.3 m …. 142

4.25 Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of focused solitary wave using model-3 at pressure gages located at the bottom surface along the length of the tank at y=0 for wave height=0.3m ……………………………... 142

LIST OF FIGURES (Continued)

Figure Page

4.26 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-3 at three locations (starting at the edge and ending at the middle) along the wavemaker for wave height=0.4 m ……………………………….. 143

4.27 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-3 at four wave gages at (x,y) m =(8.8,0), (14.9,0), (14.9,-3.99) and (18.7,0) for wave height=0.4 m ……………………………………………... 143

4.28 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-3 at wave gages along the length of the tank at y=0 for wave height=0.4 m …. 144

4.29 Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of focused solitary wave using model-3 at pressure gages located at the bottom surface along the length of the tank at y=0 for wave height=0.4m ……………………………... 144

5.1 Propagation of a solitary wave on a triangular sloping shelf in the 3D wave basin at Oregon State University ………………………… 173

5.2 Close-up of a solitary wave on a triangular sloping shelf …………. 174

5.3 Snapshot of Cnoidal waves ……………………………………...... 175

5.4 Snapshot of Airy waves ……………………………………………. 176

5.5 Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of solitary wave at three wave gages at x=8.8 m, 14.9 m and 18.7 m at y=0 for wave height=0.3 m 177

LIST OF FIGURES (Continued)

Figure Page

5.6 Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of solitary wave at wavemaker for wave height=0.45 m ……………………………………………. 177

5.7 Time (in sec on horizontal axis) versus theoretical (analytical) and numerical wave profiles (in m on vertical axis) of Cnoidal wave at wavemaker for wave height=0.3 m …………………..................... 178

5.8 Time (in sec on horizontal axis) versus theoretical (analytical) and numerical wave profiles (in m on vertical axis) of Airy wave at wavemaker for wave height=0.2 m …………………..................... 178

5.9 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of Cnoidal waves at three wave gages at x=8.8 m, 14.9 m and 18.7 m at y=0 for wave height = 0.3 m ……………. 179

5.10 Performance (#of CPUs versus clock time in sec) using model-1 … 180

5.11 Performance (#of CPUs versus clock time in sec) using model-2 … 181

5.12 Performance (#of CPUs versus clock time in sec) using model-3 … 182

5.13 Performance (number of nodes versus clock time) using several models with nlevels=8 ……………………………...................... 183

5.14 Performance using several models with nlevels=5, 6, 7 …………... 184

5.15 Performance using several models with nlevels=5, 6, 7, 8 ………... 185

5.16 Performance (#of CPUs versus clock time in hours) using the benchmark problem ………………………………………………... 186

LIST OF TABLES

Table Page

2.1 Application-1: # of nodes *N* =2,592, # of tree levels in FMA (*n*-levels) = 6; 5 time steps ………………………………………………. 60

2.2 Application-2: # of nodes *N* =13,838, # of tree levels in FMA (*n*-levels) = 7; 5 time steps ………………………………………………. 61

3.1 Free surface peak elevation in m over still water shore line using FEM, BEM and lab experiment …………………………………… 104

3.2 Benchmark model: # of nodes=40228, # of tree levels in FMA=nlevels=7; 100 time steps ………………………………….... 106

4.1 Model of 3D wave basin: # of nodes=63150, # of tree levels in FMA=nlevels=8; 5 time steps …………………………………….... 130

5.1 Test model-1: # of nodes=2592, # of tree levels in FMA=nlevels=6; 5 time steps …………………………………….... 180

5.2 Test model-2: # of nodes=13838, # of tree levels in FMA=nlevels=7; 5 time steps …………………………………….... 181

5.3 Test model-3: # of nodes=63150, # of tree levels in FMA=nlevels=8; 5 time steps …………………………………….... 182

5.4 Benchmark model: # of nodes=40228, # of tree levels in FMA=nlevels=7; 100 time steps ………………………………….... 185

Dedication

To the Omniscient, Omnipresent, Ever-existing, Ever-new Joy -
The Source of Everything

and

To my parents, teachers, family and friends

An Efficient High-Performance Computing Based Three-Dimensional Numerical Wave Basin Model for the Design of Fluid-Structure Interaction Experiments

Chapter-1: Introduction

1.1 Introduction

Fluid-structure interaction (FSI) is an interesting and challenging interdisciplinary area comprised of fields such as engineering, computational science and mathematics. FSI has several practical engineering applications such as the development of guidelines for designing coastal infrastructure (e.g., bridges, levees) subjected to harsh environments from natural forces such as tsunamis, storm surges, etc. Development of accurate input conditions to more detailed and complex models involving flexible structures in a fluid domain is an important requirement for the solution of such problems. For example, the oceans are modeled in medium to large scales (hundreds of meters to several kilometers) whereas the model surrounding the structure (which is often flexible) is prepared in a finer grid to capture the phenomena such as wave breaking in greater detail. FSI researchers often employ methods that use results from physical wave basin experiments to assess the wave forces on structures. These experiments, while closer to the physical phenomena, often tend to be time-consuming and expensive. Experiments are also not easily accessible for conducting parametric studies. Alternatively, numerical models when developed with similar capabilities will complement the experiments very well because of the lower costs and the ability to study phenomena that are not feasible in the laboratory.

This dissertation is aimed at contributing to the solution of a significant component of the FSI problem with respect to the above-mentioned engineering applications, covering accurate input to detailed models, and a numerical wave basin to complement the large-scale laboratory experiments. To this end, this work contains a description of a three-dimensional numerical wave tank (3D-NWT), its enhancements, validation using large-scale experiments in the 3D wave basin at Oregon State University (OSU) and application of fluid-structure interaction to bridge structures. This code, which uses the fully nonlinear potential flow (FNPF) and the boundary element method (BEM), builds upon the research of Prof. Stephan Grilli of the University of Rhode Island and as such, his work and the theoretical background leading to it are extensively referenced. The capabilities and the limitations of this code are highlighted in the following section on literature survey.

Performing simulations involving fluid dynamics is computational-intensive and the complexity is magnified by the presence of the flexible structure(s) in the fluid domain. The models are also required to take care of large-scale domains such as a wave basin in order to be applicable to practical problems. Therefore, undertaking these efforts requires access to high-performance computing (HPC) platforms and development of parallel codes. With these objectives in mind, parallelization of the 3D-NWT is implemented and discussed in this dissertation.

1.2 Literature survey

This section surveys the literature of the two major components of this dissertation - boundary element method, fully nonlinear potential flow and free surface waves. The numbers in the brackets therein indicate the references provided in the bibliography at the end of the dissertation.

1.2.1 BEM: The advantage of BEM as familiar to engineers is that the dimensions of the model can be reduced by one. Thus, 2D models can be analyzed using a boundary mesh and 3D models can be discretized using a surface-only mesh. The governing differential equation is satisfied exactly within the 3D domain and approximately over the boundary. Problems such as free surface waves, and stress concentration can be solved very accurately. Also, it is easier to remesh (due to surface-only mesh) unlike the domain methods (e.g., finite element method), in the case of moving boundary problems (e.g., free surface waves) wherein regridding will help redistribute the nodes evenly during the wave propagation. However, this method results in unsymmetric, fully populated linear system matrices which require faster solution methods.

The BEM or the boundary integral equations method (BIEM) as it is sometimes called, can be studied from the perspectives of classical presentation of integral equations [1], numerical mathematics [2], mathematical analysis [3-4] and engineering applications [5-13]. The work of Grilli, et al on the propagation of highly nonlinear waves (which the present dissertation builds upon) is summarized in a chapter of a recent book on the BEM [12]. Reference [14] attempts to provide a balanced presentation of BIEM between the mathematical world and engineering. This list of references is not exhaustive and serve as a pointer for further exploration by interested readers. This method is currently receiving a renewed attention from the research community (including engineers and mathematicians) and new interesting practical developments and applications using the BEM are being proposed frequently in the literature.

The solution techniques of BEM are made efficient in order to be able to solve practical problems in engineering. To this end, the fast multipole method (FMM) or fast multipole algorithm (FMA) is an important development that brought BEM to the forefront in the world of numerical methods. A review of the relevant literature is available in [15]. References [16-19] include the latest developments in the fast solution methods using FMA and BEM as implemented by mathematicians. References [20-21] include several numerical examples in elastostatic BEM-FEM coupling and 3D elastodynamics and present models of a size up to a million nodal unknowns. Reference [22] presents practical engineering applications using these technologies (i.e., BEM, FMA) in an accessible manner for students and researchers.

1.2.2 FNPF and free surface waves: Potential theory has been well-developed in the classical literature and the interested reader can be referred to its mathematical developments [23-24]. While the Laplace equation that models the potential flow is a second-order linear elliptic partial differential equation (PDE), the nonlinear flow originates from the presence and computation of the free-surface. This model is applicable for the cases of inviscid, incompressible and irrotational flows which can model several physical situations of ocean wave propagation. In general, the effects of viscosity are negligible for large-scale bodies (such as ships), and also because viscosity of water is relatively small in magnitude. The assumption of incompressibility is valid in most of the ocean wave situations where there is no air-water mixing (i.e., no bubbles). Many results of classical hydrodynamics and their applications are indeed based on this theory.

A study of the validity of potential flow theory for describing steep gravity waves produced in an experimental tank was presented in [25]. A mixed Eulerian-Lagrangian scheme was used in two dimensions and the results demonstrated that the fully nonlinear potential flow yields results very close to the physical results unlike the linearized (small amplitude) theory. The solution method- vectorized Gaussian elimination with partial pivoting and the computation of the matrix elements limited the size of the problems that can be handled. For the integration of the free-surface boundary conditions, a fourth-order multistep Adams-Bashforth-Moulton scheme (ABM4) predictor-corrector method, started by the fourth-order Runge-Kutta (RK4) scheme, was used. For the linearized problem, the RK4 scheme is conditionally stable with $\Delta t^2 \leq 8\Delta x / \pi$, where Δt =time step and Δx =minimum grid spacing. The ABM4 is weakly unstable with a growth rate proportional to $\Delta t^6 / \Delta x^3$. Regridding combined with smoothing techniques were employed to minimize the instability issues.

The mechanisms of water-wave breaking are an interesting mathematical topic [26-27]. The latter reference described some of the historical developments using numerical models of potential flows to model the early stage of wave breaking phenomena.

Reference [28] describes the work of mathematicians Dold and Peregrine, who applied the boundary integral method for the propagation of 2D water surface waves including the overturning of water waves as they break. Their work was based on the Cauchy theorem boundary integral which permitted the use of an iterative solver and calculation of the gradients of higher derivatives of the potential. Explicit time stepping was based on a truncated Taylor series. This scheme improved efficiency by permitting relatively large time-steps while maintaining a desired degree of accuracy. A discussion on stability and accuracy indicated that this method didn't exhibit sawtooth numerical instabilities reported in other methods when time step sizes (which depends on point density) are sufficiently small. Detailed mathematical formulations and numerical examples of this method including the studies of stability, smoothing, etc. were presented in [29]. This method was found to be more efficient, accurate and stable [30] than the approaches given by Longuet-Higgins and Cokelet.

Professor P.J. Zandbergen of the Applied Mathematics Department of the University of Twente, Netherlands has advised several doctoral students as part of a major project on the numerical simulation of nonlinear water waves. This initiative generated substantial research work [31-34] by using the BIEM and FNPF. It is unclear if the software developed by this group is currently being maintained and offered as an open source for the research community.

For large objects with a characteristic dimension of the order of the wave length, viscous effects of the fluid flow can be neglected as well as compressibility effects [31]. Small-amplitude theory (with wave amplitude smaller than wave length or water depth) doesn't adequately describe the situations where nonlinearity is predominant. While the assumption of potential flow is not valid at all stages of a physical process, it is valid for many practical engineering situations and at the least can be very useful for a more general model. A detailed discussion of the mathematical properties such as existence, uniqueness and well-posedness can also be found in this reference. The numerical algorithm used a classical fourth-order Runge-Kutta method for time integration. The spatial discretization was done using the panel method- wherein the surface is divided into rectangular sub-surfaces called as panels with their midpoints denoted as collocation points. These methods are based on Taylor expansions in collocation points and are more like the finite difference methods. In comparison, the BEMs are based on a finite element approach. Romate [31] used a conjugate gradient squared (CGS) algorithm for the solution of the linear equation at a given time-step. The results were accurate and stable for linear and weakly nonlinear waves but displayed stability problems in the case of highly nonlinear waves.

The work of Broeze [32] focused on improving the panel method developed in [31] for highly nonlinear wave problems. A table presented therein showed a number of researchers favoring BEM over other methods to model these problems. The time integration schemes that were frequently used are Adams-Bashforth-Moulton (ABM), Runge-Kutta (RK) and Taylor Series ($2^{nd}/3^{rd}$ order). The CGS algorithm was used in this work in similar lines of its predecessor [31] with several additional optimizations in the matrix-vector multiplications. The number of operations for performing one iteration with CGS was about quadratic in the number of panels. An increased number of iterations were needed for larger problems due to the occurrence of matrix conditioning issues because of the different length to depth ratio of the domain. Thus, the increase in run times had been worse than quadratic in the number of unknowns. Generalized minimal residual (GMRES) was considered to be faster due to its better convergence behavior but it required additional memory than the CGS. One obvious disadvantage of the panel method over the BEM is that the geometry and solution is known only at the collocation points of the panels. This will necessitate some additional techniques to find the solution at the intersection of the surfaces.

Reference [33] continued from [31-32] by simplifying the algorithms therein to two dimensions and applied them to the hydrodynamics of floating and submerged bodies. This work also provided considerable theoretical background pertaining to this topic but the bodies considered in the numerical examples were of regular shapes such as a circular cylinder.

Reference [35] presented applications of BIEMs for 2D nonlinear water wave problems and compared the use of both the free space and the periodic Green's functions. The latter function facilitates the advantage of requiring integration only over the free surface. The numerical studies presented therein had not indicated an improved computational efficiency due to the use of the periodic Green's function, and the authors stated that the 3D problems will have an advantage. This work also used the Taylor series for time-stepping.

Cooker [36] presented a method based on Cauchy's integral theorem, conformal mapping and a truncated Taylor series for time-stepping to model the 2D wave propagation over irregular beds. Discretization was needed only on the free surface, resulting in a solution method that was faster than the ones that used a complete boundary mesh.

Lin, W.M. [37] studied the nonlinear behavior of the fluid motion near the intersection point of the free surface and a floating body (which was simplified to a piston wavemaker) using the potential flow in two dimensions. A numerical algorithm was proposed to accommodate the singularity resulting from the confluence of two analytical boundary conditions of fluid motion at this point. Cauchy's integral theorem was used to solve for the complex potential function along the boundary. The new algorithm satisfied both the free surface boundary condition and the wavemaker boundary condition at the intersection point.

A more recent work [38] applied the BEM to solve nonlinear planar free-surface waves and validated with wave generation using Stokes fifth-order theory and sinusoidal motion of a piston wavemaker in two dimensions. The theory had been extended to include the roll-motion of a 2D hull section. A fourth-order Runge-Kutta (RK4) scheme was used for time-integration. A direct LU solver was used to solve the system of equations which limited the size of the problems that can be solved.

Colicchio, et al [39] have described a domain-decomposition strategy for nonlinear flows (two-phase). In this work, they coupled potential flow BEM with a Navier-Stokes (NS) solver combined with a level-set (LS) technique resulting in an efficient and accurate method. This strategy was applied only for two dimensions. LU decomposition was used to solve the equations and time stepping was done using a second-order RK method. An overlapping region was defined between the two solvers wherein the interface location, pressure and velocity information was exchanged. Both methods solved the problem at this overlapping region and the exchange of data was accomplished in time through the staggered time steps between the BEM and the NS-LS, i.e., the explicit second-order RK scheme was nested with the predictor-corrector method.

Grilli and co-workers [40-48] have developed an accurate and versatile three-dimensional numerical wave tank based on fully nonlinear potential flow (FNPF) theory. This development is based on the higher-order BEMs, combining both free-space Green’s functions and the FMA in combination with the GMRES. The numerical complexity was proportional to the problem size (e.g., the number of nodes in the mesh). The free-surface time updating was based on a second-order Taylor series expansion in an Eulerian-Lagrangian form. Wave absorption was accomplished by a combination of free-surface (pressure) damping and a lateral active absorbing wavemaker. The range of the applications that can be solved, however, has been limited by the size of problems that could be solved on a single processor. Also the implementation did not have a piston wavemaker to accept the wave motion, which is an important requirement to be able to model the physical wave basin. Further, it will be interesting to be able to implement the 3D waves (e.g., directed/focused) that FSI researchers would like to study. Validation using large-scale laboratory experiments and demonstration of its capabilities is also of significance for its wider acceptance as an engineering tool.

Mathematical details of Grilli's work in two dimensions, which can provide an insight into the more recent developments in 3D, can be found in [49-52]; references [50] and [52] have a review of the corner/edge problems- such as the intersection between the free surface and the wavemaker that required incorporation of the compatibility conditions. The time-stepping scheme in Grilli's work [49] was based on Dold and Peregrine's method [30] of using a higher-order Taylor expansion in time. This is a stable and fast method (by allowing larger time-steps) in comparison with the RK or ABM schemes, which require multiple evaluations of the BIE for several intermediate times per time step, with all the advantages of an explicit method. Reference [53] provided details of the incorporation of the wave energy absorption due to an absorbing beach (AB) and absorbing piston (AP) and extends these ideas to the implementation of bottom friction in two dimensions. Reference [54] by Grilli, et al described the use of BEM in the physical space to discretize fully nonlinear potential flow unlike the works of many other researchers who used mapping or complex variables and highlighted the numerical challenges such as accuracy and stability and handling of corners at the wavemaker and the bottom surface.

1.3 Scientific contributions of the dissertation

The scientific contributions of this dissertation are written in the form of four manuscripts which are in the process of being submitted to journals and conferences. As such, these chapters may be read in sequence or independently. The introductory parts, mathematical formulations, etc. may have some common elements so as to provide a common theme and reach a wider readership (it is assumed that a smaller number of readers peruse all journals when compared with those that read one or two journals).

Manuscript-1 presents the mathematical details behind the 3D-NWT and its enhancements made by the author- namely, generation of a solitary wave (often referred to as a tsunami wave) and periodic waves (such as Cnoidal and Airy waves) using a piston wavemaker and validation of the results obtained using these capabilities.

Manuscript-2 contains comparison of numerical results with a so-called benchmark experiment [55] performed in the 3D wave basin at OSU. The bathymetry is quite complex (has non-uniform depth) and has sloping surfaces with a triangular shelf. Numerical results are obtained using both the 3D-NWT and LS-DYNA [59] codes which use very different models and numerical methods.

Manuscript-3 describes the incorporation of directed/focused waves into the 3D-NWT. It provides detailed mathematical formulations, results and comparison of a few alternatives studied for this purpose.

Manuscript-4 discusses the parallelization efforts for solving large-scale models in general and also in a specific sense as applied to the 3D-NWT code. Performance studies included in detail therein provide a foundation for the assertion of O(n) runtimes of this code particularly when applied to large-scale problems. One of the sections includes the insight and experience gained by the author during the undertaking of this effort.

The final chapter (i.e., Chapter 6) named as general conclusions has the remarks along with the possible extensions of this research work.

The appendices at the end provide a description of the algorithms and details of the input files of the 3D-NWT. There are also instructions on how to compile and run the package and visualization (post-processing) of output. This software package is intended to be eventually made available in the public domain and is contingent upon permission from its former contributors. One of the appendices introduces some of the work that did not make the main chapters of this dissertation- using LS-DYNA for the analysis of an Oregon Department of Transportation (ODOT) bridge on the Oregon coast subjected to tsunamis. These are problems of practical interest to analysts and designers of coastal infrastructure who would otherwise have to resort to expensive and limited cases of physical experiments or empirical design methods. As such, this work would bring these latest FSI technologies to practicing engineers.

The author attempted to provide a balanced presentation of the materials by combining both visual and mathematical elements. Thus it is hoped that this work will be interesting to the readers that have either inclination and also to those who tend to be somewhere on the spectrum.

An Efficient Parallelized 3-Dimensional Numerical Wave Tank to Model Large-Scale Wave Basin Experiments

Seshu B. Nimmala and Solomon C. Yim
School of Civil & Construction Engineering, Oregon State University (OSU),
Corvallis, OR 97331, USA

Stephan T. Grilli
Department of Ocean Engineering, University of Rhode Island (URI),
Narragansett, RI 02882, USA

Abstract

This paper presents the parallel implementation of an accurate and efficient three-dimensional numerical wave tank (3D-NWT), based on fully nonlinear potential flow (FNPF) theory, and its extension to incorporate the motion of a laboratory piston wavemaker, to simulate experiments in Oregon State University's large-scale 3D wave basin. Particular effort was devoted to make the code efficient for large-scale simulations using high-performance computing (HPC) platforms (e.g., large multi-core systems), in order to complement the experimental 3D wave basin at OSU (or similar facilities elsewhere), with a means of simulating, and thus optimizing, experiments ahead of time. To date, waves that can be generated include solitary waves, Cnoidal waves, and Airy waves. In this paper, we detail the model, mathematical formulation, wave generation, and analyze the performance of the parallelized code. Experimental or analytical/theoretical comparisons with numerical results are then provided for several cases, which demonstrate both the accuracy and practical applicability of the code to engineering problems.

1 Introduction

Over the past decade, following the increasing power of modern computing platforms, accurate and efficient three-dimensional (3D) numerical wave tanks (NWTs) have been developed and refined, which simulate complex processes of wave generation, propagation over arbitrary bottom topography, interaction with ocean structures, and dissipation over sloping beaches. Such phenomena have typically been investigated by performing laboratory experiments in large-scale 3D wave basins. Using NWTs, however, one can simulate and optimize planned physical experiments, ahead of time, and thus more efficiently devote time and efforts to the targeted laboratory experiments, since such facilities are both expensive and lengthy to operate. Additionally, NWTs allow calculation of time series of detailed flow parameters (e.g., velocity, pressure) everywhere in the domain, while these are usually available only at a limited number of experimental probes (and at the sacrifice of flow-field intrusion), in laboratory experiments.

It is beyond the scope of this paper to provide an exhaustive literature review of all methods used to develop NWTs. We will only present and discuss a limited list of references, targeted to the type of models used in our work, i.e., models simulating nonlinear waves based on inviscid fully nonlinear potential flow (FNPF) theory, and implemented based on a higher-order boundary integral equation (BIE) method, in finite element (FEM) formalism, which is referred to as the boundary element method (BEM).

The main advantage of the BEM, which benefits engineering applications, is that the dimensionality of the model is reduced by one. Thus 3D problems can be discretized using a surface-only mesh, which reduces the effort devoted to developing relevant numerical grids. Additionally, the governing equation (here the Laplace equation) is satisfied exactly within the 3D domain and only approximately over the boundary. However, due to the reduced dimensionality, this can be done using both a higher-order and highly resolved BEM surface grid. Hence, problems such as free surface waves or stress concentration (with close proximity to the surface), can be solved very accurately. Also it is easier to regrid (due to the surface-only grid) unlike domain-discretization based methods (e.g., FEM) in the case of moving boundary problems (e.g., free surface waves), wherein regridding will help redistributing the nodes evenly during the wave propagation. One drawback of the standard BEM, however, is that it yields unsymmetric and fully populated linear system matrices, which require fast solution methods or a more advanced implementation, which creates sparse matrices. This is further discussed later.

The BEM can be studied from the perspectives of a classical presentation of integral equations [1], numerical mathematics [2], mathematical analysis [3-4] and engineering applications [5-13]. Reference [14] attempts to provide a balanced presentation of BIE methods, between the mathematical and engineering worlds. The BEM is currently receiving a renewed attention from the mathematical and engineering research communities, and new interesting practical developments and applications are being proposed in the area of high-performance computing (HPC) using the BEM. In particular, in the context of the present work, we present the parallelization and efficient HPC implementation of a solution technique which makes the BEM much more efficient, in order to be able to solve the large-scale practical problems that occur in engineering. This technique, the fast multipole method (FMM) or fast multipole algorithm (FMA) is an important, and relatively recent, development that has brought the BEM to the forefront in the world of numerical methods. While more details are provided later, the principle of the FMA is first to approximate the free space Green's function of the problem governing equation(s) by a series of spherical harmonics. Then, for each BEM discretization node, a hierarchy of increasingly distant sub-domains is defined, in which the full Green's function is used in the nearest sub-domain and a decreasing number of harmonics is used to represent the Green's function, down to no harmonics, for increasingly distant sub-domains. Doing so both accelerates the set-up of the non-zero coefficients in the BEM system matrices, through numerical integration, and creates large empty blocks in such matrices for sub-domains beyond the cut-off distance. When properly implemented, a BEM-FMA code can achieve a N log N numerical complexity (where N denotes the size of the boundary discretization). A review of the relevant FMA literature is available in [15]. References [16-19] include the latest developments in the fast solution methods using FMA and BEM, as implemented by mathematicians. References [20-21] include several numerical examples in elastostatics, BEM-FEM coupling, and elastodynamics, in which grid sizes up to millions of nodal unknowns were used. Reference [22] presents practical engineering applications using FMA-BEM models, in an accessible manner for students and researchers.

2. FNPF theory and modeling of free surface waves

Potential flow theory, which solves inviscid, incompressible, Euler equations for irrotational flows, has been well-developed in the classical literature, and for a long time; the interested reader can be referred to the [23-24] for its mathematical developments. While the Laplace equation, which governs potential flow is a second-order linear elliptic partial differential equation (PDE), nonlinearity in wave processes originates from the presence of, and equations, governing the free-surface (i.e., the dynamic and kinematic free surface boundary conditions). The FNPF model is accurate to model ocean wave propagation since, outside of the surfzone, the effects of viscosity are usually negligible, except in thin bottom and free surface boundary layers; additionally, in the presence of submerged or floating ocean structures, viscosity effects are also negligible for large-scale bodies (such as ships) and, for small bodies (e.g., pipelines), while they may be important for the flow around the structure, they will typically be negligible for the large scale wave flows. The assumption of incompressibility is valid in most ocean wave situations where there is no air-water mixing (i.e., no bubbles), which is mostly the case for non-breaking waves. Many results of classical hydrodynamics and their applications are thus based on FNPF theory, or other equations derived from it through perturbation expansions. One severe limitation of FNPF theory, however, is that wave overturning and breaking will cause flow (i.e., breaking jet) penetration, which violates the governing equations and thus will interrupt any computations based on this flow model. Hence, in FNPF-NWT, numerical absorbing beaches have been developed to prevent waves from overturning, through the absorption of the energy of steeper waves; more details on this aspect of NWT simulations are given later.

A study of the validity of potential flow theory for describing steep gravity waves produced in an experimental tank, up to and including overturning, was presented in [25] for deep water plunging breakers. This was based on work detailed in Refs. [26-29], in which a Mixed Eulerian-Lagrangian (MEL) scheme is used in two dimensions (2D) for the time updating, combined to a higher-order BIE, in a conformally mapped space. A similar 2D study was later performed for solitary wave shoaling and breaking over plane slopes [30,31], based on a MEL and a higher-order BEM in the physical space [52,53,55], which allowed simulating a variety of wavemakers (or more exact methods [33]) to generate waves, as well as surface piercing slopes and absorbing beaches [33,34] (see also Refs. [54,58] for details). Results in these studies demonstrated that a FNPF model yields results very close to the physical results for strongly nonlinear waves, up to and including wave overturning (unlike linearized (small amplitude) or low-order wave theories).

In the 2D-FNPF models initially developed [25-29,52,55], both the solution method, which was based on a Gaussian elimination with partial pivoting (or similar), and the computation of the matrix elements, limited the size of the problems that could be handled. For the integration of the free-surface boundary conditions, a fourth-order multistep Adams-Bashforth-Moulton scheme (ABM4) predictor-corrector method (started by the fourth-order Runge-Kutta (RK4) scheme), or an explicit second-order Taylor expansion scheme, were used. Regridding, combined in some cases with smoothing techniques were employed to minimize numerical (sawtooth) instabilities.

More specifically Ref. [26] describes the work of applied mathematicians Dold and Peregrine, who applied the BIE method for the propagation of 2D water surface waves, including overturning. Their work was based on Cauchy integral theorem in a conformally mapped space, which permitted using an iterative solver and calculating the gradients of higher derivatives of the potential. An explicit time stepping scheme was used, based on truncated Taylor series. This scheme improved both the stability of the solution and the efficiency, by permitting to use relatively large time-steps while maintaining a desired degree of accuracy. A discussion on stability and accuracy indicates that this method did not exhibit sawtooth instabilities reported in other methods, which used ABM or RK schemes, when time step sizes (which depends on point density) are sufficiently small. Detailed mathematical formulations and numerical examples of this method including the studies of stability, smoothing, etc. were presented in reference [27]. This method was found to be more efficient, accurate and stable [28] than the approaches given by Longuet-Higgins and Cokelet [32].

Lin [29] studied the nonlinear behavior of the fluid motion near the intersection point of the free surface and a floating body (which was simplified to a piston wavemaker) using the potential flow in two dimensions. A numerical algorithm was proposed to accommodate the singularity resulting from the confluence of two analytical boundary conditions of fluid motion at this point. [This aspect was revisited and extended by Grilli, et al [53,55] in the context of a different BEM model implementation as well as the generation of waves by paddle wavemakers.] In [29], Cauchy's integral theorem was used to solve for the complex potential function along the boundary. The new algorithm satisfies both the free surface boundary condition and the wavemaker boundary condition at the intersection point.

Zandbergen and co-workers, at the Applied Mathematics Department of the University of Twente (Netherlands), pursued a major project on the numerical simulation of nonlinear water waves, particularly in 3D, in the late 1980's, early 1990's. This initiative generated substantial research work [35-38] based on using the BEM and FNPF approaches. This work confirmed that, for large objects with characteristic dimensions on the order of the wavelength, viscous as well as compressibility effects can be neglected in the fluid flow [35]. While the assumption that the flow is potential is not valid at all stages of a physical process, it is valid for many practical engineering situations, and at least should be very useful for a more general model. A detailed discussion of mathematical properties, such as existence, uniqueness, and well-posedness of BIE-BEM problems can also be found in this reference. In these first 3D-NWTs based on FNPF, the time updating scheme was based on a classical fourth-order Runge-

Kutta method. The spatial discretization of the BIE was done using a low-order panel method, wherein the surface was divided into rectangular sub-surfaces called panels with their midpoints denoted as collocation points. When extended to higher-order, such methods are based on Taylor expansions at the collocation points and are more akin to finite difference methods. By contrast, the BEM models are more akin to a FEM approach. Romate [35] used a conjugate gradient squared (CGS) algorithm for solving the linear system of discretized BIE equations at a given time-step. The results were accurate and stable for linear and weakly nonlinear waves, but displayed significant stability problems in the case of highly nonlinear waves.

Broeze [36] focused on improving the panel method developed in [35] for highly nonlinear wave problems. A table presented in this reference showed that many researchers favored a BEM over other methods to model waves problems, because of the well-known advantages of the methods discussed above. In most works, time integration schemes were Adams-Bashforth-Moulton (ABM), Runge-Kutta (RK) and Taylor Series (2nd/3rd order). The CGS algorithm was also used in this work, as well as similar numerical approaches as for its predecessor [35], with several additional optimizations in the matrix-vector multiplications. The number of operations for performing an iteration with CGS is about quadratic in the number of panels/nodes (i.e., a N^2 complexity). More iterations are needed for larger problems due to the occurrence of matrix conditioning issues, when using large length to depth ratios of the computational domains, which typically occurs in wave propagation problems. In the latter cases, the increase in computational times will be worse than quadratic in the number of unknowns. The generalized minimal residual (GMRES) method is considered to be faster and to have better convergence, but it requires additional memory than the CGS. One obvious disadvantage of panel methods over the BEM is that the geometry and solution are known only at the discrete collocation points of the panels. This necessitates additional techniques to find the solution at the intersection of various surfaces.

For completeness, Ref. [39] presents applications of BIEs for 2D nonlinear water wave problems and compares the use of both free space and periodic Green's functions. The latter function is expected to offer the advantage of requiring integration only over the free surface but good computational efficiency is not demonstrated in the numerical studies. The authors state that 3D problems will have an advantage following this approach. This work also used the Taylor series for time-stepping. Cooker [40] further developed the method in [26], based on Cauchy's integral theorem, conformal mapping and truncated Taylor series for time-stepping, to model 2D wave propagation over irregular beds. Discretization is needed only on the free surface, resulting in a solution method, which is faster than the ones that use a full boundary mesh. A more recent work, by Vinayan, et al [41], applied the BEM to solving nonlinear 2D free-surface waves and validated it with wave generation using Stokes fifth-order theory and sinusoidal motion of a piston wavemaker. The theory has been extended to include the roll-motion of a 2D hull section. A fourth-order Runge-Kutta (RK4) scheme is used for time-integration. A direct LU solver is used to solve the system of equations which limits the size of the problems that can be solved. Colicchio, et al [42] proposed a domain-decomposition strategy for nonlinear flows (two-phase) in two dimensions. In this work, they coupled potential flow BEM with a Navier-Stokes solver, combined with a level-set technique resulting in an efficient and accurate method. LU decomposition is used to solve the equations and time stepping is done using a second-order RK method. An overlapping region is defined between the two solvers wherein the interface location, pressure and velocity information is exchanged.

As a premise to the present work, Grilli and co-workers [43-51] developed an accurate and versatile 3D-NWT based on FNPF theory. This development, which directly extended their earlier 2D work (e.g., [30,31,52-55]) was initially based on combining a higher-order (3rd-order) boundary element methods (BEM) based on a free-space Green's function in the physical space, to an explicit time updating based on 2nd-order Taylor series expansions in an Eulerian-Lagrangian formulation [43-46,48]. To improve the NWT computational efficiency, the method was later extended [47,49-51] to combine both a free-space Green's function (for the near field) to a fast multipole algorithm (FMA) (for the far-field). The solution method was GMRES and in the FMA implementation, the numerical complexity was nearly proportional to the problem size *N*. Wave absorption is accomplished by a combination of free-surface (pressure) damping and a lateral active absorbing wavemaker (such as detailed in [33] for the 2D-NWT). This work on the propagation of highly nonlinear waves in a 3D-NWT was summarized in a recent book chapter [51].

Despite its demonstrated success for academic-type applications, the range of engineering applications that could practically be solved using this 3D-NWT, however, was limited by the size of problems that could be solved on a single processor. This was because the FMA and the BEM had been implemented in scalar mode in the original 3D-NWT. Also, while wavemakers had been used in other applications (e.g., [44,45,49]) the FMA implementation of the 3D-NWT did not have a piston wavemaker for simulating laboratory wave motion, which is an important requirement to be able to model the physical wave basin, nor was there a fully operational absorbing beach (AB).

It is thus the purpose of this work to improve the 3D-NWT implementation by: (i) fully parallelizing the code, (ii) adding a snake piston wavemaker for arbitrary wave generation, and (iii) an efficient absorbing beach. This will bring the NWT to the level of an engineering tool, able to simulate meaningful laboratory experiments in a large-scale wave basin, such as used at Oregon State University (OSU).

In addition to the references to the earlier work on developing the 3D-NWT [43-51], mathematical and numerical details of the work of Grilli, et al in 2D can be useful in providing insight into more recent work in 3D; these can be found in [33,52-55]; references [53] and [55] have the review of corner/edge problems, such as the intersection between the free surface and the wavemaker, which required incorporation of extended compatibility conditions. The time stepping scheme in Grilli [52] is similar to the method in Dold and Peregrine [28], which based on higher-order Taylor expansions in time (however, implemented in the physical space, rather than in a conformally mapped space that has no corners). This method is stable and fast (by allowing larger time-steps) in comparison to the RK or ABM schemes, which require multiple evaluations of the BIE for several intermediate times per time step, with all the advantages of an explicit method. Reference [56] provides details of the incorporation of the wave energy absorption due to absorbing beach (AB) and absorbing piston (AP) in 2D, based on the method first proposed by Grilli and Horrillo [33], and extends these ideas to the implementation of bottom friction in 2D as well. Reference [57] further highlights the numerical challenges of the physical space implementation of the NWT (again in 2D), such as accuracy and stability, and handling of corners at the wavemaker and bottom surface.

In the following, we first present a summary of the mathematical model and numerical algorithms underlying the 3D-FNPF-NWT, then describe the enhancements implemented into the NWT and their validation. As discussed above, these enhancements include, in particular, arbitrary wave generation by a snake piston wavemaker (with the development of a wave generation module based on the control software driving the actual wavemaker at the OSU 3D wave basin), and wave dissipation in a numerical absorbing beach. Applications for the generation of solitary waves, periodic waves, such as Cnoidal and Airy waves, are presented, which validate the new implementation. A discussion on the parallelization of the NWT algorithms (particularly FMA-BEM) and resulting performance characterization is included, which demonstrate the newly added capabilities.

3 Mathematical model

The 3D-NWT code is based on FNPF theory, which solves incompressible, inviscid, Euler equations for irrotational free surface flows. In this case, mass conservation is expressed as the Laplace's equation, which represents the governing PDE (of second-order linear elliptic type) for such flows. The nonlinearity in the flows originates from two free surface boundary conditions (i.e., kinematic and dynamic, as will be discussed later).

Specifically, in potential flows one defines, $\phi =$velocity potential function, with the velocity given by $\mathbf{u} = \nabla\phi = (u, v, w)$. Laplace's equation is obtained from continuity equation as, $\nabla.\,\mathbf{u} = 0 \Rightarrow \nabla.\,\nabla\phi = 0$, i.e.,

$$\nabla^2\phi = 0 \qquad (1)$$

In the above, the Laplacian operator can be expressed as $\Delta = \nabla.\nabla = \nabla^2$.

Figure 1 shows the general set-up and typical geometry of the 3D numerical wave tank (NWT), in the case of wave generation by a snake piston wavemaker on boundary Γ_w, with stroke function $x_p(y,t)$. The free surface boundary Γ_f has an elevation defined with respect to the still water level corresponding to the (x,y) plane (that is, the vertical coordinate z is negative downwards toward the bottom boundary Γ_b). The origin of axes at $y = 0$ is located at the middle of the wavemaker and coordinate x is positive in the direction of the wave propagation pointing away from the wavemaker. The four sides of the tank are vertical boundaries, one of them being the moving snake wavemaker, the opposite one, parallel to the (*y,z*) plane, is a moving absorbing piston boundary Γ_{ap}. The two sidewalls Γ_s, parallel to the (*x,z*) plane, are fixed in location. The impermeable bottom boundary, which is represented with constant depth h_0 on the can be specified to be sloping or with an arbitrary geometry.

On the free surface, the potential satisfies the nonlinear kinematic (KBC) and dynamic (DBC) boundary conditions. Both conditions are required since there are two unknowns (position and potential) on the free surface. The KBC states that the normal velocity of the free surface interface must be equal to the normal fluid velocity at the surface, following a fluid particle at the free surface; this means that such particles remain at the free surface. The DBC, obtained from Bernoulli's equation (i.e., an integration of Euler equations), states that the pressure on the free surface equals the atmospheric pressure, which is assumed to be zero. If ***R(t)*** is the position vector (*x,y,z*) of a fluid particle on the free surface, *g* is the acceleration due to gravity, then we have, from KBC and DBC respectively:

$$\frac{D\mathbf{R}}{Dt} = \mathbf{u} = \nabla\phi \tag{2}$$

$$\frac{D\phi}{Dt} = -gz + \frac{1}{2}\nabla\phi . \nabla\phi \tag{3}$$

For fixed boundaries, the no flow condition (zero flux) is specified as $\frac{\partial\phi}{\partial n} = 0$, where **n** is the normal vector to the surface.

For a moving (piston) wavemaker boundary, the motion (stroke function) and velocity are prescribed based on waves to be generated (by way of a wavemaker theory); hence, the specified BCs are:

$$x = x_p(y,t); \quad \frac{\partial \phi}{\partial n} = \mathbf{u}_p . \mathbf{n} \qquad (4)$$

where $\mathbf{u}_\mathrm{p}$ is the velocity of the wavemaker, and **n** the normal to the wavemaker. The initial free surface boundary condition (at $t = 0$) is given by the still water level ($z = 0$) and zero potential (Dirichlet BC), and all other surfaces have zero or specified normal flux (Neumann BC).

4 Numerical Methods

In the 3D-NWT developed by Grilli's group, a higher-order boundary element method (BEM) was implemented to solve the Laplace equation. In this method, Green's second identity is applied to transform the PDE into a BIE, which is discretized on the boundary. The reduction of the problem dimensionality by one in the BEM results in considerable saving in mesh size, as compared with methods such as FEM. However, the BEM yields system matrices, which are unsymmetric and fully populated. The BEM version used here is referred to as a direct BEM, since the unknowns are potential and normal flux (and their time derivatives for the second Laplace's equation solved, see below).

Governing equation: Green's second identity transforms Laplace's equation into a BIE expressed on the boundary Γ of the fluid domain:

$$\alpha(\mathbf{x}_l)\phi(\mathbf{x}_l) = \int_{\Gamma(t)} \left\{ \frac{\partial \phi}{\partial n}(\mathbf{x}) G(\mathbf{x},\mathbf{x}_l) - \phi(\mathbf{x}) \frac{\partial G}{\partial n}(\mathbf{x},\mathbf{x}_l) \right\} d\Gamma \qquad (5)$$

where $\alpha(\mathbf{x}_l)$ is proportional to the exterior solid angle [47] made by the boundary at the collocation point $\mathbf{x}_l$ and the 3D free space Green's function of Laplace's equation is defined as,

$$G(\mathbf{x},\mathbf{x}_l)=\frac{1}{4\pi r} \tag{6}$$

where $r=\left|\mathbf{x}-\mathbf{x}_l\right|$ is the distance from the source point $\mathbf{x}$ to the collocation point $\mathbf{x}_l$ (both on the boundary), $\mathbf{n}$ is the outwards normal vector on the boundary. The normal derivative of the Green's function reads,

$$\frac{\partial G}{\partial n}(\mathbf{x},\mathbf{x}_l)=\nabla G.\,\mathbf{n}=-\frac{1}{4\pi}\frac{\mathbf{r}.\,\mathbf{n}}{r^3} \tag{7}$$

Time integration: Second-order explicit Taylor series expansions are used to find both the new free surface position and potential at the next time step,

$$\mathbf{R}(t+\Delta t)=\mathbf{R}(t)+\Delta t\frac{D\mathbf{R}}{Dt}(t)+\frac{(\Delta t)^2}{2}\frac{D^2\mathbf{R}}{Dt^2}(t)+O\{(\Delta t)^3\} \tag{8}$$

$$\phi(t+\Delta t)=\phi(t)+\Delta t\frac{D\phi}{Dt}(t)+\frac{(\Delta t)^2}{2}\frac{D^2\phi}{Dt^2}(t)+O\{(\Delta t)^3\} \tag{9}$$

These expressions are found to yield an explicit, stable, and efficient time stepping scheme (as compared to, e.g., ABM-RK schemes).

In Eqs. (8-9), zero$^{\text{th}}$-order coefficients in the Taylor series are given by the free surface geometry and the solution of the BIE for potential ϕ at time t. First-order coefficients are evaluated from the BCs (KBC and DBC) of free surface, also as a function of geometry and the BIE solution for ϕ at time t. Second-order coefficients are expressed as:

$$\frac{D^2\mathbf{R}}{Dt^2}=\frac{D\mathbf{u}}{Dt}=\frac{\partial\mathbf{u}}{\partial t}+\mathbf{u}.\,\nabla\mathbf{u}=\nabla\frac{\partial\phi}{\partial t}+\nabla\phi.\nabla(\nabla\phi) \tag{10}$$

$$\frac{D^2\phi}{Dt^2}=-g\frac{Dz}{Dt}+\frac{1}{2}\frac{D(\nabla\phi.\nabla\phi)}{Dt}=-gw+\mathbf{u}.\frac{D\mathbf{u}}{Dt} \tag{11}$$

where w = vertical component of velocity. These can be calculated as a function of the geometry, the BIE solution for potential ϕ and the solution of another similar BIE for the time derivatives of the potential at time *t*. It should be emphasized that this second BIE uses the same system matrix as the first one and has boundary conditions, which can be calculated as a function of the solution of the first BIE. Details can be found in Refs. [43,46].

5 Salient features of the algorithms

The salient features of the numerical algorithm combining higher-order BIE solutions and an explicit time updating can be summarized as follows:

- Second-order coefficients of the Taylor series used in the time updating are obtained from time derivatives of the potential and flux, which are solved from another BIE, expressed on the boundary. Since this is done in the same geometry as that for the first BIE for the potential, the same discretized BEM algebraic system matrix is used with a different right hand side; hence the solution of the second BIE is very fast [43].
- The time step is adapted as a function of the minimum distance between two nodes on the free surface, based on a constant mesh Courant number $C_0 \sim 0.45$ [43].
- The solution uses GMRES [48] wherein the matrix vector products are replaced by the fast multipole algorithm (FMA) for distant sources points relative to a given collocation node [47,51]. This algorithm uses multipole expansions and tree data structures, and avoids the assembling of the discretized system matrix. The theoretical computational

complexity of the FMA is $O(N \log N)$, where N is the number of nodes on the boundary. This is a very good improvement over the pure GMRES implementation, which resulted in a $O(N^2)$ performance. More details on the FMA implementation can be found in [47].

- The majority of the 3D-NWT code was developed in Fortran, while the FMA algorithms were written in C language.
- The user input to the 3D-NWT was designed to be minimal [43], in the form of the dimensions of the NWT and number of elements in each direction, plus a few other control and FMA parameters. The 3D surface mesh is automatically generated based on input parameters, thus resulting in considerable time savings for the user.
- A node regridding technique can be automatically applied for any user-specified iterations so that free surface nodes can be redistributed evenly on the grid during computations. This option helps prevent the occurrence of inaccuracies and instabilities due to an uneven distribution of nodes during wave propagation, which occurs due to Stokes drift for strongly nonlinear waves.

6 Piston motion input

The boundary conditions for a plane (2D) piston wavemaker were derived in Ref. [58], for generating solitary, Cnoidal, or other type of elementary waves. Grilli, et al [59] extended this 2D wavemaker generation to arbitrary irregular waves, based on a specified wave energy spectrum. A 3D snake flap piston wavemaker was implemented in the 3D-NWT to study wave focusing [44,49]. Here, a snake piston wavemaker is implemented in the 3D-NWT, with the wavemaker boards/paddle motion driven by actuators such as those used in OSU's laboratory tank (these expressions are developed to be general enough so that each rigid individual paddle in the snake wavemaker can oscillate independently while maintaining flexible connections at the edges with other boards):

The wavemaker displacement is defined by the stroke displacement: $x = x_p(y,t)$ (Fig. 1) velocity: $u = u_p = dx_p/dt$, and acceleration $a = a_p = du_p/dt.$

Let **n** be the outward normal vector to the piston wavemaker boundary, and (**s**, **m**) are tangential vectors with, **s**.**m** = 0 (**s** pointing in the global z direction; see Fig. 1) and **n** = **s** x **m**. We define,

$$\phi_n = \frac{\partial \phi}{\partial n} = \mathbf{u}_p . \mathbf{n} \; ; \; \phi_m = \frac{\partial \phi}{\partial m} = \mathbf{u}_p . \mathbf{m} \; ; \; \phi_s = \frac{\partial \phi}{\partial s} = \mathbf{u}_p . \mathbf{s} \qquad (12)$$

with $\mathbf{x}_p = x_p\,\mathbf{i}$ and $\mathbf{u}_p = u_p\,\mathbf{i}$ (where (**i, j, k)** denote unit vectors along the global x, y, and z axes respectively), which yields the boundary condition for the second BIE problem as,

$$\frac{\partial^2 \phi}{\partial t \partial n} = \frac{D\mathbf{u}_p}{Dt} . \mathbf{n} + \mathbf{u}_p . \frac{d\mathbf{n}}{dt} + \phi_n(\phi_{ss} + \phi_{mm}) - \phi_m \phi_{nm} \qquad (13)$$

Now, let the coordinates of the edges of a plane individual waveboard in the snake piston wavemaker be (x_2,y_2,z) and (x_1, y_1,z), so that,

$$\mathbf{m} = (x_1 - x_2)\mathbf{i} + (y_1 - y_2)\mathbf{j} \; ; \mathbf{s} = \mathbf{k} \; ; \mathbf{n} = -(y_1 - y_2)\mathbf{i} + (x_1 - x_2)\mathbf{j}$$

Finally, with these definitions,

$$\frac{d\mathbf{h}}{dt} = (\mathbf{u}_{p1} - \mathbf{u}_{p2})\mathbf{j} \qquad (14)$$

since the actuators move the wavemaker paddles only in the *x* direction. In the 3D-NWT, derivatives with respect to *n* are obtained from the BIE solutions while derivatives with respect to *s* and *m* are obtained from analytical derivatives in sliding 4th-order polynomials [43,46].

7 Wave generation

To compare 3D-NWT results with those of laboratory experiments, a new module (named Wavegen), capable of generating the piston wavemaker motion, was developed based on the algorithms used in the driver software of the wavemaker of the 3D wave basin at OSU. As Wavegen was implemented as a separated module from the 3D-NWT, this ensures that either code can be used independently- for instance, the output of Wavegen can be used in any other numerical code (such as LS-DYNA [60]) or a different module can be used to feed the piston motion into the 3D-NWT. In its present state of development, the Wavegen module offers three types of wave generation capabilities: solitary, Cnoidal and Airy. A detailed discussion for generating more complex wave profiles (such as Stokes 5th order waves) can be found in [70,71].

7.1 Solitary Wave: The mathematical formulations and algorithms implemented to generate a first-order (Boussinesq) solitary wave are summarized below. These are similar to the implementation made in the earlier 2D-NWT [54,58];

additional discussions can be found in references [61] and [62].

Let H be the wave height and h the water depth. The solitary wave celerity and (equivalent) wave number are,

$$C=\sqrt{g(h+H)} \quad ; \quad k=\sqrt{\frac{3H}{4h^3}} \tag{15}$$

The target solitary wave profile is given, for $t \in \{-\infty,+\infty\}$ by,

$$\eta(x,t) = H\,\mathrm{sech}^2\left\{k(Ct-x_p)\right\} \tag{16}$$

With these definitions, the paddle time history is implicitly given by,

$$x_p(t) = \frac{H}{kh}\tanh\left\{k(Ct-x_p)\right\} \tag{17}$$

which yields the total waveboard displacement as,

$$x_p(t\rightarrow\pm\infty) = \pm\frac{H}{kh} = \pm\sqrt{\frac{4Hh}{3}} \tag{18}$$

Let $x_{pf} = 0.999\frac{H}{kh}$; hence, the time taken by paddle to travel from x_p = 0 to x_{pf} is,

$$t_f = \frac{1}{kC}\left\{3.8+\frac{H}{h}\right\} \tag{19}$$

Due to wave symmetry with respect to $x = 0$, the total stroke is from $x_p = -x_{pf}$ to $+x_{pf}$, and its duration is thus, $t_s = 2t_f$

Fig. 2 shows snapshot of a solitary wave in 3D-NWT.

7.2 Cnoidal Waves: The mathematical formulations and algorithms implemented to generate the Cnoidal wave are summarized below. These are similar to the implementation made in the earlier 2D-NWT [54,58]; additional discussions can be found in [61-64]. The approximate range of validity [63] for the Cnoidal wave theory is $h/L < 0.125$ and $L^2H/h^3 > 26$ (the latter being known as the Ursell number or Stokes parameter).

Let H be the wave height, h the water depth, T the wave period, and L the wavelength. For Cnoidal waves, the wave number is defined as, $k = \frac{2K(m)}{L}$, with the celerity, $C = L/T$. The wave shape is defined as,

$$\eta_c(x,t) = (h_t - h) + Hcn^2(\theta_c, m) = Hf(\theta) \tag{20}$$

$$f(\theta) = \frac{(h_t - h)}{H} + cn^2(\theta_c, m) \tag{21}$$

$$\theta_c = 2K(m)\left(\frac{t}{T} - \frac{X_o}{L}\right); X_o(t) = L\left(\frac{t}{T} - \frac{\theta_c(t)}{2K(m)}\right) \tag{22}$$

with the following parameters: h_t = distance between trough and bottom; h_c = distance between crest and bottom; $x_p(t)$ = wave paddle displacement; m = elliptic parameter = κ^2; κ = modulus of the elliptic integrals; $cn(\theta,m)$ = Jacobian elliptic function; $K(m) = K(\kappa)$ = first kind complete elliptic integral; $E(\theta,m)$ = second kind incomplete elliptic integral, and $E(\kappa) = E(\pi/2,\kappa)$ = second kind complete elliptic integral. Note, $cn(\theta,m)$, $K(m)$, $E(\kappa)$, m and κ are dimensionless. Further details of these parameters are available in Ref. [65].

General equation for the wave paddle displacement is,

$$x_p(t) = \frac{H}{kh}\int_0^\theta f(\theta)\,d\theta \tag{23}$$

Also,

$$L = \kappa\sqrt{\frac{16h^3}{3H}}K(\kappa) \tag{24}$$

$$h_t = h_c - H = \frac{H}{\kappa^2 K(\kappa)}[K(\kappa) - E(\kappa)] + h - H \tag{25}$$

$$T = \sqrt{\frac{16hh_t}{3gH}\frac{h}{h_t}} \left\{ \frac{\kappa K(\kappa)}{1+\frac{H}{h_t\kappa^2}\left(\frac{1}{2}-\frac{E(\kappa)}{K(\kappa)}\right)} \right\} \tag{26}$$

7.3 Airy Waves: Airy waves, also known as linear sinusoidal waves, are generated as in the earlier 2D-NWT [54,58], based on linear wavemaker theory, whose equations are summarized below. More detailed discussion can be found in reference [66]. For a sinusoidal wave of height *H,* period *T* = *2π/σ,* and wavelength *L* = *2π/k_p*, in water depth *h,* free surface elevation is defined as η,

$$\sigma^2 = gk_p \tanh(k_p h) \tag{27}$$

$$\eta = \frac{H}{2}\cos(k_p x - \sigma t) \tag{28}$$

with the Eq. (27) being the linear dispersion relationship. To generate such a wave in a wave tank, linear wavemaker theory predicts the maximum stroke *S* of the piston wavemaker paddle as,

$$\frac{H}{S} = \frac{2(\cosh(2k_p h) - 1)}{\sinh(2k_p h) + 2k_p h} \tag{29}$$

The paddle displacement is then obtained as,

$$x_p(t) = \frac{S}{2}\sin\sigma t \tag{30}$$

8 Discussion of results

The following results are obtained using a 3D numerical wave tank of length 25 m, depth 0.75 m, and width 1 m or 0.5m, for solitary wave or Cnoidal and Airy waves, respectively. Although the 3D wave basin at OSU has larger dimensions (length 48.8 m and width 26.5 m), the reduced dimensions (along length and width) are acceptable since in this application we are generating 2D waves (in the vertical plane). The main benefit of using a smaller NWT is to yield faster computational times. Results include the comparison with experiments [67] and theoretical wave profiles. Observations are discussed below as well as in figure captions.

8.1 Solitary Wave: The NWT-BEM grid has 200 elements along the length, 5 elements along the width and 8 elements along the depth. This yields a total number of nodes N = 6,138 and elements M = 5,280. Two cases are studied for this purpose, with wave heights H = 0.3 and 0.45 m. Results obtained with H = 0.3 m are shown in Figs. 3 to 7. It took 7.82 hrs with 8 CPUs for 600 time steps and a 13 sec maximum time. Results obtained with H = 0.45 m are shown in Figs. 8 to 12 and took 7.91 hrs with 8 CPUs for 600 time steps and a 12.75 sec maximum time. The numerical wave profiles (at both wavemaker and wave gages) and velocities are found to agree well with the experimental results for both the 0.3 and 0.45 m wave heights, although the experimental wave heights are slightly taller than the numerical values for the 0.45 m case. The wave absorption capabilities (absorbing piston and absorbing pressure beach) are working well so no reflected waves can be seen in the wave gage records.

8.2 Cnoidal Waves: The NWT-BEM grid has 200 elements along the length, 5 elements along the width, and 5 elements along depth. This yields a total number of nodes N = 4,896 and elements M = 4,050. The wave height is H = 0.3 m and period T = 3.5 s. Results are illustrated in Figs. 13 to 17 and took 20.5 hrs on a 8 CPU computer with a 27.6 sec maximum time. The numerical wave profile at the wavemaker is compared with the theoretical (analytical) wave profile in Fig. 13, which shows a good agreement. Wave absorption is working well as indicated in results from wave gages and pressure gages. The periodicity of Cnoidal waves is illustrated well in this case (from the repeatability).

8.3 Airy Waves: The grid used is the same as for Cnoidal waves. The wave height H = 0.2 m and period T = 3.0 s. Results are illustrated in Figs. 18 to 22 and took 30.5 hrs with 8 CPUs, for a 40.6 s maximum time. The numerical wave profile at the wavemaker is compared with the analytical wave profile in Figure 18 and shows a good agreement. Wave absorption is working well as indicated in results from wave gages, pressure gages, and velocity gages. We see, these results become quasi-steady for time t > 20 s.

9 Parallelization and performance studies

This section reviews numerical features of the 3D-NWT's implementation, discusses alternatives explored for parallel computing, choices adopted, and presents performance studies and insight gained in the process.

9.1 Serial Version: The initial BEM algorithms of the 3D-NWT code [43-46] were implemented in scalar mode using Fortran 77, an implementation language that for a long time has been dominant in the domain of scientific computing owing to its speed, capabilities, and familiarity with scientists and engineers. In standard Fortran, the memory is statically assigned in common blocks, during compilation. Such a design, which was the frequent design choice in most large scientific codes before the advent of more recent object-oriented programming, permits subroutines called in sequence to access the data that is of interest in a straight-forward fashion. One disadvantage for evolving codes (such as part of research projects), however, is that the data structure is spread all over the code and it is thus more difficult to modify the code as other associated code may directly dependent on such modifications. With the addition of new features over several years (which is inevitable in evolving computational/numerical software), this dependency between data and implementation becomes increasingly more complex to manage.

9.2 Alternatives: In order to parallelize the 3D-NWT with the FMA implementation [47] (which was initially only scalar mode), an off-the-shelf component, the distributed parallel multipole tree algorithm (DPMTA) library, developed by the Electrical Engineering department of Duke University [68] needed to be included. This library was written in C using the message passing interface (MPI) library. Tests on DPMTA indicated that it yielded nearly linear speedup with respect to the number of processors, for the same problem size, and also with increase in data size.

In its original form, the 3D-NWT has a significant serial part in the computations (e.g., mesh generation/regridding, iterative scheme of solution at each time step, and also time-stepping until a user-specified number of time steps or a maximum time). In order to gain maximum advantage from MPI, there should be a division of mesh data and work among the processors and minimization of message passing. However, this would require a major redesign and rewrite of the code. The risks involved in doing this are the introduction of software defects (bugs) and the requirement of extensive testing. An alternative that can be considered is to entirely rewrite/design the code with the latest software architectures, but this would mean reimplementing lots of research accomplished over a decade, and this approach is also not immune to risks mentioned in case of redesign either. So a practical strategy that is considered in the present case is that it is far more efficient (in terms of developmental time and efforts) to implement the NWT parallelization using OpenMP [69], rather than the DPMTA library (and MPI), to make the computations run faster. This approach will also take advantage of multi-core systems - the current paradigm shift in the processor manufacturing industry.

Parallel codes using OpenMP work by creating a user-specified number of threads. These threads are different from processes (an example of a process is a serial program) in that they are light weight, independent instructions that execute within a process. All threads created by the same process share the same memory space and have access to the same data in physical memory, without duplicating the data, which avoids transferring data as needed in the message passing paradigm (e.g., as used by MPI). However, an OpenMP software developer has to be extremely cautious to take care that threads do not write on the same physical location in memory at the same time and avoid the several possible pitfalls of multi-threaded programming. The number of threads can be as many as the number of processors, in order to achieve load balancing (ideally each processor taking equal load), but this is not necessarily the case. However, using too many threads (e.g., more than the number of processors) can cause the system to become unresponsive to other tasks. A practical point of concern is that the overhead for parallelization is not negligible (including that for the creation of threads). The size of and speed-up resulting from parallel components must be significant enough to overcome this overhead. In other words, the problem size (or model size in a simulation) should be large enough as to achieve good speedups.

9.3 Implementation Using OpenMP: Although this was initially thought to be a fairly straight-forward task, the efforts required to parallelize the 3D-NWT using OpenMP turned out to be quite a bit more significant and challenging. The BEM and time stepping parts of the 3D-NWT code were written in Fortran while the FMA library, which the BEM algorithms use, was written in C. Fortran routines make calls to the FMA library, which in turn calls the Fortran routines. Hence, in the initial code, there was an inherent heterogeneous language implementation as well use of a hybrid compiler technology, i.e., gcc (GNU C compiler) and Intel Fortran, with different paradigms of passing memory chunks from one part to another. Hence, it proved to be quite a challenging task to identify the blocks (via profiling) in code that are major bottlenecks, and implement OpenMP in such areas, with careful modifications such that no two threads write to the same memory location, while making the code "work" in an incremental development cycle.

9.4 Performance Studies: Here, we assess the performance of the newly added parallelization capability of the 3D-NWT code. The FMA uses multipole expansions and tree data structure [15,22], and avoids the full assembling of the algebraic system matrix in the solution. Although this could have been anticipated, our results clearly show that, in addition to problem size (i.e., *N, M*), the code run time is also influenced by the choice of FMA parameters such as the number of tree levels.

The applications used in this performance study correspond to wave generation/propagation in a 3D tank with the geometry of OSU's wave basin. The length and width of the wave basin are 48.8 and 26.5 m, respectively. For simplicity, we use the canonical case of a plane solitary wave of height $H = 0.39$ m, generated over a water depth of $h = 0.78$ m. The number of terms in the multipole expansion are 8 and the number of time steps are 5 (this small number is aimed at speeding up the performance tests and CPU time for larger problems is simply linear with the number of time steps). The FMA cube length (which should be specified to be at least twice the sum of the maximum dimension with an allowance for wave absorption) covering the wave basin is 105 m. Further details such as the number of tree levels (*n*-levels), etc. are provided below for each model application. A Dell Precision WorkStation 690 with eight Intel Xeon

3 GHz processors, 64GB of RAM and a 64 bit Redhat Linux Enterprise 5 was our dedicated test bed. This is an example of a small shared memory type of computing systems. Larger multi-processor computers could of course be used to realize improved CPU times for larger models.

Tables 1-2 summarize the tested performance of the 3D-NWT, using two applications in the same wave basin, with coarse to fine discretization, and Figs. 23-24 graphically illustrate these results. Application-1 has N = 2,592 nodes and Application-2 has N = 13,838 nodes. The number of tree levels is varied from 6 to 7, so as to obtain the best clock times with 8 processors. Reported speedups are given by the ratio of CPU time, using a single processor, divided by total CPU time using multiple processors (= T_1/Tp). The maximum speedups that have been achieved in these applications range from 2.48 to 2.51 (i.e., about one-third the number of processors), which implies that CPU time for an application, using a single processor, can be divided by these numbers to get an estimate of total CPU time on a multi-processor system. It is observed that the speedup is not markedly increasing beyond ~5-6 processors. This can only be expected as the speedups are dependent on the size of parallel components out of the total computation and the associated overhead. The code has inherently sequential (the code that necessarily has to run in sequence and cannot be parallelized) components and some parallelizable components; using additional processors would only reduce CPU times of the parallelizable components. For smaller models, an increase in the number of processors does not cause a significant reduction in processing times because of the additional overheads (due to thread creation, setup, etc) overcoming the gains in parallel computation. From the user's perspective, this would mean that one has to increase the model size (which gives rise to larger parallelizable components in the code) in general to realize better speedups with larger number of processors. Nevertheless, the maximum speedup that can be realized for a given model is limited by the sequential component of the code and thus cannot increase indefinitely with the number of processors. These observations are demonstrated by the first two applications (where the number of nodes in the second application is 6 times that of the first but both have about the same speedup for 8 processors).

Figure 25 illustrates the performance of the parallel 3D-NWT using one or eight processors, for the same solitary wave problem, using grids for the same wave basin (the largest one having 26,838 nodes). The FMA parameter number of tree levels is varied from 5 through 7. It can be observed that the fastest clock times of all these levels with 8 processors (solid lines) indicate a nearly linear performance (O*(N))* with the BEM node numbers *(N).* This behavior was already identified for the scalar 3D-NWT using the FMA [47,50], for large enough node numbers (usually more than 6,000 or so), and is also visible on Fig. 25 for the one-processor curve, when using the new parallel version.

Figure 26 shows the same kind of results as in Fig. 25, but for much larger grids in the applications (the largest one having 149,250 nodes), with variations in the FMA level parameter (number of tree levels varied from 5 to 8). [Note, for completeness, this figure includes results of Fig. 25 as well, for the smaller grids.] Again, the smaller CPU times are obtained with the maximum number of levels (8), when using 8 processors, and for large enough *N*, the increase with *N* becomes nearly linear numbers.

Some additional observations can be made based on results in Figs. 25 and 26. The smaller CPU times are achieved when increasing the FMA *n*-levels, with increasing grid size *N*. This is applicable with processors ranging from 1 to 8. Based on these figures, the 3D-NWT user can thus select a relevant FMA parameter *n*-levels, as a function of the number of processors, to achieve an optimal clock time for a given application. For instance, one can reduce the number of processors on a given system, to accommodate other users or other processes, by adopting an appropriate *n*-levels, without being too penalized in CPU time.

9.5 Insight Gained: Some of the insight and experience gained during this parallelization effort are summarized below, to serve as guidelines for developers undertaking similar efforts and for users to get the present code to run faster.

1. *Fast multipole algorithm (FMA) input parameter - number of tree levels*: This parameter indicates the number of hierarchical levels into which the "cube" (containing the discretization) is divided. The CPU speedup is considerably affected by this parameter as the work shared by the threads is dependent on the number of cells in the hierarchy. The user should be

able to experiment and determine the optimum parameter for a specified number of threads and given problem size. The practical values of this parameter range from 4 to 8. A much larger number of tree levels (>8) would considerably increase the memory requirement and the run may not be able to fit within the memory resources of the computing system. This limitation, however, can be alleviated by the future availability of bigger (and better) hardware resources.

2. *FMA input parameter - cube length*: This is the “cube” size surrounding the entire geometry of the discretization. The cube is defined with the origin of model at its center. The user has to identify the maximum dimension from the origin of the model, estimate the possible change in length (due to wave absorption for instance) and use twice this value for the cube length. One should use just as low a value as possible, for optimal performance of the FMA algorithms.

3. *FMA input parameter - number of terms in a multipole expansion*: The larger this number, the better the accuracy of the expansions but the larger the execution (CPU) time. While this parameter’s range can be 4 to 16, the practical value is around 6 through 8.

4. *The distribution of nodes* (or the grid geometry and number of nodes) in the FMA cube can also have an effect on the CPU time. A study done for one of the applications, using a cube size of 60 m instead of 120 m,

showed a reduction in CPU time from 19.76 to 8.53 hrs. The reason is that a cube size of 60 m caused nodes to be more uniformly distributed in FMA cells and thus reduced the computational effort.

10 Concluding remarks

This paper presented improvements in the implementation of an existing 3D-NWT, based on FNPF theory, to make it an efficient tool to complement similar experimental facilities, such as OSU's 3D wave basin or elsewhere. The numerical algorithms and mathematical formulations involved in furthering the development of wave generation capabilities (i.e., for solitary, Cnoidal and Airy waves) are provided. Comparisons with experiments for simple waves illustrate that the numerical and experimental results are in good agreement and provide a strong basis for the use of the 3D-NWT code for all appropriate and practical engineering purposes.

The insight gained from the parallelization effort, along with the performance studies, provide guidelines for a more efficient use of the code and also could help with similar efforts on developing other numerical codes that are to be run on multi-core processors. Linear performance of the parallel algorithms in the 3D-NWT, for a sufficiently large number of BEM nodes, is demonstrated and the BEM + GMRES + FMA + parallel capabilities provide an "optimal" combination of tools for solving problems, such as 3D nonlinear free surface waves, in a reasonable CPU time, on small (and larger) computer clusters.

Additional investigations, not detailed here, in improving wave generation using solid wavemakers, were also conducted to try to implement the wavemaker motion necessary to generate waves based on Stokes 5th order wave theory, but currently there is no direct, practical, methodology to implement this capability. A few references in this regard have been provided in a previous section on wave generation. Some possible ideas include the integration of the wave profile to obtain the wavemaker motion. It should also be noted that the generation of exact stream function waves using a "porous" piston wavemaker was already made in the 2D-NWT in an earlier work by the third author that could be easily incorporated in the 3D-NWT. However, such a wavemaker cannot be simulated in a physical facility such as OSU. Nevertheless, finding a relevant piston wavemaker motion that could create approximate fully nonlinear periodic waves in a physical facility could be investigated using the present improved 3D-NWT. Such an addition, which could be done in future work, would benefit both the numerical wave tank and equivalent laboratory wave basins.

11 Figures and tables

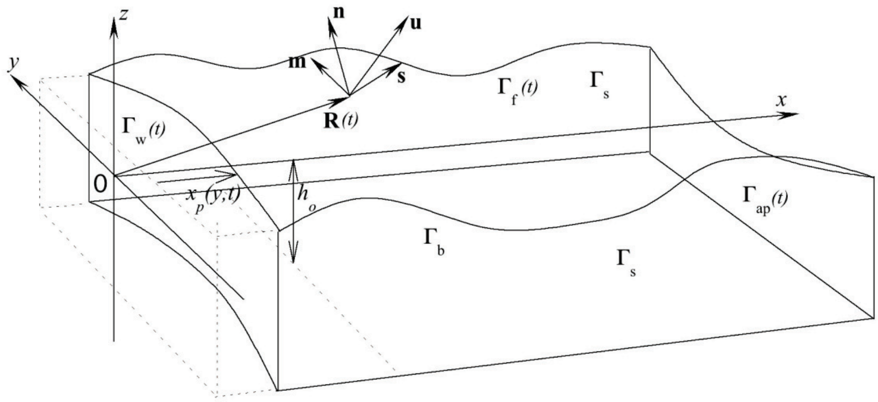


Fig. 1 Sketch of 3D-NWT geometry and parameters, for wave generation by a snake piston wavemaker (notation and details of mathematical model can be found in section 3)

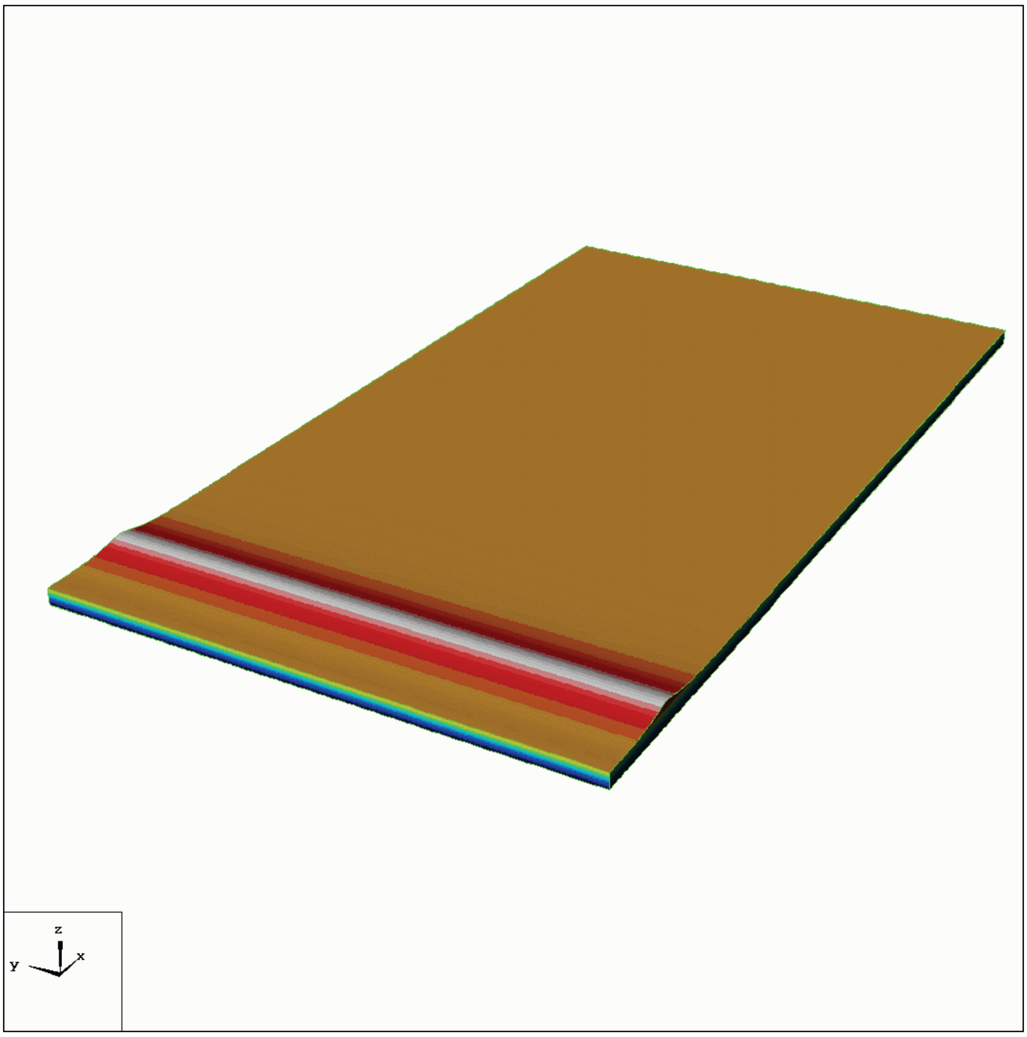


Fig. 2   Snapshot of 3D numerical wave tank propagating a solitary wave

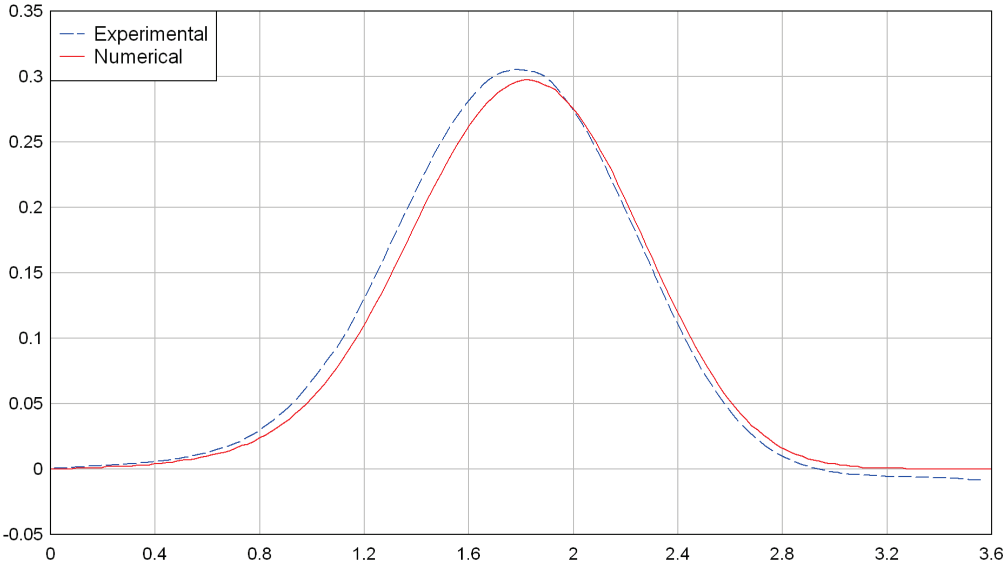


Fig. 3 Surface elevation (in m on vertical axis) versus time (in sec on horizontal axis), at the wavemaker, for a solitary wave of height H = 0.3 m, in depth h = 0.75 m.

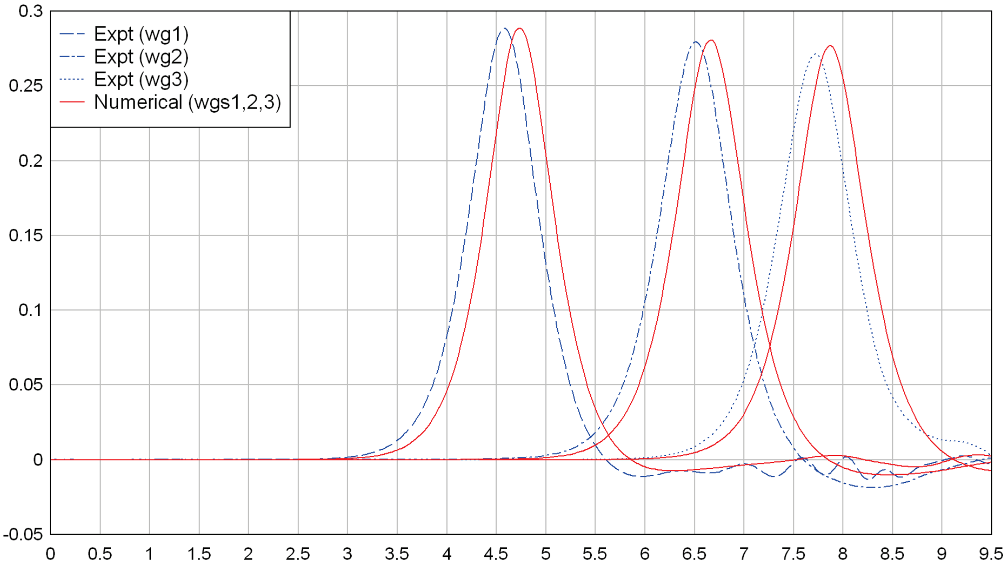


Fig. 4 Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of solitary wave elevations at three gages, at x = 8.8 m, 14.9 m, and 18.7 m, at y = 0 for wave height = 0.3 m; the numerical waves are found to lag slightly behind the experimental waves, but the results show a good overall agreement.

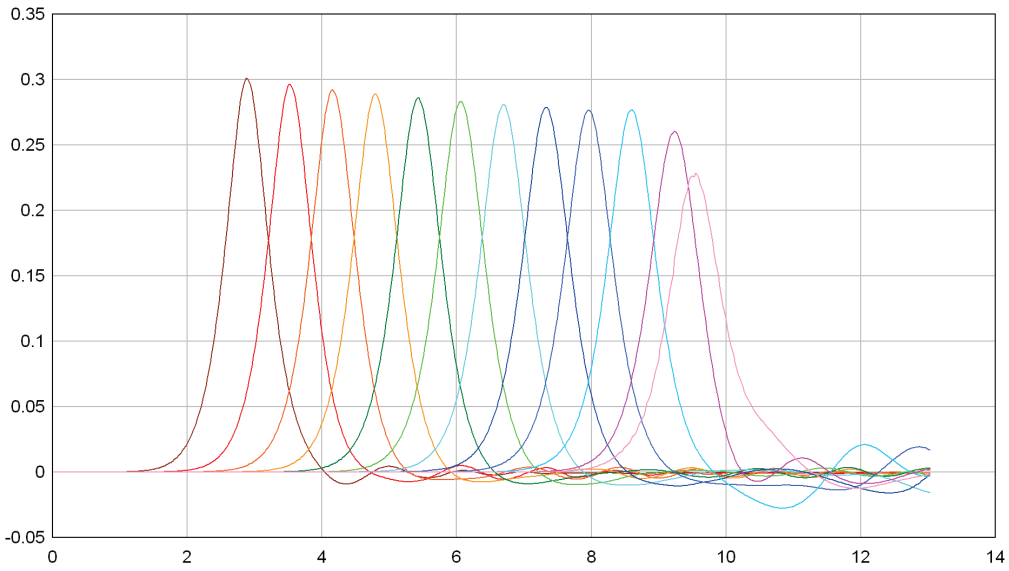


Fig. 5 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of solitary wave elevations at wave gages along the length of the NWT, at $y = 0$ for wave height=0.3 m; we see, wave gage records are free from reflected waves, indicating that the NWT wave absorption capability near the far end is working well to absorb the incident wave.

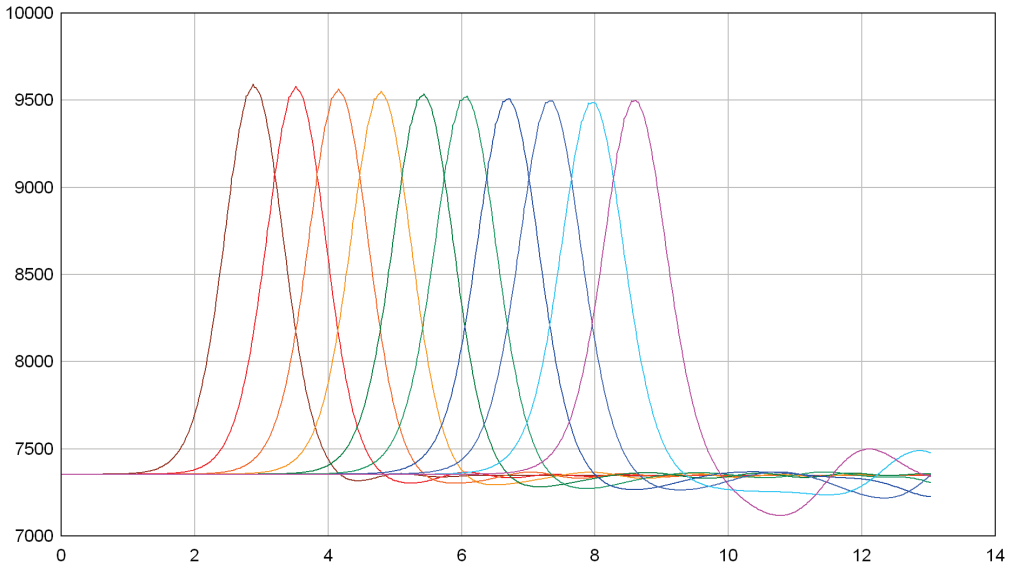


Fig. 6 Solitary wave pressure on the bottom (in N/m^2 on vertical axis) as a function of time (in sec on horizontal axis), along the length of the tank at $y = 0$ for wave height = 0.3 m; baseline is static pressure = 0.75*9810=7357.5 N/m^2

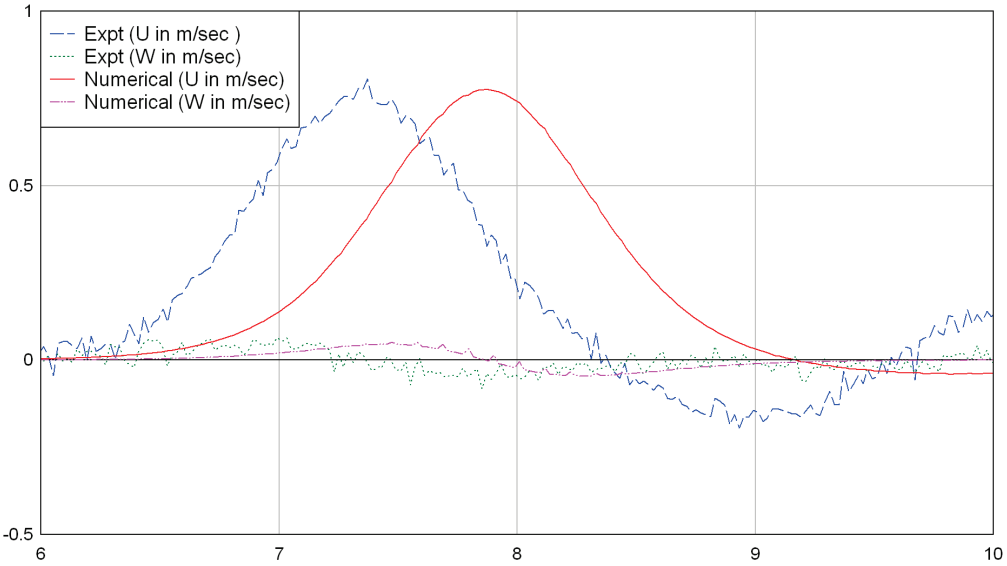


Fig. 7 Solitary wave particle velocity components (u, w) (in m/sec on vertical axis) as a function of time (in sec on horizontal axis) at x = 18.7 m, z = -0.61m, at y = 0 for wave height = 0.3 m; numerical profiles are lagging behind experimental profiles, but there is a very good agreement of their magnitude and shape (the experimental velocity measurement is more noisy)

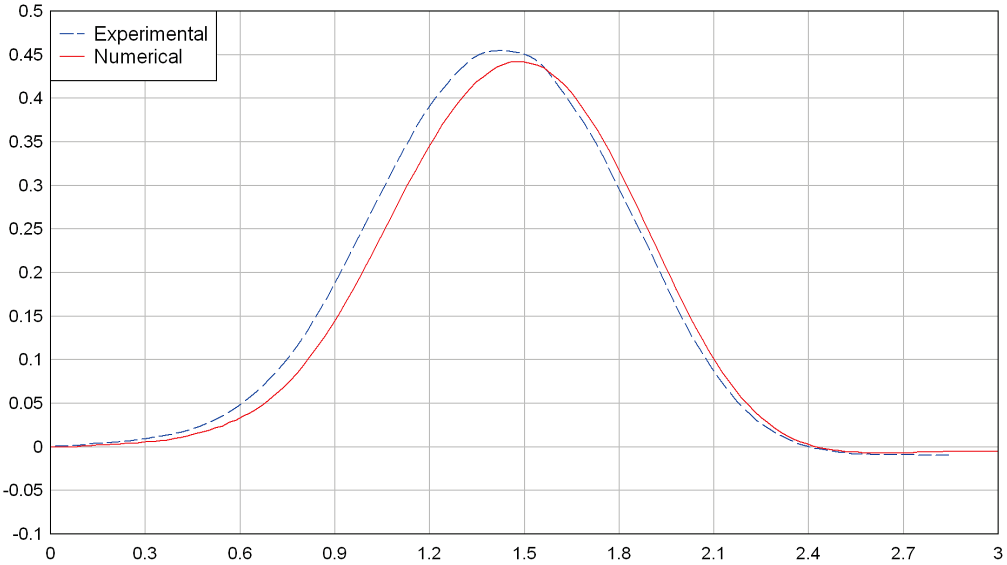


Fig. 8 Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of solitary wave at wavemaker with wave height=0.45 m

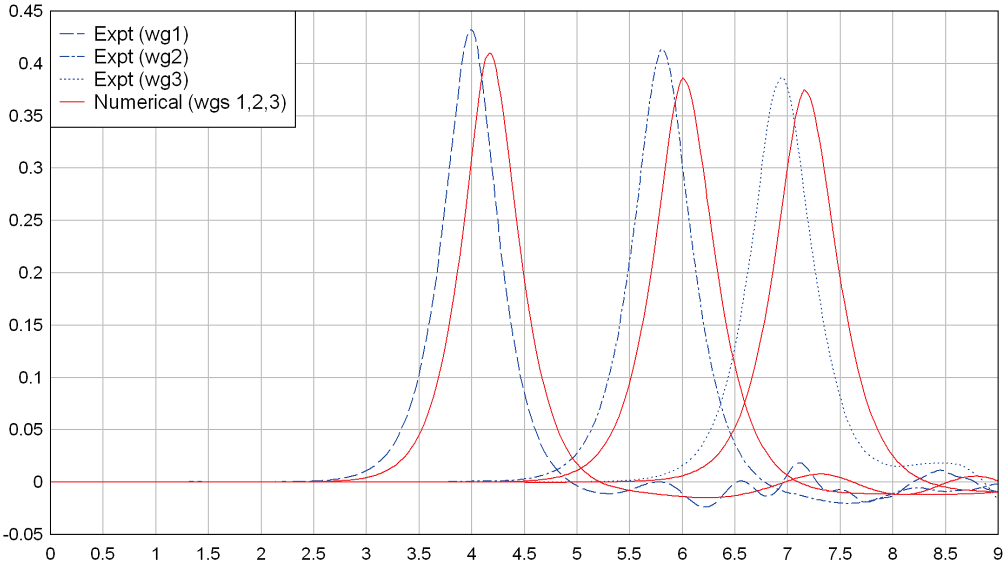


Fig. 9 Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of solitary wave at three wave gages at x=8.8 m, 14.9 m and 18.7 m at y = 0 for wave height = 0.45 m; the numerical waves are found to lag slightly behind the experimental waves, but the results show a general agreement of geometry with the experimental wave heights being a little taller

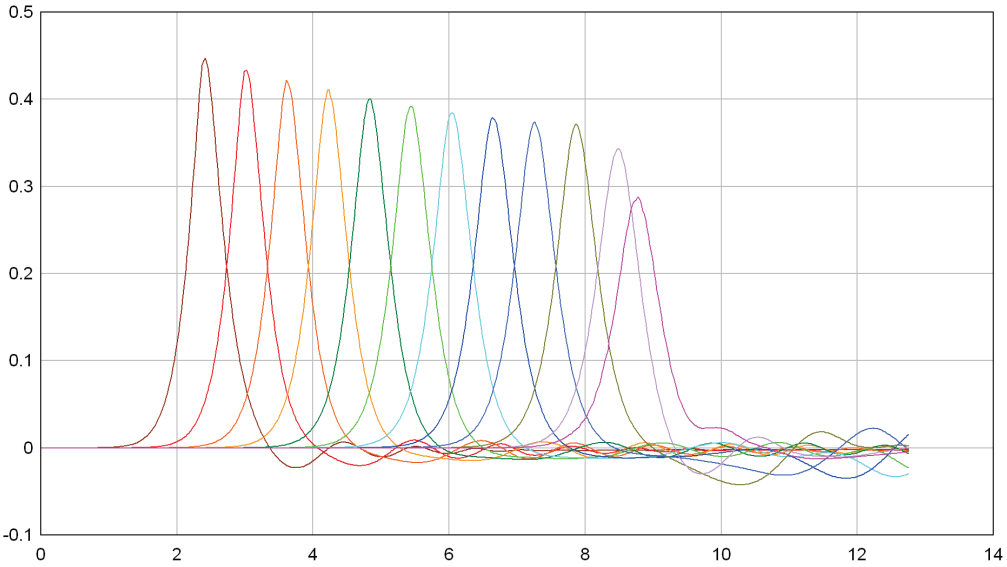


Fig. 10 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of solitary wave at wave gages along the length of the tank at y=0 for wave height=0.45 m; the wave absorption capability near the end of the tank is working well to absorb the incident wave (wave gage records are free from reflected waves)

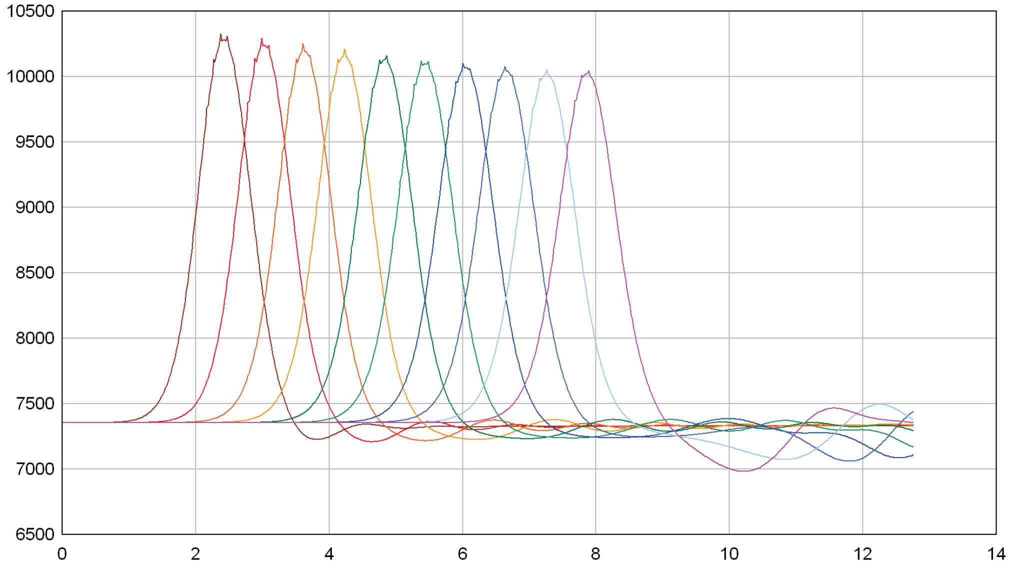


Fig. 11 Solitary wave pressure on the bottom (in N/m^2 on vertical axis) as a function of time (in sec on horizontal axis), along the length of the tank at y = 0 for H=0.45 m; baseline is static pressure = 0.75*9810=7357.5 N/m^2

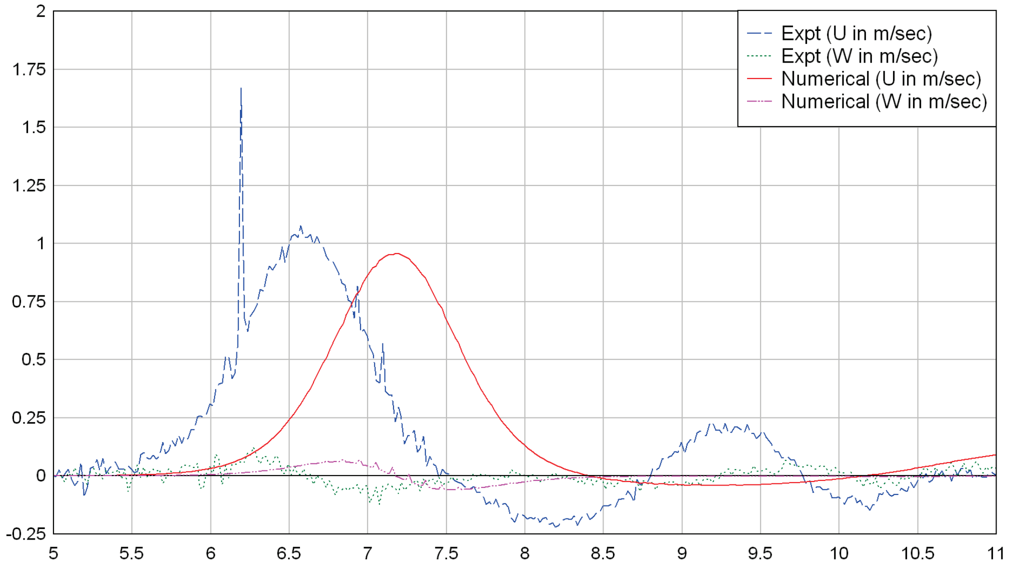


Fig. 12 Solitary wave particle velocity components (u, w) (in m/sec on vertical axis) as a function of time (in sec on horizontal axis) at x = 18.7 m, z = -0.61m, at y = 0 for wave height=0.45 m; numerical profiles are lagging behind experimental profiles, but there is a very good agreement of their magnitude and shape (the experimental velocity measurement is more noisy)

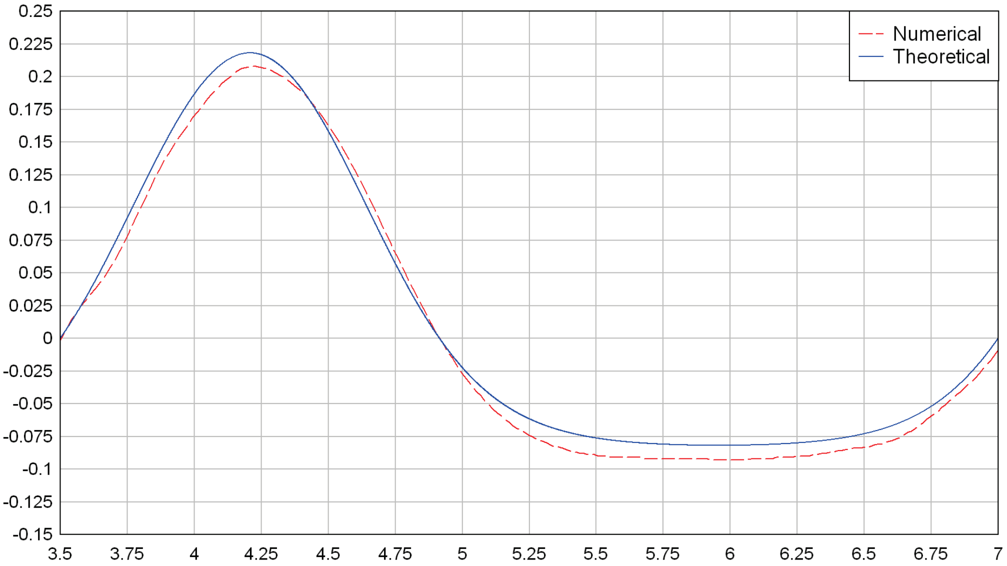


Fig. 13 Surface elevation at the wavemaker (in m on vertical axis) as a function of time (in sec on horizontal axis) for a Cnoidal wave of height H = 0.3 m and period T = 3.5 s, in depth h = 0.75 m.

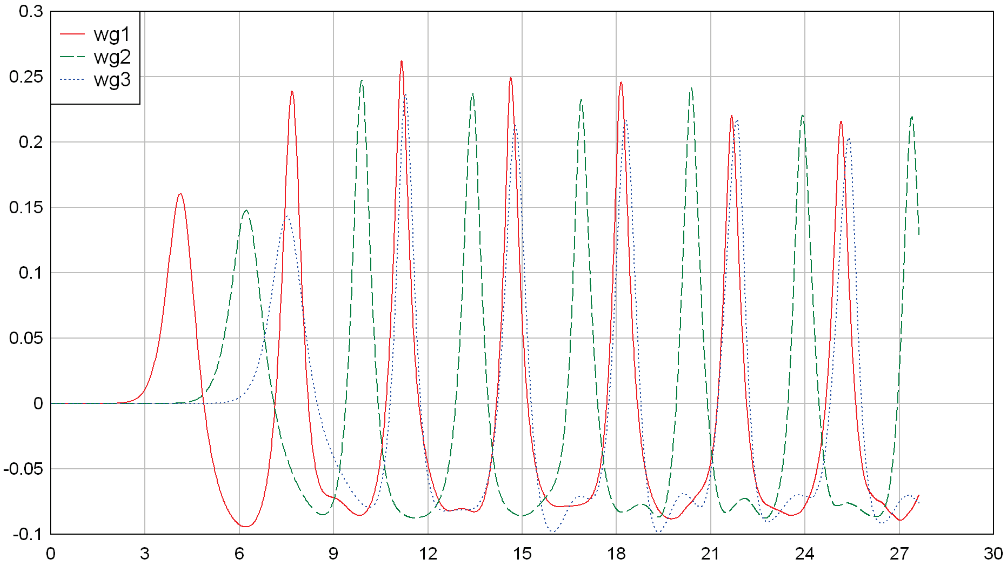


Fig. 14 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of Cnoidal waves at three wave gages at x=8.8 m, 14.9 m and 18.7 m with y=0, height H = 0.3 m and period T = 3.5 s.

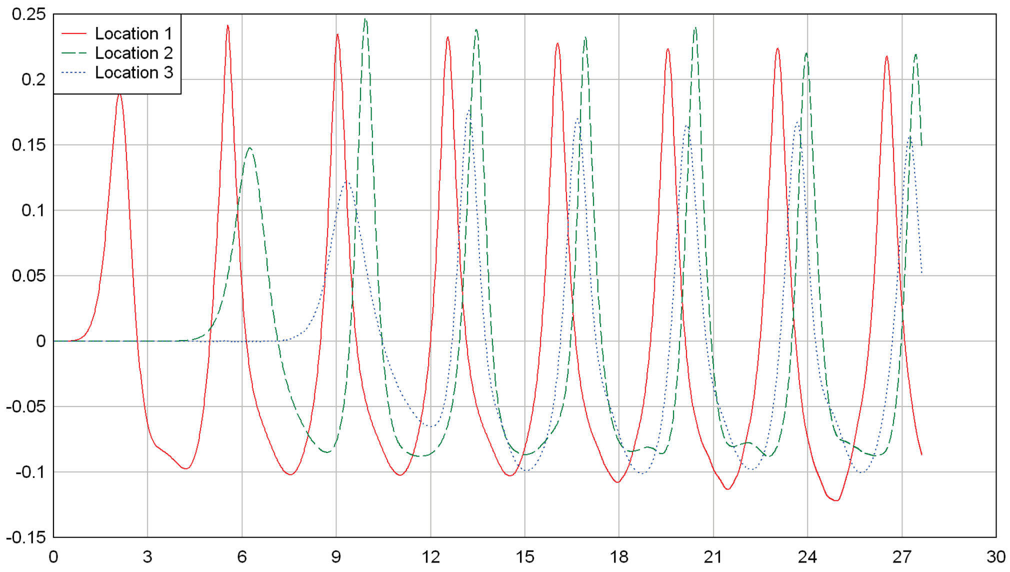


Fig. 15 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of Cnoidal waves at three wave gages at x=4m, 16m and 24.95m at y=0 for wave height=0.3 m; the wave absorption capability near the end of the tank is working well to absorb the waves (wave gage records are free from reflected waves)

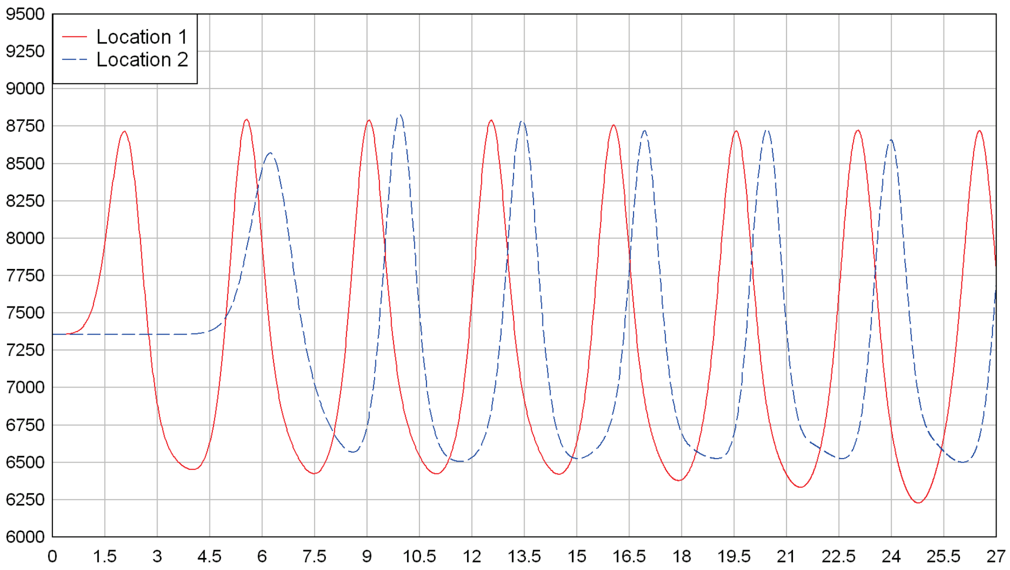


Fig. 16 Numerical bottom pressure (in N/m^2 on vertical axis) as a function of time (in sec on horizontal axis), for Cnoidal waves at two gages at x = 4 and 16 m with y = 0 and wave height=0.3 m; the baseline is the static pressure = 0.75*9810=7357.5 N/m^2.

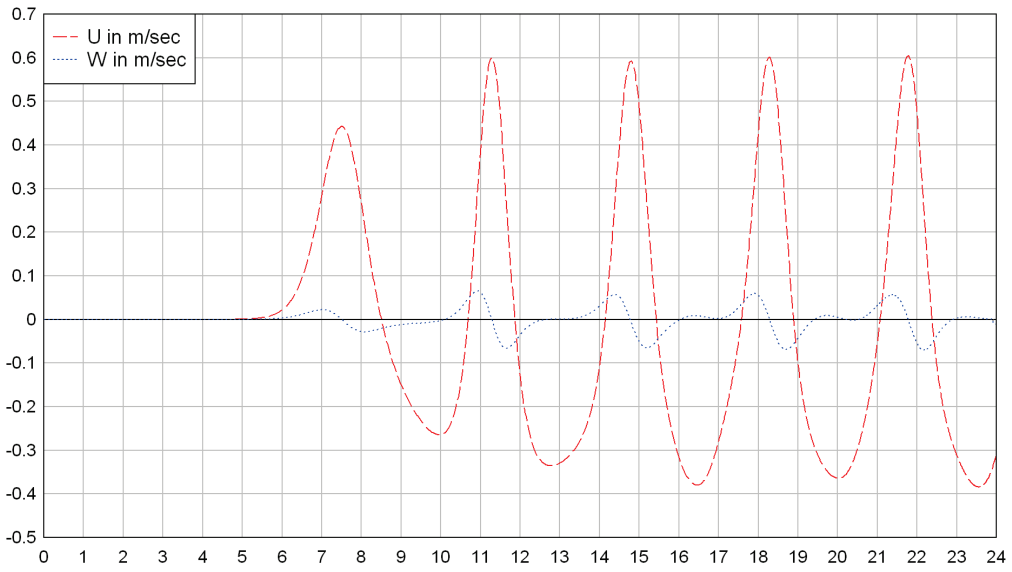


Fig. 17 Numerical Cnoidal wave particle velocity components (u, w) (in m/sec on vertical axis) as a function of time (in sec on horizontal axis) at x = 18.7 m, z = -0.61m, at y = 0 and for wave height = 0.3 m.

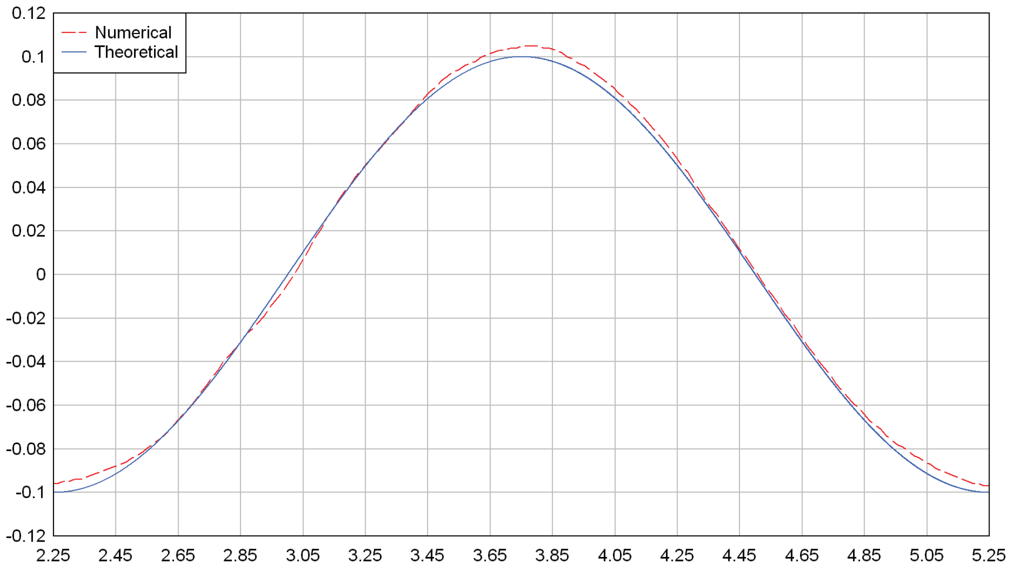


Fig. 18 Elevation of an Airy wave at wavemaker (in m on vertical axis) as a function of time (in sec on horizontal axis) for a wave height H = 0.2 m and period T = 3.0 s.

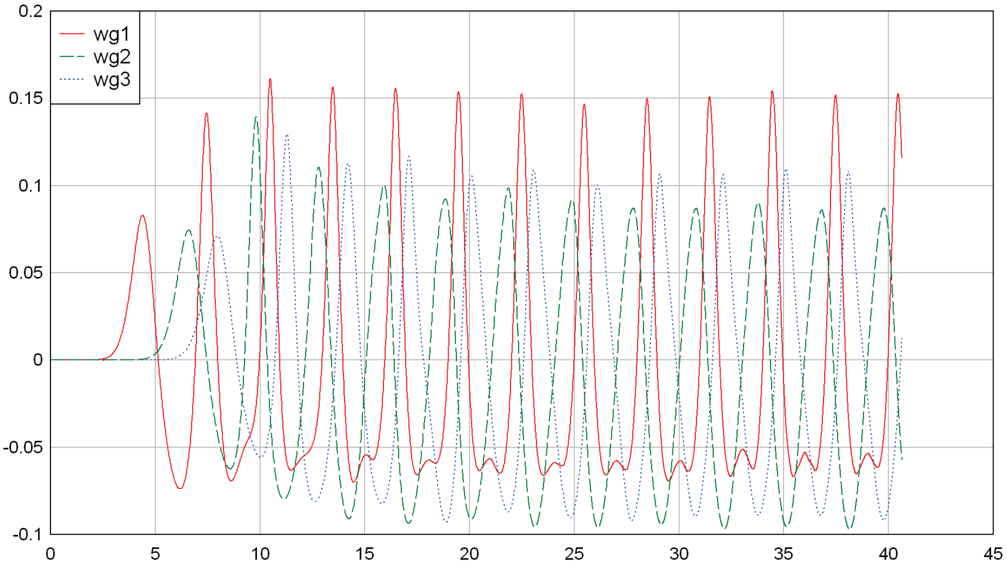


Fig. 19 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of Airy waves at three wave gages at x=8.8 m, 14.9 m and 18.7 m at y=0 for wave height = 0.2 m

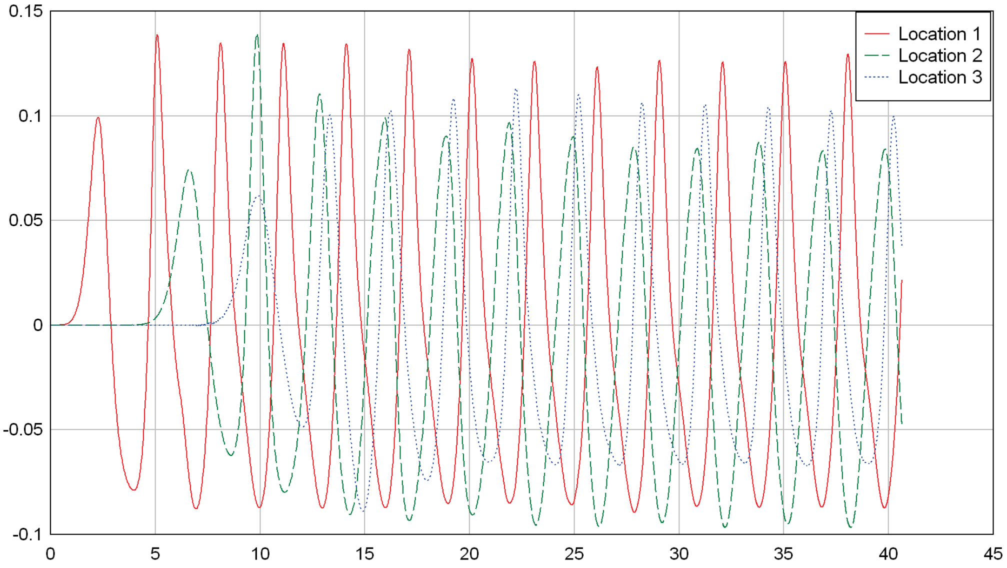


Fig. 20 Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of Airy waves at three wave gages at x=4m, 16m and 24.95m at y=0 for wave height=0.2 m; the wave absorption capability near the end of the tank is working well to absorb the waves (wave gage records are free from reflected waves)

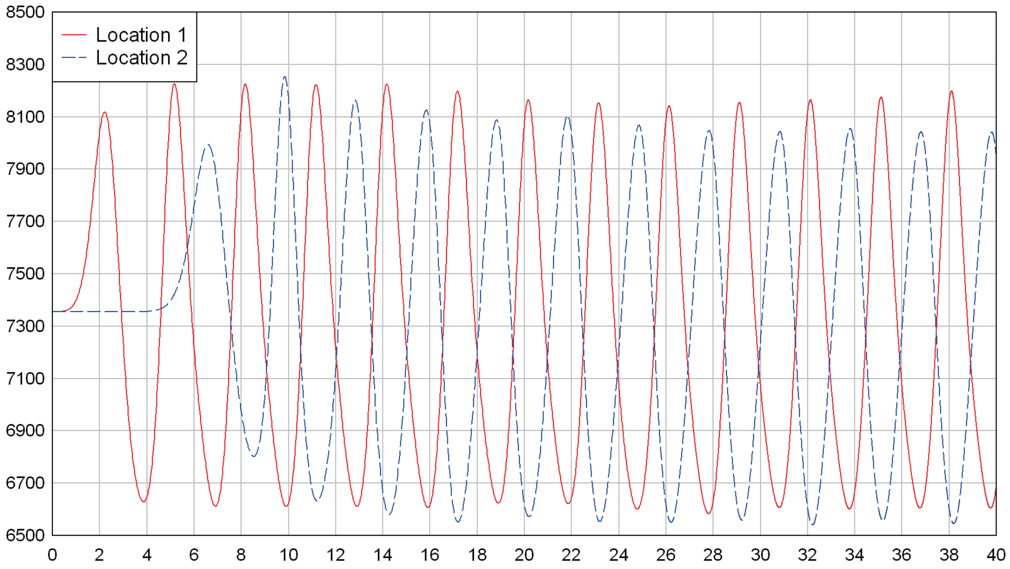


Fig. 21 Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of Airy waves at two pressure gages at x=4m and 16m located at the bottom surface with y=0 and wave height=0.2 m; static pressure =0.75*9810=7357.5 N/m^2

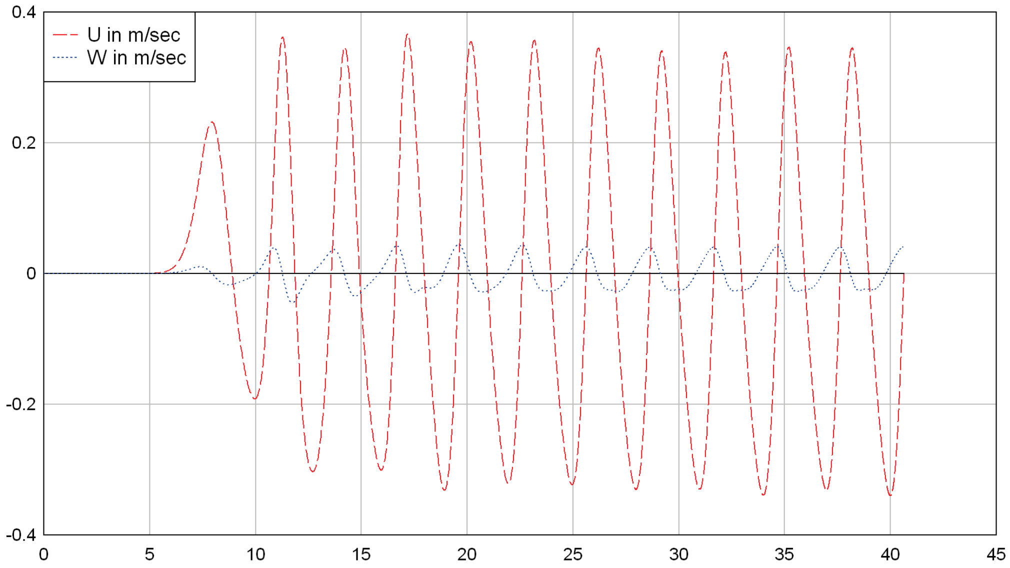


Fig. 22 Time (in sec on horizontal axis) versus numerical velocity components u,w (in m/sec on vertical axis) of Airy waves at x=18.7 m, z=-0.61m with y=0 and wave height=0.2 m

Table 1: Application-1: # of nodes N =2,592, # of tree levels in FMA (n-levels) = 6; 5 time steps

Number of processors (# CPU)	Total CPU time in sec	Speedup (T_1/Tp)
1	126.5	1.00
2	84.5	1.50
3	67	1.89
4	58	2.18
5	54.5	2.32
6	53	2.39
7	51	2.48
8	50.5	2.51

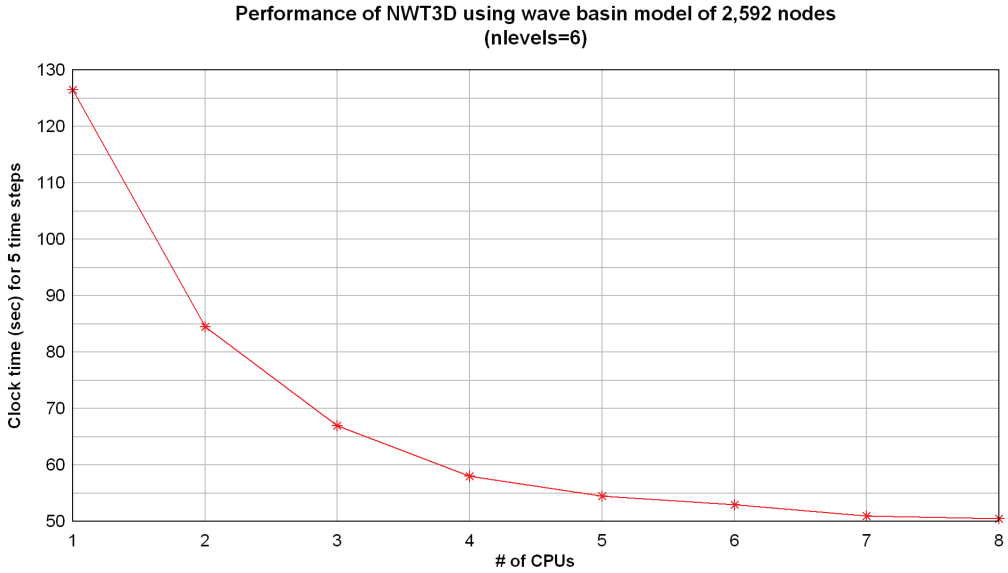


Fig. 23 Performance (clock time in sec versus # of CPU) for Application-1.

Table 2:- Application-2: # of nodes N =13,838, # of tree levels in FMA (n-levels) = 7; 5 time steps		
Number of processors (ncpu)	Execution time in sec	Speedup (T_1/Tp)
1	1102	1.00
2	751	1.47
3	609	1.81
4	514	2.14
5	511	2.16
6	480	2.30
7	456	2.47
8	444	2.48

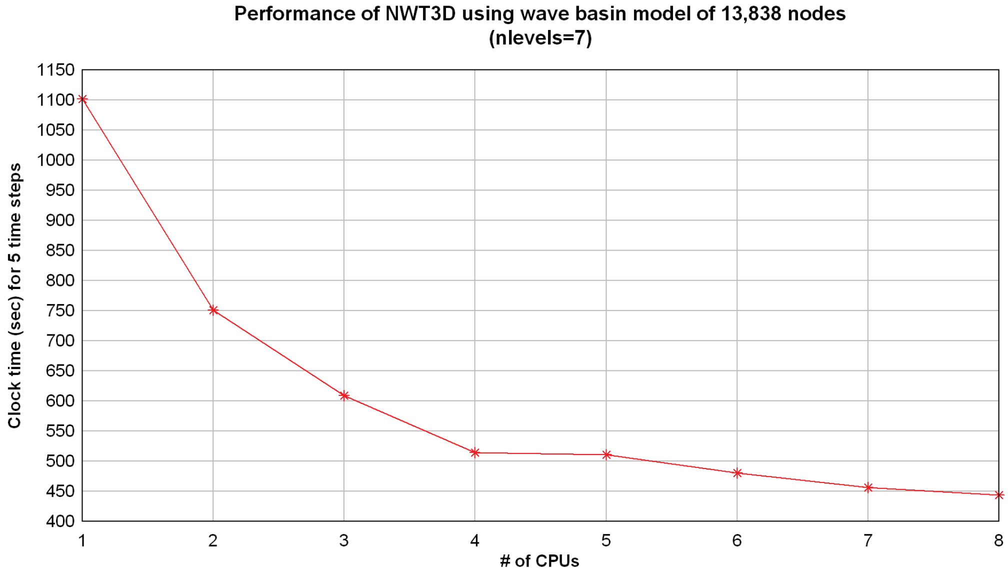


Fig. 24 Performance (clock time in sec versus # of CPU) for Application-2

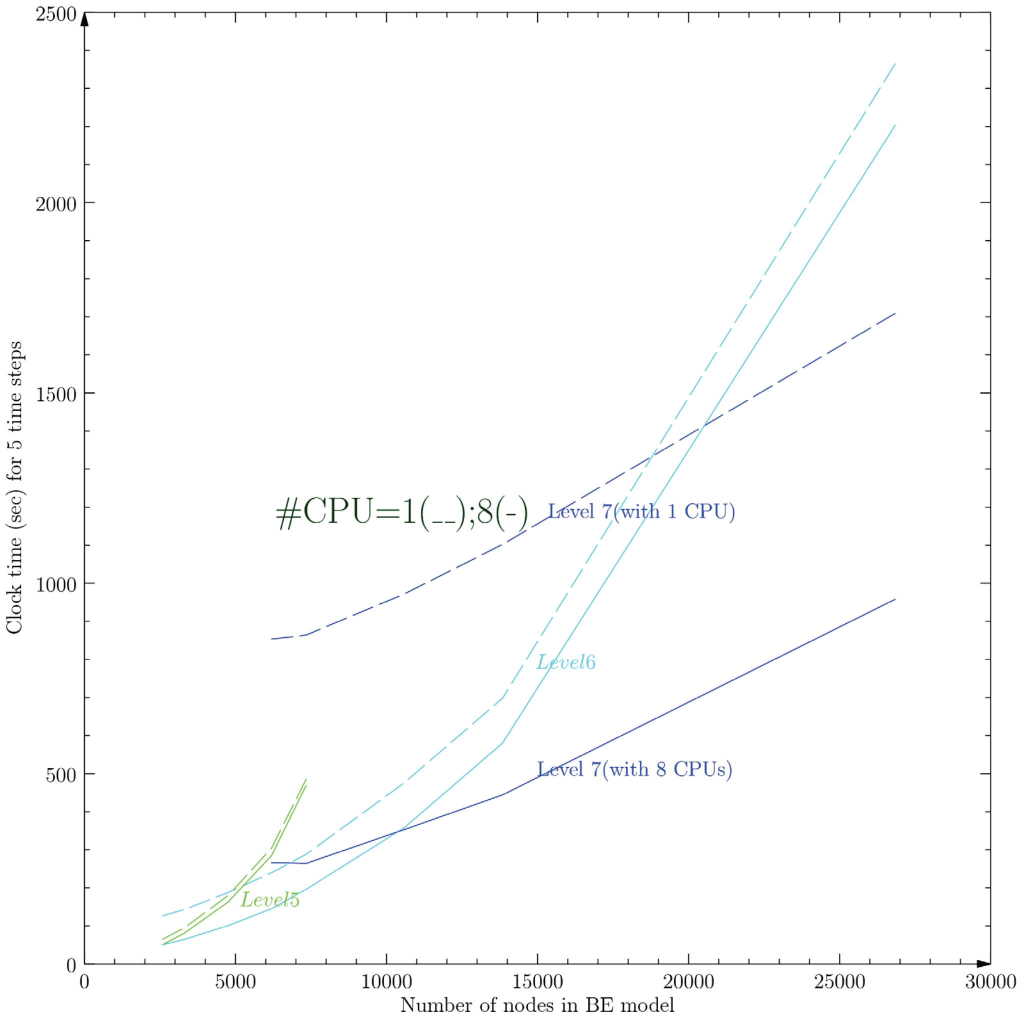


Fig. 25 Performance of the parallel 3D-NWT, tested using several applications with n-levels=5, 6, 7 in the FMA; solid line (-) is with 8 processors; the dashed line (--) with a single processor; there is nearly linear performance of run times with 8 processors (for instance, using solid lines comprising of the fastest clock times of levels 6 and 7)

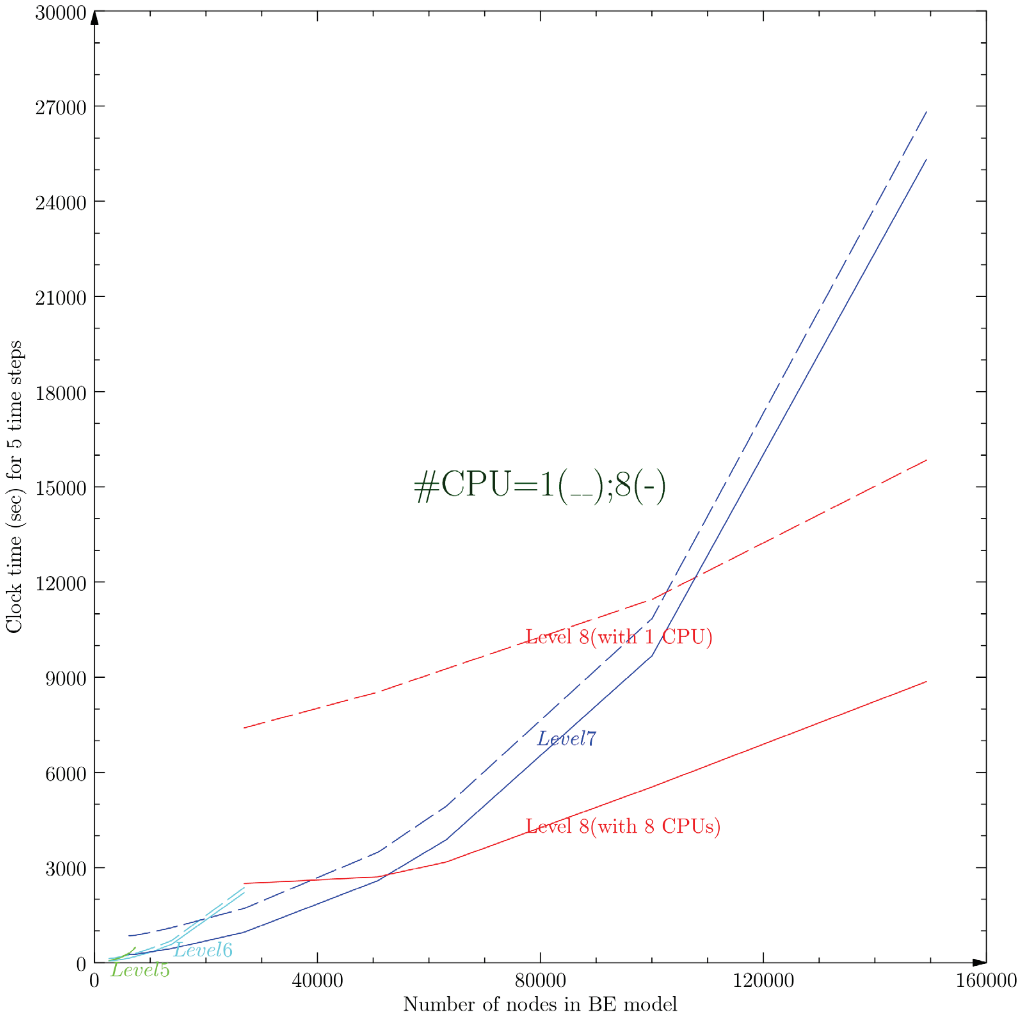


Fig. 26 Performance of the parallel 3D-NWT, tested using several applications with *n*-levels=5, 6, 7, 8 in the FMA; solid line (-) is with 8 processors; dashed line (--) with single processor; there is nearly linear performance of run times with 8 processors (for large enough *N* value).

Acknowledgement

The first two authors are thankful to the Office of Naval Research (ONR) for the generous financial support that made this work possible.

References

[1] Kanwal, R.P., 1997, "Linear integral equations- theory & technique," Birkhauser, Boston.

[2] Delves, L.M., and Mohamed, J.L., 2008, "Computational methods for integral equations," Cambridge University Press.

[3] Chen, G., and Zhou, J., 1992, "Boundary Element Methods," Academic Press.

[4] Hsiao, G.C., and Wendland, W.L., 2008, "Boundary integral equations," Springer-Verlag, Berlin, Heidelberg.

[5] Banerjee, P.K., and Butterfield, R., 1981, "Boundary element methods in engineering science," McGraw-Hill Book Co (UK) Ltd.

[6] Liggett, J.A., and Liu, P.L.F., 1983, "The boundary integral equation method for porous media flow," George Allen & Unwin Publishers, London.

[7] Brebbia, C.A., Telles, J.C.F., and Wrobel, L.C., 1984, "Boundary element techniques: theory and applications in engineering," Springer-Verlag, Berlin.

[8] Hartmann, F., 1989, "Introduction to Boundary elements," Springer-Verlag.

[9] Brebbia, C.A., and Dominguez, J., 1989, "Boundary elements: an introductory course," Computational Mechanics Publications, Southampton, Boston.

[10] Banerjee, P.K., 1994, "The boundary element methods in engineering," McGraw-Hill Book Company.

[11] Power, H., and Wrobel, L.C., 1995, "Boundary integral methods in fluid mechanics," Computational Mechanics Publications, Southampton, Boston.

[12] Wrobel, L.C., 2002, "The boundary element method- Applications in thermo-fluids and acoustics," Volume 1, John Wiley & Sons, Ltd.

[13] Aliabadi, M.H., 2002, "The boundary element method- Applications in solid and structures," Volume 2, John Wiley & Sons, Ltd.

[14] Bonnet, M., 1995, "Boundary integral equation methods for solids and fluids," John Wiley & Sons Ltd.

[15] Nishimura, N., 2002, "Fast multipole accelerated boundary integral equation methods," Applied Mechanics Review, 55(4), pp. 299-324.

[16] Ying, L., Biros, G., Zorin, D., and Langston, H., 2003, "A new parallel kernel-independent fast multipole method," Proceedings of the 2003 ACM/IEEE Conference on Supercomputing, 196(2), pp. 591-626.

[17] Ying, L., 2004, "An efficient and high-order accurate boundary integral solver for the Stokes equations in three dimensional complex geometries," PhD thesis, Department of Computer Science, New York University.

[18] Ying, L., Biros, G., and Zorin, D., 2004, "A kernel-independent adaptive fast multipole algorithm in two and three dimensions," Journal of Computational Physics, 196(2), pp. 591-626.

[19] Ying, L., Biros, G., and Zorin, D., 2006, "A high-order 3D boundary integral equation solver for elliptic PDEs in smooth domains," Journal of Computational Physics, 219(1), pp. 247-275.

[20] Margonari, M., and Bonnet, M., 2005, "Fast multipole method applied to elastostatic BEM-FEM coupling," Computers and Structures, 83, pp. 700-717.

[21] Chaillat, S., Bonnet, M., and Semblat, J-F., 2008, "A multi-level fast multipole BEM for 3-D elastodynamics in the frequency domain," Computer Methods in Applied Mechanics and Engineering, 197(49-50), pp. 4233-4249.

[22] Liu, Y., 2009, "Fast multipole boundary element method: theory and applications in engineering," Cambridge University Press.

[23] Sternberg,W.J., and Smith, T.L., 1946, "The theory of potential and spherical harmonics," The University of Toronto Press.

[24] Kellogg, O.D., 1953, "Foundations of Potential theory," Dover Publications Inc.

[25] Dommermuth, D.G., Yue, D.K.P., Lin, W.M., and Rapp, R.J., 1988, "Deep-water plunging breakers: a comparison between potential theory and experiments," Journal of Fluid Mechanics, 189, pp. 423-442.

[26] Dold, J.W., and Peregrine, D.H., 1986, "An efficient boundary-integral method for steep unsteady water waves," Numerical Methods for fluid dynamics II, Edited by Morton, K.W., and Baines, M.J., Clarendon Press, Oxford.

[27] Dold, J.W., 1992, "An efficient surface-integral algorithm applied to unsteady gravity waves," Journal of Computational Physics, 103, pp. 90-115.

[28] Dold, J.W., and Peregrine, D.H., 1984, "Steep unsteady water waves: An efficient computational scheme," Proceedings of 19th International Conference on Coastal Engineering, Houston, pp. 955-967.

[29] Lin, W.M., 1984, "Nonlinear motion of the free surface near a moving body," PhD thesis, Department of Ocean Engineering, Massachusetts Institute of Technology.

[30] Grilli, S., Subramanya, R., Svendsen, I.A. and Veeramony, J., 1994, “Shoaling of Solitary Waves on Plane Beaches,” Journal of Waterway, Port, Coastal and Ocean Engineering, 120(6), pp. 609-628.

[31] Grilli, S.T., Svendsen, I.A. and Subramanya, R., 1997, “Breaking Criterion and Characteristics for Solitary Waves on Slopes,” Journal of Waterway, Port, Coastal and Ocean Engineering, 123(3), pp. 102-112.

[32] Longuet-Higgins, M.S. and Cokelet, E.D., 1976, “The deformation of steep surface waves on water. I. A numerical method of computation,” Proceedings of the Royal Society, London,, A 350, 1-26.

[33] Grilli, S.T., and Horrillo, J., 1997, “Numerical Generation and Absorption of Fully Nonlinear Periodic Waves,” Journal of Engineering Mechanics, 123 (10), pp. 1060-1069.

[34] Grilli, S.T., and Horrillo, J., 1999, “Shoaling of periodic waves over barred-beaches in a fully nonlinear numerical wave tank,” Intl. J. Offshore and Polar Engng, 9(4), pp. 257-263.

[35] Romate, J.E., 1989, “The numerical simulation of nonlinear gravity waves in three dimensions using a higher order panel method,” Zandbergen, P.J., (advisor), PhD thesis, University of Twente, Netherlands.

[36] Broeze, J., 1993, “Numerical modelling of nonlinear free surface waves with a 3D panel method,” Zandbergen, P.J., (advisor), PhD thesis, University of Twente, Netherlands.

[37] Daalen van, E.F.G., 1993, “Numerical and theoretical studies of water waves and floating bodies,” Zandbergen, P.J., (advisor), PhD thesis, University of Twente, Netherlands.

[38] Berkvens, P.J.F., 1998, “Floating bodies interacting with water waves- development of a time-domain panel method,” Zandbergen, P.J., (advisor), PhD thesis, University of Twente, Netherlands.

[39] Liu, P.L.F., and Hsu, H.W., 1992, “Applications of boundary integral equation methods for two-dimensional non-linear water wave problems,” International Journal for numerical methods in fluids, 15, pp. 1119-1141.

[40] Cooker, M.J., 1990, “A boundary-integral method for water wave motion over irregular beds,” Engineering Analysis with Boundary Elements, 7(4), pp. 205-213.

[41] Vinayan, V., and Kinnas, S.A., 2007, “A BEM for the propagation of Nonlinear Planar Free-surface Waves,” Electronic Journal of Boundary Elements, 5(1), pp. 17-40.

[42] Colicchio, G., Greco, M., and Faltinsen, O.M., 2006, “A BEM-level set domain-decomposition strategy for non-linear and fragmented interfacial flows,” International Journal for Numerical Methods in Engineering, 67, pp. 1385-1419.

[43] Grilli, S.T., Guyenne, P., and Dias, F., 2001, "A fully nonlinear model for three-dimensional overturning waves over arbitrary bottom," International Journal for Numerical Methods in Fluids, **35**(7), pp. 829-867.

[44] Brandini, C., and Grilli, S.T., 2001, "Modeling of freak wave generation in a 3D-NWT," In Proc. 11th Offshore and Polar Engng. Conf. (ISOPE01, Stavanger, Norway, June 2001), Vol III, pp. 124-131.

[45] Grilli, S.T., Vogelmann, S., and Watts, P., 2002, "Development of a 3D numerical wave tank for modeling tsunami generation by underwater landslides," Engineering Analysis with Boundary Elements, **26**(4), pp. 301-313.

[46] Fochesato, C., Grilli, S.T., and Guyenne, P., 2005, "Note on non-orthogonality of local curvilinear coordinates in a three-dimensional boundary element method," International Journal for Numerical Methods in Fluids, **48**, pp. 305-324.

[47] Fochesato, C., and Dias, F., 2006, "A fast method for nonlinear three-dimensional free-surface waves," Proceedings of the Royal Society, London, **A 462**, pp. 2715–2735.

[48] Guyenne, P., and Grilli, S.T., 2006, "Numerical study of three-dimensional overturning waves in shallow water," Journal of Fluid Mechanics, **547**, pp. 361-388.

[49] Fochesato, C., Grilli, S.T., and Dias F., 2007, "Numerical modeling of extreme rogue waves generated by directional energy focusing," Wave Motion, **44**, pp. 395-416.

[50] Sung, H.G., and Grilli, S.T., 2008, "BEM Computations of 3D Fully Nonlinear Free Surface Flows Caused by Advancing Surface Disturbances," International Journal of Offshore and Polar Engineering, **18**(4), pp. 292-301.

[51] Grilli, S.T., Dias, F., Guyenne, P., Fochesato, C., and F. Enet, 2010, "Progress In Fully Nonlinear Potential Flow Modeling Of 3D Extreme Ocean Waves," Chapter 3 in Advances in Numerical Simulation of Nonlinear Water Waves ,Vol. 11 in Series in Advances in Coastal and Ocean Engineering, World Scientific Publishing Co. Pte. Ltd., pp. 75-128.

[52] Grilli, S.T., Skourup, J., and Svendsen, I.A., 1989, "An efficient boundary element method for nonlinear water waves," Engineering Analysis with Boundary Elements, 6(2), pp. 97-107.

[53] Grilli, S.T., and Subramanya, R., 1996, "Numerical modeling of wave breaking induced by fixed or moving boundaries," Computational Mechanics, 17, pp. 374-391.

[54] Grilli, S.T., 1993, "Modeling of nonlinear wave motion in shallow water," Computational methods for free and moving boundary problems in heat and fluid flow, Wrobel, L.C., and Brebbia, C.A., (editors), Computational Mechanics Publications, Southampton, Boston.

[55] Grilli, S., and Svendsen, I.A., 1990, “Corner problems and global accuracy in the boundary element solution of nonlinear wave flows,” Engineering Analysis with Boundary Elements, 7(4), pp. 178-195.

[56] Grilli, S.T., Voropayev, S., Testik, F.Y., and Fernando, H.J.S., 2003, “Numerical Modeling and Experiments of Wave Shoaling over Buried Cylinders in Sandy Bottom,” In Proc. 13th Offshore and Polar Engineering Conference (ISOPE03, Honolulu, USA, May 2003), pp. 405-412.

[57] Svendsen, I.A., Otta, A.K., and Grilli, S.T., 1992, “Unsteady free surface waves,” Breaking waves, International Union of Theoretical and Applied Mechanics (IUTAM) Symposium, Sydney, Australia, 1991, Banner, M.L., and Grimshaw, R.H.J., (eds), Springer-Verlag.

[58] Grilli, S.T., 1997, “Fully nonlinear potential flow models used for long wave runup prediction,” Long-Wave Runup Models (eds. H. Yeh, P. Liu, and C. Synolakis), World Scientific Publishing, Singapore, pp.116-180.

[59] Grilli, S.T., Harris, J., and N. Greene, 2009, “Modeling of Wave-induced Sediment Transport around Obstacles,” In Proc. 31st Intl. Coastal Engng. Conf. (J. McKee Smith, ed.) (ICCE08, Hamburg, Germany, September, 2008), World Scientific Publishing Co. Pte. Ltd, pp.1,638-1,650.

[60] John O. Hallquist, 2006, “LS-DYNA theory manual,” Livermore Software Technology Corporation.

[61] Steven A. Hughes, 1993, “Physical models and laboratory techniques in coastal engineering,” World Scientific Publishing.

[62] Derek G. Goring, 1978, “Tsunamis-The propagation of long waves onto a shelf, W.M.Keck Laboratory of Hydraulics and Water Resources,” Division of Engineering and Applied Science, California Institute of Technology, Pasadena, California, Report no. KH-R-38.

[63] Department of the Army, 1984, “Shore Protection Manual, Volume I,” Waterways Experiment Station, Corps of Engineers, Coastal Engineering Research Center.

[64] Chris Lakhan, 1982, “Calculating modulus k of Cnoidal waves, J. Waterway, Port, Coastal and Ocean Engineering,” Vol 108, No. 3, pp.435-438.

[65] Abramowitz, M., and Stegun, I.A., 1972, “Handbook of Mathematical functions,” Dover Publications, New York.

[66] Robert G. Dean, Robert A. Dalrymple, 2000, “Water wave mechanics for engineers and scientists,” World Scientific Publishing.

[67] “Tsunami Wave Basin (TWB) Experiment Notebook,” College of Engineering, NACSE, Oregon State University: Tsunami wave impact forces on cylinders (Phase1Exp2: run5, run6, run11, run17 and run18), http://neespop.wave.oregonstate.edu/nees/ui/index.php?maintab=200&tab=0&proID=21&expID=72

[68] Rankin, W.T., 1999, "Efficient Parallel Implementations of Multipole based N-Body algorithms,” PhD Thesis, Dept. of Electrical Engineering, Duke University.

[69] Chapman, B., et al., 2007, “Using OpenMP: Portable shared memory parallel programming,” MIT Press.

[70] John D. Fenton, 1985, “A fifth-order Stokes theory for steady waves,” J. Waterway, Port, Coastal and Ocean Engineering, Vol 111, Issue 2, pp.216-234.

[71] Kiyoshi Horikawa (editor), 1988, “Nearshore dynamics and coastal processes,” University of Tokyo Press.

Comparison and Validation of Nonlinear Potential Flow (BEM) and Navier-Stokes (FEM) Free Surface Flow Models Using a 3D Wave Basin Experiment

Seshu B. Nimmala and Solomon C. Yim
School of Civil & Construction Engineering, Oregon State University (OSU),
Corvallis, OR 97331, USA

Abstract

It has been of considerable interest to the engineers and scientists working in the areas of hydrodynamics and fluid dynamics to study how well the numerical predictions from the potential flow model (which is widely used in hydrodynamics) and the general viscous flow model (which is widely used in computational fluid dynamics or CFD) compare with each other in the case of nonlinear free surface flows. In this context, this paper presents the numerical results of the fully nonlinear potential flow (FNPF) using the Laplace equation and general viscous flow using the Navier-Stokes (NS) equation in the backdrop of the comparison with a large-scale 3D wave basin experiment (i.e., benchmark). The FNPF model is implemented using the boundary element method (BEM) and the NS model is implemented using the finite element method (FEM). The authors have recently incorporated several enhancements into the BEM code (3D-NWT) which include generation of waves such as solitary, Cnoidal and Airy, etc. using a piston wavemaker and parallelized it to solve large-scale problems on high-performance computing (HPC) platforms. The three-way comparison (between two numerical methods and an experiment) evaluated the relative merits and provided an understanding of the strengths and

weaknesses of two popular models (FNPF and NS) and numerical methods (BEM and FEM) and in the process validated some of the capabilities of the 3D-NWT code. It is also for the first time that the 3D-NWT code with the newly incorporated developments is run to simulate a large-scale problem of the 3D wave basin with a complex bathymetry.

The comparison of results includes that of the development of free surface, and velocities at salient locations in both quantitative and qualitative (visual) forms. The results indicate that the potential flow model can be successfully employed in many situations where a Navier-Stokes model is computationally demanding for attaining the same accuracy and 3D-NWT is very appropriate for the free surface flows due to the advantages of surface-only mesh creation for 3D models. Some of the results for performance of the parallelized 3D-NWT code, including that of the benchmark problem, are also presented. Although the FE model ran faster for the same model size, the total turnaround time (including model preparation) is significantly shorter for the BE model and the latter has been more accurate and offered convenient input (i.e., with fewer parameters) for the user.

1 Introduction

This paper first presents the background information, then describes a large-scale laboratory experiment, simulations using FNPF-BEM and NS-FEM models, comparison of results, challenges encountered, and insight and experience gained in the process. Results of parallel performance are also presented in one of the sections. The experiment, herewith referred to as the benchmark problem, is conducted in the 3D wave basin at Oregon State University (OSU). The stage is set with the propagation of a solitary wave over a complex bathymetry.

2 Background

The term computational fluid dynamics (CFD) is more commonly used in the disciplines of mechanical and aerospace engineering whereas hydrodynamics is generally used in the field of civil engineering (or its sub-area ocean engineering). The former term broadly describes the numerical techniques for solving mathematical models of fluids and this effort often includes Navier-Stokes equations (for describing both incompressible and compressible flows) and more often than not these are internal flows (i.e., without requiring a free surface). The latter field uses models based on potential flow along with wavemakers and free surfaces flows. One of the goals of this paper is to understand the relative merits of either model with practical problems in consideration. The following subsections introduce the reader to both the codes that implemented BEM and FEM.

2.1 The BE model: 3D-NWT is a special purpose research code that implemented the boundary element method (BEM) to solve the FNPF. References [1-10] can be perused for a detailed study of the BEM. The BEM offers an advantage that the 3D problems can be solved using a surface-only mesh. This method however results in unsymmetric and fully populated linear system matrices requiring faster solution methods. To this end, the fast multipole method (FMM) or fast multipole algorithm (FMA) is an important development that brought the BEM to the forefront in the world of numerical methods [11]. References [12-15] include the latest developments in the fast solution methods using both the FMA and the BEM whereas [16] presents practical engineering applications using these technologies in an accessible manner for students and researchers. References [17] and [18] contain mathematical analysis of the BEM.

Nonlinearity of FNPF occurs from the presence of the free surface and its boundary conditions. Grilli, et al [19-34] have developed an accurate and versatile three-dimensional numerical wave tank based on the FNPF, and higher-order boundary element methods, combining both the free-space Green's functions and the fast multipole algorithm in combination with generalized minimal residual method (GMRES). The numerical complexity is proportional to the problem size. The free-surface time updating is based on a second-order Taylor series expansion in an Eulerian-Lagrangian form.

The authors made several enhancements to the numerical wave tank and named the new implementation as the 3D-NWT [35]. Some of the major contributions include the incorporation of a piston wavemaker, wave generation capabilities (including Solitary wave generation), visualization and parallelization. Some of the mathematical details of the 3D-NWT are summarized below:

The Laplace equation is given by

$$\nabla^2\phi = 0 \tag{1}$$

where $\phi =$ velocity potential function. Also, velocity $\vec{u} = \nabla\phi$.

If $\vec{R}(t)$ is the position vector (x,y,z) of a fluid particle on the free surface, g is the acceleration due to gravity, $\vec{u}$ is the velocity vector (u,v,w),

$$\frac{D\vec{R}}{Dt} = \vec{u} = \nabla\phi \tag{2}$$

(from kinematic boundary condition)

$$\frac{D\phi}{Dt} = -gz + \frac{1}{2}\nabla\phi.\nabla\phi \tag{3}$$

(from dynamic boundary condition)

For fixed boundaries, the no flow condition (zero flux) is specified as $\frac{\partial\phi}{\partial n} = 0$, where n=normal vector to the surface.

In the case of moving boundaries, the wavemaker generates waves with motion and velocity that are specified as BCs:

$$x = x_p; \quad \frac{\partial\phi}{\partial n} = \vec{u}_p.\vec{n}, \tag{4}$$

where $\vec{n}$ is the normal to the wavemaker.

Second-order explicit Taylor series expansions are used to find the new position and potential on free surface at each new time step.

$$\vec{R}(t+\Delta t)=\vec{R}(t)+\Delta t\frac{D\vec{R}(t)}{Dt}+\frac{(\Delta t)^2}{2}\frac{D^2\vec{R}(t)}{Dt^2}+O\left[(\Delta t)^3\right] \tag{5}$$

$$\phi(\vec{R}(t+\Delta t))=\phi(t)+\Delta t\frac{D\phi(t)}{Dt}+\frac{(\Delta t)^2}{2}\frac{D^2\phi(t)}{Dt^2}+O\left[(\Delta t)^3\right] \tag{6}$$

The boundary conditions for the three-dimensional piston wavemaker with the boards/paddles driven by the actuators are derived by the authors [35].

A solitary wave is generated using the following mathematical formulations. Further details can be obtained from Refs. [36] and [37].

Let H = wave height and h = water depth

Total stroke, $S_s=\sqrt{\frac{16Hh}{3}}$; Celerity, $C=\sqrt{g(h+H)}$

Wave number,

$$k=\sqrt{\frac{3H}{4h^3}} \tag{7}$$

Waveboard time history,

$$X_o(t)=\frac{H}{kh}\tanh\left[k(Ct-X_o)\right] \tag{8}$$

Waveboard displacement,

$$X_o(at\,t=\pm\infty)=\pm\frac{H}{kh}=\pm\sqrt{\frac{4Hh}{3}} \tag{9}$$

Let $X_l=0.999\frac{H}{kh}$.

Time taken by board to travel (from $X_o = 0$ to $+X_l$)=

$$t_f = \frac{1}{kC}(3.8 + \frac{H}{h}) \qquad (10)$$

Total stroke duration, $t_s = 2t_f$

Waveboard starts at an initial resting position $X_o(-t_f) = -\sqrt{\frac{4Hh}{3}}$ and a final resting position $X_o(+t_f) = +\sqrt{\frac{4Hh}{3}}$ for practical purposes.

2.2 The FE model: For finite-element analysis, we use a version of the general-purpose code LS-DYNA ([38] and [39]) with capabilities of nonlinear mechanics – namely, impact and contact algorithms along with several advanced constitutive models. This code is selected based on its capabilities to perform fluid-structure interaction, flexibility of its license for academic/research use and its availability on most HPC platforms (in research as well as production-oriented environments). The version used for this study works on distributed memory platforms (such as clusters). The fluid solver available in this code is a compressible flow solver with Arbitrary-Lagrangian Eulerian (ALE) formulation. As of this writing, we are developing an incompressible Reynolds averaged Navier-Stokes (RANS) solver based on particle finite element method (PFEM) which will soon be included in an upcoming release.

3 Description of the benchmark experiment

The 3D wave basin at OSU has dimensions of 48.8 m length, 26.5 m width and 2.1 m depth. The piston-type wavemaker spanning across the width has 29 boards and a height of 2 m. This wavemaker has the capability to generate regular, irregular waves, user-defined waves, etc. with a period range of 0.5 to 10 sec and maximum wave height of 0.8 m in 1 m depth. Maximum stroke can be 2.1 m whereas the maximum velocity is 2 m/s. As part of the ISEC community workshop [40] held in summer 2009 at OSU, the participants with expertise in the numerical modeling of ocean engineering phenomena were asked to compare results from their software codes with a laboratory experiment (referred to as the benchmark). This experiment involved the propagation of a solitary wave on complex bathymetry with a triangular shaped sloping shelf. The foot of the shelf starts at 10.2 m from the wavemaker. The water depth near the wavemaker is 0.78 m, whereas the solitary wave height at the wavemaker is 0.39 m. The bathymetry is made available to the modelers in the form of a large number of closely spaced points (i.e., coordinates X, Y and Z). Views (3D and cross-sectional) of the bathymetry are in Figures 1 and 2 (taken from [40], [41]). A 3D view of the plotted bathymetry is shown in Figure 3 and a plan view is shown in Figure 4 (taken from [40]). A grid corresponding to the bathymetry is generated for both the BE and FE models and is illustrated in Fig. 5. More detailed information about the benchmark problem and the results can be found in references [40] and [41]. Snapshots of the wave propagation and its breaking in the 3D experimental wave basin at OSU are in Figs. 6-10.

4 Input motion to the wavemaker

Figure 11 illustrates the displacement, velocity and acceleration of the wavemaker required to generate the solitary wave. The maximum motion of the wavemaker is 1.27 m in a duration of 3.2 sec.

5 Simulation using the FNPF-BEM

In the simulation done using the 3D-NWT code, the actual length (48.8 m) of the tank is reduced to 25.75 m length to match with the SWS (still water shoreline) as specified in the benchmark so as to focus on the area of interest before wave breaking happens. This strategy also facilitates finer discretization in a model of smaller dimensions and faster execution. Width of the tank is 26.5 m and the depth of still water is 0.78 m above datum. Motion of the piston wavemaker is created using a software module named "Wavegen", which is developed by the authors based on the wave generation algorithms used in the OSU's 3D wave basin for such purposes. Celerity of the wave is computed to be 3.42 m/sec.
Since the bathymetry (defined as a large set of points) is not directly adaptable to automatic mesh generation available in 3D-NWT, a Matlab script is developed to produce a grid (Fig. 5) of the bottom surface using cubic interpolation. A special option is incorporated in 3D-NWT to be able to accept this bottom grid.

The BEM mesh (which is a surface mesh of the tank with the above-mentioned dimensions) has 303 elements along the length (X direction), 58 elements along the width (Y direction) and 5 elements along the depth (Z direction). The mesh has a total of 40,228 nodes and 38,758 elements. Figure 12 shows some of the mesh details and a solitary wave. The wall clock time of the execution is about 32 hours using 8 CPUs on a test bed machine using shared memory for a solution time of 5.38 sec with a total of 375 time steps. This is the first time a FNPF model of a real-world problem of this size has been run using the 3D-NWT code on multiple CPUs to result in a "reasonably" acceptable clock time.

6 Simulation using the NS-FEM

The wavemaker is explicitly modeled and prescribed with a time-history of motion (mentioned in section 4) to interact with the fluid (water) domain. In addition to the fluid part, the air part is also required to be modeled to provide the interface to compute the propagation of free surface. The air part is situated at the top of the water part as well as behind the wavemaker.

In order to compare with the BEM model, the number of elements in this model is kept to be the same as in the BEM mesh along the X, Y and Z directions to form the domain mesh (unlike the surface mesh in BEM). Thus, the fluid part has 303 elements along X, 58 along Y and 5 along Z. The mesh has a total of 205,733 nodes and 182,700 elements. Figure 13 illustrates the FE mesh used for the run.

Some details of the parameters used in model are provided below. Further details on the keywords can be referenced from [39].

An equation of state (EOS) defines the thermodynamic state of a homogeneous material without undergoing any chemical reactions or phase changes and is generally a function of two state variables [38]. The EOS to represent the behavior of the water is input as the linear polynomial type with user-defined constants C_0, C_1, etc.

Pressure,

$$P = C_0 + C_1\mu + C_2\mu^2 + C_3\mu^3 + (C_4 + C_5\mu + C_6\mu^2) * E \qquad (11)$$

where $\mu = \frac{\rho}{\rho_0} - 1 = \frac{V_0 - V}{V} = -\frac{\delta V}{V}$

ρ = current density

ρ_0 = nominal or reference density

E = internal energy per initial volume

V = current volume

V_0 = reference volume

Fluid properties of water are defined by the bulk modulus which gives the relation between the change of volume and pressure. For water this relation can be assumed to be linear: $P = C_1 * \mu$, where C_1=K=bulk modulus of water=2300e+6 N/m^2 and the remaining coefficients C0, C_2, C_3, C_4, C_5, C_6 are specified to be zero in the above EOS.

The wavemaker is represented as a rigid body and air is treated as a void to simplify the calculations as its influence is very minimal. The wavemaker's coupling with water is specified and the run took 42 hours for the 9 sec solution time on 8 processors.

The compressible flow behavior can be made to simulate more closely an incompressible flow by reducing the sound speed to the order of about 10 times the celerity of the wave (this factor came into light from the discussions held by the second author with some of the CFD experts) which in effect causes a reduction in bulk modulus, thereby resulting in a faster execution time (since the time step becomes larger). The problem in question involves the gravity waves which travel slower than the sound, thus the compressibility effects are negligible.

Celerity of wave =3.42 m/s

So, sound speed in water is taken to be 34.5 m/s

Therefore, K = C_1 = Speed of sound^2 * mass density = 34.5^2*1000 ~1,190,000N/m^2. Acceleration due to gravity =9.81 m/s^2 (used in linear load curve of gravity applied during first 0.5 sec of the execution).

The modified (i.e., with reduced C_1) run took only 3 hrs on 4 processors for a solution time of 9 sec, which is a great improvement in speed. The propagation of the wave has been observed to be smoother when compared with the previous run.

7 Comparison of results of the 3D-NWT with the benchmark

There is a very good agreement in the free surface elevation computed by this model and the benchmark experiment. The location (spatial and temporal) of the beginning of the breaking of the wave is also observed to be very close among the results of the computation and experiment. The numerical simulation could not progress beyond the onset of the breaking wave due to mesh distortion and the limitations inherent in the potential theory.

Figure 14 has snapshots of the propagation of the solitary wave and Figure 15 shows the solitary wave at different locations of the wave basin at y=0.

The following figures illustrate the numerical results along with the experimental data. Figures 16 and 17 display the free surface at three locations each. There is about 1 to 2 cm difference in peak heights between the numerical wave and experimental wave which is a relatively small difference considering the wave height (at the wavemaker) is 0.39 m. It is also noticeable that there is a phase difference of the order of 0.1 sec, the numerical wave lagging the benchmark wave (it may be noted that the abscissa in the figures is time in sec, so the longer the time, slower the wave) in all situations. The reasons can be attributed to the surface approximation using the mesh, roughness of bed and the limitations of potential flow itself (which is based on the assumptions of inviscid, irrotational, incompressible flow) in simulating the "real" flow.

Velocities U, V, W (U=Longitudinal along X; V=Lateral along Y; W=Vertical along Z) are compared at a location close to the free surface where the experimental data are made available in digital form. These results are shown in Figures 18 and 19. The magnitudes of U are very close (the real flow peak value is slightly smaller than that of the ideal flow) to each other and the same phase difference as in free surface peak can be noticed. These results are obtained close to wave breaking (just past the apex of the triangular shelf) causing numerical problems so the little perturbation in the U value in Figure 18 can be attributed to this reason. Velocity components V and W are very small in magnitude when compared with U in either case.

Qualitative comparison of velocities is made at several other locations where digitized experimental data are not available. In such cases, experimental figures are taken from reference [41]. The corresponding figures are numbered from 20 through 23. A general observation is that the peak velocity of U is qualitatively (in terms of both the shape of the curve and the magnitude of the peak) close in both the numerical and benchmark results in these additional plots. Figures 20 and 21 have velocity V as the smallest of the three velocities and the order of magnitude is also close in case of both numerical and experimental results. Velocity W (i.e., vertical) has a magnitude in between U and V and there is a close agreement here as well. However, it can be observed that the second part of the s-curve of W is somewhat different in the case of the numerical experiment. The bed is not very smooth and the assumption of cubic interpolation, degree of discretization of bottom surface, and idealized flow conditions in the model are some of the reasons for these little discrepancies in vertical velocity. Nevertheless, these values seem to fall within the level of accuracies acceptable to practicing engineers.

Figures 22 and 23 indicate a very good match of U and W. The patterns of variation in W with time are quite close in both numerical and benchmark results. V is relatively small and is more oscillatory in the case of the benchmark. Considering that these results are obtained on a sloping shelf, 3D-NWT can be concluded to result in a very good prediction of the velocities.

8 Comparison of results of the NS-FEM, the FNPF-BEM and the benchmark

Snapshots illustrating FEM results are provided in Figure 24. The commencement of the breaking wave over the triangular shelf is shown in Figure 25. Table-1 has the peak free surface elevation at different locations in the wave basin obtained from the FEM, the BEM and the laboratory experiment. The reason for comparing the peaks only is due to the non-availability of complete free surface elevation as a user-defined option in the FEM code. The 3D-NWT did not continue execution after the wave breaking thus limiting the locations of the BEM free surface results in this table. The velocities in the X,Y and Z directions (U,V,W in m/sec) are compared between the FEM and the experiment in Figures 26 through 28. There is a very good general agreement of both the free surface results and the velocities. It can be observed from Table-1 that with X<=13 m, the FEM and the BEM have magnitudes of peaks and their times of occurrences very close to each other. Both the FEM and the BEM results lag (by about 0.1-0.15 sec) behind the experimental wave, whereas the magnitudes of all three match very well. Once wave breaking happens, the FE mesh requires more refinement to capture the physics more accurately and the difference in peak free surface magnitudes in a few locations can be attributed to this reason.

9 Results of parallel performance

Figure 29 illustrates the performance using one and eight processors with several wave basin models (the largest one having 149,250 nodes) with variation of the FMA parameter – number of tree levels varied from 5 through 8. The number of tree levels is the number of hierarchical levels into which the cube surrounding the model has been divided. It can be observed that the fastest clock times of all these levels with 8 processors (solid lines) indicate a nearly linear performance (O(n)) with the node numbers (n). It can also be noticed that the single processor runs are not as linear as theoretically expected from the complexity measures of the FMA (except in some ranges of the nodes it may be called "piecewise linear") unlike the parallel runs.

Faster clock times can be achieved by increasing nlevels with an increase in model size. This is applicable with processors ranging from 1 to 8. As evident from this study (Fig. 29) , the user is now equipped with interesting design choices with the number of processors and the parameter nlevels to achieve an optimal clock time for a given model. For instance, one can reduce the number of processors to accommodate other users or other processes on the same machine by adopting appropriate parameter nlevels without incurring much inflation in clock time.

Tabl e 2 indicates the results of the benchmark model and Figure 30 also shows these results. In this study, 100 time steps reached a termination time of 1.52 sec. and the FMA cube length is 60 m. The speedup gain with the addition of processors is not as significant for this model because the transition of optimal time for nelvels=7 to 8 happens in between nodes 50000 and 60000 (from Figure 29). For instance, a model with greater than or equal to 60000 nodes and nlevels=8 can be expected to provide a better speedup for an increased accuracy. More detailed studies of the performance are presented in [42].

10 Insight and experience gained

The insight and experience gained in this work are summarized below. Although this experience refers to two numerical methods (BEM and FEM) and their implementations, the ideas are generic enough to be applicable to other (similar) situations.

- There is very good agreement between the results of the two models and the experiment. Fully nonlinear potential flow yields very good results and can be effectively used to model free surface flows. General viscous flows are more appropriate if viscosity is to be taken account in situations such as small-scale bodies or areas in proximity to boundaries.

- The wave breaking phenomenon can continue in the FEM unlike the BE model (which is actually a limitation of the potential flow model).
- The BE model uses a higher-order polynomial (cubic) for the element shape functions and tracks the free surface very accurately. The FE model uses hexahedral solid (brick) elements.
- Usually, finer discretization is required in the case of the FE models (due to use of lower-order shape functions) to yield comparably accurate results including free surface tracking. Such an increase, however, will affect the entire domain mesh and convert it into a very large-sized 3D model.
- The BE model requires only surface meshes, so finer discretization affects only the surface.
- The FE model required additional elements to model air and wavemaker parts.
- If the FE model is to be revised or made with a finer mesh, the analyst has to restart another model from scratch. In the case of the BE model, one can simply change a few parameters and proceed with execution.
- 3D-NWT in its current stage of development will save lot of time for the engineer in preprocessing at the additional cost of more computational time, which can be reduced with further work on software redesign in

order to run it faster (discussed in detail in another paper by the authors [42]).

- The FE model required lot of preprocessing effort (manual preparation) for the model. There is no direct way to input the complex bathymetry in the preprocessor, so the input file is edited to incorporate the same.
- 3D-NWT's input is relatively simpler (and smaller) and geared toward the NWT within its capabilities. An internal flag and a modification to read the grid is incorporated in order to read the bathymetry.
- Both models required the prior generation of the grid of cubic interpolation for the bathymetry.
- 3D-NWT is a special-purpose code for modeling the numerical wave tank (NWT), whereas the FE code is a general-purpose code. Thus, the latter can be used for solving other computational mechanics problems as well.
- The runtimes of the FE model are much faster (especially with the reduced sound speed) than the BE model for a comparable mesh size. However, the total turnaround time (i.e., elapsed time from the start to the finish) of the models is of particular interest to assess how long it takes to obtain the results. The FE model actually consumed more time in creating the mesh - it required about a week of effort to be able to create the basic mesh and input parameters and another week to be able to run the model successfully (which required some more debugging). This timeframe is given assuming the first author's familiarity with the FE code and its pre and post-processor before undertaking this study.

- There is no facility for the user to track the free surface in an automated fashion in the FE code. Velocities can be obtained only at the mesh nodes. By comparison, 3D-NWT can be input with gauges and internal points to obtain the velocities at desired locations.
- The FE model solves fluid flows based on direct numerical simulation (DNS). However, turbulence studies (which are more appropriate in case of impact on a structure) using FEM require a very fine-grained mesh with DNS or the use of more specialized models such as the Reynolds-Averaged Navier-Stokes Equation (RANS) which require further enhancements to the FE code and are presently in an advanced stage of development.

11 Concluding remarks

This work presented the comparison of the potential flow/Laplace equation and NS models using BEM and FEM, respectively, with the benchmark experiment conducted in the 3D wave basin at OSU, and validated some of the capabilities of the 3D-NWT code. It is also the first time that the 3D-NWT code with the newly incorporated developments is run to simulate a large-scale problem for the 3D wave basin with a complex bathymetry and parallel performance evaluated.

The comparison of results includes development of free surface and velocities at salient locations in both quantitative and qualitative (visual) forms. The results indicate that the potential flow model can be successfully used in many situations where a Navier-Stokes model is computationally demanding for attaining the same accuracy, and 3D-NWT is very appropriate for the free surface flows due to the advantages of surface-only mesh creation for 3D models. Some of the results for performance of the parallelized 3D-NWT code, including that of the benchmark problem, are also presented. Although the FE model ran faster for the same model size, the total turnaround time (including model preparation) is significantly shorter for the BE model and the latter has been more accurate and offered convenient input for the user. A very good comparison of results with the benchmark experiment indicates that the code (and hence the FNPF+BEM) can be confidently applied to real-world wave generation problems/wave basins. A detailed listing of the challenges, insight and experiences provided therein will be helpful for designing such experiments and research studies in future. Parallel performance studies demonstrate a good scale-up and linear performance for large-scale simulations.

A summary of the applicability of the 3D-NWT code to physical situations can be of help to physicists and engineers: this code is in general valid up to the moment the tip of the breaker jet hits the free surface and can be applied to deep to shallow water situations with arbitrary length waves. It doesn’t have small parameter assumptions, etc. as in analytical/numerical expansion wave theories. Bottom friction cannot be modeled directly and is not an important consideration in many cases. Many flow situations are inviscid (with large bodies of size greater than the wave length) and incompressible (with no air-water mixing) where the potential flow model can be applied with confidence. Thus it may be stated that (with the “caution” that there can be several situations where the numerical results are to be validated further on a case-by-case basis) numerical models in the area of hydrodynamics and/or CFD can strongly complement the experiments and are being prepared for engineers to use in their practice.

12 Figures and tables

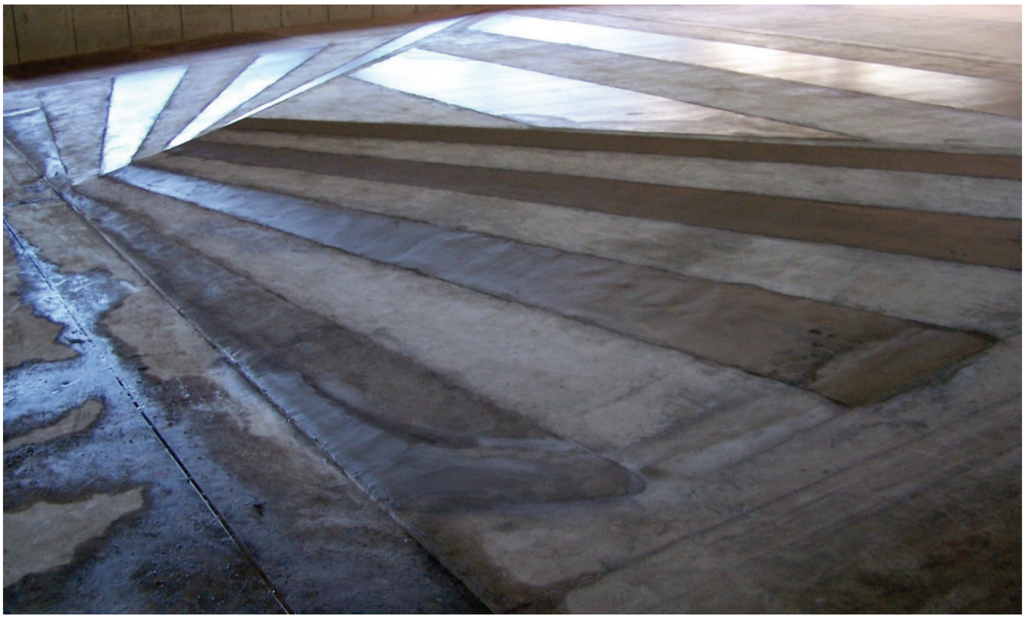


Figure 1: 3D view of the bathymetry used in the benchmark experiment (taken from [40])

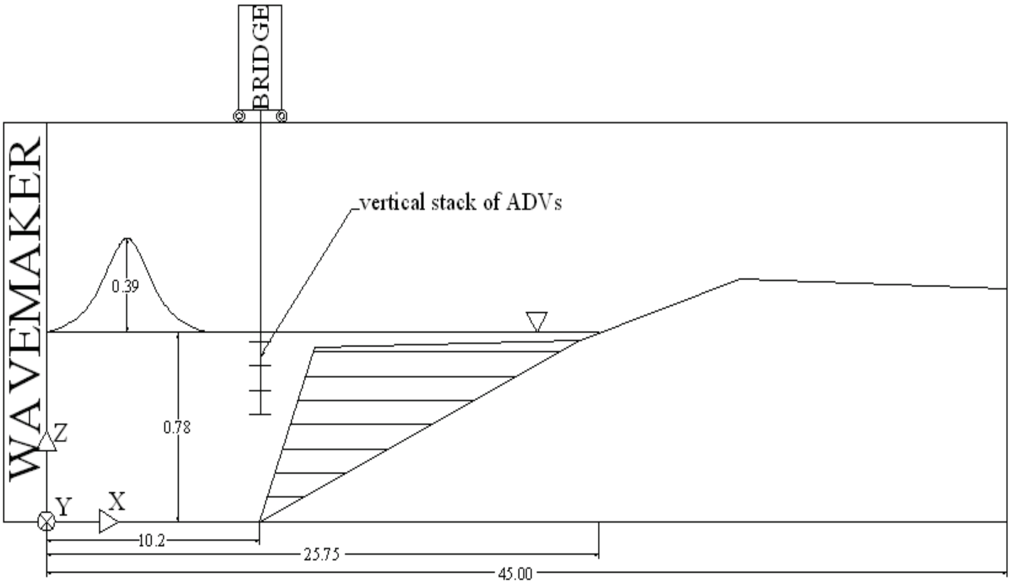


Figure 2: Cross-sectional view of the bathymetry used in the benchmark experiment (taken from [41]); ADVs = Acoustic Doppler Velocimeters (to measure velocity); Dimensions are in m

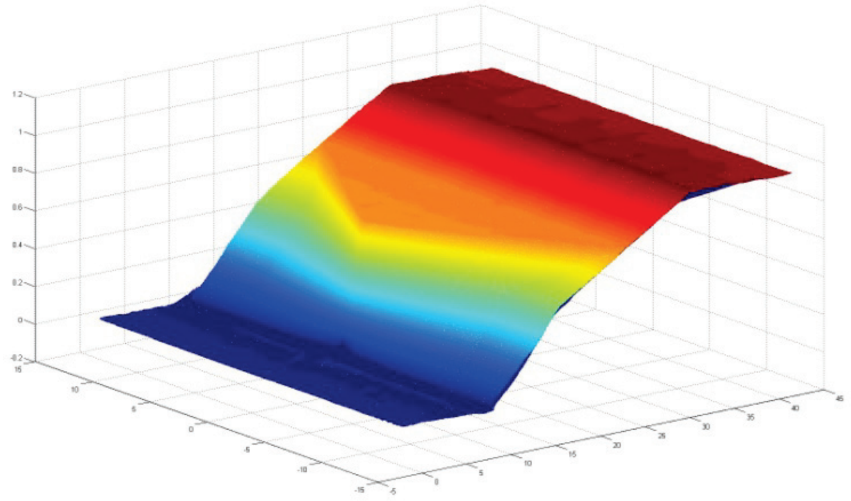


Figure 3: 3D view of the bathymetry, used in the benchmark experiment, plotted using a Matlab script (taken from [40])

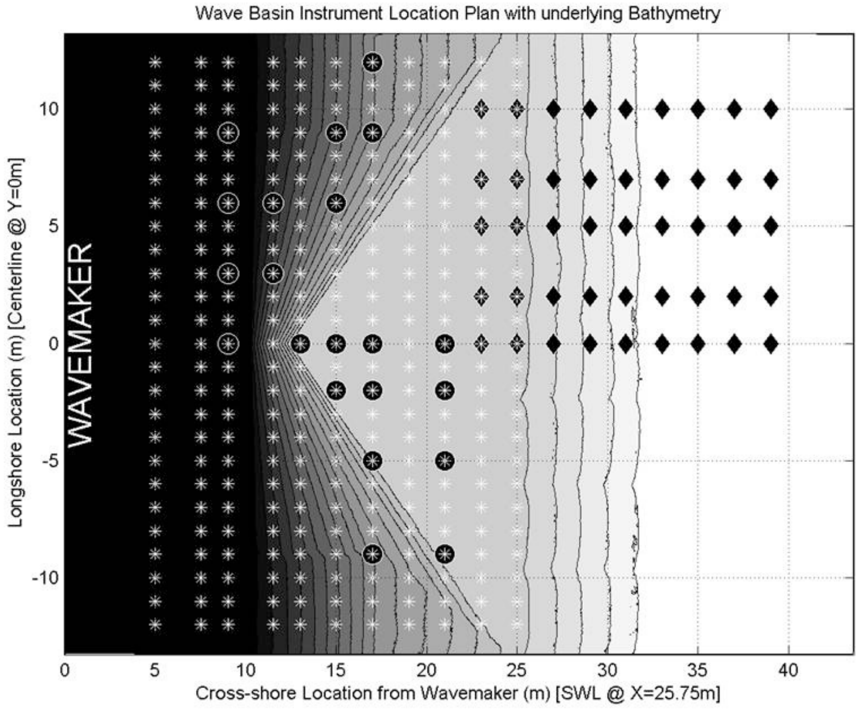

Wave Basin Instrument Location Plan with underlying Bathymetry
WAVEMAKER
Longshore Location (m) [Centerline @ Y=0m]
10
5
0
-5
-10
0
5
10
15
20
25
30
35
40
Cross-shore Location from Wavemaker (m) [SWL @ X=25.75m]

Figure 4: Plan view of the bathymetry used in the benchmark experiment (taken from [40]); *=Wave gauges, circle=ADVs (acoustic doppler velocimeter), diamond=ultrasonic wave gauges, SWL=still water line

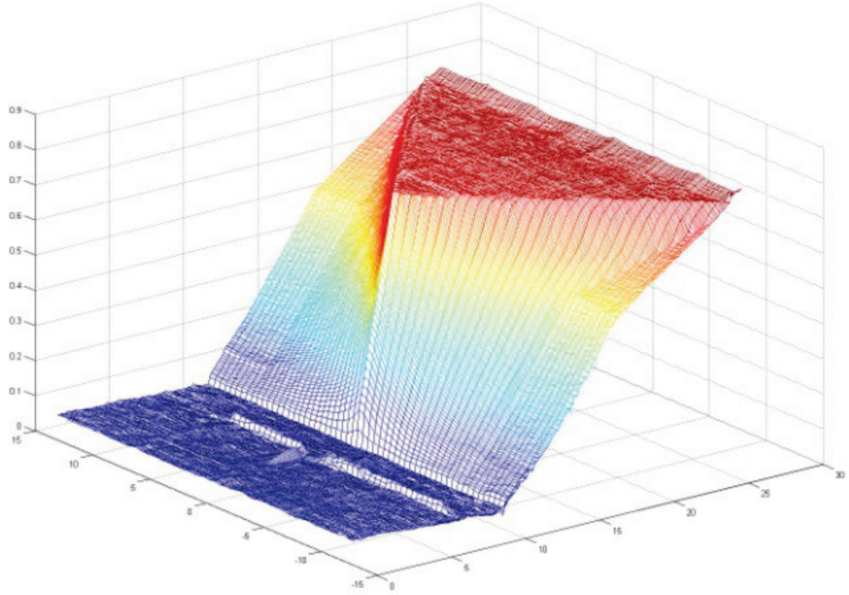


Figure 5: 3D view of the grid of the bathymetry used in BE and FE models

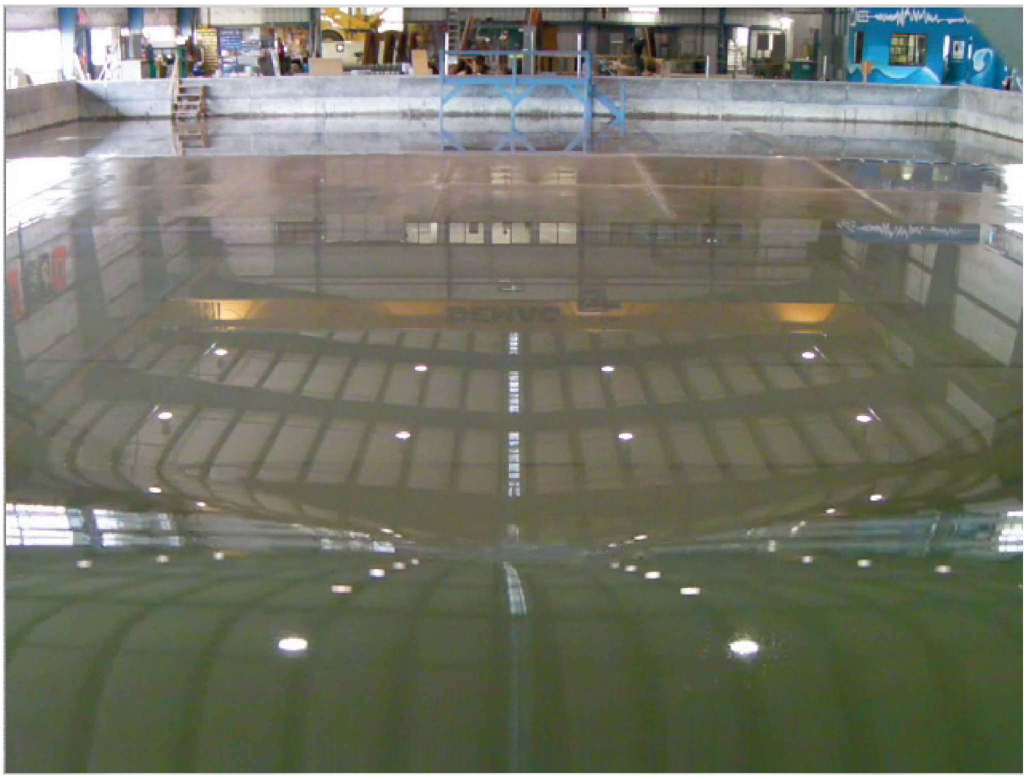


Figure 6: View of the wave propagation in the 3D wave basin at OSU

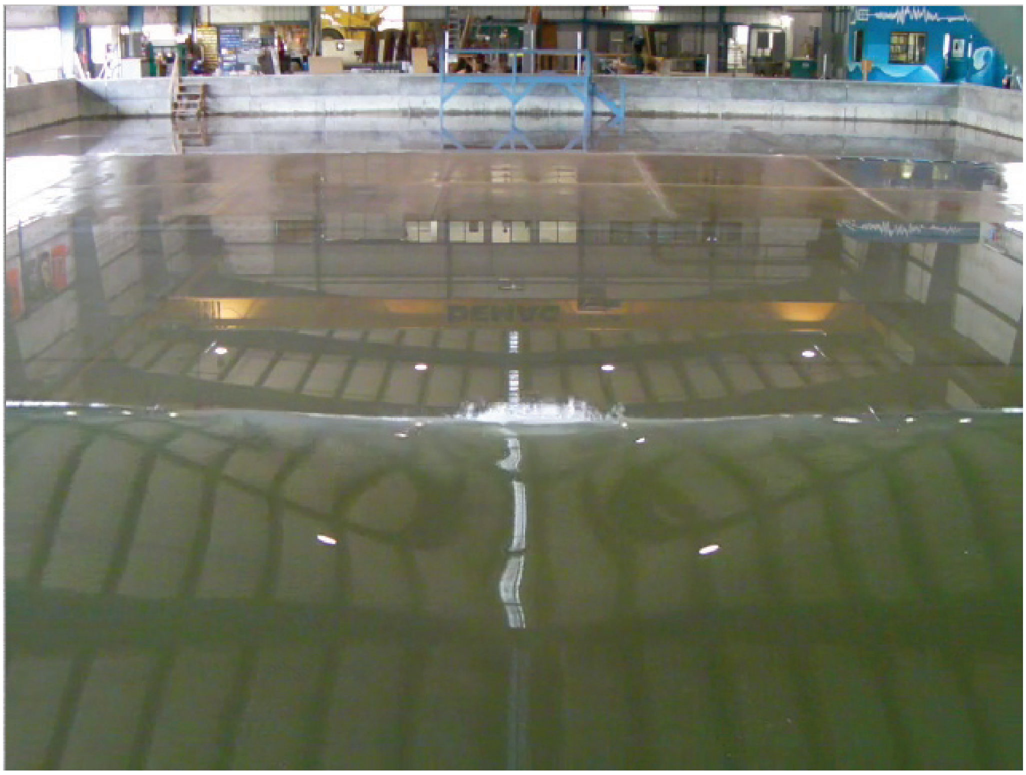


Figure 7: View of the wave propagation in the 3D wave basin at OSU; shows a breaking wave over the triangular sloping shelf

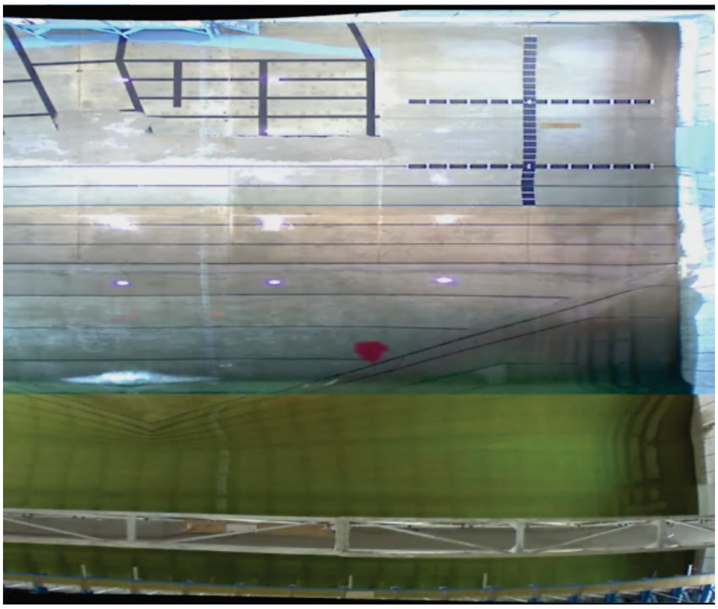


Figure 8: Top view of the wave propagation in the 3D wave basin at OSU; wave breaking and progress

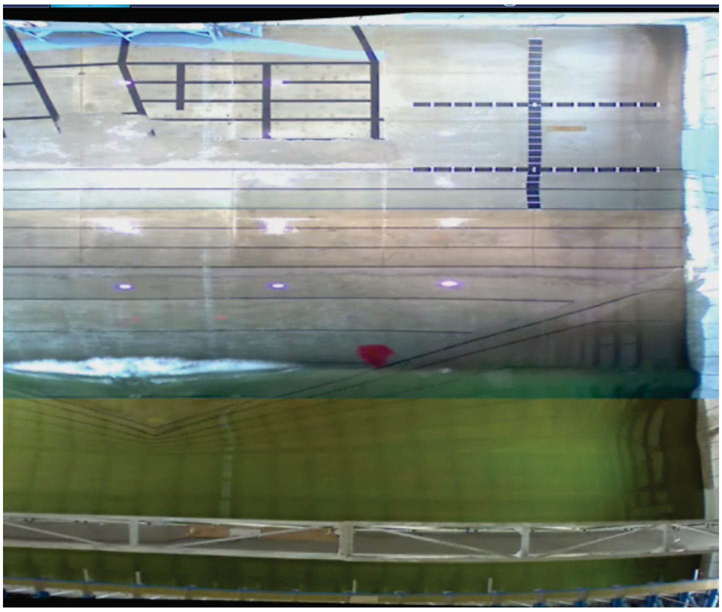


Figure 9: Top view of the wave propagation in the 3D wave basin at OSU; wave breaking and progress (following Fig. 8)

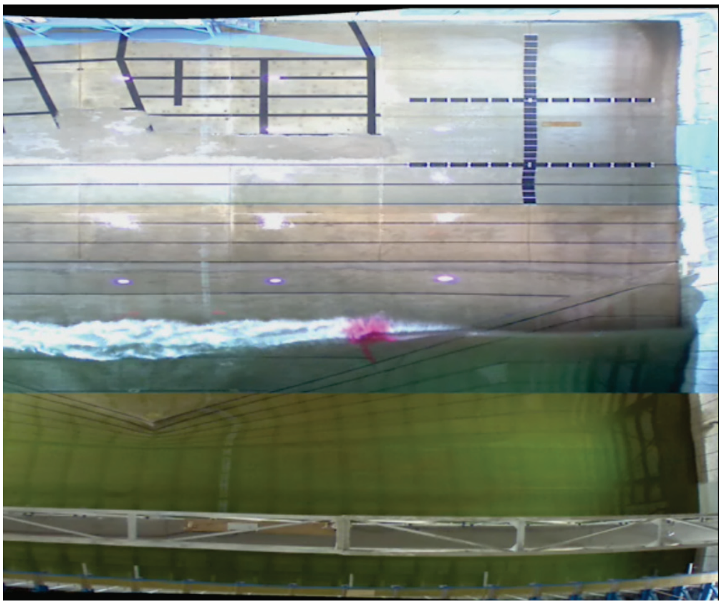


Figure 10: Top view of the wave propagation in the 3D wave basin at OSU; wave breaking and progress (following Fig. 9)

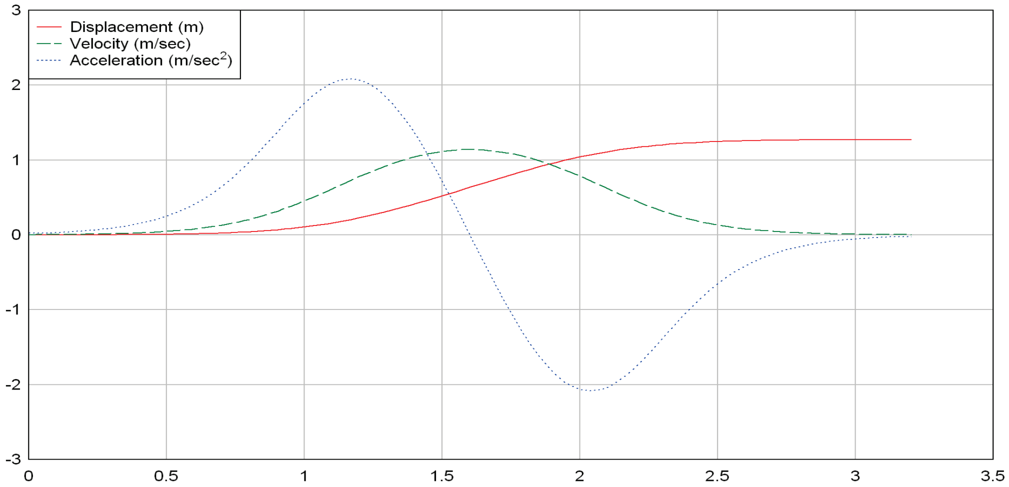


Figure 11: Motion input to the wavemaker: Time in sec (along X-axis) versus displacement in m (solid lines), velocity in m/sec (semi-solid) and acceleration in m/sec^2 (dotted)

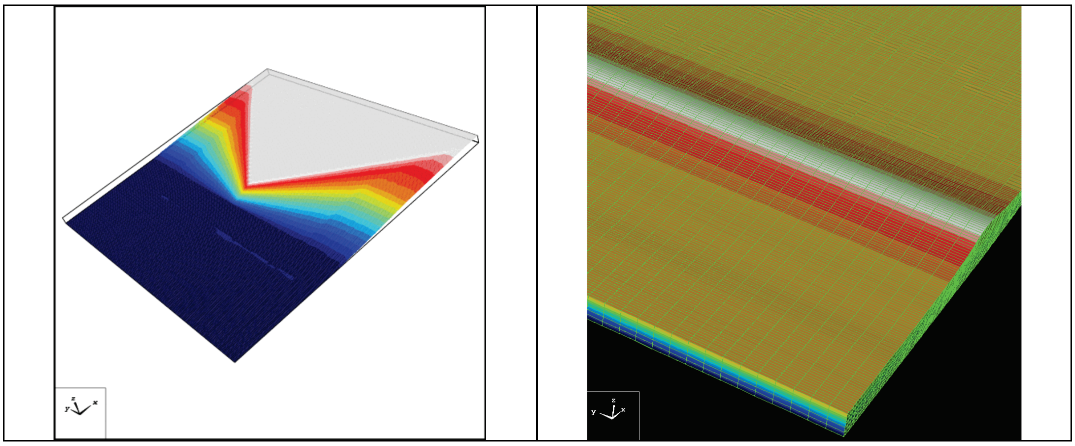


Figure 12: 3D view of the bathymetry used in the benchmark experiment (dark blue part indicates the deepest areas, white part indicates the highest areas of the triangular shelf; the colors in-between represent the elevations in-between); on the right side is the BE model using 3D-NWT showing mesh details along with a solitary wave

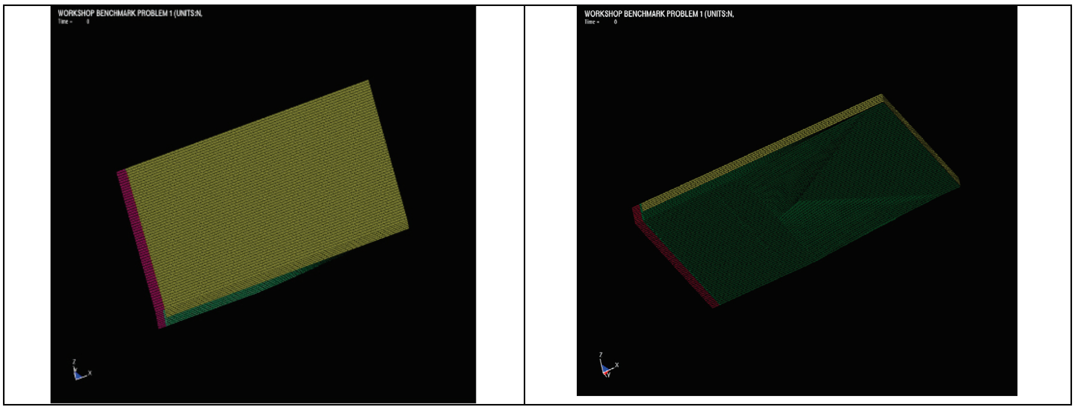

WORKSHOP BENCHMARK PROBLEM 1 (UNITS:N,
WORKSHOP BENCHMARK PROBLEM 1 (UNITS:N,

Figure 13: FE models (top and bottom views): Green part is water, yellow and red parts are air, wavemaker is situated between red and yellow parts.

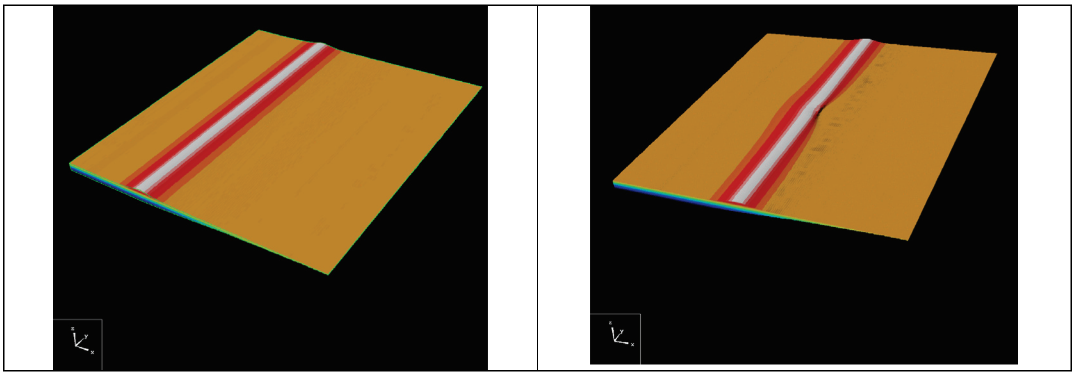


Figure 14: Benchmark problem-propagation of solitary wave; Snapshot on the right shows the result at the onset of breaking wave

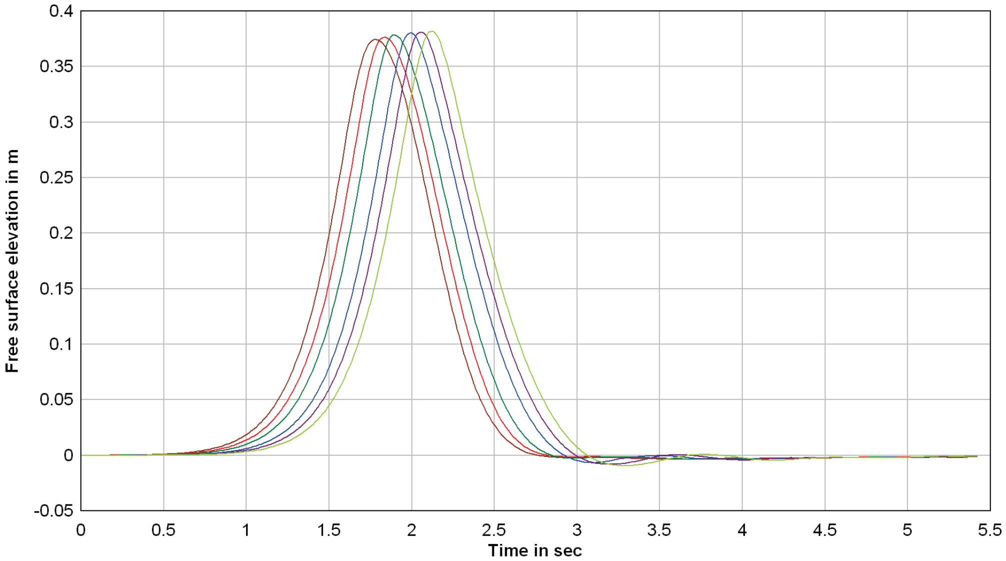


Figure 15: Propagation of solitary wave displayed at locations x=1.3m, 1.5m, 1.7m, 2.0m, 2.2m and 2.4m (all locations are at y=0), respectively

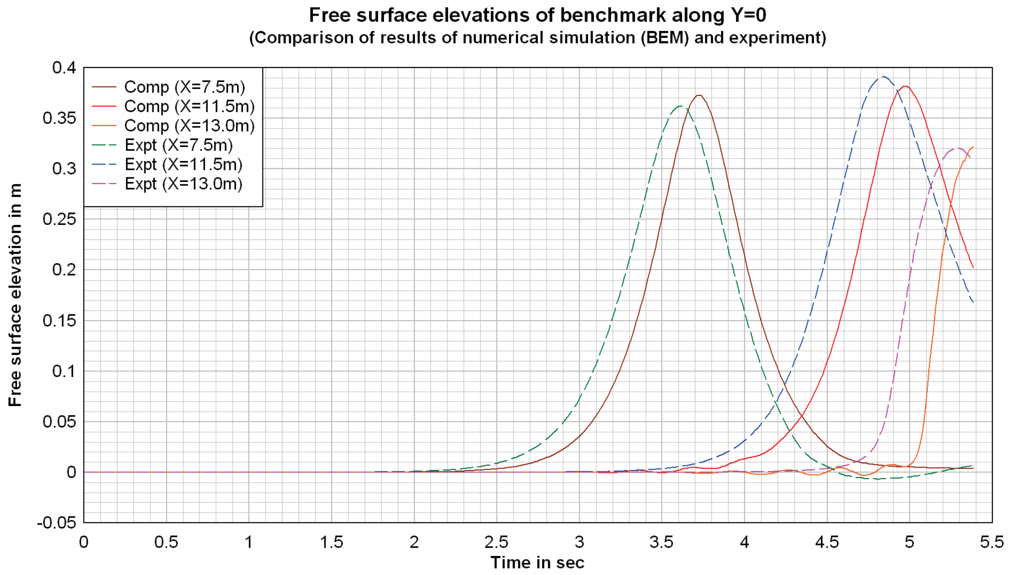


Figure 16: Free surface elevations at y=0 (time in sec versus free surface elevation in m): solid lines are from 3D-NWT at x=7.5m, 11.5m, 13.0m; dashed lines are from benchmark experiment at x=7.5m, 11.5m, 13.0m, respectively, as wave propagates. More recent peak corresponds to x=7.5m and farthest peak corresponds to x=13.0m. Numerical results "lag" behind the benchmark results by about 0.1 sec in general, but there is good agreement in the free surface elevations.

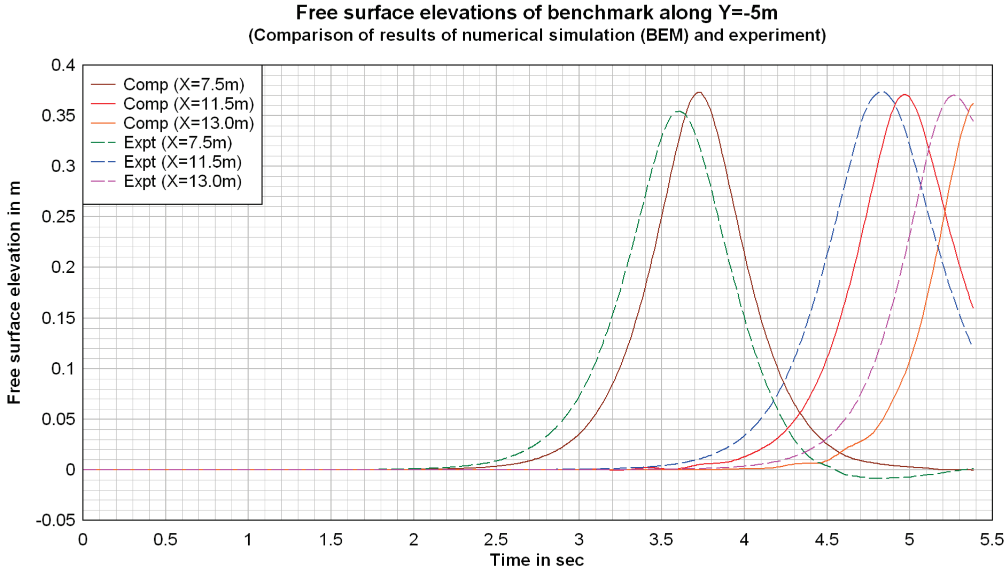


Figure 17: Free surface elevations at y=-5m (time in sec versus free surface elevation in m): solid lines are from 3D-NWT at x=7.5m, 11.5m, 13.0m; dashed lines are from benchmark experiment at x=7.5m, 11.5m, 13.0m, respectively, as wave propagates. More recent peak corresponds to x=7.5m and farthest peak corresponds to x=13.0m. Numerical results "lag" behind the benchmark results by about 0.1 sec in general, but there is good agreement in the free surface elevations.

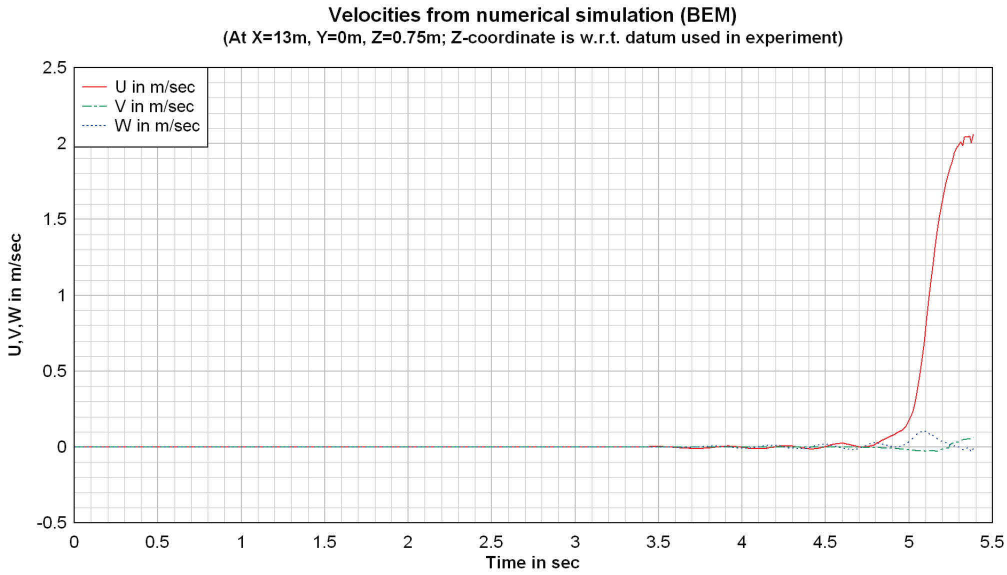


Figure 18: Velocities U, V, W (in m/sec) from 3D-NWT at x=13.0m, y=0m, z=0.75m (Z is w.r.t. datum used in the benchmark experiment)

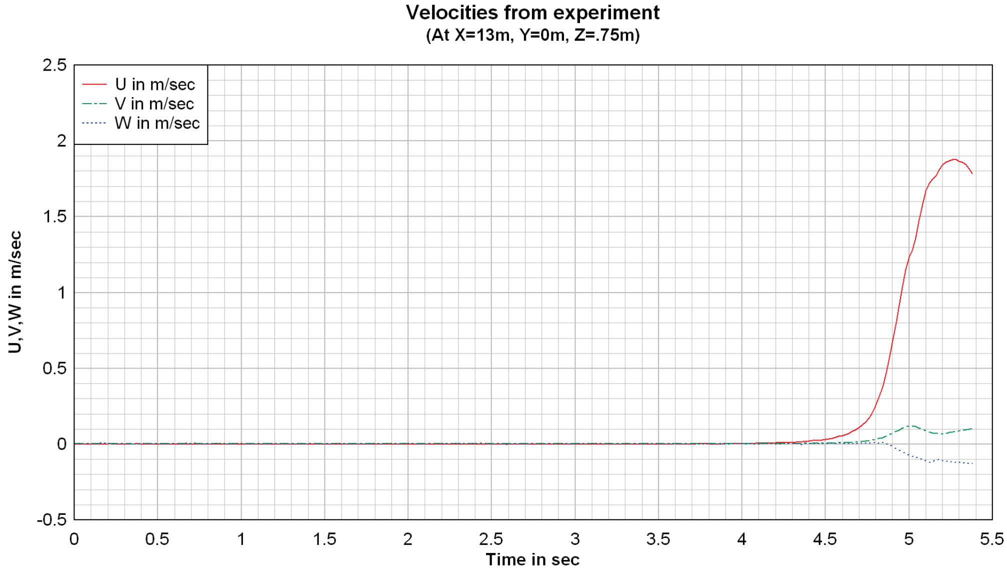


Figure 19: Velocities U, V, W (in m/sec) from the benchmark experiment at x=13.0m, y=0m, z=0.75m

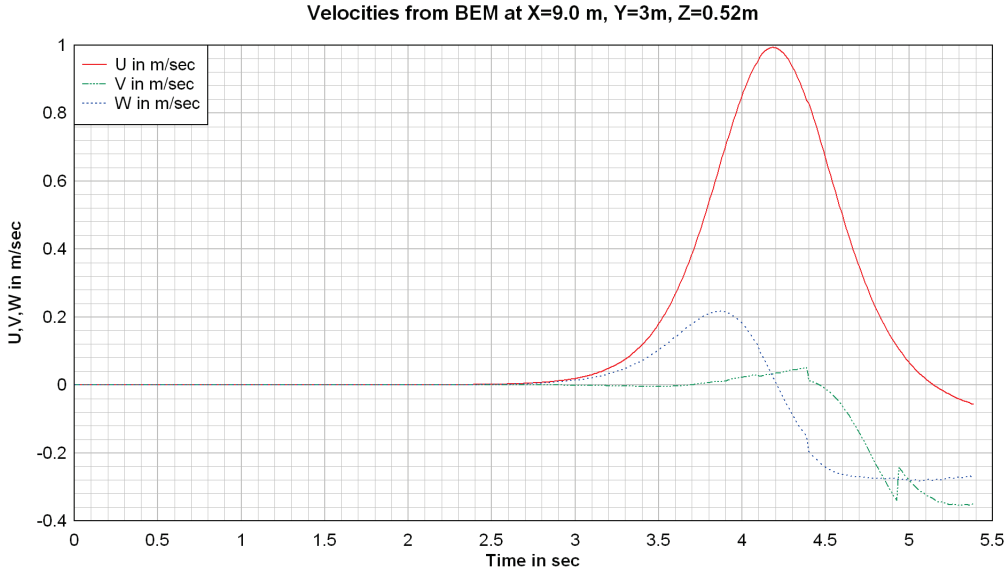


Figure 20: Velocities U, V, W (in m/sec) from 3D-NWT at x=9.0m, y=3m, z=0.52m (Z is w.r.t. datum used in the benchmark experiment)

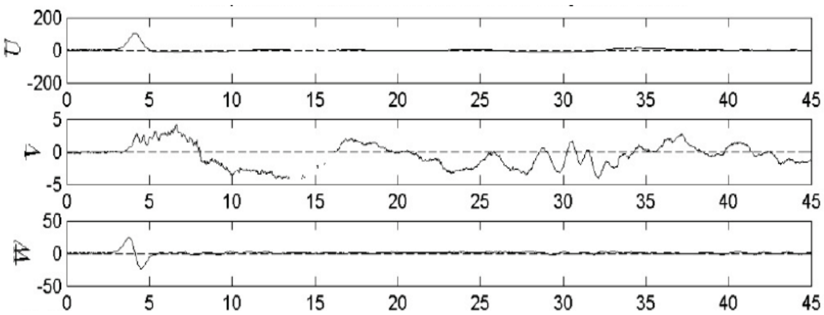


Figure 21: Velocities U, V, W (in cm/sec) from the benchmark experiment at x=9.0m, y=3m, z=0.52m

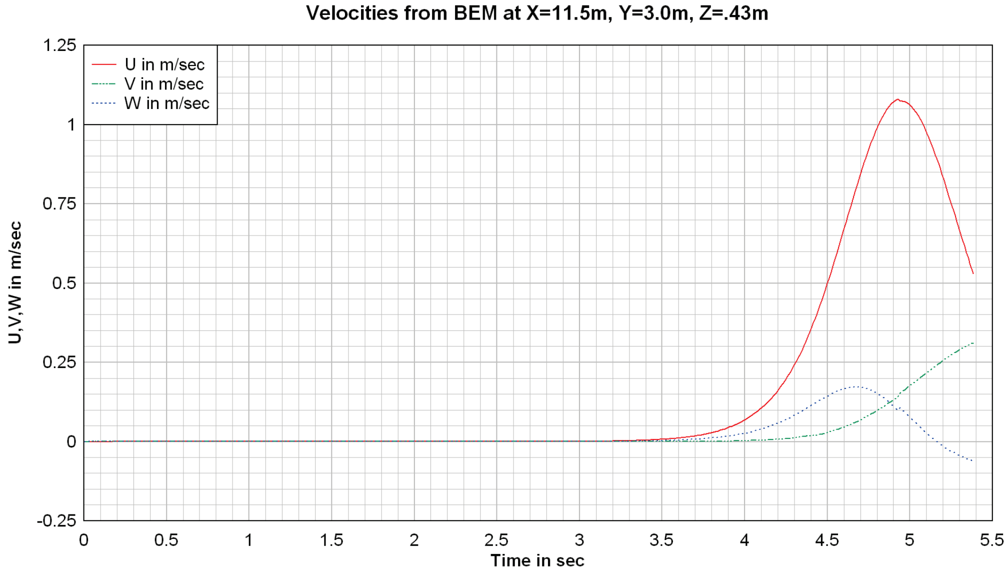


Figure 22: Velocities U, V, W (in m/sec) from 3D-NWT at x=11.5m, y=3m, z=0.43m (Z is w.r.t. datum used in the benchmark experiment)

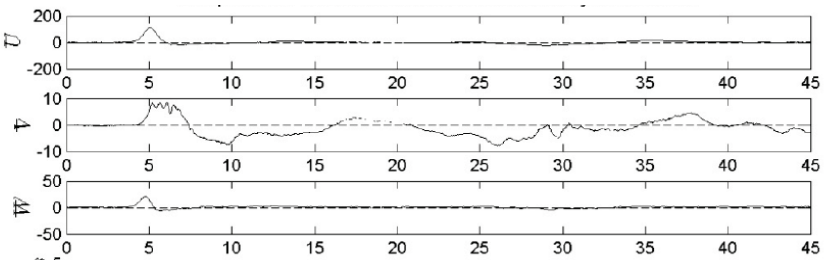


Figure 23: Velocities U, V, W (in cm/sec) from the benchmark experiment at X=11.5m, Y=3m, Z=0.43m

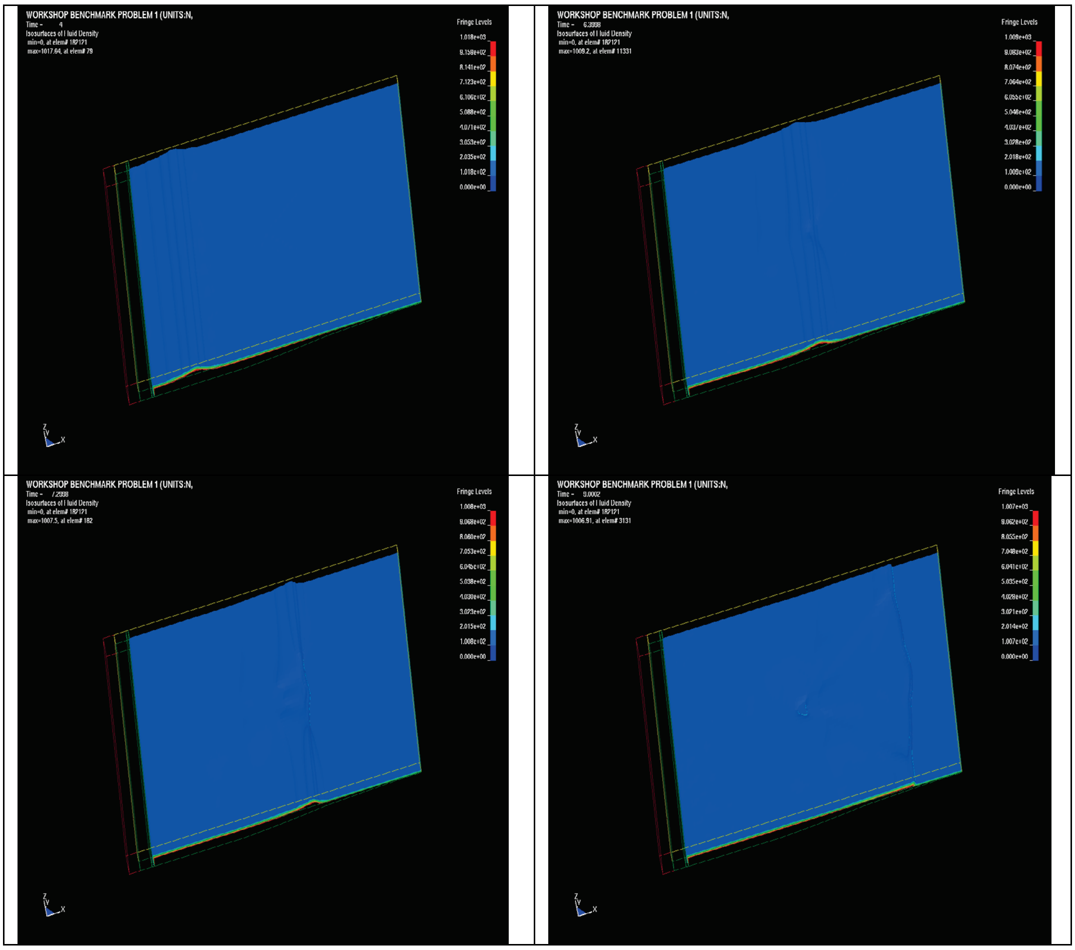


Figure 24: Screenshots of propagation of solitary wave using FEM

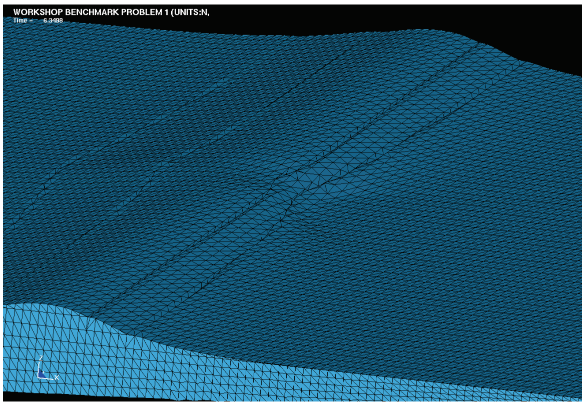


Figure 25: Close-up screenshot of the commencement of breaking wave over the triangular shelf in FE model

Table-1: Free surface peak elevation in m over still water shore line using FEM, BEM and lab experiment (Expt)

X (m)	Y (m)	Time (s)	FEM	Time (s)	BEM	Time (s)	Expt
7.5	0	3.7	0.37	3.7	0.37	3.6	0.36
11.5	0	4.95	0.37	4.97	0.38	4.84	0.39
13.0	0	5.4	0.31	5.37	0.32	5.28	0.32
15.0	0	5.85	0.27	-	-	5.98	0.25
17.0	0	6.45	0.24	-	-	6.5	0.18
21.0	0	7.6	0.14	-	-	7.76	0.14
7.5	-5	3.7	0.37	3.7	0.37	3.6	0.36
11.5	-5	4.95	0.37	4.97	0.37	4.82	0.37
13.0	-5	5.4	0.36	5.37	0.36	5.26	0.37
15.0	-5	5.95	0.36	-	-	5.84	0.37
17.0	-5	6.55	0.28	-	-	6.4	0.35
21	-5	7.75	0.15	-	-	7.56	0.13

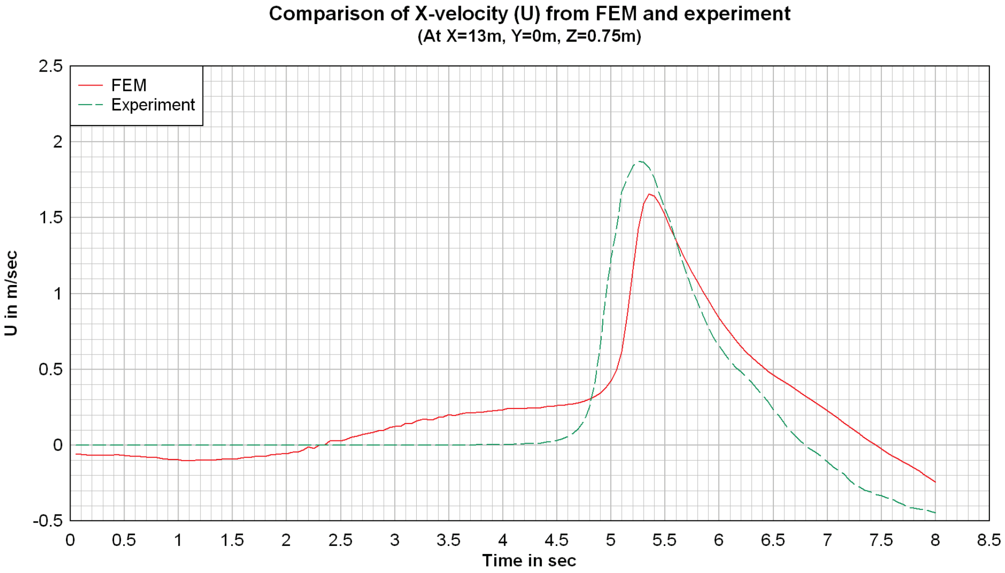


Figure 26: Comparison of X-velocity (U in m/sec): FEM (solid line) and experiment (dashed line)

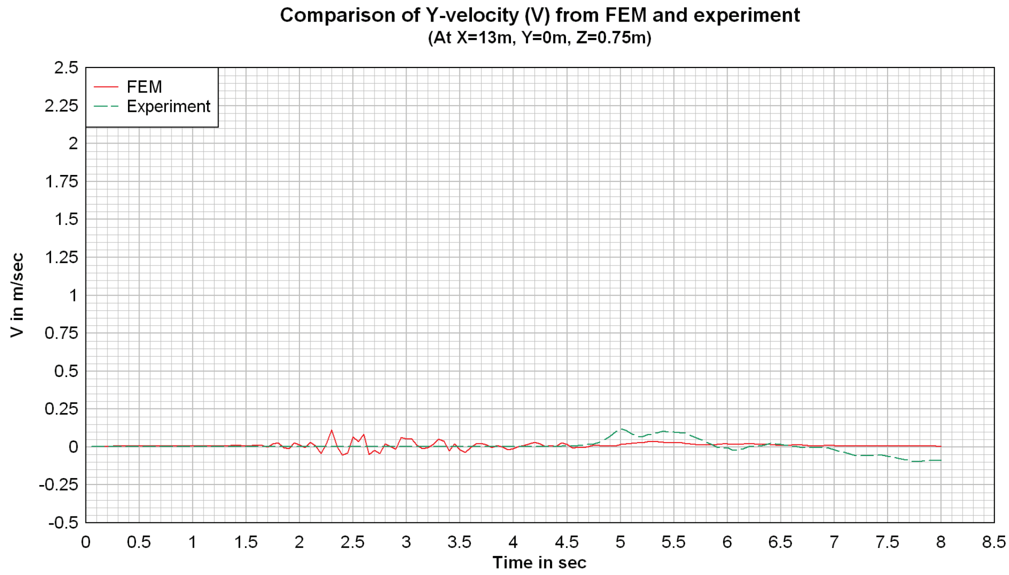


Figure 27: Comparison of Y-velocity (V in m/sec): FEM (solid line) and experiment (dashed line)

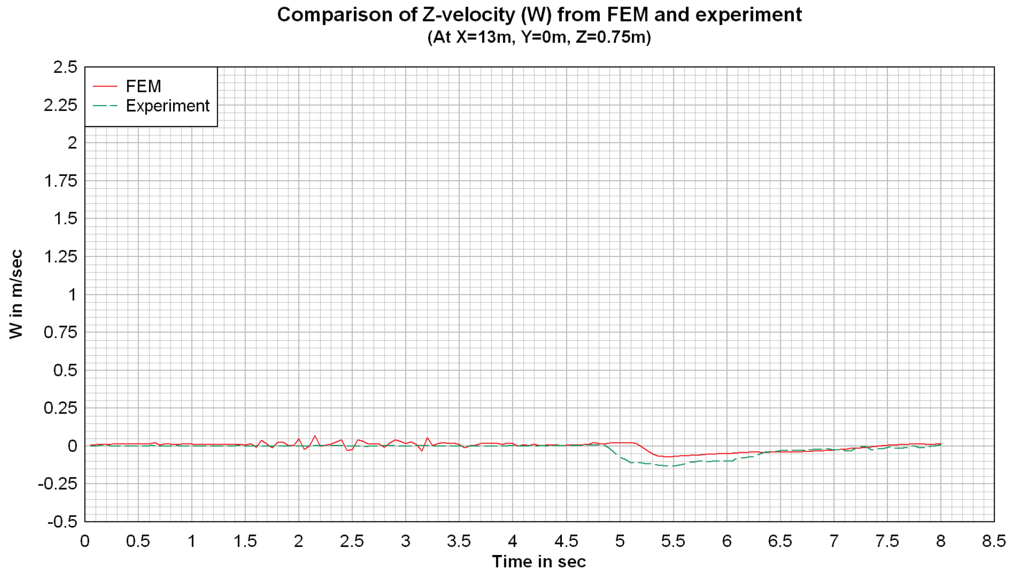


Figure 28: Comparison of Z-velocity (W in m/sec): FEM (solid line) and experiment (dashed line)

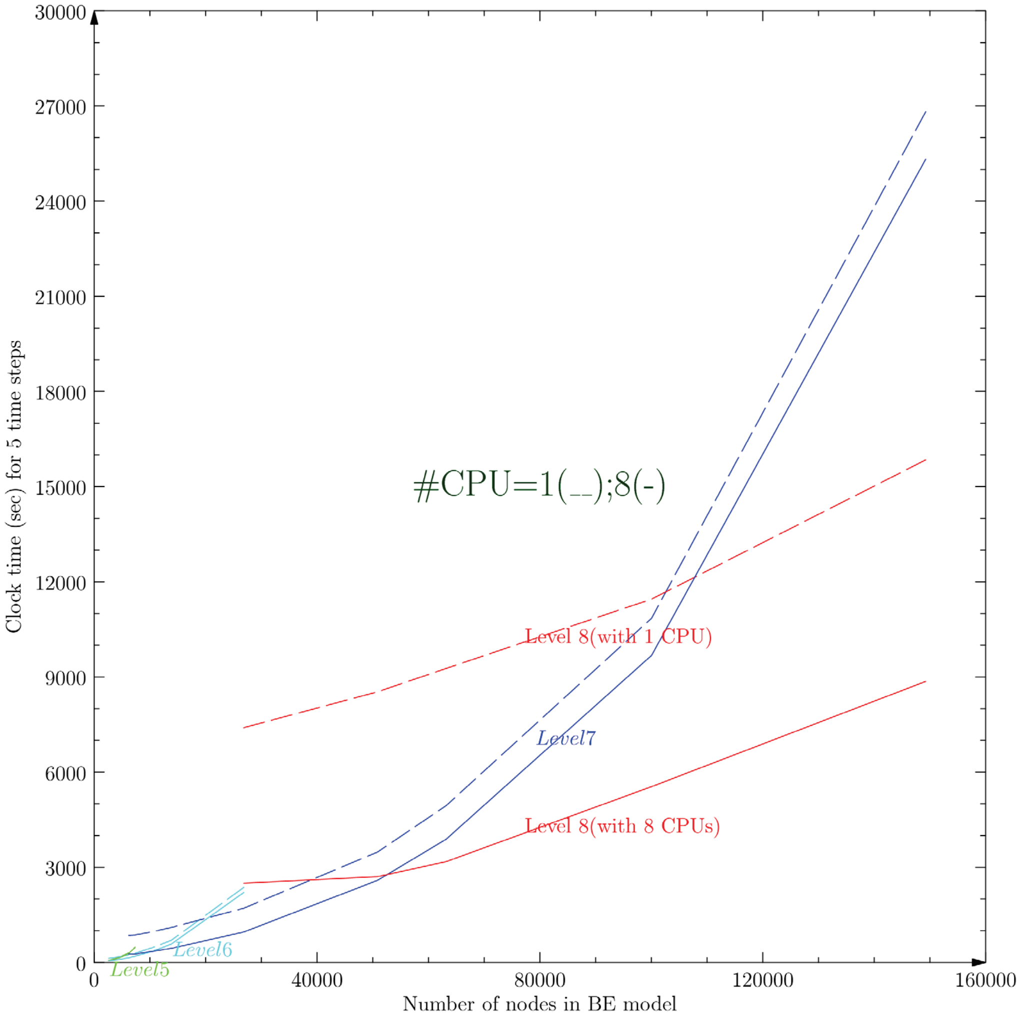


Figure 29: Performance using several models with nlevels=5, 6, 7, 8; solid line (-) is with 8 processors; dashed line (--) with single processor; there is nearly linear performance of run times with 8 processors (for instance, using solid lines of the fastest clock times of levels 6, 7 and 8)

Table 2:- Benchmark model: # of nodes=40228, # of tree levels in FMA=nlevels=7; 100 time steps		
Number of processors (ncpu)	Execution time in sec (hrs)	Speedup (T1/Tp)
1	51768 (14.38)	1.00
2	40495 (11.25)	1.28
4	34068 (9.46)	1.52
6	31578 (8.77)	1.64
8	30709 (8.53)	1.69

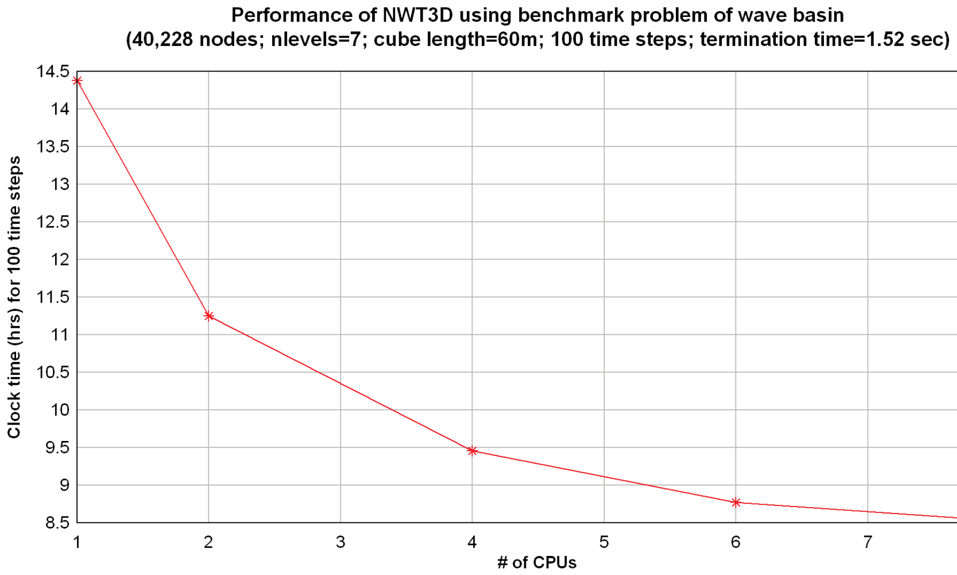


Figure 30: Performance (#of CPUs versus clock time in hours) using benchmark problem

Acknowledgement

The authors are thankful to the Office of Naval Research (ONR) for their generous financial support that made this work possible.

References

[1] Banerjee, P.K., and Butterfield, R., 1981, "Boundary element methods in engineering science," McGraw-Hill Book Co (UK) Ltd.

[2] Liggett, J.A., and Liu, P.L.F., 1983, "The boundary integral equation method for porous media flow," George Allen & Unwin Publishers, London.

[3] Brebbia, C.A., Telles, J.C.F., and Wrobel, L.C., 1984, "Boundary element techniques: theory and applications in engineering," Springer-Verlag, Berlin.

[4] Hartmann, F., 1989, "Introduction to Boundary elements," Springer-Verlag.

[5] Brebbia, C.A., and Dominguez, J., 1989, "Boundary elements: an introductory course," Computational Mechanics Publications, Southampton, Boston.

[6] Banerjee, P.K., 1994, "The boundary element methods in engineering," McGraw-Hill Book Company.

[7] Power, H., and Wrobel, L.C., 1995, "Boundary integral methods in fluid mechanics," Computational Mechanics Publications, Southampton, Boston.

[8] Wrobel, L.C., 2002, “The boundary element method- Applications in thermo-fluids and acoustics,” Volume 1, John Wiley & Sons, Ltd.

[9] Aliabadi, M.H., 2002, “The boundary element method- Applications in solid and structures,” Volume 2, John Wiley & Sons, Ltd.

[10] Bonnet, M., 1995, “Boundary integral equation methods for solids and fluids,” John Wiley & Sons Ltd.

[11] Nishimura, N., 2002, “Fast multipole accelerated boundary integral equation methods,” Applied Mechanics Review, 55(4), pp. 299-324.

[12] Ying, L., Biros, G., Zorin, D., and Langston, H., 2003, “A new parallel kernel-independent fast multipole method,” Proceedings of the 2003 ACM/IEEE Conference on Supercomputing, 196(2), pp. 591-626.

[13] Ying, L., Biros, G., and Zorin, D., 2004, “A kernel-independent adaptive fast multipole algorithm in two and three dimensions,” Journal of Computational Physics, 196(2), pp. 591-626.

[14] Margonari, M., and Bonnet, M., 2005, “Fast multipole method applied to elastostatic BEM-FEM coupling,” Computers and Structures, 83, pp. 700-717.

[15] Chaillat, S., Bonnet, M., and Semblat, J-F., 2008, “A multi-level fast multipole BEM for 3-D elastodynamics in the frequency domain,” Computer Methods in Applied Mechanics and Engineering, 197(49-50), pp. 4233-4249.

[16] Liu, Y., 2009, “Fast multipole boundary element method: theory and applications in engineering,” Cambridge University Press.

[17] Chen, G., and Zhou, J., 1992, “Boundary Element Methods,” Academic Press.

[18] Hsiao, G.C., and Wendland, W.L., 2008, “Boundary integral equations,” Springer-Verlag, Berlin, Heidelberg.

[19] Grilli, S.T., Skourup, J., and Svendsen, I.A., 1989, “An efficient boundary element method for nonlinear water waves,” Engineering Analysis with Boundary Elements, 6(2), pp. 97-107.

[20] Grilli, S.T., and Subramanya, R., 1996, “Numerical modeling of wave breaking induced by fixed or moving boundaries,” Computational Mechanics, 17, pp. 374-391.

[21] Grilli, S.T., 1993, “Modeling of nonlinear wave motion in shallow water,” Computational methods for free and moving boundary problems in heat and fluid flow, Wrobel, L.C., and Brebbia, C.A., (editors), Computational Mechanics Publications, Southampton, Boston.

[22] Grilli, S., and Svendsen, I.A., 1990, “Corner problems and global accuracy in the boundary element solution of nonlinear wave flows,” Engineering Analysis with Boundary Elements, 7(4), pp. 178-195.

[23] Grilli, S.T., Voropayev, S., Testik, F.Y., and Fernando, H.J.S., 2003, "Numerical Modeling and Experiments of Wave Shoaling over Buried Cylinders in Sandy Bottom," In Proc. 13th Offshore and Polar Engineering Conference (ISOPE03, Honolulu, USA, May 2003), pp. 405-412.

[24] Svendsen, I.A., Otta, A.K., and Grilli, S.T., 1992, "Unsteady free surface waves," Breaking waves, International Union of Theoretical and Applied Mechanics (IUTAM) Symposium, Sydney, Australia, 1991, Banner, M.L., and Grimshaw, R.H.J., (eds), Springer-Verlag.

[25] Grilli, S.T., Guyenne, P., and Dias, F., 2001, "A fully nonlinear model for three-dimensional overturning waves over arbitrary bottom," International Journal for Numerical Methods in Fluids, **35**(7), pp. 829-867.

[26] Brandini, C., and Grilli, S.T., 2001, "Modeling of freak wave generation in a 3D-NWT," In Proc. 11th Offshore and Polar Engng. Conf. (ISOPE01, Stavanger, Norway, June 2001), Vol III, pp. 124-131.

[27] Grilli, S.T., Vogelmann, S., and Watts, P., 2002, "Development of a 3D numerical wave tank for modeling tsunami generation by underwater landslides," Engineering Analysis with Boundary Elements, **26**(4), pp. 301-313.

[28] Fochesato, C., Grilli, S.T., and Guyenne, P., 2005, "Note on non-orthogonality of local curvilinear coordinates in a three-dimensional boundary element method," International Journal for Numerical Methods in Fluids, **48**, pp. 305-324.

[29] Fochesato, C., and Dias, F., 2006, "A fast method for nonlinear three-dimensional free-surface waves," Proceedings of the Royal Society, London, **A 462**, pp. 2715–2735.

[30] Guyenne, P., and Grilli, S.T., 2006, "Numerical study of three-dimensional overturning waves in shallow water," Journal of Fluid Mechanics, **547**, pp. 361-388.

[31] Fochesato, C., Grilli, S.T., and Dias F., 2007, "Numerical modeling of extreme rogue waves generated by directional energy focusing," Wave Motion, **44**, pp. 395-416.

[32] Sung, H.G., and Grilli, S.T., 2008, "BEM Computations of 3D Fully Nonlinear Free Surface Flows Caused by Advancing Surface Disturbances," International Journal of Offshore and Polar Engineering, **18**(4), pp. 292-301.

[33] Grilli, S.T., Dias, F., Guyenne, P., Fochesato, C., and F. Enet, 2010, "Progress In Fully Nonlinear Potential Flow Modeling Of 3D Extreme Ocean Waves," Chapter 3 in Advances in Numerical Simulation of Nonlinear Water Waves, Vol. 11 in Series in Advances in Coastal and Ocean Engineering, World Scientific Publishing Co. Pte. Ltd., pp. 75-128.

[34] Grilli, S.T., 1997, "Fully nonlinear potential flow models used for long wave runup prediction," Long-Wave Runup Models (eds. Yeh, H., Liu, P., and Synolakis, C.,), World Scientific Publishing, Singapore, pp.116-180.

[35] Nimmala, S.B., Yim, S.C., Grilli, S.T., “An Efficient Parallelized 3-Dimensional Numerical Wave Tank to Model Large-Scale Wave Basin Experiments,” (to be published).

[36] Steven A. Hughes, 1993, “Physical models and laboratory techniques in coastal engineering,” World Scientific Publishing.

[37] Derek G. Goring, 1978, “Tsunamis- The propagation of long waves onto a shelf,” W.M. Keck Laboratory of Hydraulics and Water Resources, Division of Engineering and Applied Science, California Institute of Technology, Pasadena, California, Report no. KH-R-38.

[38] John O. Hallquist, 2006, “LS-DYNA theory manual,” Livermore Software Technology Corporation.

[39] “LS-DYNA keyword user’s manual (volumes I and II)- version 971,” 2007, Livermore Software Technology Corporation, Livermore, CA.

[40] “ISEC Community Workshop: Simulation & Large-Scale Testing of Nearshore Wave Dynamics,” July 8-10, 2009- Corvallis, Oregon (http://isec.nacse.org/workshop/isec_workshop_2009/ ; http://isec.nacse.org/workshop/isec_workshop_2009/benchmarks.html).

[41] Swigler, D.T., 2009, “Laboratory study investigating the three-dimensional turbulence and kinematic properties associated with a breaking solitary wave,” M.S. thesis, Department of Civil Engineering, Texas A&M University.

[42] Nimmala, S.B., Yim, S.C., Grilli, S.T., “Performance Studies of a 3D Numerical Wave Tank Model for Large-Scale Wave Basin Experiments,” (to be published).

Generation of Focused Waves Using an Efficient 3D Numerical Wave Tank and Validation with Large-Scale Experiments

Seshu B. Nimmala and Solomon C. Yim
School of Civil & Construction Engineering, Oregon State University (OSU),
Corvallis, OR 97331, USA

Stephan T. Grilli
Department of Ocean Engineering, University of Rhode Island (URI),
Narragansett, RI 02882, USA

Abstract

Focused solitary waves exhibit real 3D wave phenomena and are of importance to study the directed wave impact on coastal infrastructure (structures such as bridges, levees, etc) and in laboratory experiments. In this context, this paper provides a description of the three alternatives of generating focused solitary waves that are incorporated in the three-dimensional numerical wave tank (3D-NWT) and validates the results using large-scale experiments. The authors developed the necessary theoretical background and algorithms for the three alternatives and implemented them in the numerical wave basin. The focused wave capability takes advantage of the newly added piston wavemaker and the high-performance computing (parallelization) features that are also recently incorporated by the authors. The actuators of the piston wavemaker's paddles are controlled by the user-specified time history of motion.

The 3D-NWT code is developed using the boundary element method (BEM) to discretize the fully nonlinear potential flow (FNPF) using the Laplace equation.

The numerical results demonstrate a very good comparison to physical experiments and thus strongly complement the latter with the potential capabilities of conducting parametric studies to obtain faster insight while incurring only a fraction of the cost. Performance studies of the parallelized 3D-NWT code are also presented in this paper to illustrate its applicability on high-performance computing (HPC) platforms.

1 Introduction

This paper first provides a summary of the literature survey, mathematical formulations that drive the 3D-NWT code, high-performance computing (performance studies) and discusses in detail the three alternatives of generating focused solitary waves and finally validates the results using physical experiments.

The code 3D-NWT is based on several years of research work of Grilli, et al [1-9]. It implemented the fully nonlinear potential flow (FNPF) using the boundary element method (BEM), a surface-only discretization method for 3D problems.

This reduction in dimension by surface modeling is of advantage in moving boundary problems which require regridding at several time steps besides the smaller problem size (number of nodes or elements) when compared to domain-based methods such as the finite element method (FEM). An engineering-oriented approach of BEM can be found in [10-18] and mathematical analysis can be found in [19,20]. The resulting advantages of the reduction in dimension are somewhat offset by the fact that the coefficient matrices are unsymmetric and fully-populated, thus requiring fast solution methods. Fast multipole algorithm (FMA) [21-26] is one of the methods that made BEM competitive in the world of numerical methods. The following discussion reviews some of the other references that cover the nonlinear free surface waves.

A study of the validity of potential flow theory for describing steep gravity waves produced in an experimental tank was attempted in [27]. A mixed Eulerian-Lagrangian (MEL) scheme was used in two dimensions and the results demonstrated that FNPF yields results very close to the physical results. References [28-30] describe the work of mathematicians Dold and Peregrine, who applied the boundary integral method for the propagation of 2D water surface waves including the overturning of water waves as they break. Their work was based on the Cauchy theorem boundary integral which permits the use of an iterative solver and calculation of the gradients of higher derivatives of the potential. Explicit time-stepping was based on a truncated Taylor series. This scheme improved efficiency by permitting relatively large time-steps to be made while maintaining a desired degree of accuracy. A discussion on stability and accuracy indicated that this method didn’t exhibit sawtooth numerical instabilities reported in other methods, when time steps are sufficiently small.

Professor P.J. Zandbergen of the Applied Mathematics Department of the University of Twente, Netherlands has advised several doctoral students as part of a major project- the numerical simulation of nonlinear water waves. This initiative generated substantial research work [31-34] by using the boundary integral equation method (BIEM) and FNPF. For large objects with a characteristic dimension of the order of the wave length, viscous effects of the fluid flow can be neglected as well as compressibility effects [31]. Small-amplitude theory (with wave amplitude smaller than the wave length or water depth) doesn't adequately describe the situations where nonlinearity is predominant. While the assumption of potential flow is not valid at all stages of a physical process, it is valid for many practical engineering situations and is at least very useful for a more general model. A detailed discussion of the mathematical properties such as existence, uniqueness and well-posedness can also be found in this reference. The numerical algorithm used a classical fourth-order Runge-Kutta method for time integration. The spatial discretization was done using the panel method- wherein the surface is divided into rectangular sub-surfaces called panels with their midpoints denoted as collocation points. The results were accurate and stable for linear and weakly nonlinear waves but displayed stability problems in the case of highly nonlinear waves.

Broeze's work [32] focused on improving the panel method developed in [31] for highly nonlinear wave problems. A table presented therein showed a number researchers favoring BEM over other methods to model these problems. Time integration schemes frequently used are the Adams-Bashforth-Moulton (ABM), Runge-Kutta (RK) and Taylor Series (2nd/3rd order). One obvious disadvantage of the panel method over the BEM is that the geometry and solution are known only at the collocation points of the panels which will necessitate some additional techniques to find the solution at the intersection of the surfaces.

Reference [35] presented applications of BIEMs for 2D nonlinear water wave problems and compared the use of both free space and periodic Green’s functions. The latter function was expected to offer the advantage of requiring integration only over the free surface but the computational efficiency was not realized in the numerical studies. This work also used the Taylor series for time-stepping. Cooker [36] presented a method based on Cauchy’s integral theorem, conformal mapping and a truncated Taylor series for time-stepping to model 2D wave propagation over irregular beds.

Lin, W.M. [37] studied the nonlinear behavior of the fluid motion near the intersection point of the free surface and a floating body (which was simplified to a piston wavemaker) using the potential flow in two dimensions. Singularity results from the confluence of two analytical boundary conditions at this point. Cauchy’s integral theorem is used to solve for the complex potential function along the boundary. The algorithm proposed therein satisfies both the free surface boundary condition and the wavemaker boundary condition at the intersection point.

A more recent work by Vinayan, et al [38] applied the BEM to solve nonlinear planar free-surface waves and validated with wave generation using Stokes fifth-order theory and a sinusoidal motion of a piston wavemaker in two dimensions. The theory has been extended to include the roll-motion of a 2D hull section. A direct LU solver is used to solve the system of equations which limits the size of the problems that can be solved.

Colicchio, et al [39] have described a domain-decomposition strategy for nonlinear flows (two-phase) in two dimensions. In this work, they coupled potential flow BEM with a Navier-Stokes solver combined with a level-set technique resulting in an efficient and accurate method. An overlapping region is defined between the two solvers wherein the interface location, pressure and velocity information is exchanged.

The 3D-NWT code described in this paper uses higher-order boundary element methods, combining both the free-space Green's functions and a fast multipole algorithm in combination with the generalized minimal residual method (GMRES). The numerical complexity is proportional to the problem size. The free-surface time updating is based on a second-order Taylor series expansion in an Eulerian-Lagrangian form. Wave absorption is accomplished by a combination of free-surface (pressure) damping and a lateral active absorbing wavemaker. The authors incorporated a piston wavemaker to accept the wave motion to be able to model the physical wave basin and parallelized it so as to be able to take advantage of high-performance computing platforms and to solve large-scale problems in a reasonable time frame.

2 Mathematical formulations

The 3D-NWT code is based on fully nonlinear potential flow, which uses the Laplace equation as the governing partial differential equation (of linear second-order elliptic type).

The Laplace equation is given by

$$\nabla^2\phi = 0 \tag{1}$$

where $\phi =$velocity potential function and velocity $\vec{u} = \nabla\phi$.

The nonlinear flow originates from the presence and computation of the free surface. On the free surface, the potential satisfies nonlinear kinematic (KBC) and dynamic (DBC) boundary conditions. We need these two conditions since there are two unknowns (position and potential) of the free surface. The KBC states that the normal velocity of the free surface interface must be equal to the normal fluid velocity at the surface. Following a fluid particle at the free surface, this means that the particle remains at the free surface. The DBC, obtained from Bernoulli's equation, states that the pressure on the free surface equals atmospheric pressure which will be taken to be zero.

If $\vec{R}(t)$ is the position vector (x,y,z) of a fluid particle on the free surface, g is the acceleration due to gravity, $\vec{u}$ is the velocity vector (u,v,w),

$$\frac{D\vec{R}}{Dt} = \vec{u} = \nabla\phi \tag{2}$$

(from KBC)

$$\frac{D\phi}{Dt}=\frac{1}{2}\nabla\phi.\nabla\phi-gz \tag{3}$$

(from DBC)

For fixed boundaries, the no flow condition (zero flux) is specified as $\frac{\partial\phi}{\partial n}=0$, where n=normal vector to the surface. In the case of moving boundaries, the wavemaker generates waves with motion and velocity (both along X direction) that are specified as BCs:

$$x=x_p;\quad \frac{\partial\phi}{\partial n}=\vec{u}_p.\vec{n}, \tag{4}$$

where $\vec{n}$ is the normal to the wavemaker.

The initial free surface boundary condition (at t=0) is given by the still water level and zero potential (Dirichlet BC) and all other surfaces have zero or specified flux (Neumann BC).

Second-order explicit Taylor series expansions are used to find the new position and potential on free surface at each new time step. These expressions are found to yield explicit, stable and efficient solution techniques.

$$\vec{R}(t+\Delta t)=\vec{R}(t)+\Delta t\frac{D\vec{R}(t)}{Dt}+\frac{(\Delta t)^2}{2}\frac{D^2\vec{R}(t)}{Dt^2}+O\left[(\Delta t)^3\right] \tag{5}$$

$$\phi(\vec{R}(t+\Delta t))=\phi(t)+\Delta t\frac{D\phi(t)}{Dt}+\frac{(\Delta t)^2}{2}\frac{D^2\phi(t)}{Dt^2}+O\left[(\Delta t)^3\right] \tag{6}$$

A piston wavemaker in three dimensions with the wavemaker boards/paddles driven by the actuators is described below; these expressions are developed to be general enough such that each board can oscillate independently while maintaining connections at the edges with other boards:

Wavemaker displacement $\vec{x} = \vec{x}_p$; Velocity $\vec{u} = \vec{u}_p = \frac{d\vec{x}_p}{dt}$;

Acceleration $\ddot{\vec{u}} = \ddot{\vec{u}}_p = \frac{d\vec{u}_p}{dt}$

Direction of $\vec{x}_p$, $\vec{u}_p$ and $\ddot{\vec{u}}_p$ is along global X.

Let $\vec{n}$ be the outward normal (pointing away from the domain) to the piston wavemaker, $\vec{s}, \vec{m}$ are tangential vectors, $\vec{s}.\vec{m} = 0$ and $\vec{s} =$ Tangent vector along global Z (i.e., vertical axis), $\vec{n} = \vec{s} \times \vec{m}$

$$\phi_n = \frac{\partial \phi}{\partial n} = \vec{u}_p.\vec{n};\ \phi_m = \frac{\partial \phi}{\partial m} = \vec{u}_p.\vec{m};\ \phi_s = \frac{\partial \phi}{\partial s} = \vec{u}_p.\vec{s} = 0 \tag{7}$$

$$\frac{\partial^2 \phi}{\partial t \partial n} = \ddot{\vec{u}}_p.\vec{n} + \vec{u}_p.\frac{d\vec{n}}{dt} + \phi_n(\phi_{ss} + \phi_{mm}) - \phi_m \phi_{nm} \tag{8}$$

Let the coordinates of the edges of a waveboard be (x2,y2,z) and (x1, y1,z). So,

$\vec{m} = (x1 - x2)i + (y1 - y2)j;\ \vec{s} = k;\ \vec{n} = -(y1 - y2) * i + (x1 - x2) * j$

where i,j,k are unit vectors along the global X,Y and Z axes, respectively.

$$\frac{d\vec{n}}{dt} = (\vec{u}_{p1} - \vec{u}_{p2})j, \tag{9}$$

since the actuators move the boards only in the X direction.

3 High-performance computing (HPC)

Since HPC is moving in a direction where the multi-core machines are going to be ubiquitous (as evident from the current direction of the processor making industry) it is imperative that the numerical applications are redesigned to utilize these resources. The authors analyzed a few alternative options and found that the most optimal way of parallelizing 3D-NWT is by using OpenMP [40]. A detailed study on the performance of the parallel 3D-NWT code is presented by the authors in another paper [41] and a few results and observations are highlighted in this section. A Dell Precision WorkStation 690 with eight processors, Intel Xeon 3 GHz, 64GB RAM, 64 bit Redhat Linux Enterprise 5 is used as a test bed. This is an example of the shared memory type of computing systems.

Table 1 indicates the computational performance using a model (with 63150 nodes) of the 3D wave basin at Oregon State University (OSU) and Figure 1 illustrates these results. Speedup reported is given by the ratio of clock time using single processor divided by the clock time using multiple processors (= T1/Tp). It can be observed that the maximum speedup with 8 processors is 2.92 and the speedup is not gaining a significant increase after the initial addition of a few (~5-6) processors. This can be expected as the speedups are dependent on the parallel component of the total computation and the associated overhead. From a user's perspective, it would mean that one has to increase the model size in general to realize better speedups with a larger number of processors.

Figure 2 shows the performance of the 3D-NWT using one and eight processors for a variable number of nodes of wave basin models (node numbers range from 26838 to 149250; number of tree levels or nlevels in FMA=8). The speedups of the five models tested are 2.97, 3.15, 2.92, 2.07, 1.79. The value of speedup is found to show a downward trend after reaching a maximum of 3.15 with the increase in the model size for the same nlevels. It has been later "discovered" that speedups can also be improved considerably by increasing nlevels in FMA.

The observations from this parallelization effort are summarized below:

1. Parallel codes using OpenMP work by creating the user-specified number of threads. The threads are different from processes (an example of process is a serial program) in that they are lightweight, independent instructions that execute within a process. All threads created by the same process share the same memory space. Thus threads can take advantage of accessing the same data in centralized memory without duplicating the data and avoid transfer of data as needed by message passing paradigm (as used by MPI). However the users of OpenMP have to be extremely cautious to take care that threads do not write to the same location in memory at the same time and avoid the several possible pitfalls of multi-threaded programming. The number of threads can be as many as the number of processors in order to achieve load balancing (ideally each processor taking equal load), but not necessarily limited to them. However, using too many threads more than the number of available processors can cause the system to become nonresponsive to other tasks (and also to other users). A practical point of significance is that the overhead (including the creation of threads) for parallelization is not negligible. The size of the parallel component of computation must be significant enough to overcome the overhead. In other words, the problem size (or model size in a simulation) should be large enough to realize good speedups. It is prudent to start with a small

number of time steps for a given model, and study the speedup by varying the number of threads and grow the termination time from there.

2. FMA input parameter - number of tree levels: This parameter indicates the number of hierarchical levels into which the cube (containing the model) is divided. The speedup is considerably affected by this parameter as the work shared by the threads is dependent on the number of cells in the hierarchy. The user should be able to experiment and determine the optimum parameter for a specified number of threads for a given problem size. The practical values of this parameter range from 4 to 8.

3. FMA input parameter - cube length: This is the cube size surrounding the entire dimensions of the model. The cube is defined with the origin of the model at its center. The user has to identify the maximum dimension from the origin of the model, estimate the possible change in length (due to wave absorption for instance) and use twice of it for the cube length. One should use as low a value as possible for optimal performance of the algorithms.

4. FMA input parameter - number of terms in a multipole expansion: The larger this number, the better the accuracy of the expansions and higher the execution time.

5. The distribution of nodes (or the model geometry and number of nodes) in the FMA cube can also have an effect on the clock time.

4 Focused waves

Three models of focused waves have been developed and implemented in 3D-NWT. The following sections describe the theoretical details of the three models. The general expressions of piston motion input described in the previous section are applicable for the waveboards driven by the actuators to generate the focused waves. The additional information used in these models is the celerity (C) of the waves and the coordinates of the focus where the wave is directed.

4.1 Model-1: This model uses the time-lag algorithm. Each actuator moves according to a pre-computed time-lag and the closer is the actuator to the focus, the slower the beginning of the movement. Time-lag is the time after which the specific actuator starts following the farthest actuator from the focus.

1. Compute the radial distance of the focus from all actuators.

$$R_i = \sqrt{\left(x_f - x_i\right)^2 + \left(y_f - y_i\right)^2} \tag{10}$$

where (x_f, y_f) and (x_i, y_i) are the coordinates of the focus and the initial position of the actuator, respectively.

2. Compute the time taken for the wave to reach the focus from the actuator,

$$t_i = \frac{R_i}{C} \tag{11}$$

3. Compute the maximum time, $t_{\max} = \max(t_i),\ i = 1,...,N$

(N=total number of actuators)

4. Compute the time-lag from each actuator, $t_{Li} = t_{\max} - t_i$

5. Start the farthest actuator first (or one with $t_{Li} = 0$) at time step, t=0. Compare the time-lags of each actuator with the current time step and decide to move if the time-lag is less than or equal to the current time step. Repeat this process until all actuators are activated. The initial and final positions of the wave boards (that constitute the wavemaker), i.e., when they are at rest, are parallel for focused solitary wave generation, but the intermediate positions follow curved paths.

4.2 Model-2: The wavemaker is specified with the geometry of a circular arc computed from the radial distance R_s=R_i (equation (10)), which is the shortest distance between the focus and the wavemaker, as the initial condition in this model. Then all the actuators start moving along the X direction from the first time step itself following the specified uniform motion.

4.3 Model-3: The algorithm of this model is based on the idea of motion of the actuators with a uniform radial velocity toward the focus. Since there is only a limited length of time- the duration of wave generation for the wave boards to influence the wave characteristics, it is therefore of research interest to compare the results from this model with the rest of the models.

From equation (10):

$$\frac{dR_i(t)}{dt}=\frac{1}{R_i(t)}\left|x_i(t)-x_f\right|\frac{dx_i(t)}{dt}, \tag{12}$$

since the motion of the actuator in 3D wave basin at OSU is constrained to be in the X-direction only.

$$\frac{dx_i}{dt}=f(x_i(t))\frac{dR_i(t)}{dt}=f(.)V(t) \tag{13}$$

$V(t)$ is computed from the location on the wavemaker where $y_i=y_f$:

$$\frac{dx_v}{dt}=\frac{dR_i(t)}{dt}=V(t) \tag{14}$$

$$\frac{d^2x_i}{dt^2}=\frac{1}{(x_i-x_f)}\left[R_i\frac{d^2R_i}{dt^2}-\left(\frac{y_i-y_f}{x_i-x_f}\right)^2\left(\frac{dR_i}{dt}\right)^2\right], \tag{15}$$

where $\frac{d^2R_i}{dt^2}$ is the acceleration of the wavemaker prescribed as the time series

$$x_{i+1}=x_i+(t_{i+1}-t_i)\frac{dx_i}{dt} \tag{16}$$

This algorithm modifies the velocity and acceleration of each actuator based on equations (13) and (15), wherein V(t) is the prescribed velocity time series for a plain solitary wave as in equation (14). Motion of the actuators is stepped forward using equation (16).

5 Discussion of results

The results include a comparison with laboratory experiments run in the 3D wave basin at OSU [42]. The following results are obtained using a numerical wave tank of length of 25 m, width of 26.5 m and depth of 0.75 m. Although the 3D wave basin of OSU is 48.8 m long, a reduced length is incorporated in the models to yield faster runs representing the three dimensional propagation of the waves. The mesh has 250 elements along the length, 58 elements along the width (thus 2 elements per wave board) and 5 elements along the depth. Total number of nodes is 33338 and number of elements is 32080. Two cases are studied with each model- with wave heights of 0.3 m and 0.4 m. The celerity is computed to be 3.209 m/sec and 3.359 m/sec, respectively for these cases. Focus is located at X=21.0m and Y=0. Comments below the individual figures provide the detailed observations corresponding to the results in the figures.

5.1 Model-1: The evolution and propagation of the wave is shown in Figure 3. Results obtained with wave height of 0.3 m are shown in Figures 6 to 9. This run clocked 45.3 hrs with 4 CPUs for 636 time steps and 10.9 sec termination time. Results obtained with wave height of 0.4 m are shown in Figures 10 to 13. This run took 35.5 hrs with 4 CPUs for 490 time steps and 8.3 sec termination time. The numerical wave profiles (at both wavemaker and wave gage locations) are found to match very well with the experimental results for both 0.3 m and 0.4 m wave heights, although the experimental wave heights at location 2 are taller (Fig. 11) than numerical values for the 0.4 m case. The wave absorption capabilities (due to the absorbing piston and absorbing beach) are working well so no reflected waves can be observed in the wave gage records.

5.2 Model-2: The evolution and propagation of the wave is shown in Figure 4. Results obtained with wave height of 0.3 m are shown in Figures 14 to 17. This run had a clock time of 35.5 hrs with 4 CPUs for 385 time steps and 5.3 sec termination time. Results obtained with a wave height of 0.4 m are shown in

Figures 18 to 21. This run clocked 35 hrs with 4 CPUs for 376 time steps and 5.1 sec termination time. Both the runs went slower than the rest of the cases reported in this paper because of partial sharing of CPUs with other users. Only numerical results are shown since the experimental facility has implemented only model-1 at the present time. The wave height is found to increase during the propagation along the center line in the X-direction of the tank. Both of these runs resulted in early termination due to mesh distortion that occurred in the proximity of the edge of the wavemaker (also noticeable as the oscillations of the corresponding wave profile in Figs. 14 and 18). The wave profiles are developed at the wavemaker sooner than for the case of model-1 and at the same time (since there is no time-lag) for all locations along the wavemaker. Thus, the wave is found to propagate faster than for the case of model-1 as reported at the wave gages.

5.3 Model-3: The evolution and propagation of the wave is shown in Figure 5. Results obtained with wave height of 0.3 m are shown in Figures 22 to 25. This run took 32 hrs with 4 CPUs for 450 time steps and 7.7 sec termination time. Results obtained with a wave height of 0.4 m are shown in Figures 26 to 29. This run clocked 43.7 hrs with 4 CPUs for 621 time steps and 10.5 sec termination time. Only numerical results are shown since the experimental facility has implemented only model-1 at the present time. The wave absorption capabilities (absorbing piston and absorbing beach) are working well so no reflected waves can be seen in the wave gage records. An interesting observation is that in contrast to model-1, the focused wave has not increased in wave height during its propagation. At the same time, it didn't drop the height much during its travel as noticeable in the behavior of solitary waves. Also, this wave traveled faster than model-1 owing to the faster generation at the wavemaker (since there is no time-lag). The pressures computed are found to be smaller in magnitude when compared with models-1 and 2.

6 Concluding remarks

Focused solitary waves exhibit real 3D wave phenomena and are of importance to study the directed wave impact on coastal infrastructure (structures such as bridges, levees, etc) and in laboratory experiments. This paper presented the theoretical background including algorithms, demonstrated the generation of three dimensional focused solitary waves using the 3D-NWT code and validated the results using large-scale experiments. The focused wave capability takes advantage of the newly added piston wavemaker and the high-performance computing (parallelization) features that are also recently incorporated by the authors. The actuators of the piston wavemaker's paddles are controlled by the user-specified time history of motion.

Out of the three alternatives implemented, model-1 has exhibited generation of a taller wave than the rest of the models and is the one implemented in the 3D wave basin at OSU. The comparison of results of this model with the experiments has been excellent. This indicates that the FNPF discretized by BEM and coupled with fast solution methods such as FMA provides results very close to the real physics even for three-dimensional waves. Model-2 has been relatively less successful because of the mesh distortion causing an early termination. The possible fixes in a future work would require the use of much finer meshes and some local refinement of meshes as well. The initial curvature of the wavemaker is very high for some cases of the location of the focus, which makes it impossible to implement in the existing experimental wave basin with restricted motion of wavemaker. As such this model may serve theoretical purposes in many situations. Model-3 is an interesting study- although the part of the wave closer to the center line of the tank along the X-direction is created shorter (but matches the specified wave height) than at the edges, it picked up the height from the edge parts as the propagation proceeded further down toward the absorbing beach and wave gages indicate that all the parts of the wave travel at about the same speed (Figs. 23 and 27). Models 2 and 3 are to be implemented in the 3D experimental wave basin to facilitate comparison with results.

Performance studies of a parallelized 3D-NWT code indicate very good speedups and the applicability of the code to solve large-scale problems on high-performance computing (HPC) platforms. This has been demonstrated by the models used in the paper which have taken 1-2 days of clock time to run on 4 processors which is a reasonable turnaround time for CFD problems (that are quite computational-intensive in nature). It is for the first time (to the authors' knowledge) that a BEM application code has been run to propagate 3D nonlinear waves with model sizes of 26-149K, thus indicating significant improvements in the use of BEM and its capabilities to solve practical engineering problems.

7 Figures and tables

Table 1:- Model of 3D wave basin: # of nodes=63150, # of tree levels in FMA=nlevels=8; 5 time steps

Number of processors (ncpu)	Execution time in sec	Speedup (T1/Tp)
1	9267	1.00
2	5963	1.55
3	4790	1.94
4	4145	2.24
5	3792	2.44
6	3376	2.75
7	3266	2.84
8	3176	2.92

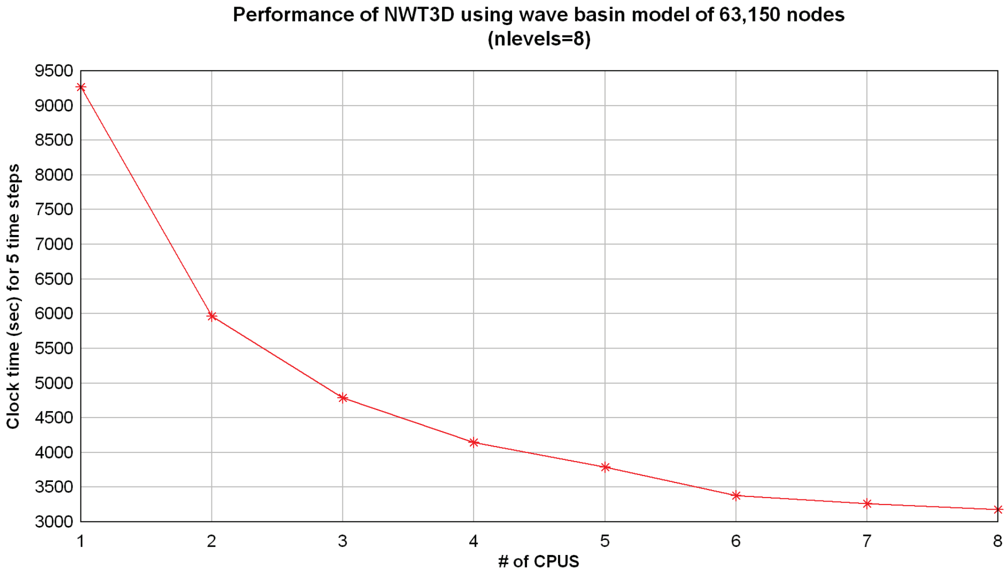


Figure 1: Performance (#of CPUs versus clock time in sec) using wave basin model

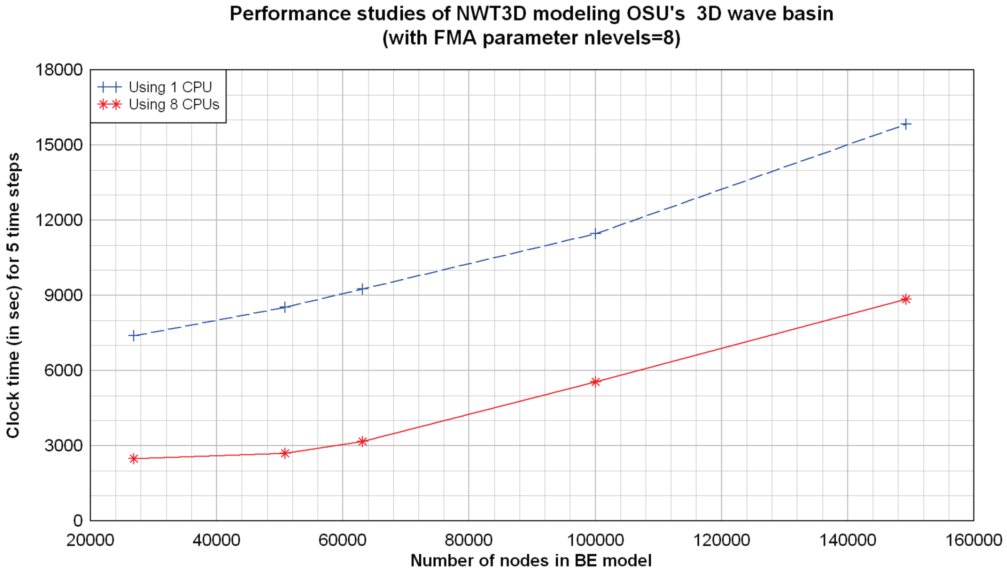


Figure 2: Performance (number of nodes versus clock time) using several models with nlevels=8; solid line (-) is with 8 processors; dashed line (--) is with single processor

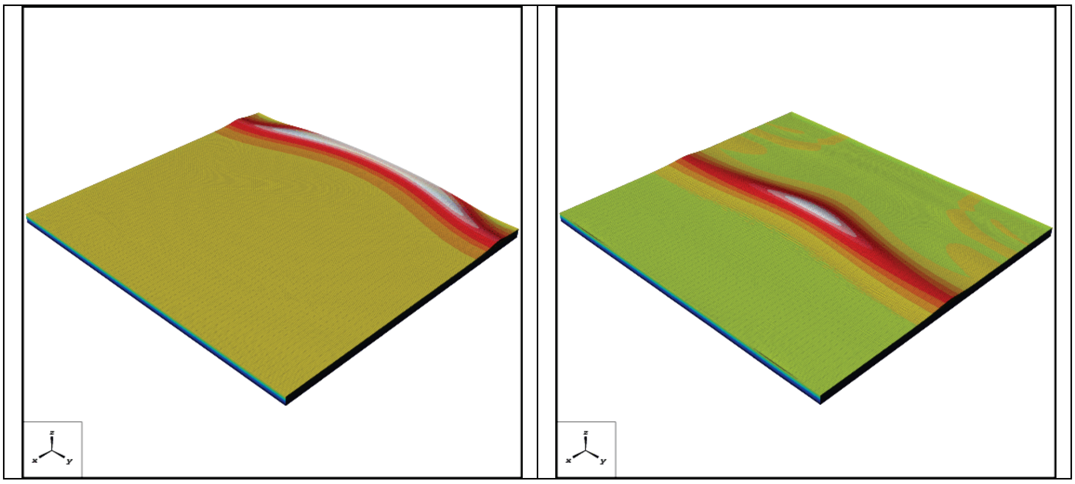


Figure 3: Evolution and propagation of the focused solitary wave using model-1

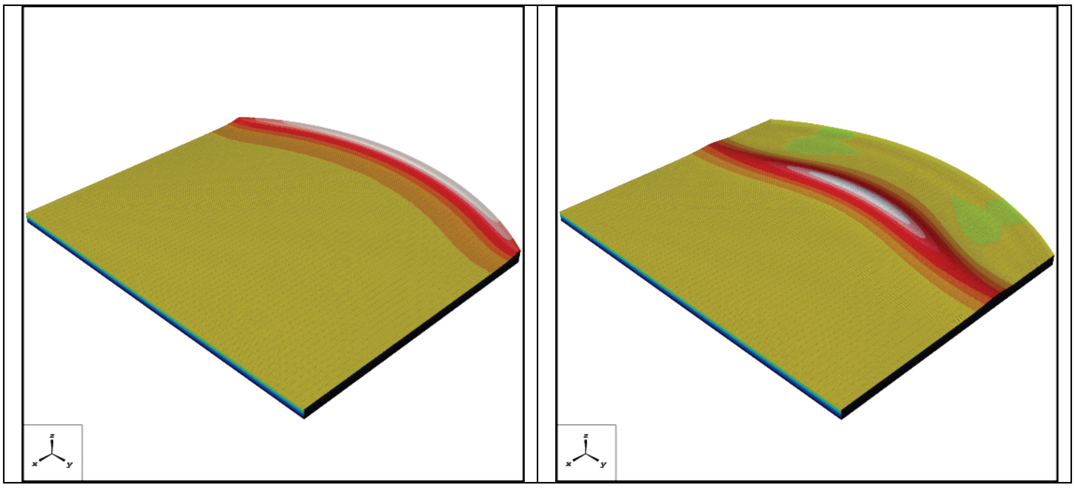


Figure 4: Evolution and propagation of the focused solitary wave using model-2

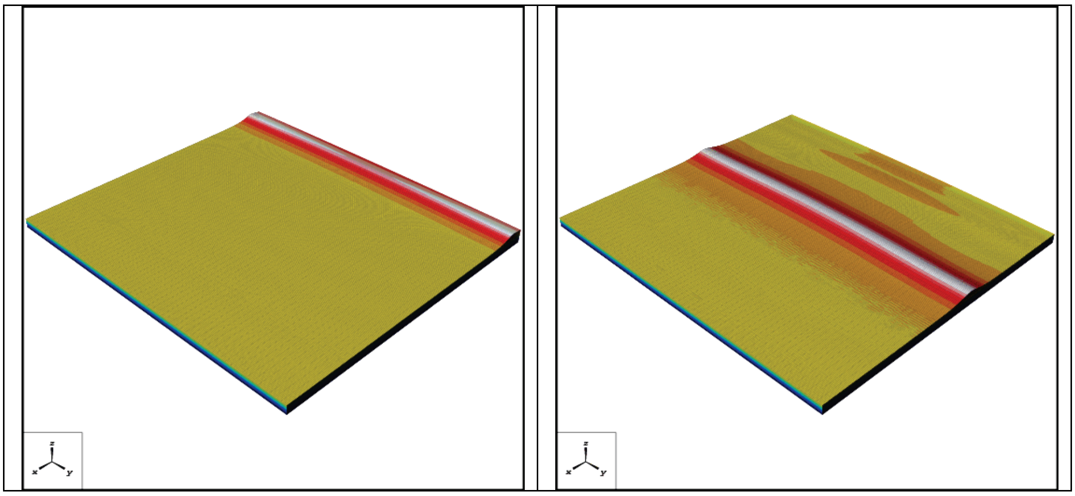


Figure 5: Evolution and propagation of the focused solitary wave using model-3

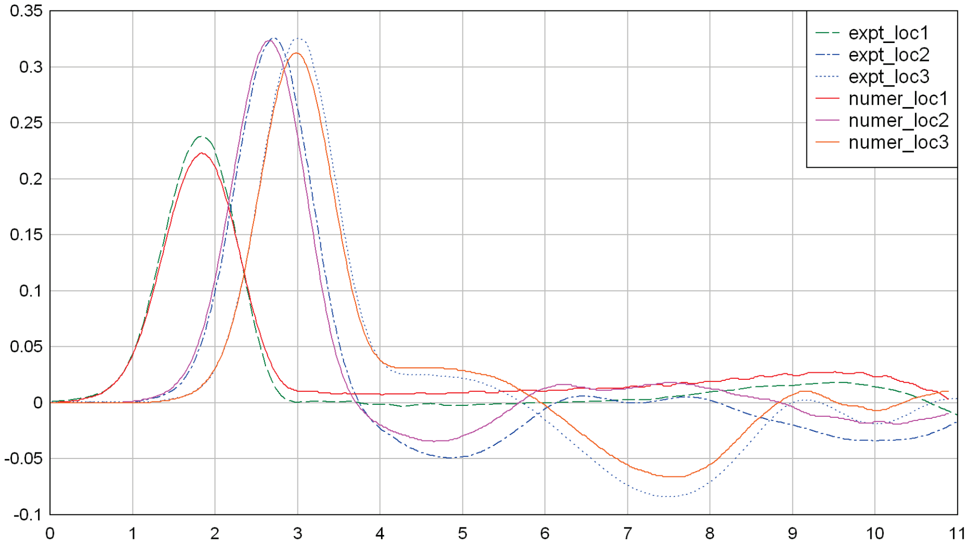


Figure 6: Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of focused solitary wave using model-1 at three locations (starting at the edge first and ending at the middle) along the wavemaker for wave height=0.3 m; Wave profile (that showed up first in time) corresponds to the edge of the wavemaker and the last wave profile corresponds to the middle of the wavemaker. Results indicate a very good agreement.

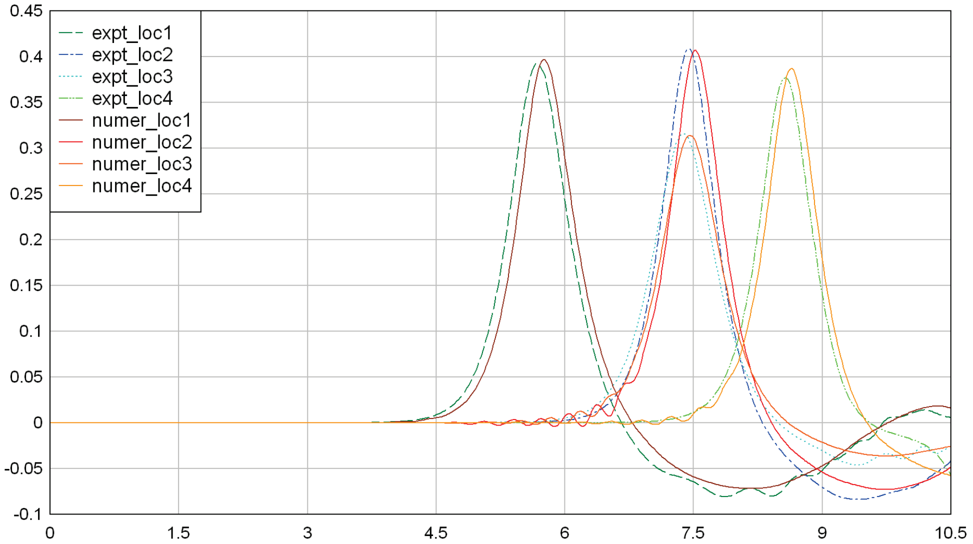


Figure 7: Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of focused solitary wave using model-1 at four wave gages at (x,y) m =(8.8,0), (14.9,0), (14.9,-3.99) and (18.7,0) for wave height=0.3 m; the numerical waves are found to lag slightly behind the experimental waves, but the results show a very good agreement; the wave height at location 3 is the shortest of all four locations because it is located off the center line of the numerical wave tank.

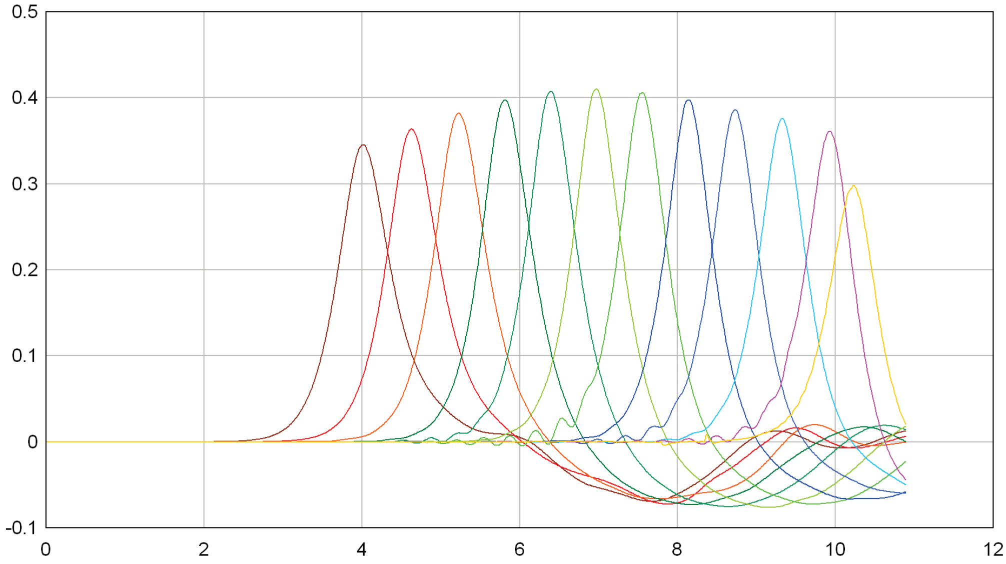


Figure 8: Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-1 at wave gages along the length of the tank at y=0 for wave height=0.3 m; the wave absorption capability near the end of the tank is working well to absorb the wave (wave gage records are free from reflected waves); the focused wave at the middle reached a maximum.

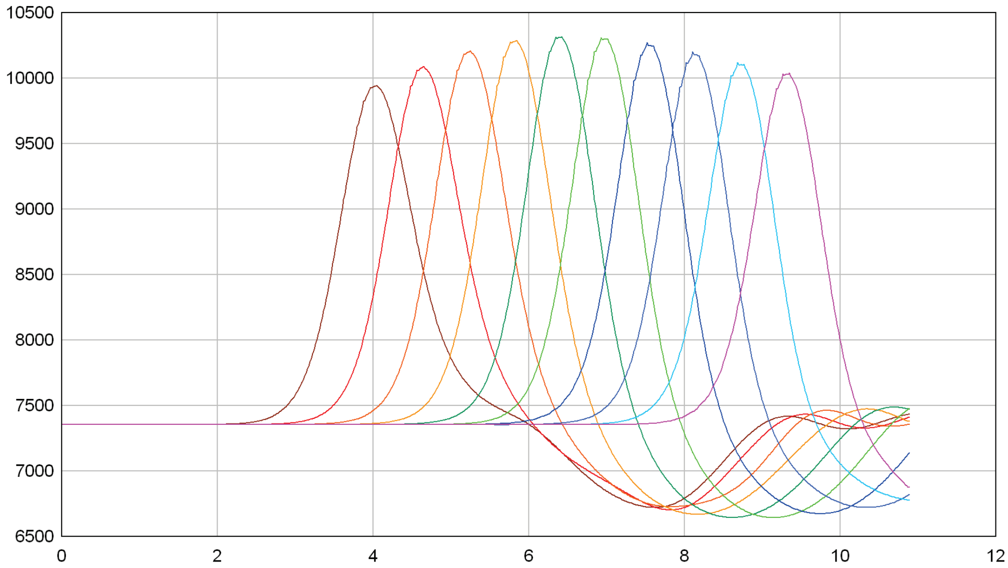


Figure 9: Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of focused solitary wave using model-1 at pressure gages located at the bottom surface along the length of the tank at y=0 for wave height=0.3 m; static pressure =0.75*9810=7357.5 N/m^2

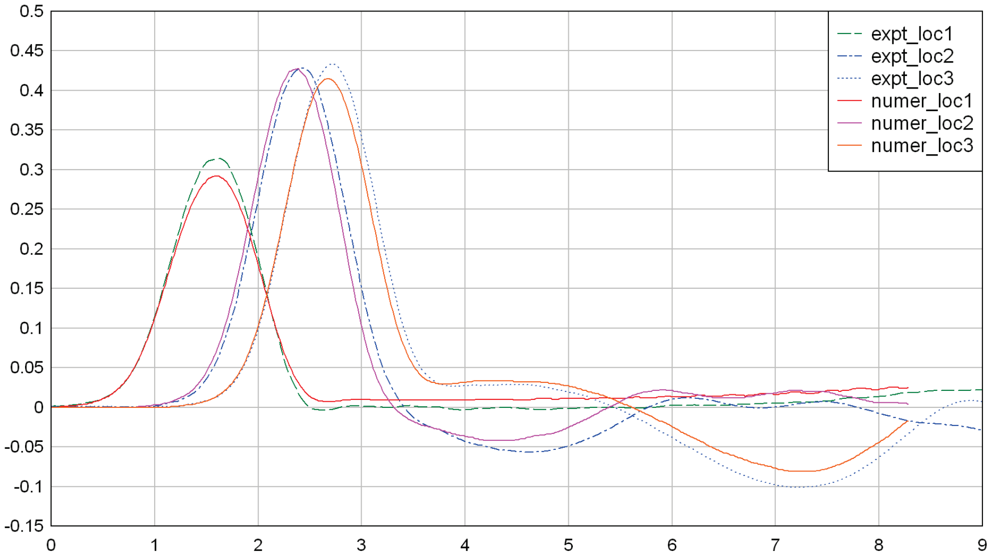


Figure 10: Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of focused solitary wave using model-1 at three locations (starting at the edge first and ending at the middle) along the wavemaker for wave height=0.4 m; Wave profile (that showed up first in time) corresponds to the edge of the wavemaker and the last wave profile corresponds to the middle of the wavemaker. Results indicate a very good agreement.

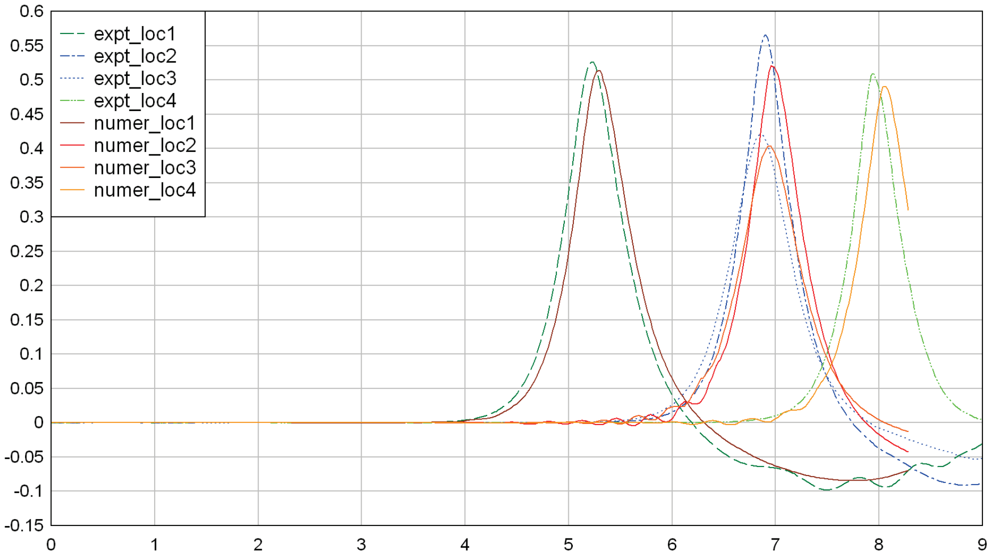


Figure 11: Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of focused solitary wave using model-1 at four wave gages at (x,y) m =(8.8,0), (14.9,0), (14.9,-3.99) and (18.7,0) for wave height=0.4 m; the numerical waves are found to lag slightly behind the experimental waves, but the results show good agreement; the wave height at location 3 is the shortest of all four locations because it is located off the center line of the numerical wave tank.

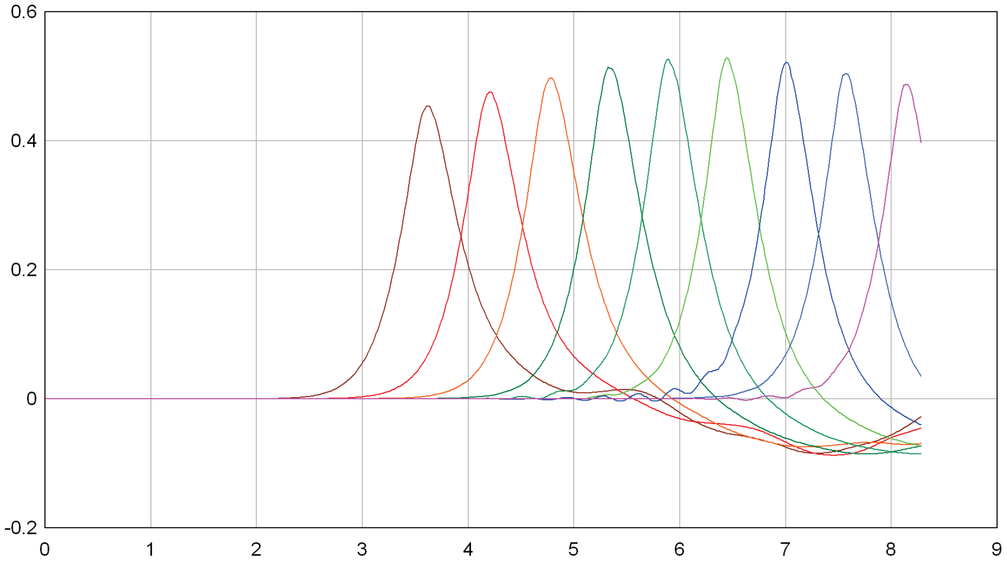


Figure 12: Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-1 at wave gages along the length of the tank at y=0 for wave height=0.4 m; the wave absorption capability near the end of the tank is working well to absorb the wave (wave gage records are free from reflected waves); the focused wave at the middle reached a maximum.

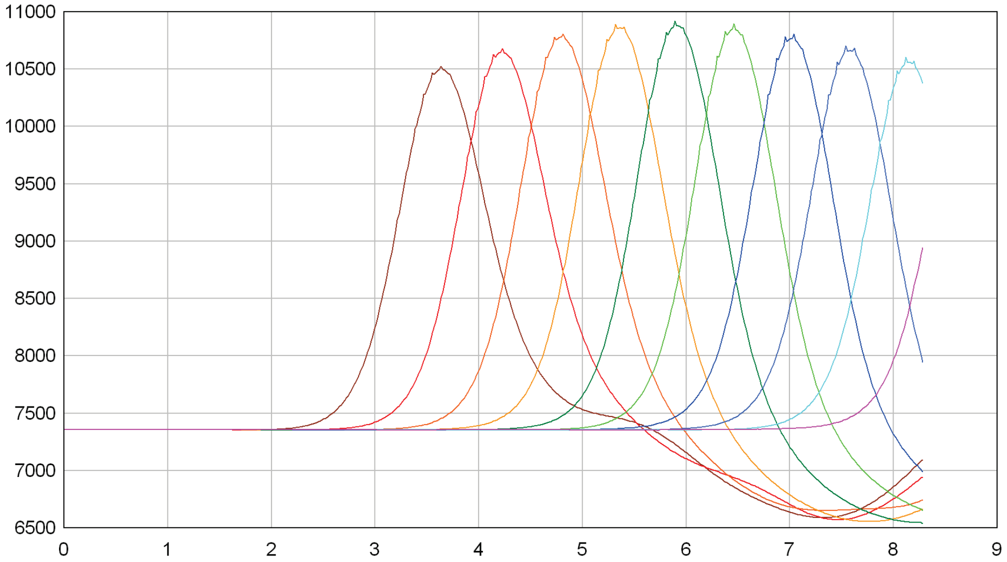


Figure 13: Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of focused solitary wave using model-1 at pressure gages located at the bottom surface along the length of the tank at y=0 for wave height=0.4 m; static pressure =0.75*9810=7357.5 N/m^2

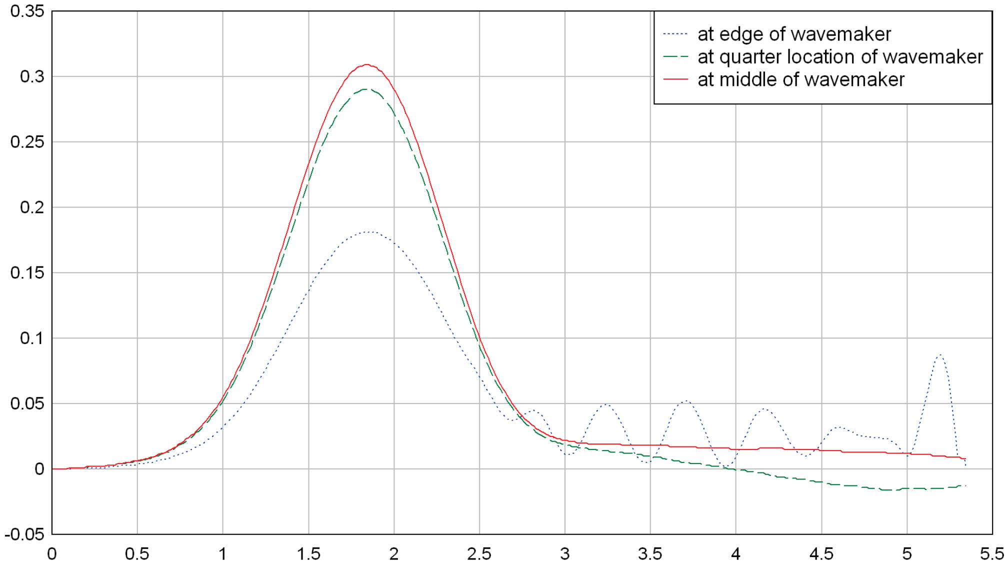


Figure 14: Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-2 at three locations (starting at the edge and ending at the middle) along the wavemaker for wave height=0.3 m; Wave profile (that is the shortest) corresponds to the edge of the wavemaker and the tallest wave profile corresponds to the middle of the wavemaker.

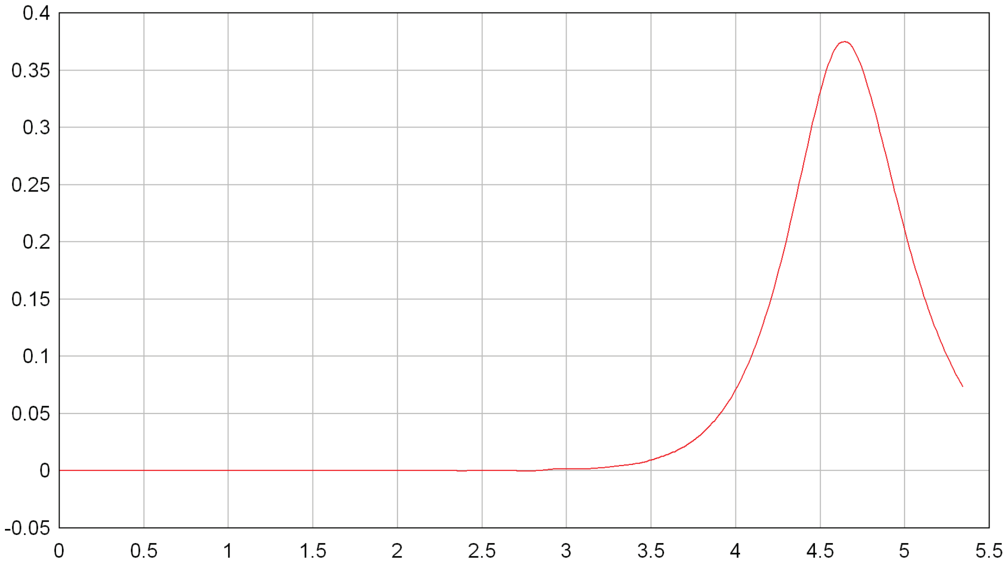


Figure 15: Time (in sec on horizontal axis) versus numerical wave profile (in m on vertical axis) of focused solitary wave using model-2 at a wave gage at (x,y) m =(8.8,0) for wave height=0.3 m.

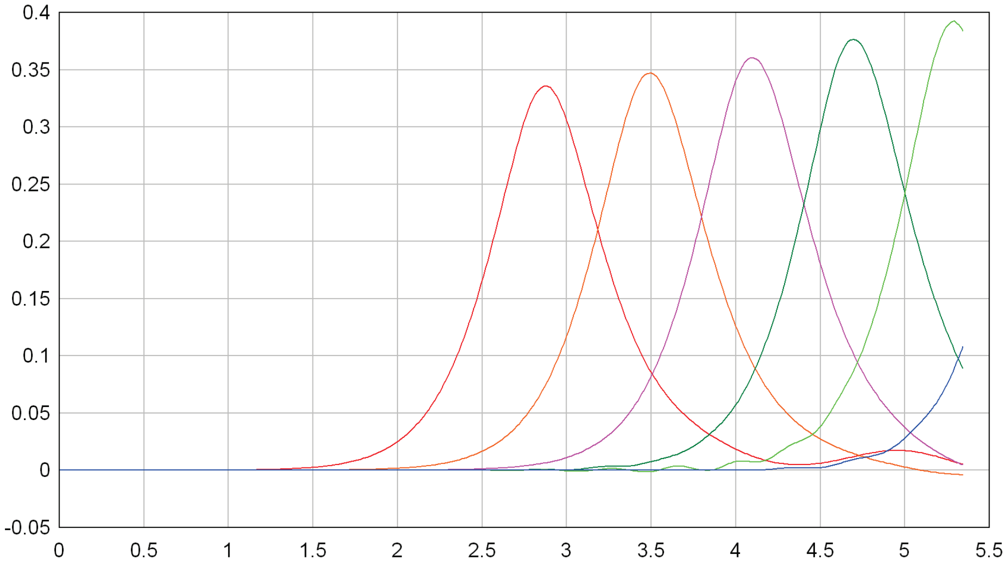


Figure 16: Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-2 at wave gages along the length of the tank at y=0 for wave height=0.3 m

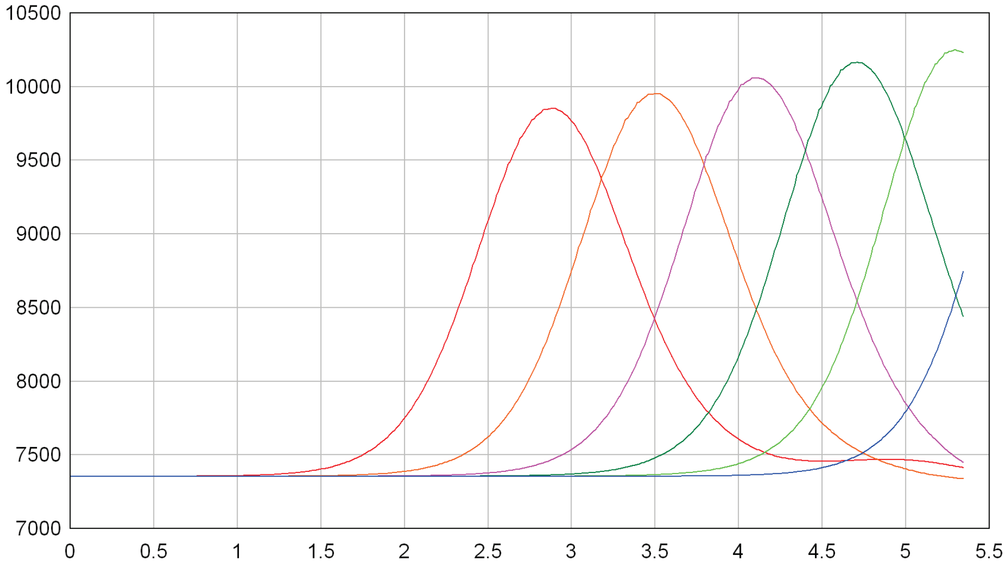


Figure 17: Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of focused solitary wave using model-2 at pressure gages located at the bottom surface along the length of the tank at y=0 for wave height=0.3m; static pressure =0.75*9810=7357.5 N/m^2

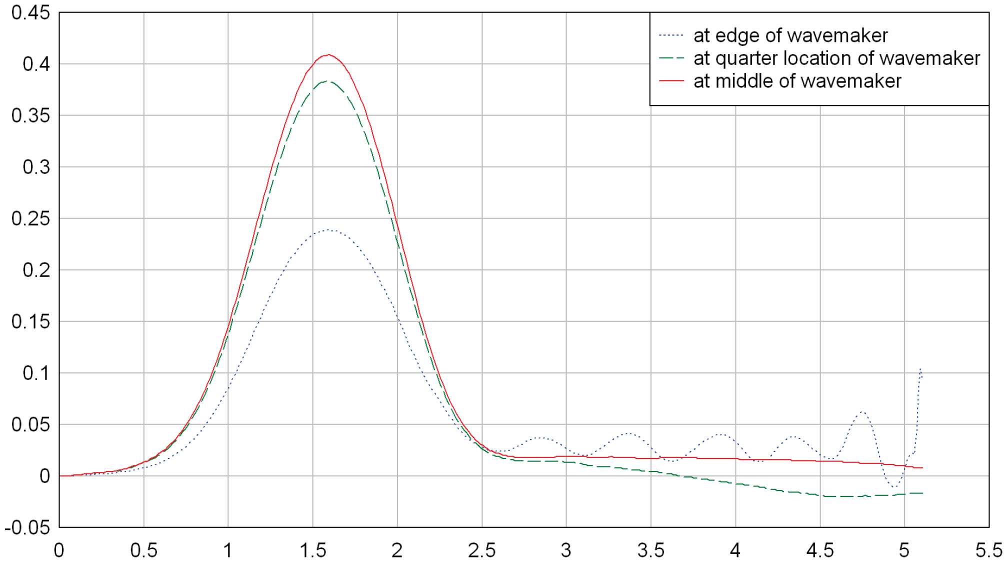


Figure 18: Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-2 at three locations (starting at the edge and ending at the middle) along the wavemaker for wave height=0.4 m; Wave profile (that is the shortest) corresponds to the edge of the wavemaker and the tallest wave profile corresponds to the middle of the wavemaker.

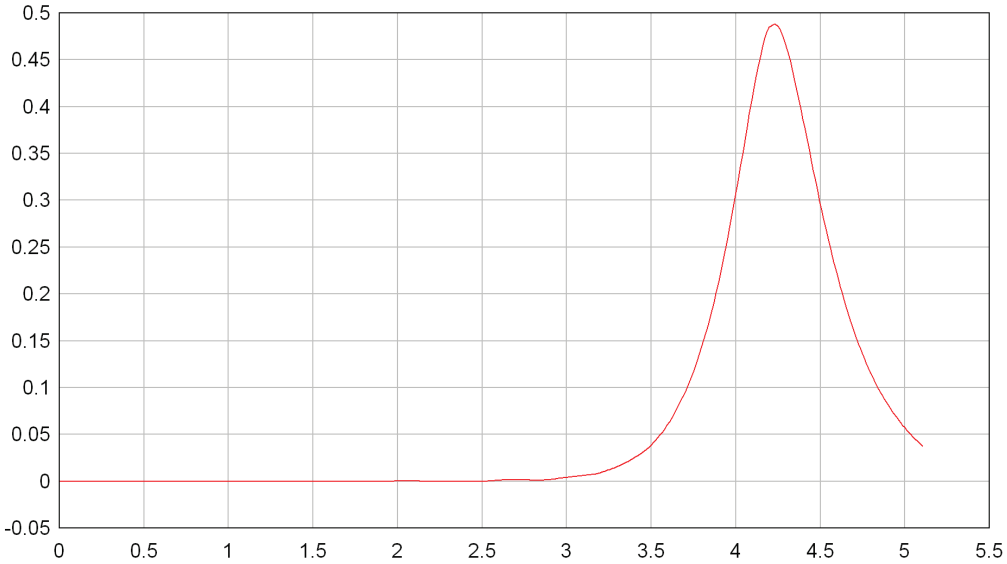


Figure 19: Time (in sec on horizontal axis) versus numerical wave profile (in m on vertical axis) of focused solitary wave using model-2 at a wave gage at (x,y) m =(8.8,0) for wave height=0.4 m.

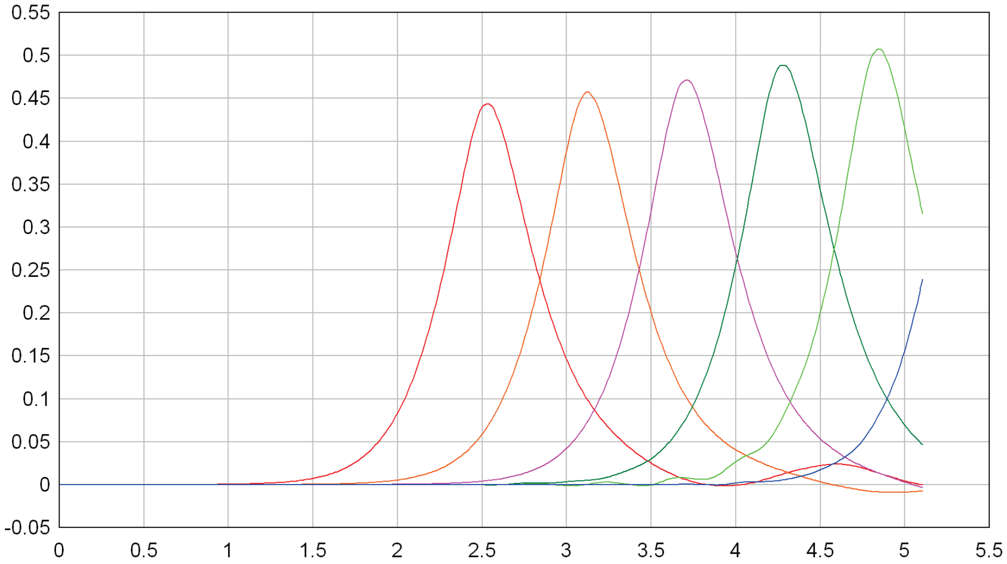


Figure 20: Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-2 at wave gages along the length of the tank at y=0 for wave height=0.4 m

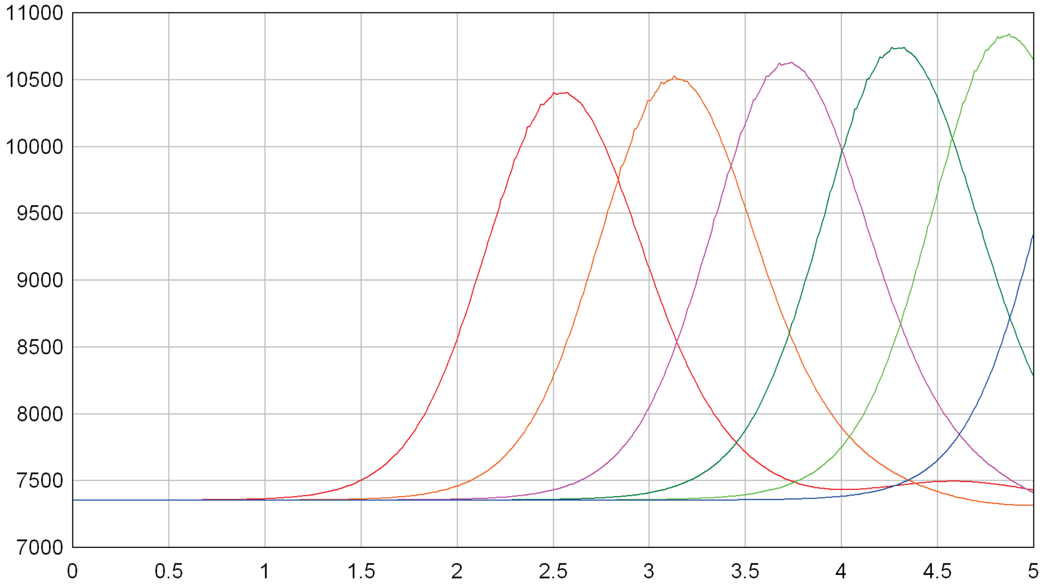


Figure 21: Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of focused solitary wave using model-2 at pressure gages located at the bottom surface along the length of the tank at y=0 for wave height=0.4 m; static pressure =0.75*9810=7357.5 N/m^2

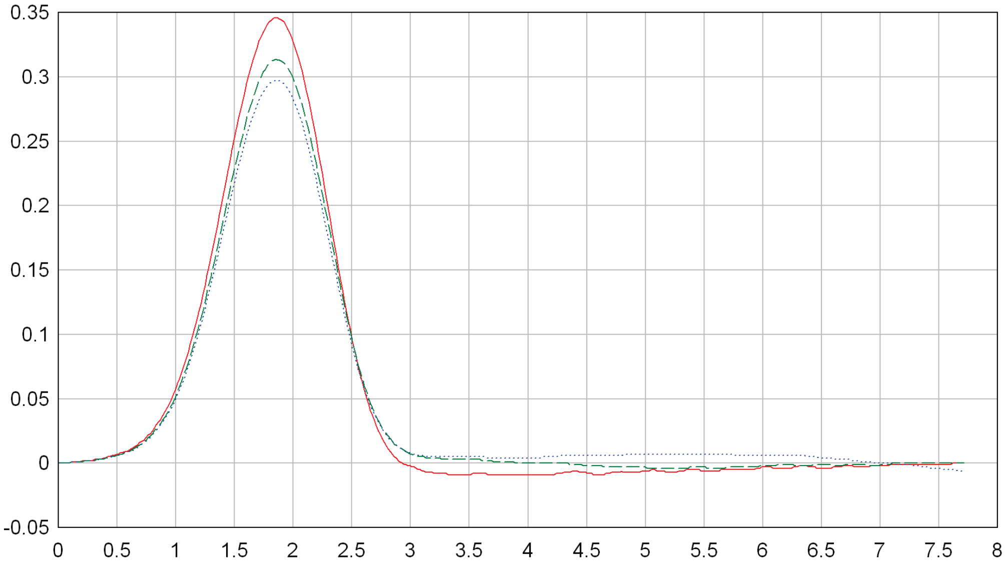


Figure 22: Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-3 at three locations (starting at the edge and ending at the middle) along the wavemaker for wave height=0.3 m; Wave profile (that is the tallest) corresponds to the edge of the wavemaker and the shortest wave profile corresponds to the middle of the wavemaker.

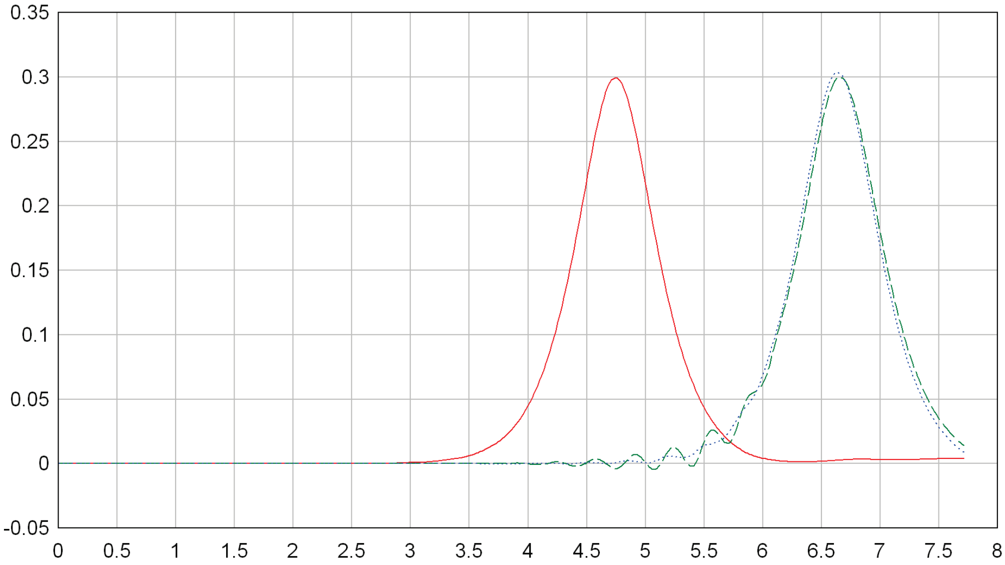


Figure 23: Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-3 at three wave gages at (x,y) m =(8.8,0), (14.9,0), (14.9,-3.99) for wave height=0.3 m. The wave gage 3, which is off the center line also reports the wave height of 0.3 m- same as the rest of the gages.

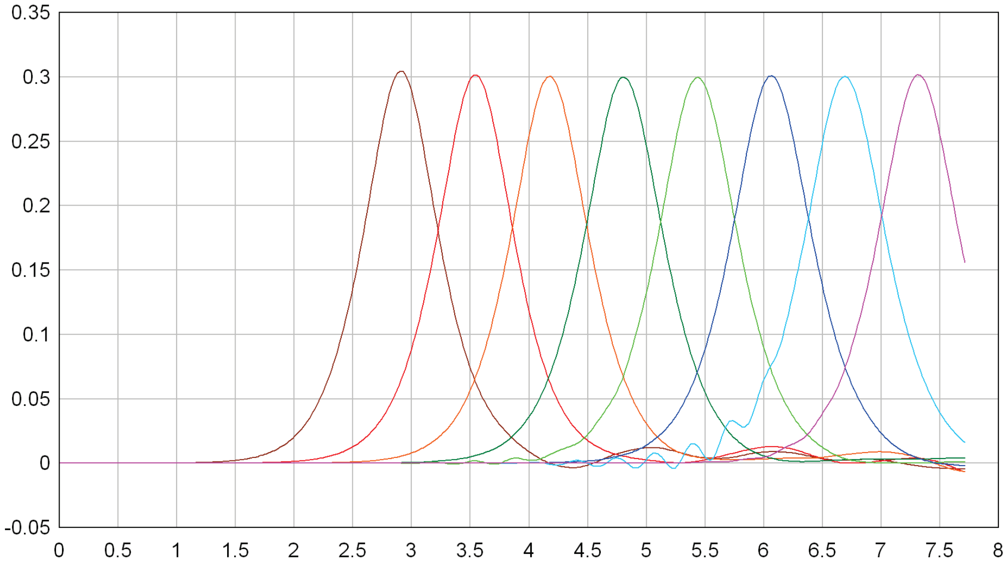


Figure 24: Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-3 at wave gages along the length of the tank at y=0 for wave height=0.3 m; the wave absorption capability near the end of the tank is working well to absorb the wave (wave gage records are free from reflected waves). Wave height remained close to 0.3 m throughout.

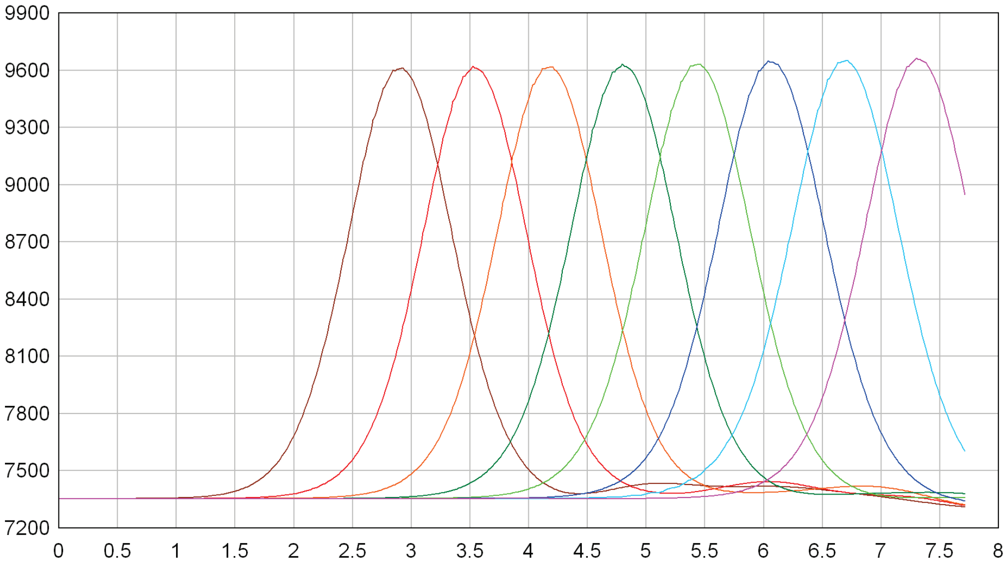


Figure 25: Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of focused solitary wave using model-3 at pressure gages located at the bottom surface along the length of the tank at y=0 for wave height=0.3m; static pressure =0.75*9810=7357.5 N/m^2

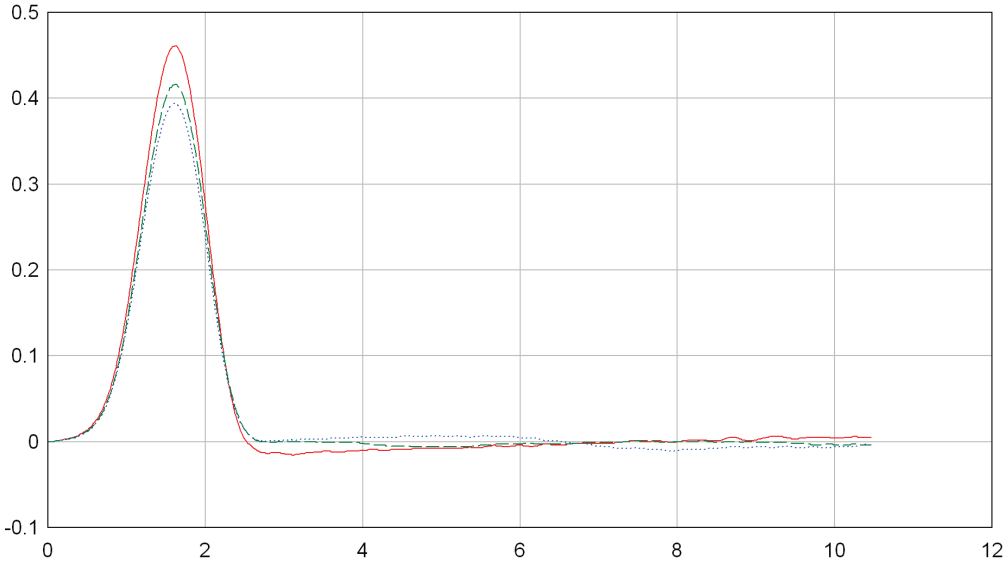


Figure 26: Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-3 at three locations (starting at the edge and ending at the middle) along the wavemaker for wave height=0.4 m; Wave profile (that is the tallest) corresponds to the edge of the wavemaker and the shortest wave profile corresponds to the middle of the wavemaker.

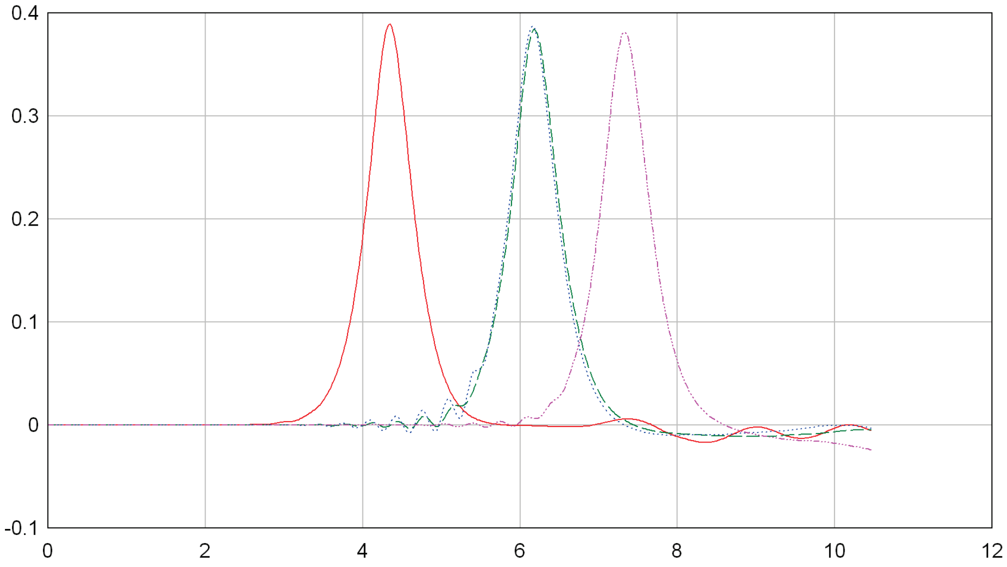


Figure 27: Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-3 at four wave gages at (x,y) m =(8.8,0), (14.9,0), (14.9,-3.99) and (18.7,0) for wave height=0.4 m. The wave gage 3, which is off the center line also reports the wave height of 0.4 m- same as the rest of the gages.

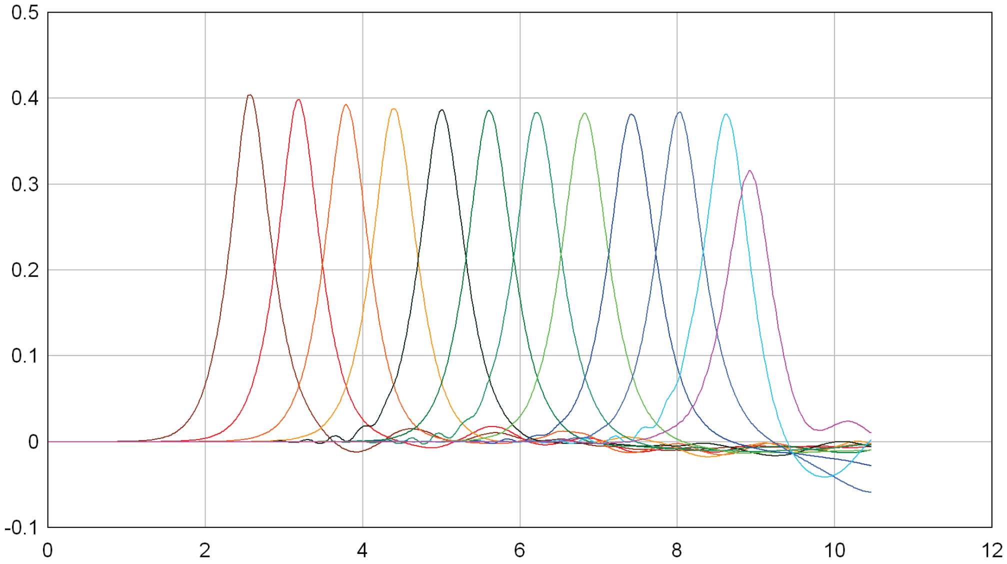


Figure 28: Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of focused solitary wave using model-3 at wave gages along the length of the tank at y=0 for wave height=0.4 m; the wave absorption capability near the end of the tank is working well to absorb the wave (wave gage records are free from reflected waves). Wave height came down slightly below 0.4 m near the middle of the tank and then remained almost uniform.

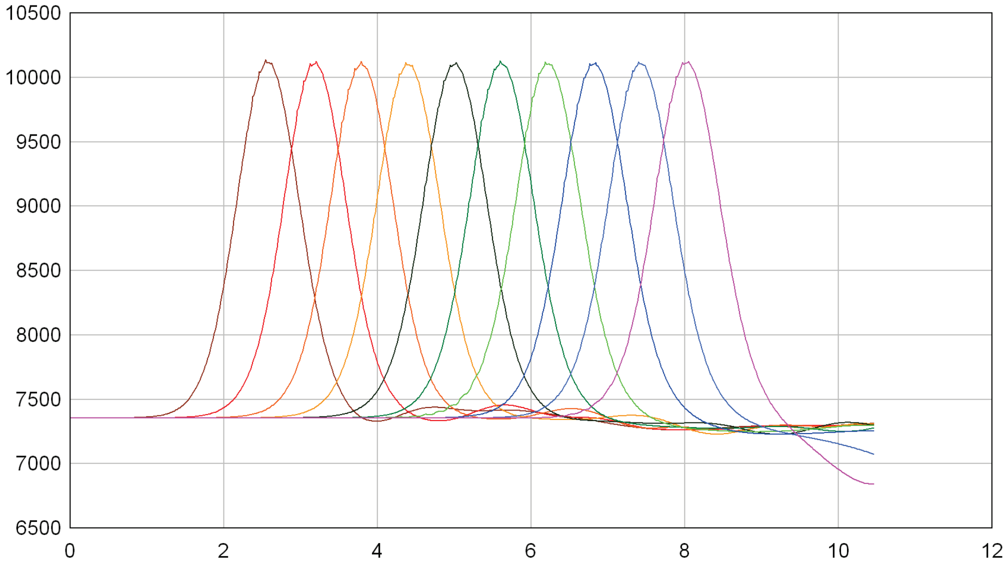


Figure 29: Time (in sec on horizontal axis) versus numerical pressure profiles (in N/m^2 on vertical axis) of focused solitary wave using model-3 at pressure gages located at the bottom surface along the length of the tank at y=0 for wave height=0.4m; static pressure =0.75*9810=7357.5 N/m^2

Acknowledgement

The authors are thankful to the Office of Naval Research (ONR) for their generous financial support that made this work possible.

References

[1] Grilli, S.T., Guyenne, P., and Dias, F., 2001, "A fully nonlinear model for three-dimensional overturning waves over arbitrary bottom," International Journal for Numerical Methods in Fluids, **35**(7), pp. 829-867.

[2] Brandini, C., and Grilli, S.T., 2001, "Modeling of freak wave generation in a 3D-NWT," In Proc. 11th Offshore and Polar Engng. Conf. (ISOPE01, Stavanger, Norway, June 2001), Vol III, pp. 124-131.

[3] Grilli, S.T., Vogelmann, S., and Watts, P., 2002, "Development of a 3D numerical wave tank for modeling tsunami generation by underwater landslides," Engineering Analysis with Boundary Elements, **26**(4), pp. 301-313.

[4] Fochesato, C., Grilli, S.T., and Guyenne, P., 2005, "Note on non-orthogonality of local curvilinear coordinates in a three-dimensional boundary element method," International Journal for Numerical Methods in Fluids, **48**, pp. 305-324.

[5] Fochesato, C., and Dias, F., 2006, "A fast method for nonlinear three-dimensional free-surface waves," Proceedings of the Royal Society, London, **A 462**, pp. 2715–2735.

[6] Guyenne, P., and Grilli, S.T., 2006, "Numerical study of three-dimensional overturning waves in shallow water," Journal of Fluid Mechanics, **547**, pp. 361-388.

[7] Fochesato, C., Grilli, S.T., and Dias F., 2007, "Numerical modeling of extreme rogue waves generated by directional energy focusing," Wave Motion, **44**, pp. 395-416.

[8] Sung, H.G., and Grilli, S.T., 2008, "BEM Computations of 3D Fully Nonlinear Free Surface Flows Caused by Advancing Surface Disturbances," International Journal of Offshore and Polar Engineering, **18**(4), pp. 292-301.

[9] Grilli, S.T., Dias, F., Guyenne, P., Fochesato, C., and F. Enet, 2010, "Progress In Fully Nonlinear Potential Flow Modeling Of 3D Extreme Ocean Waves," Chapter 3 in Advances in Numerical Simulation of Nonlinear Water Waves, Vol. 11 in Series in Advances in Coastal and Ocean Engineering, World Scientific Publishing Co. Pte. Ltd., pp. 75-128.

[10] Banerjee, P.K., and Butterfield, R., 1981, "Boundary element methods in engineering science," McGraw-Hill Book Co (UK) Ltd.

[11] Brebbia, C.A., Telles, J.C.F., and Wrobel, L.C., 1984, “Boundary element techniques: theory and applications in engineering,” Springer-Verlag, Berlin.

[12] Hartmann, F., 1989, “Introduction to Boundary elements,” Springer-Verlag.

[13] Brebbia, C.A., and Dominguez, J., 1989, “Boundary elements: an introductory course,” Computational Mechanics Publications, Southampton, Boston.

[14] Banerjee, P.K., 1994, “The boundary element methods in engineering,” McGraw-Hill Book Company.

[15] Power, H., and Wrobel, L.C., 1995, “Boundary integral methods in fluid mechanics,” Computational Mechanics Publications, Southampton, Boston.

[16] Wrobel, L.C., 2002, “The boundary element method- Applications in thermo-fluids and acoustics,” Volume 1, John Wiley & Sons, Ltd.

[17] Aliabadi, M.H., 2002, “The boundary element method- Applications in solid and structures,” Volume 2, John Wiley & Sons, Ltd.

[18] Bonnet, M., 1995, “Boundary integral equation methods for solids and fluids,” John Wiley & Sons Ltd.

[19] Chen, G., and Zhou, J., 1992, “Boundary Element Methods,” Academic Press.

[20] Hsiao, G.C., and Wendland, W.L., 2008, “Boundary integral equations,” Springer-Verlag, Berlin, Heidelberg.

[21] Nishimura, N., 2002, “Fast multipole accelerated boundary integral equation methods,” Applied Mechanics Review, 55(4), pp. 299-324.

[22] Ying, L., Biros, G., Zorin, D., and Langston, H., 2003, “A new parallel kernel-independent fast multipole method,” Proceedings of the 2003 ACM/IEEE Conference on Supercomputing, 196(2), pp. 591-626.

[23] Ying, L., Biros, G., and Zorin, D., 2004, “A kernel-independent adaptive fast multipole algorithm in two and three dimensions,” Journal of Computational Physics, 196(2), pp. 591-626.

[24] Margonari, M., and Bonnet, M., 2005, “Fast multipole method applied to elastostatic BEM-FEM coupling,” Computers and Structures, 83, pp. 700-717.

[25] Chaillat, S., Bonnet, M., and Semblat, J-F., 2008, “A multi-level fast multipole BEM for 3-D elastodynamics in the frequency domain,” Computer Methods in Applied Mechanics and Engineering, 197(49-50), pp. 4233-4249.

[26] Liu, Y., 2009, “Fast multipole boundary element method: theory and applications in engineering,” Cambridge University Press.

[27] Dommermuth, D.G., Yue, D.K.P., Lin, W.M., and Rapp, R.J., 1988, “Deep-water plunging breakers: a comparison between potential theory and experiments,” Journal of Fluid Mechanics, 189, pp. 423-442.

[28] Dold, J.W., and Peregrine, D.H., 1986, “An efficient boundary-integral method for steep unsteady water waves,” Numerical Methods for fluid dynamics II, Edited by Morton, K.W., and Baines, M.J., Clarendon Press, Oxford.

[29] Dold, J.W., 1992, “An efficient surface-integral algorithm applied to unsteady gravity waves,” Journal of Computational Physics, 103, pp. 90-115.

[30] Dold, J.W., and Peregrine, D.H., 1984, “Steep unsteady water waves: An efficient computational scheme,” Proceedings of 19th International Conference on Coastal Engineering, Houston, pp. 955-967.

[31] Romate, J.E., 1989, “The numerical simulation of nonlinear gravity waves in three dimensions using a higher order panel method,” Zandbergen, P.J., (advisor), PhD thesis, University of Twente, Netherlands.

[32] Broeze, J., 1993, “Numerical modelling of nonlinear free surface waves with a 3D panel method,” Zandbergen, P.J., (advisor), PhD thesis, University of Twente, Netherlands.

[33] Daalen van, E.F.G., 1993, “Numerical and theoretical studies of water waves and floating bodies,” Zandbergen, P.J., (advisor), PhD thesis, University of Twente, Netherlands.

[34] Berkvens, P.J.F., 1998, “Floating bodies interacting with water waves- development of a time-domain panel method,” Zandbergen, P.J., (advisor), PhD thesis, University of Twente, Netherlands.

[35] Liu, P.L.F., and Hsu, H.W., 1992, “Applications of boundary integral equation methods for two-dimensional non-linear water wave problems,” International Journal for numerical methods in fluids, 15, pp. 1119-1141.

[36] Cooker, M.J., 1990, “A boundary-integral method for water wave motion over irregular beds,” Engineering Analysis with Boundary Elements, 7(4), pp. 205-213.

[37] Lin, W.M., 1984, “Nonlinear motion of the free surface near a moving body,” PhD thesis, Department of Ocean Engineering, Massachusetts Institute of Technology.

[38] Vinayan, V., and Kinnas, S.A., 2007, “A BEM for the propagation of Nonlinear Planar Free-surface Waves,” Electronic Journal of Boundary Elements, 5(1), pp. 17-40.

[39] Colicchio, G., Greco, M., and Faltinsen, O.M., 2006, “A BEM-level set domain-decomposition strategy for non-linear and fragmented interfacial flows,” International Journal for Numerical Methods in Engineering, 67, pp. 1385-1419.

[40] Chapman, B., et al., 2007, “Using OpenMP: Portable shared memory parallel programming,” MIT Press.

[41] Nimmala, S.B., Yim, S.C., Grilli, S.T., “Performance Studies of a 3D Numerical Wave Tank Model for Large-Scale Wave Basin Experiments,” (to be published).

[42] "Tsunami Wave Basin (TWB) Experiment Notebook," College of Engineering, NACSE, Oregon State University: Tsunami wave impact forces on cylinders (Phase1Exp2: run5, run6, run11, run17 and run18), (http://neespop.wave.oregonstate.edu/nees/ui/index.php?maintab=200&tab=0&proID=21&expID=72).

Performance Studies of a 3D Numerical Wave Tank Model for Large-Scale Wave Basin Experiments

Seshu B. Nimmala and Solomon C. Yim
School of Civil & Construction Engineering, Oregon State University (OSU),
Corvallis, OR 97331, USA

Stephan T. Grilli
Department of Ocean Engineering, University of Rhode Island (URI),
Narragansett, RI 02882, USA

Abstract

This paper presents parallelization efforts and performance studies of an accurate three-dimensional numerical wave tank (3D-NWT) based on fully nonlinear potential flow theory (FNPF) and the boundary element method (BEM). The solution algorithms use the generalized minimal residual method (GMRES) along with the fast multipole method (FMM). With the advent of multi-core processors in computers, it has become imperative for serial codes to be enhanced to take advantage of the new resources. These enhancements are also aimed at making the code applicable to large-scale simulations using high-performance computing (HPC) platforms to complement the 3D wave basin at OSU (or similar facility elsewhere). The discussions in this paper include the alternatives considered for parallelization, the challenges encountered and the insight/experience gained by the authors that would be applicable for similar efforts considered in the case of computational/numerical codes in engineering/physics domains.

1 Introduction

This paper begins with a brief discussion of the background of the 3D-NWT code, the need for its parallelization, and moves on to describe the various options available to do so. There is also a follow-up discussion of the reasons behind the choice made, the challenges encountered in the parallelization effort, the computational performance, the insight and experience gained in the process. Also included are the examples illustrating some of the capabilities of wave generation incorporated recently by the authors. These developments demonstrate the practical applicability of the boundary element method (BEM) along with the fast solution methods, now with parallelization, to solve real-world problems of nonlinear free surface waves (within the limitations of the FNPF).

2 Background

Grilli and co-workers [1-9] have developed an accurate and versatile three-dimensional numerical wave tank based on fully nonlinear potential flow (FNPF) theory. This development is based on higher-order BEMs, combining both free-space Green's functions and a fast multipole algorithm (FMA) in combination with the generalized minimal residual method (GMRES). The numerical complexity is proportional to the problem size. The free-surface time updating is based on a second-order Taylor series expansion in an Eulerian-

Lagrangian form. Wave absorption is accomplished by a combination of free-surface (pressure) damping and a lateral active absorbing wavemaker. This work on the propagation of highly nonlinear waves is summarized in a chapter of a recent book on BEM [10]. The range of the applications that can be solved by this development, however, have been limited by the size of problems that could be solved on a single processor. Also the implementation doesn't have a piston wavemaker to accept the wave motion input which is an important requirement to be able to model the physical wave basin. Further, it will be interesting to be able to implement the 3D waves (e.g., directed/focused) that the researchers like to study. Validation using large-scale laboratory experiments and demonstration of its capabilities is also of significance. This work of Grilli, et al is selected by the first two authors as the basis for the enhancements envisaged as part of their fluid-structure interaction research studies.

This code is based on fully nonlinear potential flow, which uses the Laplace equation as the governing partial differential equation (second-order linear elliptic). The Laplace equation is given by

$$\nabla^2\phi = 0 \tag{1}$$

where $\phi =$velocity potential function. Also, velocity $\vec{u} = \nabla\phi$. Nonlinear flow originates from the presence and computation of the free-surface.

On the free surface, the potential satisfies nonlinear kinematic (KBC) and dynamic (DBC) boundary conditions. The presence of two unknowns (position and potential) on the free surface requires both of these boundary conditions.

The KBC states that the normal velocity of the free surface interface must be equal to the normal fluid velocity at the surface. Following a fluid particle at the free surface, this means that the particle remains at the free surface. The DBC, obtained from Bernoulli's equation, states that the pressure on the free surface equals atmospheric pressure which will be taken to be zero. If $\vec{R}(t)$ is the position vector (x,y,z) of a fluid particle on the free surface, g is the acceleration due to gravity, $\vec{u}$ is the velocity vector (u,v,w), then from KBC and DBC, respectively:

$$\frac{D\vec{R}}{Dt} = \vec{u} = \nabla\phi \tag{2}$$

$$\frac{D\phi}{Dt} = -gz + \frac{1}{2}\nabla\phi.\nabla\phi \tag{3}$$

For fixed boundaries, the no flow condition (zero flux) is specified as $\frac{\partial\phi}{\partial n} = 0$, where n = normal vector to the surface.

In the case of moving boundaries, the wavemaker generates waves with motion and velocity that are specified as BCs:

$$x = x_p; \quad \frac{\partial\phi}{\partial n} = \vec{u}_p.\vec{n}, \tag{4}$$

where $\vec{u}_p$, $\vec{n}$ are the velocity of the wavemaker and its normal, respectively.

The initial free surface boundary condition (at t=0) is given by the still water level (Z=0) and zero potential (Dirichlet BC) and all other surfaces have zero or specified flux (Neumann BC).

Direct BEM is used to solve the Laplace equation with the potential, flux and their time derivatives as the unknowns.

Second-order explicit Taylor series expansions determine the new position and potential on free surface at each new time step.

$$\vec{R}(t+\Delta t)=\vec{R}(t)+\Delta t\frac{D\vec{R}(t)}{Dt}+\frac{(\Delta t)^2}{2}\frac{D^2\vec{R}(t)}{Dt^2}+O\left[(\Delta t)^3\right] \tag{5}$$

$$\phi(\vec{R}(t+\Delta t))=\phi(t)+\Delta t\frac{D\phi(t)}{Dt}+\frac{(\Delta t)^2}{2}\frac{D^2\phi(t)}{Dt^2}+O\left[(\Delta t)^3\right] \tag{6}$$

In the above, zeroth-order coefficients of the Taylor series are given by the geometry and the solution of the BIE at time t. First-order coefficients are evaluated from the BCs (KBC and DBC) of free surface. Second-order coefficients are given by:

$$\frac{D^2\vec{R}}{Dt^2}=\frac{D\vec{u}}{Dt}=\frac{\partial\vec{u}}{\partial t}+\vec{u}.\nabla\vec{u}=\nabla\frac{\partial\phi}{\partial t}+\nabla\phi.\nabla(\nabla\phi) \tag{7}$$

$$\frac{D^2\phi}{Dt^2}=-g\frac{Dz}{Dt}+\frac{1}{2}\frac{D(\nabla\phi.\nabla\phi)}{Dt}=-gw+\frac{1}{2}\frac{D(\vec{u}.\vec{u})}{Dt}=-gw+\vec{u}.\frac{D\vec{u}}{Dt}, \tag{8}$$

where w= vertical component of velocity

The authors have recently implemented a variety of wave generation options and enhanced the 3D-NWT code with the incorporation of a piston wavemaker and its motion input [11]. The waves that can be generated include solitary, Cnoidal and Airy.

Following are the mathematical formulations of the solitary wave. The reader is referred to [12,13] for further details on this topic.

Let H = wave height and h = water depth

Total stroke, $S_s = \sqrt{\dfrac{16Hh}{3}}$; Celerity, $C = \sqrt{g(h+H)}$

Wave number,

$$k = \sqrt{\frac{3H}{4h^3}} \qquad (9)$$

Waveboard time history,

$$X_o(t) = \frac{H}{kh}\tanh\left[k(Ct - X_o)\right] \qquad (10)$$

Waveboard displacement,

$$X_o(at\, t = \pm\infty) = \pm\frac{H}{kh} = \pm\sqrt{\frac{4Hh}{3}} \qquad (11)$$

Let $X_l = 0.999\dfrac{H}{kh}$.

Time taken by board to travel (from $X_o = 0$ to $+X_l$)=

$$t_f = \frac{1}{kC}(3.8 + \frac{H}{h}) \qquad (12)$$

Total stroke duration, $t_s = 2t_f$

Waveboard starts at an initial resting position $X_o(-t_f) = -\sqrt{\dfrac{4Hh}{3}}$ and a final resting position $X_o(+t_f) = +\sqrt{\dfrac{4Hh}{3}}$ for practical purposes.

The algorithms of the Cnoidal and Airy waves are in [11].

3 Parallelization options

Solving real-world/practical engineering analysis problems often requires creation and use of large scale numerical models (which can have several thousands or millions of nodes and elements) to achieve improved accuracy and an access to the high-performance computing (HPC) platforms. In this context, it is also imperative to have a state-of-the-art computational mechanics software capable of taking advantage of the HPC resources, so that the scientist/engineer can get interesting results within a reasonable time frame (usually within 12-24 hrs of turnaround). These requirements often mean development of parallel codes able to solve the problems faster.

Many advanced “legacy” computational codes (existing codes with decades of man-year efforts) in research and commercial domains are generally developed based on a serial paradigm, the most popular way to design and implement in the recent past. When parallel computing arrived into practice, there were several attempts to convert the legacy codes into parallel. The success rate (in terms of resources and the return on investment) of these efforts, however, depends on how well the original software design (modularity, etc.) is done and the language chosen for implementation. It usually is more effective and efficient to start the design with parallelization as the goal. However, one has to examine the practicality of initiating such an “original” work in the light of work already accomplished. Some of the codes containing millions of lines cannot be easily rewritten, without a heavy investment of resources and the added risk of introducing software defects. So it is quite reasonable to expect interoperability with legacy codes with enhancements to parallelize as much as possible whereas the newly developed codes should be parallelized from the start.

Most codes typically have some parts of computations done serially (i.e., inherently serial) and some other parts that are amenable for parallelization. One can derive the advantage of parallelization only in the latter part. So larger (i.e., major computational time is spent here) this area, the greater will be the returns on developmental investment. One has to consider overhead such as communication, creation of threads, etc. as well in the realization of speedup depending on the approach adopted.

In the present case, the 3D-NWTcode is a serial version, i.e., it works only on a single processor. While this limitation can be somewhat reasonable for small to medium sized models (<20,000 nodes), one will not be able to take full advantage of the HPC platforms in order to model larger domains using fine discretization. Thus, parallelization of this code is a very important consideration.

We have to examine the topic of parallelization options from the user's (or application developer's) perspective of hardware, software and economics. A brief discussion that follows will help clarify some of the background information.

From the hardware point of view, shared memory machines (or SMPs or Symmetric Multiprocessors) have several processors (multi-cores) accessing common memory. The term cluster refers to a group of machines (called nodes) connected through high-speed networks. Each of these nodes in a cluster can have multiple cores (dual, quad, etc.), thus making it a combination of both the types (cluster as well as locally shared). The current and near future trends of major improvement in speeds of the processors will be from the contribution of multi-core processors rather than from the direct improvement in a single processor. This trend no longer follows the so-called Moore’s law (i.e., a single processor’s computational power approximately doubling every two years). Previously, it was quite convenient to take advantage of the serial code by moving it to a newer processor without any modifications to the code. Nowadays it has become very important to develop multi-threaded applications to take advantage of the current computing platforms. Major chipmakers in the semiconductor industry are strongly proposing a paradigm shift in software development techniques (to be increasingly parallel) to take advantage of multi-core processors (shared memory type). Based on this trend, one can visualize a scenario in which there would be hundreds (or even thousands) of cores on a desktop and several such desktops are connected to form a cluster. Therefore, many of the computational-intensive applications that are somewhat constrained in terms of memory and clock speeds at this point of time would become very amenable to be solved in a few years.

We constrain our discussion here to the use of the more relevant parallelization of existing or the so-called legacy codes since it is not often, in the authors’ experience, that a code is written from “scratch” (in which case the choices of parallelization are more obvious). From the application (software) point of view, most of the present-day computational mechanics/physics work is designed and implemented using MPI (Message Passing Interface [14]) and/or OpenMP

(Open Multi-processing [15]). Both MPI and OpenMP work with C and Fortran languages, the two most popular languages used in such codes.

MPI is more often used to take advantage of a network of clusters whereas OpenMP is used on shared memory computers. The presence of a large number of multi-core processors sharing a big memory is of advantage to an OpenMP program. A serial code which uses an in-core memory can be modified into a parallel one using OpenMP and the efforts to make such a modification are in general considerably less than using MPI. With MPI, the software will often require a major redesign as many issues such as message passing and division of work among processors are to be addressed. MPI also adds to the overhead of communication between the processes. OpenMP also has some difficulties of its own such as threads cannot be allowed to write into the same memory location at the same time, etc., so it is rare that the modifications can be so simple and straight-forward. In any case, the choice depends on the original design of the serial code that is in consideration and the availability of resources (hardware, developers and their expertise). It is also possible to program using both MPI and OpenMP, but the relative advantages and resources in terms of using such a combination are to be assessed in practical terms before the undertaking of such a venture.

Taking an economical perspective, individual PCs linked via networks have traditionally been less expensive compared to the shared memory type of systems. This explains the proliferation of clusters at engineering consultancies, university laboratories and national research laboratories. At present, 8/16 processor machines sharing 64GB/128GB RAM are considered to be high-end systems. However, such machines will become more economical with mass production in the coming years and one can be expected to find even better versions as desktops.

4 Choice of an optimal strategy

Major BEM algorithms of the 3D-NWT code were implemented using Fortran, an implementation language that has been quite popular in the domain of scientific computing owing to its speed, capabilities and familiarity with scientists and engineers. The memory is assigned in Fortran common blocks (at least in older Fortran releases) during compile time, which can be found to be a very common design choice before the advent of modern software techniques (e.g., object-oriented design). Such a design, besides being simplistic for smaller-sized codes, permits the constituent subroutines, called in sequence, to access the data that are of interest in a straight-forward fashion. Nevertheless, there is a disadvantage that the data structures are spread all over the code and it is quite difficult to modify the code as the associated code is directly dependent on them. With the addition of features (the inevitability with most of the software) over several years, this dependency between data and implementation became increasingly more complex to manage.

In order to address the issue of parallelizing the 3D-NWT code, an off-the-shelf component - the Distributed Parallel Multipole Tree Algorithm (DPMTA) library [16] developed by the Electrical Engineering department of Duke University is considered. This library was written in C using MPI. Tests on DPMTA indicated that it has linear speedup with respect to the number of processors for the same data and also linear runtime with an increase in data.

3D-NWT has a significant serial part in the computation (mesh generation/regridding, iterative scheme of solution at each time step and also time-stepping until a user-specified number of time steps or termination time is reached). In order to gain maximum advantage from MPI, there should be a division of mesh data and work among the nodes and minimization of message passing. However, this would require a major redesign and rewrite of the code. The risks involved are the introduction of software defects (bugs) and the requirement of extensive testing. An alternative that can be considered is to start from scratch with the latest software design techniques but it would mean reimplementation of a lot of research accomplished so far (over a decade) and this approach is not immune to the risks mentioned in case of redesign either. So a practical strategy that is considered in the present case is that it is far more efficient (in terms of developmental time and efforts) to implement parallelization using OpenMP rather than DPMTA library (and MPI) to make the computations run faster. As discussed in a previous section, this approach will take advantage of multi-core systems - the current major shift in the processor making industry.

5 Implementation using OpenMP

The efforts to address and realize the parallelization in 3D-NWT turned out to be quite involved and more challenging than it seemed at the phase of preliminary analysis. The BEM part of the 3D-NWT code was written using Fortran and the the fast multipole (FMA) library which the BEM algorithms use under the hood was written using C. Some of the Fortran subroutines in the BEM modules make calls to the FMA library which in turn call other Fortran subroutines in the BEM modules to fill-in information on the details of the mesh. There is an inherent heterogeneous language as well as compiler technology (gcc and Intel Fortran) with different paradigms of passing of memory from one part to another. It proved to be quite a challenging task to identify the blocks (profiling) in code that are major bottlenecks and implement OpenMP in such areas with careful modifications such that no two threads write to the same memory location while making the code “work” in an incremental development cycle.

6 Examples using 3D-NWT

The examples included in this section illustrate the wave generation capabilities that are incorporated recently by the authors in 3D-NWT. Figure 1 shows the propagation of a solitary wave on a triangular sloping shelf in the 3D wave basin at Oregon State University. This model has a length of 25.75 m and width of 26.5 m. The foot of the shelf starts at 10.2 m from the wavemaker. The water depth and wave height at the wavemaker are 0.78 m and 0.39 m, respectively.

Motion of the piston wavemaker is created using a software module developed by the authors based on the wave generation algorithms used in OSU's 3D wave basin. The BEM mesh generated has 303 elements along the length, 58 elements along the width and 5 elements along the depth resulting in a total of 40,228 nodes and 38,758 elements. Figure 2 has a close-up snapshot of the incipient breaking wave formation of this model.

The following results are obtained using a numerical wave tank of length of 25 m, width of 1 m for solitary wave (0.5 m width for Cnoidal and Airy waves) and 0.75 m depth.

Free surface elevation of the solitary wave obtained from the 3D-NWT with a wave height of 0.3 m is compared with experimental results at three wave gage locations in Figure 5. The results show a very good agreement although the numerical wave is found to be slightly slower than the experimental wave.

Figures 3 and 4 contain the 3D examples of Cnoidal and Airy waves. For Cnoidal waves, wave height is 0.3 m and time period is 3.5 sec. For Airy waves, wave height is 0.2 m and time period is 3.0 sec. Figures 6, 7 and 8 illustrate the comparison of numerical wave profiles near the wavemaker with experimental/analytical results for solitary, Cnoidal and Airy waves. Cnoidal wave profiles (numerical) reported at three wave gages are shown in Figure 9.

7 Performance studies

In this section, we study the performance of the newly added parallelization capability of the 3D-NWT code. The Fast Multipole Algorithm (FMA) uses multipole expansions and tree data structure [17,18] and avoids the assembling of the coefficient matrix in the solution. It has been discovered that in addition to model size, the run times are also influenced by the choice of the parameters of FMA such as the number of tree levels. So, a detailed study including the effects of such parameters is attempted to serve as a future guideline in arriving at the optimum run time for a given model. Such studies help the engineers or scientists taking the runs to proceed with a balanced approach (in terms of available hardware, model size, accuracy, etc).

The models prepared for this study correspond to the 3D wave basin at Oregon State University. The length and width of the wave basin are 48.8m and 26.5m, respectively. A solitary wave is generated over a water depth of 0.78m using 0.39 m of wave height at the wavemaker. The number of terms in the multipole expansion is 8 and the number of time steps is 5. The FMA cube length (which should be specified to be at least twice of the sum of the maximum dimension and the allowance for wave absorption) covering the wavebasin is 105m. Further details such as the number of tree levels (nlevels), etc. are provided for each model.

A Dell Precision WorkStation 690 with eight nodes (two processor socket quad core), Intel Xeon 3 GHz, 64GB RAM and loaded with 64 bit Redhat Linux Enterprise 5 is dedicated as a test bed. This is an example of a shared memory type of computing systems.

Tables 1-3 indicate the performance of 3D-NWT using three models of the wave basin with coarse to fine discretization and Figures 10-12 illustrate these results graphically. Model-1 has 2592 nodes, model-2 has 13838 nodes and model-3 has 63150 nodes. The number of tree levels is varied from 6 to 8 so as to obtain the best clock times with 8 processors. A discussion on how the best tree level corresponding to a given model size may be identified follows later. Speedup reported is given by the ratio of clock time using a single processor divided by the clock time using multiple processors (= T1/Tp). The maximum speedups that have been realized with these models range from 2.48 to 2.92 which implies that the run time using a single processor can be divided by these numbers to get the best possible clock time. It is observed that speedup is not gaining a lot of increase after the initial addition of a few (~5-6) processors. This can only be expected as the speedups are dependent on the parallel component of the total computation and the associated overhead. From a user’s perspective, it would mean that one has to increase the model size in general to realize better speedups with a larger number of processors.

Figure 13 shows the performance of 3D-NWT using one and eight processors for a variable number of nodes of wave basin models (node numbers range from 26838 to 149250; nlevels=8). The speedups of the five models are 2.97, 3.15, 2.92, 2.07, 1.79. The value of speedup is found to show a downward trend after reaching a maximum of 3.15 with the increase in the model size for the same nlevels. It has been “discovered” later that speedups can also be improved considerably by increasing nlevels (this idea can be observed in results below). It is also found that a number greater than 8 cannot be accommodated in the memory of the test bed machine.

Figure 14 illustrates the performance using one and eight processors with several wave basin models (the largest one having 26838 nodes) with variation of the FMA parameter – number of tree levels varied from 5 through 7. It can be observed that the fastest clock times of all these levels with 8 processors (solid lines) indicate a nearly linear performance (O(n)) with the node numbers (n). It can also be noticed that the single processor runs are not as linear as theoretically expected from the complexity measures of FMA (except in some ranges of the nodes).

Figure 15 illustrates the performance using one and eight processors with several wave basin models (the largest one having 149,250 nodes) with variation of the FMA parameter – number of tree levels varied from 5 through 8. This figure includes a part of Figure 14 as well, however, the results of levels 5 through 7 appear to be magnified in Figure 14 because of the smaller scale used. It can be observed that the fastest clock times of all these levels with 8 processors (solid lines) indicate a nearly linear performance (O(n)) with the node numbers (n). The previous observation for Figure 14 on single processor runs holds good in this case as well.

Some observations on Figures 14 and 15 are:

Faster clock times can be achieved by increasing nlevels with an increase in model size. This is applicable with processors ranging from 1 to 8. As evident from these figures, the user is now equipped with interesting design choices with the number of processors and the parameter nlevels to achieve an optimal clock time for a given model. For instance, one can reduce the number of processors to accommodate other users or other processes on the same machine by adopting appropriate parameter nlevels without incurring much inflation in clock time.

Table 4 indicates the results of a so-called benchmark model with 40228 nodes, 100 time steps, and Figure 16 also shows these results. The dimensions along directions X,Y,Z are 25.755m, 26.5m, 0.78m; numbers of elements along X,Y,Z are 303, 58, 5; 100 time steps reached a termination time of 1.52 sec; FMA cube length=60m. This model has different dimensions (length considered is shorter than the actual wave basin in order to focus on wave propagation in the area of interest with faster execution) from the previous models. The speedup gain is not as significant for this model because the transition of optimal time for nelvels=7 to 8 happens in between nodes 50000 and 60000 (from Figure 15). For instance, a model with greater than or equal to 60000 nodes and nlevels=8 can be expected to provide a better speedup for an increased accuracy.

8 Insight and experience gained

Some of the understanding and the experience gained during this parallelization effort are summarized below to serve as guidelines for developers undertaking similar efforts and for users to have the code run faster.

1. Parallel codes using OpenMP work by creating the user-specified number of threads. The threads are different from processes (an example of process is a serial program) in that they are lightweight, independent instructions that execute within a process. All threads created by the same process share the same memory space. Thus threads can take advantage of accessing the same data in centralized memory without duplicating the data and avoid transfer of data as needed by message passing paradigm (as used by MPI). However, an OpenMP software developer has to be extremely cautious to take care that threads do not write to the same location in memory at the same time and avoid the several possible pitfalls of multi-threaded programming. The number of threads can be as many as the number of processors in order to achieve load balancing (ideally each processor taking equal load), but is not necessarily limited to them. However, using too many threads more than the number of processors can cause the system to become nonresponsive to other tasks (and also to other users). A practical point of significance is that the overhead (including the creation of threads) for parallelization is not negligible. The size of the parallel component must be significant enough to overcome the overhead. In other

words, the problem size (or model size in a simulation) should be large enough to realize good speedups. It is prudent to start with a small number of time steps for a given model, study the speedup by varying the number of threads and grow the termination time from there.

2. FMA input parameter - number of tree levels: This parameter indicates the number of hierarchical levels into which the cube (containing the model) is divided. The speedup is considerably affected by this parameter as the work shared by the threads is dependent on the number of cells in the hierarchy. The user should be able to experiment and determine the optimum parameter for a specified number of threads for a given problem size. The practical values of this parameter range from 4 to 8. A very large number of tree levels (>8) would considerably increase the memory requirement and the run may not be able to fit within the memory resources of the computing system. This limitation, however, can be alleviated by the availability of larger (and better) hardware resources.

3. FMA input parameter - cube length: This is the cube size surrounding the entire dimensions of the model. The cube is defined with the origin of the model at its center. The user has to identify the maximum dimension from the origin of the model, estimate the possible change in length (due to wave absorption for instance) and use twice of it for the cube length. One should use as low a value as possible for optimal performance of the algorithms.

4. FMA input parameter - number of terms in a multipole expansion: The larger this number, the better the accuracy of the expansions and higher the execution time. While its range could be between 4 and 16, the practical value is around 6 through 8. An investigation taken up in this direction using a wave basin model indicated that the difference in the time steps is insignificant with this parameter between 8 and 12, unlike the comparison between 4 and 8.

5. The distribution of nodes (or the model geometry and number of nodes) in the FMA cube can also have an effect on the clock time. A study done using a cube size of 60 m instead of 120 m caused a reduction in clock time from 19.76 hrs to 8.53 hrs. The reason is that a cube size of 60 caused nodes to be more uniformly distributed in cells and reduced the computational effort.

6. Location of the nodes in the model can have considerable effect on clock time which can be found by numerical experimentation. An example is the number of nodes distributed along the length versus breadth versus depth.

9 Concluding remarks

This paper presented parallelization of the 3D-NWT code, demonstrated the applicability of this code to large-scale simulations by the performance studies and illustrated the recently incorporated wave generation capabilities by examples. The performance studies provided very good indicators of O(n) performance which enabled the FNPF with BEM to be applied to solve several medium to large-scale problems (as evidenced by 3D wave basin models) in a reasonable time frame. A maximum speedup close to 3 is realized in quite a few cases. The parallelization demonstrated linear performance unlike the single processor runs that indicated linear performance ("piecewise linear") only in some ranges of number of nodes.

While this is an exciting development in a BEM application by itself, an analyst may be interested in realizing better speedups in the future in order to solve very large-scale problems of millions of nodes. In this context, further development will need to focus on the following serial/computational-intensive components of run times of the 3D-NWT code: the iterative (GMRES) algorithm is used during each time step, a number of time steps (adaptively computed based on the free surface; the finer the mesh results in a smaller time step) needed to reach a specified termination time, regridding is needed to distribute the nodes on free surface uniformly, and higher-order elements are used. Thus, further improvements in speedup can be realized only by designing algorithms that share these tasks among several nodes on cluster (distributed memory) with as little overhead as possible, a task that would require a major undertaking and resources.

In the context of this paper, it is considered that it will be of benefit to the computational scientists and engineers in the simulation field to be aware of some recent developments in software technologies that are aimed at improving productivity to solve their applications. The "next-generation" HPC programming languages such as Fortress (by Sun), X10 (by IBM) and Chapel (by Cray) are in development under the DARPA initiative [19] and have the potential to significantly improve the quality and efforts of the HPC. These languages offer higher-level constructs than MPI and OpenMP (which are actually external constructs/libraries that are popularly used along with C/C++ and Fortran) and have the programmers' productivity in focus. Fortress and X10 are either directly or indirectly associated with Java/Java Virtual Machine, whereas Chapel [20] is designed to be a new parallel programming language and can be more appealing to users (with a background in C/C++, Java, Fortran, etc) who do not want to use Java as the platform. These languages also provide interfaces to the (now) low-level languages such as Fortran and C. For new developments that start from scratch and those that plan to use old libraries, it will be prudent to explore and use one of these languages to boost productivity and robustness. With abstraction, the scientist or engineer can focus on his/her problem-domain and interpretation of results instead of low-level details such as memory allocation/deallocation, array sizes, pointers, etc. These ideas are similar to the times when assembly language (although faster) was replaced by the then so-called higher-level languages such as C and Fortran. It is also worth looking at the functional programming languages (e.g., Erlang, Haskell) that are being mentioned in the computing world as promising tools to take advantage of multi-core systems and for building shorter, easily maintainable, extendable and better quality software for HPC and related research.

10 Figures and tables

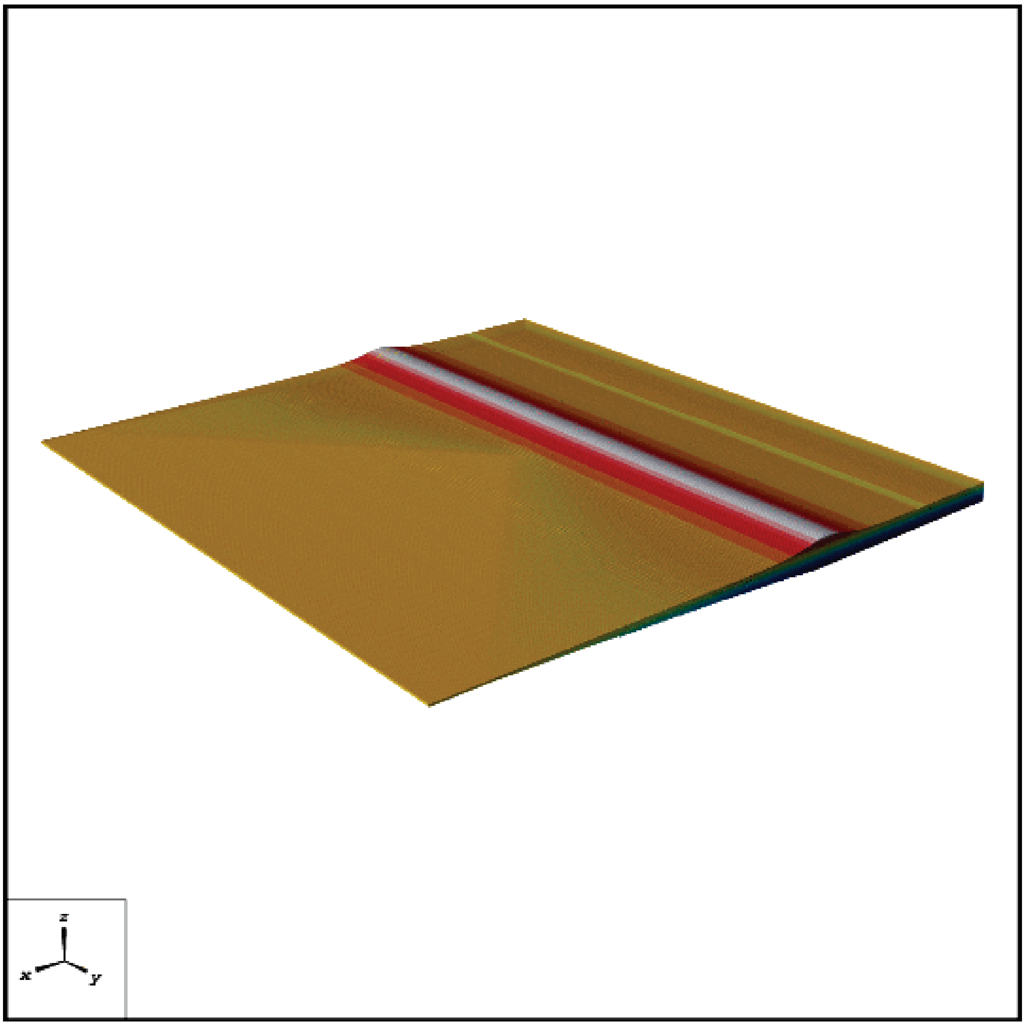


Figure 1: Propagation of a solitary wave on a triangular sloping shelf in the 3D wave basin at Oregon State University. The wave starts at x=0 and propagates toward the positive X direction

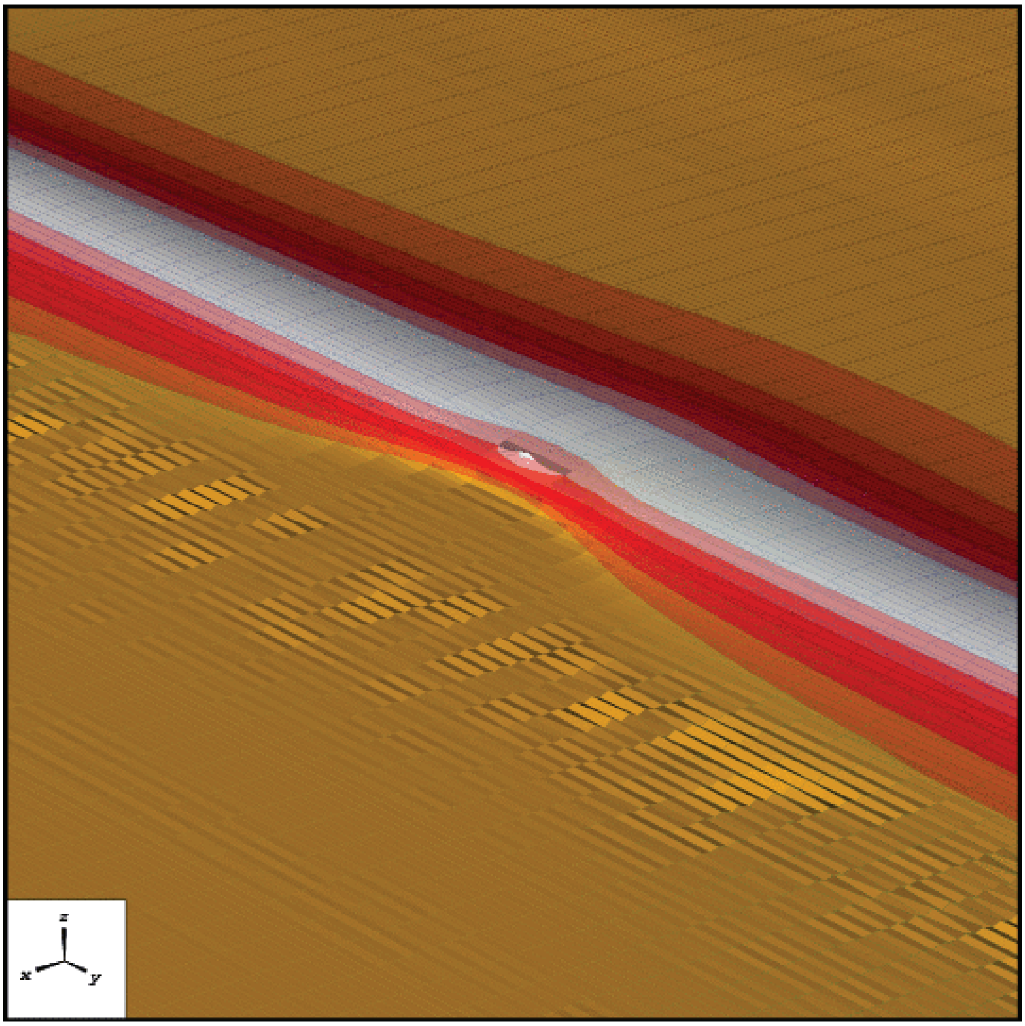


Figure 2: Close-up of a solitary wave on a triangular sloping shelf; development of a breaking wave just past the triangle’s apex is visible here

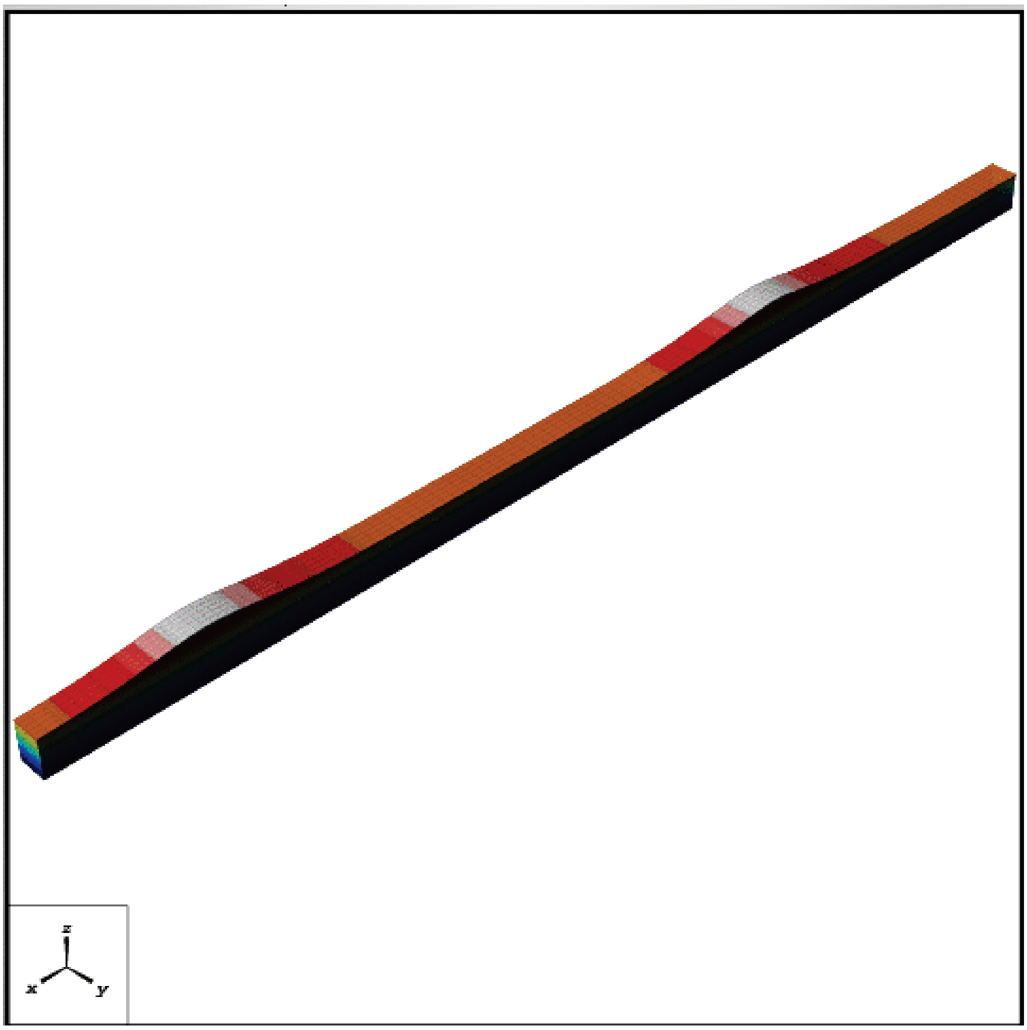


Figure 3: Snapshot of Cnoidal waves

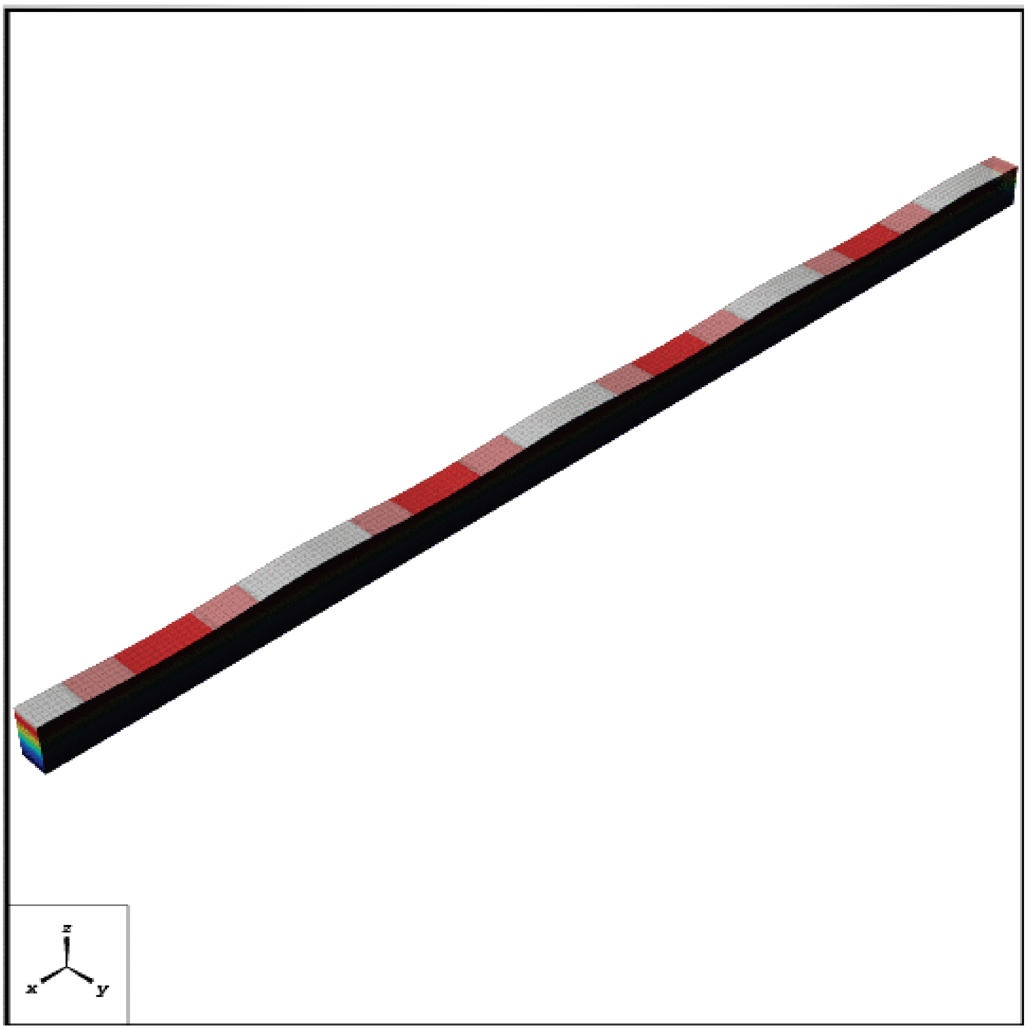


Figure 4: Snapshot of Airy waves

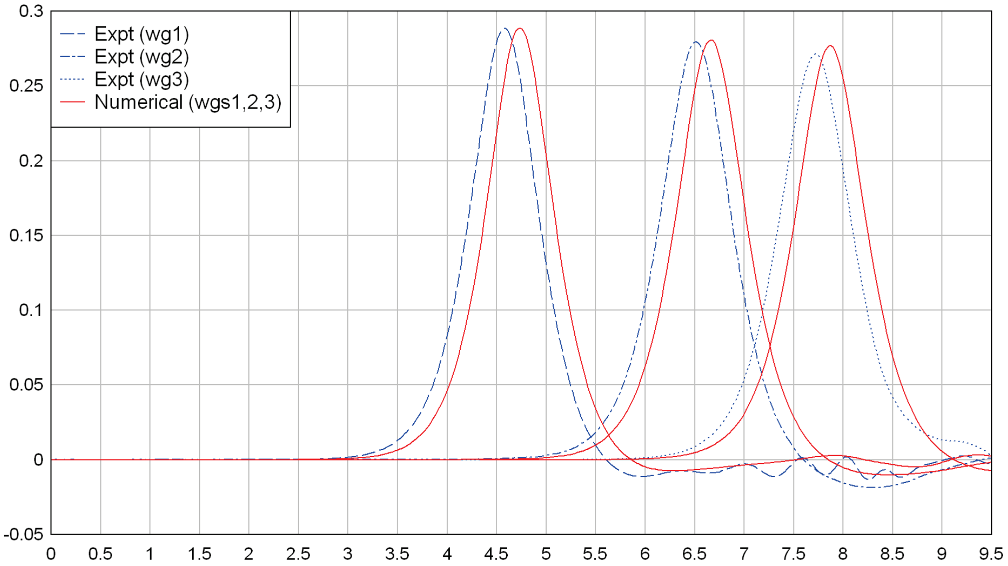


Figure 5: Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of solitary wave at three wave gages at x=8.8 m, 14.9 m and 18.7 m at y=0 for wave height=0.3 m; the numerical waves are found to lag slightly behind the experimental waves, but the results show a very good agreement

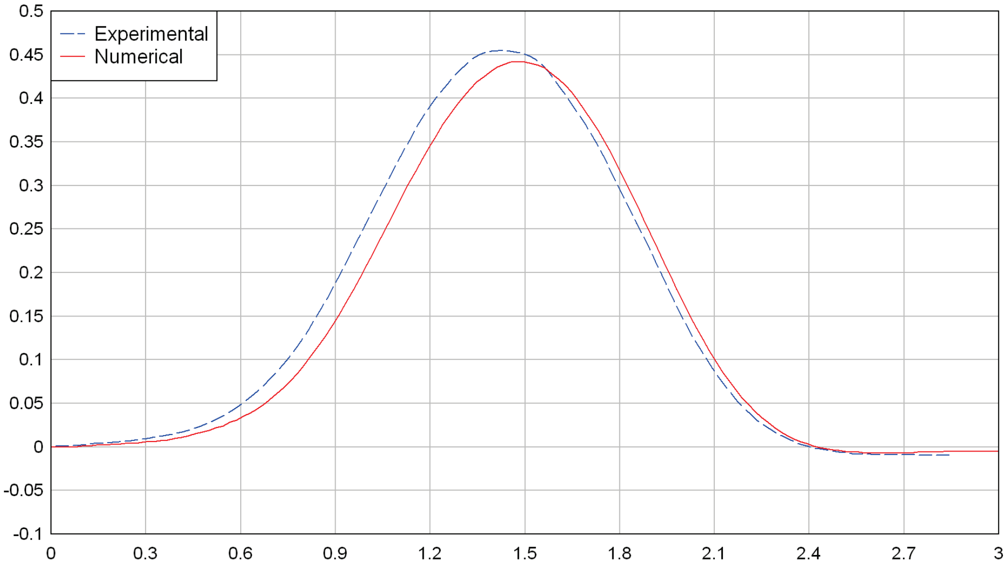


Figure 6: Time (in sec on horizontal axis) versus experimental and numerical wave profiles (in m on vertical axis) of solitary wave at wavemaker for wave height=0.45 m

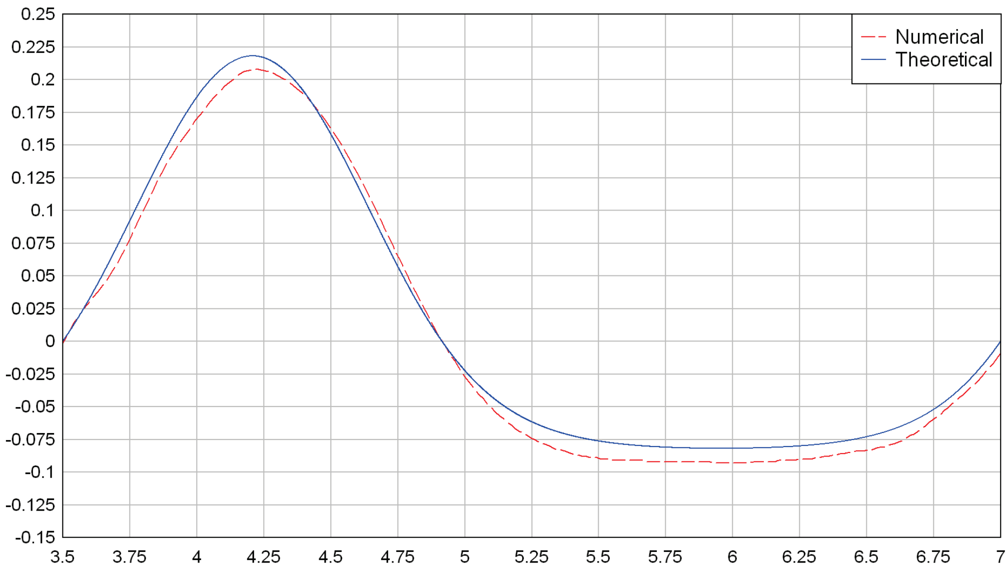


Figure 7: Time (in sec on horizontal axis) versus theoretical (analytical) and numerical wave profiles (in m on vertical axis) of Cnoidal wave at wavemaker for wave height=0.3 m

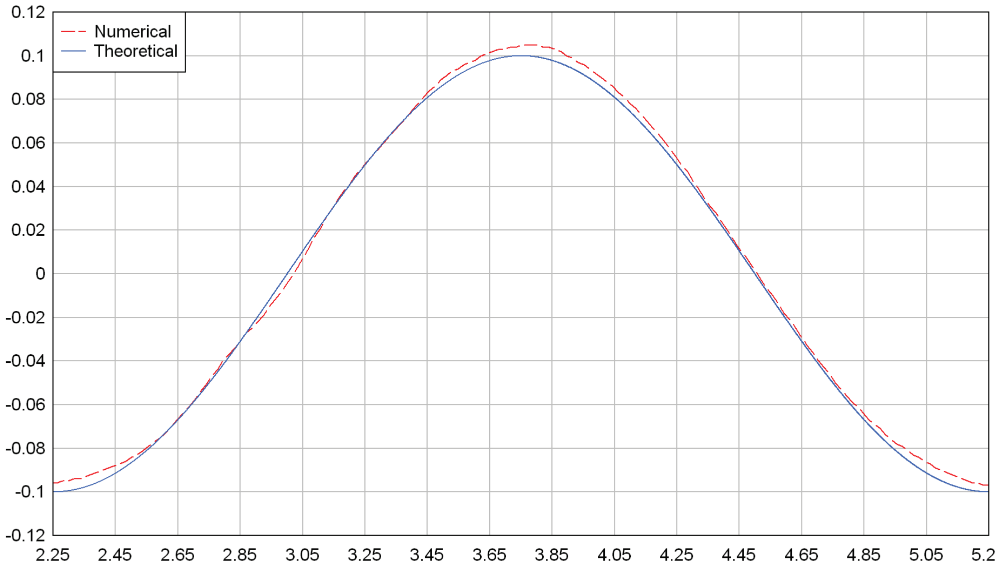


Figure 8: Time (in sec on horizontal axis) versus theoretical (analytical) and numerical wave profiles (in m on vertical axis) of Airy wave at wavemaker for wave height=0.2 m

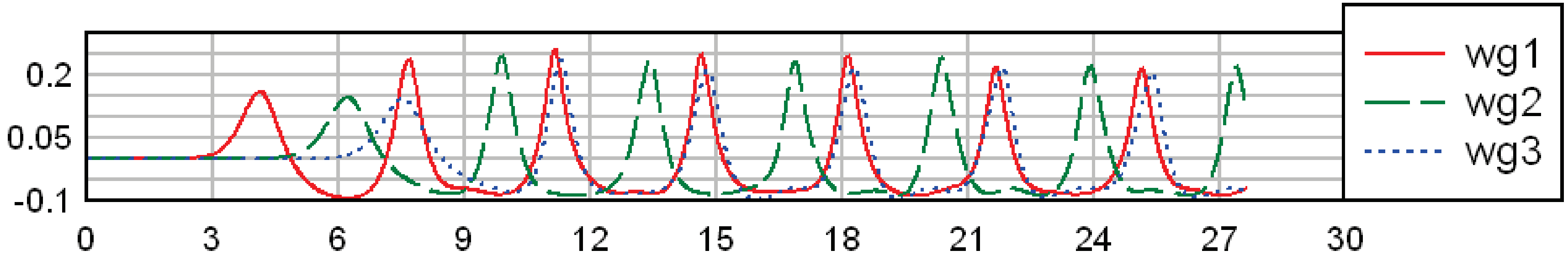


Figure 9: Time (in sec on horizontal axis) versus numerical wave profiles (in m on vertical axis) of Cnoidal waves at three wave gages at x=8.8 m, 14.9 m and 18.7 m at y=0 for wave height = 0.3 m

Table 1:- Test model-1: # of nodes=2592, # of tree levels in FMA=nlevels=6; 5 time steps

Number of processors (ncpu)	Execution time in sec	Speedup (T1/Tp)
1	126.5	1.00
2	84.5	1.50
3	67	1.89
4	58	2.18
5	54.5	2.32
6	53	2.39
7	51	2.48
8	50.5	2.51

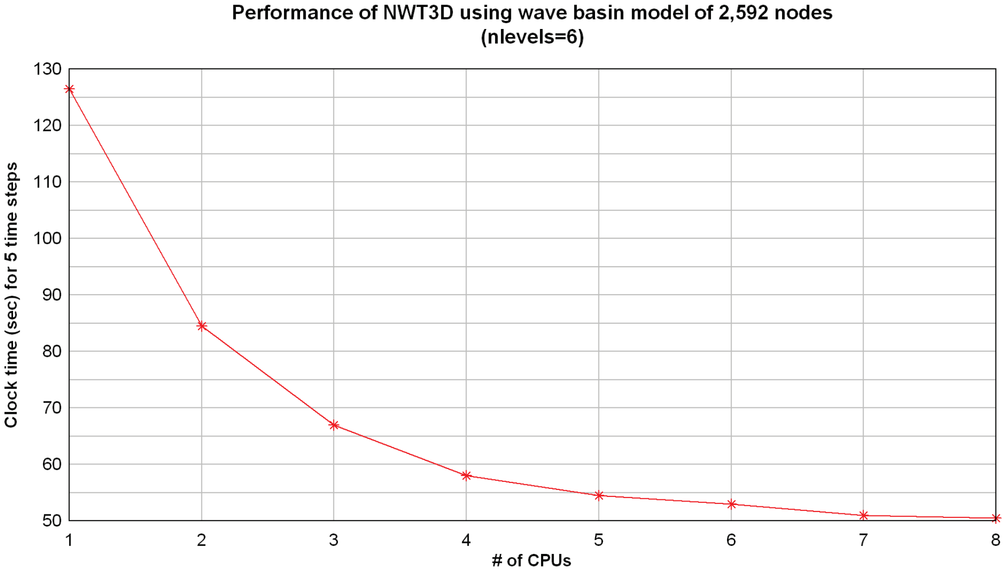


Figure 10: Performance (#of CPUs versus clock time in sec) using model-1

Table 2:- Test model-2: # of nodes=13838, # of tree levels in FMA=nlevels=7; 5 time steps		
Number of processors (ncpu)	Execution time in sec	Speedup (T1/Tp)
1	1102	1.00
2	751	1.47
3	609	1.81
4	514	2.14
5	511	2.16
6	480	2.30
7	456	2.47
8	444	2.48

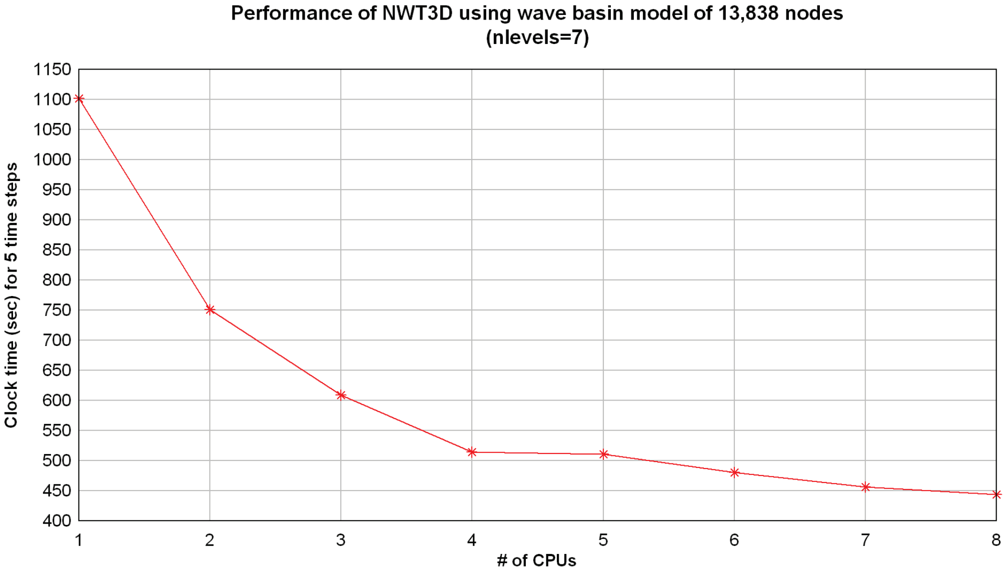


Figure 11: Performance (#of CPUs versus clock time in sec) using model-2

Table 3:- Test model-3: # of nodes=63150, # of tree levels in FMA=nlevels=8; 5 time steps		
Number of processors (ncpu)	Execution time in sec	Speedup (T1/Tp)
1	9267	1.00
2	5963	1.55
3	4790	1.94
4	4145	2.24
5	3792	2.44
6	3376	2.75
7	3266	2.84
8	3176	2.92

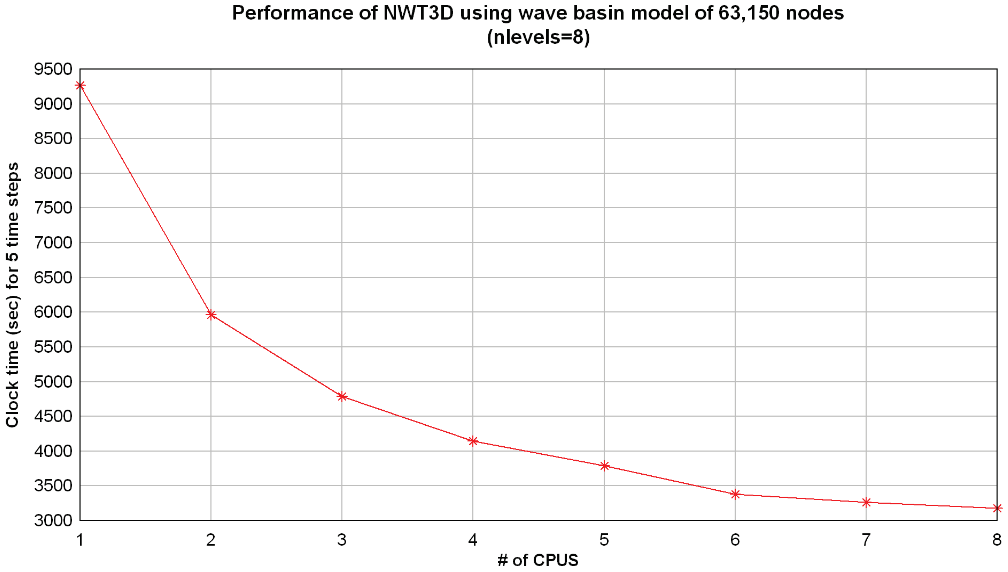


Figure 12: Performance (#of CPUs versus clock time in sec) using model-3

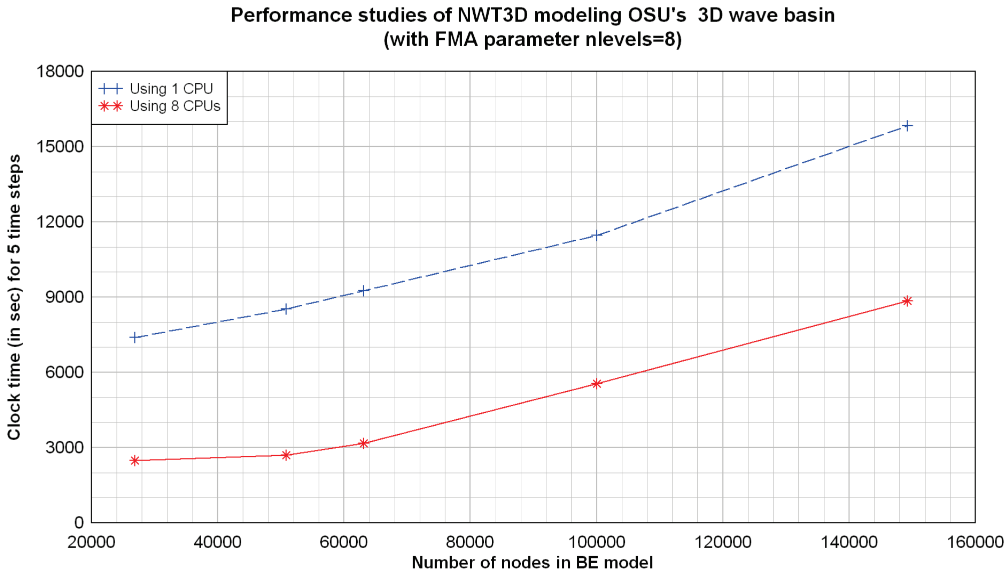


Figure 13: Performance (number of nodes versus clock time) using several models with nlevels=8; solid line (-) is with 8 processors; dashed line (--) is with single processor

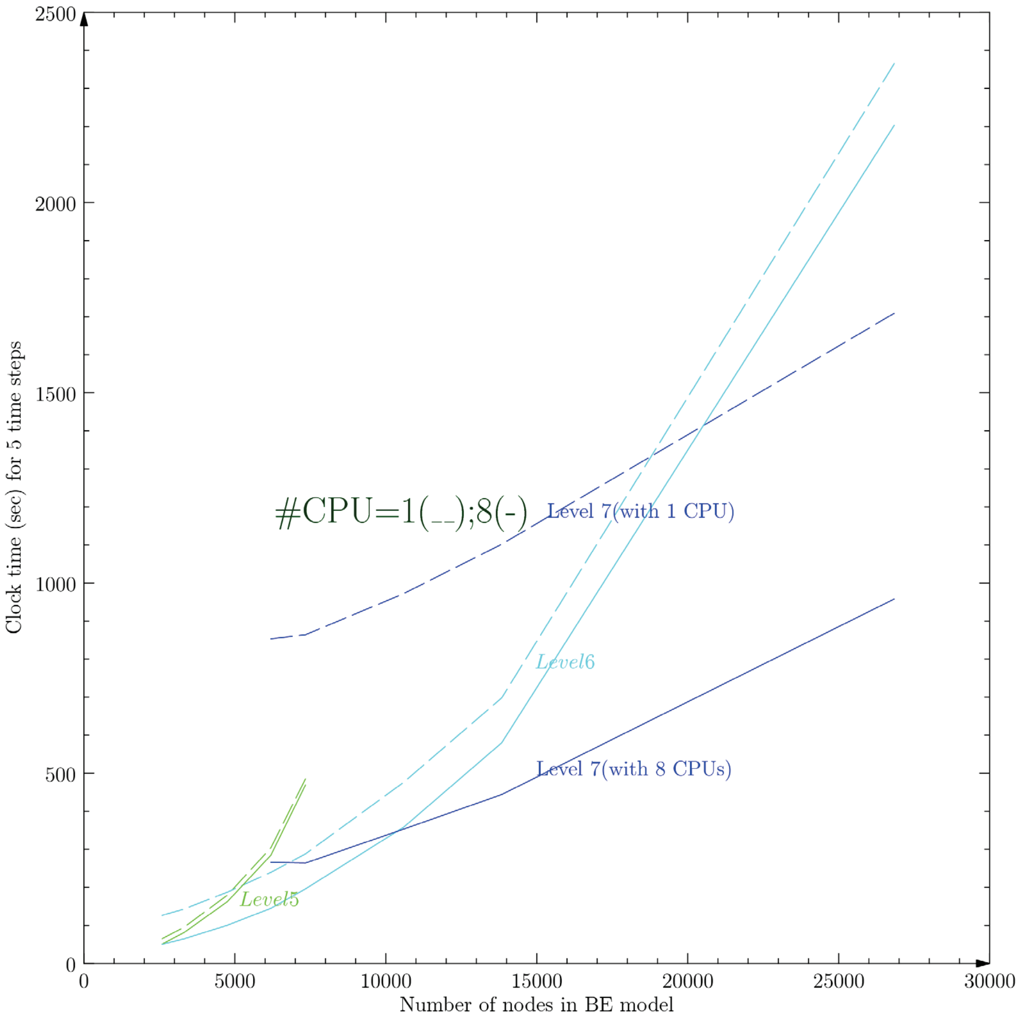


Figure 14: Performance using several models with nlevels=5, 6, 7; solid line (-) is with 8 processors; dashed line (--) with single processor; there is nearly linear performance of run times with 8 processors (for instance, using solid lines comprising of the fastest clock times of levels 6 and 7)

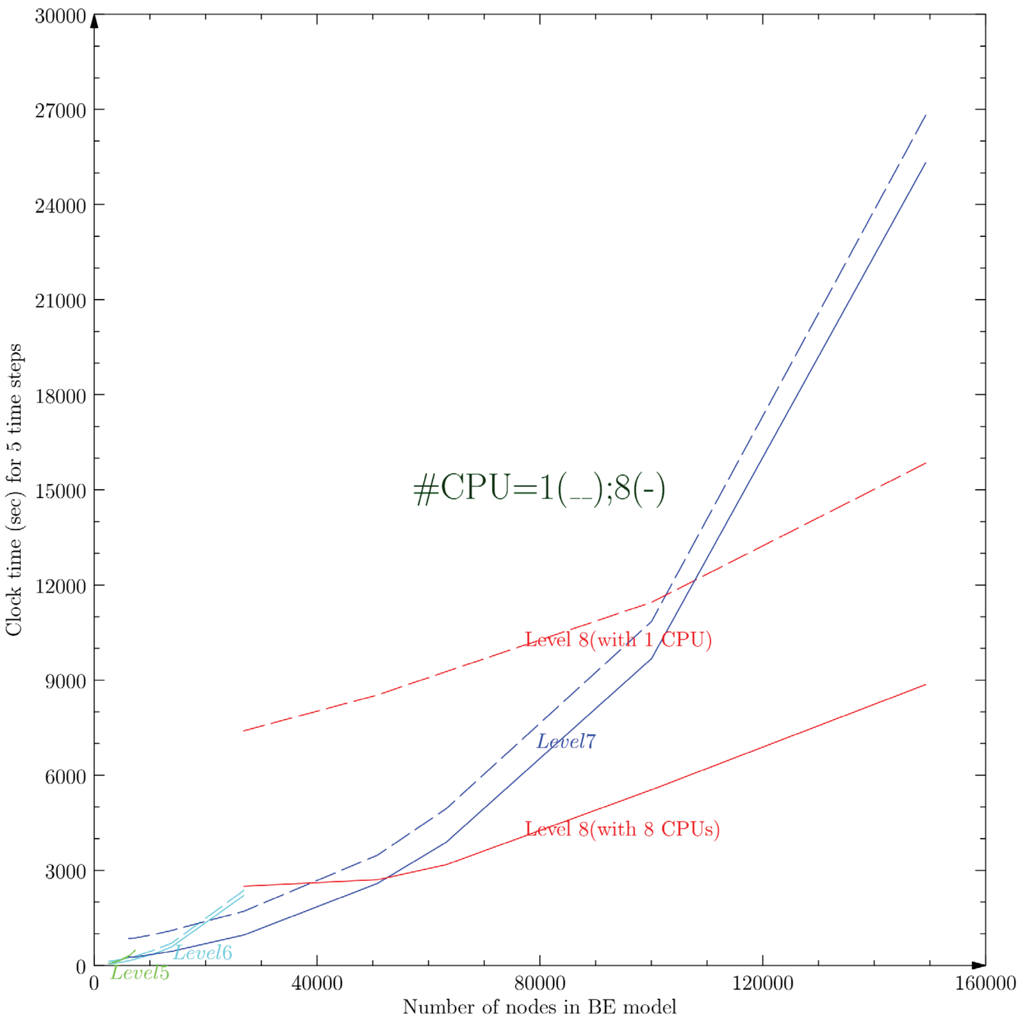


Figure 15: Performance using several models with nlevels=5, 6, 7, 8; solid line (-) is with 8 processors; dashed line (--) with single processor; there is nearly linear performance of run times with 8 processors (for instance, using solid lines of the fastest clock times of levels 6, 7 and 8)

Table 4:- Benchmark model: # of nodes=40228, # of tree levels in FMA=nlevels=7; 100 time steps		
Number of processors (ncpu)	Execution time in sec (hrs)	Speedup (T1/Tp)
1	51768 (14.38)	1.00
2	40495 (11.25)	1.28
4	34068 (9.46)	1.52
6	31578 (8.77)	1.64
8	30709 (8.53)	1.69

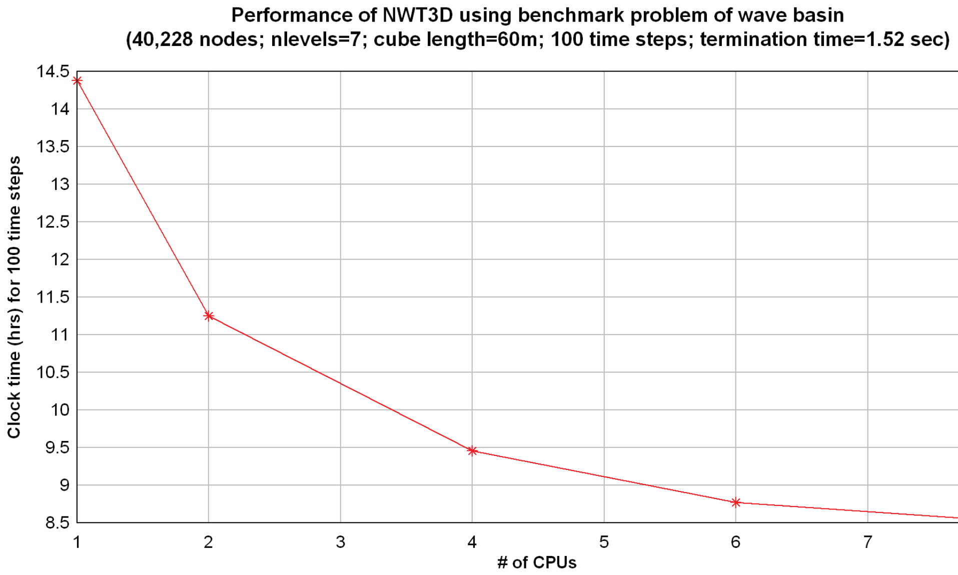


Figure 16: Performance (#of CPUs versus clock time in hours) using the benchmark problem

Acknowledgement

The first two authors are thankful to the Office of Naval Research (ONR) for their generous financial support that made this work possible.

References

[1] Grilli, S.T., Guyenne, P., and Dias, F., 2001, “A fully nonlinear model for three-dimensional overturning waves over arbitrary bottom,” International Journal for Numerical Methods in Fluids, **35**(7), pp. 829-867.

[2] Brandini, C., and Grilli, S.T., 2001, “Modeling of freak wave generation in a 3D-NWT,” In Proc. 11th Offshore and Polar Engng. Conf. (ISOPE01, Stavanger, Norway, June 2001), Vol III, pp. 124-131.

[3] Grilli, S.T., Vogelmann, S., and Watts, P., 2002, “Development of a 3D numerical wave tank for modeling tsunami generation by underwater landslides,” Engineering Analysis with Boundary Elements, **26**(4), pp. 301-313.

[4] Fochesato, C., Grilli, S.T., and Guyenne, P., 2005, “Note on non-orthogonality of local curvilinear coordinates in a three-dimensional boundary element method,” International Journal for Numerical Methods in Fluids, **48**, pp. 305-324.

[5] Fochesato, C., and Dias, F., 2006, “A fast method for nonlinear three-dimensional free-surface waves,” Proceedings of the Royal Society, London, **A 462**, pp. 2715–2735.

[6] Guyenne, P., and Grilli, S.T., 2006, “Numerical study of three-dimensional overturning waves in shallow water,” Journal of Fluid Mechanics, **547**, pp. 361-388.

[7] Fochesato, C., Grilli, S.T., and Dias F., 2007, “Numerical modeling of extreme rogue waves generated by directional energy focusing,” Wave Motion, **44**, pp. 395-416.

[8] Sung, H.G., and Grilli, S.T., 2008, “BEM Computations of 3D Fully Nonlinear Free Surface Flows Caused by Advancing Surface Disturbances,” International Journal of Offshore and Polar Engineering, **18**(4), pp. 292-301.

[9] Grilli, S.T., Dias, F., Guyenne, P., Fochesato, C., and F. Enet, 2010, “Progress In Fully Nonlinear Potential Flow Modeling Of 3D Extreme Ocean Waves,” Chapter 3 in Advances in Numerical Simulation of Nonlinear Water Waves, Vol. 11 in Series in Advances in Coastal and Ocean Engineering, World Scientific Publishing Co. Pte. Ltd., pp. 75-128.

[10] Wrobel, L.C., 2002, “The boundary element method- Applications in thermo-fluids and acoustics,” Volume 1, John Wiley & Sons, Ltd.

[11] Nimmala, S.B., Yim, S.C., Grilli, S.T., “An Efficient Parallelized 3-Dimensional Numerical Wave Tank to Model Large-Scale Wave Basin Experiments,” (to be published).

[12] Steven A. Hughes, 1993, “Physical models and laboratory techniques in coastal engineering,” World Scientific Publishing.

[13] Derek G. Goring, 1978, “Tsunamis- The propagation of long waves onto a shelf,” W.M. Keck Laboratory of Hydraulics and Water Resources, Division of Engineering and Applied Science, California Institute of Technology, Pasadena, California, Report no. KH-R-38.

[14] Quinn, M.J., 2003, “Parallel programming in C with MPI and OpenMP,” McGraw Hill.

[15] Chapman, B., et al., 2007, “Using OpenMP: Portable shared memory parallel programming,” MIT Press.

[16] Rankin, W.T., 1999, "Efficient Parallel Implementations of Multipole based N-Body algorithms,” PhD Thesis, Dept. of Electrical Engineering, Duke University.

[17] Nishimura, N., 2002, “Fast multipole accelerated boundary integral equation methods,” Applied Mechanics Review, 55(4), pp. 299-324.

[18] Liu, Y., 2009, “Fast multipole boundary element method: theory and applications in engineering,” Cambridge University Press.

[19] DARPA (Defense Advanced Research Projects Agency) High Productivity Computer Systems (http://www.highproductivity.org/).

[20] The Chapel parallel programming language (http://chapel.cray.com/).

Chapter-6: General Conclusions

This dissertation presented the applicability of the 3D-NWT to large-scale simulations by the implementation of parallelization, and demonstration using the performance studies and by comparison with the results of the experiments conducted in the 3D wave basin at OSU. The performance studies provided very good indicators of O(n) (i.e., linear) performance which enabled the fully nonlinear potential flow (FNPF) model with the boundary element method (BEM) to be applied to solve several medium to large-scale problems (as evidenced by 3D wave basin models) in a reasonable time frame. A maximum speedup close to 3 is realized in several cases. While this is an exciting development in a BEM application by itself, an analyst may be interested in realizing better speedups in the future in order to solve very large-scale problems of millions of nodes. In this context, further development will need to focus on the following serial/computational-intensive components of run times of the 3D-NWT code: the iterative (GMRES) algorithm used during each time step, the number of time steps (adaptively computed based on the free surface; a finer mesh results in a smaller time step) needed to reach a specified termination time, regridding needed to distribute the nodes uniformly on the free surface, and higher-order elements used in the discretization. Thus, further improvements in speedup can be realized only by designing algorithms that share these tasks among several nodes on a cluster (distributed memory) with as little overhead as possible, a task that would be quite challenging and requires a major undertaking and resources.

A very good comparison of results with the physical experiments and also with the commercial code LS-DYNA [59,60] in the case of the benchmark experiment (described in the second manuscript) indicates that the 3D-NWT code can be confidently applied to the real-world wave generation problems in 3D wave basins. The three-way comparison (physical experiment, 3D-NWT and LS-DYNA) provided in the benchmark experiment, which is an application of solitary wave generation to complex bathymetry, is a good illustration of the large-scale simulations. The motion of the piston wavemaker can be activated by the input provided by the Wavegen module which is developed for the purpose, and also by another software module as per the user's choice. Although the waves such as solitary (tsunami), Cnoidal, Airy and three alternatives of focused waves are implemented, there is further scope to implement more specialized type of waves- the snake waves, etc. It could however be an extension/adaptation of the existing subroutines. It may be of interest to some practicing ocean engineers to implement the bottom friction as well, which is an extension to three dimensions from the two dimensional case implemented by Grilli, et al [53].

The development of the 3D-NWT comes with all the advantages of the BEM - requiring surface-only discretization (thereby reducing the mesh data in the three dimensions and the regridding efforts at each time step unlike the case of the domain discretization methods) and high accuracy. In general, fully nonlinear free surface waves are considered (independent of the magnitudes of the water depth and wave height) without making any assumptions about the solution. The wave propagation can proceed as long as there is no excessive mesh distortion due to phenomena such as wave breaking, in which case the run has to be restarted using a finer mesh. The fluid flow is physically no longer irrotational after the wave breaking and air bubbles are included, thus the model equation is strictly not valid in such situations. The automatic mesh generation is specified using simple commands and the input files have only a few lines editable in text format. BEM coupled with the fast multipole algorithm (FMA) (which led to linear runtimes), GMRES and parallelization to take advantage of multicore machines (e.g., personal computers with several cores) of HPC and the incorporation of the piston wavemaker to generate a variety of waves has brought the realization of a 3D numerical wave tank complementing the physical wave basin. Visualization capabilities added to the software facilitate a faster feedback system to the analyst by the automatic generation of snapshots and animations.

The mesh generation in 3D-NWT is done automatically at each time step due to moving boundaries. However, a mesh generation scheme based on the inspection of the analyst (not only in the regions of overturning waves but also at other critical regions of interest) at a given time step and restart (which is already available in some form) will be of good practical utility. Development of a graphical user interface (GUI) to run the code, and perform pre-and post-processing will also be a good addition to the productivity of the analyst.

While the developments mentioned in this dissertation are provided with comparisons with the physical experiments, it will be of interest to study mathematical/numerical analyses (nonlinear stability, etc.) of the problems to provide a comprehensive theoretical basis in future. Meshless version of the BEM is just appearing in the literature and it can be advantageous in solving problems with moving boundaries. Studies of mesh refinement and its effect on convergence rate will offer guidance to the practicing engineers.

A summary of the limitations (compiled from literature) of the code is provided here. This code is in general valid up to the moment the tip of the breaker jet hits the free surface and can be applied to deep to shallow water situations with arbitrary length waves. It does not have the small parameter assumptions, etc. that are used in the analytical or numerical expansion wave theories. Bottom friction and flow separation cannot be modeled directly, however, bottom friction is not an important consideration in many cases. Potential flow models are not accurate for small-scale problems where viscosity may have large influences (such as boundary layers) on the solution. Runup of non-breaking waves on a steep or gentle slope can be accurately calculated as long as wave reflection doesn't cause another incoming wave break. Physical phenomena such as air-water interface breaking, water-structure impact, air cushioning, and vorticity generation cannot be adequately represented by the potential flow model and require a more general solver [39] which is however applied only for small-scale domains due to computational limitations.

The performance studies of LS-DYNA and its application to an ODOT bridge (presented in Appendix-E) indicate that the computer hardware capabilities have yet to catch up with the computational needs of a full-scale 3D CFD simulation. However, currently 2D CFD simulations are a feasibility as demonstrated in an application to the Spencer Creek bridge [71]. More such studies are being actively pursued in a follow-up thesis [72].

This dissertation presented the solution to an important component of the FSI problem- namely accurate input generation to the more detailed models containing the structure. The work being conducted in Prof. Solomon Yim's group at OSU will be comprised of the ideas of domain-decomposition [39] with FNPF (the current work) representing the outer fluid domain and the inner domain represented by Navier-Stokes or Reynolds Averaged Navier Stokes (RANS) or such appropriate models. When this work is accomplished, it is expected to present a comprehensive solution to the FSI problem.

Since the computational scientists and engineers in the simulation field have to deal with development of better and faster software tools to solve their applications, it may be appropriate to mention the following software technologies being developed under the DARPA initiative [74] that have the potential to significantly improve the quality and efforts of HPC- especially the "next-generation" programming languages such as Fortress (by Sun), X10 (by IBM) and Chapel (by Cray). These languages offer higher-level constructs than MPI and OpenMP (which are actually external constructs/libraries that are popularly used along with C/C++ and Fortran) and have the programmers' productivity in focus. The serial code can be converted into parallel by means of a simplified syntax. The application developer does not have to deal with the low-level details such as data transfer (send/receive), processor information, etc. Fortress and X10 are either directly or indirectly associated with Java/Java Virtual Machine, whereas Chapel [75] is designed to be a new parallel programming language and can be more appealing to users that have a background in C/C++, Java, Fortran, etc and do not want to use Java as the platform. These languages also offer Foreign Function Interfaces (FFIs) to interoperate with the (now) low-level languages such as Fortran and C. For the new developments that start from scratch and those that plan to use old libraries, it will be prudent to explore and use one of these languages to boost productivity and robustness. With abstraction, the scientist or the engineer can focus on his/her problem-domain and interpretation of results instead of low-level details such as memory allocation/deallocation, array sizes, pointers, etc. These ideas are similar to the times when assembly language (although faster) was replaced by the then so-called higher-level languages such as C and Fortran. It is also worth looking at the functional programming languages (e.g., Erlang, Haskell) that are being mentioned in the computing world as promising tools to take advantage of the multi-core systems and for building shorter, easily maintainable, extendable and better quality software for HPC and related research.

Bibliography

[1] Kanwal, R.P., 1997, "Linear integral equations- theory & technique," Birkhauser, Boston.

[2] Delves, L.M., and Mohamed, J.L., 2008, "Computational methods for integral equations," Cambridge University Press.

[3] Chen, G., and Zhou, J., 1992, "Boundary Element Methods," Academic Press.

[4] Hsiao, G.C., and Wendland, W.L., 2008, "Boundary integral equations," Springer-Verlag, Berlin, Heidelberg.

[5] Banerjee, P.K., and Butterfield, R., 1981, "Boundary element methods in engineering science," McGraw-Hill Book Co (UK) Ltd.

[6] Liggett, J.A., and Liu, P.L.F., 1983, "The boundary integral equation method for porous media flow," George Allen & Unwin Publishers, London.

[7] Brebbia, C.A., Telles, J.C.F., and Wrobel, L.C., 1984, "Boundary element techniques: theory and applications in engineering," Springer-Verlag, Berlin.

[8] Hartmann, F., 1989, "Introduction to Boundary elements," Springer-Verlag.

[9] Brebbia, C.A., and Dominguez, J., 1989, "Boundary elements: an introductory course," Computational Mechanics Publications, Southampton, Boston.

[10] Banerjee, P.K., 1994, "The boundary element methods in engineering," McGraw-Hill Book Company.

[11] Power, H., and Wrobel, L.C., 1995, "Boundary integral methods in fluid mechanics," Computational Mechanics Publications, Southampton, Boston.

[12] Wrobel, L.C., 2002, "The boundary element method- Applications in thermo-fluids and acoustics," Volume 1, John Wiley & Sons, Ltd.

[13] Aliabadi, M.H., 2002, "The boundary element method- Applications in solid and structures," Volume 2, John Wiley & Sons, Ltd.

[14] Bonnet, M., 1995, “Boundary integral equation methods for solids and fluids,” John Wiley & Sons Ltd.

[15] Nishimura, N., 2002, “Fast multipole accelerated boundary integral equation methods,” Applied Mechanics Review, 55(4), pp. 299-324.

[16] Ying, L., Biros, G., Zorin, D., and Langston, H., 2003, “A new parallel kernel-independent fast multipole method,” Proceedings of the 2003 ACM/IEEE Conference on Supercomputing, 196(2), pp. 591-626.

[17] Ying, L., 2004, “An efficient and high-order accurate boundary integral solver for the Stokes equations in three dimensional complex geometries,” PhD thesis, Department of Computer Science, New York University.

[18] Ying, L., Biros, G., and Zorin, D., 2004, “A kernel-independent adaptive fast multipole algorithm in two and three dimensions,” Journal of Computational Physics, 196(2), pp. 591-626.

[19] Ying, L., Biros, G., and Zorin, D., 2006, “A high-order 3D boundary integral equation solver for elliptic PDEs in smooth domains,” Journal of Computational Physics, 219(1), pp. 247-275.

[20] Margonari, M., and Bonnet, M., 2005, “Fast multipole method applied to elastostatic BEM-FEM coupling,” Computers and Structures, 83, pp. 700-717.

[21] Chaillat, S., Bonnet, M., and Semblat, J-F., 2008, “A multi-level fast multipole BEM for 3-D elastodynamics in the frequency domain,” Computer Methods in Applied Mechanics and Engineering, 197(49-50), pp. 4233-4249.

[22] Liu, Y., 2009, “Fast multipole boundary element method: theory and applications in engineering,” Cambridge University Press.

[23] Sternberg, W.J., and Smith, T.L., 1946, “The theory of potential and spherical harmonics,” The University of Toronto Press.

[24] Kellogg, O.D., 1953, “Foundations of Potential theory,” Dover Publications Inc.

[25] Dommermuth, D.G., Yue, D.K.P., Lin, W.M., and Rapp, R.J., 1988, "Deep-water plunging breakers: a comparison between potential theory and experiments," Journal of Fluid Mechanics, 189, pp. 423-442.

[26] Peregrine, D.H., 1992, "Mechanisms of water-wave breaking," Breaking waves, International Union of Theoretical and Applied Mechanics (IUTAM) Symposium, Sydney, Australia, 1991, Banner, M.L., and Grimshaw, R.H.J., (eds), Springer-Verlag.

[27] Cooker, M.J., and Peregrine, D.H., 1992, "Violent motion as near breaking waves meet a vertical wall," Breaking waves, International Union of Theoretical and Applied Mechanics (IUTAM) Symposium, Sydney, Australia, 1991, Banner, M.L., and Grimshaw, R.H.J., (eds), Springer-Verlag.

[28] Dold, J.W., and Peregrine, D.H., 1986, "An efficient boundary-integral method for steep unsteady water waves," Numerical Methods for fluid dynamics II, Edited by Morton, K.W., and Baines, M.J., Clarendon Press, Oxford.

[29] Dold, J.W., 1992, "An efficient surface-integral algorithm applied to unsteady gravity waves," Journal of Computational Physics, 103, pp. 90-115.

[30] Dold, J.W., and Peregrine, D.H., 1984, "Steep unsteady water waves: An efficient computational scheme," Proceedings of 19th International Conference on Coastal Engineering, Houston, pp. 955-967.

[31] Romate, J.E., 1989, "The numerical simulation of nonlinear gravity waves in three dimensions using a higher order panel method," Zandbergen, P.J., (advisor), PhD thesis, University of Twente, Netherlands.

[32] Broeze, J., 1993, "Numerical modelling of nonlinear free surface waves with a 3D panel method," Zandbergen, P.J., (advisor), PhD thesis, University of Twente, Netherlands.

[33] Daalen van, E.F.G., 1993, "Numerical and theoretical studies of water waves and floating bodies," Zandbergen, P.J., (advisor), PhD thesis, University of Twente, Netherlands.

[34] Berkvens, P.J.F., 1998, "Floating bodies interacting with water waves-development of a time-domain panel method," Zandbergen, P.J., (advisor), PhD thesis, University of Twente, Netherlands.

[35] Liu, P.L.F., and Hsu, H.W., 1992, "Applications of boundary integral equation methods for two-dimensional non-linear water wave problems," International Journal for numerical methods in fluids, 15, pp. 1119-1141.

[36] Cooker, M.J., 1990, "A boundary-integral method for water wave motion over irregular beds," Engineering Analysis with Boundary Elements, 7(4), pp. 205-213.

[37] Lin, W.M., 1984, "Nonlinear motion of the free surface near a moving body," PhD thesis, Department of Ocean Engineering, Massachusetts Institute of Technology.

[38] Vinayan, V., and Kinnas, S.A., 2007, "A BEM for the propagation of Nonlinear Planar Free-surface Waves," Electronic Journal of Boundary Elements, 5(1), pp. 17-40.

[39] Colicchio, G., Greco, M., and Faltinsen, O.M., 2006, "A BEM-level set domain-decomposition strategy for non-linear and fragmented interfacial flows," International Journal for Numerical Methods in Engineering, 67, pp. 1385-1419.

[40] Grilli, S.T., Guyenne, P., and Dias, F., 2001, "A fully nonlinear model for three-dimensional overturning waves over arbitrary bottom," International Journal for Numerical Methods in Fluids, **35**(7), pp. 829-867.

[41] Brandini, C., and Grilli, S.T., 2001, "Modeling of freak wave generation in a 3D-NWT," In Proc. 11th Offshore and Polar Engng. Conf. (ISOPE01, Stavanger, Norway, June 2001), Vol III, pp. 124-131.

[42] Grilli, S.T., Vogelmann, S., and Watts, P., 2002, "Development of a 3D numerical wave tank for modeling tsunami generation by underwater landslides," Engineering Analysis with Boundary Elements, **26**(4), pp. 301-313.

[43] Fochesato, C., Grilli, S.T., and Guyenne, P., 2005, “Note on non-orthogonality of local curvilinear coordinates in a three-dimensional boundary element method,” International Journal for Numerical Methods in Fluids, **48**, pp. 305-324.

[44] Fochesato, C., and Dias, F., 2006, “A fast method for nonlinear three-dimensional free-surface waves,” Proceedings of the Royal Society, London, **A 462**, pp. 2715–2735.

[45] Guyenne, P., and Grilli, S.T., 2006, “Numerical study of three-dimensional overturning waves in shallow water,” Journal of Fluid Mechanics, **547**, pp. 361-388.

[46] Fochesato, C., Grilli, S.T., and Dias F., 2007, “Numerical modeling of extreme rogue waves generated by directional energy focusing,” Wave Motion, **44**, pp. 395-416.

[47] Sung, H.G., and Grilli, S.T., 2008, “BEM Computations of 3D Fully Nonlinear Free Surface Flows Caused by Advancing Surface Disturbances,” International Journal of Offshore and Polar Engineering, **18**(4), pp. 292-301.

[48] Grilli, S.T., Dias, F., Guyenne, P., Fochesato, C., and F. Enet, 2010, “Progress In Fully Nonlinear Potential Flow Modeling Of 3D Extreme Ocean Waves,” Chapter 3 in Advances in Numerical Simulation of Nonlinear Water Waves, Vol. 11 in Series in Advances in Coastal and Ocean Engineering, World Scientific Publishing Co. Pte. Ltd., pp. 75-128.

[49] Grilli, S.T., Skourup, J., and Svendsen, I.A., 1989, “An efficient boundary element method for nonlinear water waves,” Engineering Analysis with Boundary Elements, 6(2), pp. 97-107.

[50] Grilli, S.T., and Subramanya, R., 1996, “Numerical modeling of wave breaking induced by fixed or moving boundaries,” Computational Mechanics, 17, pp. 374-391.

[51] Grilli, S.T., 1993, “Modeling of nonlinear wave motion in shallow water,” Computational methods for free and moving boundary problems in heat and fluid flow, Wrobel, L.C., and Brebbia, C.A., (editors), Computational Mechanics Publications, Southampton, Boston.

[52] Grilli, S., and Svendsen, I.A., 1990, "Corner problems and global accuracy in the boundary element solution of nonlinear wave flows," Engineering Analysis with Boundary Elements, 7(4), pp. 178-195.

[53] Grilli, S.T., Voropayev, S., Testik, F.Y., and Fernando, H.J.S., 2003, "Numerical Modeling and Experiments of Wave Shoaling over Buried Cylinders in Sandy Bottom," In Proc. 13th Offshore and Polar Engineering Conference (ISOPE03, Honolulu, USA, May 2003), pp. 405-412.

[54] Svendsen, I.A., Otta, A.K., and Grilli, S.T., 1992, "Unsteady free surface waves," Breaking waves, International Union of Theoretical and Applied Mechanics (IUTAM) Symposium, Sydney, Australia, 1991, Banner, M.L., and Grimshaw, R.H.J., (eds), Springer-Verlag.

[55] "ISEC Community Workshop: Simulation & Large-Scale Testing of Nearshore Wave Dynamics," July 8-10, 2009- Corvallis, Oregon (http://isec.nacse.org/workshop/isec_workshop_2009/ ; http://isec.nacse.org/workshop/isec_workshop_2009/benchmarks.html).

[56] Swigler, D.T., 2009, "Laboratory study investigating the three-dimensional turbulence and kinematic properties associated with a breaking solitary wave," M.S. thesis, Department of Civil Engineering, Texas A&M University.

[57] Chapman, B., et al., 2007, "Using OpenMP: Portable shared memory parallel programming," MIT Press.

[58] Rankin, W.T., 1999, "Efficient Parallel Implementations of Multipole based N-Body algorithms," PhD Thesis, Dept. of Electrical Engineering, Duke University.

[59] Hallquist, J.O., 2006, "LS-DYNA theory manual," Livermore Software Technology Corporation, Livermore, CA.

[60] "LS-DYNA keyword user's manual (volumes I and II), version 971," 2007, Livermore Software Technology Corporation, Livermore, CA.

[61] Grilli, S.T., 1997, "Fully nonlinear potential flow models used for long wave runup prediction," Long-Wave Runup Models (eds. Yeh, H., Liu, P., and Synolakis, C.,), World Scientific Publishing, Singapore, pp. 116-180.

[62] Hughes, S.A., 1993, “Physical models and laboratory techniques in coastal engineering,” World Scientific Publishing.

[63] Goring, D.G., 1978, “Tsunamis- The propagation of long waves onto a shelf,” W.M. Keck Laboratory of Hydraulics and Water Resources, Division of Engineering and Applied Science, California Institute of Technology, Pasadena, California, Report no. KH-R-38.

[64] Abramowitz, M. and Stegun, I.A., 1972, “Handbook of Mathematical functions,” Dover Publications, New York.

[65] “Shore Protection Manual, Volume I,” 1984, Department of the Army, Waterways Experiment Station, Corps of Engineers, Coastal Engineering Research Center.

[66] Fenton, J.D., 1985, “A fifth-order Stokes theory for steady waves,” J. Waterway, Port, Coastal and Ocean Engineering, 111(2), pp. 216-234.

[67] Lakhan, C., 1982, “Calculating modulus k of Cnoidal waves,” J. Waterway, Port, Coastal and Ocean Engineering, 108(3), pp. 435-438.

[68] Horikawa, K (ed)., 1988, “Nearshore dynamics and coastal processes,” University of Tokyo Press.

[69] Dean, R.G., and Dalrymple, R.A., 2000, “Water wave mechanics for engineers and scientists,” World Scientific Publishing.

[70] “Tsunami Wave Basin (TWB) Experiment Notebook,” College of Engineering, NACSE, Oregon State University: Tsunami wave impact forces on cylinders (Phase1Exp2: run5, run6, run11, run17 and run18), (http://neespop.wave.oregonstate.edu/nees/ui/index.php?maintab=200&tab=0&proID=21&expID=72).

[71] Nimmala, S.B., Yim, S.C., and Cheung, K.F., 2006, “Tsunami design criteria for coastal infrastructure: A case study for Spencer Creek Bridge, Oregon,” Oregon Department of Transportation (ODOT) (http://www.oregon.gov/ODOT/TD/TP_RES/docs/Reports/Tsunami.pdf).

[72] Boon-intra, S., M.S. thesis (in progress), Department of Civil and Construction Engineering, Oregon State University.

[73] Tsunami wave generator operations manual, 2003, Oregon State University, prepared by MTS Systems Corporation.

[74] DARPA (Defense Advanced Research Projects Agency) High Productivity Computer Systems (http://www.highproductivity.org/).

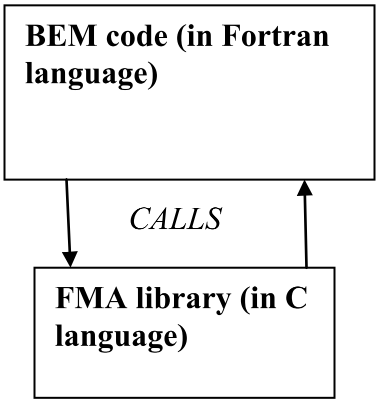
[75] The Chapel Parallel Programming Language (http://chapel.cray.com/).

[76] xd3d: Scientific visualization tool (http://www.cmap.polytechnique.fr/~jouve/xd3d/).

Appendices

Appendix-A: Algorithms of the 3D-NWT

The structure of 3D-NWT is as follows:



A high-level description of the algorithm (as extracted from the implementation) is provided below:

Two equations of BEM are solved- one for PHI, PHIN ($\phi, \frac{\partial\phi}{\partial n}$) and another for PHIT, PHITN ($\frac{\partial\phi}{\partial t}, \frac{\partial^2\phi}{\partial t\partial n}$). From the second loop (or time step) onwards, the latter equation uses the same geometry as the first one so the steps involved in FMA proceed quite faster.

1. Read the input parameters and generate the BEM mesh.
2. Initialize PHI, PHIN, PHIT, PHITN (*i.e.* $\phi, \frac{\partial\phi}{\partial n}, \frac{\partial\phi}{\partial t}, \frac{\partial^2\phi}{\partial t\partial n}$), velocities, pressures, DPRDT (i.e.,$\frac{Dp}{Dt}$), DUVWDT (i.e.,$\frac{Du}{Dt}, \frac{Dv}{Dt}, \frac{Dw}{Dt}$) for all nodes to zero.
3. Set IFLAGS=1 (request to perform BEM analysis using FMA).

4. If the current time step is less than the total termination time specified and also the current number of loops is less than the total number of loops specified, do the following (some of the steps below provide the necessary input to the subsequent steps):

4.1 Increment the loop number by one.

4.2 Do regrid as per the user-specified intervals of loop number.

4.3 Compute the current velocities over the Dirichlet (free surface) and moving (such as wavemaker) boundaries.

4.4 Compute PHIT $\left(i.e., \frac{\partial \phi}{\partial t} \right)$ by Bernoulli equation on all Dirichlet boundaries.

4.5 Set IFLAGV=2 (request to update boundary conditions).

4.6 Assign the Neumann boundary conditions at time t for impermeable and wavemaker boundaries; solve the equations using FMA and GMRES for PHIT, PHITN.

4.7 Calculate the accelerations- $\frac{Du}{Dt}, \frac{Dv}{Dt}, \frac{Dw}{Dt}$

4.8 Post-processing (calculate the results at internal points and gage points).

4.9 Compute the time derivative of pressure- $\frac{Dp}{Dt}$ on Dirichlet boundaries.

4.10 Update the geometry and potential of the Dirichlet boundaries based on second-order Taylor expansions.

4.11 Calculate the next time step and increment the time.

4.12 Set IFLAGV=1 (request to update both boundary conditions and geometry) and IFLAGS=1 (request to perform BEM analysis using FMA).

4.13 Solve PHI, PHIN using FMA and GMRES; update for the new boundary geometry (free surface, other boundaries) and the Neumann conditions (impermeable, wavemaker) at new time step.

4.14 Set IFLAGS=2 (BEM analysis using FMA is not required as current geometry is to be used, so execution is faster) and proceed to step 4.1 to solve for PHIT, PHITN.

Appendix-B: Build Instructions and Visualization Capabilities

The 3D-NWT code can be compiled using both Intel Fortran (ifort) and gcc compilers. It has also been found to compile successfully by using Intel C (icc) in the place of gcc. A makefile has automated the build process. Model-dependent parameters are set in two include files- param.inc and PMTA.h; the former is for the Fortran part of the code and the latter for the C part. With each new model, the parameters (such as the number of segments in directions X, Y, Z) in these two files are to be updated appropriately and the code is to be rebuilt. This is a required step because the Fortran part of the code uses static allocation of memory for its internal arrays.

Visualization tools are developed using xd3d [76], which can plot 2D and 3D meshes, contour plots, vector fields, 3D surfaces, etc. It also generates PostScript files for scientific publications. It is installed on the Linux platform for the visualization of the results of the 3D-NWT.

The 3D-NWT is modified to output the mesh information at each time step. A post-processing utility developed in Python reads in this data and writes in a format readable by xd3d. Another Python utility first calls xd3d to generate the PostScript files at specified intervals of time steps and then a tool- Convert (part of the ImageMagick tool; should be downloadable if not already available on the user’s Linux system) to transform these files into gif format. These gif images are joined together by a tool called Gifsicle (another free tool that is downloadable) to create an animation. On Linux platforms, web browser Firefox can be used to display this animation.

The wave generation module- Wavegen is written in C language and is compiled using gcc. It generates the piston motion and wave profiles for solitary wave, Cnoidal wave and Airy wave.

The source code of the Wavegen module, 3D-NWT code and the utilities in Python are archived in a .gz file (with gzip format) for running on a Linux system. This code is made available for use within the research group of my Advisor Prof. Solomon Yim. It will also be handed over to the code’s originator Prof. Stephan Grilli of the University of Rhode Island.

Appendix-C: Details of the I/O Files

Some of the details of the input and output (I/O) files are presented below for the convenience of the user. All the files are in ASCII (text) format, thus can be edited using any standard text editor. The input is also very minimal, often only a few (~2-10) lines. More detailed explanation of the parameters can be found from the sample input files and comments in the source code in the distribution package.

C.1 Wavegen module

This module requires the input file - wavegen.dat2, wherein the input is specified as the typeid (indicating the type of wave), water depth, wave height, time period (for periodic waves), acceleration due to gravity and the number of points of time history of piston motion.

Output is generated in the form of three files. For example, for a solitary or tsunami wave: tsunami_par.dat, tsunami_wave.dat and tsunami_wm.dat. These files contain the salient parameters computed, wave profile (time versus wave height) and piston motion (time versus piston displacement) respectively.

C.2 3D-NWT

This code requires the following input files:

input.dat2: Contains the number of segments of the BEM mesh in the X, Y, Z directions, dimensions of the wave tank, boundary condition indices of six faces of the tank, maximum time step, maximum loop number, fluid density, acceleration due to gravity, number of gages, number of internal points, options for grid generation, indices for periodic/focused wave, etc.

fmadata.dat2: Contains the parameters of the FMA such as cube size, number of levels, number of terms in the multipole expansion, etc.

piston.dat2: Time history of piston motion (time versus displacement, velocity and acceleration).

intern.dat2: The coordinates of the internal points (inside the domain) where the output is to be printed. This file is optional and is dependent on the user's choice of internal points in file input.dat2.

A few of the output files are highlighted below:

listing.dat: Lists out the data read in and the progress of execution.

gageb.dat, gagef.dat: Contain the time history of the coordinates, velocities, pressures at the bottom surface and the free surface respectively at the gage points specified.

intout.dat: Contains the coordinates, velocities, accelerations and pressures at each time step at the internal points specified.

tstep.dat: Contains the iteration number of the GMRES and the residual at each time step.

mesh.dat, mesh2dat: Files containing mesh data for post-processing.

The author found it helpful to redirect the terminal output to a log file and to submit the execution in the background. Error messages are typically printed out in the terminal log, files listing.dat and tstep.dat.

Appendix-D: Details of the FE Code Input

Following are some of the details of the finite element model (using LS-DYNA [59,60]) described in the second manuscript of this dissertation.

The Equation of State (EOS) to represent the behavior of water is input using the keyword *EOS_LINEAR_POLYNOMIAL with user-defined constants C_0, C_1, etc.

Pressure,

$$P = C_0 + C_1\mu + C_2\mu^2 + C_3\mu^3 + (C_4 + C_5\mu + C_6\mu^2) * E$$

where $\mu = \frac{\rho}{\rho_0} - 1 = \frac{V_0 - V}{V} = -\frac{\delta V}{V}$

ρ = current density

ρ_0 = nominal or reference density defined in *MAT_NULL card

E = internal energy per initial volume

V = current volume

V_0 = reference volume

Fluid properties of water are defined by the bulk modulus which gives the relation between the change of volume and pressure. For water, this relation can be assumed to be linear: $P = C_1 * \mu$, where C_1=K=bulk modulus of water=2300e+6 N/m^2 and the remaining coefficients C0, C_2, C_3, C_4, C_5, C_6 are specified to be zero in the above EOS. Card *MAT_NULL is specified with a mass density of 1000 kg/m^3 and a dynamic viscosity coefficient of 0.001 N.s/m^2.

The wavemaker is represented as a rigid body by using the keyword *MAT_RIGID. Air (located above the water part; needed in order to compute the free surface) is treated with the keyword *INITIAL_VOID_SET to simplify the calculations as its influence is very minimal.

The wavemaker's coupling with water is specified by the card *CONSTRAINED_LAGRANGE_IN_SOLID wherein the master part set includes water and air and the slave part includes the wavemaker. Number of coupling points (quadrature rule) distributed over each coupled lagrangian surface segment, NQUAD is specified to be 3 and CTYPE=4 for penalty coupling. *CONTROL_ALE is set so that the ALE (i.e., Arbitrary Lagrangian Eulerian) analysis is activated.

Appendix-E: Application of FSI to an ODOT Bridge

As part of the preparatory extension to the research work presented in this dissertation, the author and his Advisor Prof. Yim have studied the impact analysis of an ODOT (Oregon Department of Transportation) bridge subjected to a tsunami in collaboration with the University of Hawaii. This is a practical engineering application that required the use of multi-disciplinary interaction such as computational structural mechanics (CSM), computational fluid dynamics (CFD), ocean engineering, mathematics, computer science, etc.

The bridge is a replacement work planned on Highway 101, at Spencer Creek, Newport, Oregon. A variety of tsunami scenarios have been envisaged and both 2D and 3D models of the bridge were developed and analyzed. This study was reviewed and published online by the ODOT [71]. An extension to other locations of the bridges on the Oregon coast is in progress and is to be presented in an upcoming MS thesis [72]. In the future developments, the 3D-NWT code is envisaged to provide the input in terms of the wave height and the velocities to the codes such as LS-DYNA [59,60] that use some form of the Navier-Stokes model, including turbulence in future, to solve the fluid-structure interaction problems.