



Boundary Element Simulation of Linear Water Waves in a Model Basin

Clemens Hofreither

DK Computational Mathematics, University of Linz,
Altenberger Str. 69, 4040 Linz, Austria

Ulrich Langer

Institute of Computational Mathematics, University of Linz,
Altenberger Str. 69, 4040 Linz, Austria

Satyendra Tomar

Johann Radon Institute of Computational and Applied Mathematics
(RICAM), University of Linz,
Altenberger Str. 69, 4040 Linz, Austria

NuMa-Report No. 2009-02

February 2009

Technical Reports before 1998:

1995

- 95-1 Hedwig Brandstetter
Was ist neu in Fortran 90? March 1995
- 95-2 G. Haase, B. Heise, M. Kuhn, U. Langer
Adaptive Domain Decomposition Methods for Finite and Boundary Element Equations. August 1995
- 95-3 Joachim Schöberl
An Automatic Mesh Generator Using Geometric Rules for Two and Three Space Dimensions. August 1995

1996

- 96-1 Ferdinand Kickingger
Automatic Mesh Generation for 3D Objects. February 1996
- 96-2 Mario Goppold, Gundolf Haase, Bodo Heise und Michael Kuhn
Preprocessing in BE/FE Domain Decomposition Methods. February 1996
- 96-3 Bodo Heise
A Mixed Variational Formulation for 3D Magnetostatics and its Finite Element Discretisation. February 1996
- 96-4 Bodo Heise und Michael Jung
Robust Parallel Newton-Multilevel Methods. February 1996
- 96-5 Ferdinand Kickingger
Algebraic Multigrid for Discrete Elliptic Second Order Problems. February 1996
- 96-6 Bodo Heise
A Mixed Variational Formulation for 3D Magnetostatics and its Finite Element Discretisation. May 1996
- 96-7 Michael Kuhn
Benchmarking for Boundary Element Methods. June 1996

1997

- 97-1 Bodo Heise, Michael Kuhn and Ulrich Langer
A Mixed Variational Formulation for 3D Magnetostatics in the Space $H(\text{rot}) \cap H(\text{div})$ February 1997
- 97-2 Joachim Schöberl
Robust Multigrid Preconditioning for Parameter Dependent Problems I: The Stokes-type Case. June 1997
- 97-3 Ferdinand Kickingger, Sergei V. Nepomnyaschikh, Ralf Pfau, Joachim Schöberl
Numerical Estimates of Inequalities in $H^{\frac{1}{2}}$. August 1997
- 97-4 Joachim Schöberl
Programmbeschreibung NAOMI 2D und Algebraic Multigrid. September 1997

From 1998 to 2008 technical reports were published by SFB013. Please see

<http://www.sfb013.uni-linz.ac.at/index.php?id=reports>

From 2004 on reports were also published by RICAM. Please see

<http://www.ricam.oeaw.ac.at/publications/list/>

For a complete list of NuMa reports see

<http://www.numa.uni-linz.ac.at/Publications/List/>

Boundary Element Simulation of Linear Water Waves in a Model Basin

Clemens Hofreither¹, Ulrich Langer², and Satyendra Tomar³

¹ DK Computational Mathematics, JKU Linz
Altenberger Straße 69, 4040 Linz, Austria

² Institute of Computational Mathematics, JKU Linz
Altenberger Straße 69, 4040 Linz, Austria

³ Johann Radon Institute for Computational and Applied Mathematics (RICAM)
Altenberger Straße 69, 4040 Linz, Austria

Abstract. We present the Galerkin boundary element method (BEM) for the numerical simulation of free-surface water waves in a model basin. In this work, as a first step we consider the linearized model of this time-dependent three-dimensional problem. After time discretization by an explicit Runge-Kutta scheme, the problem to be solved at each time step corresponds to the evaluation of a Dirichlet-to-Neumann map on the free surface of the domain. We use the Galerkin BEM for the approximate evaluation of the Dirichlet-to-Neumann map. To solve the resulting large, dense linear system, we use a data-sparse matrix approximation method based on hierarchical matrix representations. The proposed algorithm is quasi-optimal. Finally, some numerical results are given.

1 Introduction

Numerical simulation of free-surface water waves in a model test basin is important when maritime structures, such as freight carriers, ferries, and oil rigs are tested on a model scale. Before the actual construction, the owners and designers need precise information about the hydrodynamic properties of their design. Several numerical algorithms have been proposed for such problems, see e.g. [8] (a combination of finite element method and finite difference method), or [7] (a complete discontinuous Galerkin finite element method). For such problems, only the discretization of the surface is of interest, which is a typical characteristic of the boundary element method (BEM). Unfortunately, the resulting system of linear equations from BEM is dense. On the other hand, while the finite element method (FEM) requires the discretization of the whole domain, the resulting matrices are sparse, and fast iterative solvers are available for these discrete systems. Thus, asymptotically, FEM has been considered the favorable choice as compared to BEM, at least, for three-dimensional problems, see e.g. [4]. However, data-sparse approximations of the dense BEM matrices can overcome this drawback of the BEM, see [2, 5] and the references therein.

In this paper, we present the Galerkin BEM for the numerical simulation of linear free-surface water waves in a model basin. We use a linearized model

for representing the dynamics of the water, which is introduced in Section 2. Its basic structure is that of an operator ordinary differential equation involving a Dirichlet-to-Neumann map on its right-hand side. For the evaluation of this operator, we employ a boundary element method, which is briefly described in Section 3. The resulting large, dense matrices are approximated via data-sparse H -matrix techniques, as outlined in Section 4. Finally, we present numerical results in Section 5 and conclude our findings in Section 6.

2 Modeling

We briefly outline the mathematical model used to describe the behavior of a model basin. The detailed derivation can be found in [9].

We start from the Navier-Stokes equations for an ideal, incompressible fluid. By assuming an irrotational flow in a simply connected domain, we may introduce a potential $\phi(x, y, z, t)$ for the velocity $v = \nabla\phi$. If the amplitude of the waves is small in comparison to the depth of the basin, we may linearize the problem; our computational domain $\Omega \subset \mathbb{R}^3$ is then time-independent. Without loss of generality, let Ω be oriented such that gravity applies in negative z direction with a magnitude g , and the free surface Γ_F of Ω is a subset of the plane at $z = 0$. The remainder of the surface, $\Gamma_N = \partial\Omega \setminus \Gamma_F$, represents the walls of the basin, where we prescribe a given Neumann boundary condition g_N , possibly time-dependent, for the potential ϕ . We denote the vertical perturbation due to waves at the free surface by a scalar function $\zeta(x, y, t)$ with $(x, y, 0) \in \Gamma_F$; see Figure 1 for a sketch.

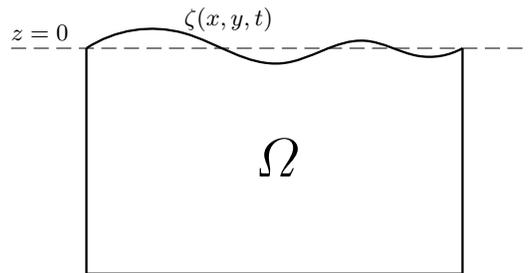


Fig. 1. A sketch of the free surface parametrization ζ .

Under these assumptions, we obtain the following system of equations. The first and the second condition on the free surface are called the *kinematic* and the *dynamic* boundary condition, respectively.

$$\left. \begin{aligned} \Delta\phi &= 0 & \text{in } \Omega, & \quad \frac{\partial\phi}{\partial n} = g_N(t) & \text{on } \Gamma_N, \\ \frac{\partial\zeta}{\partial t} = \frac{\partial\phi}{\partial n} & & \text{on } \Gamma_F, & \quad \frac{\partial\phi}{\partial t} = -g\zeta & \text{on } \Gamma_F. \end{aligned} \right\} \quad (1)$$

We now introduce the Dirichlet-to-Neumann map or Steklov-Poincaré operator $\mathcal{S}(t) : H^{1/2}(\Gamma_F) \rightarrow H^{-1/2}(\Gamma_F)$. For given Dirichlet data $g_D \in H^{1/2}(\Gamma_F)$, let $u \in H^1(\Omega)$ be the weak solution of the mixed boundary value problem

$$\Delta u = 0 \quad \text{in } \Omega, \quad \frac{\partial u}{\partial n} = g_N(t) \quad \text{on } \Gamma_N, \quad \text{and } u = g_D \quad \text{on } \Gamma_F. \quad (2)$$

We then define $\mathcal{S}(t)$ to be the Dirichlet-to-Neumann map

$$\mathcal{S}(t) : g_D \mapsto \frac{\partial u}{\partial n} \Big|_{\Gamma_F}.$$

Using this operator, the behavior of the model problem (1) on the free surface may be specified in the form of a system of two coupled operator ordinary differential equations,

$$\frac{d}{dt} \begin{pmatrix} \phi_F \\ \zeta \end{pmatrix} = \begin{pmatrix} 0 & -g \\ \mathcal{S}(t) & 0 \end{pmatrix} \begin{pmatrix} \phi_F \\ \zeta \end{pmatrix}, \quad (3)$$

where ϕ_F represents the trace of the potential ϕ on Γ_F .

As initial values, we use constant zero functions for both ϕ_F and ζ . In the context of our model problem, this corresponds to the free surface being undisturbed and at rest initially.

For the solution of this ODE system in a time interval $[0, T]$, we introduce a discretization of the time axis, $0 = t_0 < t_1 < \dots < t_N = T$, with the step sizes $\tau^{(n)} := t_{n+1} - t_n$. After spatial discretization, the Steklov-Poincaré operator \mathcal{S} exhibits moderate stiffness with a Lipschitz constant of the order $\mathcal{O}(h^{-1})$. The use of an explicit scheme for time integration, e.g. a classical fourth-order Runge-Kutta method, is thus justifiable as long as the time step size is not chosen too large.

For the discretization of the Steklov-Poincaré operator \mathcal{S} , we use a boundary element method, as motivated in the introduction.

3 The Boundary Element Method

The BEM operates only on the *Cauchy data*, that is, on the Dirichlet and Neumann traces on the boundary of the computational domain. The Cauchy data are related to each other via an integral equation on the boundary. This equation is then solved via e.g. a collocation or Galerkin approach. For a thorough treatment of the boundary element method, we refer the reader to e.g. [5].

For the remainder of this section, let u refer to the Dirichlet values of the solution of the PDE on the boundary Γ , and let $v := \frac{\partial u}{\partial n}|_{\Gamma}$ refer to its Neumann data.

With the help of a fundamental solution $E(x, y) = \frac{1}{4\pi|x-y|}$ of the Laplace equation in 3D, we now define the boundary integral operators

$$V : H^{-1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma), \quad K : H^{1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma), \quad D : H^{1/2}(\Gamma) \rightarrow H^{-1/2}(\Gamma)$$

called the *single layer potential operator*, *double layer potential operator* and *hypersingular operator*, respectively. We have the relation

$$\begin{pmatrix} u \\ v \end{pmatrix} = \mathcal{C} \begin{pmatrix} u \\ v \end{pmatrix} := \begin{pmatrix} \frac{1}{2}I - K & V \\ D & \frac{1}{2}I + K' \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}, \quad (4)$$

where the two-by-two block operator \mathcal{C} is called the *Calderón projector*.

Consider now the mixed boundary value problem (2) for a fixed t . We define extensions $\tilde{g}_D \in H^{1/2}(\Gamma)$ and $\tilde{g}_N \in H^{-1/2}(\Gamma)$ of the given boundary values and choose the ansatz

$$u = \tilde{g}_D + u_N, \quad v = \tilde{g}_N + v_D$$

with $u_N \in H^{1/2}(\Gamma)$, $v_D \in H^{-1/2}(\Gamma)$. Substituting this in (4) and restricting the equations to suitable parts of the boundary yields

$$Vv_D - Ku_N = \frac{1}{2}\tilde{g}_D + K\tilde{g}_D - V\tilde{g}_N \quad \text{in } H^{1/2}(\Gamma_D), \quad (5)$$

$$K'v_D + Du_N = \frac{1}{2}\tilde{g}_N - D\tilde{g}_D - K'\tilde{g}_N \quad \text{in } H^{-1/2}(\Gamma_N). \quad (6)$$

We then choose a trial space $\Lambda := \tilde{H}^{-1/2}(\Gamma_D) \times \tilde{H}^{1/2}(\Gamma_N)$, where

$$\begin{aligned} \tilde{H}^{1/2}(\Gamma') &:= \left\{ v = \tilde{v}|_{\Gamma'} : \tilde{v} \in H^{1/2}(\Gamma), \text{supp } \tilde{v} \subset \Gamma' \right\}, \\ \tilde{H}^{-1/2}(\Gamma') &:= \left(H^{1/2}(\Gamma') \right)' \end{aligned}$$

for any open subset $\Gamma' \subset \Gamma$. With an arbitrary test function $(s, t) \in \Lambda$, we multiply (5) by s and (6) by t in order to obtain a variational formulation.

We discretize Ω by a quasi-uniform and shape-regular triangulation with mesh size h . On this mesh, we define the space of piecewise linear and continuous functions $S_h^1(\Gamma_N) \subset \tilde{H}^{1/2}(\Gamma_N)$ on the Neumann boundary, and the space of piecewise constant functions $S_h^0(\Gamma_D) \subset \tilde{H}^{-1/2}(\Gamma_D)$ on the Dirichlet boundary. This gives us the natural choice $\Lambda_h := S_h^0(\Gamma_D) \times S_h^1(\Gamma_N) \subset \Lambda$ for a finite-dimensional trial space. We thus obtain the Galerkin variational formulation: Find $(v_{Dh}, u_{Nh}) \in \Lambda_h$ such that for all $(s_h, t_h) \in \Lambda_h$, there holds

$$a(v_{Dh}, u_{Nh}; s_h, t_h) = F(s_h, t_h), \quad (7)$$

with the bilinear form

$$a(v, u; s, t) = \langle Vv, s \rangle_{\Gamma_D} - \langle Ku, s \rangle_{\Gamma_D} + \langle K'v, t \rangle_{\Gamma_N} + \langle Du, t \rangle_{\Gamma_N}$$

and an analogous abbreviation $F(s, t)$ of the right-hand side. The resulting linear system may be solved by a suitable Krylov subspace method, e.g. MINRES.

4 Data-sparse approximation

The system matrices obtained from (7) are dense. By discretizing the boundary only, we obtain $N_{\text{BEM}} = \mathcal{O}(h^{-2})$ unknowns, resulting in a fully populated system

matrix with $\mathcal{O}(h^{-4})$ non-zero entries. Even with an optimally preconditioned iterative solver, the solution of the corresponding linear system will thus require at least $\mathcal{O}(h^{-4})$ arithmetical operations.

In contrast to the BEM, the FEM, where we have to discretize the entire domain, results in a system with $N_{\text{FEM}} = \mathcal{O}(h^{-3})$ unknowns. However, the finite element stiffness matrices are sparse, i.e. they have only $\mathcal{O}(h^{-3})$ non-zero entries. With an optimal preconditioner, a numerical solver with $\mathcal{O}(h^{-3})$ operations is attainable.

However, it is possible to avoid the difficulties associated with classical BEM discretization by using *data-sparse approximation* of the involved system matrices. In particular, we apply hierarchical matrix (or *H-matrix*) techniques to represent the system matrix. This results in a considerable reduction of the memory demand for matrix coefficients than the storage of the full matrix would require; see e.g. Bebendorf [2].

The core idea is to approximate a matrix $A \in \mathbb{R}^{m \times n}$ by a sum of outer products of vectors $u_i \in \mathbb{R}^m, v_i \in \mathbb{R}^n$, i.e. $A \approx \tilde{A} = \sum_{i=1}^r u_i v_i^T$. We call this a low-rank approximation of A with rank (at most) r . One way to construct such approximations is to compute a singular value decomposition of A and then discard all singular values which are below a certain threshold. This is called *truncated singular value decomposition*. While this produces optimal approximations in the spectral norm [2], it is quite slow. Therefore, faster methods have been developed in recent years. We only mention here the Adaptive Cross Approximation (ACA) method. Its concept is to construct \tilde{A} from a sum of *crosses*, that is, outer products of a column and a row of A . Once a desired error threshold is reached, the process is stopped. The approximation rank is thus determined adaptively.

Our BEM system matrices share the common property that they have a near-singularity along the diagonal. For such matrices, typically no suitable low-rank approximation with desirable accuracy exists. Thus, the algorithms described above fail to give useful results if applied to the entire matrix. Instead, it is required to split the matrix recursively into sub-matrices which may be better approximated. This is done by clustering the degrees of freedom which are represented by the matrix rows and columns into so-called *cluster trees*. Pairs of clusters are then chosen according to some admissibility condition for low-rank approximation. By this method, we obtain a so-called *H-matrix* approximation. Various matrix operations like addition, factorization or inversion may be generalized to *H-matrices*.

We refer the interested reader to the comprehensive monograph [2] for a detailed discussion of the techniques mentioned above. We also mention that software libraries which implement the methods sketched above are available, e.g. HLib [3] and AHMED [1]. The latter was used in our numerical experiments.

Data-sparse approximation is also used for preconditioning. From (7), we get the system of boundary element equations in the form

$$\begin{pmatrix} -V_h & K_h \\ K_h^T & D_h \end{pmatrix} \begin{pmatrix} \underline{v}_{Dh} \\ \underline{u}_{Nh} \end{pmatrix} = \begin{pmatrix} -\underline{f}_{Dh} \\ \underline{f}_{Nh} \end{pmatrix}.$$

As a preconditioner for this block matrix, we choose

$$C = \begin{pmatrix} I & 0 \\ -K_h^T V_h^{-1} & I \end{pmatrix} \begin{pmatrix} -V_h & K_h \\ 0 & D_h \end{pmatrix},$$

where hierarchical Cholesky factorizations of V_h and D_h are used to apply this preconditioner in an approximate way.

5 Numerical results

In the following, we extend a numerical example from [7] into three dimensions. We assume that we have a test basin with the dimensions $\Omega = (0, 10) \times (0, 1) \times (-1, 0)$. The free surface at rest is the quadrilateral $\Gamma_D = (0, 10) \times (0, 1) \times \{0\}$. On the remaining walls Γ_N , we prescribe the normal velocity $g_N(x, y, z, t)$ that is $(1 + z)a \sin(\omega t)$ for all $(x, y, z) \in \{0\} \times (0, 1) \times (-1, 0)$ and 0 otherwise. That is, the left wall of the basin is assumed to be equipped with a wave maker which exhibits periodic oscillations with maximum amplitude $a = 0.02$ and frequency $\omega = 1.8138$. The other walls are assumed to be stationary. Note that the oscillations exhibit maximum amplitude at the top of the basin and vanish at the bottom.

For time discretization, we use a fixed time step $\tau = 0.1$ over the time interval $[0, T = 80]$. This results in 800 time steps, each of which requires four solutions of the mixed boundary value problem in the domain Ω . The surface Γ is discretized by triangles using the software package NETGEN. We use a series of uniformly refined boundary meshes where each refinement step quadruples the number of triangles. The computations are performed on a machine with four Opteron-852 processors and 32 GB of RAM.

# triangles ($N = \mathcal{O}(h^{-2})$)	init	solve	total	ratio
704	9	20	29	-
2816	62	370	432	16.78
11264	520	3214	3734	8.78
45056	4162	20473	24635	6.60

Table 1. Performance of the algorithm for T=80

Table 1 summarizes the performance of our data-sparse BEM algorithm. The first column indicates the number of triangles in the boundary mesh. The second and third columns show the CPU time (in seconds) used for initialization and solution of the problem, while the fourth column shows total CPU time. Finally, the fifth column gives the ratio between total time for the current and the previous smaller problem.

Note that there is a significant cost for the generation and factorization of the system matrices. This however has to be performed only once at startup. The scheme is thus better suited for long simulations where many time iterations are to be performed. Also, parallelization was used for this initialization phase, so

actual measured times were lower. No appreciable parallelization overhead could be measured, so the CPU times given above can be taken as wall clock times for serial execution on one CPU.

As discussed at the beginning of Section 4, a FEM-based implementation of the Dirichlet-to-Neumann map would have $\mathcal{O}(h^{-3})$ unknowns and thus, if optimally preconditioned, a time complexity of $\mathcal{O}(h^{-3}) = \mathcal{O}(N^{3/2})$. Since evaluating the Dirichlet-to-Neumann map is the main bottleneck in the numerical simulation, we could then expect a constant ratio $4^{\frac{3}{2}} = 8$ in the last column of Table 1. The numbers we have obtained here thus suggest that our scheme may outperform a FEM-based approach for large problems.

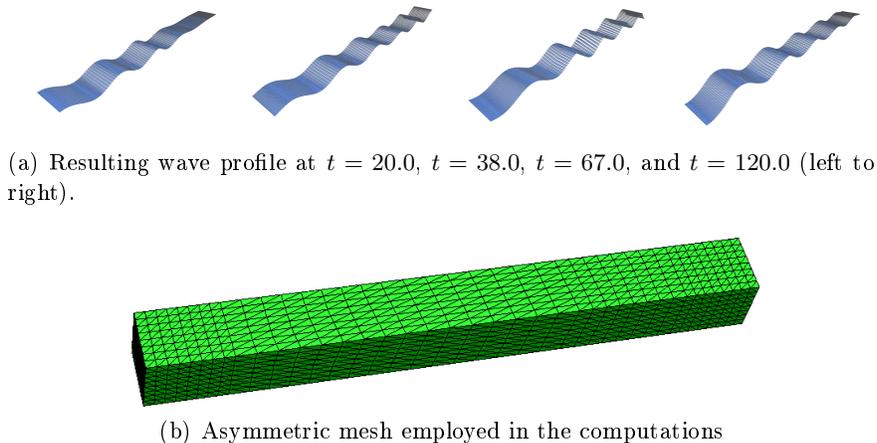


Fig. 2. Wave profiles computed on an asymmetric mesh

To investigate the stability of the proposed scheme over a long period of time, we performed the simulations up to $T = 120$ with $N = 11264$ and $N = 45056$ triangles, and did not face any stability problems like those reported in [8]. Figure 2(a) shows the computed wave profile with 45056 boundary triangles at times $t = 20.0, 38.0, 67.0$ and $t = 120.0$. At $t = 20.0$ the wave starts approaching the wall opposite to the wave maker, at $t = 38.0$ the wave gains full height, and $t = 67.0$ the wave reflected from the wall is affecting the pattern in the basin. The wave is also exhibiting a very nice pattern at $t = 120.0$, without any stability problems arising from the numerical scheme. Since no experimental data is available to validate our results (for a 3-D linear problem), we can rely only on visual results. However, they are in excellent agreement with a behavior similar to that of the 2-D problem reported in [7]. It is also important to note here that for a non-uniform/asymmetric mesh (Figure 2(b) shows the structure of the mesh employed in our computations) we do not need any artificial stabilization in our numerical scheme. This property is highly desirable for simulations over a

long period of time. A detailed analysis of the effect of mesh-asymmetry in the numerical computations was carried out in [6], and the resulting mechanism was used to stabilize the numerical scheme in [6, 7].

6 Conclusion

We have presented a numerical scheme for solving a linearized, time-dependent potential flow problem using the fourth-order explicit Runge-Kutta method and the boundary element method for the time and space discretizations, respectively. Data-sparse approximations of the resulting large-scale, dense matrices were used to make an efficient solution feasible. While this technique incurs a large one-time overhead for setting up the hierarchical matrices, it is worthwhile when many boundary value problems are to be solved on a fixed geometry, as the results in Section 5 indicate. For the same reason, however, a direct generalization to the case of a non-linearized, time-dependent domain seems to be more problematic at the first glance, since the system matrices would have to be recomputed at every iteration step. This problem may be alleviated by using the same preconditioner in every iteration and taking advantage of the fact that the matrix generation step may be trivially parallelized. The numerical scheme does not require a separate velocity reconstruction, or different order polynomials for velocity field and potential to preserve the accuracy in the wave height. A rigorous analysis of the numerical scheme proposed in this paper is still in progress.

References

1. M. Bebendorf. AHMED. Another software library on hierarchical matrices for elliptic differential equations, Universität Leipzig, Fakultät für Mathematik und Informatik.
2. M. Bebendorf. *Hierarchical Matrices*. Springer-Verlag Berlin Heidelberg, 2008.
3. S. Börm and L. Grasedyck. HLib. A program library for hierarchical and H^2 -matrices, Max Planck Institute for Mathematics in the Sciences, Leipzig.
4. X. Cai, H. P. Langtangen, B. F. Nielsen, and A. Tveito. A finite element method for fully nonlinear water waves. *J. Comput. Physics*, 143(2):544–568, 1998.
5. S. Rjasanow and O. Steinbach. *The Fast Solution of Boundary Integral Equations (Mathematical and Analytical Techniques with Applications to Engineering)*. Springer-Verlag New York, Inc., 2007.
6. I. Robertson and S. Sherwin. Free-surface flow simulation using hp /spectral elements. *J. Comput. Phys.*, 155(1):26–53, 1999.
7. S. K. Tomar and J. J. W. van der Vegt. A Runge-Kutta discontinuous Galerkin method for linear free-surface gravity waves using high order velocity recovery. *Comput. Methods Appl. Meth. Engrg.*, 196:1984–1996, 2007.
8. J.-H. Westhuis. *The Numerical Simulation of Nonlinear Waves in a Hydrodynamic Model Test Basin*. PhD thesis, Universiteit Twente, 2001.
9. J. Whitham. *Linear and Nonlinear Waves*. John Wiley, New York, 1974.

Latest Reports in this series

2009

- | | | |
|---------|---|---------------|
| 2009-01 | Clemens Pechstein and Robert Scheichl
<i>Scaling Up through Domain Decomposition</i> | January 2009 |
| 2009-02 | Clemens Hofreither, Ulrich Langer and Satyendra Tomar
<i>Boundary Elements Simulation of Linear Water Waves in a Model Basin</i> | February 2009 |

From 1998 to 2008 reports were published by SFB013. Please see

<http://www.sfb013.uni-linz.ac.at/index.php?id=reports>

From 2004 on reports were also published by RICAM. Please see

<http://www.ricam.oeaw.ac.at/publications/list/>

For a complete list of NuMa reports see

<http://www.numa.uni-linz.ac.at/Publications/List/>