

Parallel Efficiency of a Boundary Integral Equation Method for Nonlinear Water Waves

P. Strating

*Faculty of Applied Mathematics, University of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands*¹

and

P.C.A. De Haas

*Delft Hydraulics, P.O. Box 177, 2600 MH Delft, The Netherlands*²

Presenting and corresponding author: P. Strating

E-mail: P.Strating@math.utwente.nl

Phone: +31-(0)53-4893384

Fax: +31-(0)53-4894888

Abstract

We describe the application of domain decomposition on a boundary integral method for the study of nonlinear surface waves on water in a test case for which the domain decomposition approach is an important tool to reduce the computational effort. An important aspect is the determination of the optimum number of domains for a given parallel architecture. Previous work on heterogeneous clusters of workstations is extended to (dedicated) parallel platforms. For these systems a better indication of the parallel performance of the domain decomposition method is obtained because of the absence of varying speed of the processing elements.

Keywords: water waves, domain decomposition, boundary integral equation method

¹Part of this work was supported by the HPCN program NICE (Netherlands Initiative for CFD in Engineering)

²These investigations were supported by the Dutch Technology Foundation (STW)

1 Introduction

Knowledge about the propagation of water waves in near and offshore seas and in harbors is of importance in many studies in offshore and coastal engineering. The occurrence and growth of for example low frequency waves is of importance for vessels moored in a harbor which have their resonance frequency near the frequency of these long waves. Usually long waves are split into locked long waves which travel along with wave groups and free long waves which propagate with their own celerity. See e.g. Bowers [1].

The mechanisms behind the transformation of locked long waves into free long waves are not well understood yet and different approaches have been employed to understand this matter. An analysis based on a multiple-scales method has lead to the development of several formulations for the description of the transformation under influence of bottom topography and wave-wave interaction. One such model has been described by Dingemans et al. [2]. An alternative to this approach is the direct numerical simulation of nonlinear water waves. Such computations can be very valuable because they can provide verification material for other simplified models. See for example [3]. A promising numerical approach is the Panel Method reported by Broeze e.a. [4]. This method is based on a Boundary Integral Equation Method. Because many interesting problems involve very large domains, the simulation of free surface flows with the Panel Method is still time consuming, and the simulations are restricted to waves propagating in one direction only at present. For this reason, ways are sought to accelerate the computations. In previous work it was reported that domain decomposition by itself can speedup the calculations [5]. As an additional advantage, parallel processing can be introduced in a natural way. From the combination of both a super-linear speedup can be expected.

The computations presented here show a wave group signal which propagates with fixed form over an even bottom according to the third-order evolution equations described in [2]. The simulations with the present model originated from the wish to investigate to what extent this signal propagates undisturbed using the fully non-linear conditions, and to simulate the transformation of the wave group as it travels over a shoaling bottom. In this paper however, we discuss the parallel performance rather than the physical meaning of the computations.

From the viewpoint of parallel computing the problem is interesting because the number of subdomains influences the convergence of the solving method, and because there is no a priori obvious choice of the coupling of the subdomains. This is a common feature of many parallel computations, e.g. of multiblock solvers for stationary flows in computational fluid dynamics: to obtain flexible multiblock algorithms the communication between the blocks may be reduced, with the aim of saving more computation time than losing due to a slightly slower convergence rate. The present computations, done on both supercomputing parallel systems and on a cluster of workstations give an impression of this interplay between parallelism, convergence rate and interface couplings.

2 Mathematical Model and Numerical Method

In the simulation of three dimensional nonlinear water waves the water is usually assumed to be incompressible and inviscid, and the flow is assumed irrotational. Hence, the motion of the water can be described by a potential flow. This water wave problem is described by the field ϕ , from which the velocity field \mathbf{v} is derived as $\mathbf{v} = \nabla\phi$. The field ϕ satisfies Laplace's equation

$$\Delta\phi = 0, \quad (1)$$

and the boundary conditions on the free surface $\partial\Omega_{FS}$

$$\left. \begin{aligned} \frac{\partial\phi}{\partial t} + \frac{1}{2}(\nabla\phi)^2 + gz &= -\frac{p}{\rho} \\ \frac{D\mathbf{x}}{Dt} \cdot \mathbf{n} &= \frac{\partial\phi}{\partial n} \end{aligned} \right\} \mathbf{x} \in \partial\Omega_{FS} \quad (2)$$

and on the bottom $\partial\Omega_B$

$$\frac{\partial\phi}{\partial n} = 0, \quad \mathbf{x} \in \partial\Omega_B. \quad (3)$$

In addition, appropriate in- and outflow boundary conditions are formulated on the lateral boundaries.

The Panel Method is based on the work of Broeze [4] for the simulation of three-dimensional nonlinear water waves. The numerical method consists of a time marching scheme for the evolution of the free surface and the boundary conditions. At every time-step Laplace's equation for the velocity potential has to be solved. This is done with a Boundary Integral Equation Method. In the Boundary Integral Equation Method, Laplace's equation is solved by writing it as a set of integral equations over the boundary. Discretizing these integral equations and using the boundary conditions, a system of linear equations is built and subsequently solved. The solution provides the time derivatives in eqs. 2 which are needed for the time marching step.

Boundary Integral Equation Methods are very suitable for solving Laplace's equation on such domains because they only require a discretization of the boundary of the domain. Compared with field discretization methods, the advantages of a Boundary Integral Equation Method are a much smaller amount of grid points and a natural description of the free surface.

3 Domain Decomposition

For large-scale wave problems the solution algorithm for Laplace's equation is the bottleneck, the time marching scheme requiring only a minor part of the total CPU-time. The solution algorithm for Laplace's equation involves both the discretization of the boundary integral equations and the solution of the resulting system of linear equations. Furthermore, memory requirements for solving Laplace's equation depend quadratically on the number of grid points. These problems can be reduced considerably, both in two and three dimensions, by the use of domain decomposition.

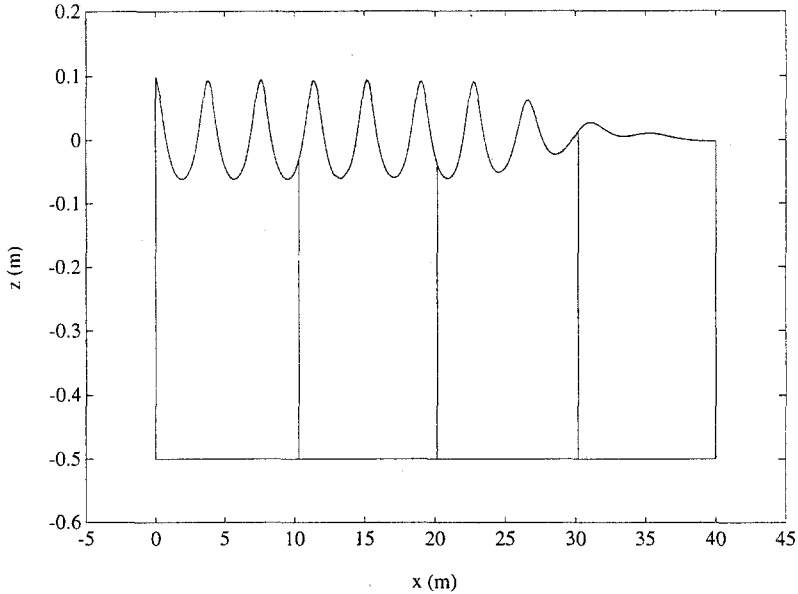


Figure 1: Decomposition of a 2d domain

The domain decomposition method described here consists of a division of the computational domain into subdomains (see Figure 1) and an iterative procedure which generates a sequence of solutions on the subdomains that converges towards the solution on the original domain. Every step of the iterative procedure consists of first solving Laplace's equation for the potential ϕ on the separate subdomains and then formulating new boundary conditions on the subdomain interfaces by using the solution in two adjacent subdomains. Only the latter part involves the actual exchange of information between the subdomains. Every odd step of the iterative procedure Dirichlet conditions are imposed on all interfaces and all even steps Neumann conditions are imposed. This scheme is also known as a Neumann-Neumann preconditioner in the context of domain decomposition methods for field discretization techniques. See e.g. [6].

The performance of the domain decomposition method is determined by the convergence of the iterative process. The convergence of the process can be judged by computing the jump across the interface between the solutions on both sides of each interface. The convergence on different interfaces can differ and depends on the geometrical form of the subdomains. This aspect has been subject of previous investigations [5] and the main conclusions given there are that 1) given a fixed number of subdomains, the convergence of the iterative procedure deteriorates as the length-to-height ratio of the subdomains decreases, and 2) the convergence of the iterative procedure deteriorates if there is more asymmetry near the interfaces due to a disturbed free surface or an uneven bottom, and 3) given a fixed length of the computational domain, the convergence of the iterative procedure deteriorates as the number of subdomains increases.

An interesting feature of the domain decomposition method emerges when considering subdomains with a fixed size. By increasing the number of subdomains, a larger computational domain is modelled. In [5] it was shown that the number of iterations required in the iteration process is limited and therefore the CPU time per subdomain is limited. This implies that the total computational time is proportional to the size of the full computational domain, so that the algorithm is scalable in the sense of Gustafson, i.e. with n processors we can solve an n -times bigger problem in the same amount of time. This may be important for realistic simulations.

It is noted that the domain decomposition method has also been developed for three-dimensional problems, where the domain subdivision is in one spatial direction [7]. Preliminary computations indicate that convergence of the iterative process is comparable with the convergence in two dimensions. More general cases will be the subject of future research.

4 First problem

4.1 Sequential performance, previous work

The efficiency of the domain decomposition for water wave problems has been shown in [5], where it was shown that for a two-dimensional problem a speedup factor of almost 6 could be obtained by using 12 subdomains. This speedup is due to the quadratic dependence of the computational costs on the size of a subdomain. Because of the deterioration of the convergence of the iteration process with increasing number of subdomains, there is an optimal number of subdomains in minimizing the computational costs of the original problem.

For a division into 1, 2, 4, 8 and 12 subdomains we have measured CPU-times and the average number of required iterations for a simulation of 200 time steps on one single HP workstation in Table 1. Although more iterations

# <i>subdomains</i>	# <i>iterations</i>	<i>CPU-time</i> (10^3 s)	<i>Speedup</i>
1	-	12.7	1
2	4.23	13.3	0.97
4	5.39	5.6	2.3
8	6.09	2.7	4.7
12	6.94	2.2	5.8
16	6.74†	0.98†	-

Table 1: Speedup due to the domain decomposition technique. The dagger denotes that the iteration process for the run with 16 subdomains diverged after 197 time steps (see text).

are required as the number of subdomains increases, the total required CPU-time decreases and gives a considerable speedup in the case of 4, 8 and 12 subdomains.

In the course of the simulation the number of required iterations increases due to the wave signal entering the subdomains. The asymmetry near the interfaces increases which slows down the convergence rate. Because of this, in combination with the small length-to-height ratio of the subdomains, the iterative process for 16 subdomains diverges after 197 time steps.

4.2 Parallel performance

The domain decomposition method is very suitable when using a parallel system of processors. The largest part of the computational work is done independently in the subdomains and the information volume which has to be communicated between subdomains is relatively small.

In [8] studies have been reported on the speedup and efficiency of a heterogeneous cluster of 5 HP workstations when using domain decomposition and explicit communication with the PVM library on a simulation of fixed size. Depending on the number of subdomains a reasonable speedup was achieved. These results, however, are not conclusive about the parallel efficiency of the algorithm because of the heterogeneity of the cluster. In the present study we will consider dedicated parallel platforms to establish the parallel efficiency as function of the number of subdomains and as a function of the number of processors.

The experiments described in this paper were done on a SGI Power Challenge computer. This system represents a modern RISK-processor based shared memory system with 10 processors. Because of the homogeneity of the systems, a partitioning into subdomains of equal size excludes the effect of load imbalance.

Each processor performs the computations for the subdomains that have been assigned to it. Global synchronization points occur in monitoring the convergence of the overall iterative process after each iteration step, and the determination of the (adaptive) time step size after each time step. Communication between the processors dealing with neighboring parts of the flow domain takes place every step of the iterative procedure, when constructing new boundary conditions for the subdomain interfaces.

The numerical problem of the previous section has been implemented with explicit message passing between the processing elements. Although on many supercomputers shared memory is available, message passing still is the most portable form of parallel computing. Both the PVM and MPI message passing libraries were used with comparable results. Through the use of generic send and receive instructions in the code, a quick change between the message passing libraries was possible. The generic send and receive commands in turn the appropriate PVM or MPI instructions.

In Table 2 we present performance results for the SGI Power Challenge with the MPICH message passing library. At the time of measurement, 8 processors were supposed to be dedicated for these computations. From the wall clock time for the 8 processor run, it was seen that not all processors were available at all times. For the calculation of the speedup the user times were used, because we

<i>#</i> <i>subdomains</i>	<i>#</i> <i>procs</i>	<i>Wall Clock</i> <i>time (s)</i>	<i>User time</i> <i>per proc (s)</i>	<i>Theor.</i> <i>speedup</i>
1	1	95.4	91.3	1.00
2	2	62.4	59.2	1.54
4	4*	34.3	15.5	4.27
8	-*	-	5.3	17.2
12	-*	-	3.4	26.9
16	-*	-	3.5†	26.1†

Table 2: Speedup due to the combination of domain decomposition and parallel processing on an SGI Power Challenge. The theoretical speedup is based on the user time for each subdomain reported by the system. The * denotes that the requested processors were not dedicated. The † value for the sixteen subdomain run was extrapolated from the 15 time steps run.

believe that these are indicative of the performance. If we compare the results with the results in Table 1 for the HP workstation, we see that the domain decomposition technique is less efficient on the SGI Power Challenge. This is probably because the problem under consideration is relatively small on the SGI computer. For this reason, and because this problem is not well suited for systems with larger numbers of processors, we will discuss in the following experiments for the larger problem that was mentioned in the introduction.

5 The traveling wave group problem

5.1 Sequential performance

We now discuss the problem that describes an almost stationary wave group traveling from left to right as depicted in Figure 2, over a total length of 5 km. During the simulation of this wave, smaller wave groups segregate from the main wave group as seen in the figure. Because the problem is very large (1600 panels at the free surface), the minimum subdomain number chosen here is as much as 25. For a number of subdomains (25, 32, 40, 50, 80, 100, 160 and 200), 20 time steps were taken. This number of time steps corresponds to a displacement of the main wave group of about one wavelength of the main wave group. Figure 3 clearly shows the effect of increasing the number of subdomains. An optimum number of subdomains, from the viewpoint of total computation time, is seen around a hundred subdomains.

5.2 Parallel performance

The above problem was parallelized using compiler directives in front of the loops over the subdomains. An analysis of the 50 subdomain problem showed that for these calculations almost 98 percent of the total computational work was done inside a parallelized loop. The remaining two percent is mostly initializing time of the program. Although the parallelized fraction is very high, it

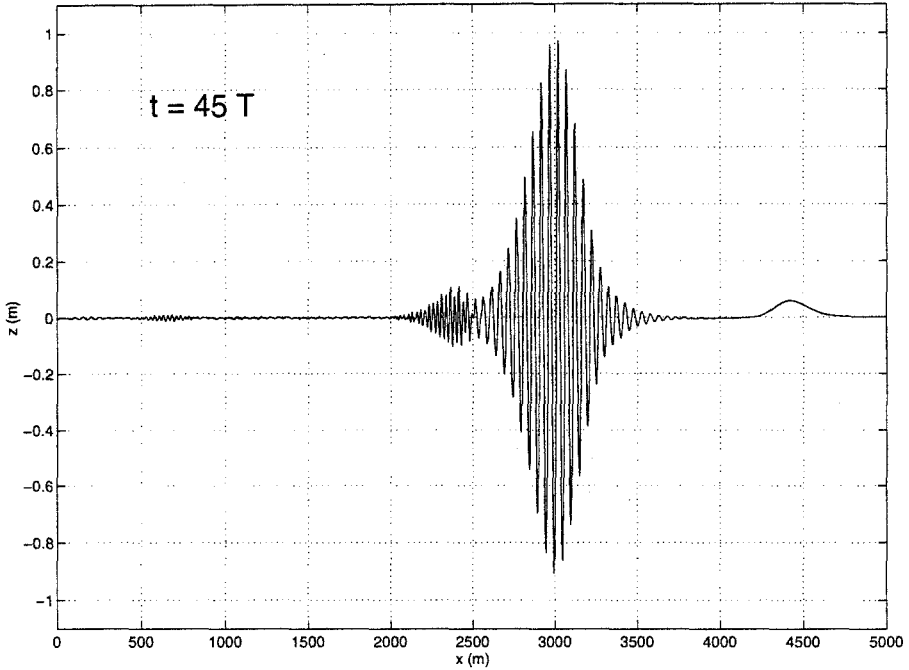


Figure 2: The shape of the wave during the simulation. Note the difference in scales for the axes.

should be noted that for the present computations Amdahl's law already limits the maximum achievable speedup to 7 for 8 processors.

For all subdomain divisions of the previous section, the program was run on 2, 4, 6 and 8 processors. The overhead generated to use more processors on the SGI Power Challenge is substantial, as is illustrated in Figure 4. These computations illustrate that the least expensive calculation is the single processor run with a hundred subdomains.

To see the parallel performance of our do-loop parallelized program we plot the speedup of the various subdomain divisions relative to their single processor runs (Figure 5). This figure shows that the parallel efficiency decreases if more subdomains are used. This is probably due to the decrease of the work done in smaller subdomains, and a relatively larger overhead because division of parallel loops takes place more often.

Finally we plot the overall performance as a speedup relative to the 'simplest' 25 subdomain, single processor run for all computed subdomain and processor numbers, the result of which is found in Figure 6. This plot shows that when the goal is to reduce the wall-clock time of the simulation, the 8 processor run with a hundred subdomains gives the fastest simulation.

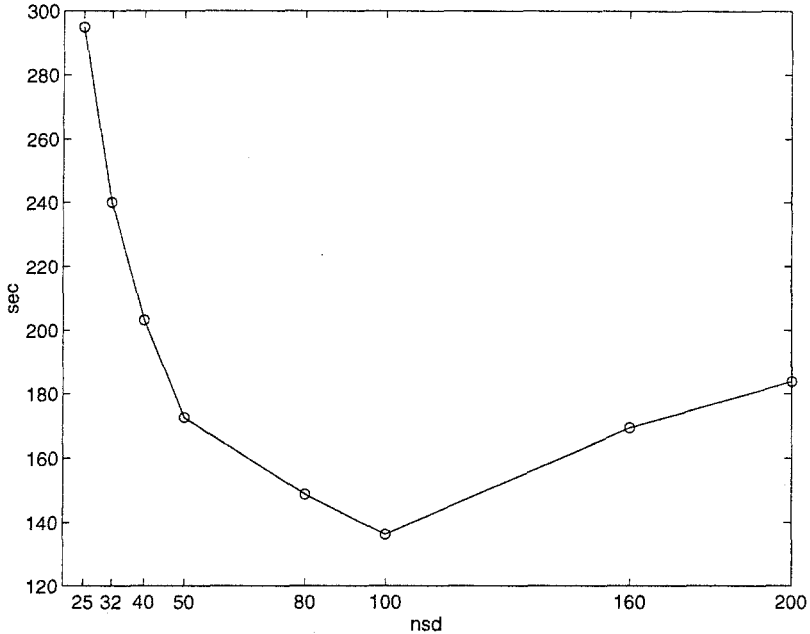


Figure 3: Computation time as a function of the number of subdomains.

6 Conclusions and Future Work

Domain decomposition has been proven to be a means to reduce the turn-around time of a nonlinear water wave simulation, based on a Boundary Integral Equation formulation. The domain decomposition alone can give a substantial speedup, which can be enhanced by parallel processing. It has been shown that for a given problem an optimum number subdomains exists that minimizes the total computational effort. The structure of the domain decomposed program allows very simple do-loop parallelization, while MIMD-like parallelization strategies with explicit message passing are also possible.

Because the iterative method is capable of dealing with large problems with large numbers of subdomains, the method may also be ported to Massively Parallel Platforms.

Future work will involve the convergence characteristics and a more detailed study of the message passing as compared to the shared memory do-loop parallelization. Then, when the two-dimensional problem is properly understood, three-dimensional problems will be considered.

References

- [1] E.C. Bowers, *Harbour resonance due to set-down beneath wave groups*, J. Fluid Mech. **79** (1), 1977, pp. 71-92.

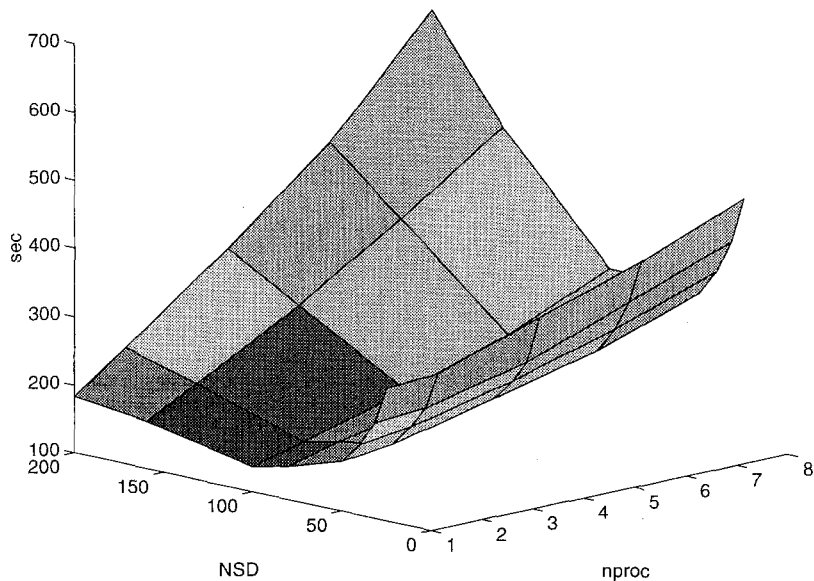


Figure 4: Total CPU time as a function of both the number of subdomains and the number of processors. Increasing the number of processors gives a computational overhead.

- [2] M.W. Dingemans, H.A.H. Petit, Th.J.G.P. Meijer and J.K. Kostense, *Numerical evaluation of the third-order evolution equations for weakly nonlinear water waves propagating over uneven bottoms*, in *Computer Modelling in Ocean Engineering 91*, Barcelona, 1991, pp. 361-370.
- [3] P.C.A. De Haas, M.W. Dingemans and G. Klopman, *Simulation of propagating nonlinear wave groups*, to appear in *Proc. 25th Int. Conf. on Coast. Eng.*, Orlando, 1997.
- [4] J. Broeze, E.F.G. Van Daalen and P.J. Zandbergen, *A three-dimensional panel method for nonlinear free surface waves on vector computers*, *Comp. Mech.* **13**, 1993, pp. 12-28.
- [5] P.C.A. De Haas and P.J. Zandbergen, *The application of domain decomposition to time-domain computations of nonlinear water waves with a panel method*, *J. Comp. Phys.* **129** No. 2, 1996, pp. 332-344.
- [6] P. Le Tallec, *Domain decomposition methods in computational mechanics*, *Comp. Mech. Adv.* **1**, 1994, pp. 121-220.
- [7] P.C.A. De Haas, J. Broeze and P.J. Zandbergen, *Proc. 15th Int. Conf. on Numerical Methods in Fluid Dynamics*, Monterey, 1996.
- [8] P.C.A. De Haas, J. Broeze and M. Streng, *Domain decomposition and parallel computing in a numerical method for nonlinear water waves*, *Proc. of the*

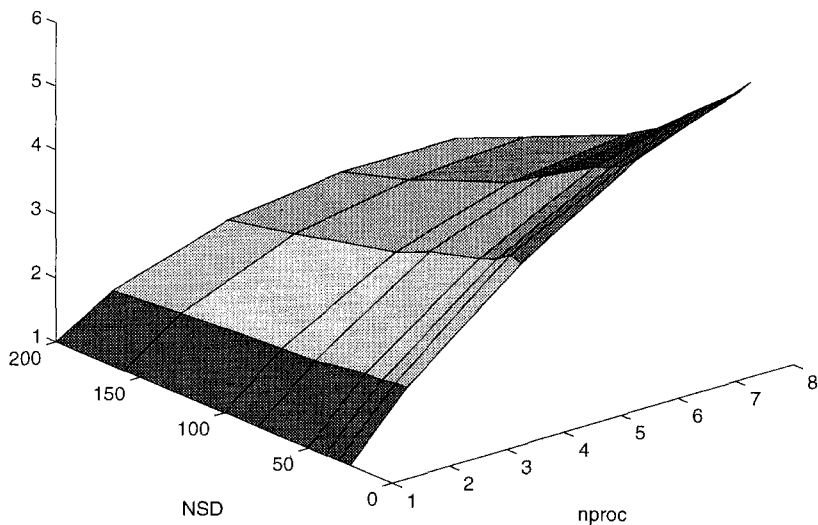


Figure 5: Speedup as compared to the runs with only one processor for all subdomain divisions.

EUROSIM 1996 Int. Conf., eds. L. Dekker, W. Smit and J.C. Zuidervaart, Delft, The Netherlands, 1996, pp. 437-444.

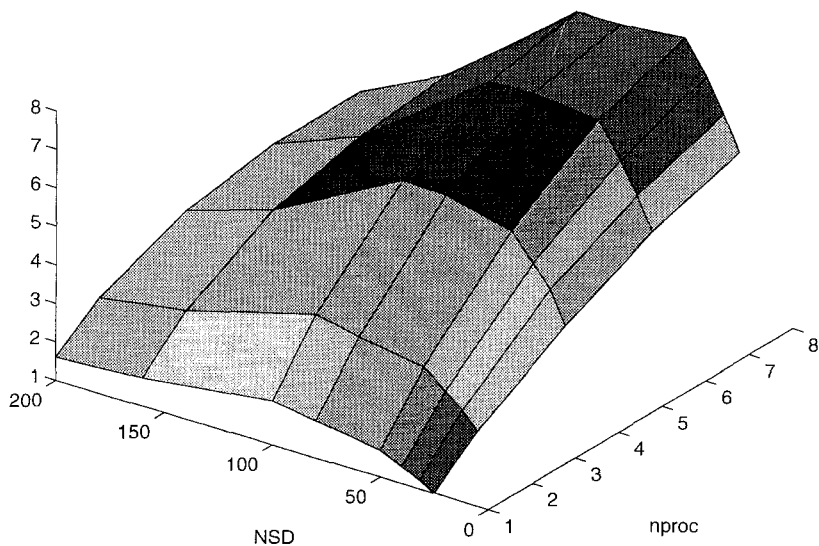


Figure 6: Speedup as compared to the 'smallest' run with 25 subdomains for runs up to 8 processors.