# Evaluation of a Vector Hypercube for Seismic Modelling

Rosemary Renaut<sup>†</sup> and Johnny Petersen<sup>‡</sup> Chr. Michelsen Institute

### Abstract

Seismic modelling to produce realistic seismic traces is a computationally intensive problem. A 2D synthetic wave propagation code using explicit finite differences with absorbing boundary conditions has been implemented on an Intel Hypercube with 32 processors. The algorithm is highly parallel with good load balancing between processors and speed up proportional to the number of processors being used. A vectorised version of the code has been used to evaluate the performance of the Intel IPSC-VX 5d vector hypercube. Mflop rates up to 71 are achieved.

## 1. Introduction.

Many geophysical problems, including the modelling of the Earth's crust and the discovery of gas and oil accumulations are problems in wave propagation (Kelly et. al. [4]). In recent years there has been great interest in the production of synthetic seismograms for complex subsurface geometries as a means of extracting and understanding fine detail in seismograms ([1], [3], [4], [6]). A realistic solution of the two-dimensional wave equation is the first step in the generation of a synthetic seismogram as an aid in the interpretation of seismic data.

The two-dimensional wave equation which describes the propagation of stress waves in a horizontally stratified elastic medium is easily solved by the method of finite differences. Numerical solutions have been obtained by

Present Address:

† Department of Mathematics, Arizona State University, Tempe, Arizona, USA

‡ Bergen Scientific Centre, Allégaten 36, N-5002 Bergen, Norway

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specfic permission.

© ACM 1988 0-89791-273-X/88/0007/1187 \$1.50

Alford et.al. [1] and Kelly et. al. [4]. For the generation of a synthetic seismogram an essentially infinite physical domain must be mapped onto a finite region restricted in size by the computer memory available. A solution over a timescale of seconds is required. Consequently, this problem is large scale, requiring intensive computation and large memory.

Such large scale problems are ideal candidates for parallelisation. Clayton [3] implemented an algorithm for the solution of the acoustic wave equation on a concurrent processor. Petersen and Renaut [6] have demonstrated that this algorithm can be used to solve a realistic problem and produce seismic traces.

In this paper we continue the work of Petersen and Renaut [6]. The algorithms for the solution of the wave equation by either a five-point or a nine-point stencil have been implemented on a hypercube, where each processor has a vector capability. The effects that the use of these stencils have on numerical solutions are determined by their dispersive and stability properties. Here we present the results of a theoretical analysis which show these properties. The characteristics can be compared by examining the numerical solutions.

In Section 3 we present our results. For comparison results without vectorisation are also presented. The code vectorized well. With a problem of maximal size a mflop rate of 71 is achievable. This compares favourably with smaller supercomputers, which are, however, more expensive than a hypercube. Our results also demonstrate that the nine-point stencil may be preferred to the five-point stencil. It is computationally more intensive and possesses better dispersive properties. The ratio of timestep to gridsize is also less restricted by the stability requirement. These characteristics mean that either larger problems can be solved or higher accuracy attained, than with the five-point stencil.

#### 2. The Acoustic Wave Equation.

The acoustic wave equation in two dimensions is

$$rac{1}{
ho}iggrappi^2 P = rac{1}{\kappa}rac{\partial^2}{\partial t^2}P$$

where  $\rho = \rho(x, y)$  is density,  $\kappa$  is bulk modulus and P = P(x, y, t). The velocity of the pressure wave in the medium is given by  $v = v(x, y) = \sqrt{\frac{\kappa}{\rho}}$ . This equation is solved numerically on a horizontally stratified domain  $350m \times 240m$  (cf. Figure 1). The initial condition is a shot explosion behind the boat which tows an array of hydrophones for picking up the trace data. As in Petersen and Renaut [6] the shot is modelled by a Gaussian pulse.

This scheme is said to be accurate of order two. It allows numerically stable solutions provided that the Courant number,  $\mu$ , is bounded

$$\mu = v \frac{\Delta t}{\Delta x} \le \frac{1}{\sqrt{2}}$$



Figure 1. The physical domain.

A finite difference scheme for the solution of this equation is obtained by substituting the standard forward and backward difference operators for the continuous derivatives. This leads to the traditional five-point stencil for the solution of the wave equation

$$(2.1)$$

$$P(x, y, t + \Delta t) = 2P(x, y, t) - P(x, y, t - \Delta t)$$

$$+ (\frac{v\Delta t}{h})^{2} [P(x + \Delta x, y, t)$$

$$+ P(x - \Delta x, y, t) + P(x, y + \Delta y, t)$$

$$+ P(x, y - \Delta y, t) - 4P(x, y, t)]$$

Here we assume that the grid is square and set  $\Delta x = \Delta y = h$ . The velocity within a layer is given by v = v(x, y) and the time increment is  $\Delta t$ . The points used to advance a point to the new time level are seen in Figure 2.



Figure 2. The five-point stencil.

An analysis of the dispersive properties of this method is given by Petersen and Renaut [6]. It propagates waves with phase and group velocities which are direction dependent. This dispersion can be controlled by using a small enough grid size allowing about 12 points per wavelength. The pulse used in our experiments has a wavelength of about 30m and so we require  $h \leq 2.5m$ . With h = 2.5m dispersion is less than 2% and the energy of the wave travels at about 97% of its correct value.

An alternative to the above five-point stencil is obtained by rotating the stencil  $45^{\circ}$  and using a weighted combination of the resulting stencil with 2.1. A second order accurate stencil using nine points at the k time level is obtained:

$$P(x, y, t + \Delta t) = 2P(x, y, t) - P(x, y, t - \Delta t) + \left(\frac{v\Delta t}{h}\right)^2 [(\beta - 1)(P(x + \Delta x, y, t)) + P(x - \Delta x, y, t) + P(x, y + \Delta y, t) + P(x, y - \Delta y, t) - 4P(x, y, t)) + \frac{\beta}{2}(P(x + \Delta x, y + \Delta y, t)) + P(x - \Delta x, y - \Delta y, t) + P(x - \Delta x, y - \Delta y, t) + P(x - \Delta x, y - \Delta y, t) - 4P(x, y, t))]$$

This stencil is shown in Figure 3.

this we have used the second-order accurate boundary conditions derived in Clayton and Engquist [2]. At the right hand boundary we model the equation

$$P_{xt}+\frac{1}{v}P_{tt}-\frac{v}{2}P_{yy}=0$$

An approximation of this on the discrete domain is given by the first order difference scheme:

$$D_x^- D_t^0 P_{Njk} + \frac{1}{2v} D_t^+ D_t^- (P_{Njk} + P_{N-1jk}) - \frac{v}{4} D_y^+ D_y^- (P_{N-1jk+1} + P_{Njk-1}) = 0$$

Here

$$P_{ijk} \simeq P(i\Delta x, j\Delta y, k\Delta t).$$

 $D_q^+, D_q^- \mbox{ and } D_q^0$  are the forward, backward and central difference operators.

This boundary scheme is stable for  $\mu < 1$  when implemented with either of the operators 2.1 or 2.2. A comparison of other suitable boundary operators has been performed by Renaut and Petersen [7].



Figure 3. The nine-point stencil.

Dispersion is minimised by using  $\beta = \frac{1}{3}$ . With this value for  $\beta$  the method is stable for  $\mu \leq \frac{\sqrt{3}}{2}$ . Dispersion is controlled by using at least 6 points per wavelength. For gridsize, h = 5m, dispersion is less than 1% and energy travels at 96% of its correct value.

As explained the physical domain is mapped onto a finite domain with boundaries that are artificial. Ideally, a solution computed on the finite region should coincide with the free space solution. Therefore, provided that there are no external mechanisms which cause reflection back into the domain, the artificial boundaries should simulate the outward radiation of energy. To achieve The finite difference method is implemented on a hypercube by dividing the spatial domain to be modelled into a regular grid of patches. As Clayton [3] each patch is mapped onto a processor of the cube which has also been arranged as a two-dimensional grid. Neighbouring patches have one layer of grid points overlapping on each side (cf. Figure 5). If a patch lies on the edge of the physical domain then only interior sides overlap. By overwriting the  $(k-1)^{st}$  time level with the updated  $(k+1)^{st}$  time level the pressure array need only be of size  $(nx+2) \times (ny+2) \times 2$ , where nx+2 and ny+2 are the number of points in the x and y directions respectively, per processor. The velocity distribution is stored in an array of size  $(nx+2) \times (ny+2)$ .



right hand boundary

Figure 4. Right hand boundary.



Figure 5. The node communication.

## 3. Results.

In order to compare the efficiency of the two methods we need a measure of the amount of work involved in each case. We count the number of floating point operations per iteration step. For (2.1) this gives

288nx ny + 88(nx + ny) + 48 + 680,

compared to

$$602nxny + 88(nx + ny) + 48 + 680$$

for (2.2). The major contribution here is from the calculation at the interior points. Work at the boundaries is less than 1% of the total for 0(nx) = 0(ny) = 100. It appears that the five-point stencil (2.1) is more than twice as fast. Results for a problem with nx = ny = 70,  $\Delta x = \Delta y =$ 2.5m and  $\Delta t = .00055$  for 4096 iterations are given in Table 1.

Scheme	Dimension	Time/	Total Time	mflop
1	of Cube	Iteration		Rate
5pt	5	.9995s	1 hr 8 min 14 s	1.41
9pt	5	1.6289s	1 hr 57 min 12 s	1.83
5pt	4	2.1758s	21 hr 28 min 32 s	.65
5pt	0	32.4531s	36 hr 55 min 46 s	.04

	Table	1.	Results	for	non-vectorized	code.
--	-------	----	---------	-----	----------------	-------

We notice that the scheme (2.1) does not live up to expectations. It is only 1.63 times faster than (2.2). The communication time, however, is the same in either case and thus is the cause of the discrepancy with the theoretical prediction. Greater than linear speedup is attained using the 5-d cube as compared with the 0d cube. This phenomenum which is an effect of the 80286 chip is fully explained in Petersen and Lindheim [5]. Obviously the program is highly parallel with good load balancing.

Using the Intel hypercube it is possible to apply the preprocessor to vectorise the code. We discovered, however, that in this case more efficient code was obtained by carrying out the vectorisation by hand. Better results are achieved by considering the pressure array as a long vector, thus enabling vectorisation of both loops of the iteration. Results are presented in Table 2. In this case we used an array of maximum size, nx = ny = 190.

Scheme	Dimension	Time/	15 min 54 s	mflop
	of Cube	Iteration		Rate
5pt	5	.2329	15 min 54 s	44.63
9pt	5	.3060	20 min 53 s	71.73
5pt	4			18.88
9pt	4			36.16
9pt	3			18.5

Table 2. Results for vectorized code.

Almost linear speedup is obtained with higher dimensional cubes. The advantage of (2.1) is now even less apparent as it is only 1.31 times faster than (2.2). We performed the same problem but without calculation. This required 5 mins 52 secs. Thus one third of the time is required for communication with (2.1) as compared to one quarter for (2.2).

In the experiments we performed the gridsize was the same for both (2.1) and (2.2). As explained, however, in Section 2 the scheme (2.2) should perform equally well as (2.1) on a grid twice as coarse. Also the bound on Courant number is higher so that larger time steps can be used. Therefore, we can expect to solve a problem to a specified accuracy with less work using (2.2). A grid twice as coarse reduces the computation by a factor 4. As some portion of the time is involved in communication and as vector lengths are reduced, the timings will not actually be reduced by this factor. Whatever, the ninepoint stencil will outperform the five-point stencil. More importantly, in this context, scheme (2.2) can be used to model larger problems. In fact problems four times as large with accuracy comparable to that attained by (2.1) on the original problem can be modelled. The extra computation required for (2.2) may explain why it is not much used in practice. In a parallel setting, however, where communication takes a large proportion of the time, the advantages of (2.2) are more apparent.

In either case the added capability of the vector processor evidently improves the algorithm's performance. Computer time is substantially reduced. Synthetic seismograms can be produced on a realistic timescale. A vector hypercube is a viable alternative to a more expensive supercomputer.

#### 4. References

[1] R. M. Alford, K. R. Kelly and D. M. Boore, Accuracy of finite-difference modelling of the acoustic wave equation, Geophysics 39 (1974), pp. 834-842.

[2] R. Clayton and B. Engquist, Absorbing boundary conditions for acoustic and elastic wave equations, Bull. Seismol. Soc. Amer., 67 (1977), pp. 1529-1540. [3] R. W. Clayton, Finite difference solutions of the acoustic wave equation on a concurrent processor, Seismological Laboratory, Caltech, Pasadena (1986).

[4] K. R. Kelly, R. W. Ward, S. Treitel, and R. M. Alford, Synthetic seismograms: a finite difference approach, Geophysics 41 (1976), pp. 2-27.

[5] J. Petersen and J. de Pillis-Lindheim, Simulation of larger dimensions on the Intel IPSC Hypercube, Chr. Michelsen Inst., Bergen, Norway (1987). [6] J. Petersen and R. A. Renaut, Synthetic 2D-seismic wave propagation using a hypercube parallel computer, presented at the 49th Meeting of the European Association of Exploration Geophysicists, June 1987, Yugoslavia. Accepted, Geophysical Transactions of Hungary.

[7] R. A. Renaut and J. Petersen, *Absorbing boundary* conditions for the wave equation, presented at the SIAM 35th Anniversary Meeting, October, 1987, Denver.