# Paper II

# Simulation of ground vibration on the CM-200

M. Berglund.

Center for Computational Mathematics and Mechanics ($C^2M^2$)
Royal Institute of Technology, S-100 44 Stockholm

## 2.1 Introduction

The rhythmic motion of a crowd of rock fans in a large sports arena in Gothenburg, Sweden, lead to violent structural vibrations resulting in structural damage. The incident has intensified research in elastic wave propagation in soil materials such as is found in the foundation.

The wave propagation in soil is approximated by a three dimensional linearly elastic model in which the modulus of elasticity, $E$, varies with depth, *i.e.* $E = E(x_3)$. The density $\rho$ and Poisson's ratio $\nu$ are constant. The damping is assumed to follow the common Rayleigh model with coefficients $\gamma$ and $\kappa$. This gives opportunities to study both viscous and hysteretical types of damping by just changing the coefficients $\gamma$ and $\kappa$. Thus, the governing equations are

$$\sigma_{ij,i} + K_j + \gamma \frac{\partial}{\partial t}\sigma_{ij,j} + \kappa \frac{\partial u_j}{\partial t} = \rho \frac{\partial^2 u_j}{\partial t^2}, \quad j = 1, 2, 3, \tag{1}$$

in which

$$\sigma_{ij} = \frac{E}{1+\nu}\left(\varepsilon_{ij} + \frac{\nu}{1-2\nu}\varepsilon_{kk}\delta_{ij}\right), \tag{2}$$

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{3}$$

where $\sigma_{ij}$ is the stress tensor, $u_j$ the j:th component in the displacement field and $K_j$ are the external volume forces. Equation 1 is solved in the space domain

$$\Omega = \{(x_1, x_2, x_3) : 0 \le x_1 \le M_1, 0 \le x_2 \le M_2, h(x_1, x_2) \le x_3 \le 0\},$$

and $t \ge 0$, see figure 1.

The boundary condition on the upper surface, $x_3 = 0$, is

$$\sigma_{i,j} = f_j(x_1, x_2, t), \tag{4}$$

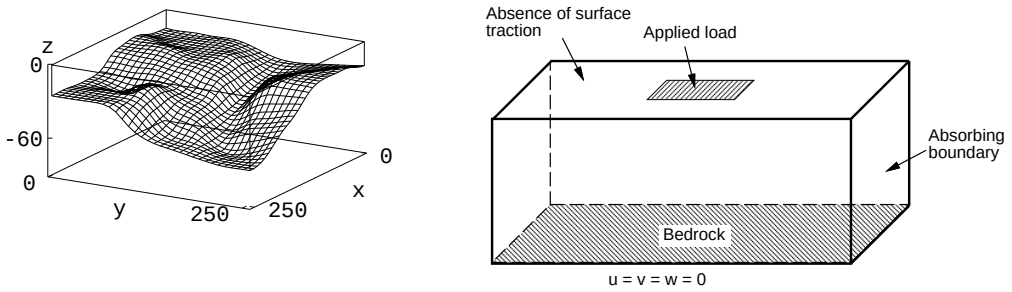


Figure 1: The physical domain and the computational domain with the different boundary conditions

which expresses the prescribed surface tractions. If $f_j = 0$ for some $j$ then we have absence of surface tractions in that direction. At the bedrock boundary $x_3 = h(x_1, x_2)$ displacements are zero, *i.e.*

$$u_i(x_1, x_2, h(x_1, x_2), t) = 0. \tag{5}$$

On the remaining faces of $\Omega$ absorbing boundary conditions are applied, which are paraxial approximations of order 1 of equation 1, see [CE] for details. The initial conditions for this problem are

$$u_i(x_j, 0) = f_0(x_j),$$
$$\frac{\partial}{\partial t} u_i(x_j, 0) = g_0(x_j).$$

This problem is of general interest in soil dynamics with all kinds of site response as application, such as earthquakes, machine vibrations, pile drivings and blastings. Also effects of moving loads like trains and trucks where the three dimensional effects are of great interest can be analyzed.

The choice of algorithm is guided by the demand of high accuracy within a realistic computation time. This is achieved by a compromise in which the computing stencil and data structure are simple enough for good flop-rate on a parallel computer but general enough for numerical accuracy. The computational domains we have in mind can be transformed by a smooth mapping to a parallelepiped. This makes it possible to use a structured regular grid for the unknowns $u_i$. The computational stencil is explicit, *i.e.* the new value of an unknown displacement in a time-step is given by a linear combination of known old-time values at neighbors in coordinate and diagonal directions. The same structure is used for the different boundary conditions. Second order finite differences are used on transformed equations in the interior of the domain. The absorbing boundary conditions are of first order numerical accuracy which is enough for overall second order accuracy [G].

The program ELWA has been implemented in an array extension of FORTRAN 90, CM Fortran, on a CM-200 with 256 floating point units (FPU) and 1 Gbyte RAM [T1]. The current performance is 436 Mflops in single precision with 128 FPU and a VP-ratio of 2048. The program has been validated by comparing results with an analytic solution of a one dimensional problem and with the finite-element package ABAQUS [A] in a two dimensional problem. ELWA has been used by S. Erlingsson in his thesis [E].

## 2.2 Numerical algorithm

Let the physical domain be described in the coordinates $(x_1, x_2, x_3)$ and the computational domain in $(x, y, z)$. Due to the simple geometry of the physical domain we can transform by a smooth mapping

$$x = x_1,\ y = x_2,\ z = x_3/h(x_1, x_2), \tag{6}$$

the physical domain into a parallelepiped, which becomes the computational domain, see figure 1. Using the chain rule, the equation (1) is written in the new coordinates as,

$$\frac{\partial^2}{\partial t^2}u_j = P_1(x,y,z)\frac{\partial^2 u_1}{\partial x^2} + P_2(x,y,z)\frac{\partial^2 u_2}{\partial x^2} + \ldots, \tag{7}$$

where $P_r(x, y, z)$, $r = 1, 2, 3$, are functions of $h(x, y)$, $E(z)$ and their partial derivatives of first and second order. Let $\mu^n(i, j, k)$ be the numerical approximation of $u_r(x, y, z, t)$, where $r = 1, 2, 3$, $x = i\Delta x$, $y = j\Delta y$, $z = k\Delta z - L_z$, $t = n\Delta t$, $\Delta x = L_x/N_x$, $\Delta y = L_y/N_y$, $\Delta z = L_z/N_z$, $\Delta t$ is the time-step and $N_x$, $N_y$ and $N_z$ are the number of grid nodes in each direction in the three dimensional (regular) grid. Let inner nodes be the nodes where $\mu^n$ is determined by the discretization of the equation (7) and let the boundary nodes be the nodes where $\mu^n$ are determined by the discretization of the boundary conditions. Let inner nodes be the nodes where $\mu^n$ is determined by the discretization of the equation (7) and let the boundary nodes be the nodes where $\mu^n$ are determined by the discretization of the boundary conditions.

We can now describe the explicit time stepping scheme for the inner nodes as follows

1. for each time-step
    2. for each grid node collect displacements from all neighbor grid nodes
        3. evaluate the stencil
4. if one more time-step, go to 1, else stop.

Step 2-3 consists of a single matrix multiplication with a large sparse matrix,

$$u_r^{n+1}(i,j,k) = \sum_{q=1}^{3} \sum_{s1=-1}^{1} \sum_{s2=-1}^{1} \sum_{s3=-1}^{1} \sum_{s4=-1}^{0} [m^r_{s1,s2,s3,s4,q} \; u_q^{n+s4}(i+s1, j+s2, k+s3)], \tag{8}$$

for $i = 0, 1, \ldots N_x$, $j = 0, 1, \ldots N_y$, $k = 0, 1, \ldots N_z$ and $r = 1, 2, 3$ and $m^r_{s1,s2,s3,s4,q}$ are weights from the discretization. In figure 2 we see the communication pattern for step 2 in the time stepping scheme.

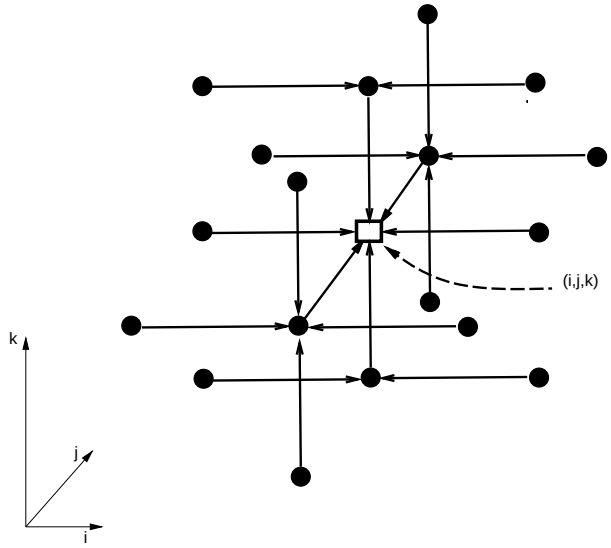


Figure 2: Communication pattern for the stencil operation at an inner node.

Discretizing (4) in the new coordinates gives,

$$\begin{aligned}
D_0^z u_1^n(i,j,N_z) &= -h(i,j,N_z) D_0^x u_3^n(i,j,N_z) + f_1(n\Delta t)/M \\
D_0^z u_2^n(i,j,N_z) &= -h(i,j,N_z) D_0^y u_3^n(i,j,N_z) + f_2(n\Delta t)/M \\
D_0^z u_3^n(i,j,N_z) &= h(i,j,N_z) \Lambda \left( D_0^x u_1^n(i,j,N_z) + D_0^y u_2^n(i,j,N_z) \right) + f_3(n\Delta t)/M,
\end{aligned} \tag{9}$$

where $D_0^x \mu(i,j,k) = (\mu(i+1,j,k) - \mu(i-1,j,k))/2\Delta x$ etc, $\Lambda = \nu/(\nu-1)$, $M = \frac{E(1-\nu)}{(1-2\nu)(1+\nu)}$ is the constrained modulus, for $i = 1, 2, \ldots, N_x - 1$ and $j = 1, 2, \ldots, N_y$.

The displacements at the bottom of the parallelepiped are zero, *i.e.* $\mu(i,j,0) = 0$ for $r = 1, 2, 3$.

On the remaining faces absorbing boundary conditions are applied. For simplicity of presentation we consider the case $h$ and $E$ constant. The paraxial approximation of order 1 of equation (1), see [CE], at $x = N_x \Delta x$ is,

$$\frac{\partial u_r}{\partial t} = -C_r \frac{\partial u_r}{\partial x}, \quad r = 1, 2, 3, \tag{10}$$

in which

$$C_1 = [E(\alpha + 2\beta)]^{1/2},$$
$$C_2 = C_3 = [E\beta]^{1/2},$$

where $\alpha = \nu/[\rho(1+\nu)(1-2\nu)]$ and $\beta = 1/[2\rho(1+\nu)]$. $C_1$ and $C_2$ are the P- and S-wave velocities respectively.

Discretize (10) with forward differences in time and in space,

$$D_+^t u_r^n(N_x+1,j,k) = -C_r D_+^x u_r^n(N_x,j,k), \tag{11}$$

where $D_+^t u_r^n(i,j,k) = (u_r^{n+1}(i,j,k) - u_r^n(i,j,k))/\Delta t$ etc, which is used to update the ghost nodes $(N_x+1,j,k)$, where $j = 1,2,\ldots,N_y-1$ and $k = 1,2,\ldots,N_z+1$. Good absorbing boundary conditions on the edges between absorbing boundary faces is constructed by rotating the coordinate system and the equations 45° and applying the paraxial approximation in the bisector direction to the adjoining faces [BE]

## 2.3 Implementation

For massively data-parallel computers of SIMD type, the operations are done on all the data with the same instruction at once [T1]. For example, matrix addition is well suited where the matrix elements are the data and the instruction is addition. Also, explicit time stepping methods, such as the one employed here, are well suited. The program is implemented in CM Fortran on a CM-200 massively data-parallel computer with 256 FPUs and 1 Gbyte RAM. Timings will be given for runs in single precision and 128 FPUs on this configuration.

The most important thing when programming a massively data-parallel computer (as CM-200) is to distribute the data in a clever way onto the processors to minimize communications and do computations locally in every FPU (elemental computation). One should also avoid operations on a subset of the data, unless the data-layout is intended for this operation. Otherwise the operation is (probably) performed on the whole data set with a mask operation [T1].

Since nearest neighbor processor or `NEWS` communication is fast on a CM-200, the natural distribution with all data related to each grid node on separate FPUs is well suited for our case. The main problem in our algorithm is the updating of $\mu^n$at the ghost nodes. They are updated with help of the boundary condition so that $\mu^n$should approximate the correct boundary conditions. Since the stencil operation for all ghost nodes which belong to absorbing and zero displacement boundary conditions is of the same type as for the inner nodes, it can be made at the same time as the stencil for the inner nodes. But the stencil operation for the ghost nodes belonging to the prescribed surface tractions boundary condition has to be evaluated alone. Fortunately we have this boundary condition on only one face which suggest another data-layout than the ordinary `(:news,:news,:news)`

which would make the updating of $u^n$ in the ghost nodes very slow. With the data-layout `(:serial,:news,:news)` this updating can be done with high performance which is about ten times faster than with `(:news,:news,:news)`-layout.

The five-point stencil applied on a two dimensional array,

$$u^{n+1}(i,j) = c_0 u^n(i,j) + c_1 u^n(i+1,j) + c_2 u(i-1,j) + c_3 u(i,j+1) + c_4 u(i,j-1),$$

is implemented in FORTRAN 90 as

```
 u = c0*u + c1*CSHIFT(u, DIM=1, SHIFT= 1) +
&           c2*CSHIFT(u, DIM=1, SHIFT=-1) +
&           c3*CSHIFT(u, DIM=2, SHIFT= 1) +
&           c4*CSHIFT(u, DIM=2, SHIFT=-1)
```

This strategy did not give satisfactory results and timings on communication versus elemental computation time showed that communication took 3.3 times longer time then the elemental computations [BE]. Improvements have been done by using a parallel shift routine from the CMSSL library [T2]. The communication pattern is then at compiled run-time and can be used many times one wants to. For example, the five-point stencil with `PSHIFT`,

```
 pattern_nr = pshift_setup(4,u,ier,
&   CMSSL_CSHIFT,1, 1,
&   CMSSL_CSHIFT,1,-1,
&   CMSSL_CSHIFT,2, 1,
&   CMSSL_CSHIFT,2,-1)
...
 DO 100 i=1,no_iter
    call pshift(4,pattern_nr,ier,
&      a1,u,1, 1,
&      a2,u,1,-1,
&      a3,u,2, 1,
&      a4,u,2,-1)
    u = c0*u + c1*a1 + c2*a2 + c3*a3 + c4*a4
...
```

performs the stencil operation much faster then with `CSHIFT`.

Saving the shifted arrays in temporary arrays can also save time. They can be used in several stencil operations without any extra communication. For example,

$$u^{n+1}(i,j) = a_0 u^n(i,j) + a_1 u^n(i+1,j) + a_2 u(i-1,j) + a_3 u(i,j+1) + a_4 u(i,j-1)$$
$$v^{n+1}(i,j) = b_0 u^n(i,j) + b_1 u^n(i+1,j) + b_2 u(i-1,j) +$$

$$
\begin{aligned}
& b_3u(i,j+1) + b_4u(i,j-1), \\
w^{n+1}(i,j) = {} & c_0u^n(i,j) + c_1u^n(i+1,j) + c_2u(i-1,j) + \\
& c_3u(i,j+1) + c_4u(i,j-1),
\end{aligned}
$$

is then implemented as,

```
 pattern_nr = pshift_setup(4,u,ier,
&   CMSSL_CSHIFT,1, 1,
&   CMSSL_CSHIFT,1,-1,
&   CMSSL_CSHIFT,2, 1,
&   CMSSL_CSHIFT,2,-1)
 ...
 call pshift(6,pattern_nr,ier,
&      u1,u,1, 1,
&      u2,u,1,-1,
&      u3,u,2, 1,
&      u4,u,2,-1)
    u = a0*u + a1*u1 + a2*u2 + a3*u3 + a4*u4
    v = b0*u + b1*u1 + b2*u2 + b3*u3 + b4*u4
    w = c0*u + c1*u1 + c2*u2 + c3*u3 + c4*u4
```

For more complicated stencil operations we can do yet another optimization. This kind of optimization gives better performance on stencil operations in more dimensions, like for our stencil operation in figure 2, but is illustrated on 2D for simplicity.

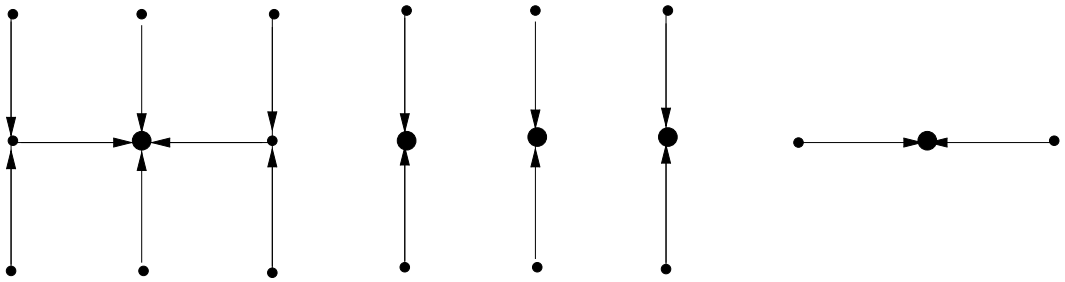

Figure 3: The nine-point stencil (12) in the first figure is decomposed into two communication and computation steps.

Consider the stencil operation,

$$
\begin{aligned}
u^{n+1}(i,j) = {} & a_0u^n(i,j) + a_1u^n(i+1,j) + a_2u(i-1,j) + \\
& a_3u(i,j+1) + a_4u(i,j-1) + a_5u(i-1,j-1) + \\
& a_6u(i-1,j+1) + a_7u(i+1,j-1) + a_8u(i+1,j+1), \qquad (12)
\end{aligned}
$$

It is implemented as,

```
 pattern1 = pshift_setup(2,u,ier,
&   CMSSL_CSHIFT,1, 1,
&   CMSSL_CSHIFT,1,-1)
 pattern2 = pshift_setup(2,u,ier,
&   CMSSL_CSHIFT,2, 1,
&   CMSSL_CSHIFT,2,-1)
 ...
 call pshift(2,pattern1,ier,
&      u1,u,1, 1,
&      u2,u,1,-1)
 u3 = a6*u1 + a5*u2 + a2*u
 u4 = a8*u1 + a7*u2 + a1*u
 u  = a3*u1 + a4*u2 + a*u
 call pshift(2,pattern2,ier,
&      u1,u4,1, 1,
&      u2,u3,1,-1)
 u = u + u1 + u2
```

to save communication operations. Note that we do only 4 communication operations, which is the same number of communication operations needed for the five-point stencil. Note also that some of the stencil coefficient arrays are shifted, for example `a2(i,j)`$:= a_2(i+1,j)$.

## 2.4 Results

The dynamic characteristics of Gothenburg clay was studied by Andreasson [An]. Based on his results the unit weight $\rho$ of the clay material can be assumed to be 1550 kg/m$^3$, nearly constant with depth. The Young's modulus $E$ on the other hand increases with depth and can be estimated from $E = -1.39z + 16$, giving $E$ in MPa when $z$ is in meters. Poisson's ratio was estimated to $\nu = 0.495$. The damping characteristics were studied in a resonant column apparatus and in the low strain amplitude range $< 10^{-5}$, the material damping constant was estimated to be 2 %.

The load consists of thirty thousand people jumping on the ground in front of the stage. There are approximately 4–5 persons/m$^2$ leading to a load amplitude of 3.0 kPa. Pernica has studied dynamic live loads at a rock concerts [P]. He found out that audiences can easily produce rhythmic forces in the frequency range 1–3 Hz. Sahlin estimated the frequency to be in the 2.2–2.5 Hz range for the "Nya Ullevi" concerts [S]. In figure 4 results from a simulation with $\nu = 0.47$, force frequency 2.0 Hz and damping constants $\gamma = 0.241$ and $\kappa = 0.0016$.

With all optimization above we get a performance of 436 Mflops on a CM-200 with 128 FPUs and a VP-ratio [T1] of 2048. To solve a transient problem 5 s forward in time with $64^3$ nodes and $3 \cdot 64^3 \approx 785000$ degrees of freedom on the

Figure 4: In the first three figures the total displacement field at the surface is plotted together with the bedrock at times t= 0.5 s, 5 s and 9.75 s. All displacements are multiplied by a factor 7000. In the last figure the vertical displacement component $w$ is plotted versus time at an observation point at the surface. Every second point is plotted.

CM-200 with 256 FPUs takes about 6.5 hours.

## 2.5 Conclusions

A computer code `ELWA` has been developed for the simulation of elastic wave propagation in soil material. The numerical algorithm was designed for efficient implementation on parallel computers. A performance of 850 Mflops in single precision was achieved on an 256 FPUs CM-200. With this efficiency realistic three dimensional cases can be simulated. The accuracy of the code with different boundary conditions has been evaluated in comparison with other codes and with known analytical solutions. `ELWA` has been used as the tool in 3D-simulation of ground vibrations induced from a jumping rock audience [E].

## 2.6 Acknowledgment

Part of this work has been a joint work with S. Erlingsson. Thanks to F. Hedman and P. Hammarlund who gave me some ideas how `ELWA` should be optimized on the CM-200.

# References

[A] Hibbitt, Karlson & Sorensen, Inc. *ABAQUS User Manual. Version 4.8.*

[An] B. Andreasson. *Deformation Characteristics of Soft High-Plastic Clays under Dynamic Loading Conditions.* PhD Thesis, Chalmers University of Technology, Göteborg, Sweden, 1979.

[BE] M. Berglund and S. Erlingsson. *Simulation of ground vibrations on a massively data-parallel computer.* In Ch. Hirsch et al., editor. *Numerical Methods in Engineering '92.* Elsevier Science Publishers B.V., 1982.

[CE] R. Clayton and B. Engquist. *Absorbing boundary conditions for acoustic and elastic wave equations.* Bulletin of the Seismological Society of America, 67(6):1519–1540, December 1977.

[E] S. Erlingsson. *Dynamic Soil Analysis with an Application to Rock Music Induced Vibrations in Ullevi Stadium.* PhD Thesis, Royal Institute of technology, Stockholm, Sweden, 1993.

[G] B. Gustafsson. *The convergence rate for difference approximations to general mixed initial boundary value problems.* SIAM J. Num. Analysis, 18:179–190, 1981.

[P] G. Pernica. *Dynamic live loads at a rock concerts.* Canadian Journal of Civil Engineering, 10(2):185–191, 1988.

[S] S. Sahlin. *On Site Measurements of Soil and Structure Response of the Soccer Stadium "Nya Ullevi" in Göteborg.* In *A course in Fundamentals of Earthquake Engineering 1989.* Statens Provningsanstalt, 1989.

[T1] Thinking Machines Corporation. *Getting started in CM Fortran.*

[T2] Thinking Machines Corporation. *CMSSL for CM Fortran. Version 2.2.*