Note: This is the pdf version of the presentation with only one illustrating figure included instead of the corresponding MPEG, VRML and netCDF files. Full version of the presentation with active links to MPEG, VRML and netCDF files is available at http://kfe.fjfi.cvut.cz/~sinor/docs/mptl2004/.
Contents

1 Introduction ................................................................. 3

2 LISA/SIM in 1D .............................................................. 4

3 LISA/SIM in 2D .............................................................. 5
   3.1 Discretization .......................................................... 5
   3.1.1 Zero stress boundary conditions ............................... 8
   3.2 Code generation ....................................................... 9

4 LISA/SIM in 3D .............................................................. 9
   4.1 Discretization .......................................................... 10
   4.2 Code generation ....................................................... 12

5 Numerical programs ....................................................... 13
   5.1 Geometric shapes of computational domains .................... 13
   5.2 Input signal ............................................................ 13
   5.3 Boundary conditions ................................................ 13
      5.3.1 Illustration of boundary conditions ......................... 14
   5.4 Data storage .......................................................... 15
      5.4.1 Example 1. Monitoring of numerical errors ............... 15
      5.4.2 Example 2. Tracing of numerical instabilities ........... 16
      5.4.3 Example 3. Brasil Nut Sandwich modeling ................. 16

6 Several illustrating results ............................................. 17
1 Introduction

Propagation of acoustic waves is important in many areas, e.g. material characterization, material diagnostics and defectoscopy.

Analytical solutions of the wave equations are well known, but severe difficulties are encountered when dealing with real materials, because of the possible presence of textures, stress fields, inhomogeneities and other irregularities or defects.

Development of an efficient simulation procedure for the study of ultrasonic wave propagation in a material specimen of arbitrary complexity. Reflection, refraction, absorption and mode conversion effects are automatically obtained.

We suppose elastodynamic wave equation of the form

\[
\frac{\partial}{\partial x_l} \left( S_{klmn} \frac{\partial w^m}{\partial x_n} \right) + f_k = \rho \frac{\partial^2 w^k}{\partial t^2} + R_k \frac{\partial w^k}{\partial t}, \quad k, l, m, n = 1, 2, 3, \quad (1)
\]

where

- \( w^k(\vec{x}) \) – particle displacement component,
- \( \rho(\vec{x}) \) – material density,
- \( S(\vec{x}) \) – stiffness tensor,
- \( f_k(\vec{x}) \) – external volume force,
- \( R_k(\vec{x}) \) – viscous friction coefficient.

For the system of equations (1) we have adopted the method of solution based on the local interaction simulation approach (LISA) and the sharp interface model (SIM) introduced for 2D case in [1].

In this presentation we are interested in two aspects of the LISA/SIM formalism:

- LISA/SIM method is derived in 2D and 3D cases by means of the computer algebra system Reduce [3].

  Computer algebra systems (Mathematica, Maple, MuPAD . . .) are powerful software packages capable to deal with complicated mathematical formulas on a computer. CAS can support the development of simulation codes for numerical solving of partial differential equations in several stages:

  - discretization,
  - analysis of properties of the numerical methods (e.g. stability),
  - numerical code generation.

  All steps of this methodology have been used for the development of 2D and 3D simulation codes for elastodynamic wave equation.

- LISA/SIM 2D, 3D and LISA/ICI 2D methods are implemented in computer programs, part of which have been generated semi-automatically by system Reduce.

- Parallel versions of the programs: OpenMP, MPI.
2 LISA/SIM in 1D

The LISA/SIM method can be nicely demonstrated in the case of 1D. We assume simplified version of the elastodynamic wave equation

\[ S \frac{\partial^2 w}{\partial x^2} = \rho \frac{\partial^2 w}{\partial t^2} \]  

(2)

which can be written as

\[ Sw'' = \rho \ddot{w} \]  

(3)

Figure 1: Numerical grid around an interface point with additional points M, N, P, Q.

We impose

\[ \ddot{w}_P = \ddot{w}_Q = \ddot{w}_I, \]  

(4)

where \( P \) and \( Q \) are infinitely close to the interface. We then apply the wave equation and impose the matching of \( \tau_1 \):

Wave equation in \( P \):

\[ Sw''_P = S \frac{w'_P - w'_M}{\varepsilon/2} = \rho \ddot{w}_I. \]  

(5)

Wave equation in \( Q \):

\[ \bar{S}w''_Q = \bar{S} \frac{w'_N - w'_Q}{\bar{\varepsilon}/2} = \bar{\rho} \ddot{w}_I. \]  

(6)

Matching of \( \tau_1 \):

\[ (\tau_1)_P - (\tau_1)_Q = Sw'_P - \bar{S}w'_Q = 0, \]  

(7)

where \( M \) and \( N \) are the middle points of the two cells bordering at the interface.

By multiplying Eq. (5) by \( \varepsilon/2 \) and Eq. (6) by \( \bar{\varepsilon}/2 \), summing the 2 eqs. and subtracting Eq. (7), one obtains

\[ \bar{S}w'_N - Sw'_M = \frac{1}{2}(\varepsilon \rho + \bar{\varepsilon} \bar{\rho})\ddot{w}_I, \]  

(8)

\[ \ddot{w}_I = \frac{2}{\varepsilon \rho + \bar{\varepsilon} \bar{\rho}} \left[ \bar{S} \frac{w_{I+1} - w_I}{\bar{\varepsilon}} - \frac{Sw_{I} - w_{I-1}}{\varepsilon} \right] = \tilde{t}w_{I-1} + \tilde{t}'w_{I+1} - (\tilde{t} + \tilde{t}')w_I, \]  

(9)

where

\[ \tilde{t} = \frac{2S/\varepsilon}{\varepsilon \rho + \bar{\varepsilon} \bar{\rho}}, \quad \tilde{t}' = \frac{2\bar{S}/\bar{\varepsilon}}{\varepsilon \rho + \bar{\varepsilon} \bar{\rho}}. \]  

(10)

Then

\[ w_{I,t+1} = 2w_I - w_{I,t-1} + \delta^2 \left( \tilde{t}w_{I-1} + \tilde{t}'w_{I+1} - (\tilde{t} + \tilde{t}')w_I \right). \]  

(11)
3 LISA/SIM in 2D

We use the Local Interaction Simulation Approach (LISA) and Sharp Interface Model (SIM) according to P. P. Delsanto et al. [1]. In this section we summarize the basic ideas of the method in 2D.

Assuming orthotropic specimen and symmetry (translation invariance) with respect to $x_3$, Eq. (1) becomes

\[
\begin{align*}
\frac{\partial}{\partial x_1} \left( S_{llmm} \frac{\partial w^m}{\partial x_m} \right) &+ f_k = \rho \frac{\partial^2 w^k}{\partial t^2} + R_k \frac{\partial w^k}{\partial t}, \quad k, l, m = 1, 2, \\
\frac{\partial}{\partial x_1} \left( S_{3333} \frac{\partial w^3}{\partial x_1} \right) &+ f_3 = \rho \frac{\partial^2 w^3}{\partial t^2} + R_3 \frac{\partial w^3}{\partial t}, \quad l = 1, 2,
\end{align*}
\]

(12)

(13)

where Eq. (13) corresponding to an anti-plane shear wave is completely uncoupled from Eq. (12).

Eq. (12) can be written more explicitly as

\[
\begin{align*}
\sigma_k \frac{\partial^2 w^k}{\partial x_k^2} + \mu \frac{\partial^2 w^k}{\partial x_k \partial x_h} + \nu \frac{\partial^2 w^k}{\partial x_h \partial x_k} + f_k &= \rho \frac{\partial^2 w^k}{\partial t^2} + R_k \frac{\partial w^k}{\partial t}, \quad k = 1, 2, h = 3 - k = 2, 1,
\end{align*}
\]

(14)

where \( \sigma_k = S_{kkkk}, \quad \lambda = S_{1122}, \quad \mu = S_{1212} \)

and \( \nu = \lambda + \mu = \sigma - \mu \).

In matrix notation Eq. (14) can be written as

\[
AW_{xx} + BW_{yy} + CW_{xy} + f = \rho W_{tt} + RW_t,
\]

(15)

where

\[
\begin{align*}
A &= \begin{pmatrix} \sigma_1 & 0 \\ 0 & \mu \end{pmatrix}, & B &= \begin{pmatrix} \mu & 0 \\ 0 & \sigma_2 \end{pmatrix}, & C &= \begin{pmatrix} 0 & \nu \\ \nu & 0 \end{pmatrix}, \\
f &= \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}, & R &= \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}, & W &= \begin{pmatrix} u \\ v \end{pmatrix},
\end{align*}
\]

and

\[
\begin{align*}
W_{xx} &= \frac{\partial^2}{\partial x^2} \begin{pmatrix} u \\ v \end{pmatrix}, & W_{xy} &= \frac{\partial^2}{\partial x \partial y} \begin{pmatrix} u \\ v \end{pmatrix}, \quad \text{etc.}
\end{align*}
\]

3.1 Discretization

In this section the elastodynamic wave equation in 2D, Eq. (15), is discretized.

Assumption of the model:

- The wave equation is discretized on the rectangular uniform grid with grid steps $\Delta x, \Delta y$.
- The displacements $W^{ij}$ are discretized at points $(i, j)$.
- The additional points $(i \pm \delta, j \pm \delta)$ at distance $\delta \ll \Delta x \sim \Delta y$ from grid lines are introduced as shown in the Fig. 2.
- A cell is marked by its lower left corner, so the cell with center at $(i + 1/2, j + 1/2)$ is marked by $i, j$. 
Material matrices are constant inside each cell with center at \((i + 1/2, j + 1/2)\). Material matrices can be discontinuous with jumps across the cells boundaries.

The wave equation (15) is taken in 4 points \((i \pm \delta, j \pm \delta)\). These 4 equations after considering the constant material matrices in each cell can be written as

\[
A^{i + a, j + b} W_{xx}^{i + \delta, j + \delta} + B^{i + a, j + b} W_{yy}^{i + \delta, j + \delta} + C^{i + a, j + b} W_{xy}^{i + \delta, j + \delta} + f = \rho W_{tt}^{i + \delta, j + \delta} + R W_{t}^{i + \delta, j + \delta}
\]

where \(a, b = -1, 0\) and the overbar denotes the shift operator \(\bar{0} = 1, \bar{1} = -1\).

The finite differences replacements for second order derivatives are for \(a, b = \pm 1\) defined, in terms of the first order derivatives:

\[
\begin{align*}
W_{xx}^{i + a, j + b} & = \frac{W_{x}^{i + a/2, j} - W_{x}^{i + a\delta, j + b\delta}}{a\Delta x/2}, \\
W_{yy}^{i + a, j + b} & = \frac{W_{y}^{i + b/2, j} - W_{y}^{i + a\delta, j + b\delta}}{b\Delta y/2}, \\
W_{xy}^{i + a, j + b} & = \frac{W_{y}^{i + a, j + b/2} - W_{y}^{i + a\delta, j + b\delta}}{a\Delta x}.
\end{align*}
\]

where for now the first order derivatives at the points \((i \pm \delta, j \pm \delta)\) remain unevaluated. These unevaluated derivatives will be eliminated by using the stress continuity equations.
The first order derivatives at the points \((i \pm 1/2, j), (i, j \pm 1/2)\) are approximated by

\[
W_i^{x,i+1/2,j} = \frac{W_{i+1,j} - W_{i,j}}{\pm \Delta x},
\]
\[
W_i^{y,i,j+1/2} = \frac{W_{i,j+1} - W_{i,j}}{\pm \Delta y}.
\]

For imposing stress continuity we define additional points \((i \pm \delta, j \pm \epsilon)\) and \((i \pm \epsilon, j \pm \delta)\) with \(\epsilon \ll \delta \ll \Delta x \sim \Delta y\), see the Fig. 3.

Figure 3: Close neighbourhood of a cross point with additional points \((i \pm \delta, j \pm \epsilon)\) and \((i \pm \epsilon, j \pm \delta)\), \((i \pm \delta, j \pm \delta)\) with \(\epsilon \ll \delta \ll \Delta x \sim \Delta y\).

The components of the stress tensor \(\tau\) in 2D for orthotropic materials with \(\pi\) symmetry are given by

\[
\tau_{ij}^1 = A^{ij}W_{x}^{ij} + D^{ij}W_{y}^{ij},
\]
\[
\tau_{ij}^2 = E^{ij}W_{x}^{ij} + B^{ij}W_{y}^{ij},
\]

where \(D, E\) are also material matrices:

\[
D = \begin{pmatrix}
0 & \lambda \\
\mu & 0
\end{pmatrix}, \quad E = \begin{pmatrix}
0 & \mu \\
\lambda & 0
\end{pmatrix}.
\]

As we have already pointed out the material matrices \(A, B, C, D, E\) are constant inside each cell, i.e.

\[
A^{i+\epsilon,j+\delta} = A^{i+\delta,j+\epsilon} = A^{i+\delta,j+\delta} = A^{i,j} \quad \text{etc.}
\]
Stress continuity across the interface will be considered at the points $\pm \epsilon$. Now while still keeping the first order derivatives at the points $(i \pm \delta, j \pm \delta)$ unevaluated we approximate the first order derivatives at the points $\pm \epsilon$, for $a, b = \pm 1$, by

$$
W_{x}^{i+a\delta,j+b\epsilon} = \frac{W^{i+a,j} - W^{i,j}}{a\Delta x},
$$

$$
W_{y}^{i+a\delta,j+b\epsilon} = \frac{W^{i+a,j} - W^{i,j}}{b\Delta y},
$$

$$
W_{x}^{i+a\epsilon,j+b\delta} = \frac{W^{i,j+b} - W^{i,j}}{a\Delta x},
$$

$$
W_{y}^{i+a\epsilon,j+b\delta} = \frac{W^{i,j+b} - W^{i,j}}{b\Delta y},
$$

where $a, b = \pm 1$.

The continuity of stress tensor at the points $\pm \epsilon$ across the cell boundaries is expressed by the equations

$$
\tau^{i-\epsilon,j\pm\delta}_{1} = \tau^{i+\epsilon,j\pm\delta}_{1},
$$

$$
\tau^{i\pm\delta,j-\epsilon}_{2} = \tau^{i\pm\delta,j+\epsilon}_{2}. \tag{17}
$$

Combining 4 equations (16) and 4 equations (17) the unknown first order derivatives at the points $(i \pm \delta, j \pm \delta)$ can be eliminated to obtain

$$
\sum_{k=-1}^{1} \sum_{l=-1}^{1} P_{kl}^{ij} W^{i+k,j+l} + f = \rho W^{i,j}_{tt} + RW^{i,j}. \tag{18}
$$

The standard discretization for the time derivatives is used:

$$
W^{i,j}_{tt} \approx \frac{W^{i,j,n+1} - 2W^{i,j,n} + W^{i,j,n-1}}{dt^2}, \quad W^{i,j}_{t} \approx \frac{W^{i,j,n+1} - W^{i,j,n-1}}{2dt}
$$

and the final explicit finite difference scheme is obtained:

$$
W^{i,j,n+1} = -W^{i,j,n-1} + \sum_{k=-1}^{1} \sum_{l=-1}^{1} P_{kl}^{ij} W^{i+k,j+l,n}. \tag{19}
$$

All derivation of the difference scheme has been done in the computer algebra system Reduce [3]. The obtained results agree with [1]. The derivation has been done in vector-matrix notation.

### 3.1.1 Zero stress boundary conditions

The discrete formulas for the zero stress boundary conditions has been derived in the same way as the difference scheme for the general cross point in the previous section. Stress in the cells out of the boundaries taken to be zero in the stress continuity equations (17) and all remaining derivation is the same as for the inside cross point. Eight different boundary conditions has been derived at four edges and four corners.

We have found that the same boundary conditions can be derived from the inside difference scheme (19) by setting all material constants to zero ($\rho = 0, A = B = C = D = E = 0$) in the cells out of boundaries. This has been verified by computer algebra.
3.2 Code generation

Having derived the difference scheme by means of the computer algebra we have the formulas for the scheme and the code generation facilities of the computer algebra systems can be used to directly write out the numerical source code implementing the difference scheme.

The derivation of the finite difference scheme (19) has been done in the vector-matrix notation. However our numerical code can deal only with the scalar quantities. The difference scheme (19) in the scalar form can be written as

\[
\begin{align*}
    u^{i,j,n+1} &= -u^{i,j,n-1} + \sum_{k=-1}^{1} \sum_{l=-1}^{1} \left( c^{ijkl}_{uu} u^{i+k,j+l,n} + c^{ijkl}_{uu} v^{i+k,j+l,n} \right), \\
    v^{i,j,n+1} &= -v^{i,j,n-1} + \sum_{k=-1}^{1} \sum_{l=-1}^{1} \left( c^{ijkl}_{uv} u^{i+k,j+l,n} + c^{ijkl}_{uv} v^{i+k,j+l,n} \right),
\end{align*}
\]

where the coefficients \(c^{ijkl}_{pq}\), \(p = u, v; q = u, v\) are functions of the material constants \(\mu, \nu, \lambda, \sigma_1, \sigma_2, R\) and \(f\). These coefficients do not depend on the time and can be calculated only once at the beginning of the computation. A numerical routine for their calculation has been automatically generated.

A routine implementing the finite difference scheme (19) using the numerically precalculated coefficients, i.e., doing one time step, is also generated. The code for zero stress boundary conditions and for absorbing boundary conditions has been automatically generated, too. This part of the program performing the actual numerical computation is the core part of the code and is represented by about 1300 lines of the code (60 kBytes).

4 LISA/SIM in 3D

Derivation similar to paper by P. P. Delsanto et al. [2]. In addition included terms for external volume forces and attenuation.

Elastodynamic wave equation in 3D with coordinates \((x_1, x_2, x_3)\) and displacements \(W = (w^1, w^2, w^3)\):

\[
(S_{klmn} w^n_{x_n})_{x_l} + f = \rho w^{k}_{tt} + R w^{k}_{l}, \quad k, l, m, n = 1, 2, 3.
\]

(20)

where

\[
S_{klmn} = S_{V(k,l)V(m,n)} = \begin{pmatrix}
\sigma_1 & \lambda_6 & \lambda_5 & \tau_{14} & \tau_{15} & \tau_{16} \\
\sigma_2 & \lambda_4 & \tau_{24} & \tau_{25} & \tau_{26} \\
\sigma_3 & \tau_{34} & \tau_{35} & \tau_{36} \\
\lambda_4 & \tau_{45} & \tau_{46} \\
\mu_4 & \tau_{56} \\
\mu_5 & \mu_6
\end{pmatrix}.
\]

(21)

and where the Voight’s convention is used:

\[V(i, j) = i\delta_{ij} + (1 - \delta_{ij})(9 - i - j).\]
Orthotropic materials:

\[ S_{klmn} = S_{V(k,l)V(m,n)} = \begin{pmatrix} \sigma_1 & \lambda_{12} & \lambda_{13} & \mu_{23} \\ \sigma_2 & \lambda_{23} & \mu_{13} \\ \sigma_3 & \mu_{12} \end{pmatrix}. \]  

(22)

It is convenient to write wave equation \((20)\) in the form

\[ \sum_{n=1}^{3} \left( A_{kn} w_{x_n}^k + N_{kn} w_{x_n}^n \right) + f = \rho w_{tt}^k + R w_t^k, \]

(23)

where

\[ A = \begin{pmatrix} \sigma_1 & \mu_{12} & \mu_{13} \\ \mu_{12} & \sigma_2 & \mu_{23} \\ \mu_{13} & \mu_{23} & \sigma_3 \end{pmatrix}, \quad N = \begin{pmatrix} 0 & \nu_{12} & \nu_{13} \\ \nu_{12} & 0 & \nu_{23} \\ \nu_{13} & \nu_{23} & 0 \end{pmatrix}, \]

\[ \nu_{ij} = \lambda_{ij} + \mu_{ij}. \]

4.1 Discretization

LISA, SIM method in 3D:

\(w\) in the following stands for any component of displacement vector \(W = (w^1, w^2, w^3)\).

Rectangular uniform grid with grid steps \(\Delta x, \Delta y, \Delta z\) with additional points at distance \(\delta \ll \Delta x \sim \Delta y\).

Finite differences for derivatives:

- First order:
  
  \[ w_i^{\pm 1/2,j,k} = \frac{w_i^{\pm 1,j,k} - w_i^{j,k}}{\pm \Delta x}, \]
  
  \[ w_i^{\pm 1/2,j,k} = \frac{w_i^{j+1,k} - w_i^{j,k}}{\pm \Delta y}, \]
  
  \[ w_i^{\pm 1/2,j,k} = \frac{w_i^{j,k+1} - w_i^{j,k}}{\pm \Delta z}. \]

- Second order, for \(a, b, c = \pm 1:\)
  
  \[ w_{xx}^{i+a} = \frac{w_x^{i+a/2} - w_x^{i+a}}{a \Delta x/2}, \]
  
  \[ w_{yy}^{i+b} = \frac{w_y^{i+b/2} - w_y^{i+b}}{b \Delta y/2}, \]
  
  \[ w_{zz}^{i+c} = \frac{w_z^{i+c/2} - w_z^{i+c}}{c \Delta z/2}, \]
  
  \[ w_{xy}^{i+a+b} = \frac{w_x^{i+a+b/2} - w_y^{i+b/2}}{a \Delta x}. \]
The derivatives at the points \( \pm \delta, j \pm \delta, k \pm \delta \), for \( a, b, c = -1, 0, m = 1, 2, 3 \):

\[
\begin{align*}
    w_{i+\delta,j+b\delta,k+c\delta}^x &= w_{i+\delta,j+b\delta,k+c\delta}^x - w_{i,j,k}^x, \\
    w_{i+\delta,j+b\delta,k+c\delta}^y &= \frac{w_{i+\delta,j+b\delta,k+c\delta}^y - w_{i,j,k}^y}{b\Delta y}, \\
    w_{i+\delta,j+b\delta,k+c\delta}^z &= \frac{w_{i+\delta,j+b\delta,k+c\delta}^z - w_{i,j,k}^z}{c\Delta z},
\end{align*}
\]

3 equations in each of 8 points \( i \pm \delta, j \pm \delta, k \pm \delta \):

\[
\sum_{n=1}^{3} \left( A_{i+\delta,j+b\delta,k+c\delta} + N_{i+\delta,j+b\delta,k+c\delta} w_{i+\delta,j+b\delta,k+c\delta}^x \right) + f = \rho w_{i+\delta,j+b\delta,k+c\delta}^x + R w_{i+\delta,j+b\delta,k+c\delta}^x,
\]

where \( \rho = 1, -1 = -1 \)

To obtain continuity of stress tensor components, additional points are introduced: \( (i \pm \delta, j \pm \epsilon, k \pm \delta) \) and \( (i \pm \delta, j \pm \delta, k \pm \epsilon) \) with \( \epsilon \ll \delta \ll \Delta x \sim \Delta y \sim \Delta z \).

In addition, \( A_{i+\delta,j+b\delta,k+c\delta} \) are supposed to be constant in each cell, i.e.

\[
\begin{align*}
    A_{i+\delta,j+b\delta,k+c\delta} &= A_{i+\delta,j+b\delta,k+c\delta}^1 = A_{i+\delta,j+b\delta,k+c\delta}^2 = A_{i+\delta,j+b\delta,k+c\delta}^3,
\end{align*}
\]

eetc., cell is marked by a corner with minimal indices.

The derivatives at the points \( \pm \epsilon, \), for \( a, b, c = \pm 1 \), have the form:

\[
\begin{align*}
    w_{i+a\delta,j+b\delta,k+c\epsilon}^x &= w_{i+a\delta,j+b\delta,k+c\epsilon}^x - w_{i,j,k}^x, \\
    w_{i+a\delta,j+b\delta,k+c\epsilon}^y &= \frac{w_{i+a\delta,j+b\delta,k+c\epsilon}^y - w_{i,j,k}^y}{b\Delta y}, \\
    w_{i+a\delta,j+b\delta,k+c\epsilon}^z &= \frac{w_{i+a\delta,j+b\delta,k+c\epsilon}^z - w_{i,j,k}^z}{c\Delta z},
\end{align*}
\]

The first order derivatives at the points \( i \pm \delta, j \pm \delta, k \pm \delta \) remain unevaluated.

The components of the stress tensor

\[
\tau_{kl} = S_{klmn} w_{x_n}^m
\]

can be for orthotropic material written as

\[
\begin{align*}
    \tau_{kk} &= \sigma_k w_{x_k}^k + \sum_{n \neq k} \lambda_{kn} w_{x_n}^n, \\
    \tau_{kn} &= \mu_{kn} (w_{x_k}^k + w_{x_k}^n), \quad k \neq n.
\end{align*}
\]
Next we impose the stress continuity at the planes

- $x = \text{const.}$:
  \[ \tau_{n1}^{i+\delta,j,k+\delta} = \tau_{n1}^{i+\epsilon,j,k+\epsilon}, \quad n = 1, 2, 3, \]

- $y = \text{const.}$:
  \[ \tau_{n2}^{i,j+\delta,k+\delta} = \tau_{n2}^{i,j+\epsilon,k+\epsilon}, \quad n = 1, 2, 3, \]

- $z = \text{const.}$:
  \[ \tau_{n3}^{i,j+\delta,k-\epsilon} = \tau_{n3}^{i,j+\delta,k+\epsilon}, \quad n = 1, 2, 3. \]

An appropriate combination of the discretized wave equations in 8 cells adjacent to a cross point with stress continuity equations allow to eliminate unknown first order derivatives at the points $i \pm \delta, j \pm \delta, k \pm \delta$.

After a standard discretization of time derivatives we obtain an explicit difference scheme for displacements $(w^1, w^2, w^3)$:

\[ w^{m, (i,j,k),n+1} = -w^{m, (i,j,k),n-1} + \sum_{l=1}^{3} \sum_{a,b,c=-1}^{1} c_{lm}^{ijkabc} w^{l, (i+a,j+b,k+c),n}. \quad (25) \]

All derivation done in component form in the computer algebra system Reduce and checked with results of Delsanto et al. [2].

### 4.2 Code generation

\[ w^{m, (i,j,k),n+1} = -w^{m, (i,j,k),n-1} + \sum_{l=1}^{3} \sum_{a,b,c=-1}^{1} c_{lm}^{ijkabc} w^{l, (i+a,j+b,k+c),n}. \quad (26) \]

- Many coefficients $c_{lm}^{ijkabc}$ in the Eq. (26) are zero, i.e. all with the shifts $a \neq 0 \land b \neq 0 \land c \neq 0$.
- These coefficients have to be stored in each point $(i, j, k)$ of the computational grid. A need to save memory!!
- It is enough to store only $3 \times 25 = 75$ coefficients in each point instead of $3^5 = 243$ (3 equations, 3 components and 3 shifts in each of the 3 coordinates).
- Rearrangement before code generation to include only non-zero coefficients.
- Names of arrays storing coefficients are generated.
- The values of coefficients $c_{lm}^{ijkabc}$ do not depend on time and can be precalculated (as we have already seen for 2D case).
  
  In this case routines for initial calculation of coefficients and for one time step have been automatically generated (29 kB of C code).
- If the coefficients are not precalculated: 6600 lines, 0.5 MB of C code.
- The code with full elastic constants tensor with 21 independent coefficients: 21000 lines, 1.5 MB of C code.
5 Numerical programs

The implementation of LISA/SIM in 2D, 3D and of LISA/ICI in 2D to computer codes is done in a universal way. Very brief description of the several characteristics of the computer programs follows.

5.1 Geometric shapes of computational domains

Geometric shapes of the predefined physical properties can have manifold form with an arbitrary dimensions and orientation in space.

2D (typical examples): rectangle, circle, ring, sector, ellipse, polygon

3D (typical examples): cuboid, sphere, spherical sector, spherical section, ellipsoid, paraboloid, cylinder

Or geometric shapes with predefined physical properties can be absolutely arbitrary generated by external tools.

5.2 Input signal

Input signal can be in the form of initial wave displacements or/and external volume forces. There can be arbitrary number, position, incidence angle of the input signals. Shape in space and time can be, e.g., gaussian, sine, rectangle, polygon, error function, explosion, Ricker function. Modulation of the signal by another function is an option.

Example of the input pulse (Gaussian modulated by sine in time, Gaussian in space, normal incidence):

\[
\begin{align*}
   w_k(\vec{x},t) &= \sum_{i=1}^{N} c_n \exp \left( -\frac{(t - t_d - i/f_r - (x - x_0)/v)^2}{c_t^2} \right) \\
   &\times \sin(2\pi f(t - t_d - i/f_r - (x - x_0)/v + \delta_s)) \\
   &\times \exp \left( -\frac{(y - y_0)^2}{c_y^2} \right) \exp \left( -\frac{(z - z_0)^2}{c_z^2} \right),
\end{align*}
\]

(27)

where

- \( c_n, c_t, c_y, c_z \) – parameters of the pulse,
- \( v \) – velocity of the wave,
- \( t_d \) – delay of the pulse,
- \( f \) – sine wave frequency,
- \( \delta_s \) – phase shift of the wave,
- \( N \) – number of pulse repetitions,
- \( f_r \) – repetition frequency.

5.3 Boundary conditions

Different types of boundary conditions (BC) are implemented in the codes:

- Free boundary conditions (\( \tau_{kl} = 0 \)).
- Solution at the boundary set to an arbitrary value.
- Periodic BC.
- Several methods to implement non-reflecting (absorbing, radiation) boundary conditions have been tried:
Extrapolation of the zero and first order works, but the results are not very good. Extrapolations of the second and higher orders are usually unstable.

Damping (in buffers) can be simulated by several different terms in the elastodynamic wave equation (proportional to the first space and time derivatives of the displacement amplitudes), e.g.:

\[
\begin{align*}
    r_{kl}(\vec{x}) \frac{\partial w_k}{\partial x_l}, \quad R_k(\vec{x}) \frac{\partial w_k}{\partial t},
\end{align*}
\]

where \( r_{kl}(\vec{x}) \), \( R_k(\vec{x}) \) are, for example, products of the power functions of an arbitrary degree or logarithm functions.

Paraxial approximation of the elastic wave equation \[4\] can be characterized by the reflection coefficient \( r_j(\theta) \) of the form

\[
    r_j(\theta) = \left( \frac{1 - \cos \theta}{1 + \cos \theta} \right)^j,
\]

where

- \( j \) – degree of approximation,
- \( \theta \) – angle of incidence measured from the normal to the boundary.

From this formula it is apparent that this approximation cannot be used in generality.

Superposition of solutions with Dirichlet boundary conditions \( (w^k = 0) \) and Neumann boundary conditions \( (\partial w^k / \partial x_l = 0) \) \[5\]. The analytical formulation is exact, independent of both frequency and incidence angle. Works very well with exception of the source waves overpassing corners of the computation region.

5.3.1 Illustration of boundary conditions

Free boundary conditions
Absorbing boundary conditions

5.4 Data storage

Due to a large amount of data generated as a rule by LISA/SIM 2D, 3D programs and of LISA/ICI in 2D, we have decided to use netCDF file format [6] for data storage. netCDF represents an efficient set of software for scientific-data storage, retrieval and manipulation. netCDF file format is:

- in the public domain.
- machine-independent, access to netCDF files over computer networks without data modification,
- direct-access,
- self-describing.
- Programming interface in Fortran, C, C++, Perl, Java, etc.
- Common Data Language, CDL, a printable text-equivalent of the contents of a netCDF file.
- Large number of subroutines, operators and utilities for processing netCDF files are available.
- netCDF files can be read by visualization and data processing programs:
  - commercial: Iris Explorer, AVS, IDL, Matlab, NCAR graphics etc.
  - in public domain: Khoros, Vis5D, LinkWinds, PolyPaint, SciAn, Ferret, PolyView etc.

5.4.1 Example 1. Monitoring of numerical errors

Plane wave reflection on Plexiglas-Aluminum bilayer
5.4.2 Example 2. Tracing of numerical instabilities

5.4.3 Example 3. Brasil Nut Sandwich modeling

Energy as a function of time in Brasil Nut Sandwich.
6 Several illustrating results


Reflection of a Gaussian longitudinal pulse on Plexiglas-Aluminum bilayer. Absorbing boundary conditions, (II).
Reflection of a Gaussian longitudinal pulse on Plexiglas-Aluminum bilayer. Absorbing boundary conditions, (III). (MPEG movie, 172 kB)

An interaction of four Gaussian pulses in a homogeneous medium. (MPEG movie, 1456 kB)

A longitudinal Gaussian pulse scattering from a crack in a glass, (I). (MPEG movie, 155 kB)

A longitudinal Gaussian pulse scattering from a crack in a glass, (II). (MPEG movie, 145 kB)
A longitudinal Gaussian pulse scattering from a crack in a glass, (III). (MPEG movie, 584 kB)

A longitudinal Gaussian pulse scattering from a crack oriented at $90^\circ$ in a glass, (I). (MPEG movie, 208 kB)

A longitudinal Gaussian pulse scattering from a crack oriented at $90^\circ$ in a glass, (II). (MPEG movie, 208 kB)

A longitudinal Gaussian pulse scattering from a crack oriented at $90^\circ$ in a glass, (III). (MPEG movie, 182 kB)
A longitudinal Gaussian pulse scattering from a void in glass, (I). (MPEG movie, 286 kB)

A longitudinal Gaussian pulse scattering from a void in glass, (II). (MPEG movie, 293 kB)

A longitudinal Gaussian pulse scattering from a void in glass, (III). (MPEG movie, 743 kB)

A longitudinal Gaussian pulse scattering from a void in glass, (IV). (MPEG movie, 1.030 MB)
A longitudinal Gaussian pulse scattering from a void in glass. Longitudinal velocity field, (V). (MPEG movie, 316 kB)

Scattering of the longitudinal Gaussian pulse modulated by sine from an aluminum inclusion in plexiglas, (I). (MPEG movie, 255 kB)

Scattering of the longitudinal Gaussian pulse modulated by sine from an aluminum inclusion in plexiglas, (II). (MPEG movie, 711 kB)

Scattering of the longitudinal Gaussian pulse modulated by sine from an aluminum inclusion in plexiglas, (III). (MPEG movie, 818 kB)
Plane wave and acoustic lens. Amplitude field, (I).
(MPEG movie, 246 kB)

Plane wave and acoustic lens. Amplitude field, (II).
(MPEG movie, 234 kB)

Plane wave and acoustic lens. Kinetic energy, (III).
(MPEG movie, 230 kB)
Plane wave and acoustic lens. Kinetic energy, (IV).
(MPEG movie, 293 kB)

(MPEG movie, 158 kB)

(Gzipped VRML file, 179 kB)
Scattering of a Gaussian longitudinal pulse from a spherical aluminum inclusion in Plexiglas. A longitudinal component of the displacement amplitude. Free boundary conditions, (III). Time = 17. (Gzip compressed VRML file, 181 kB)

Scattering of a Gaussian longitudinal pulse from a spherical aluminum inclusion in Plexiglas. A longitudinal component of the displacement amplitude. Free boundary conditions, (IV). Time = 19. (Gzip compressed VRML file, 180 kB)

References


