ACADEMIC PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Sound and Vibration 261 (2003) 819–837

JOURNAL OF SOUND AND VIBRATION

www.elsevier.com/locate/jsvi

Functional integral formulation of classical wave equations

E. Pardo*

INTEMA-Facultad de Ingeniería, Universidad de Mar del Plata, Juan B. Justo 4302, 7600 Mar del Plata, Argentina

Received 8 December 2000; accepted 7 May 2002

Abstract

It is shown in this paper that classical wave equations admit path integral formulations. For this, the evolution of the system is first set-up in terms of a fundamental solution or propagator. We choose this last name because it suggests a connection with functional integrals, which are exploited in this work. A functional integral in terms of non-singular functions is then proposed and shown to converge to the propagator in the appropriate limit for the case of scalar wave equations. One of the advantages of such formulation is that it provides an adequate framework for mesh-free numerical methods. This is demonstrated through a computational implementation that combines a simple second-degree polynomial local approximation of the continuous field and an approximate statement of the exact evolution equations. Numerical simulations of modal analysis and transient dynamics indicate the feasibility of the technique.

1. Introduction

Computation of linear wave propagation phenomena in continuous media can start with either the differential equations of motion or an equivalent variational statement. The former leads to the so-called "strong" formulations since they require evaluation of second order derivatives. Variational principles, on the other hand, lead to weak formulations in which only first order derivatives appear. The standard computational substantiations of strong and weak formulations are the finite difference method and finite element method, respectively. In particular, whenever flexibility in geometry is important—and this is the case in most engineering problems—the latter is preferred. However, the difficulty of generating acceptable finite element meshes in very demanding situations—specially in three dimensions—has prompted the development of numerical methods that do not rely on partition of the domain into elements. The so-called meshless methods based on strong formulations (such as smoothed particle hydrodynamics

*Tel.: +54-223-481-6600; fax: +54-223-481-0046.
E-mail address: epardo@fi.mdp.edu.ar (E. Pardo).

0022-460X/03/$ - see front matter
PII: S0022-460X(02)01005-2

(SPH), generalized finite differences, finite point method and several point collocation schemes [1,2]) typically attain low precision even though research continues to improve them. Weak formulations, on the other hand, pose continuity requirements throughout the domain that the trial functions (or "shape" functions) must satisfy. As a consequence, meshless methods based on weak formulations [2–7] (such as the element free Galerkin method, reproducing kernel particle methods, etc.) are built upon rather elaborate approximations which are computer intensive and frequently lead to poorly conditioned matrixes that may be deleterious in three dimensions.

A much less explored alternative, which may lead to a new family of solution procedures, is to start with a reformulation of the physical phenomena in terms of propagators and the associated path integral [8,9]. It has been shown that this can be done for parabolic and elliptic problems, which lead naturally to a meshless method that combines simplicity of implementation and good accuracy. Briefly, the idea is to replace the original differential equation by an equivalent expression of the form

$$u(x, t+\tau) = \int_{-\infty}^{\infty} P(\bar{x} - x, \tau) u(\bar{x}, t)\, \mathrm{d}\bar{x}, \tag{1}$$

where $P(\bar{x} - x, \tau)$ is the Green function or *propagator*. The next step is to assume that the propagator can be generated as a path of the *functional integral*

$$P(\bar{x} - x, \tau) = \lim_{\substack{\delta \to 0 \\ N \to \infty}} \int_{-\infty}^{\infty} A(\bar{x} - x_1, \delta)\, \mathrm{d}x_1 \int_{-\infty}^{\infty} A(x_1 - x_2, \delta)\, \mathrm{d}x_2 \cdots \times \int_{-\infty}^{\infty} A(x_{N-1} - x_N, \delta)\, \mathrm{d}x_N, \tag{2}$$

where in the limit $\delta \to 0, N \to \infty$ it is assumed that the condition $N\delta = \tau = \text{constant}$ holds. The objects $A(\bar{x} - x, \delta)$ in Eq. (2) are assumed to be non-singular functions that were called *infinitesimal propagators* in other articles. From the preceding equation it follows that the evolution of the system for a sufficiently small time step, δ, can be approximated as

$$u(x, t+\delta) \cong \int A(\bar{x} - x, \delta) u(\bar{x}, t)\, \mathrm{d}\bar{x}. \tag{3}$$

Eq. (3) can be interpreted as an approximate non-local principle, which provides the state of the system at time $(t + \delta)$ as a weighted average of the state at a previous instant, t. Hence, the action of the weighting function $A(\bar{x} - x, \delta)$ over the state, $u(x, t)$, is to push it forward in time an infinitesimal amount, δ. The form of this statement makes it specially adequate to devise simple and effective meshless computational methods. For this, it suffices to fill the domain with a finite number of points, N, (not necessarily equally spaced) with co-ordinates x_i, $i = 1, \ldots, N$ (the nodes), and define on them local values of the unknown field: $u_i = u(x_i)$. Define now an approximation of the field $u(x)$ as a linear combination of prescribed "shape" functions $\varphi_i(x)$:

$$\tilde{u}(\bar{x}, t) = \sum \varphi_i(\bar{x}) u_i(t). \tag{4}$$

Combining Eqs. (4) and (3) yields

$$u_i(t + \varepsilon) - \sum_j K_{ij} u_j(t) = 0, \tag{5}$$

which is the ith equation of the final linear system. To solve elliptic equations it is assumed that the system has attained its stationary state so that the time dependence can be eliminated from Eq. (5), i.e., letting $u_i(t+\varepsilon) = u_i(t)$. It has also been shown that the formulation can be generalized to non-linear problems using blurred derivatives. Details on this method can be found elsewhere [10].

The main purpose of this paper is to show that classical wave equations also admit a natural path integral formulation, which can be used as the basis for a meshless computational method. The scope of the paper is as follows. In Section 2 it is shown how classical wave phenomena can be described in terms of propagators as in Eq. (1). Section 3 indicates how a functional integral can be set-up in terms of non-singular infinitesimal propagators. Section 4 is devoted to prove that the proposed functional integral does converge to the exact propagator in the appropriate limit. Finally, the last two sections describe a simple numerical technique and explore its numerical performance through some examples.

2. Formulation

Since the scalar wave equation can be recovered mathematically as a special case of Lamé's equations of elasticity, we consider elastodynamics (infinitesimal deformation) of an isotropic medium neglecting eigenstrains and body forces. The state of the body is given by the instantaneous displacement and velocity fields, which can be conveniently merged into a state vector

$$\mathbf{v}_{(\mathbf{x},t)} = (\mathbf{u}_{(\mathbf{x},t)}, \dot{\mathbf{u}}_{(\mathbf{x},t)}). \tag{6}$$

The state evolution (i.e., the equations of motion) are

$$\mu \frac{\partial^2 u_i}{\partial x_j^2} + (\lambda + \mu) \frac{\partial^2 u_j}{\partial x_i \partial x_j} = \rho \frac{\partial u_i}{\partial t}. \tag{7}$$

Here we seek a formulation in terms of an integral equation that maps the initial state into the actual one

$$\mathbf{v}_{(\mathbf{x},t+\tau)} = \int \mathbf{\Delta}_{(\mathbf{x},\bar{\mathbf{x}},\tau)} \mathbf{v}_{(\bar{\mathbf{x}},t)} \, \mathrm{d}^3\bar{x}. \tag{8}$$

The operator $\mathbf{\Delta}$ in Eq. (8) which in three space dimensions is a 6×6 matrix, is the source function, fundamental solution or propagator for this problem. We choose this last name because it suggests a connection with functional integrals, which is what we actually exploit in this work. The matrix elements of $\mathbf{\Delta}$ are found from the well-known integral equation for an unbounded domain [11]

$$u_{(x,t)} = \int u_{(\bar{x},0)} \dot{u}^*_{(x-\bar{x},t)} \, \mathrm{d}^3\bar{x} + \int \dot{u}_{(\bar{x},0)} u^*_{(x-\bar{x},t)} \, \mathrm{d}^3\bar{x}, \tag{9a}$$

where u^* is the Green function—the solution of the wave equation with a point source. Time differentiation of Eq. (9a) yields the corresponding formula for velocities

$$\dot{u}_{(x,t)} = \int u_{(\bar{x},0)} \ddot{u}^*_{(x-\bar{x},t)} \, \mathrm{d}^3\bar{x} + \int \dot{u}_{(\bar{x},0)} \dot{u}^*_{(x-\bar{x},t)} \, \mathrm{d}^3\bar{x}. \tag{9b}$$

Eq. (4) can be recast into a single evolution equation in terms of the propagator putting

$$\underline{\mathbf{\Delta}}^{12} = u^*, \quad \Delta^{11} = \Delta^{22} = \frac{\partial u^*}{\partial t}, \quad \Delta^{21} = \frac{\partial^2 u^*}{\partial t^2}. \tag{10}$$

For the scalar wave equation the matrix elements in Eq. (10) are scalars, while for Lamé's equations they are 3×3 matrixes. As an example, in the latter the Green function is [12]

$$\Delta_{ij}^{12} = U_{ij}^* = \frac{t}{4\pi y^2}\left\{\frac{1}{y}\left(3\frac{\partial y}{\partial x_i}\frac{\partial y}{\partial x_j} - \delta_{ij}\right)(U_{(y-c_1 t)} - U_{(y-c_2 t)}) + \frac{\partial y}{\partial x_i}\frac{\partial y}{\partial x_j}\left(\frac{\delta_{(y-c_1 t)}}{c_1} - \frac{\delta_{(y-c_2 t)}}{c_2}\right) + \frac{\delta_{(y-c_2 t)}}{c_2}\delta_{ij}\right\}, \tag{11}$$

where c_1 and c_2 are the longitudinal and transverse wave speeds ($c_1 = \sqrt{(\lambda + 2\mu)/\rho}$, $c_2 = \sqrt{\mu/\rho}$), and we used the abbreviation $\mathbf{y} = \mathbf{x} - \bar{\mathbf{x}}$.

In a finite domain the general solution (9) includes boundary terms. For a scalar wave equation for instance, we have

$$u_{(x,t)} = \int_0^t \int_\Gamma \left(u_{(\bar{x},0)}\frac{\partial u^*}{\partial n} - \frac{\partial u}{\partial n}u^*\right) \mathrm{d}\Gamma \, \mathrm{d}t' + \int u_{(\bar{x},0)} \dot{u}^*_{(x-\bar{x},t)} \, \mathrm{d}^3\bar{x} + \int \dot{u}_{(\bar{x},0)} u^*_{(x-\bar{x},t)} \, \mathrm{d}^3\bar{x}, \tag{12}$$

but in a broad class of problems boundary term can be made to vanish by a suitable choice of the Green function. For instance, if boundary conditions are $u = 0$ on Γ_u and $\partial u/\partial n = 0$ on $\Gamma - \Gamma_u$, it suffices to choose u^* satisfying those conditions to annihilate the boundary integrals.

3. Functional integral

Existence of the propagator for classical wave equations—addressed above—guarantees the existence of a path integral. Indeed, from Eqs. (8)–(10) it follows that

$$\underline{\mathbf{\Delta}}_{(x-\bar{x},t+\tau)} = \int \underline{\mathbf{\Delta}}_{(x-x_1,t)} \cdot \underline{\mathbf{\Delta}}_{(x_1-\bar{x},\tau)} \, \mathrm{d}x_1. \tag{13}$$

Hence, repeating convolution (13) for N small time steps ε and taking the limit $N \to \infty$, $N\varepsilon = t$, a functional integral is generated. But it is built in terms of the propagator. What we are interested in is just the converse: to *obtain* the propagator from a path integral built in terms of *more general objects* which are not propagators themselves

$$\Delta_{(x,\bar{x},\tau)} = \lim_{N\to\infty, N\varepsilon=\tau} \int A_{(x,x_1,\varepsilon)} \int A_{(x_1,x_2,\varepsilon)} \cdots \int A_{(x_N,\bar{x},\varepsilon)} \, \mathrm{d}^3x_1 \, \mathrm{d}^3x_2 \ldots \mathrm{d}^3x_N. \tag{14}$$

Matrixes $\mathbf{A}$ in formula (14) play the role of *infinitesimal propagators*—i.e., they are only valid for sufficiently small time intervals. To avoid confusion with the actual propagator they will be designated as *precursors*. To simplify derivations we specialize for plane strain taking $u_3 = 0$. Generalization to 3-D is most natural and simple. A possible representation of the *precursor* $\mathbf{A}$ is

$$\mathbf{A} = \begin{bmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{P}' & \mathbf{Q}' \end{bmatrix}. \tag{15}$$

The symbols **P** and **Q** in Eq. (15) stand now for 2×2 matrixes which are described below, while the prime denotes derivative with respect to ε:

$$\mathbf{P}' = \frac{\partial \mathbf{P}}{\partial \varepsilon}, \quad \mathbf{Q}' = \frac{\partial \mathbf{Q}}{\partial \varepsilon}. \tag{16}$$

In particular, **Q** is a diagonal matrix proportional to a delta function, $\mathbf{Q} = \varepsilon \mathbf{I}\delta_{(x-\bar{x})}$, where **I** is the identity. The form of the matrix elements **P** is not unique. A possible representation is

$$\mathrm{P}^{11}_{(x,\bar{x},\varepsilon)} = \frac{\mathrm{e}^{((-y_1^2/a\varepsilon^2)-(y_2^2/b\varepsilon^2))}}{\sqrt{\pi a}\varepsilon\sqrt{\pi b}\varepsilon}, \quad \mathrm{P}^{22}_{(x,\bar{x},\varepsilon)} = \frac{\mathrm{e}^{((-y_1^2/b\varepsilon^2)-(y_2^2/a\varepsilon^2))}}{\sqrt{\pi a}\varepsilon\sqrt{\pi b}\varepsilon},$$
$$\mathrm{P}^{12}_{(x,\bar{x},\varepsilon)} = \mathrm{P}^{21}_{(x,\bar{x},\varepsilon)} = \frac{\mathrm{e}^{-((y_1^2+y_2^2)/c\varepsilon^2)}}{\pi c\varepsilon^2}\frac{y_1 y_2}{c\varepsilon^2}, \tag{17}$$

where

$$y_i = x_i - \bar{x}_i, \quad a = 4\frac{(\lambda+2\mu)}{\rho}, \quad b = 4\frac{\mu}{\rho}, \quad c = 4\frac{(\lambda+\mu)}{\rho}. \tag{18}$$

To show that this is a correct description consider a sufficiently small time interval ε and approximate Eq. (14) as an ordinary integral

$$\mathbf{v}_{(\mathbf{x},t+\varepsilon)} \cong \iint \mathbf{A}_{(\mathbf{y},\varepsilon)} \cdot \mathbf{v}_{(\mathbf{x}+\mathbf{y},t)}\, \mathrm{d}^2 y, \tag{19}$$

Now, expand the left-hand side of Eq. (19) in powers of ε and the right side in powers of **y**, both up to second order, and notice that the only non-zero integrals of **P** are

$$\iint \mathrm{P}^{11}_{(yx,\varepsilon)}\, \mathrm{d}y^2 = 1, \quad \iint y_1^2 \mathrm{P}^{11}_{(y,\varepsilon)}\, \mathrm{d}y^2 = \frac{(\lambda+2\mu)}{\rho}\varepsilon^2,$$
$$\iint \bar{x}_2^2 \mathrm{P}^{11}_{(y,\varepsilon)}\, \mathrm{d}y^2 = \frac{\mu}{\rho}\varepsilon^2, \quad \iint y_1 y_2 \mathrm{P}^{12}_{(y,\varepsilon)}\, \mathrm{d}y^2 = \frac{(\lambda+\mu)}{\rho}\varepsilon^2. \tag{20}$$

Taking these into account, collecting terms and dividing by ε^2 the first Eq. (19) yields

$$\frac{\partial^2 u_1}{\partial t^2} - \left\{(\lambda+2\mu)\frac{\partial^2 u_1}{\partial x_1^2} + \mu\frac{\partial^2 u_1}{\partial x_2^2} + (\lambda+\mu)\frac{\partial^2 u_2}{\partial x_1 \partial x_2}\right\} = O_{(\varepsilon)}. \tag{21}$$

The second Eq. (14) yields a similar expression for u_2,

$$\frac{\partial^2 u_2}{\partial t^2} - \left\{(\lambda+2\mu)\frac{\partial^2 u_2}{\partial x_1^2} + \mu\frac{\partial^2 u_2}{\partial x_2^2} + (\lambda+\mu)\frac{\partial^2 u_1}{\partial x_1 \partial x_2}\right\} = O_{(\varepsilon)}. \tag{22}$$

Hence, in the limit $\varepsilon \to 0$ Eq. (21) are plain strain Lamé's equations. The second pair of Eq. (19), on the other hand, is equivalent to the time derivative of the first pair, so that it gives the time evolution of velocities. This comes from using definitions (16) and noticing that $\partial u_{i(x,t+\varepsilon)}/\partial\varepsilon = \dot{u}_{(x,t+\varepsilon)}$.

4. Convergence of the path integral

The fact that for an infinitesimal time step the wave equation is recovered—discussed in the previous section—does not guarantee the convergence of the path integral (14) to the exact propagator (7). Actually, this is the most difficult to prove for the equations of elastodynamics. However, setting $c_1 = c_2$ (i.e., $\lambda + \mu = 0$), Lamé's equations reduce to three uncoupled ordinary wave equations, one for each component of displacements, which are much easier to tackle. It is shown in this section that when precursors (17) of elastodynamics are restricted to this scalar case, the corresponding path integral does converge to the exact propagator. Since the path integral is defined as the limit of several convolutions, it is convenient to use a shorthand defining

$$a \cdot b = \int a_{(x-x_1)} \cdot b_{(x_1-\bar{x})}\, \mathrm{d}x_1 = f_{(x-\bar{x})}. \tag{23}$$

This operation is commutative and associative so that

$$\underline{\mathbf{A}}^{n+1} = \underline{\mathbf{A}}^n \cdot \underline{\mathbf{A}} = \underline{\mathbf{A}} \cdot \underline{\mathbf{A}}^n. \tag{24}$$

A sub-index will be used to designate the nth member of the family, namely

$$\underline{\mathbf{A}}_n = \underline{\mathbf{A}}^n = \begin{bmatrix} P_n & Q_n \\ P'_n & Q'_n \end{bmatrix}. \tag{25}$$

Notice however that $P_n \neq P^n$. Now, calling $\mathbf{L} = \lim_{\varepsilon \to 0, n \to \infty, n\varepsilon = ct}$, the conjecture to be proved is that

$$L(P_n) = \Delta^{11}, \quad L(Q_n) = \Delta^{12}. \tag{26}$$

We begin by looking for a recursion that provides P_n and Q_n in terms of the precursors Considering that $P_n Q = \varepsilon P_n$ and $Q_n Q' = Q_n$, and calling $\mathcal{Q} = Q/\varepsilon$ it follows from formulas (23) to (25) that

$$P_{n+1} = P_n P + \mathcal{Q}_n R, \quad \mathcal{Q}_{n+1} = P_n + \mathcal{Q}_n. \tag{27}$$

Defining $P_0 = \mathcal{Q}_1 = \delta$, the second expression in Eq. (27) can be eliminated by repeated feedback into itself

$$\mathcal{Q}_n = P_{n-1} + \mathcal{Q}_{n-1} = P_{n-1} + P_{n-2} + \mathcal{Q}_{n-2} = \cdots = \sum_{i=1}^{n-1} P_i + \mathcal{Q}_1 \equiv \sum_{i=0}^{n-1} P_i. \tag{28}$$

Substituting Eq. (28) into Eq. (27), there is now a single recursion formula to be solved

$$P_{n+1} = P_n P + R G_{n-1} \quad \text{where } G_l = \sum_{i=0}^{l} P_i. \tag{29}$$

To eliminate the sub-index "n"—which designates unknown elements—from the right-hand side of Eq. (27) we again feed expression (29) into itself and use the fact that $P_0 = P^0$ and $P_1 = P^1$. This process gives rise to a huge expression, so that we use the abbreviation $H^l = \sum_{k=0}^{l} P^k$. The

final result is

$$P_n = P^n + R\sum_{i_1=0}^{n=2} P^{i_1} H^{n-2-i_1} + R^2 \sum_{i_1=2}^{n=2}\sum_{i_2=0}^{i_1=2} P^{i_2} H^{n-2-i_1} H^{i_1-2-i_2} + \cdots$$
$$+ R^k \sum_{i_1=2k-2}^{n=2}\sum_{i_2=2k-4}^{i_1=2} \cdots \sum_{i_k=0}^{i_{k-1}=2} P^{i_k} H^{n-2-i_1} \cdots H^{i_{k-1}-2-i_k} + \cdots. \tag{30}$$

Formula (30) is valid for one, two- and three-dimensional scalar wave equations, and for any choice of precursors P since $R = \varepsilon\partial P/\partial\varepsilon$ by definition. To evaluate it explicitly we now specialize it to the space one dimension

$$P = \frac{\mathrm{e}^{-y^2/\varepsilon^2}}{\sqrt{\pi}\varepsilon}, \quad Q = \varepsilon\delta_{(y)}, \quad Q' = \delta_{(y)},$$

$$P' = \frac{1}{\varepsilon}R, \quad R = P(h-1) \quad \text{and} \quad h = 2\frac{y^2}{\varepsilon^2}. \tag{31}$$

The Green function and its first time derivative are $\Delta^{12} = (1/2c)U_{(\|x-\bar{x}\|-ct)}$, $\Delta^{11} = 1/2[\delta_{(x-\bar{x}-ct)} + \delta_{(x-\bar{x}+ct)}]$, where U is the step function ($U = 1$ for $x > 0$ and $U = 0$ for $x < 0$).

Calculations are preformed from now on in the Fourier Transform (FT) space—designated with an overbar—since it transforms convolutions into products of FTs. Hence, we have $\overline{P^\alpha} = \bar{P}^\alpha$. The variable in transformed space is $\underline{s}$. The FT of these operators are found to be [13]

$$\bar{P} = \mathrm{e}^{-s^2\varepsilon^2/4}, \quad \overline{Ph} = \bar{P}(1-\bar{b}) \quad \text{where } \bar{b} = \varepsilon^2 s^2/2. \tag{32}$$

Hence

$$\bar{R} = \overline{P(h-1)} = -\bar{b}\bar{P} \Rightarrow \overline{R^k} = \bar{R}^k = (-\bar{b})^k \bar{P}^k. \tag{33}$$

Expression (30) can now be written in the FT space. For this recall that the factors H^k are sums over powers of P so that they transform into sums of powers of $\bar{P}$ called $\overline{H^k}$. We have

$$\bar{P}_n = \bar{P}^n + (-\bar{b})\sum_{i_1=0}^{n=2} \bar{P}^{i_1+1}\overline{H^{n-2-i_1}} + (-\bar{b})^2 \sum_{i_1=2}^{n=2}\sum_{i_2=0}^{i_1=2} \bar{P}^{i_2+2}\overline{H^{n-2-i_1}}\,\overline{H^{i_1-2-i_2}} + \cdots$$
$$+ (-\bar{b})^k \sum_{i_1=2k-2}^{n=2}\sum_{i_2=2k-4}^{i_1=2} \cdots \sum_{i_k=0}^{i_{k-1}=2} P^{i_k+k} \bar{H}^{n-2-i_1} \cdots \bar{H}^{i_{k-1}-2-i_k} + \cdots. \tag{34}$$

Expression (34) can be viewed as a sum of *even powers* of the product (εs) with alternating signs. This suggests a natural road to follow, namely to expand *all* exponentials $\bar{P}$ in a power series

$$\bar{P}^\alpha = \sum_{l=0}^{\infty} \frac{(-\alpha)^l}{l!}\left(\frac{\varepsilon s}{2}\right)^{2l}$$

and collect all terms with equal powers of the product (εs). Although this may seem inconvenient due to the generation of a very large number of terms, it will be seen in the sequel that a massive cancellation of terms takes place as the limit $\varepsilon \to 0$, $n \to \infty$, $n\varepsilon = ct$ is approached. Replacing the expansion in Eq. (34) and collecting powers of (εs) leads obviously to an expression

of the form

$$\overline{P_n} = \sum_{l=0}^{\infty} (-1)^l (\varepsilon^2 s^2)^l a_{(n,l)}, \tag{35}$$

where $a_{(n,l)}$ are yet unknown coefficients.

Now, the *key point* of the derivation is that the coefficients $a_{(n,l)}$ are just polynomials of $\boldsymbol{n}$ of degree $2l$, which can be written

$$a_{(n,l)} = d_0^l n^{2l} + d_1^l n^{2n-1} + d_2^l n^{2n-2} + \cdots + d_{2l}^l. \tag{36}$$

Hence, in the limit $\varepsilon \to 0$, $n \to \infty$, $n\varepsilon = ct =$ constant all terms of Eq. (36) except the leading one, n^{2l}, will vanish so that the only relevant coefficient is d_0^l. Let us see this in detail.

The first polynomial coefficient of Eq. (35), $\boldsymbol{a}_{(n,0)}$, depends only on the power expansion of $\bar{P}^n$ in Eq. (34) since all remaining terms are pre-multiplied by powers of $(\varepsilon s)^2$. Hence we have $d_0^0 = 1$. The second term of Eq. (35), with $\boldsymbol{l} = 1$, depends on the second term in the power expansion of $\bar{P}^n$, $-n(\varepsilon s/2)^2$, and on the first sum in Eq. (34) which contributes with a factor

$$\sigma_1 = \sum_{i_1=0}^{n-2} \sum_{j=0}^{n-2-i_1} 1 = \sum_{i_1=0}^{n-2} (n-1-i_1) = \frac{n(n-1)}{2} \approx \frac{1}{2} n^2. \tag{37}$$

Consequently, only the first sum in Eq. (34) actually contributes to the leading term, d_0^1, of the polynomial coefficient $\boldsymbol{a}_{(n,1)}$. Similarly, the polynomial coefficient $\boldsymbol{a}_{(n,2)}$ of the following term, $(\varepsilon s)^4$, is made up from three contributions: $\bar{P}^n$, the first sum in Eq. (34) and the second sum in it. But $\bar{P}^n$contributes with a factor of order $\boldsymbol{n}^2$ and the first sum with a factor of order $\boldsymbol{n}^3$. Hence, only the *second* sum contributes to the leading coefficient d_0^2. Its contribution is

$$\begin{aligned}\sigma_2 &= \sum_{i_1=2}^{n-2} \sum_{i_2=0}^{i_1-2} \sum_{j_1=0}^{n-2-i_1} \sum_{j_2=0}^{i_1-2-i_2} 1 = \sum_{i_1=2}^{n-2} \sum_{i_2=0}^{i_1-2} (n-1-i_1)(i_1-2-i_2) = \frac{1}{2} \sum_{i_1=2}^{n-2} (n-1-i_1)(i_1^2 - i_1) \\ &\approx \frac{1}{2} \left\{ n \left[\sum_{i=1}^{n-2} i^2 - 1 \right] - \left[\sum_{i=1}^{n-2} i^3 - 1 \right] \right\} \approx \frac{(n-1)(n-2)}{2} \left\{ \frac{n(2n-3)}{6} - \frac{(n-1)(n-2)}{4} \right\}. \end{aligned} \tag{38}$$

Notice that in the preceding calculations we have systematically neglected lower order terms since only the leading coefficient is relevant for our purposes. That final result is $d_0^2 \approx n^4/4!$.

Generalizing the previous line of reasoning it can be seen that only the kth sum in Eq. (34) contributes to the leading coefficient d_0^k yielding the proper term $(-1)^k (n\varepsilon s)^{2k} d_0^k$ to the power expansion of $\bar{P}_n$ we are building. Hence, all that is required is to estimate

$$\sigma_k = \sum_{i_i=2k-2}^{n-2} \cdots \sum_{i_k=0}^{i_{k-1}-2-i_k} (n-1-i_1)(i_1-1-i_2) \cdots (i_{k-1}-1-i_k). \tag{39}$$

The sums in Eq. (39) are processed from right to left. After performing the rightmost sum, over the index i_k, the last factor in the preceding Eq. (39) is now $i_{k-1}(i_{k-1}-1)/2$. Hence, this operation contributes a factor $\frac{1}{2}$ to the leading coefficient d_0^k we are seeking. And the last index, i_{k-1}, now appears with two powers: 1 and 2. Elimination of the following sum, over i_{k-1}, will contribute to a

factor of the form

$$F_{k-1} \approx i_{k-2} \sum_{i_{k-1}=1}^{i_{k-2}-2} i_{k-1}^2 - \sum_{i_{k-1}=1}^{i_{k-2}-2} i_{k-1}^3 \approx \frac{i_{k-2}^4}{3 \times 4}.$$

So that the leading coefficient d_0^k now contains a total factor: $1/(2 \times 3 \times 4) = 1/4!$.

This process continues so that elimination of a sum increases the power of the last remaining index in two units, i.e.,

$$F_{k-j+1} \approx i_{k-j} \sum^{i_{k-j}} i_{k-j+1}^{2j-1} - \sum^{ik-j} i_{k-j+1}^{2j}. \tag{40}$$

It is now possible to calculate all factors contributing to the leading coefficient. In fact, calling $S_l^m = \sum_{r=1}^m i_v^l$, and taking into account that [14]

$$\binom{l+1}{1} S_1^m + \binom{l+1}{2} S_2^m + \cdots + \binom{l+1}{l} S_l^m = (m+1)^{l+1} - (m+1),$$

it can be seen that

$$mS_{2j-2}^m - S_{2j-1}^m \approx \frac{1}{2j-1} m m^{2j-1} - \frac{1}{2j} m^{2j} = \frac{m^{2j}}{(2j-1)(2j)}. \tag{41}$$

Collecting all factors, the contribution of the kth sum to the leading coefficient d_0^k will then be

$$d_0^k = \frac{1}{1.2} \frac{1}{3.4} \cdots \frac{1}{(2k-1)(2k)} = \frac{1}{(2k)!}.$$

Hence, as $\boldsymbol{n}$ tends to infinity the power expansion of $\bar{P}_n$ can be approximated as

$$\bar{P}_n \approx \sum_{k=0}^{\infty} (-1)^k \frac{(n\varepsilon s)^{2k}}{(2k)!} = \cos(n\varepsilon s). \tag{42}$$

But the cosine is the FT of two delta functions symmetric about the origin

$$\overline{\tfrac{1}{2}(\delta_{(x-g)} + \delta_{(x-g)})} = \cos(gs). \tag{43}$$

Hence,

$$\mathbf{L}(P_n) = \tfrac{1}{2}[\delta_{(y-n\varepsilon)} + \delta_{(y+n\varepsilon)}] = \Delta^{11}. \tag{44}$$

It is now easy to see from formulas (30) and (43) that

$$\mathbf{L}(Q_n) = \mathbf{L}\left(\varepsilon \sum_{i=0}^{n-1} P_i\right) \approx \mathbf{L}\left(\tfrac{1}{2}\varepsilon \sum_{i=0}^{n-1} [\delta_{(y-i\varepsilon)} + \delta_{(y+i\varepsilon)}]\right) = \Delta^{12}.$$

This completes the proof.

5. Numerical scheme

5.1. Approximation

The path integral formulation of elastodynamics provides an adequate framework for mesh-free numerical methods. For this, notice that the evolution of the system during a sufficiently small time interval can be adequately approximated by "updating formula" (19) that we reproduce below for convenience

$$\mathbf{v}_{(\mathbf{x},t+\varepsilon)} \cong \iint \mathbf{A}_{(\mathbf{y},\varepsilon)} \cdot \mathbf{v}_{(\mathbf{x}+\mathbf{y},t)}\, \mathrm{d}^2 y.$$

Taking into account that precursors (17) are rapidly decaying functions of position, Eq. (19) can be interpreted as follows: the state of a point $\mathbf{x}$ and at time $t+\varepsilon$ is obtained by properly weighting the state at time t in the neighborhood of $\mathbf{x}$. Hence, unlike variational principles, application of Eq. (19) does not require partitions of the domain for integration such as finite element discretization. Moreover, evaluation of Eq. (19) does not involve derivatives of approximation functions. So that regular nodal arrays as those used for finite differencing are not required.

To obtain a computational method based on the present formulation the continuous field $\mathbf{u}_{(x)}$ is approximated, as usual, in terms of a suitable linear combination of N basis functions $\Phi_{i(\mathbf{x})}$ with coefficients u^i_α ($i = 1, \ldots, N$; $\alpha = 1, 2$)

$$u^i_{\alpha(x)} = \sum_{i=1}^{N} \Phi_{i(x)} u^i_\alpha. \tag{45}$$

The simplest approximation for the displacement field is ordinary least-squares fitting of a quadratic polynomial for each node. For this, the domain is filled with points (nodes) which need not be connected forming "elements". For each node, a convenient cloud of nodes is selected around it and the displacement field is locally approximated as $\tilde{\mathbf{u}}_{(x,y)} = \mathbf{a} \cdot \mathbf{R}_{(x,y)}$, where

$$\tilde{\mathbf{u}}_{(x,y)} = (u_{1(x,y)}, u_{2(x,y)})^{\mathrm{t}}, \quad \mathbf{R}_{(x,y)} = (1, x, y, x^2, xy, y^2)^{\mathrm{t}}, \tag{46}$$

$$\mathbf{a} = \begin{bmatrix} a^1_1 & a^1_2 & a^1_3 & a^1_4 & a^1_5 & a^1_6 \\ a^2_1 & a^2_2 & a^2_3 & a^2_4 & a^2_5 & a^2_6 \end{bmatrix}. \tag{47}$$

The coefficient matrix $\mathbf{a}$ is calculated in terms of M nodal displacements $\mathbf{u}$ solving [15]

$$\mathbf{a} = (\mathbf{V}^{\mathbf{t}} \cdot \mathbf{V})^{-\mathbf{1}} \cdot \mathbf{V}^{\mathbf{t}} \cdot \tilde{\mathbf{u}}, \tag{48}$$

where

$$\tilde{\mathbf{u}} = \begin{bmatrix} u^0_1 & u^1_1 & \cdots & u^M_1 \\ u^0_2 & u^1_2 & \cdots & u^M_2 \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} 1 & x_0 & y_0 & x_0^2 & x_0 y_0 & y_0^2 \\ 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ \vdots & & & & & \vdots \\ 1 & x_M & y_M & x_M^2 & x_M y_M & y_M^2 \end{bmatrix}. \tag{49}$$

The "shape" functions for this scheme are

$$\Phi = \mathbf{R}^{\mathbf{t}} \cdot (\mathbf{V}^{\mathbf{t}} \cdot \mathbf{V})^{-\mathbf{1}} \cdot \mathbf{V} \quad \text{where } \boldsymbol{\Phi} = (\varphi_{1(x)}, \varphi_{2(x)}, \ldots, \varphi_{N(x)}). \tag{50}$$

Feeding this approximation of the displacement field into the evolution Eqs. (19) yields the following updating formula for the first component of displacement of each node:

$$u^1_{i(t+\varepsilon)} =^i a^1_{11(t)} + (\lambda+2\mu)\varepsilon^2 i a^1_{4(t)} + \tfrac{1}{2}(\lambda+\mu)\varepsilon^2 i a^2_{5(t)} + \mu\varepsilon^2 i a^1_{6(t)} + \varepsilon \dot{u}^\alpha_{i(t)}, \tag{51}$$

$$\dot{u}^1_{i(t+\varepsilon)} = 2(\lambda+2\mu)\varepsilon^i a^1_{4(t)} + (\lambda+\mu)\varepsilon^i a^2_{5(t)} + 2\mu\varepsilon^i a^1_{6(t)} + \dot{u}^1_{i(t)}$$

and two similar equations for the other component.

At the assembly stage, the coefficients a^j_i are not fixed numbers but linear combinations of nodal displacements which depend on the number and location of nodes in the local cloud of the current node. As an example, if the local cloud contains nine nodes forming a regular array with nodal distance h, the relevant coefficients are

$$\begin{aligned}
a_1 &= \tfrac{1}{9}[5u_0 + 2(u_1+u_3+u_5+u_7) - (u_2+u_4+u_6+u_8)],\\
a_4 &= \frac{1}{3h^2}\big[-u_0 + \tfrac{1}{2}(u_1+u_2+u_4+u_5+u_6+u_8) - (u_3+u_7)\big],\\
a_5 &= \frac{1}{4h^2}[u_2+u_6-u_4-u_8],\\
a_6 &= \frac{1}{3h^2}\big[-u_0 + \tfrac{1}{2}(u_2+u_3+u_4+u_6+u_7+u_8) - (u_1+u_5)\big].
\end{aligned} \tag{52}$$

In Eq. (52) a local numbering was used with node "0" at the center and the surrounding nodes labelled from 1 to 8 counterclockwise. In this particular case the nodal equations can be written as

$$\begin{aligned}
u^1_{0(t+\varepsilon)} = {} & \big[\tfrac{5}{9} - \zeta\big]u^1_{0(t)} + \big[\tfrac{2}{9} + A\zeta\big](u^1_1+u^1_5) + \big[\tfrac{2}{9} - B\zeta\big](u^1_3+u^1_7)\\
& + \big(-\tfrac{1}{9} + \tfrac{1}{2}\zeta\big)(u^1_2+u^1_4+u^1_6+u^1_8) + C\zeta(u^2_2-u^2_4+u^2_6-u^2_8) + \varepsilon\dot{u}^1_{0(t)},
\end{aligned} \tag{53a}$$

$$\begin{aligned}
\varepsilon\dot{u}^1_{0(t+\varepsilon)} = {} & -2\zeta u^1_{0(t)} + 2\zeta A(u^1_1+u^1_5) - 2\zeta B(u^1_3+u^1_7)\\
& + \zeta(u^1_2+u^1_4+u^1_6+u^1_8) + 2\zeta C(u^2_2-u^2_4+u^2_6-u^2_8) + \varepsilon\dot{u}^1_{0(t)},
\end{aligned} \tag{53b}$$

where

$$\zeta = \frac{(\lambda+3\mu)\varepsilon^2}{3\rho.h^2}, \quad A = \frac{\lambda}{2(\lambda+3\mu)}, \quad B = \frac{(2\lambda+3\mu)}{2(\lambda+3\mu)}, \quad C = \frac{3}{8}\frac{(\lambda+\mu)}{(\lambda+3\mu)}.$$

And a similar set of equations results for the other component, u^2_0.

It is interesting to notice that for model equilibrium problems—i.e., elastostatics—it suffices to enforce a no evolution condition. For this, velocities are set to zero and Eq. (19) is replaced by $\mathbf{v}_{(\mathbf{x})} \cong \iint \mathbf{A}_{(\mathbf{y},\varepsilon)} \cdot \mathbf{v}_{(\mathbf{x}+\mathbf{y})}\, \mathrm{d}^2 y$. It can be shown that this leads to the same algebraic equations as the numerical scheme presented in Ref. [9].

5.2. Boundary conditions

While weak formulations provide a natural framework for numerical implementation of traction boundary conditions—stemming from the boundary integral term—the present formulation requires some *ad hoc* procedure for this. Actually, the same is true of methods based on strong formulations such as SPH or GFDM. In our implementation we proceeded as follows. Dirichlet boundary conditions are satisfied in a trivial manner by just enforcing the

prescribed displacement at those points which lie in this part of the boundary. For nodes on the boundary where a traction condition is to be satisfied, we assumed that the dynamic statement (19) is also valid, but the local approximation of the displacement field is enforced to satisfy the prescribed condition. To clarify this issue let us consider a boundary point with outward normal parallel to the Y co-ordinate axis. The two components of the traction vector—assuming as before plain strain—are

$$\frac{\partial u_1}{\partial x_1}+\frac{\partial u_2}{\partial x_2}=\frac{t_1}{\mu},$$
$$\frac{\partial u_2}{\partial x_1}+\kappa\frac{\partial u_1}{\partial x_2}=\frac{t_2}{\lambda+2\mu},\quad \kappa=\frac{\lambda}{\lambda+2\mu}. \tag{54a}$$

In the neighborhood of the current point the displacement field is given by the polynomial approximation described above, namely

$$u_1(x,y)=a_1^1+a_2^1x+a_3^1y+a_4^1x^2+a_5^1xy+a_6^1y^2, \tag{55a}$$

$$u_2(x,y)=a_1^2+a_2^2x+a_3^2y+a_4^2x^2+a_5^2xy+a_6^2y^2. \tag{55b}$$

where, as stated before, the origin of co-ordinates x and y is locally placed on the current boundary point. Hence, by enforcing conditions (54) on approximation (55) we obtain two relations among the coefficients a_i^j of the latter

$$a_3^1+a_2^2=\frac{t_1}{\mu},\quad a_3^2+\kappa a_2^1=\frac{t_2}{\lambda+2\mu}. \tag{56}$$

Using Eq. (56) the local approximation (55) of the displacement field can be rewritten

$$u_1(x,y)=a_1^1+a_2^1x-a_2^2y+a_4^1x^2+a_5^1xy+a_6^1y^2+\frac{t_1}{\mu}y, \tag{57a}$$

$$u_2(x,y)=a_1^2+a_2^2x-\kappa a_2^1y+a_4^2x^2+a_5^2xy+a_6^2y^2+\frac{t_2}{\lambda+2\mu}y. \tag{57b}$$

The structure of matrix $\mathbf{V}$ is no longer given by Eq. (49) but has to be recalculated as

$$\mathbf{V}=\begin{bmatrix}
1 & x_0 & -y_0 & x_0^2 & x_0y_0 & y_0^2 & 0 & 0 & 0 & 0\\
1 & x_1 & -y_1 & x_1^2 & x_1y_1 & y_1^2 & 0 & 0 & 0 & 0\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
1 & x_M & y_M & x_M^2 & x_My_M & y_M^2 & 0 & 0 & 0 & 0\\
0 & -\kappa y_0 & x_0 & 0 & 0 & 0 & 1 & x_0^2 & x_0y_0 & y_0^2\\
0 & -\kappa y_1 & x_1 & 0 & 0 & 0 & 1 & x_1^2 & x_1y_1 & y_1^2\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
0 & -\kappa y_M & x_M & 0 & 0 & 0 & 1 & x_M^2 & x_My_M & y_M^2
\end{bmatrix}. \tag{58}$$

The coefficient matrix $\mathbf{a}$ is again given by Eq. (48) but now matrix $\mathbf{V}$ is given by Eq. (58). This structure is only valid when the outward normal points to the positive Y-axis. Hence, for each boundary point with prescribed tractions the program first rotates locally the co-ordinates so that Eq. (54) apply, it then calculates the two contributions to the global matrix (one for each

component of displacement) using the preceding equations and store them in an auxiliary array. Finally, these equations are rotated to the original co-ordinate system and added to the global matrix.

For node on a corner there are two traction vectors, one on each side. In this case, Eq. (56) account for one these vectors, while a new one has to be added. Assuming for simplicity that the corner is traction free, this is

$$\sigma_{11} = (\lambda + 2\mu)\frac{\partial u_1}{\partial x_1} + \lambda\frac{\partial u_2}{\partial x_2} = 0 \;\Rightarrow\; a_2^1 = 0. \tag{59}$$

There are now nine coefficients a_j^i to specify. Hence, the new matrix $\mathbf{V}$ is obtained from Eq. (58), eliminating the second column.

6. Dynamics

6.1. Eigenvalues and eigenvectors

Determination of normal modes of vibration and eigenfrequencies starts by assuming a harmonic motion

$$\mathbf{u}_{(\mathbf{x},t)} = \mathbf{u}_{(\mathbf{x})}\mathrm{e}^{\mathrm{i}wt}. \tag{60}$$

Evolution of displacements and velocities are then given by

$$\mathbf{u}_{(\mathbf{x},t+\varepsilon)} = \mathbf{u}_{(\mathbf{x})}\mathrm{e}^{\mathrm{i}wt}\mathrm{e}^{\mathrm{i}w\varepsilon}, \quad \dot{\mathbf{u}}_{(\mathbf{x},t)} = (\mathrm{i}w)\mathbf{u}_{(\mathbf{x})}\mathrm{e}^{\mathrm{i}wt}. \tag{61}$$

Recalling that the diagonal sub-matrix matrix $\mathbf{Q}$ is proportional to a delta function

$$Q^{11} = Q^{22} = \varepsilon\delta_{(x-\bar{x})}, \tag{62}$$

the infinitesimal updating prescription for displacements (first pair of Eq. (19)) then reads

$$\mathbf{u}_{(\mathbf{x})}\mathrm{e}^{\mathrm{i}w\varepsilon} = \int \mathbf{P}_{(\mathbf{x},\mathbf{y},\varepsilon)} \cdot \mathbf{u}_{(\mathbf{y})}\, \mathrm{d}^2 y + \mathrm{i}w\varepsilon\mathbf{u}_{(\mathbf{x})}. \tag{63}$$

Discretization of Eq. (63) proceeds as explained in the preceding section. In particular, if each internal node is linked to its eight neighbors which form a regular array, Eq. (63) is just Eq. (53a) making the replacement (61) in the nodal displacements. The result is a non-symmetric system of algebraic equations, whose eigenvalues are pairs of complex conjugate numbers

$$\lambda = \pm\mathrm{i}w\varepsilon - \exp(\pm\mathrm{i}w\varepsilon). \tag{64}$$

From this the eigenfrequencies can be computed as

$$w = \frac{1}{\varepsilon}\left[\mathrm{Im}(\lambda) + \sqrt{1 + \mathrm{Re}^2(\lambda)}\right]. \tag{65}$$

As is usual in numerical analysis, only the lower modes of vibration with wavelengths sufficiently larger than the nodal spacing can be accurately calculated. This implies $w\varepsilon \ll 1$. Hence, for the lower frequencies the real part of λ is very close to unity and the imaginary part is very close to zero. An alternative procedure is to use the *second* pair of Eq. (19) for eigenvalue

extraction, which leads to

$$\mathrm{i}we^{\mathrm{i}w\varepsilon}\mathbf{u}_{(\mathbf{x})} = \int \mathbf{P}'_{(\mathbf{x},\mathbf{y},\varepsilon)} \cdot \mathbf{u}_{(\mathbf{y})}\, \mathrm{d}^2 y + \mathrm{i}w\mathbf{u}_{(\mathbf{x})}. \tag{66}$$

The eigenvalues are now

$$\lambda = \mathrm{i}w[1 - e^{\mathrm{i}w\varepsilon}] \to \|\lambda\|^2 = w^2 2[1 - \cos(w\varepsilon)]. \tag{67}$$

Hence, eigenfrequencies can be extracted from the modulus of eigenvalues using a second order Taylor expansion of the cosine

$$w \cong \sqrt{\frac{\|\lambda\|}{\varepsilon}}. \tag{68}$$

6.2. Numerical example

To test the performance of the technique the free vibrations of a cantilever plate were analyzed. Bending frequencies of plates are related to the wave vector κ by

$$f_n^T = \frac{\kappa_n^2}{2\pi}\sqrt{\frac{t^2 E}{12\rho(1-\nu^2)}}, \tag{69}$$

where t is the plate's thickness. For a cantilever plate of length l the wave vectors are given by the transcendental equation $\cos(\kappa l)\cosh(\kappa l) + 1 = 0$. Longitudinal vibration frequencies on the other hand are given by

$$f_n^L = \frac{(2n+1)}{4l}\sqrt{\frac{E}{\rho(1-\nu^2)}}, \quad n = 0, 1, 2, \ldots. \tag{70}$$

A plate of length $l = 15$ cm and thickness 1.5 cm was modelled using the set of 205 nodes depicted in Fig. 1. In order to define the local cloud of each node we used the eight closest neighbors as described in Section 5. Material properties of polypropylene were used: $E = 1400$ MPa, $\nu = 0.3$ and $\rho = 0.905\,\mathrm{g/cm^3}$. The bests results were obtained with a non-dimensional time step $\sqrt{\zeta} = \sqrt{(\lambda + 3\mu)/3\rho}(\varepsilon/h) = 0.25$ where h is the nodal spacing. Eigenvalue and eigenvector extraction was performed using a standard routine DEVCRG of IMSL package. After solution the performance index τ was computed using routine DEPIRG of that package. The solution is considered excellent for $\tau < 1$, good if $1 < \tau < 100$ and poor if $\tau > 100$. We obtained $\tau = 0.001$. Comparison of analytical and numerical solutions for the first eight modes as shown in Table 1 indicates that the agreement is excellent. It is worth mentioning that up to the 25th mode the

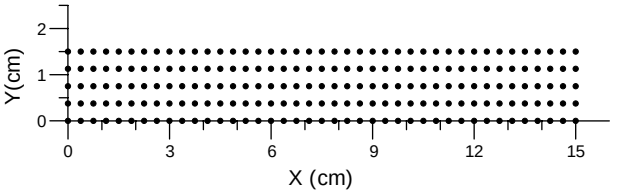

Fig. 1. Array of 205 points used to model cantilever plate.

Table 1
First eight vibration frequencies for cantilever plate; comparison between analytical and numerical solutions

Mode	Frequency (Hz)		Relative error (%)	Mode type
	Exact	Numerical		
1	140	144	2.8	Transverse
2	880	873	0.8	Transverse
3	2172	2180	0.4	Longit.
4	2464	2320	5.7	Transverse
5	4828	4206	12.	Transverse
6	6520	6531	0.2	Longit.
7	8001	8900	11.	Transverse
8	10 866	10 850	0.1	Longit.

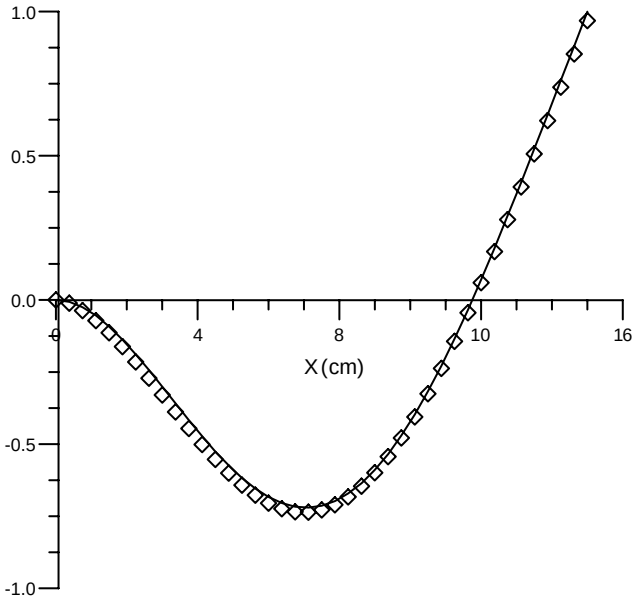

Fig. 2. Second vibration mode (traverse displacement) of cantilever plate; ◇, is analytical solution.

imaginary part of eigenvectors was computed as zero by the routine, so that frequencies were extracted from the real part which is very close to one. Obviously, eigenvectors are real for these frequencies. Figs. 2 and 3 show the second and third vibration modes, respectively, where the former is traverse and the latter longitudinal. It can be seen that in both cases the numerical prediction is in very close agreement with the exact solution, specially taking into account that the mesh used is rather coarse. It is important to point out that the preceding results were obtained with an empirically estimated optimum non-dimensional parameter ζ. This is only valid for the present connectivity (i.e., eight closest neighbors for each node). We have found that departure of ζ up to 20% from its optimal value has little effect on numerical results but larger variations are deleterious.

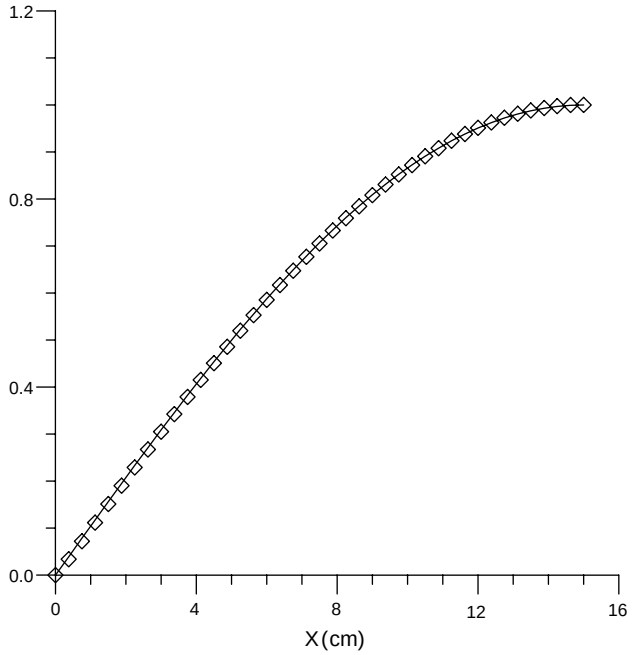

Fig. 3. Third vibration mode (longitudinal displacement) of cantilever plate; ◇, analytical solution.

6.3. Transient problems

Direct application of the approximate updating prescription (19) leads to an explicit time marching scheme. Since velocities are now non-zero there are four equations per node in two dimensions

$$u^{\alpha}_{i(t+\varepsilon)} = \int \mathrm{P}^{\alpha\beta}_{(x^i,y,\varepsilon)} \Phi_{j(y)}\, \mathrm{d}^2 y u^{\beta}_{j(t))} + \varepsilon \dot{u}^{\alpha}_{i(t)}, \quad \dot{u}^{\alpha}_{i(t+\varepsilon)} = \int \mathrm{P}'_{(x^i,y,\varepsilon)} \alpha\beta \Phi_{j(y)}\, \mathrm{d}^2 y u^{\beta}_{j(t))} + \dot{u}^{\alpha}_{i(t)}. \tag{71}$$

Eq. (71) make it clear that, as mentioned above, this is an explicit scheme and hence it is expected to be conditionally stable. Although it is difficult to perform a stability analysis in the most general case, an indication of the limiting time step ε can be obtained in the special case of a regular array of nodes, each one "connected" to its eight closest neighbors, and using quadratic polynomial approximation. In this case, the updating formula for the first component of displacements and velocities at an internal node are given by Eq. (53) of Section 5. Hence, the stability locus can be estimated using standard arguments: the discrete Eq. (53) are expressed in the general form: $\underline{\mathbf{T}} \cdot \mathbf{v}_{n+1} = \underline{\mathbf{S}} \cdot \mathbf{v}_n$ where $\mathbf{v}_n = (u^1, u^2, \dot{u}^1, \dot{u}^2)^{\mathrm{t}}$, and the solution is assumed to be of the form $\mathbf{v}_{n+k} = \lambda^k \mathbf{v}_n$. The characteristic equation of the system is then $\det[\lambda \underline{\mathbf{T}} - \underline{\mathbf{S}}] = 0$, which yields a polynomial of fourth degree in λ:

$$\left[\lambda^2 + \left(\zeta - \tfrac{14}{9}\right)\lambda + \zeta + \tfrac{5}{9}\right]^2 = 0. \tag{72}$$

Oscillatory motion requires the solutions of Eq. (72) to be complex eigenvalues which in turn demands the discriminant to be negative. Under this condition the eigenvalues can be expressed in

polar form as $\lambda = \rho e^{i\theta}$ and their modulus is found to be: $\rho = \zeta + \frac{5}{9}$. The numerical solution will be *stable* only if $\rho \leqslant 1$ which implies $\zeta \leqslant \frac{4}{9}$. Hence, stability requires the time step to be

$$\varepsilon \leqslant \varepsilon_{\lim} = \frac{2}{\sqrt{3}} \frac{\sqrt{\rho} h}{\sqrt{(\lambda + 3\mu)}}. \tag{73}$$

6.4. Numerical example

To test the performance of the simple time marching scheme just described we simulated the free vibrations of the cantilever plate described in the previous example. Initial velocities were set to zero and displacements correspond to the third free vibration mode shown in Fig. 3. Numerical simulations required the time step to be substantially smaller than estimated by formula (73), which can be attributed to the boundary conditions. Indeed, for nodes with traction boundary conditions Eq. (53) do not apply because the coefficients a_i^j are no longer given by Eq. (52) but have to be recalculated as indicated in Section 5.2. Figs. 4 and 5 show the evolution of displacements and velocities for the first three cycles. They were obtained using a time step $\varepsilon = 0.03\varepsilon_{\lim}$. It can be seen that in both cases the numerical simulation agrees very well with theory. Notice, however, that stability imposes a stringent limitation on the time step, as is usual in explicit schemes. In fact, the time marching scheme just described is analogous to the Euler forward time integration formula. But it is not the only possible one, and in fact there exist many other possibilities. Since elastodynamics is a conservative process it is possible to move backwards in time—as is the case of all wave equations without dissipation. Hence, if the updating formula (19) is replaced by its backward evolving counterpart an implicit scheme resembling Euler backward would be obtained. For this, it suffices to change the sign of the sub-matrixes $\mathbf{Q}$ and $\mathbf{Q}'$

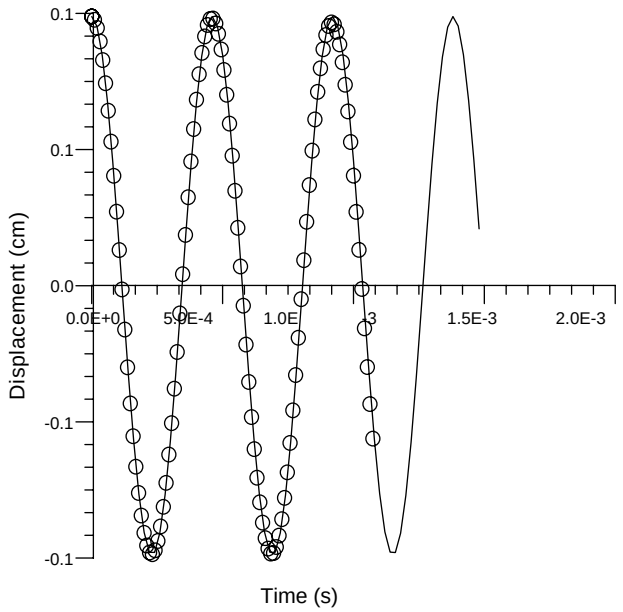

Fig. 4. Beam tip displacement versus time: ——, analytical; ○, numerical.

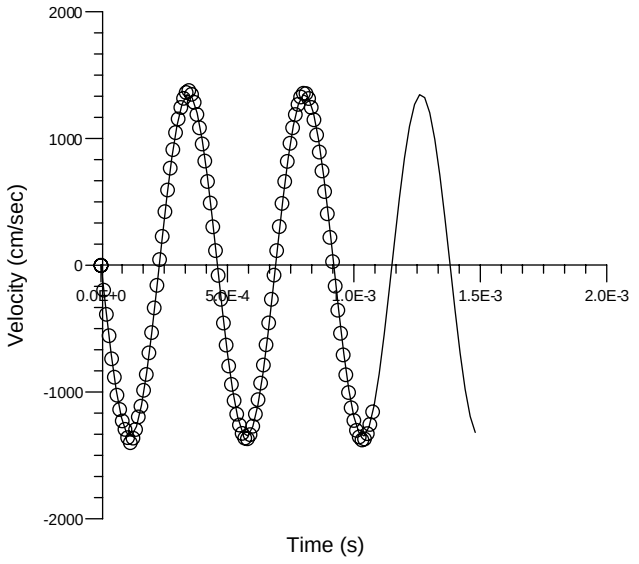

Fig. 5. Beam tip velocity versus time: ——, analytical; ○, numerical.

in the precursor $\mathbf{A}$ to obtain a the backward evolving precursor $\overleftarrow{\mathbf{A}}$. More generally, it is also possible to move partly forward and partly backward in time. Hence, a general updating prescription is

$$\int \overleftarrow{\mathbf{A}}_{(\varepsilon_2)} \cdot \mathbf{v}_{(\bar{x},t+\varepsilon_1+\varepsilon_2)}\, \mathrm{d}\bar{s} = \int \mathbf{A}_{(\varepsilon_1)} \cdot \mathbf{v}_{(\bar{x},t)}\, \mathrm{d}\bar{s}. \tag{74}$$

Calling $\varepsilon_1 = \eta\varepsilon$ and $\varepsilon_2 = (1-\eta)\varepsilon$, $0 \leqslant \eta \leqslant 1$ it is seen that Eq. (74) contains a new parameter η which could be adjusted to provide a wide stability locus while minimizing numerical damping. Stability analysis of such a general formula, which we have not yet performed might provide a reliable and accurate time marching scheme.

7. Final remarks

The preceding examples show that beyond its academic interest, the path integral formulation of elastodynamics—and of other classical wave equations—provides an appropriate framework for meshless computational implementations. In this respect, it is worth emphasizing that the simple approximation used here for the displacement field is not the only possible one. This suggests that a whole new breed of meshless computational methods can be devised on the basis of the functional integral formalism.

Finally, it is important to mention that the present formalism can be generalized to some dissipative problems in a very natural way. For this, notice that elastodynamics corresponds to the case where the $\mathbf{Q}$ sub-matrix of the precursor $\mathbf{A}$ is proportional to a delta function. Now,

assume that the elements of **Q** are Gaussians of the form

$$Q^{ij} = \varepsilon \frac{e^{-(y^2/d\varepsilon^2)}}{\sqrt{\pi}\, d\varepsilon} \delta_{ij} \tag{75}$$

and repeat the reasoning used in Section 3, Eqs. (14)–(16), to obtain the associated differential equation in the limit $\varepsilon \to 0$. In such case, the associated differential equation would be similar to Lamé's but terms proportional to the gradient of the rate-of-deformation tensor would appear. The coefficients of such terms—i.e., the parameter of the Gaussians of **Q**—can be adjusted to account for viscous effects such as viscoelasticity. Elastodynamics is then recovered as the limit of negligible viscosity, $d \to 0$.

References

[1] T. Liszka, An interpolation method for an irregular net of nodes, International Journal for Numerical Methods in Engineering 20 (1984) 1599–1612.
[2] J.J. Monaghan, An introduction to SPH, Computer Physics Communications 48 (1988) 89–96.
[3] T. Belytschko, et al., Meshless methods: an overview and recent developments, Computer Methods in Applied Mechanics and Engineering 139 (1996) 3–47.
[4] B. Nayroles, G. Touzot, P. Villon, Generalizing the finite element method: diffuse approximation and diffuse elements, Computer Mechanics 10 (1992) 307–318.
[5] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, International Journal for Numerical Methods in Engineering 20 (1995) 1081–1106.
[6] C.A. Duarte, J.T. Oden, H–p adaptive method using clouds, Computer Methods in Applied Mechanics and Engineering 139 (1996) 237–262.
[7] I. Babuska, J.M. Melenk, The partition of unity method, International Journal for Numerical Methods in Engineering 40 (1997) 727–758.
[8] E. Pardo, Meshless method for linear elastostatics based on a path integral formulation, International Journal for Numerical Methods in Engineering 47 (2000) 1463–1480.
[9] E. Pardo, Convergence and accuracy of the path integral approach for elastostatics, Computer Methods in Applied Mechanics and Engineering 191 (20) (2002) 2219–2247.
[10] E. Pardo, Blurred derivatives and meshless methods, International Journal for Numerical Methods in Engineering 56 (2) (2003) 295–324.
[11] C.A. Brebbia, J.C.F. Telles, L.C. Wrobel, Boundary Element Techniques, Springer, Berlin, 1984.
[12] A.C. Eringen, E.S. Suhubi, Elastodynamics, Academic Press, London, 1975.
[13] P.M. Morse, H. Feshback, Methods of Theoretical Physics, McGraw-Hill, New York, 1953.
[14] M.R. Spiegel, Mathematical Handbook of Formulas and Tables, McGraw-Hill, New York, 1988.
[15] P. Lancaster, K. Salkauskas, Curve and Surface Fitting, Academic Press, New York, 1986.