Chapter 9
Meshfree Methods in Moving Load Problems

9.1 Meshless Methods (Element-Free Galerkin Method)

The idea of meshless methods is to eliminate the mesh generation stage, which is the main disadvantage of the finite element method (or other classical discrete methods). In a meshless method, the set of separated points is placed in the domain of the structure. Interpolation functions (shape functions) are then generated not in element subdomains, but in arbitrarily placed nodal points.

In the case of very large deformations it is necessary to modify the mesh of finite elements step by step. In order to minimize this significant cost of computation, researchers are trying to use meshfree methods, e.g., [27, 28, 46, 90]. These methods also seem to be appropriate for the task of a moving inertial load.

The shape functions are stretched on points which are in the neighbourhood of the given point, in the stepping subdomain Ω (Figure 9.1). We determine the shape function on the basis of the moving least squares criterion (MLS) [82]. This method consists in minimizing the differences between the exact and approximate solution. The sum of squared errors of the approximation at all the nodes ($i = 1, \ldots, m$) is a functional:

$$J = \sum_{i=1}^{m} W(x - x_i) \left[w^h(x, x_i) - w_i \right]^2 , \tag{9.1}$$

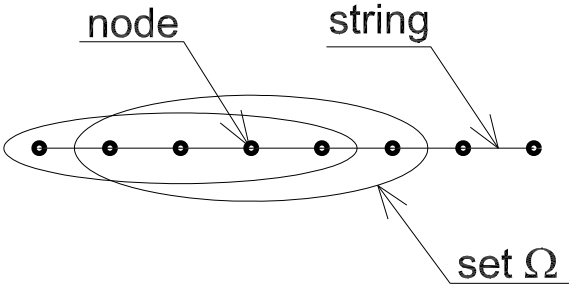

Fig. 9.1 The set Ω moving along a string.

C.I. Bajer and B. Dyniewicz: Numerical Analysis of Vibrations of Structures, LNACM 65, pp. 241–246.
springerlink.com

where $W(x-x_i)$ is the weight function, $w^h(x,x_i)$ is an approximation function, and the w_i are the nodal values. We assume exponential shape functions:

$$W(x-x_i) = \begin{cases} \mathrm{e}^{-\left(\frac{x-x_i}{\alpha}\right)^2} & \text{if } (x-x_i) \le 1, \\ 0 & \text{if } (x-x_i) > 1 \ . \end{cases} \tag{9.2}$$

The coefficient α depends on the size of the domain Ω and the number of points in the domain.

In this case, the exact solution has been approximated by an nth degree polynomial

$$w^h(x,x_i) = \sum_{j=1}^{n} p_j(x_i)a_j(x) = \mathbf{p}^{\mathsf{T}}(x_i)\mathbf{a}(x) \ , \tag{9.3}$$

where the monomials in the interpolation polynomial are

$$\mathbf{p}^{\mathsf{T}} = \left[1, x, x^2, \ldots\right] \ , \tag{9.4}$$

with approximation coefficients

$$\mathbf{a}^{\mathsf{T}}(x) = \left[a_0(x), a_1(x), a_2(x), \ldots\right] \ . \tag{9.5}$$

The functional (9.1) can be written in matrix form:

$$J = (\mathbf{P}\mathbf{a} - \mathbf{w})^{\mathsf{T}} \mathbf{W} (\mathbf{P}\mathbf{a} - \mathbf{w}) \ , \tag{9.6}$$

where $\mathbf{P}$ is a full matrix of dimension $(n \times m)$

$$\mathbf{P} = \begin{pmatrix} p_0(x_1) & p_1(x_1) & \cdots & p_m(x_1) \\ p_0(x_2) & p_1(x_2) & \cdots & p_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_0(x_n) & p_1(x_n) & \cdots & p_m(x_n) \end{pmatrix} \ , \tag{9.7}$$

and $\mathbf{W}$ is a diagonal matrix of dimension $(n \times n)$

$$\mathbf{W} = \begin{pmatrix} W(x-x_1) & 0 & \cdots & 0 \\ 0 & W_2(x-x_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W(x-x_n) \end{pmatrix} \ . \tag{9.8}$$

As a result of minimizing the functional (9.6) at the coefficients of the approximating polynomial ($\partial J / \partial a_i = 0$) we obtain the sought form of the vector $\mathbf{a}$ from equation (9.3)

$$\mathbf{a}(x) = \mathbf{A}^{-1}\mathbf{B}\,\mathbf{w} \ , \tag{9.9}$$

where $\mathbf{A} = \mathbf{P}^{\mathsf{T}}\mathbf{W}\mathbf{P}$ and $\mathbf{B} = \mathbf{P}^{\mathsf{T}}\mathbf{W}$. Eqs (9.3) and (9.9) lead to:

$$w^h(x,x_i) = \mathbf{p}^{\mathsf{T}}(x_i)\,\mathbf{A}^{-1}\,\mathbf{B}\,\mathbf{w} = \sum_{j=1}^{n} \phi_j^m(x)\; w_j \;, \tag{9.10}$$

where $\phi_j^m(x)$ is the shape function and m is the degree of the approximation polynomial

$$\phi_j^m(x) = \mathbf{p}^{\mathsf{T}}(x_i)\,\mathbf{A}^{-1}\,\mathbf{B} = [\phi_1^m(x), \phi_2^m(x), \ldots \phi_n^m(x)] \;. \tag{9.11}$$

The general criterion of MLS approximation requires differentiation of the equation $\mathbf{p}^{\mathsf{T}}(x_i)\mathbf{A}^{-1}\mathbf{B}$. In order to simplify this task, we apply a zero degree polynomial approximation ($m = 0$). This leads to the shape functions called the Shepard functions, dependent only on a weight function of the nodes

$$\phi_i^0 \;=\; \frac{W(x-x_i)}{\sum_{j=1}^{n} W(x-x_j)} \;. \tag{9.12}$$

Selecting these functions is a significant problem when the nodes are irregularly distributed. In the general case, the designation of the matrix describing the test structure requires numerical integration, resulting in additional computational costs.

The resulting shape function (9.12) was used for discretization of a string. Using virtual work, we can obtain the stiffness and inertia matrices:

$$\begin{aligned} k_{ij} \;&=\; \int_{\Omega} B_i^T\, N B_j \; d\Omega \;, \\ m_{ij} \;&=\; \int_{\Omega} \phi_i^T\, \rho A\, \phi_j \; d\Omega \;, \end{aligned} \tag{9.13}$$

where $B = \partial\phi/\partial x$, N is the tensile force, and ρA is the mass of the string. The nodal matrices $\mathbf{k}$ and $\mathbf{m}$ computed for domains Ω are assembled into global matrices $\mathbf{K}$ and $\mathbf{M}$. The proper choice of the parameter α is still the fundamental problem of the weight function (9.2). In the case of a regular distribution of points in the set Ω, where the distance between the nodes is denoted by b, the coefficient α can described with high accuracy by $\alpha = b/\sqrt{\pi}$.

9.2 Results

The numerical results will be compared with the analytical results. We used the following data: length of the string $l = 1$, tensile force $N = 1$, mass of the a string $\rho A = 1$, external force $P = -1$ travelling with constant speed $\mathsf{v} = 0.1c$ (where c, as usual, is the wave speed in the string).

In Figure 9.2 we plot the displacements under a moving force and at the middle of the string length. The analytical solution coincides well with the one obtained by the finite element method or the central difference method. The meshless method results in a good approximation to the accurate solution. In this method we can notice the phase error, visible especially in the second half of the observation period. This means that the rigidity is higher than required. However, the smoothing effect is the main reason for the differences in the results. A significantly lower number of

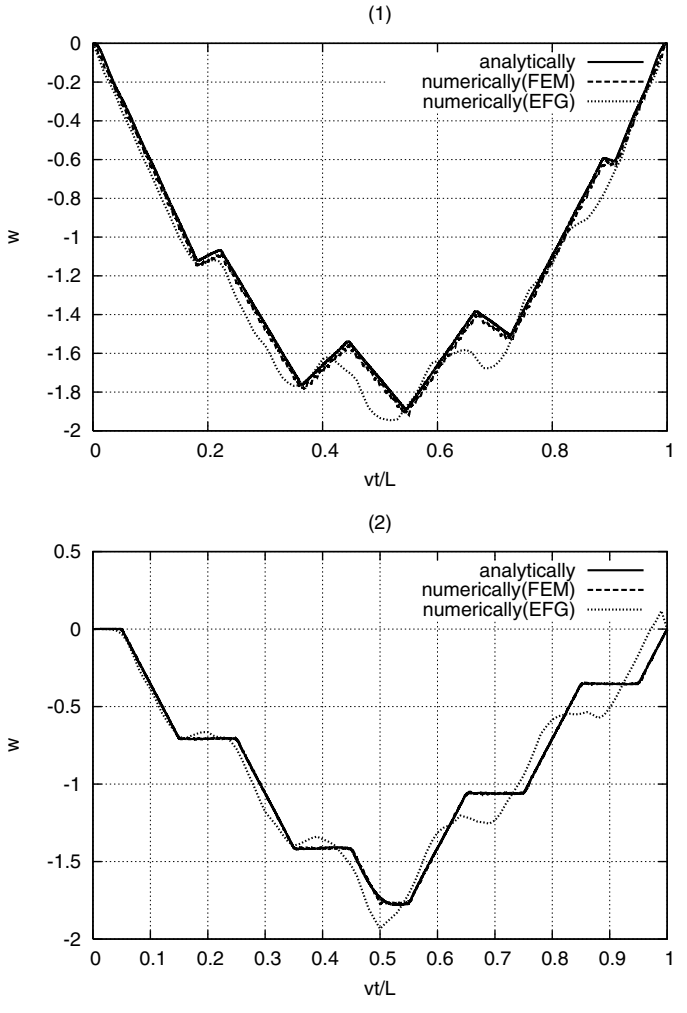

Fig. 9.2 Displacements under constant moving force: (1) under moving force, (2) at the middle of the string (number of points in Ω is 10).

points in the moving domain (2 instead of 10 as in the previous case) (Figure 9.3) results in quite similar plots. Both the amplitude and the period of the vibrations are acceptable.

The numerical approach in the case of a Winkler foundation does not differ from the analytical solution. We must emphasize here that a continuously moving force in the numerical application is replaced by a sequence of marching pairs of forces, applied to nodal points. In such a case we neglect the mixed derivatives of the formulation. We can notice that such an analysis does not introduce a significant error (Figure 9.4).

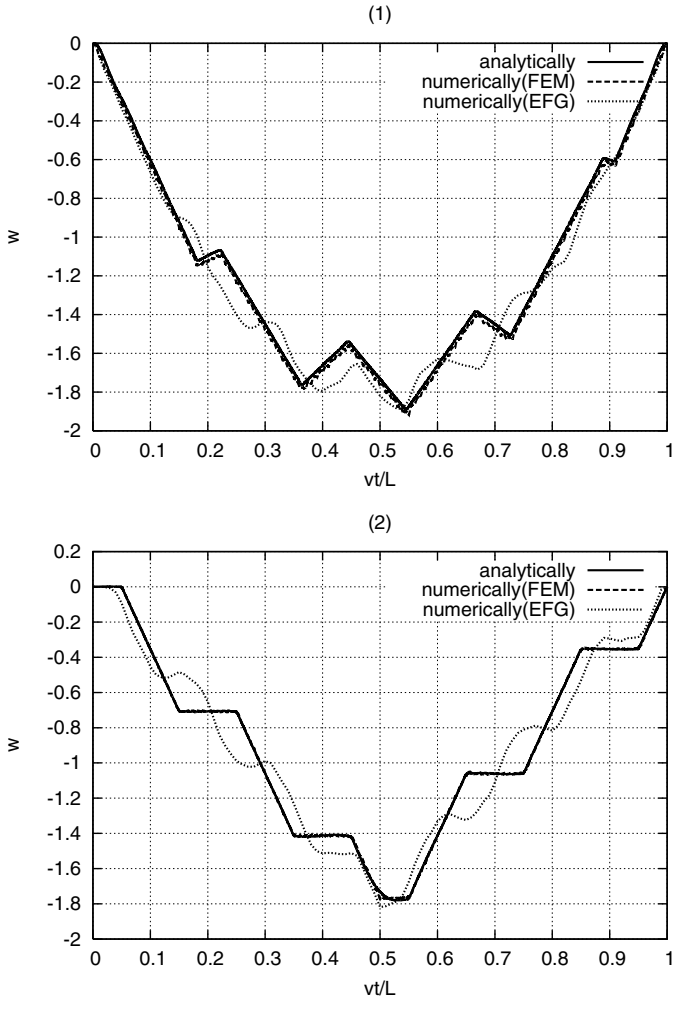

Fig. 9.3 Displacements under constant moving force: (1) under moving force, (2) at the middle of the string (number of points in Ω is 2).

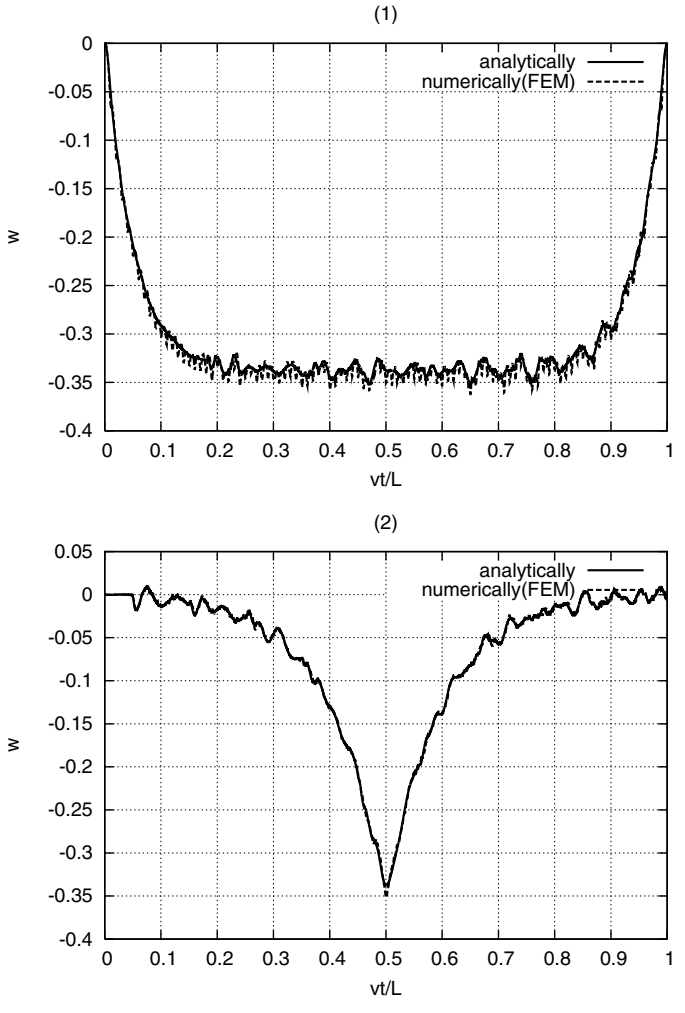

Fig. 9.4 Displacements under constant moving force—string put on a Winkler foundation: (1) under moving force, (2) at the middle of the string.