

# Using generalized functions in continuum mechanics

Petr Procházka  
Ústav matematiky



## OBJECTIVES

- Create a mathematical model of a beam vibrations. The beam has  $n$ -flexible supports. Solution is assumed on the generalized functions principle.
- Create software for the solution of the own and forced beam vibrations.
- Consider the use of generalized functions for hydrodynamic purposes.

## GENERAL MODEL FOR BEAM

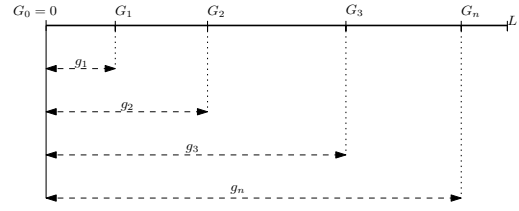


FIGURE: Generally loaded beam with point forces/moments/springs

$$y'''' + \frac{\rho A}{EI} \ddot{y} + \frac{k_e}{EI} \dot{y} + \frac{c_e}{EI} y = \sum_{i=0}^k \frac{F_i(t) \delta(x - g_i)}{EI} + \sum_{i=0}^k \frac{M_i(t) \delta'(x - g_i)}{EI} - \sum_{i=0}^k \frac{k_{pi} y(g_i, t) \delta(x - g_i)}{EI} + \sum_{i=0}^k \frac{k_{oi} y'(g_i, t) \delta'(x - g_i)}{EI} \quad (1)$$

## GENERAL MODEL FOR BEAM

Applying double *Laplace transform* to the system of equations (1) produces unknown coefficients, which can be written as an vector

$$\hat{\mathbf{Y}} = \left\{ \begin{bmatrix} y(0, t) \\ y'(0, t) \\ y(g_1, t) \\ y'(g_1, t) \\ \vdots \\ y(g_n, t) \\ y'(g_n, t) \end{bmatrix} \right\}_{t \rightarrow s} \quad 2(k+1)$$

Rewriting the system of equations (1) into the matrix form yields

$$\mathbf{Z} \cdot \hat{\mathbf{Y}} = \mathbf{J}$$

## MATRIX Z

$$\mathbf{Z} = \begin{bmatrix} A_1 & A_1^* & -1 & 0 & 0 & 0 & 0 & 0 & \dots \\ \hat{A}_1 & \hat{A}_1^* & 0 & -1 & 0 & 0 & 0 & 0 & \dots \\ A_2 & A_2^* & B_{2,1} & B_{2,1}^* & -1 & 0 & 0 & 0 & \dots \\ \hat{A}_2 & \hat{A}_2^* & \hat{B}_{2,1} & \hat{B}_{2,1}^* & 0 & -1 & 0 & 0 & \dots \\ A_3 & A_3^* & B_{3,1} & B_{3,1}^* & B_{3,2} & B_{3,2}^* & -1 & 0 & \dots \\ \hat{A}_3 & \hat{A}_3^* & \hat{B}_{3,1} & \hat{B}_{3,1}^* & \hat{B}_{3,2} & \hat{B}_{3,2}^* & 0 & -1 & \dots \\ A_4 & A_4^* & B_{4,1} & B_{4,1}^* & B_{4,2} & B_{4,2}^* & B_{4,3} & B_{4,3}^* & \dots \\ \hat{A}_4 & \hat{A}_4^* & \hat{B}_{4,1} & \hat{B}_{4,1}^* & \hat{B}_{4,2} & \hat{B}_{4,2}^* & \hat{B}_{4,3} & \hat{B}_{4,3}^* & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ A_k & A_k^* & B_{k,1} & B_{k,1}^* & B_{k,2} & B_{k,2}^* & B_{k,3} & B_{k,3}^* & \dots \\ \hat{A}_k & \hat{A}_k^* & \hat{B}_{k,1} & \hat{B}_{k,1}^* & \hat{B}_{k,2} & \hat{B}_{k,2}^* & \hat{B}_{k,3} & \hat{B}_{k,3}^* & \dots \\ \hat{C} & \hat{C}^* & C_1 & C_1^* & C_2 & C_2^* & C_3 & C_3^* & \dots \\ \hat{D} & \hat{D}^* & D_1 & D_1^* & D_2 & D_2^* & D_3 & D_3^* & \dots \end{bmatrix}$$

## MATRIX Z-TERMS

$$\begin{aligned} A_i &= S(\lambda g_i) - \frac{V(\lambda g_i) k_{p0}}{\lambda^3 EI} & B_{i,j} &= -\frac{V(\lambda(g_i - g_j)) k_{pj}}{\lambda^3 EI} \\ \hat{A}_i &= \lambda V(\lambda g_i) - \frac{U(\lambda g_i) k_{p0}}{\lambda^2 EI} & \hat{B}_{i,j} &= -\frac{U(\lambda(g_i - g_j)) k_{pj}}{\lambda^2 EI} \\ A_i^* &= \frac{T(\lambda g_i)}{\lambda} + \frac{U(\lambda g_i) k_{o0}}{\lambda^2 EI} & B_{i,j}^* &= \frac{U(\lambda(g_i - g_j)) k_{oj}}{\lambda^2 EI} \\ \hat{A}_i^* &= S(\lambda g_i) + \frac{T(\lambda g_i) k_{o0}}{\lambda EI} & \hat{B}_{i,j}^* &= \frac{T(\lambda(g_i - g_j)) k_{oj}}{\lambda EI} \\ \hat{C} &= \lambda^2 U(\lambda L) - \frac{T(\lambda L) k_{p0}}{\lambda EI} & \hat{C}^* &= \lambda V(\lambda L) + S(\lambda L) \frac{k_{o0}}{EI} \\ C_i &= -\frac{T(\lambda(L - g_i)) k_{pi}}{\lambda EI} & C_i^* &= S(\lambda(L - g_i)) \frac{k_{oi}}{EI} \\ \hat{D} &= \lambda^3 T(\lambda L) - S(\lambda L) \frac{k_{p0}}{EI} & \hat{D}^* &= \lambda^2 U(\lambda L) + \lambda V(\lambda L) \frac{k_{o0}}{EI} \\ D_i &= -S(\lambda(L - g_i)) \frac{k_{pi}}{EI} & D_i^* &= \lambda V(\lambda(L - g_i)) \frac{k_{oi}}{EI} \end{aligned}$$

where  $S, T, U, V$  are Rayleigh functions. The matrix  $\mathbf{J}$  is given in the similar way.

## SOFTWARE

Enter Length L [m]	2.5
Enter Young's modulus E [Pa]	205000000000
Enter density rho [kg/m^3]	7830
Enter time t [s]	1
Enter coefficient of external damping [-]	0
Enter coefficient of flexible subsoil [-]	0
Enter width [m]	0.3
Enter height [m]	0.05
Enter Position [m]	0
Enter Force F [N]	0
F Omega [rad s^-1]	0
Enter Moment M [Nm]	0
M Omega [rad s^-1]	0
Enter Stiffness K_pi [Nm^-1]	0
Enter Stiffness K_oi [Nm]	0
Enter 1. values	Finish entry exit

FIGURE: Software programmed in Python for solving beam problems, based on the general model.

## SIMPLE PIPE

Continuity equation

$$\frac{S}{\rho a^2} \frac{\partial p}{\partial t} + \frac{\partial Q}{\partial x} = 0$$

Equilibrium fluid flow equation

$$\frac{\rho}{S} \frac{\partial Q}{\partial t} + \frac{\partial p}{\partial x} + \frac{b}{S} Q = 0$$

Denoting  $\mathbf{w}(x, t) = [p(x, t) \quad Q(x, t)]^T$  applies

- Initial conditions

$$\mathbf{w}(x, 0) = [p(x, 0) \quad Q(x, 0)]^T = [p_0 \quad Q_0]^T, \forall x \in \langle 0, L \rangle$$

- Boundary conditions

$$\mathbf{w}(0, t) = \mathbf{a}_1 \alpha + \mathbf{m}(t), \forall t \geq 0$$

$$\mathbf{b}_1^T \mathbf{w}(L, t) = n(t), \forall t \geq 0$$

## DISTRIBUTIONS APPROACH IN HYDRODYNAMIC

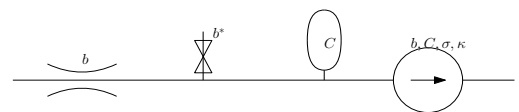


FIGURE: Various hydraulic elements in the pipeline system applied at the position  $\zeta$ , described by the coefficients  $b, b^*, C, \sigma, \kappa$

Pipes equations rewritten using distributions

$$\begin{aligned} \frac{\partial p(x, t)}{\partial x} + \frac{\rho}{S} \frac{\partial Q(x, t)}{\partial t} - \sigma p(\zeta, t) \delta(x - \zeta) + b Q(\zeta, t) \delta(x - \zeta) &= 0 \\ \frac{\partial Q(x, t)}{\partial x} + \frac{S}{\rho a^2} \frac{\partial p(x, t)}{\partial t} + C \frac{\partial p(\zeta, t)}{\partial t} \delta(x - \zeta) + \kappa \frac{\partial Q(\zeta, t)}{\partial t} \delta(x - \zeta) &= 0 \end{aligned}$$