Code Overview

for Periodic Cable Dynamics Solver

S. Boral, J. Hickey and B. Basu

Overview

This supplementary document provides **non-executable pseudocode** describing the numerical procedure used in the article:

> *"Modeling and Simulation of Periodic Cable Dynamics with Bending Stiffness Using Time-Domain and Multi-Harmonic Balance Methods"*

The MATLAB implementation used in this study forms part of a larger, proprietary research codebase that is actively being developed for multiple future publications. For this reason, the full source code cannot be released publicly in its current form.

To ensure transparency and reproducibility, we provide in the Electronic Supplementary Material a complete and self-contained pseudocode description of all computational steps, including the static configuration solver, the time-domain integrator, the force models, and the reconstruction procedure. This pseudocode fully specifies the numerical workflow and is sufficient for independent re-implementation of the methods presented in the paper.

While the full MATLAB source code cannot be made publicly available, it can be shared with bona fide researchers upon reasonable request for academic and non-commercial use, subject to the constraints of the larger codebase and ongoing research commitments.

Model Parameters

Table 1 summarizes the physical and numerical parameters used in the model.

Symbol	Description	Typical Value
N	Number of nodes along cable	51
L	Cable unstretched length	902.2 m
EA	Axial stiffness	3.84243×10^7 N
EI	Bending stiffness	(given in paper)
W	Submerged weight per unit length	698.094 N/m
ρ_w	Water density	1025 kg/m^3
d	Cable diameter	0.09 m
C_d^t	Tangential drag coefficient	1.6
C_d^n	Normal drag coefficient	1.6
m	Structural mass per unit length	77.7066 kg/m
C_a	Added mass coefficient	1.0
ω	Forcing frequency	$2\pi f$
A_x, A_z	Fairlead motion amplitudes	(given in paper)
U_c, V_c	Background current components	(given in paper)

Table 1: Model parameters used in the pseudocode description.

Algorithm S1: Static Configuration of the Cable

Algorithm 1 Static configuration (non-executable pseudocode)

1: Discretize unstretched arc length:

$$s_i = \frac{i-1}{N-1}L, \quad i = 1, \ldots, N.$$

2: Initialize arrays:

- $\phi_{\text{static}}[i]$ (initial angle guess, e.g. catenary)
- $\varepsilon_{\text{static}}[i]$ (initial axial strain guess)

3: For each interior node $i = 2, \ldots, N-1$, compute:

- curvature and its derivatives from finite differences of ϕ_{static}
- $T[i] = EA\,\varepsilon_{\text{static}}[i]$
- dT/ds via central differences

4: Form static equilibrium residuals:

$$R_1[i] = B[i] + \frac{dT}{ds} - W\sin(\phi_{\text{static}}[i]),$$

$$R_2[i] = -EI\frac{d^3\phi}{ds^3} + T[i]\kappa[i] - W\cos(\phi_{\text{static}}[i]).$$

5: Impose boundary conditions:

$$(x, z)(0) = (0, 0), \qquad (x, z)(L) = (x_{\text{end}}, z_{\text{end}})$$

using geometric reconstruction from $(1 + \varepsilon_{\text{static}})\{\cos\phi_{\text{static}}, \sin\phi_{\text{static}}\}$.

6: Solve the nonlinear residual system for ϕ_{static} and $\varepsilon_{\text{static}}$ using an iterative method (e.g. Newton-type).

Algorithm S2: Time-Domain Dynamic Solver

Algorithm 2 Dynamic simulation (non-executable pseudocode)

Define state variables at node i:

$$u[i],\ v[i],\ e[i],\ \phi[i].$$

All initialized to zero at $t = 0$.

At each time t:

1. Compute fairlead velocity:

$$\dot{X}_f(t) = A_x\omega(-\sin\omega t), \qquad \dot{Z}_f(t) = A_z\omega(\cos\omega t).$$

2. Project $(\dot{X}_f, \dot{Z}_f)$ onto cable tangent/normal at node N:

$$u[N],\ v[N].$$

3. Enforce anchored-end condition:

$$u[1] = 0, \qquad v[1] = 0.$$

For each interior node $i = 2, \ldots, N-1$:

1. Compute derivatives du/ds, dv/ds, de/ds, $d\phi/ds$ using finite differences.

2. Compute relative fluid velocities:

$$u_{\text{rel}} = u - (U_c \cos\phi_{\text{static}} + V_c \sin\phi_{\text{static}}),$$

$$v_{\text{rel}} = v - (-U_c \sin\phi_{\text{static}} + V_c \cos\phi_{\text{static}}).$$

3. Drag forces:

$$F_t = -\tfrac{1}{2}\rho_w d C_d^t |u_{\text{rel}}| u_{\text{rel}} \sqrt{1+e},$$
$$F_n = -\tfrac{1}{2}\rho_w d C_d^n |v_{\text{rel}}| v_{\text{rel}} \sqrt{1+e}.$$

4. Evaluate semi-discrete equations:

$$\frac{du}{dt} = \frac{1}{m}\left[\text{bending} + EA\,\frac{de}{ds} - W\phi\cos\phi_{\text{static}} + F_t\right],$$

$$\frac{dv}{dt} = \frac{1}{m+m_a}\left[\text{bending} + T_{\text{static}}\frac{d\phi}{ds} + EA\,e\,\frac{d\phi_{\text{static}}}{ds} + W\phi\sin\phi_{\text{static}} + F_n\right],$$

$$\frac{de}{dt} = \frac{du}{ds} - v\frac{d\phi_{\text{static}}}{ds},$$
$$\frac{d\phi}{dt} = \frac{dv}{ds} + u\frac{d\phi_{\text{static}}}{ds}.$$

Advance the state vector using a time-integration scheme such as explicit Runge–Kutta.

Algorithm S3: Reconstruction and Outputs

Algorithm 3 Cable shape and fairlead tension (non-executable pseudocode)

1: At time t_k, compute total angle and strain:

$$\phi_{\text{tot}}[i] = \phi_{\text{static}}[i] + \phi[i](t_k), \qquad \varepsilon_{\text{tot}}[i] = \varepsilon_{\text{static}}[i] + e[i](t_k).$$

2: Reconstruct geometry:

$$x[i+1] = x[i] + ds\,(1 + \varepsilon_{\text{tot}}[i])\cos\phi_{\text{tot}}[i],$$

$$z[i+1] = z[i] + ds\,(1 + \varepsilon_{\text{tot}}[i])\sin\phi_{\text{tot}}[i].$$

3: Compute fairlead tension:

$$T_f(t_k) = T_{\text{static}}[N] + EA\,e[N](t_k).$$

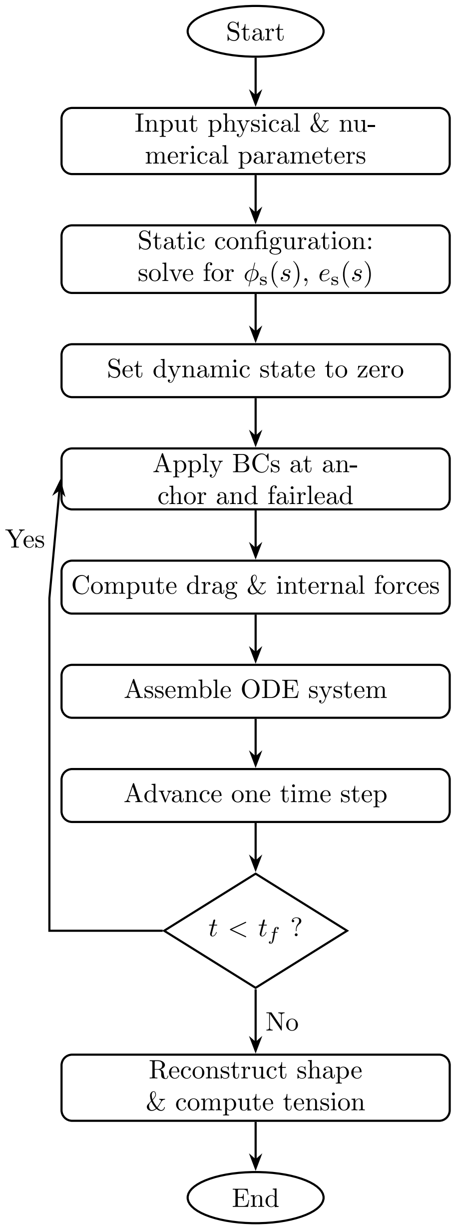

Figure 1: Compact flow diagram of the simulation procedure.