Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXIX, págs. 845-852 (artículo completo)
F.E. Dotti, M. Febbo, S.P. Machado, M. Saravia, M.A. Storti (Eds.)
Bahía Blanca, 1-4 Noviembre 2022

COUPLING COMPUTATIONAL STRUCTURAL AND CROWD DYNAMICS

Rainald Löhner[a], Daniel Andres[b] and Roland Wüchner[b]

[a]*Center for Computational Fluid Dynamics, George Mason University, Fairfax, VA 22030-4444, USA, rlohner@gmu.edu, http://www.cfd.science.gmu.edu*

[b]*Institute of Structural Analysis, TU Braunschweig, Braunschweig, Germany, daniel.andres@tu-braunschweig.de,r.wuechner@tu-braunschweig.de, https://www.tu-braunschweig.de*

Keywords: Computational Structural Dynamics, Computational Crowd Dynamics, Pedestrian Loads, Vibration.

Abstract. A loosely coupled algorithm is presented that allows to combine Computational Structural Dynamics (CSD) and Computational Crowd Dynamics (CCD) codes in order to solve, in a cost-effective manner, pedestrian-structure interaction problems. The basic structural and crowd dynamics codes are altered as little as possible. The surfaces of the structure where pedestrians transit are used to transfer the required information between codes. The transfer of loads, displacements, and velocities is carried out via fast interpolation and projection algorithms.
The loosely coupled algorithm can also be modified in order to obtain a fully coupled algorithm via iterative loops. Experience gained from several calculations indicates that the proposed approach offers a convenient and cost-effective way of coupling CSD and CCD codes without a complete re-write of them.

1 INTRODUCTION

Many pedestrian bridges or passageways are being built attempting to combine a visually appealing structure with minimum material/weight requirements. The result does not always result in a safe structure: the Millennium Bridge in London (5) provides a good reminder that one must still account for possible vibrations and resonances that can render such structures unsafe for operation (1; 2). At the same time, due to aging, environmental impact or accidents the structural integrity of these structures can deteriorate over time. Given the drastic miniaturization, connectivity, standardization, price reduction and proliferation of in-situ measuring devices, monitoring the structural integrity throughout the lifetime of such structures has become a reality. This implies the need to answer the question: where should these sensors be placed in order to detect at the earliest possible stage the maximum number of different deterioration mechanisms of the bridges/structures ?
The need to predict (during design) and guarantee (during its lifetime) the safe operation of pedestrian bridges and passageways has in turn led to the need to couple computational structural dynamics codes with computational crowd dynamics codes.
Computational structural dynamics (CSD) codes have a long history (since the 1960's) of development and application in engineering practice. In particular for the linear elastic regime in which bridges operate, finite element methods are well established (6) and used on a daily basis. Computational crowd dynamics (CCD) codes have been developed since the early 2000's, and have been used on industrial projects during the last decade (4). Unlike elastic materials, humans are more complex: they usually have a series of targets they want to reach at a certain time, can think, react, have different levels of emotions, will force and cultural backgrounds. It is therefore not surprising that the development and application of computational crowd dynamics codes took longer to develop.

2 COUPLING ALGORITHM

The so-called loose coupling technique is used in order to link the CSD and CCD codes. The technique is shown in Figure 1.

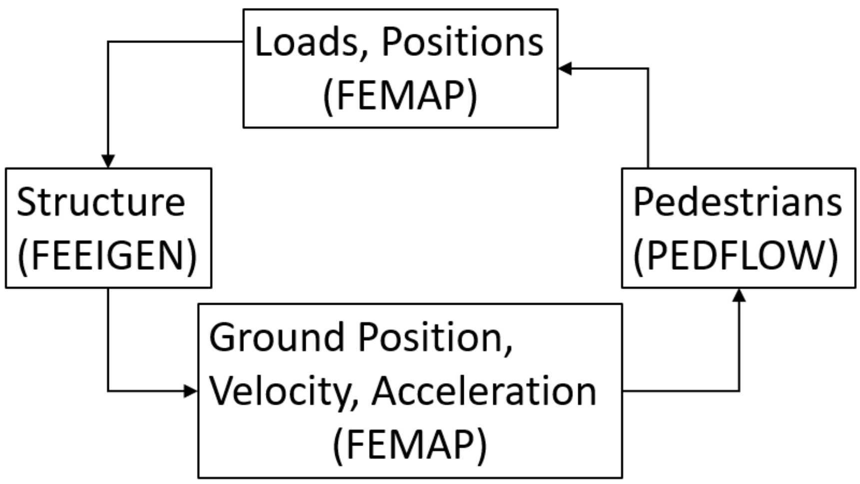

Figure 1 Loose Coupling of CSD and CCD Codes

The global timestepping algorithm proceeds as follows:

```
Initialization:
       Set:  istar=1,istop=0
       call CSD-code(.., istar, ..)
         read in CSD-Data
         initialize all CSD Arrays
       call CCD-code(.., istar, ..)
         read in CCD-Data
         initialize all CCD Arrays
Run:
       do:  Loop over the timesteps
         do:  Until convergence
            call CSD-code(.., istop, tends, tendc, ..)
            call CCD-code(.., istop, tends, tendc, ..)
         enddo
       enddo
```

Here `tends, tendc` denote the ending times for the CSD and CCD code respectively. The algorithm outlined above clearly leaves the possibility open to perform N-CSD-code steps, followed by M-CCD-code steps, i.e. asynchronous timestepping. It is felt that this is of considerable importance in order to keep the algorithm as general as possible. As one can see, both the CSD and the CCD codes are called as subroutines. The argument list passed contains all the variables required for the inter-grid transfer of information.

3 CSD CODE: FEEIGEN

Given a Finite Element discretization of an elastic structure, the resulting system of equations will be of the form (6):

$$\mathbf{M}\ddot{\mathbf{w}} + \mathbf{D}\dot{\mathbf{w}} + \mathbf{K}\mathbf{w} = \mathbf{f} \quad . \tag{1}$$

Here $\mathbf{M}, \mathbf{D}, \mathbf{K}$ denote the mass, damping and stiffness matrices respectively, and $\mathbf{w}$ is the vector of nodal displacement variables. The matrices $\mathbf{M}, \mathbf{K}$ are symmetric positive definite, and can be used to obtain a system of eigenmodes by solving the eigenvalue problem:

$$\left(-\omega_i^2\mathbf{M} + \mathbf{K}\right) \cdot \mathbf{e}^i = 0 \ , \ \ i = 1, n \ . \tag{2}$$

The eigenvectors satisfy the following important orthogonality properties:

$$\mathbf{e}^j\mathbf{M}\mathbf{e}^i = \delta^{ij} \ \ ; \ \ \mathbf{e}^j\mathbf{K}\mathbf{e}^i = \omega_i^2\delta^{ij} \quad , \tag{3}$$

where δ^{ij} is the Kronecker-δ. The vector of unknowns $\mathbf{w}$ can now be written in terms of these eigenvectors as

$$\mathbf{w} = \mathbf{e}^i a_i \quad , \tag{4}$$

resulting in

$$\mathbf{M}\mathbf{e}^i\ddot{a}_i + \mathbf{D}\mathbf{e}^i\dot{a}_i + \mathbf{K}\mathbf{e}^i a_i = \mathbf{f} \quad . \tag{5}$$

Assuming $\mathbf{e}^j \mathbf{D} \mathbf{e}^i = d^{ii}\delta^{ij}$, one can decompose the former equation by multiplication with $\mathbf{e}^j$. This results in:

$$\ddot{a}_j + d^{jj}\dot{a}_j + \omega_j^2 a_j = \mathbf{f} \cdot \mathbf{e}^j = f^j \quad , \tag{6}$$

i.e. a **decoupled system of ordinary differential equations**. FEEIGEN is a simple ODE integrator for eigenmodes. A variety of integration schemes are available, from explicit high-order Runge-Kutta to second-order, implicit Newmark schemes. FEEIGEN obtains the forces from the CCD code, projects these into the eigenmodes, updates the solution, and computes the updated shape and velocity of the structure. This information is then passed to the CCD code.

4 CCD CODE: PEDFLOW

The PEDFLOW model (4) is a combination of social force and agent based models. Individuals move according to Newton's laws of motion; they follow (via will forces) 'global movement targets'; at the local movement level, the motion also considers the presence of other individuals or obstacles via avoidance forces (also a type of will force) and, if applicable, contact forces. Newton's laws:

$$m\frac{d\mathbf{v}}{dt} = \mathbf{f} \;\; , \;\; \frac{d\mathbf{x}}{dt} = \mathbf{v} \;\; , \tag{7}$$

where $m, \mathbf{v}, \mathbf{x}, \mathbf{f}, t$ denote, respectively, mass, velocity, position, force and time, are integrated in time using a 2nd order explicit timestepping technique. The main modeling effort is centered on $\mathbf{f}$. PEDFLOW separates these forces into internal (or will) forces [I would like to move here or there] and external forces [I collided with another pedestrian or an obstacle]. For more information, as well as verification and validation studies, see (4; 3).
PEDFLOW obtains the position and velocity of the terrain from the CSD code, obtains the reaction forces of the pedestrians, and then passes this information to the CSD code in turn.

4.1 Pedestrian Loads on the Structure

The proper understanding and the modelling of the transient forces/loads of the pedestrians is of paramount importance (2). The vertical force can be decomposed into a static part corresponding to the weight of the person and a dynamic part as the sum of harmonic functions ("harmonics") with frequencies an integer multiple of the pacing or the jumping frequency, respectively, which is the fundamental frequency of the person's action (so called Fourier decomposition):

$$F_p(t) = G + \Delta G_1 \cdot \sin(2\pi f_p t) + \Delta G_2 \cdot \sin(4\pi f_p t - \phi_2) + \Delta G_3 \cdot \sin(6\pi f_p t - \phi_3) \;\; , \tag{8}$$

where G denotes the weight of the person (usually of the order of 800 N), $\Delta G_{1,2,3}$ the amplitudes of the first three harmonics, f_p the pacing or jumping frequency, and $\phi_{2,3}$ the phase shifts of the 2nd and 3rd harmonic with respect to the 1st harmonic. While higher harmonics could be considered, experience indicates that three are sufficient for an accurate description of the forces (2).

5 AN EXAMPLE

A simply supported bridge of length 20 m and width 5 m subjected to pedestrian loads is considered in this first example. Two snapshots during the run are shown in Figures 2a,b. For the structure, only 1-D modes (beam-modes along the length of the bridge) are considered, i.e. torsion modes are neglected. This is clearly visible in the velocity distribution shown in Figure 2b. Pedestrians enter the bridge from both directions with a flux of 1 ped/sec. It takes about 25 seconds for the bridge to become populated and the structure to react to the loads. The run was continued to 4 minutes in order to obtain proper statistics. With the given flux of pedestrians, an average of about 50 pedestrians are on the bridge at any given time (Figure 2c). The displacement and velocity history of three points in the centerline of the the bridge along the x-direction (denoted by their x-position: $x = 5$ implying 1/4 along the bridge, $x = 10$ in the middle of the bridge, and $x = 15$ implying 3/4 along the bridge) are shown in Figure 2d,e. Given that FEEIGEN selects the closest point in the mesh to the desired position of these diagnostics points, the actual position may be close but not exactly the one specified by the user. This explains the slight discrepancy in the extreme values between the points denoted by $x = 5$ and $x = 15$ (which should be similar). The resulting eigenforces and eigenvelocities (f^j and $\dot{a}_j$ in Eqn.(6) above) are displayed in Figure 2f,g. One can see that, as expected, the dominant mode is the main bending mode. Note also that due to the random motion of the pedestrians no resonance or cyclic force, displacement or velocity pattern is discernable.

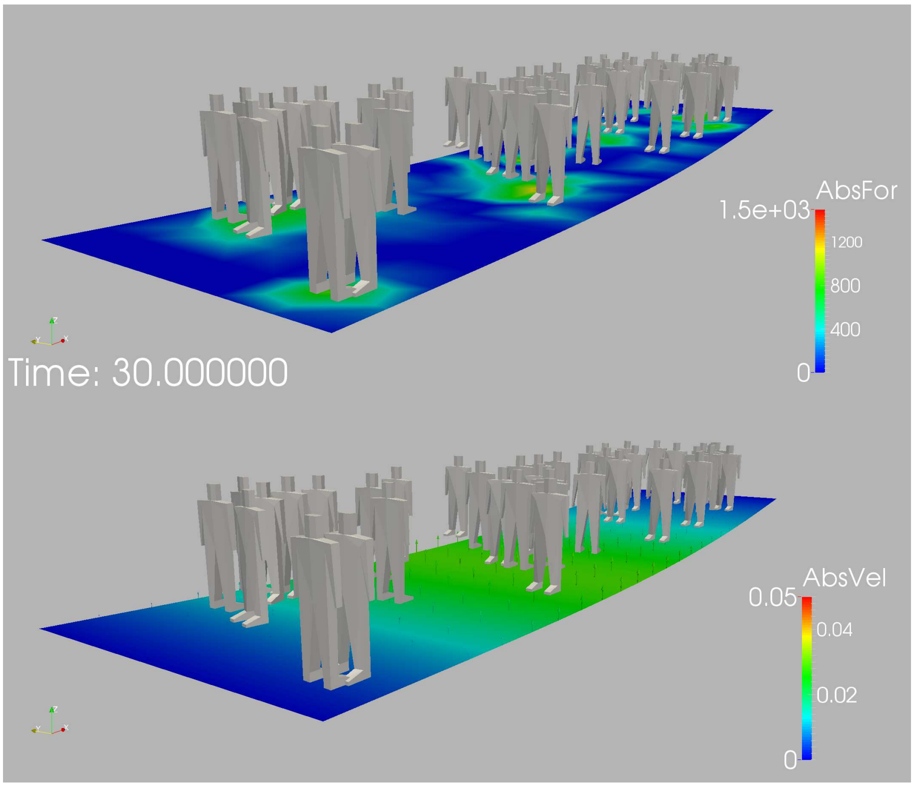

Figure 2a Force and Velocities of the Structure at T=30 sec

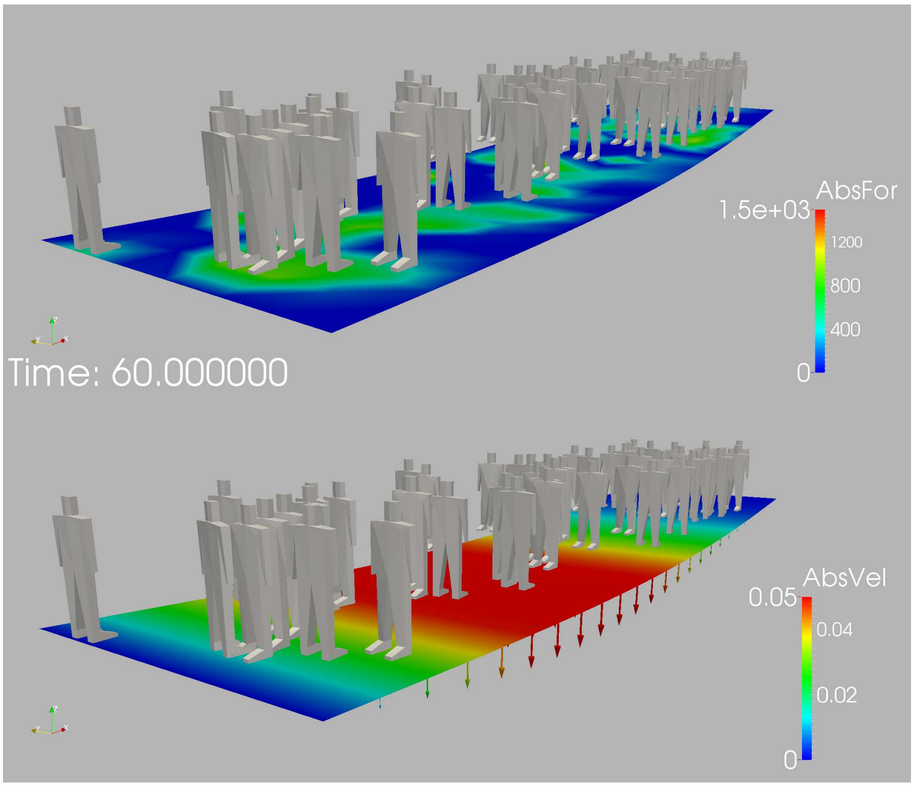

Figure 2b Force and Velocities of the Structure at T=60 sec

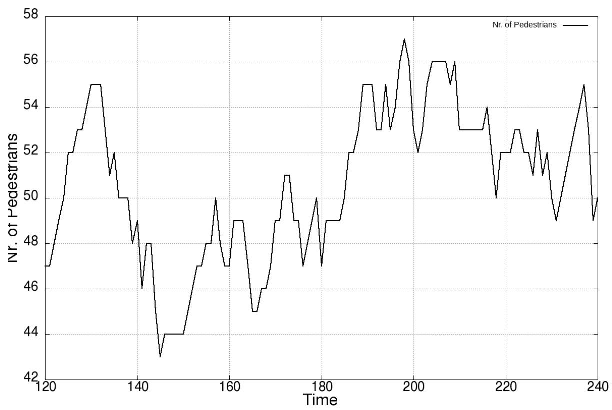

Figure 2c Number of Pedestrians on the Bridge

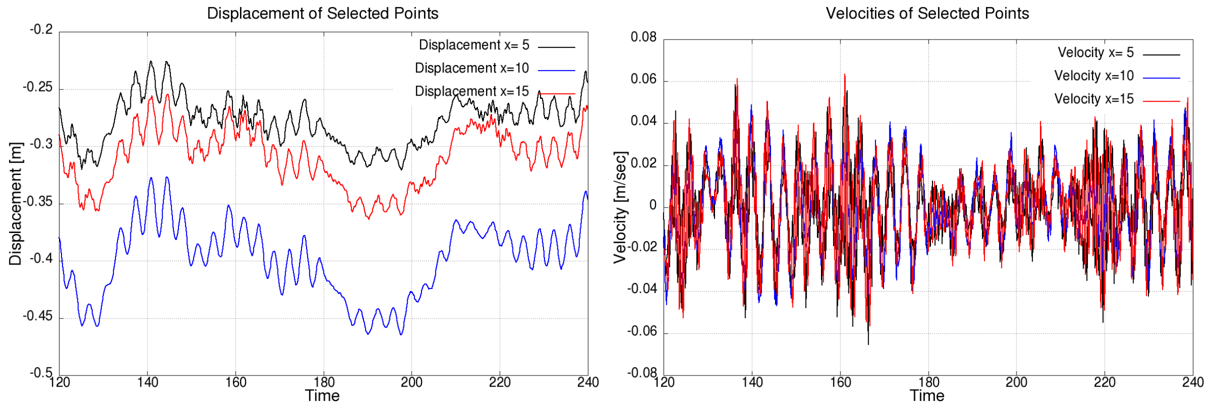

Figure 2d,e Displacements and Velocities of Three Selected Points on the Bridge

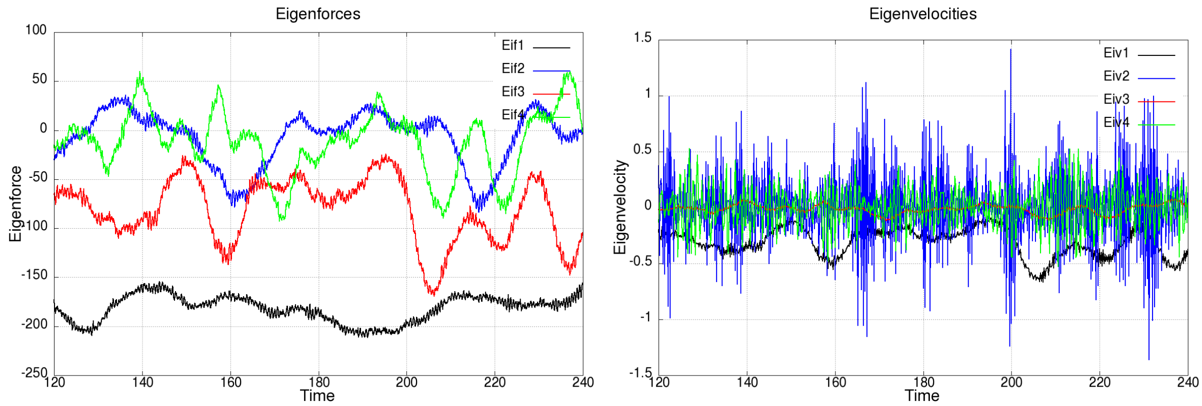

Figure 2f,g Eigenforces and eigenvelocities

6 CONCLUSIONS

A loosely coupled algorithm that allows to combine Computational Structural Dynamics (CSD) and Computational Crowd Dynamics (CCD) codes in order to solve, in a cost-effective manner, pedestrian-structure interaction problems has been developed and first results have been shown. The basic structural and crowd dynamics codes are altered as little as possible. The surfaces of the structure where pedestrians transit are used to transfer the required information between codes. The transfer of loads, displacements, and velocities is carried out via fast interpolation and projection algorithms.

This loosely coupled algorithm can also be modified in order to obtain a fully coupled algorithm via iterative loops.

Experience gained from several calculations indicates that the proposed approach offers a convenient and cost-effective way of coupling CSD and CCD codes without a complete re-write of them.

As every other technical endeavour, many improvements are possible. Among the most obvious we mention: link to other structural mechanics codes (e.g. nonlinear finite element codes to take into account dampers); analysis to obtain the optimal location of sensors for lifetime monitoring; and improvements in the modeling of the vertical and lateral forces pedestrians tend to have on moving ground.

The final paper will contain further details on all algorithmic aspects of the procedures, as well as more examples.

REFERENCES

[1] H. Bachmann and B. Weber - Tuned Vibration Absorbers For Damping Of "Lively" Structures, *Structural Engineering International* , 5, 31-36 (1995).
[2] H. Bachmann - "Lively" Footbridges - a Real Challenge; *Footbridge* (2002).
[3] M. Isenhour and R. Löhner - Verification of a Pedestrian Simulation Tool Using the NIST Recommended Test Cases; *The Conference in Pedestrian and Evacuation Dynamics 2014 (PED2014), Transportation Research Procedia* 2, 237-245 (2014).
[4] R. Löhner - On the Modeling of Pedestrian Motion; *Appl. Math. Modelling* 34, 2, 366-382 (2010).
[5] Millennium Bridge, London; https://en.wikipedia.org/wiki/Millennium_Bridge,_London (2022).
[6] O.C. Zienkiewicz and R. Taylor - *The Finite Element Method*; McGraw Hill (1988).