Dynamic analysis techniques for road bridges under large traffic flows

Alfredo Camara

Department of Continuum Mechanics and Theory of Structures, Universidad Politécnica de Madrid, Calle Profesor Aranguren s/n, 28040 Madrid, Spain.
alfredo.camara@upm.es

Jie Wang

Department of Continuum Mechanics and Theory of Structures, Universidad Politécnica de Madrid, Calle Profesor Aranguren s/n, 28040 Madrid, Spain.
jie.wang@upm.es

José M. Goicolea

Department of Continuum Mechanics and Theory of Structures, Universidad Politécnica de Madrid, Calle Profesor Aranguren s/n, 28040 Madrid, Spain.
jose.goicolea@upm.es

This work focuses on the dynamic analysis of road bridges under large microsimulated traffic flows, contributing an efficient dynamic analysis framework for future fatigue, comfort or driving safety studies in which realistic interactions between vehicles are considered. Different strategies to define the loading vector in the governing dynamic equations are presented for bridges discretised with beam and shell elements. The proposed interpolation techniques for beam-like models include a recursive binary search and a novel vectorised strategy based on shape functions expanded to the entire length of the deck. These are applied to a short simply supported bridge and a very long cable-stayed bridge under traffic flows generated with a cellular automata microsimulation model. The strategies presented here reduce the computational time required in conventional load interpolation algorithms, particularly in long bridges subject to dense traffic flows. Finally, the binary search algorithm is applied to the study of bridge decks discretised with shell elements under microsimulated traffic, demonstrating that high vehicle densities increase the contribution of the transverse vibration of the slab and the potential discomfort of pedestrians.

Keywords: Microsimulated traffic; bridge dynamics; load interpolation; algorithm efficiency.

1. Introduction

Traffic loads on bridges are stochastic dynamic actions that vary over time and are inherently uncertain, leading to potential discomfort, driving accident risks and fatigue accumulation.[1] The latter is a key problem in bridges with metallic or composite decks, and several computational frameworks have been developed for the probabilistic modelling of fatigue damage using Finite Element (FE) analysis.[2–4] However, these techniques usually simplify the traffic actions with convoys of equally spaced vehicles that are static or move at constant speeds, and this is due to the prohibitive computational cost involved in the dynamic analysis of bridges under long traffic flows in which realistic interactions between individual vehicles are considered. Indeed, it is recognised that the critical step for applying the FE analysis approach to structural fatigue stress analysis is determining how to simulate both the structural geometries and the truck loads more realistically.[3,5] On the other hand,

comfort and driving safety analysis entail complex vehicle-bridge interactions but, again, the complexity of the numerical model leads to simplified traffic load cases with a single vehicle crossing the deck,[6,7] or a convoy of equally-spaced vehicles with constant speed and straight paths.[8,9] This clearly differs from real traffic flows in road bridges, particularly in long structures, where the vehicles modify their trajectories based on the surrounding traffic.

There is a significant body of knowledge in the field of traffic microsimulation modelling to describe the behaviour of each vehicle in the traffic flow, particularly using cellular automata (CA) algorithms to define speed changes based on the vehicles ahead,[10] lane changing,[11] imperfect driving and traffic flow randomness, or the formation of jam structures, among others.[12] A number of research works have been published on bridge dynamics with microsimulated traffic actions,[13–15] but this analysis can be computationally demanding because it needs to include a large number of vehicles moving on the structure for a sufficiently long time to obtain representative results.[16–18] Most of these studies solve the bridge dynamics in time-domain using modal superposition techniques thanks to its computational efficiency, and define the deck using beam elements that account for the vertical bending and the pure torsion introduced by the moving traffic. However, this type of deck discretisation lacks the necessary detail for the fatigue assessment of structural details,[4] and it ignores the contribution of the transverse flexure of the slab to the vehicle-induced vibrations, which is very significant in relatively wide bridges and it requires a detailed FE model with shell elements in the slab.[19,20]

It appears that the computational efficiency of bridge models subject to moving loads is important, but none of the previous works focus on the strategies used to introduce them in the dynamic analysis of the structure. This work contributes new methods for massive interpolation of the moving traffic in beam- and shell-like discretisations of the deck, including (1) a binary recursive search of the deck nodes adjacent to each wheel; and (2) a vectorised algorithm that allows the simultaneous calculation of all the actions in all the nodes of the structure. Large microsimulated traffic flows of heterogeneous vehicles are applied to the dynamic anlaysis of two representative bridges: a short simply-supported deck modelled with beam and shell elements, and a very long cable-stayed bridge. The results of the one-dimensional (1D) interpolation applied to beam-like deck discretisations indicate that the proposed methods are able to reduce the computational time required in the dynamic analysis of bridges under dense traffic flows by up to 47% compared with conventional nodal load lumping strategies. This reduction is up to 16% in the proposed two-dimensional (2D) interpolation for shell-like models using the binary search.

2. One-dimensional (1D) load interpolation algorithms

Let's consider a 3D bridge model subject to moving vehicles that is discretised with beam-type elements interconnecting N_n nodes, of which N_{nd} nodes correspond to the deck. In the most general case each node of the structure has 6 different types of structural movements (SM, three translations and three rotations) per node, giving a total of $N = 6N_n$ degrees of freedom (DOF). It will be assumed that the dynamic response of the bridge under

moving traffic is linear and therefore the superposition principle holds. The time-history contribution of the j-th vibration mode to the structural response is represented with the modal coordinate q_j and its time-derivatives ($\dot{q}_j$ and $\ddot{q}_j$)

$$\ddot{q}_j(t) + 2\xi_j\omega_j\dot{q}_j(t) + \omega_j^2 q_j(t) = \frac{\boldsymbol{\phi}_j^{\mathrm{T}}\mathbf{P}(t)}{m_j}\,, \tag{2.1}$$

where $\boldsymbol{\phi}_j, \omega_j, \xi_j$ and m_j refer to the shape, circular frequency, damping ratio and mass of the j-th mode, respectively; $\mathbf{P}_{|N\times 1}(t)$ is the nodal forcing vector that contains the generalised forces introduced by the moving vehicles, with the sub-index product denoting the array dimensions: rows × columns. For convenience, in this work $\mathbf{P}$ is organised as follows

$$\mathbf{P} = \{\underbrace{P_1^1, \cdots, P_{N_n}^1}_{\text{SM=1}}, \underbrace{P_1^2, \cdots, P_{N_n}^2}_{\text{SM=2}}, \cdots, \underbrace{P_1^6, \cdots, P_{N_n}^6}_{\text{SM=6}}\}^{\mathrm{T}}\,, \tag{2.2}$$

in which P_k^{SM} is the generalised force at the k-th node of the structure related to the associated SM, with SM = $1, \cdots, 3$ referring to the forces in the X, Y, Z axes, respectively, and SM = $4, \cdots, 6$ denoting the corresponding moments. Without loss of generality this work considers a straight bridge with deck parallel to the X axis in which SM = 3 refers to the vertical Z direction, and SM = 4 to the torsional response. Horizontal (braking) and lateral wheel forces are ignored in this study, as well as the vehicle vibration and its interaction with the bridge. These effects can be incorporated in the proposed models in future works, but they are not included here to focus on the efficiency of the load interpolation techniques in the definition of $\mathbf{P}$ for very large volumes of traffic. Fig. 1 shows the two methods proposed in this study to accelerate the dynamic analysis, where t_i in this figure refers to the time instant at the step i of the analysis, with $t_0 = 0$ s, $t_i = t_{ea}$, and Δt being the start step, end step and the time between consecutive steps, respectively.

2.1. *A fast recursive method*

The recursive algorithm to define the moving loads in the analysis treats them separately and superimposes their individual contributions to the nodal force vector $\mathbf{P}$, as shown in Fig. 1(a). This algorithm includes a nested loop in which the outer layer sets the time instant under consideration (t_i) and the inner one ranges over a vector $\mathbf{W}_{v|1\times N_{wh}}$ that contains the weight of each wheel l on the deck, where N_{wh} is the total number of wheels on the deck at the instant t_i.

The calculation of $\mathbf{P}$ in the inner loop starts with the localisation of the l-th load in the longitudinal ($X_{wh,l}$) and transverse ($Y_{wh,l}$) directions at the time t_i. These are obtained from the microsimulation of the vehicle movement on the road conducted prior to the dynamic analysis.

The second step is to find the beam element in which the load l is positioned at the instant t_i. It is considered now that the vehicle moves in the positive sense of the axis X parallel to the deck, and the X-coordinates of the N_{nd} nodes of the deck sorted in ascending order are

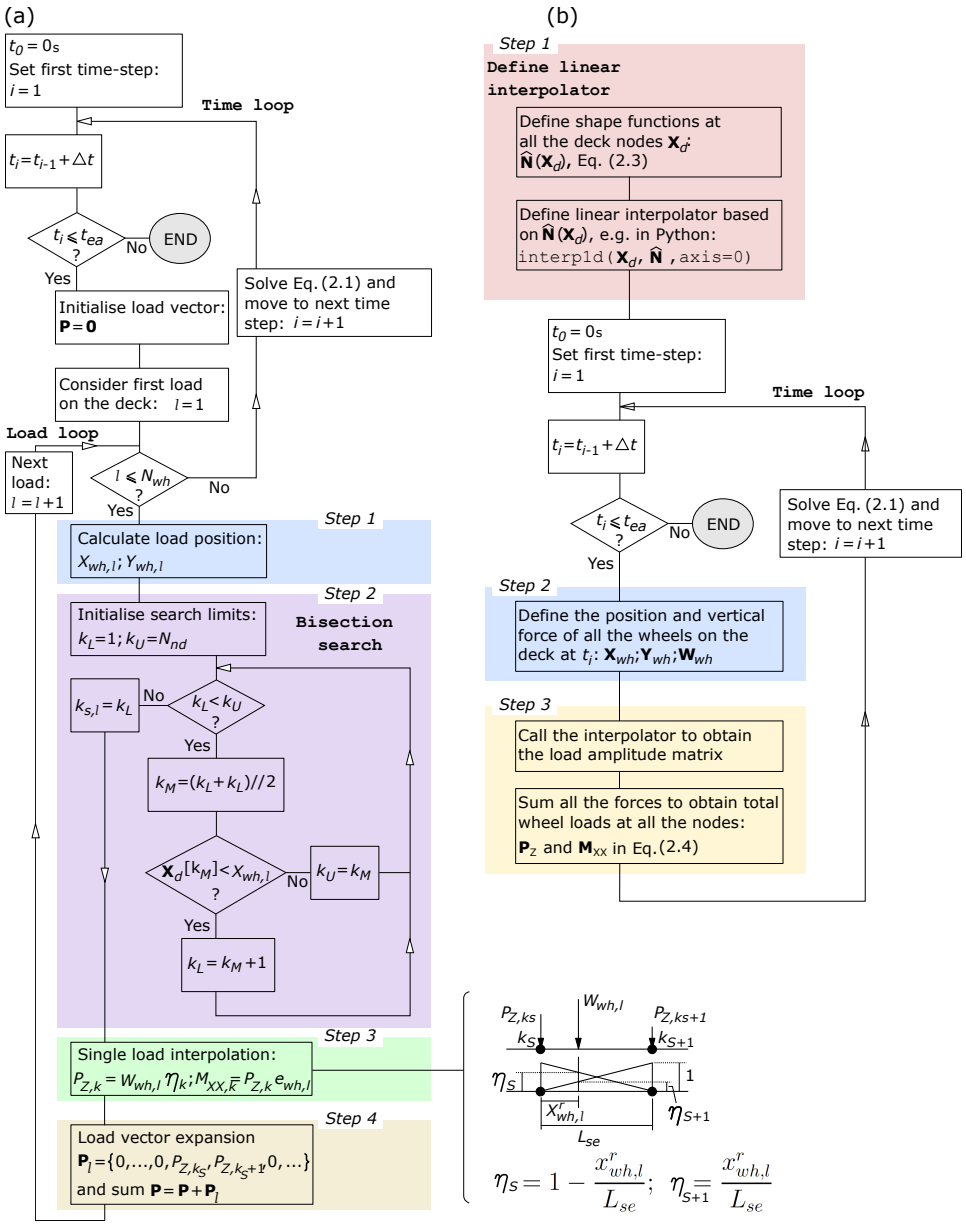


Fig. 1. Proposed algorithms for the definition of moving loads in the time-history analysis: (a) recursive method with bisection search, (b) vectorised method.'//' represents the integer division.

introduced in the vector $\mathbf{X}_{d|N_{nd}\times 1}$. In this case the search of the loaded beam element of the deck is reduced to finding in $\mathbf{X}_d$ the position (order) of the starting node of the beam element

that is closer to the load, which is referred to as k_s. Fig. 2 shows the wallclock time required by a conventional 32-GB RAM Workstation (reference computer used in this work) to find the value of k_s from a single load on a 100-m deck defined by a vector $\mathbf{X}_d$ of increasing length, ranging from $N_{nd} = 2$ to $N_{nd} = 1001$, and using different search strategies in Python programming language.[21] The thicker lines in this figure represent the arithmetic mean of 100 repetitions of the search of k_s with the load positioned randomly on the deck. The vertical error bars indicate one standard deviation above an below the mean value. The use of conditional search methods so that $\mathbf{X}_d \leq X_{wh,l} \rightarrow k_s$, such as the `numpy.where()` built-in function in Python, is straightforward. However, it has a time-complexity $O(N_{nd})$ that increases significantly the calculation time with the number of nodes in the deck due to the implicit `FOR`-loop that iterates over the entire length of $\mathbf{X}_d$ (i.e. N_{nd}). This loop can be avoided with a `while` structure that makes the time-complexity $O(\log(N_{nd}))$, as shown in Fig. 2. However, it results in large computational costs for long bridges, and if the deck has more than $N_{nd} = 30$ nodes it is the most expensive search method.

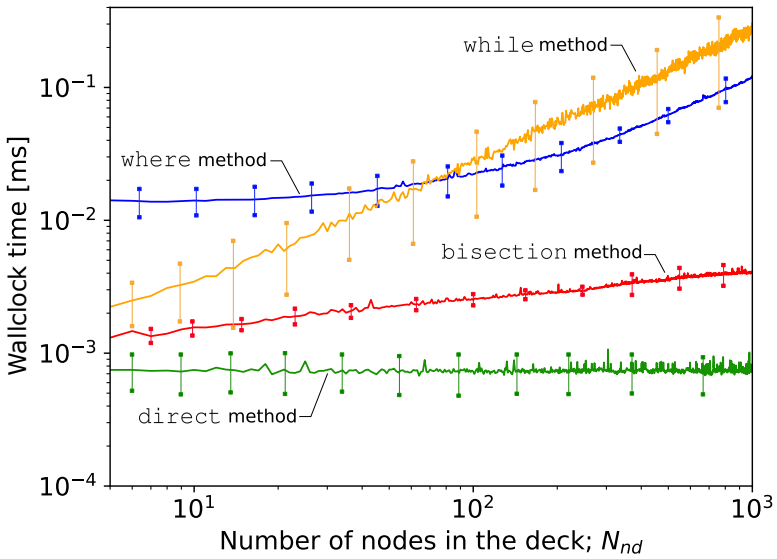


Fig. 2. Wallclock time associated with the search of the starting node (k_s) of the element in which a single wheel ($N_{wh} = 1$) is located within a 100-m line (deck) divided in N_{nd} nodes. The thicker lines refer to the arithmetic mean of 100 different results, and the error bars to their standard deviation.

The efficiency of the calculation of k_s can be improved with the `bisection` (or binary) recursive method, which halves the search interval in each iteration as shown in Fig. 1(a). The search process starts by selecting the median of $\mathbf{X}_d$ and comparing it against the target X_{wh}. If it is lower than X_{wh} the next iteration is performed on the upper half of $\mathbf{X}_d$. Otherwise the lower half of the search vector is considered. The algorithm iterates until k_s

is found but it requires less operations than the previous search methods. This is reflected in Fig. 2. Like in the `while` structure the performance of the binary search is $O(\log(N_{nd}))$, but the latter is significantly less influenced by the number of nodes in $\mathbf{X}_d$, which makes the `bisection` search suitable for the analysis of large bridges (where $N_{nd} >>$) as it will be further demonstrated later. Finally, Fig. 2 shows that the most efficient strategy to obtain k_s is to avoid searching for it in the $\mathbf{X}_d$ vector by using the `direct` method, in which $k_s = X_{wh}//\Delta X_d$, where (//) is the integer division and ΔX_d is the spacing between nodes in the deck. The `direct` method is $O(1)$-complex because it does not depend on N_d, but it can only be applied if ΔX_d and the speed of the vehicles are constants. The latter is not the case in realistic traffic flows and it will not be considered further in this work.

After the loaded beam is found (k_s), the third step is to interpolate the moving load with the amplitude factor η described in Fig. 1(a), obtaining a vertical load and a torsional moment applied at its nodes: P_Z and $M_{XX} = P_Z e_{wh,l}$, with $e_{wh,l}$ being the distance between the load and the centreline of the deck. These forces and moments are added recursively to account for the contribution of all the wheels at the time t_i, and introduced to the corresponding DOFs in the forcing vector $\mathbf{P}$.

When the process is complete and the forcing vector is obtained Eq. (2.1) can be solved and the analysis advances to the next time step $(i + 1)$, initializing $\mathbf{P} = \mathbf{0}$ and repeating the calculation of the vehicle action for the new position of the wheels on the bridge. The wallclock time associated with the four steps involved in the recursive calculation of $\mathbf{P}$ is included in Fig. 3 for the `while` and the `bisection` search methods. Fig. 3(a) shows the influence of the number of nodes in the deck (N_{nd}) for a constant number of moving loads $N_{wh} = 20$, and it indicates that the time-complexity of the entire calculation process of $\mathbf{P}$ is governed by the search algorithm, giving $O(N_{nd})$ for the `while` method and $O(\log(N_{nd}))$ for the `bisection` method. The results suggest that the latter is more efficient than the former in bridges with more than $N_{nd} = 10$ nodes in the deck. Fig. 3(b) illustrates the effect that the number of vehicle wheels on the deck (N_{wh}) has on the wallclock time of the calculation of $\mathbf{P}$, considering a 500-m long deck with a constant number of nodes ($N_{nd} = 101$). In both recursive methods the time-complexity is $O(\log(N_{wh}))$, but the computational effort increases significantly with the number of moving loads because the four steps in the calculation of the nodal forcing are repeated for each vehicle wheel.

2.2. *A vectorised method*

The strong influence of N_{wh} on the efficiency of the recursive methods is problematic in bridges subject to a large number of moving vehicles. An example of this type of traffic flow is illustrated in Fig. 4(a), which shows the positions of vehicles moving in a road segment generated with CA traffic microsimulation. We propose here to vectorise the load interpolation algorithm so that all the nodal loads at each time instant are obtained in single array-based operations, avoiding the inner loop that treats each wheel individually. This is achieved in three steps presented in Fig. 1(b).

The first step is to build an array-based interpolating function that takes the instantaneous position of all the loads in the structure at the time-step t_i and returns the corresponding

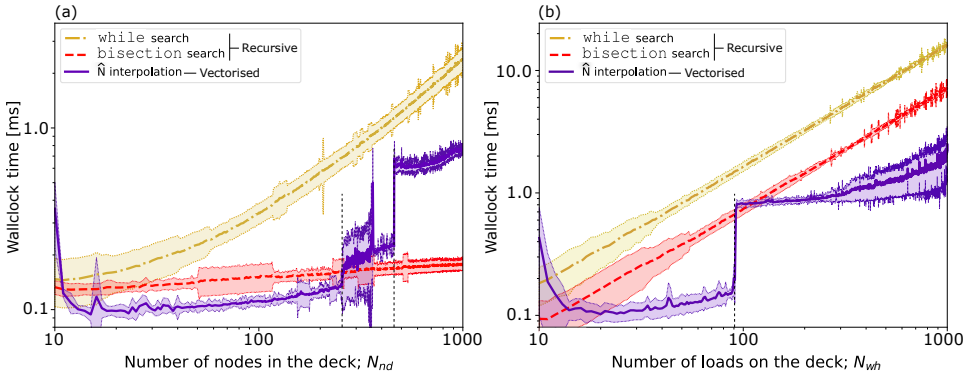


Fig. 3. Wallclock time associated with the calculation of **P** in Python; (a) influence of the number of nodes (N_{nd}) in a 100-m deck with $N_{wh} = 20$ wheels; (b) influence of the number of wheel loads (N_{wh}) in a 500-m deck with $N_{nd} = 101$ nodes.

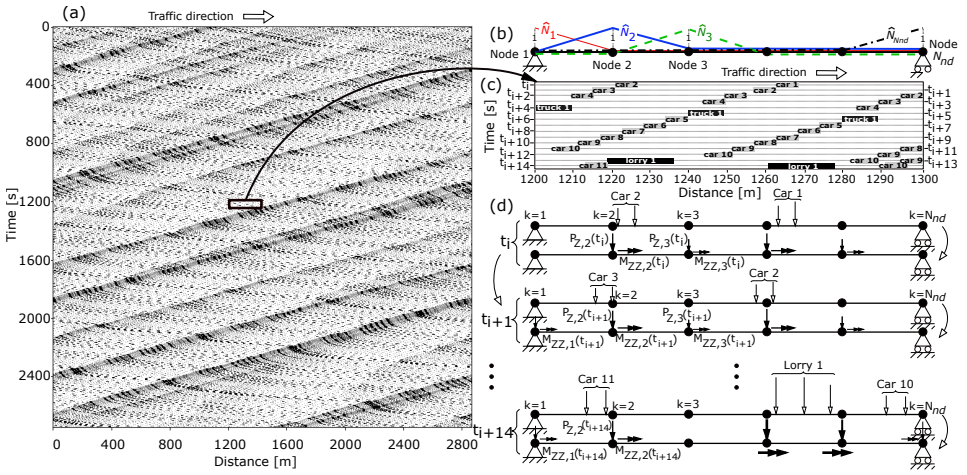


Fig. 4. Definition of the nodal loads in bridges under traffic flows; (a) example of space-time microsimulation of mixed traffic in a road lane; (b) extended shape functions; (c) detail of the vehicle distribution at different time instants in a region of the microsulated traffic that corresponds to a bridge located at the centre of the road segment for illustration purposes; (d) conversion of the point loads in the elements of the deck to nodal loads at different time instants.

nodal amplitude factors. The base of this interpolator is a set of linear shape functions $\widehat{N}_k(x)$ that are associated with each node k of the deck (with coordinates $\mathbf{X}_d$). There are N_{nd} linear interpolating shape functions that are entirely described by their nodal coordinates as

$$\widehat{\mathbf{N}}(\mathbf{X}_d) = \begin{array}{c} \\ \widehat{N}_1 \\ \widehat{N}_2 \\ \vdots \\ \widehat{N}_{nd} \\ \\ \widehat{N}_1 \end{array} \begin{array}{c} \begin{array}{cccc} k=1 & k=2 & \cdots & k=N_{nd} \end{array} \\ \left(\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{array} \right) \\ \begin{array}{cccc} \widehat{N}_2 & \cdots & \widehat{N}_{nd} \end{array} \end{array} \begin{array}{c} \\ k=1 \\ k=2 \\ \vdots \\ k=N_{nd} \\ \\ \end{array} = \mathbf{I} \,, \tag{2.3}$$

where $\mathbf{I}_{|N_{nd}\times N_{nd}}$ is the identity matrix. The functions $\widehat{N}_k$ are the same as the element-based interpolators used in the recursive method of Fig. 1(a), but extended to the entire length of the deck as shown in Figure 4(b). The extended $\widehat{N}_k$ functions allow the vectorisation of the interpolator in terms of the coordinates of all the nodes in the deck ($\mathbf{X}_d$).

In the second step, the position of all the wheels on the deck at the time instant t_i (obtained from e.g. traffic microsimulation) is stored in the arrays $\mathbf{X}_{wh|1\times N_{wh}}$ (longitudinal distance from the wheels to the left abutment) and $\mathbf{Y}_{wh|1\times N_{wh}}$ (transverse distance from the wheels to the centroid of the deck). Analogously, the value of all the wheel loads is included in $\mathbf{W}_{wh|1\times N_{wh}}$, which can be defined before the time-loop starts for computational efficiency if these are constant, or substituded by the instantaneous wheel reaction forces calculated inside the loop if there is vehicle-bridge interaction (not in this study). For illustration purposes, Fig. 4(c) shows a detail of the vehicle positions at different time-instants in a bridge located within the traffic flow of Fig. 4(a).

In the third step, the total traffic load at each node of the deck is obtained. To this end, the linear interpolator with base $\widehat{\mathbf{N}}$ is called within the time-loop to obtain the array with all the nodal loads at the i-th time-step, as shown in Fig. 1(b). The array $\mathbf{X}_{wh}$ is introduced as the argument of the vectorised interpolator to obtain the matrix $\boldsymbol{\eta}_{|N_{wh}\times N_{nd}}$ with the amplitude factors at the time-instant t_i of all the N_{wh} vehicle wheels on the deck in all its N_{nd} nodes.

The sum of all the wheel actions concentrated at each node is expressed implicitly as

$$\mathbf{P}_Z = \mathbf{W}_v \boldsymbol{\eta};\ \mathbf{M}_{XX} = (\mathbf{W}_v\ .*\ \mathbf{Y}_{wh})\boldsymbol{\eta}\ , \tag{2.4}$$

in which '.*' denotes element-wise multiplication of arrays, and $\mathbf{P}_{Z|1\times N_{nd}}$ and $\mathbf{M}_{XX|1\times N_{nd}}$ are vectors that contain in their k-th components the vertical forces and the torsional moments (respectively) at the k-th deck node due to the combined effect of all the wheels on the elements that include that node. Eq. (2.4) is introduced in the vectorised code as shown in Fig. 1(b). It is remarked that the calculation of the torsional moments in the deck ($\mathbf{M}_{XX}$) depends directly on the eccentricity of the vertical wheel loads with respect to the deck centreline ($\mathbf{Y}_{wh}$) at the instant t_i, which introduces in the bridge the dynamic effects of vehicles changing lanes. Finally, the nodal loads and moments of the deck are introduced in the corresponding positions for these actions in the vector $\mathbf{P}$ (illustrated in Fig. 4(d) at different time steps)

$$\mathbf{P} = \{ \underbrace{\mathbf{0}}_{\text{SM=1,2}}, \underbrace{0, \cdots, 0, \mathbf{P}_Z, 0, \cdots, 0}_{\text{SM=3}}, \underbrace{0, \cdots, 0, \mathbf{M}_{XX}, 0, \cdots, 0}_{\text{SM=4}}, \underbrace{\mathbf{0}}_{\text{SM=5,6}} \}^{\mathrm{T}} . \qquad (2.5)$$

The wallclock time involved in the vectorised load interpolation of a group of 20 wheels randomly positioned on a deck with increasing number of nodes is presented in Fig. 3(a). It is observed that the vectorised interpolation is faster than the recursive approaches and it is almost insensitive to the discretisation of the deck (i.e. $O(1)$) when the number of nodes ranges from $N_{nd} = 15$ to 200. Beyond this number there are sharp increments of the computational cost involved in the array-based linear interpolator due to the size of the cache memory of the computer. However, the real benefit of this method is in the interpolation of a large number of wheels, which is observed in Fig. 3(b) for a deck of 101 nodes and increasing number of wheels. The time-complexity of the vectorized algorithm in terms of N_{wh} approaches $O(1)$ for a large number of interpolated wheel loads and it is generally the fastest method, although it also shows an increment of the computational cost for more than 90 wheels.

3. Proposed bridges and microsimulated traffic records

The structures considered in this work are presented in Fig. 5. They include a short simply-supported bridge (SSB) and a long cable-stayed bridge (CSB) to study the influence of the length of the deck. The SSB is a typical 40-m span composite bridge with a single carriageway of two lanes that hold road traffic in the positive-X direction. The CSB has two main spans of 650 m each supported by a central plane of cables, and it represents the Queensferry Bridge in Scotland. It has a 4.9-m deep steel deck closed by a 39.8-m wide concrete slab. The piers and the towers of the CSB restrain the transverse and the torsional movements of the deck, but the vertical one is not constrained at the side towers (T1 and T3 in Fig. 5(b)). The deck is fully fixed to the central tower (T2) and it is restrained to vertical and torsional movements at the abutments and at the intermediate piers P1-P10. The CSB has two carriageways, each with two lanes and the eccentricities shown in Fig. 5(b). The deck of the SSB and the CSB is discretised with $N_{nd} = 101$ and 230 nodes, respectively. More details of these FE models are given by Camara.[22]

The three types of vehicles summarised in Table 1 are considered in the traffic records of this work. They include cars as well as medium- and large-size trucks described by AASHTO,[23] namely H20-44 and HS20-44. A two-lane stochastic traffic cellular automata (CA) model was implemented in Python to simulate traffic flows in each carriageway. The movement of vehicles within their lane follow the NaSchr rules,[10] and the vehicle lane changes are simulated with the asymmetric rules of Rickert *et al.*[11] This model was extended to consider mixed traffic conditions with vehicles of different characteristics (length, velocity and acceleration). The vehicle occupancy (or density) is defined as the ratio between the length occupied by all the vehicles (L_v) and two times the length of the road in which traffic is simulated, L_{road}, because the carriageway has two lanes: $\rho = L_v/(2L_{road})$. In the CSB, the traffic in both carriageways of the deck is simulated independently, with the vehicles

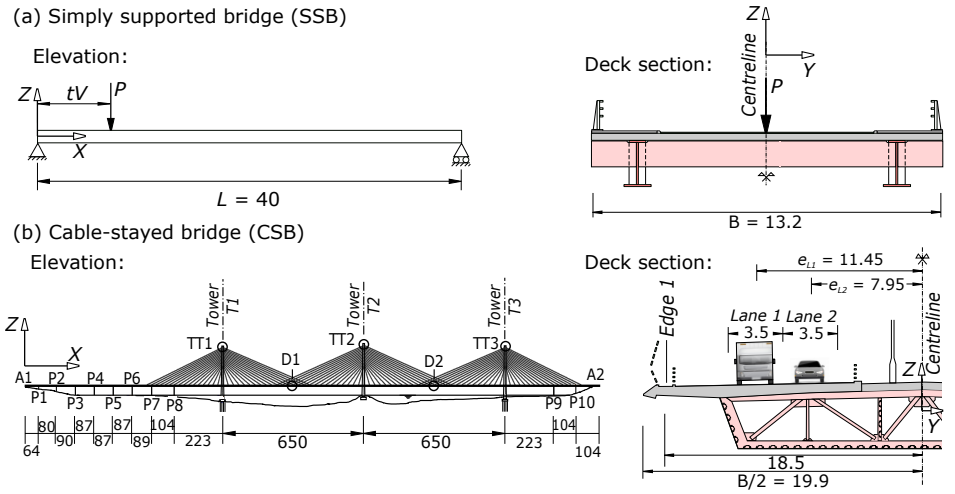


Fig. 5. Elevation and deck cross-section of the proposed bridges: (a) simply supported bridge (SSB), (b) Queensferry cable-stayed bridge (CSB). Dimensions in meters.

moving in the positive-X direction occupying Lane 1 (slow lane) and Lane 2 (fast lane), and those moving in the negative-X direction on Lane 3 (slow) and Lane 4 (fast).

The weight of each vehicle of the same type in the simulation (W) is defined from a normal probability distribution with the mean, maximum and minimum (unladen) weights defined in Table 1. The same strategy is used to randomise the vehicle acceleration and speed properties, although the latter represents the velocity that the driver would like to reach but is only possible if the surrounding traffic allows it. The stochasticity of the NaSchr/Rickter traffic model is introduced by a probability of unjustified braking, $p_b = 0.1$, and lane changing (p_c) of the vehicles in each step of the simulation. The probability of vehicles moving from the slow lane to the fast lane is $p_c = 0.8, 0.5, 0.1$ for cars, small trucks and large trucks, respectively, whilst $p_c = 0.5, 0.8, 0.9$ for changes from the fast lane to the slow one. The analysis of the macroscopic traffic properties obtained from ensemble statistics in sets of 100 random microsimulated records with different densities rendered a realistic bi-linear capacity (vehicles/s) curve in which the maximum traffic flow is achieved for a critical occupancy of $\rho_c = 0.15$, beyond which increments in the number of vehicles result in decrements of the vehicles crossing the road due to the spontaneous formation of jam structures.

The CA model presents open periodic boundary conditions so that the vehicles reaching the end of the simulated segment of the road appear immediately at the start with the same velocity and acceleration, but with a randomly allocated new weight because they are considered as new vehicles accessing the road in the dynamic analysis of the bridge. The proposed load interpolation strategies are implemented in the Python-based mode superposition solver MDyn[22] to obtain the dynamic response of bridges under traffic actions. In the first 100 s of the simulation the vehicle load on the bridge is increased linearly to

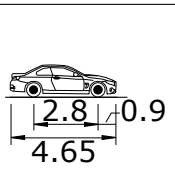
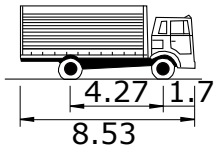
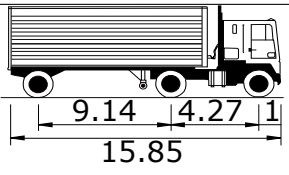
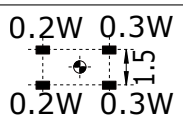
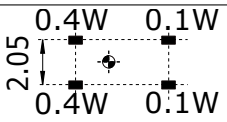
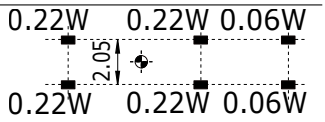
	Car	H20-44 truck	HS20-44 truck
Elevation			
Plan: wheel loads			
MaxP weight	20.6 kN	177.9 kN	320.3 kN
Mean weight	14.4 kN	124.6 kN	224.2 kN
MinP weight	5 kN	20 kN	100 kN
SD weight	1 kN	9 kN	20 kN
MaxP speed	200 km/h	144 km/h	108 km/h
Mean* speed	120 km/h	90 km/h	90 km/h
MinP* speed	72 km/h	36 km/h	36 km/h
SD speed	12 km/h	9 km/h	9 km/h
Mean* accel.	3 m/s^2	1 m/s^2	0.5 m/s^2
MinP* accel.	1 m/s^2	0.5 m/s^2	0.5 m/s^2
SD accel.	0.3 m/s^2	0.12 m/s^2	0.12 m/s^2

Table 1. Geometry and dimensions of the vehicles considered in the analysis of the CSB. MaxP, Mean, MinP and SD refer to the maximum possible, mean, minimum possible and standard deviation of the total weight of the vehicle (W), the vehicle speed (V) or its acceleration (accel.). Dimensions in m.
*The driving speeds and accelerations refer to the most likely (mean) or minimum (MinP) values chosen by the drivers if they were not affected by the surrounding traffic.

avoid impulsive loads from the vehicles located on the deck at $t = 0$ s. The dynamic analysis has a time-step of $\Delta t = 0.01$ s and gives a structural damping ratio of $\xi_j = 0.5\%$ to all the vibration modes. The simulated time is 1100 s and the length of the road is $L_{road} = 3180$ m, with both bridges centered in this segment.

4. Application of 1D interpolation methods

4.1. *Computational efficiency*

We consider now three values of the vehicle density: $\rho = 0.07, 0.15$ and 0.3, which are representative of sparse, critical and dense traffic flows, respectively. These flows give approximately 15 cars/km, 5 medium-trucks/km and 2 heavy-trucks/km (in short: 15-5-2 vehicles/km) in the sparse flow, 32-11-4 vehicles/km in the critical flow, and 65-21-8 vehicles/km in the dense flow (including both lanes). A total of 100 randomly generated records are simulated for each density level to average the computational time and account for its

potential record-to-record variability. The same density is considered in both carriageways of the CSB.

Fig. 6 gathers the wallclock time required in the reference computer used for this work to complete the dynamic analysis of both bridges. The size of the bars in this plot refers to the ensemble average of the computing time per record, whereas the size of the error bar indicates the corresponding standard deviation. The results show that the calculation time increases with the number of wheels on the deck and, particularly, with the number of nodes in which it is divided. Fig. 6 (a) gives the time required to complete the analysis in the SSB, for which the most efficient method is the vectorised $\widehat{N}$-interpolation strategy, especially for large vehicle occupancies. This is due to the increment of the computational effort in the recursive method as the number of wheels increases, and it shows that avoiding the wheel-loop with the proposed vectorised algorithm can reduce the calculation time required to study dense flows by up to 17%. This represents a reduction of 3.8 hours in the analysis of 100 different records with $\rho = 0.03$ in this work.

The vectorised algorithm is also the most efficient one in the study of the long CSB, as shown in Fig. 6 (b), but the involved wallclock time is almost the same as in the recursive `bisection` method. In the study of the CSB the vectorised method requires handling a large interpolation matrix $\widehat{N}$ with $N_{nd} = 230$ shape functions extended over the entire length of a long dual-carriageway bridge, with a large amount of wheels on the deck in each time-step, and this partly compensates the benefits of avoiding the wheel-loop in the load interpolation algorithm. However, the search of the deck nodes surrounding each wheel with conventional recursive methods like `where` functions is much more inefficient, particularly as the vehicle occupancy increases and N_{wh} is larger. The proposed `bisection` and $\widehat{N}$-interpolated vectorised methods reduced by up to 47% the wallclock time required by the conventional `where` method, which implied a total reduction of 46 hours in the analysis of 100 records.

Overall the standard deviation of the results in all the cases described in Fig. 6 is small and this suggests that the proposed strategies are not significantly affected by record-to-record variations in the number of wheels crossing the bridge.

4.2. *Response of the CSB under microsimulated traffic*

Now we look at the dynamic response of the CSB under two characteristic traffic flows shown in Fig. 7. Load case LC1 represents a traffic scenario with the same vehicle density $\rho = 0.04 < \rho_c$ in both carriageways, for which vehicles cross the bridge maintaining their desired speeds most of the time, as shown in Figs. 7(a) and (b). A second load case (LC2) is proposed to represent a typical one-way rush hour, with a very dense traffic flow in the positive-X carriageway ($\rho = 0.52$) and a lower density in the negative-X carriageway ($\rho = 0.24$). Both densities are above the critical value and lead to the formation of traffic jams, but the traffic in the positive-X direction is more congested, particularly in Lane 1 as depicted in Figs. 7(a) and (b).

Fig. 7(c) presents the time-history of the vertical movement of the deck centreline at the midspan section D1, in the two load cases. The plot compares the results obtained using

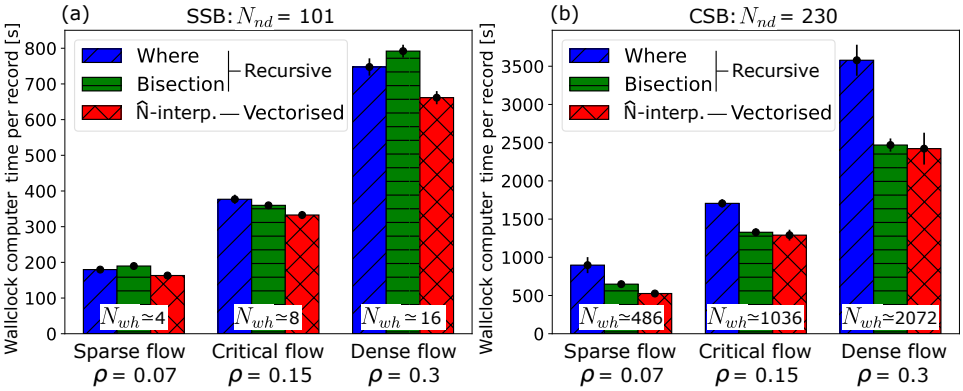


Fig. 6. Wallclock time required to complete the analysis of each record: (a) SSB, (b) CSB. The size of the color bars corresponds to the ensemble average of 100 records, and the error bars to one standard deviation above and below that value.

ABAQUS and MDyn (both with the `where` interpolation), and it shows that the differences are negligible in the whole duration of the analysis. The same results are obtained with other load interpolation techniques. It has been observed that the time intervals with larger displacements at the midspan section D1 in load case LC2 coincide with the concentration of heavy trucks at that central span, which also maximises the torsional response of the deck due to the different vehicle densities in both carriageways.

5. Two-dimensional (2D) load interpolation method

The previous beam-like FE models cannot capture the transverse bending and distortion of the deck sections, which is important in fatigue and comfort analysis of bridges with large width-to-span ratios such as the proposed SSB.[19,20] These require detailed shell- or brick-like FE models. In order to apply the analysis of long traffic flows to this type of models, the recursive strategies described previously are extended to a 2D interpolation of the loads in both the longitudinal (X) and transverse (Y) directions. The vectorised approach cannot be used directly in the 2D methodology because it is based on 1D linear interpolation functions.

At any given time t_i, the positioning of each load (*Step 1* in Fig. 1(a)) and the search of the adjacent nodes (*Step 2*) are analogous to those described for the recursive strategies, repeating the search process in X and Y directions to find the 4 nodes of the element on which the load is located. It is noted that in a simple case in which the load crosses a straight bridge in a straight path, and if the width of the elements is constant in the Y direction, the search of the loaded elements in this direction can be done before the time-loop starts. However, this is not possible with realistic traffic flows in which vehicles can change lanes, as it is the case in this work. Fig. 8(a) shows a detail of one of the loaded elements and the local node numbering used to define the relative position of the load with respect to the edges of the element. As it is normally the case in practice, it is considered here that

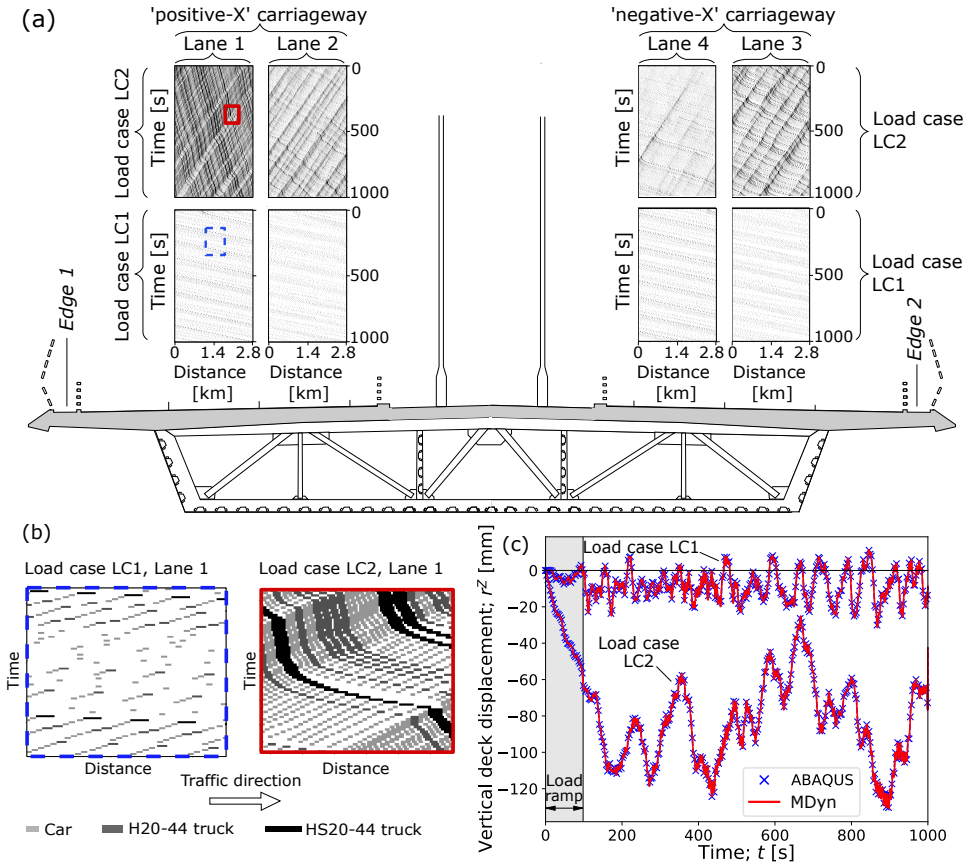


Fig. 7. (a) Microsimulated traffic flows on different lanes of the CSB deck in load cases LC1 and LC2; (b) details of the position of the vehicles at different space-time windows; (c) time-history of the vertical displacement of the deck centreline at point D1. The results are obtained with the `where` interpolation, but other methods give identical results.

the shell-type mesh of the deck is structured and therefore the elements have rectangular shape. In this case the load interpolation can be defined with the following shape functions according to the notation included in Fig. 8(a):

$$\eta_1 = \left(1 - \frac{x^r_{wh,l}}{L_{x,se}}\right)\left(1 - \frac{y^r_{wh,l}}{L_{y,se}}\right) \tag{5.1a}$$

$$\eta_2 = \left(\frac{x^r_{wh,l}}{L_{x,se}}\right)\left(1 - \frac{y^r_{wh,l}}{L_{y,se}}\right) \tag{5.1b}$$

$$\eta_3 = \left(\frac{x^r_{wh,l}}{L_{x,se}}\right)\left(\frac{y^r_{wh,l}}{L_{y,se}}\right) \tag{5.1c}$$

$$\eta_4 = \left(1 - \frac{x^r_{wh,l}}{L_{x,se}}\right)\left(\frac{y^r_{wh,l}}{L_{y,se}}\right) \tag{5.1d}$$

where $x^r_{wh,l}$ and $y^r_{wh,l}$ are the relative distances from the load to Node 1 of the loaded shell in the longitudinal and transverse directions, respectively; $L_{x,se}$ and $L_{y,se}$ are the corresponding element lengths. The interpolation of the wheel load at each of the four nodes of the shell element is: $P_{Z,k} = \eta_k W_{wh,l}$, with $k = 1, 2, 3, 4$ referring to the local numbering of the shell element in Fig. 8(a). Finally, the nodal load vector is expanded to include the degrees of freedom of other parts of the bridge considered in the modal dynamic analysis, if there are any. In this case the torsional action is introduced directly from the eccentricity of the FE nodes where the vertical load is lumped.

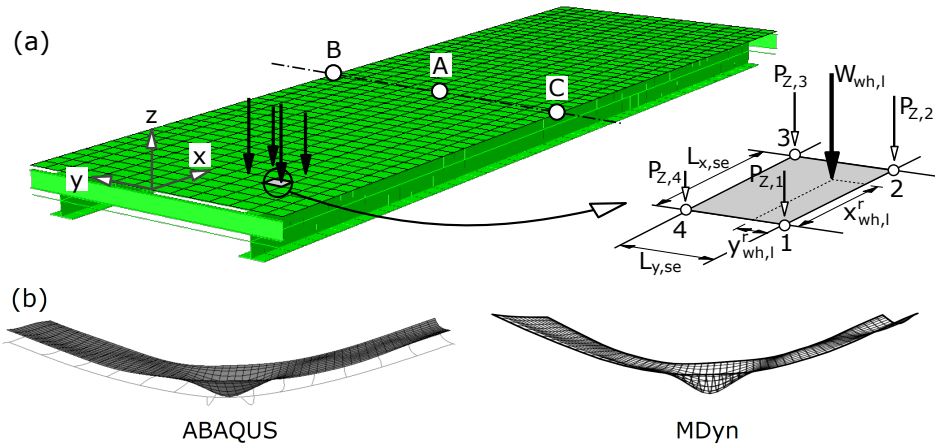


Fig. 8. (a) Load interpolation in shell-like deck models. (b) Deformation of the deck in the SSB at $t = 0.72$ s (scale factor of 1500), when the single load moving at constant speed is located at midspan

The proposed 2D load interpolation has been implemented in MDyn and applied to the FE model of the SSB described in Fig. 5(a). In this model the upper concrete slab is discretised with 728 full-integration shell elements of approximately 0.8×0.8 m. The longitudinal and transverse steel beams are modeled with linear interpolation beam elements rigidly connected to the concrete slab. In addition, lumped masses are included along the two edges of the deck to represent the mass of the sidewalks and parapets. The modal analysis conducted in the full FE model indicates that important high-order vibration modes with transverse flexure of the slab are in the range between 18 and 50 Hz. Therefore, the time-step of the dynamic analysis is $\Delta t = 0.002$ s to capture accurately these vibration modes.

First, a single moving load $P = 182.5$ kN moving with a constant speed of 100 km/h is considered in ABAQUS (including the full structure) and MDyn (with only the shell elements of the slab to reduce the computational cost). This load has an eccentricity of

2 m with respect to the centreline, in the positive sense of the y axis in Fig. 8(a). The deformation of the bridge when the load is at midspan (at $t = 0.72$ s) is included in Fig. 8(b), which shows no difference between the result obtained in ABAQUS and MDyn. Fig. 9(a) presents the time-history of the vertical displacement at the points A, B and C of the midspan cross-section (shown in Fig. 8) as the load crosses the bridge. In the same figure, the results obtained with the shell model (identical in ABAQUS and MDyn) are compared with those from the analytical solution of the bridge considered as a simply supported beam, and with the static displacement obtained when the load is applied at midspan in the same beam model. The comparison starts suggesting the importance of high-order vibration modes that excite the transverse response of the concrete slab between the two longitudinal girders, and the need for shell element discretisations to capture it.

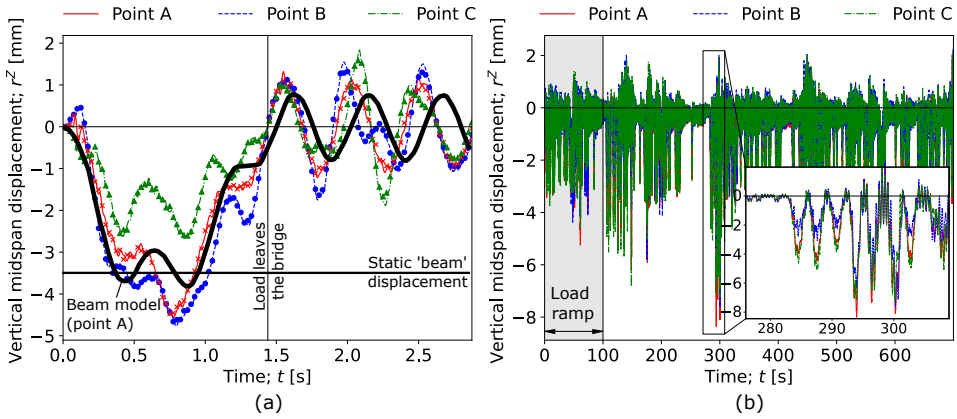


Fig. 9. Vertical displacements of the bridge at midspan subject to: (a) single moving load with constant eccentricity of 2 m, (b) critical traffic flow with $\rho = 0.15$. SSB modeled with shell elements. The lines represent the results from MDyn and the markers the results from ABAQUS.

The SSB modeled with shell elements was subject to traffic microsimulated flows of different vehicle densities in MDyn (ABAQUS is not used for this study due to the large computational time that it would require). In terms of the efficiency of the recursive load interpolation methods, the `bisection` algorithm is on average 16% faster than the `where` approach with the sub-critical traffic flow ($\rho = 0.07$), and it gives similar wallclock computer time with $\rho = 0.15$ and 0.3. Fig. 9(b) shows the time-history of the vertical displacement at the three points across the midspan section for a critical traffic flow ($\rho = 0.15$). The relatively short length of the SSB (40 m) makes it difficult for several heavy vehicles to coincide on its deck, and the passing of single HS20-44 trucks governs the displacement peaks observed in this plot.

Finally, the vibration of the shell-like SSB deck model is assessed based on the magnitude and frequency content of the vertical accelerations $a_k^Z(t)$ induced by different traffic

flows, where k indicates the node of the deck in which it is measured. The comfort criterion proposed by Irwin[24] is applied to study the vibrations sensed by pedestrians located on the bridge sidewalks. This is based on the Root Mean Square (RMS) acceleration at different one-third octave bands:

$$a^Z_{\mathrm{RMS}-k}(f_c) = \sqrt{\int_{f_l}^{f_u} S_{a^Z_k a^Z_k}\,\mathrm{d}f}\ , \tag{5.2}$$

in which $S_{a^Z_k a^Z_k}$ is the PSD of the acceleration signal $a^Z_k(t)$; f_l and f_u are the lower and the upper frequencies of each octave band, respectively. These can be expressed in terms of the corresponding central frequency (f_c) as: $f_l = 2^{-1/6} f_c$ and $f_u = 2^{1/6} f_c$. Fig. 10(a) plots the vertical RMS acceleration at midspan in one of the sidewalks (Point B in Fig. 8(a)) versus the central frequencies, and they are compared with the admissible vibration limit proposed by Irwin (a^Z_{adm}) for the three different traffic flow densities. The frequency content of the vertical acceleration at midspan shows that the risk of discomfort comes mainly from the contribution of the first vibration mode, which is a single half-wave of 2 Hz going from abutment to abutment that is also captured in the beam-like model. This source of vibrations is largely insensitive to the density of the traffic flow (ρ). However, increasing the value of ρ gives more importance to the contribution of higher-order vibration modes associated with the transverse flexure of the slab, like mode 3 (3.7 Hz) and mode 21 (24.3 Hz), that cannot be captured with the beam model. Although the tolerance of pedestrians to these frequencies is higher, the vibration due to the third mode reaches the comfort limit for critical and dense vehicle flows.

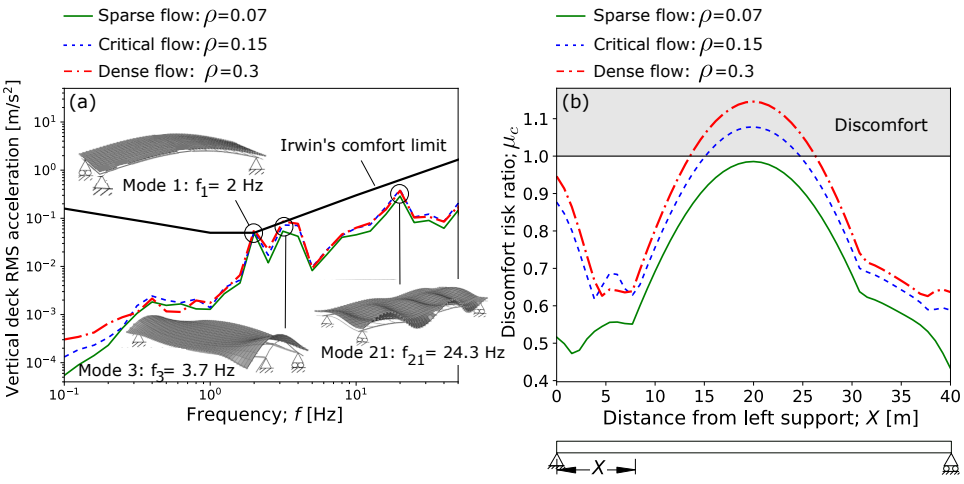


Fig. 10. Vibration comfort assessment based on the Irwin's criterion[24] (frequent conditions): (a) frequency content of the accelerations at midspan (point B), (b) discomfort risk ratio along the edge of the bridge that contains point B. SSB modeled with shell elements under microsimulated traffic with different vehicle densities (ρ).

The discomfort risk ratio proposed by Camara *et al.*[9] is used to evaluate the risk of pedestrians' discomfort along the entire sidewalk of the bridge:

$$\mu_c = \max_{f_c} \left[\frac{a^Z_{\mathrm{RMS}-k}(f_c)}{a^Z_{\mathrm{adm}}(f_c)} \right]. \tag{5.3}$$

Fig. 10(b) shows the discomfort risk ratio on the entire deck. The vertical vibrations along the sidewalk induced by the low-density flow ($\rho = 0.07$) are not classified as uncomfortable, but those caused by traffic actions with higher densities exceed the comfort limits in the midspan region. In addition, the increment of the discomfort risk near the left abutment for critical and dense flows is significant, and it is due to the larger contribution of the third vibration mode, which cannot be captured with a simple beam discretisation.

6. Conclusions

This work proposes two different one-dimensional (1D) interpolation strategies to define the loading vector in bridges under a large volume of moving vehicle forces. The first method is a recursive approach in which each wheel load is treated individually to find the adjacent nodes of the deck using a `bisection` algorithm. The second method uses extended shape functions to vectorise the node search, which allows to interpolate large amounts of moving loads without the need to consider each of them individually. The efficiency of both strategies is compared with that of conventional node-finding strategies in nodal and force vectors of increasing size, and in two structures that are representative of short and long bridges. These correspond to a 40-m span single carriageway simply supported bridge (SSB), and a 2.6-km long dual carriageway cable-stayed bridge (CSB). They are subject to large microsimulated traffic flows. In addition, recursive node-finding algorithms are extended to a two-dimensional (2D) load interpolation method that is used to study the dynamic response of the SSB modeled with shell elements. The following conclusions are drawn:

- The computational time involved in conventional methods that find recursively the nodes adjacent to each wheel force on the deck increases strongly with the number of nodes and wheels on the deck. The efficiency of the recursive search is improved by implementing a `bisection` approach in which the length of the search interval is halved, making the efficiency less dependant on the number of nodes to search and therefore more adequate in long bridges.
- The proposed vectorizaton of the 1D load interpolation algorithm is based on extended shape functions defined in the entire length of the deck, and it improves significantly the efficiency of the load definition when the number of moving vehicles is large.
- The extension of the recursive methods to a 2D load interpolation requires searching the adjacent nodes in the longitudinal and transverse directions, and it is based on structured discretisations of the deck with elements of rectangular shape. It

is observed that the `bisection` approach is faster than the `where` method in the shell-like model of the SSB under sparse traffic flows, but their computational time is similar for higher densities.

- The proposed load interpolation methods facilitate the study of large bridges under long traffic flows, and bridges modeled with detailed shell-like discretisations. The latter allowed to demonstrate that large vehicle densities can increase the contribution of high-order vibration modes that involve the transverse flexure of the slab, reducing the comfort of pedestrians.

References

1. J. Wu, S. Chen and J. V. de Lindt, Fatigue assessment of slender long-span bridges: reliability approach, *Journal of Bridge Engineering* **17(1)** (2012) 47–57.
2. T. Guo, D. Frangopol and Y. Chen, Fatigue reliability assessment of steel bridge details integrating weigh-in-motion data and probabilistic finite element analysis, *Computers and Structures* **112-113** (2012) 245–257.
3. N. Lu, M. Noori and Y. Liu, Fatigue reliability assessment of welded steel bridge decks under stochastic truck loads via machine learning, *Journal of Bridge Engineering* **22(1)** (2017) p. 04016105.
4. J. Zhu and W. Zhang, Probabilistic fatigue damage assessment of coastal slender bridges under coupled dynamic loads, *Engineering Structures* **166** (2018) 274–285.
5. X. Ye, Y. Su and J. Han, A state-of-the-art review on fatigue life assessment of steel bridges, *Mathematical Problems in Engineering* (2014) 1–13, ID:956473.
6. C. Cai and S. Chen, Framework of vehicle-bridge-wind dynamic analysis, *Journal of Wind Engineering and Industrial Aerodynamics* **92** (2004) 579–607.
7. S. Chen and C. Cai, Accident assessment of vehicles on long-span bridges in windy environments, *Journal of Wind Engineering and Industrial Aerodynamics* **92** (2004) 991–1024.
8. W. Wang, W. Han and W. Kong, Wind-vehicle-bridge coupled vibration analysis based on random traffic flow simulation, *Journal of Traffic and Transportation Engineering* **4**(1) (2014) 293–308.
9. A. Camara, I. Kavrakov, K. Nguyen and G. Morgenthal, Complete framework of wind-vehicle-bridge interaction with random road surfaces, *Journal of Sound and Vibration* **458** (2019) 197–217.
10. K. Nagel and M. Schreckenberg, A cellular automaton model for freeway traffic, *Journal de Physique I, EDP Sciences* **2**(12) (1992) 2221–2229.
11. M. Rickert, K. Nagel, M. Schreckenberg and A. Latour, Two lane traffic simulations using cellular automata, *Physica A* **231** (1996) 534–550.
12. S. Maerivoet and B. D. Moor, Cellular automata models of road traffic, *Physics reports* **419** (2005) 1–64.
13. S. Chen and J. Wu, Dynamic performance simulation of long-span bridge under combined loads of stochastic traffic and wind, *Journal of Bridge Engineering* **15**(3) (2010) 219–230.
14. J. Wu and S. Chen, Probabilistic dynamic behavior of a long-span bridge under extreme events, *Engineering Structures* **33** (2011) 1657–1665.
15. Y. Zhou and S. Chen, Fully coupled driving safety analysis of moving traffic on long-span bridges subjected to crosswind, *Journal of Wind Engineering and Industrial Aerodynamics* **143** (2015) 1–18.
16. X. Yin, L. Wang, B. Kong, G. Song and Y. Liu, Probability analysis of the vibration of bridges with rough surface under stochastic traffic, *International Journal of Structural Stability and Dynamics* **18(9)** (2018) p. 1850108.
17. J. Zhu, Z. Xiong, H. Xiang, X. Huang and Y. Li, Ride comfort evaluation of stochastic traffic

flow crossing long-span suspension bridge experiencing vortex-induced vibration, *Journal of Wind Engineering and Industrial Aerodynamics* **219** (2021) p. 104794.
18. G. Hou, S. Chen and F. Chen, Framework of simulation-based vehicle safety performance assessment of highway system under hazardous driving conditions, *Transportation Research Part C* **105** (2019) 23–36.
19. A. Camara and A. Ruiz-Teran, Multi-mode traffic-induced vibrations in composite ladder-deck bridges under heavy moving vehicles, *Journal of Sound and Vibration* **355** (2015) 264–283.
20. A. Camara, V. Vazquez, A. Ruiz-Teran and S. Paje, Influence of the pavement surface on the vibrations induced by heavy traffic in road bridges, *Canadian Journal of Civil Engineering* **12**(44) (2017) 1099–1111.
21. Python, Python 3.11.1 documentation (Python Software Foundation, 2022).
22. A. Camara, A fast mode superposition algorithm and its application to the analysis of bridges under moving loads, *Advances in Engineering Software* **151** (2021) p. 102934.
23. AASHTO, LRFD bridge design specifications (1998), 2nd Edition.
24. A. Irwin, Human response to dynamic motion of structures, *The Structural Engineer* **56A**(9) (1978) 237–244.