Multiphysics Hub

A real-time browser simulation in which one car couples
vehicle dynamics, aerodynamics, and vehicle–bridge interaction

Muhammet Emir Fil
B.Sc. Computational Engineering Science, RWTH Aachen • Technical report, June 21, 2026

Aim. This project pulls three otherwise separate simulation studies, vehicle dynamics, aerodynamics, and vehicle–bridge interaction, into one real-time application in which a single car couples all three. The wider goal is to show two things at once: that coupled multiphysics can run interactively in a plain web browser, and that such an interactive demo can still be tied to a verified numerical reference. Each model is built and checked against ground truth in Python, then re-implemented for the browser and compared back against that Python reference.

Abstract

A car is driven in real time around a user-drawn track. As it moves it couples three physics models that are recomputed every frame, with no pre-recorded data: a double-track (four-wheel) vehicle with Pacejka tyres and load transfer, a 2D fluid solver around the car silhouette that returns a drag and a downforce, and a moving-load bridge beam that deflects under the wheel force. Each model is first written in numpy and checked against an analytical or finite-element reference, then ported to JavaScript for the browser and compared against its numpy version on identical inputs (agreement of order 10^{-13} for the vehicle and 10^{-9} m for the bridge). This report describes the three models, states which couplings are physical and which are only illustrative, and gives the verification that backs the numbers shown live on screen.

1 Overview

The wider portfolio contains three standalone, separately verified projects: a minimum-lap-time racing-line optimiser, a vehicle–bridge interaction and Bridge Weigh-in-Motion simulator, and a Navier–Stokes vortex-street solver. The Multiphysics Hub connects them through the object they share, a car, into one real-time top-down application. Driving the car runs all three couplings together (Fig. 1): the tyres generate the forces, the air flowing past the body adds drag and downforce, and crossing a bridge deflects it.

2 The three models

2.1 Vehicle: a double-track car with Pacejka tyres

The car is a planar double-track (four-wheel) model. Each tyre's lateral force follows the Pacejka Magic Formula,

$$F_y = D\,\sin\Big(C\,\arctan\big(B\alpha - E(B\alpha - \arctan B\alpha)\big)\Big), \qquad D = \mu_{\text{eff}}(F_z)\,F_z, \tag{1}$$

and the longitudinal force is friction-ellipse coupled, $F_x = \text{thr}\cdot\sqrt{\max\big(0,(\mu F_z)^2 - F_y^2\big)}$, so a tyre cannot give full grip sideways and lengthways at the same time. The body-frame Newton–Euler equations, including the centripetal $v_y r$ and $v_x r$ terms, integrate the state $[X, Y, \psi, v_x, v_y, r]$ with RK4. The four wheels carry longitudinal and lateral load transfer, and since the tyre's peak friction

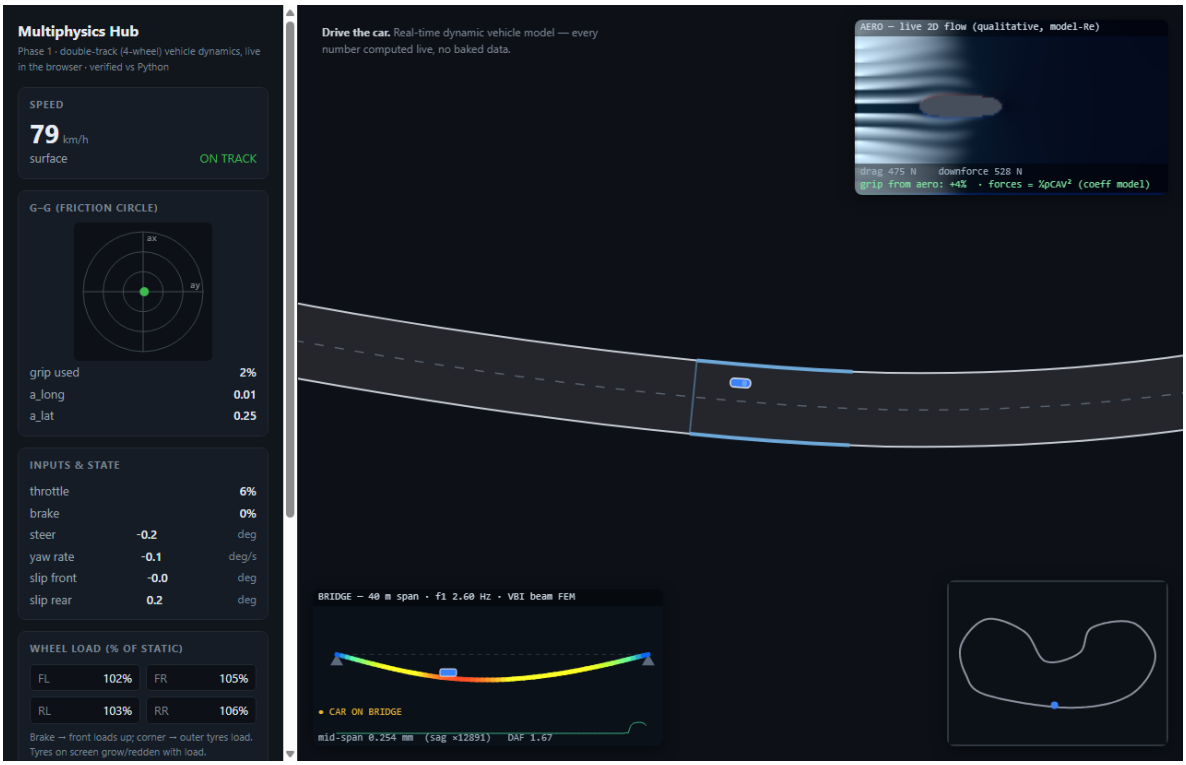


Figure 1: The three physics running together. The car crosses the bridge (centre). The aero panel (top right) shows the 2D wake with the current drag and downforce; the bridge panel (bottom) shows the 40 m span deflecting under the moving load, coloured by bending moment, with $f_1 = 2.60$ Hz and the dynamic amplification factor; the left panels show the friction-circle state and the per-wheel load transfer. Every value is computed in the current frame.

falls as load rises ($\mu_{\text{eff}}(F_z)$ concave), transfer changes each axle's total grip. That is what produces the car's understeer or oversteer balance, set here by the front load-transfer share.

2.2 Aerodynamics: a 2D fluid around the car

A 2D incompressible fluid solver runs around the car silhouette every frame, using Stam's stable-fluids scheme (semi-Lagrangian advection with a Gauss–Seidel pressure projection) and imposing the body by volume penalisation, the same immersed-boundary idea as in the portfolio's Navier–Stokes solver. It draws the wake and, through the standard coefficient relation $F = \frac{1}{2}\rho C A V^2$, returns a drag and a downforce that raise the tyre loads; at 60 m/s the downforce is about 30 % of the car's weight and lifts the rear-grip cap. This panel is labelled qualitative on screen: it is a 2D, model-Reynolds field that shows separation and the wake, not a quantitative aerodynamic prediction. The benchmark-verified staggered-grid solver is the separate vortex-street project.

2.3 Bridge: a moving-load modal beam

The bridge is a modal reduction of the Euler–Bernoulli moving-load beam from the VBI project: the first few mode shapes carry the response, integrated under the travelling wheel force. It deflects live under the car, coloured by bending moment, and reports the dynamic amplification factor. Two couplings are kept physical and quantitative: aerodynamic downforce changes the tyre grip, and the wheel load (including that downforce) deflects the span.

3 Verification

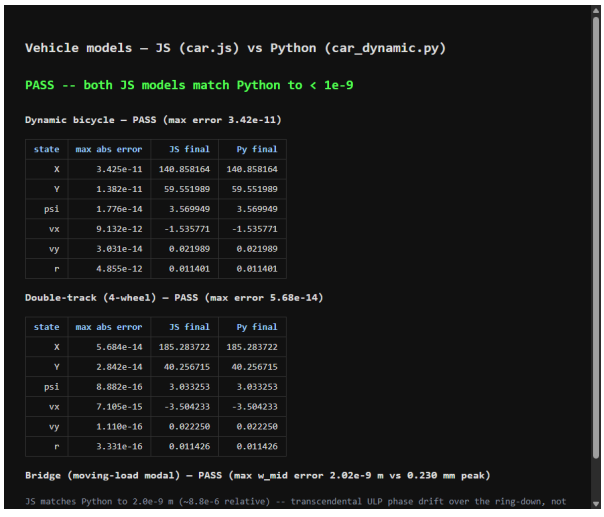
Verification runs on two layers. The numpy models are the reference and are checked against ground truth; the JavaScript ports that run in the browser are then checked against those numpy models on identical inputs. Both integrate at a fixed 250 Hz with RK4, decoupled from the render rate.

Layer 1: each model against a reference. The vehicle is checked against static and steady-state references; the bridge against the closed-form static deflection and against the independent full Euler–Bernoulli FEM eigensolve from the VBI project; the fluid solver against its own conservation properties (Table 1).

Table 1: Each model checked against an analytical or FEM reference.

Check	Reference	Result
Static wheel loads (at rest)	$mg/4$ each	exact
Vertical equilibrium (any a_x, a_y)	$\sum F_z = mg$	exact
Longitudinal / lateral load transfer	$m\lvert a_x\rvert h/L,\ 2s\,m\,a_y h/\mathrm{tr}$	match closed form
Low-speed cornering	Ackermann $\delta \approx L/R$	match
Balance trend	more front transfer $\Rightarrow$ wider R	monotonic
Bridge static mid-span deflection	$PL^3/48EI$	rel. err $< 10^{-3}$
Bridge modal frequencies	full VBI Euler–Bernoulli FEM	within 2 % ($f_1 = 2.60$ Hz)
Bridge DAF (slow / at speed)	$\to 1$ quasi-static; bounded > 1	confirmed
Fluid solver	divergence drops; stable; sheds a wake	confirmed

Layer 2: JavaScript against Python. Each browser model is run against its numpy twin on the same inputs (Fig. 2). The vehicle port agrees with Python to order 10^{-13} over thousands of steps; the bridge crossing agrees to order 10^{-9} m against a 0.23 mm peak deflection, the residual being transcendental round-off accumulated over the ring-down.


Vehicle models — JS (car.js) vs Python (car_dynamic.py)

PASS -- both JS models match Python to < 1e-9

Dynamic bicycle — PASS (max error 3.42e-11)

state	max abs error	JS final	Py final
X	3.425e-11	140.858164	140.858164
Y	1.382e-11	59.551989	59.551989
psi	1.776e-14	3.569949	3.569949
vx	9.132e-12	-1.535771	-1.535771
vy	3.031e-14	0.021989	0.021989
r	4.855e-12	0.011401	0.011401

Double-track (4-wheel) — PASS (max error 5.68e-14)

state	max abs error	JS final	Py final
X	5.684e-14	185.283722	185.283722
Y	2.842e-14	40.256715	40.256715
psi	8.882e-16	3.033253	3.033253
vx	7.105e-15	-3.504233	-3.504233
vy	1.110e-16	0.022250	0.022250
r	3.331e-16	0.011426	0.011426

Bridge (moving-load modal) — PASS (max w_mid error 2.02e-9 m vs 0.230 mm peak)

JS matches Python to 2.0e-9 m (~8.8e-6 relative) -- transcendental ULP phase drift over the ring-down, not

(a) JavaScript vs Python on identical inputs.


Real-time fluid (StableFluid) — self-check

PASS -- projection reduces divergence, stable, sheds a wake

check	value	expected	
mean |div| before projection	1.186e-2		
projection reduces divergence (200 it)	7.52e-3	< 8.9e-3	OK
field stays finite (stable) over 250 steps	1	1	OK
bluff body leaves a low-speed wake (deficit)	0.19	> 0.15	OK
freestream / wake speed (norm.)	0.22 / 0.02		

Honest scope: this browser solver is Stam "Stable Fluids" on a collocated grid — real-time and stable, but its projection leaves a small residual divergence (collocated odd–even decoupling), so it is a qualitative flow, not a verified incompressible solve. The rigorous, staggered-MAC, Ghia/Williamson-verified Navier–Stokes solver is the Python core in pillar2-cfd. The drag/downforce the car actually feels come from the coefficient model (½ρC·V²); this panel is the picture.

(b) Fluid-solver self-check (divergence, stability, wake).

Figure 2: Layer-2 verification: the shipped JavaScript is held to the numpy reference.

4 The application

The page (Fig. 3) opens in a self-driving demo loop and hands control to the user on any drive key. It is a top-down circuit with a live minimap, a friction-circle readout, per-wheel load bars

that grow and redden under transfer, a manual gearbox, handbrake and reverse, and an editable track. It runs entirely client-side: open `index.html`, with no server and no build step, and it is deployed on GitHub Pages.

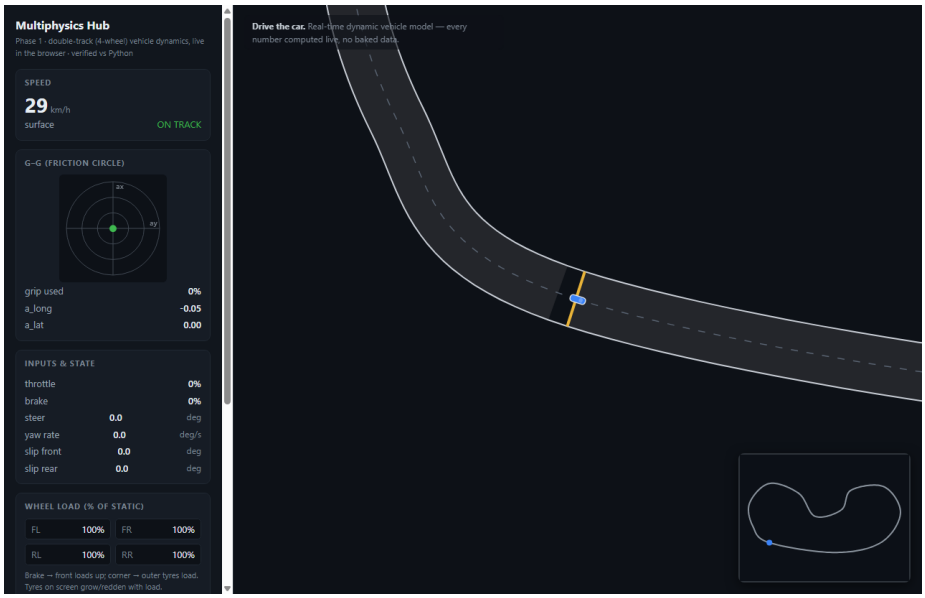


Figure 3: The driving view: friction-circle state, per-wheel load transfer, inputs and state, and the mini-map, all updated each frame from the verified vehicle model.

5 Limitations

- The whole model is two-dimensional.
- The aero field is qualitative (2D, model-Reynolds). It drives the car through standard coefficient relations, not through a quantitative pressure integration; the benchmark-verified solver is the separate vortex-street project.
- The bridge is a modal reduction of a simply-supported span, not a continuous multi-span girder. The gearbox is a driver-input layer on top of the vehicle dynamics.
- Coupling is one-way where it is only illustrative (the aero render) and two-way only where it is physical (downforce to grip, wheel load to deflection).

6 Reproducibility

The models are numpy-only and re-implemented in JavaScript. The checks ship with the code: `verify_car_dynamic.py` and `verify_doubletrack.py` for the vehicle, `verify_bridge.py` for the beam (against $PL^3/48EI$ and the full VBI FEM), `web/xcheck.html` for JavaScript versus Python, and `web/aero_test.html` for the fluid-solver self-check. The application is `web/index.html`; the live build runs on GitHub Pages.

References

[1] H. B. Pacejka, *Tyre and Vehicle Dynamics*, 3rd ed., Butterworth-Heinemann, 2012. (Magic Formula tyre model.)

[2] W. F. Milliken & D. L. Milliken, *Race Car Vehicle Dynamics*, SAE International, 1995. (Load transfer and handling balance.)

[3] J. Stam, "Stable fluids," *Proc. SIGGRAPH*, 1999. (Semi-Lagrangian advection with pressure projection.)

[4] P. Angot, C.-H. Bruneau & P. Fabrie, "A penalization method to take into account obstacles in incompressible viscous flows," *Numerische Mathematik*, 81, 1999.

[5] R. W. Clough & J. Penzien, *Dynamics of Structures*, 3rd ed., Computers & Structures, 2003. (Modal superposition.)

[6] L. Frýba, *Vibration of Solids and Structures under Moving Loads*, Noordhoff, 1972. (Moving-load reference solution.)