Modelling vehicles' air pollution

An intern's progress, and what's next

Mattéo Boissière

Last updated: August 3, 2023

Chapter 1

Introduction

One of the first things I noticed in this internship, after years of being used to engineering school and its pretty physics where most things are related to π or e somewhere down the line is that, well, we were now dealing with cars, these complicated manmade objects where who even knows how one particular model functions under the hood - and so, the values you have to deal with look more like this:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	NFR	Sector	Table	Type	Technolog	Fuel	Abatemer	Region	Pollutant	Value	Unit	CI_lower	CI_upper	Reference							
2	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Gas Turbir	Gas Oil		NA	Hg	0.053	mg/GJ	0.005	0.53	Pulles et al. (2012)							
3	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Gas Turbir	Gaseous Fuels		NA	Cd	0.00025	mg/GJ	8.00E-05	0.00075	Nielsen et al., 2012							
4	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Gas Turbir	Gaseous Fuels		NA	NMVOC	1.6	g/GJ	0.5	7.6	Nielsen et al., 2010							
5	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Gas Turbir	Gaseous Fuels		NA	TSP	0.2	g/GJ	0.05	0.8	BUWAL, 2001							
6	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Gas Turbir	Gaseous Fuels		NA	PM10	0.2	g/GJ	0.05	0.8	BUWAL, 2001							
7	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Gas Turbir	Gaseous Fuels		NA	Benzo(k)f	1.11	µg/GJ	0.4	3.3	API, 1998							
8	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Gas Turbir	Gaseous Fuels		NA	Hg	0.1	mg/GJ	0.01	1	Nielsen et al., 2010							
9	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Large stati	Gas Oil		NA	Benzo(k)f	0.0987	mg/GJ	0.0493	0.0987	US EPA (1996), chapter 3.3 ("Less than" value based on method detection limits)							
10	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Large stati	Gas Oil		NA	Indeno(1,	0.187	mg/GJ	0.0937	0.187	US EPA (1996), chapter 3.3 ("Less than" value based on method detection limits)							
11	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Large stati	Gas Oil		NA	Zn	1.81	mg/GJ	0.18	18.1	US EPA (1998), chapter 1.3							
12	1.A.1.a	Public ele	Table_3-2	Tier 2 emi	Stationary	Natural gas		NA	NMVOC	89	g/GJ	45	135	Nielsen et al., 2010							
13	1.A.1.a	Public ele	Table_3-2	Tier 2 emi	Stationary	Natural gas		NA	PM2.5	2	g/GJ	1	3	BUWAL, 2001							
14	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Large stati	Gas Oil		NA	NMVOC	37.1	g/GJ	18.5	55.6	US EPA (1996), chapter 3.3							
15	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Hard Coal		NA	Benzo(b)f	37	µg/GJ	3.7	370	Wenborn et al., 1999							
16	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Hard Coal		NA	HCB	6.7	µg/GJ	2.2	20.1	Grochowalski & Konieczyński, 2008							
17	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Hard Coal		NA	Benzo(b)f	29	µg/GJ	2.9	290	Wenborn et al., 1999							
18	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Gas Turbir	Gaseous Fuels		NA	NOx	153	g/GJ	92	245	US EPA (2000), chapter 3.1							
19	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Hard Coal		NA	Hg	1.4	mg/GJ	0.97	2.25	US EPA (1998), chapter 1.1							
20	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Hard Coal		NA	Benzo(a)p	0.736	µg/GJ	0.245	2.21	US EPA (1998), chapter 1.1							
21	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Hard Coal		NA	Cu	9	mg/GJ	0.23	15.5	Expert judgement derived from Guidebook (2006)							
22	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Hard Coal		NA	Zn	90	mg/GJ	0.39	155	Expert judgement derived from Guidebook (2006)							
23	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Hard Coal		NA	Cu	9	mg/GJ	0.23	15.5	Expert judgement derived from Guidebook (2006)							
24	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Brown Coal		NA	CO	13	g/GJ	0.1	26	European Commission (2006)							
25	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Brown Coal		NA	Cr	9.1	mg/GJ	6.55	15.3	US EPA (1998), chapter 1.7							
26	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Brown Coal		NA	PCDD/F	10	ng I-TEQ/(	5	15	UNEP (2005); Coal fired power boilers							
27	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Brown Coal		NA	Benzo(a)p	1.3	µg/GJ	0.26	6.5	US EPA (1998), chapter 1.7							
28	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Brown Coal		NA	Indeno(1,	2.1	µg/GJ	0.4	10.5	US EPA (1998), chapter 1.7							
29	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Brown Coal		NA	Zn	8.8	mg/GJ	0.5	16.8	Expert judgement derived from Guidebook (2006)							
30	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Brown Coal		NA	SOx	1680	g/GJ	330	5000	See note in Guidebook chapter							
31	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Brown Coal		NA	TSP	10.2	g/GJ	3.4	30.6	US EPA (1998), chapter 1.7							
32	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Brown Coal		NA	PM10	6.9	g/GJ	2.3	20.7	US EPA (1998), chapter 1.7							
33	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Brown Coal		NA	As	14.3	mg/GJ	10.3	24.1	US EPA (1998), chapter 1.7							
34	1.A.1.a	Public ele	Table_3-1	Tier 2 emi	Fluid Bed	Brown Coal		NA	PM2.5	2.8	g/GJ	0.9	8.4	US EPA (1998), chapter 1.7							
35	1.A.1.a	Public ele	Table_3-6	Tier 1 emi	NA	Gas oil		NA	BC	33.5	% of PM2.	28.9	38	Hildemann et al., 1981 & Bond et al., 2006							
36	1.A.1.a	Public ele	Table_3-5	Tier 1 emi	NA	Heavy Fuel Oil		NA	Ni	273	mg/GJ	140	550	US EPA (1998), chapter 1.3							
37	1.A.1.a	Public ele	Table_3-4	Tier 1 emi	NA	Gaseous fuels		NA	PCDD/F	0.5	ng I-TEQ/(	0.25	0.75	UNEP (2005); Light fuel oil/natural gas fired powe							

...and so I understood why statistics was a required course in school.

In the report that follows, I will try to detail as best I can the thoughts I've had during this internship - the work I've done, and the work I've haven't done - in the hopes that this can be useful to anyone who'd be interested in picking up the project later on.

We will review the **literature** that was useful for comprehension of the project's theoretical aspects, but also a review of the **software** that helped shape the architecture, and why it is what it is. Along the way, we will see that many great **questions** lay unsolved, and could potentially serve as direction for interesting articles - lacking time, however, I could only lay the groundwork for such topics to be covered!

Contents

Chapter 2

Methodology

2.1 On the role of ChatGPT/Copilot while learning code

ChatGPT is a tool I'm sure you know by now, and for good reason: it's very good at **putting into code, an idea you can only express with words**:

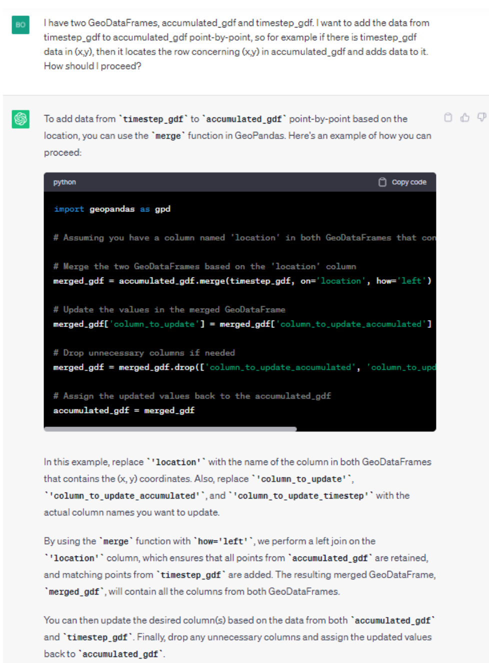


Figure 2.1: ChatGPT can help you find appropriate code...

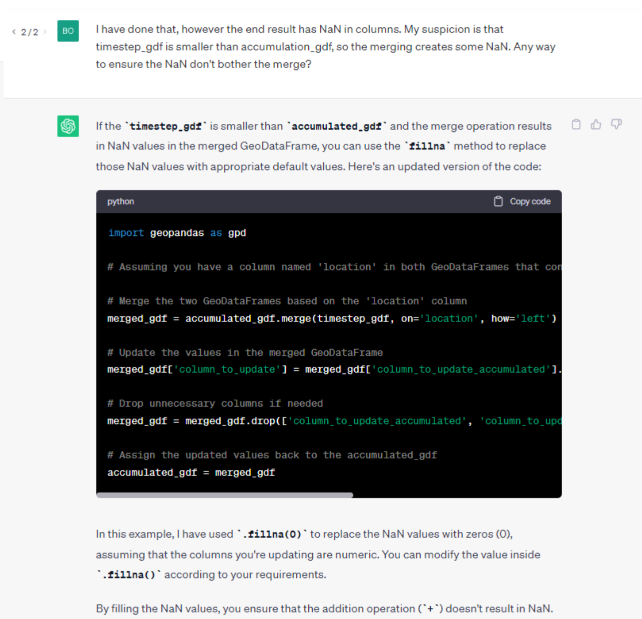


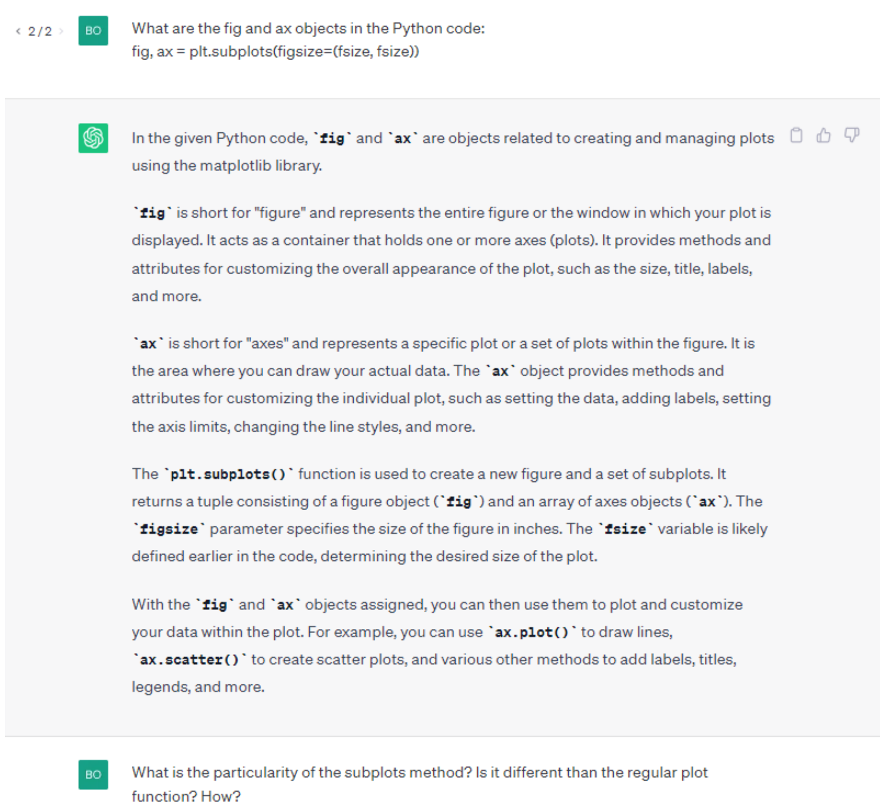
Figure 2.2: ...and help with debugging if it fails!

ChatGPT is pretty much everywhere in the news, in recent codebases, and it could even be... in this very report???

Joking aside: I have found ChatGPT to be extremely valuable for **learning new Python modules**: since programming is best learned by trying your hand at it and debugging whatever horrible crap you've made, it has significantly sped up my learning process compared to when I coded without it.

But! There is a procedure to it. I would highly recommend treating ChatGPT kinda like a human you can bother on demand: **wise, but still very much capable of making mistakes.** To avoid falling into the trap of making boilerplate code nobody understands *(and least of all the AI)*, I've found it best to stay clear in your demands, focused, and critical of what the algorithm will give you.

ChatGPT personally scratched many itches for me, along the lines of *"I don't know what this silly line does in matplotlib but I've been coding for a while so at this point I'm too afraid to ask"*, and that is why I recommend using it.



But it is of course not meant to scratch *"do all my work for me please"* itches. And most important of all in this here field of research: GPT answers can absolutely fake data, and should NOT be treated like absolute truth. GPT can be good at clearing the terrain, perhaps, if you are completely lost in a field and would like to know how to get started - but any result that could end up being used somewhere should be double-checked with a reliable source.

And careful not to forget: Google and StackOverflow haven't stopped existing, and in case of doubt, documentation reigns supreme. *(even if sometimes documentation is very poorly written and it's really damn useful to have something that can vulgarize it, I'M LOOKING AT YOU FENICS)*

(Oh, and GitHub also has a tool called Copilot where a GPT is built in your IDE, which means it has access to all of the context in your code and how it interacts! I was not personally able to get it to work, because it checks your location if you want an academic license, and it didn't like that I was in Sweden with the email of a French school. But it is supposed to have a similar effect, if not to say an even more efficient one!)

2.2 On tutorials

Originally, I was inspired by QGIS' very detailed tutorials, and wanted to make a series of Jupyter notebooks that would interactively explain, even to someone with minimal prior knowledge, how everything in the project works. Because in case you haven't figured, on page 5 of this report where I haven't even started to talk about the project itself, *I like explaining things*.

The idea was to take the reader through the reasoning of the modelling, with tests and tries to further an approach from simple to complicated - because if the reasoning is captured and explained, it can be great for learning. For example, an initial idea I've had for a series of tutorials was:

- Only assuming a working Python environment, how do we configure a traffic simulation tool?
- Let's just try a random simulation (and here's how the random works)
- Now, let's try to change the data to be closer to real life.
- What is the emission model?
- What if we are curious about emission models and their differences? *(HBEFA, COPERT, CAL3QHCR...)*
- And what are EURO norms? Can we implement them manually?
- And how does the pollution change when the fleet changes?

And so on. Sadly, it was time-consuming enough to get a simple version of the code to work. Besides, I've come to realize that in the early phases of a project, it's difficult to even know where you're headed, and your direction might drastically change from one day to another. So, making tutorials as a way of advancing when you're not even done, is not as smart as it seems compared to doing it after everything is finished.

2.3 On detailing the setting of our research

I have tried, as much as possible, to keep underlying hypotheses in mind when exploring new concepts, hopefully that will show in later chapters. I have honestly done it perhaps too much, freezing myself from advancing at times, since I'm better at finding shortcomings of an idea than just executing it. But I still encourage the practice - if data is collected for example, I think it's important to know how it was collected exactly. And at least the reasoning should be rigorous!

I will admit that the literature review will be a bit rushed, for the following reason: attempting to do a literature review in the first months left me very confused, and it took me about the whole time of my internship to slowly understand the field and gain just enough perspective to... actually be able to do a halfway decent literature review, but by the time it's too late. Still, I loved this internship very much and I'm very glad I got to learn so much about research! But now I understand what people mean when they say that 5 months is unforgivingly short for a research internship...

2.4 On keeping a record of everything

Having too often found dead URLs in my research, I will try to keep a PDF copy of any website I might refer to in this report, just in case. I will also try to keep versions of software used at time of writing whenever possible.

On citing works with a DOI/ISBN : the citation will link to a complete reference, in the Bibliography section at the end of the document.

On citing works with no DOI/ISBN : the citation will link to a quick reference, at the bottom of the page. In my "Literature gathered" folder, I have included a "By order of presentation" folder, where you will find PDFs of the works cited.

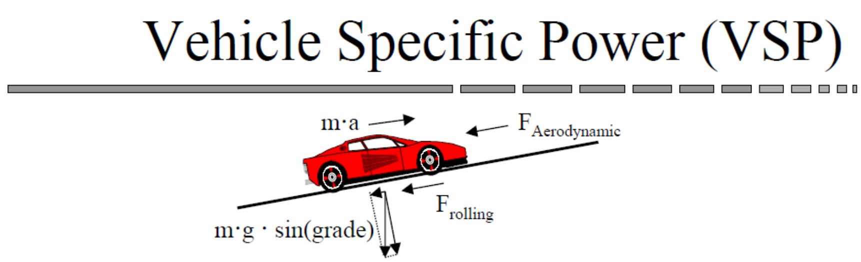
Chapter 3

Literature review, and its implementation in the theoretical model

3.1 Current theoretical model

3.1.1 Vehicle specific power: definition

There are a huge amount of variables needed to describe a vehicle's movement and so it is super annoying to model them. But around 1999, MIT Ph.D. student Jose-Luis Jimenez said *"me basically, I'm just built different"*[1] and proposed a new variable: **Vehicle Specific Power**.



Defined as power over mass, the derivation of VSP is as follows:

$$VSP = \frac{\frac{d}{dt}(E_{kinetic} + E_{potential}) + F_{rolling} \cdot v + F_{aerodynamic} \cdot v}{m} \tag{3.1}$$

with v the speed of the vehicle in m/s, and:

- $E_{kinetic} = \frac{1}{2}m \cdot (1 + \epsilon_i) \cdot v^2$
  m is the **vehicle mass**, but we adjust the mass we're studying by ϵ_i, the **mass factor**. This is because the rotating components *(wheels, gears, shifts...)* have an effect on the vehicle's inertia. The suffix i indicates that ϵ_i depends on whether or not the vehicle is in first, second, third gear...
- $E_{potential} = mgh$ with g the acceleration of gravity, and h the altitude of the vehicle.
- $F_{rolling} = C_R \cdot mg$ with C_R the dimensionless coefficient of rolling resistance.
- $F_{aerodynamic} = \frac{1}{2}\rho_a \frac{C_D \cdot A}{m}(v - v_W)^2$ with ρ_a ambient air density, C_D dimensionless drag coefficient, A frontal area of the vehicle, and v_W the wind speed[2].

[1] Note: he (probably) didn't actually say that

[2] Jimenez actually writes $v + v_W$ with v_W the headwind into the vehicle, this makes it less clear that v_W is often a negative value, and what we're looking at is really the speed of vehicle relative to the wind. Thus, I allowed myself to change the notation for clarity.

Calculations give the following formula :

$$VSP = v \cdot (a \cdot (1+\epsilon_i) + g \cdot \sin(\arctan(grade)) + g \cdot C_R) + \frac{1}{2}\rho_a \frac{C_D \cdot A}{m}(v - v_W)^2 \cdot v \tag{3.2}$$

with a the acceleration of the vehicle in m/s^2, and *grade* the ratio of vertical rise over horizontal distance. Vehicle Specific Power is in W/kg, but it is commonly expressed in kW/Metric Ton.

One thing that can catch you off guard is that much of Jimenez's thesis, and much of the literature that cites it, write *grade* as a percentage (%), and that's confusing for two reasons:

- This doesn't make it necessarily crystal clear that *grade* can be negative, even though it definitely can be. Some literature will make an effort and write "grade (±%)", though.
- Well, teeeechnically as a ratio of two lengths it could be superior to 1, couldn't it? It's not explained at all, but my guess is that, well, what kind of road has a slope so steep that rise is larger than horizontal distance?? Although physically possible, we just know we'd never run into it. Anyway, we will do some trigonometry later on, and things should make some more sense.

Jimenez showed his work with a presentation[1] during a vehicle emissions workshop *(which is the most digestible for a beginner)* and later during his Ph.D. thesis [2]. The original presentation actually had an additional term for internal friction, but in the thesis Jimenez decided against it, arguing that *"engine friction is generally small compared to the range of values of Specific Power"*, and that engine friction *"estimation from roadside measurements is much more difficult and uncertain"*.
To Jimenez, the advantages of using VSP as a variable to study vehicle emissions are many:

- The end formula is *roadside-measurable*, which basically means simple to measure directly, whereas previous models needed hard to find data such as engine speed, absolute power, throttle position, etc.
- Citing a previous article that showed emissions are mainly related to engine power and engine speed[2], Jimenez argues that, **although VSP doesn't perfectly capture all effects that might play a role on emissions**, it captures a significant part.

3.1.2 Vehicle specific power: simplified form

This is the part where we want to get a simplified formula for typical vehicles. And this is what I mean when I say we should be careful with hypotheses, it took me several months to realize the VSP formulae that follow aren't a definition but rather a simplification of the definition in specific (yet admittedly common) cases!
This is also where approaches will differ, especially depending on the type of vehicle. Jimenez's choice of "typical values for all parameters" leads to the following formula:

$$VSP = v \cdot (1.1 \cdot a + 9.81 \cdot grade + 0.132) + 3.02 \cdot 10^4 \cdot (v + v_W)^2 \cdot v \tag{3.3}$$

A slight variant formula has been adopted into the methodology of the US Environment Protection Agency as of 2002 and later cited in many works as USEPA 2002[3] or Frey et al. 2002[4]. Before we dive into that, Jimenez writes the following, concerning his simplifications:

- *"Note that these expressions are based on average values of the rolling resistance coefficient, the aerodynamic drag term coefficient ($C_D * A/m$), and the value of air density at 20°C (68°F).* ***Better estimates of the actual values of these parameters should be used whenever possible."***

[1] J. L. Jiménez, VSP origin presentation, 1999
[2] J. L. Jiménez, VSP detailed thesis, 1999 - definition of VSP page 54
[3] US EPA, Onboard Emissions Analysis, 2002
[4] According to Coelho et al. 2009[4], Frey et al. 2002 is Publication EPA420-R-02-027 which I have left in the folder, but also in that paper they cite Frey et al. 2002 and I can't really find Frey et al. 2002? I don't know what to believe anymore Mom please pick me up I'm scared

- *"The ideal gas law ($Pv/T = P/(\rho T) = constant$) can be used to correct to other temperature and pressure conditions. The formula is $\rho = \rho_0 * (P/P_0) * (T_0/T) = 1.207 * (P/101.33) * (293.16/T)$, with ρ in kg/m^3, P in kiloPascals (kPa), and T in Kelvin (K).* ***This correction may be important, e.g. if the measurements are performed at -10°C (-14°F) and 1 atmosphere the air density will be 10% higher than at 20°C."***
- *Rigorously cos(atan(grade)) should be used instead of grade, but the error of this approximation is small (less than 1% relative error for grades below 11%).*

3.1.3 This is the part where I go crazy because I trusted the geometrical computations so far and now I don't know anything anymore

It is interesting to see how the EPA have adapted his formula. For light-duty vehicles *(which is a fancy way of saying basically cars)*, here's what the EPA writes :

$$VSP_{light-duty} = v \cdot (1.1 \cdot a + 9.81 \cdot arctan(sin(grade)) + 0.132) + 0.000302 \cdot v^3 \tag{3.4}$$

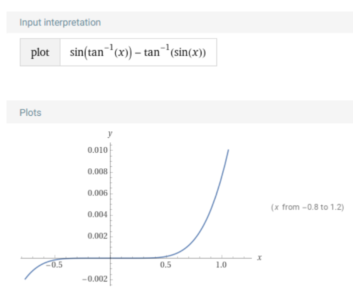
And here is the formula that many papers use, one of them being Fernandes 2017[5], where they cite the 2002 EPA report as source:

$$VSP_{light-duty} = v \cdot (1.1 \cdot a + 9.81 \cdot sin(arctan(grade)) + 0.132) + 0.000302 \cdot v^3 \tag{3.5}$$

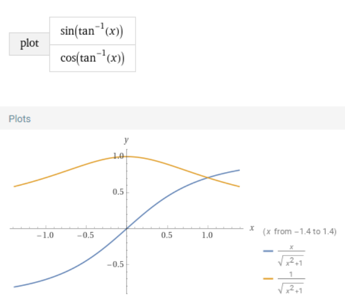
For buses, the mass factor and resistance coefficients chance, and the EPA gives the following formula, again in USEPA 2002:

$$VSP_{bus} = v \cdot (a + 9.81 \cdot sin(grade) + 0.0094) + 0.42 \cdot v^3 \tag{3.6}$$

Now, we can immediately see that from here on out, the speed of the headwind that goes into the vehicle is neglected, which is understandable enough. But another observation we could make is that they write arctan(sin(grade)) in the EPA report, but it is never explained and never reused in later papers. Granted, sin arctan and arctan sin are similar functions, but they are not the same, so it seems to me like a mistake...



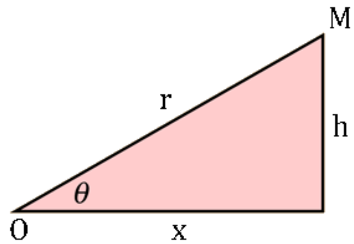
And even weirder that, although not a huge deal since Jimenez considers it to be negligible, either I'm bad at reading Jimenez's footnotes *(which is very much possible)* or everyone seems to have confused sin arctan for cos arctan, which are NOT the same function. So what's up? Did Jimenez / the PA publish an erratum later on or has just everyone ever been quoting papers wrong this whole time?? I hope not.



I will attempt to do the calculations myself. As a reminder, we're supposed to get :

$$VSP_{\text{potential part}} = \frac{\frac{d}{dt}E_{potential}}{m} = g\frac{dh}{dt} = v \cdot g \cdot sin(arctan(grade))$$

where h is altitude from sea level. Suppose in the following diagram that, at instant t, the car is at $M(t)$, and so r, h and x are also functions of time. $\theta(t)$ can also be, too, since the slope can change and not be an exact triangle. ***(Wait, if a road is curved, is it even valid to attempt a proof with this kind of "infinitesimal triangles" kind of logic? I honestly am not sure.)***



First off, it's common for slopes to be given in degrees, so let's just say we have ϕ the angle in degrees, then let's remember that the formula would be: $\theta = \frac{\pi}{180} \cdot \phi$.
(This paragraph exists because I'm not quite sure if Kaveh's CSVs are in degrees or radians. Not that it really matters because we've put everything at 0, but uh, just in case. The code could use some double-checking.)

Now then. It seems to me that, if grade is defined as the ratio of vertical rise over horizontal distance, then it's just $\frac{h}{x}$, which some of you may recognize as $tan(\theta)$. So, okay, $\theta = arctan(grade)$. Then, potential energy would be the $mgh = mgr \cdot sin(\theta)$.

Okay, I think I see what's going on.

- If we assume θ NOT to vary with time, then $g\frac{dh}{dt} = g \cdot \frac{dr}{dt} \cdot sin(\theta) = v \cdot g \cdot sin(arctan(grade))$ *(hurray!)*
- If we assume $\theta(t)$ to vary with time (which it absolutely can!), then I feel like we have:

$$\begin{aligned}
VSP_{\text{potential}} &= g\frac{dh}{dt} = g\left(\frac{dr}{dt}\sin(\theta) + r \cdot \frac{d(\sin(\theta))}{dt}\right) \\
&= v \cdot g \cdot \sin(\theta) + g \cdot r \cdot \frac{d\theta}{dt} \cdot \cos(\theta) \\
&= v \cdot g \cdot \sin(\arctan(\text{grade})) + g \cdot r \cdot \frac{d\,\arctan(\text{grade})}{dt}\cos(\arctan(\text{grade})) \\
VSP_{\text{potential}} &= VSP_{\text{potential, static }\theta} + g \cdot r(t) \cdot \frac{1}{1+\text{grade}^2(t)} \cdot \frac{d\text{grade}}{dt}(t) \cdot \cos(\arctan(\text{grade}(t)))
\end{aligned} \tag{3.7}$$

I... don't think I've ever seen this formula everywhere? And maybe this can be simplified even further but I'm not too sure how? I get that this can be negligible, if we assume flat surfaces everytime, then yes, everything is equal to zero, and it's whatever. And if we assume constant grade, then yes, the first formula is correct, and it's whatever. But I don't see this precision in papers, I see $9.81 \cdot arctan(sin(grade))$ confidently, and I feel like the discrepancies between papers here and there and Jimenez's hasty footnotes are kind of what I've just done, but more confusedly. Which is also equal to zero if we assume flat surfaces everytime anyway. But that's not what they say. So what's the big deal.

Anyway... I still have a lot of things I want to write so let's move on. So, we're at page 10 and all we have so far is a VSP. What do we do with it now ?

3.1.4 VSP, but make it discrete

By all logical means, if pollution is dependant on VSP *(and most likely other random phenomena but unless you want to die of exhaustion let's listen to Jimenez and assume we can neglect those)*, then surely, for each type of vehicle there is, a certain VSP will give a certain amount of pollution.

Figure 3.1: Clio 2 Campus Bye bye version 1.2 16V de 75 ch finition *Authentique*

And so we can figure out, say, the function :

$$f_{\text{Clio 2 Campus Bye Bye}} : \begin{cases} \mathbb{R} \to \mathbb{R} \\ VSP \mapsto \text{emitted pollutant of choice} \end{cases}$$

Great, so who wants to take an infinity of measures to account for every possible value of VSP? Yeah, me neither.

So, the EPA decided to opt for a binning approach, dividing good old continuous VSP into 14 discrete "VSP modes" - and again many later papers have followed suit. If your inner statistician is wondering how exactly they have determined these 14 modes to be the most relevant, there is more than enough to read in the famed EPA420-R-02-027 report. But the modes are as follows:

VSP Mode	Definition
1	$VSP < -2$
2	$-2 \leq VSP < 0$
3	$0 \leq VSP < 1$
4	$1 \leq VSP < 4$
5	$4 \leq VSP < 7$
6	$7 \leq VSP < 10$
7	$10 \leq VSP < 13$
8	$13 \leq VSP < 16$
9	$16 \leq VSP < 19$
10	$19 \leq VSP < 23$
11	$23 \leq VSP < 28$
12	$28 \leq VSP < 33$
13	$33 \leq VSP < 39$
14	$39 \leq VSP$

Now, once we have estimated the VSP mode of a given vehicle at a given time, how do we get an idea of how much it's polluting?

3.1.5 Hey let's make measurements of, like, what vehicles emit according to their VSP maybe

We're getting there! So, the year is 2004, and the US EPA releases MOVES[5]. And it's pretty great! As we've been leading up to so far, the idea is that, now that we've put driving patterns in different modes, **we can test many vehicles to operate in that pattern and obtain the average pollution rate in g/s for a given pollutant** *(CO, NOx, HC)*. A more detailed explanation can be found in a 2010 overview of the model[6].

This is an example of something called an **emission factor model**, very abundant due to its simplicity: now that we choose to consider that the pollution of a given vehicle in a given mode will follow a certain fixed rate *(in grams per second)*, all we have to do is add those factors at each timestep of a simulation to obtain the pattern of pollution.[0]

This figure is from a 2006 paper by Coelho et al.[3], it is actually based on table A-4 of EPA's aforementioned 2002 report, Publication EPA420-R-02-027.

Table 2
Mean Values for NO_x, HC, CO_2 and CO emissions (g/s) for VSP modes for vehicles with odometer reading < 50 000 miles and engine displacement < 3.5 L

Definition	Pollutant			
	NO_x	HC	CO_2	CO
$VSP^a < -2$	0.0009	0.0004	1.6711	0.0078
$-2 \leqslant VSP < 0$	0.0006	0.0003	1.4580	0.0039
$0 \leqslant VSP < 1$	0.0003	0.0004	1.1354	0.0033
$1 \leqslant VSP < 4$	0.0012	0.0004	2.2333	0.0083
$4 \leqslant VSP < 7$	0.0017	0.0005	2.9199	0.0110
$7 \leqslant VSP < 10$	0.0024	0.0007	3.5253	0.0170
$10 \leqslant VSP < 13$	0.0031	0.0008	4.1075	0.0200
$13 \leqslant VSP < 16$	0.0042	0.0010	4.6350	0.0292
$16 \leqslant VSP < 19$	0.0051	0.0011	5.1607	0.0355
$19 \leqslant VSP < 23$	0.0059	0.0014	5.6325	0.0551
$23 \leqslant VSP < 28$	0.0076	0.0021	6.5348	0.1138
$28 \leqslant VSP < 33$	0.0121	0.0034	7.5852	0.2076
$33 \leqslant VSP < 39$	0.0155	0.0049	9.0242	0.4418
$39 \leqslant VSP$	0.0179	0.0109	10.0884	0.8823

Source: North Carolina State University, 2002.
[a] As computed by Eq. (5).

And this is what my code has been based on! Recall that all that is needed to calculate VSP *(and thus VSP mode)* is speed, acceleration, and slope, all of which I have! That is, **assuming dimensions of a standard light-duty vehicle** *(which... it took me several months to realize I was actually reviewing mostly US literature instead of European literature. And I would wager that personal vehicles are larger in the US, actually...* ***Maybe, in order to be thorough, we could recalculate usual resistance coefficients with a Swedish fleet.***

[5]Publication EPA420-P-04-019, 2004. Actually not really a research paper, but the draft(!) of a user guide for a software implementation of the MOVES model. Didn't stop it from being cited plenty!
[6]Koupal et al., MOVES overview, 2010
[0]and if it was in grams per unit of distance, then we'd have to multiply by distance... This is the methodology most carbon footprint calculation follows actually! But we'll come back to that.

3.2 Literature review, beyond the model I used

There are plenty of different vehicle models and if you're reading this then I didn't have the time to do a solid literature review of them, but hopefully there is a solid amount of literature on it that I left you in the folder. **I find López-Martínez 2017[6] and Albuquerque et al. 2020[1] to be a particularly good start to get a wide overview of tools already developed**, their advantages, their shortcomings, etc, up to 2020 at least.

3.2.1 Some quick possible developments

Diesel v Gasoline

Coelho et al. actually have a 2009 paper[4] about how MOVES 2004 mostly covers gasoline vehicles, and so they wonder just how much the VSP method developped for "LDGVs" *(Light-Duty Gasoline Vehicles)* is transferable to "LDDVs" *(Light-Duty Diesel Vehicles)*.

Table 4
VSP normalized average emission rates for CO_2, CO, NO_x and HC by VSP mode between EURO III diesel Skoda Octavia TDI 1.9L (D) and EURO IV gasoline VW Polo 1.4 L (G).

VSP mode	Pollutant							
	CO_2 (g/s)		CO (mg/s)		NO_x (mg/s)		HC (mg/s)	
	G	D	G	D	G	D	G	D
1	0.63	0.21	0.50	0.03	0.23	1.29	0.03	0.14
2	1.05	0.61	0.27	0.07	0.68	2.62	0.03	0.11
3	1.02	0.73	0.15	0.14	0.60	3.38	0.03	0.11
4	2.07	1.50	0.58	0.25	1.75	6.05	0.07	0.17
5	2.79	2.34	1.14	0.29	2.52	9.36	0.10	0.20
6	3.47	3.29	1.76	0.69	3.34	12.53	0.15	0.23
7	4.31	4.20	4.05	0.58	4.04	15.48	0.22	0.24
8	5.19	4.94	6.13	0.64	2.63	17.82	0.35	0.23
9	5.81	5.57	9.06	0.61	3.51	21.32	0.42	0.24
10	6.43	6.26	18.97	1.01	2.89	32.53	0.52	0.28
11	7.37	7.40	44.98	1.15	1.27	55.75	0.62	0.37

For some reason I haven't been able to figure out, though, the VSP mode stops at 11 instead of 14.

3.2.2 Does it always have to be emission factor models?

Not at all! They're handy for several reasons, but definitely not an end-all be-all model depending on the situation we'r emodelling. A 2013 paper[7] by Matjaz Knez makes a pretty good inventory of what types of modeling there are, and writes the following:

> *"Emission factor models function with a simple calculation method and do not require large amounts of input data. The estimation of the emissions is expressed by the use of an emissions factor related to one type of vehicle and a specific driving mode (i.e. urban, rural or motorway). Emission factors are derived from the mean values of repeated measurements over a particular driving cycle and are usually expressed in mass of pollutant per unit distance. Emission factor models are commonly used in the development of national and regional emission inventories. This approach is not accurate on microscale, regarding the emission factors are based on average driving characteristics."*

3.2.3 Is VSP always a thing?

VSP is mainly the US EPA's thing, really, and was central to the development of their MOVES model. But there are many models that don't necessarily use VSP!

And this is where I'm proudest of my code, and the section that follows is all about that. The gist of it is, **procedures and objects are kept separate and cleanly organized, then called by a central script**. This means that, if you want to keep the heart of the code, but change the model used to study emissions, it should only take modifying of a few lines here and there!

[7] M. Knez, Vehicular emission models, 2013

Chapter 4

Software review, and its implementation in the project architecture

4.1 Current project architecture

4.1.1 Using Git for version history

Branches

Git is often said to be a powerful tool for working on code collaboratively: developers can work at the same time on seperate features by creating different branches and merging them. It's weird to get the hang of if you're not used to it, but it can prove damn useful. But not if you're coding by yourself, right? Surely if you're coding by yourself it's just as fine to edit everything on one single file, maybe create some duplicate versions if you're unsure about yourself, but no need to go through this git commit / git branch nonsense, right? **WRONG.**

I did manual versioning when I started, and it quickly proved to be annoying. All it takes is to start working on a feature from a stable version, and then later have an idea for another feature that works fine and makes your job easier, but then feature n°1 is ruining everything and needs debugging, so you get back to a stable version for testing, but the stable version is also missing feature n°2 that works just fine, to make you go *"uuurgghhh okay I wish I could work on these things separately actually"*. Which is exactly what branches are for!

Commits

Good Git etiquette requires that, everytime you've made a minor adjustment to your code and you'd like to save your changes, you write the following lines:

```
git add .\src\ [1]
git commit -m "Descriptive message of what you just did"
git pull
git push [2]
```

Doing this allows for one additional godsend methodology: **as long as you've committed your changes, it's fine, you can always go back**. Let's say for example you've messed everything up tremendously and need to go back quite a few commits. This is where the command `git reflog` comes in. Typing it in the terminal will give you the following output:

[1] Or any other file you want to save. Remember that tab autocompletion is a thing!

[2] Assuming you've configured local and remote repositories correctly, which is its own bag of worms but still worth it in the long run.

```
$ git reflog
4423af3 (HEAD -> main) HEAD@{0}: commit: Some commit message
a8e25c9 HEAD@{1}: commit (merge): Merge branch 'development'
edca205 HEAD@{2}: commit: [THE COMMIT THAT RUINED EVERYTHING]
f34e91a HEAD@{3}: commit: Implement new feature XYZ
1f23684 HEAD@{4}: commit: Fix bug ABC
98e9034 HEAD@{5}: commit: Refactor code
c548ef2 HEAD@{6}: commit: Initial commit
```

See how those commits have hash codes ? That list is from most recent to least recent, so we'd like to go back to commit f34e91a, right? All we need to do now is :

```
$ git checkout -b newbranch f34e91a
```

This will create a new branch called "newbranch" that will have the code in its exact state at time of commit f34e91a. And then you can merge it back to main, add whatever features you want more delicately, and it's all good!

4.1.2 Dependencies

Many Python modules are needed to make the project work, and the complete list can be found in requirements.txt. As is written in the README, the command:

```
pip install - requirements.txt
```

should install everything that is needed. Let's go through them quickly[1]:

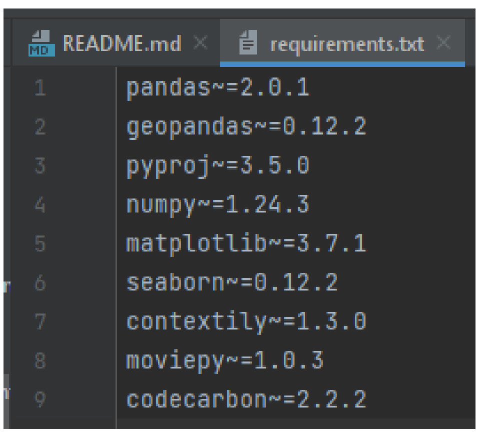


Figure 4.1: Modules and their versions, at time of writing. **Python 3.11** was used during internship.

- **Pandas** is a well-known and powerful module for handling complex data sheets/CSVs within Python.
- **GeoPandas** builds on top of Pandas for handling specifically geographical data.
- **Pyproj** is a cartographic projections library, which has been used for a CRS reprojection operation within the code.
- **Numpy** is a well-known and powerful module for handling complex numerical operations within Python.
- **Matplotlib** is a well-known and powerful module whose "**pyplot**" library is extremely useful for plotting results within Python.
- **Seaborn** is a data visualization module that builds on top of matplotlib and is especially made to be suited for operating with Pandas dataframes.
- **Contextily** is a tile map library, which has been used for placing a portion of OpenStreetMaps in the background of a geographical Matplotlib plot.
- **CodeCarbon** is a Python module that computes carbon emissions due to the power given to the computer to run the program.

CodeCarbon is mostly a small, experimental feature that I found fun to learn about. It can be enabled or disabled at will in the program. When enabled, it will print additional logs to the console and produce an Excel in the output folder with emissions in kgCO2eq.

CO2 emissions due to computation are mostly an issue in 1) **countries with a carbonated electricity mix**, which isn't the case here, and 2) **programs that require heavy computing**, such as PDE solving and AI training, which isn't the case here either. But it's still interesting!

[1] I'm starting to think I don't actually know the meaning of that word

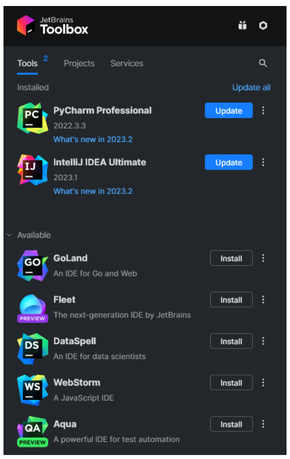
4.1.3 Necessity of an Internet connection

Once you have cloned the Git, most of the work is done locally, except for two actions that require an internet connection:

- Retrieving a background map for display *(called "basemap")* from OpenStreetMaps
- If CodeCarbon is enabled, a location check to adjust the electricity mix to that of the current country.

Perhaps some work could be done to make this work offline, but that's not a priority.

4.1.4 IDE recommendations and workflow



An IDE (Integrated Development Environment) is a software with features to help develop code more efficiently. It's much, much better than just using IDLE!

Visual Studio Code is a solid all-around choice, but the **JetBrains Toolbox** has great specialized tools for a specific language *(PyCharm for Python, IntelliJ for Java, etc etc...)*.

There are free versions, but it is possible to get an **academic license** for the paying "professional" versions with extra features: I used my @student-cs.fr email adress to confirm that I was a student for example, and it should be accessible to members of KTH as well.

Here's some among many examples of workflow can be improved with an IDE *(following examples are from PyCharm, but probably exist elsewhere.)*:

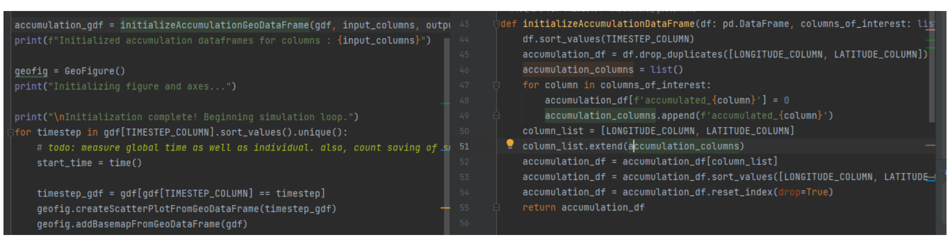


Figure 4.2: Doing `ctrl+click` on a method being called will jump to where it is defined

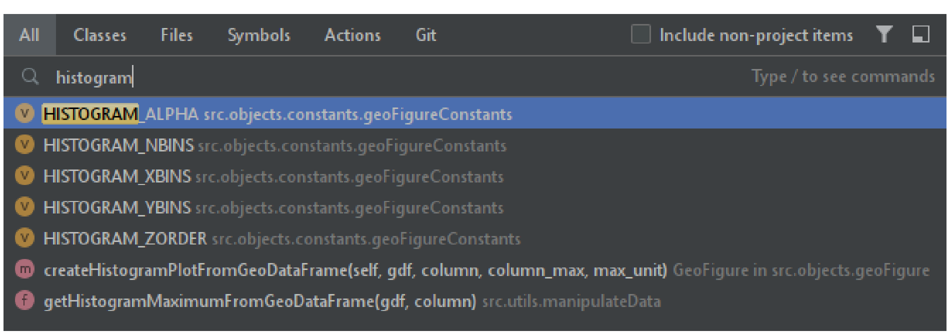


Figure 4.3: A search function that will search through the entire complex project architecture via `Navigate -> Search everywhere`

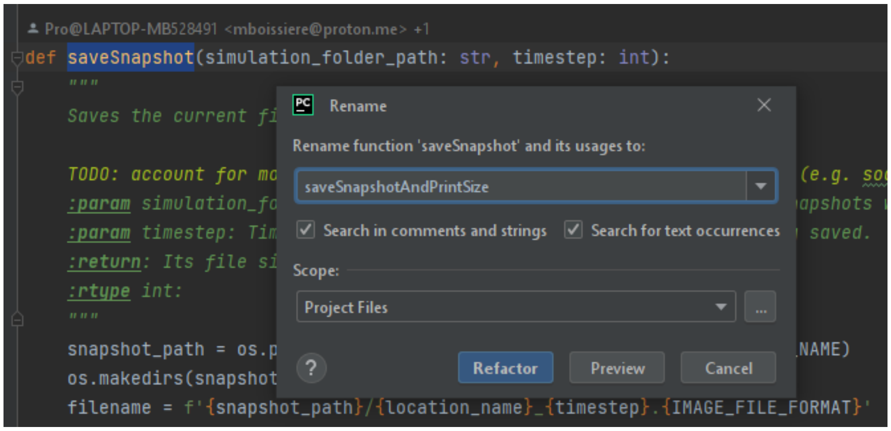


Figure 4.4: Rename an object or method via `right-click -> Refactor -> Rename...` will automatically rename all other usages in the project

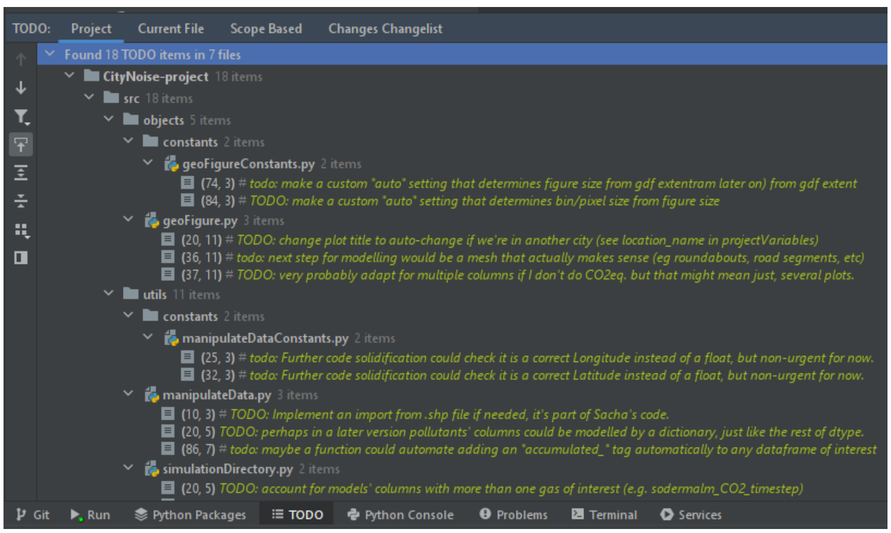


Figure 4.5: Everytime you write `"TODO"` in a comment, it will be stored in a TODO tab, so it's easy to see where there can be quick fixes or features to be implemented

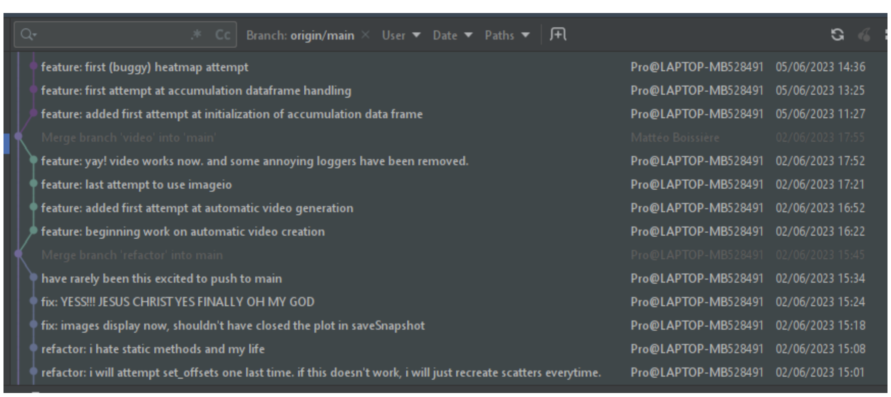


Figure 4.6: A visualization of the Git log via `Git -> Show Git Log` *(there are also buttons for push/pull/commit functionalities if you're still shy with the command line)*

And many more I don't have the time to name if I want to move on.

4.1.5 Matplotlib, Seaborn, Geoplot: what's up with modules that build on top of one another?

So, when people create Python modules, they often build on top of existing modules. After all, technically, **this very project is sort of a Python module that was built on top of matplotlib, seaborn, etc!** [3]

Usually, **the more a module is specialized** *(building functions from previous modules...)*, **the easier it is to do things** if you so happen to work in the field it covers. But! You also **lose some control**, working with auxilliary functions that reduce the number of variables for you and hide away the complicated stuff that goes on. So what does this all mean in practice? Let's look at an example.

And some modules I found in my "software review" seem to be perfect, and yet I didn't use them. Let's have a look at them

But I only used matplotlib and seaborn in my code. But *why* did I use these modules, is the big question? And should we keep using them?

see image : why$_n umpy_h as_m ore_c ontrol.pngnb : Automatedestimationofthenumberofbinsisnotsupportedforweightedd$

"Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures." anything that is built on top of something means that it will be simpler but less control basically

Geoplot Good example of KDEplot and pointplot mixing : https://residentmario.github.io/geoplot/gallery/plot$_b oston$
$https : //residentmario.github.io/geoplot/butbasically, explainedthatthemorespecialized, thesimpler, butalsothelesscon$

Consider mastering pyplot's arguments (Pierre's book?) instead of relying on seaborn which facilitates some work but also locks me (e.g. annoying to set basemap extent, to do an interpolation of histogram data for a continuous view) one section could be "architecture and module, reasons why (attempts of other things that failed), possible improvements (ie numpy)

`https://geopandas.org/en/stable/docs/user_guide/mapping.html`

Note : geoplot is actually not taht much of a lifesaver, doesn't have 2D histograms... (Consider contributing?)

investigate cartopy, seems like a better route for plotting actually

4.2 Software review, beyond the architecture I use

4.2.1 Some quick possible developments

If you're reading this, then I also haven't really got the time to write this section.

faire un mode switch : end state et dynamic pour juste faire moins de temps et avoir le truc de fin seulement si c trop lourd de faire savefig de chaque snapshot

perhaps seperate stuff by : conditional variables (pretty mode and fast mode), as well as pre-treatment and post-treatment. take care to archive what is done, like TiTAN did with the CSV copy.

perhaps a simple sum "total emissions over entire region" could also be done so we don't juste see 50 pixels of 3kg CO2 and think "oh that is low" for example

parler des tests

for accessibility : https://www.color-blindness.com/coblis-color-blindness-simulator/ (thank u paul!)

depending on the simulation time, having a fixed maximum on the colorbar is either better or annoying.

peut etre dans avenir très lointain de notebook indiquer méthode de parallélisation psk je sens que les pythons ça peut être lent as fuck

more coherent mesh that looks at roads, intersections, perhaps using .shp files / QGIS software There are some methods in GeoPandas (such as "SpatialIndex.nearest") which would enable you to find the closest road segment (or point) for each vehicle position

[3] Sort of, because it would actually need some extra refactoring for it to be entirely true, there would have to be __init__.py files here and there, but that's beside the point.

Chapter 5

Results

actually runs out of memory at some point, and not because of screenshots!

include file sizes of CSV and snapshots bc LMAO

according to my cool roommate, pyspark is a framework for distributed computing that helps with those memory problems. python doing spark behind, pandas but with pyspark -¿ is used for big data.

Chapter 6

Compilation of ideas for future developments

6.1 Future directions, topics, questions that papers could answer

As we know, "every tenth of a degree matters" when it comes to global warming. With noise pollution already being a harm for biodiversity, air pollution is also a matter of public health. So, we have chosen to look into air pollution models, to gain a better of understanding of vehicles' behavior through simulations.

However, research papers tend to answer a very specific question, or test a very specific hypothesis, and that is something we need to identify before getting lost in all the possibilities. So, where could all these simulations take us? What's the endgame?

6.1.1 An individual approach: how could a given vehicle pollute the least they can

The question I want to answer is the following. Given that "every tenth of a degree counts", a significant research is done on other things blabla other modes of transport but still people need cars and so we could model ways to reduce emissions the best (NB : SUMO actually already did a lot of this https://sumo.dlr.de/docs/Topics/Environmental$_I$*ssues.html*)

Literature on the Models and their implementation can be found at : `https://elib.dlr.de/89398/`.

Advantages : very interesting questions, scientifically speaking

Low Emission Zones and Citymapper's "Least Polluting" Path

-¿ similarly to how Sacha considers alternate paths, think of acceptability, something for an individual Citymapper could make a "least polluting" based on road traffic ? (no bc we don't have real time, so it would have to ignore traffic, which kinda doesn't make sense, except if we have something like Waze's API) another idea i had was a "least polluting path" like a citymapper app would. would require access to waze api. could be an idea suggested at the end if i dont have the time. or HERE api : `https://developer.here.com/documentation/traffic/dev_guide/topics/what-is.html` the MVP could be randomly generated traffic, and we could use the HERE api to obtain something more faithful

-¿with noise, there are hours that are best to pollute. does that existe with air pollution? uh i mean if you charge electric cars with an energy mix that isn't completely decarbonated i guess but not everyone has electric cars

Inspirations could be taken from Sacha's approaches. But does it make sense to consider "receiver-based" pollution as we do with noise? Perhaps driving around parks genuinely helps (capture?), perhaps we only need to minimize a very global function (in that case, it's more important to have stuff like green waves, cf research papers I SHOULD READ..)

Disadvantages: note that this might be super mega negligible and feed into individual consumer guilt-tripping when many activists and researchers agree that a dose of pragmatism is needed in this issue *(for example, it is useless to be a superhero when it comes to the way you use your car, if you then board a plane to go on vacation to somewhere that was easily accessible by train).*

6.1.2 An economical approach: what is the optimal trajectory for renovating the vehicle fleet

(prospective modelling approach)

current diesel v gasoline fleet : https://www.acea.auto/fuel-pc/fuel-types-of-new-cars-petrol-52-3-diesel-29-9-electric-6-8

different possible approaches : comparing results with HBEFA/COPERT/VERSIT+ (apparently copert undershoots it), comparing what happens when fleet changes, for example 33/33/33 EURO IV-V-VI is better or 50% EURO VI with poor I/II/III is better... similar to how we want to renovate buildings in france, could give leads to best action! by making these things variable. maybe, maybe then, move on to the economical stuff.

Questions are already more interesting if we wish to have direct action or to influence policy, because we're hitting broader, more systemic questions. Say we want to complete our net zero transition by 2050, like for real and not just relying on carbon offsets whose legitimacy has been put into question many times. Perhaps some cars will be better scrapped because they're so old and polluting and have done their time, or perhaps some cars are at the beginning of their life cycle and will be better renovated. Perhaps some car owners actually use their cars very little and could be convinced to favor a rent economy and public transport for their travels, or perhaps the area where they live makes it so that they really really need their car and we'd better incentivize them to carpool or renovate and such.

We already know for example that some areas give vehicle owner fines if their EURO emission standard is too low, which is a noble goal but cannot be there all by itself: consumers being punished without being helped will only feel attacked by the system, and as a great source of wisdom final fantasy tactics said (show image) "If a crime is only punishable with a fine, then it is a punishment for the poor only". All in all, fascinating questions I'm woefully unprepared to tackle head-on.

An economical model would be really cool. Like, use current prices of fuel and make the fine for trespassing a problem. A government help fund for investment could help. But ahh cycle de vie etc.

faut commencer a faire euro norms pour que y ait un levier interessant EURO standards good introduction : https://www.rac.co.uk/drive/advice/emissions/euro-emissions-standards/ (store an archive pdf in case) But talk about the controversy that some manufacturers lie. https://en.wikipedia.org/wiki/European$_{e}mission_{s}$

Modelling varies depending if it's passenger cars, motor cycles, light commercial vehicles, etc... https://en.wikipedia.org/wiki/European$_{e}mission_{s}tandardsEmission_{s}tandards_{f}or_{p}assenger_{c}arshttps: //dieselnet.com/standards/eu/ld.phphttps: //www.eea.europa.eu/publications/emep-eea-guidebook-$ 2019$/emission-factors-databaseModellingcouldstartwithpassengercarsonly, andthenaddmoretypesofvehicles...(note$

This could be studied through an economic lens *(what incentives exactly could work and change behavior on a large scale, how much would cost a renovation or recycling of the fleet)*, while keeping a very strong grasp on technical aspects *(knowledge of the current fleet and how it could pollute on average, and that's where the simulation comes in).*

See: The Shift Project

behavioral models (norway has a lot of electric cars, how are charging stations spread out and do the residents complain ?)

Also the carbon footprint stuff problem w carbon footprint approach : would be long itineraties if we simulate it ourselves, and sumo struggles w large areas

6.1.3 Time-Based Approaches for Emissions

taf, [06/04/2023 09:43] be careful for example of time effects, noise quickly is gone but not pollution

but yeah there is dissipation in most models

6.1.4 Which scale for modelling?

(there is also a french estimation of EURO fleet! i could do examples in paris for example! and ask khave how to get a simulation kinda like a SQL query of paris instead of rectangles!)

SUMO is good, but it's possible to work without it https://github.com/ethanpng2021/creatorbox be aware tho that it might have some great features for what we attempt to do, cf enviromental documentation

however it is clear that the most developped tool is SUMO, just that it isn't super beginner friendly

Talk about how one could link up with SUMO's API to study different places geographically (find correspondance with Kaveh on isolating an area, leave the .zip of polygon files ?) A good start for modelling Stockholm would be the low-emission zone https://drive2.city/stockholm/: :text=Lowhttps://dataportalen.stock

Importing OSM : https://sumo.dlr.de/docs/Networks/Import/OpenStreetMap.html

Actual simulation of Swedish fleet that is probably easier to navigate if you speak Swedish : https://www.trafa.se/vagtrafik/fordon/?cw=1

Recommend working in a clean and simple folder so that paths aren't annoying. Like creating a Sumo folder where one could copy the randomTrips.py script, the routes obtained via a WebWizard run or a modified network XML, etc etc. Start the whole process over for Södermalm, following ChatGPT advice, for the Jupyter tutorial.

NB : what is 3600 duration ? temps de génération des véhicules peut-être ? regarder aussi comment enlever certaines routes pour que ça soit plus propre, vu que sodermalm par exemple n'est pas parfaitement un rectangle, il y a des incrustes

You can manually configure right of way (traffic lights?) in sumo, perhaps try to apply 2012 results and see if I get them again. Specifying vehicle behaviour (different speeds) could be explored, and again we need a partition of EURO norms. Behaviour in vehicles, but also in infrastructure (green waves) could be explored, so far definiing traffic lights seems weird. `https://sumo.dlr.de/docs/Networks/PlainXML.html#right-of-way`

ATTEMPTS OF FIRST TEST : ran simulation on modified södermalm, but the fact that cars stack up is so weird. on simulations from the OSM Wizard, as well as ones with a command line generated randomTrips. Would it be better to just go WebWizard all the way, and ignore some routes? Or use the GUI instead of command line prompt?

-¿ dig deep on emissions chemical interaction, similarly to how Sacha looks at reflexivity, interactions with concrete buildings and grassy parks etc, perhaps there are places where it's simply best to pollute?

It might be tempting to do CO2eq because that is the approach of carbon footprint calculation, BUT Sacha recommends possibility of looking at other pollutants, instead of gathering everything in a CO2eq. Fine particules, NOx, HC etc they have a local effect so it is important to keep them in check, even if they have a lower global impact than CO2.

6.2 Future features, partially implemented or not, in the codebase

6.2.1 Software side

also, consider starting the README, and explaining some healthy code habits that i use and hope to maintain. according to paul, there is a huge statement by Google on what they personally do. there's also "clean code" the book. the conventions i try to follow for commits. etc.

Chapter 7

Conclusion and personal thoughts

Personal thoughts on my time here (i am a bit disappointed in myself in the ”research” part, my tool is honestly pretty useless and describes no physical reality. but i reckon that happens in a first-timer research internship! however i am proud of myself for having been able to share what ive learned in software-oriented internships and courses, proud of work i’ve done in codebase)

Bibliography

[1] Francisco D.B. Albuquerque, Munjed A. Maraqa, Rezaul Chowdhury, Timur Mauga, and Mohammed Alzard. Greenhouse gas emissions associated with road transport projects: current status, benchmarking, and assessment tools. *Transportation Research Procedia*, 48:2018–2030, 2020.

[2] Matthew Barth, Theodore Younglove, Tom Wenzel, George Scora, Feng An, Marc Ross, and Joseph Norbeck. Analysis of Modal Emissions From Diverse In-Use Vehicle Fleet. *Transportation Research Record: Journal of the Transportation Research Board*, 1587(1):73–84, January 1997.

[3] Margarida C. Coelho, Tiago L. Farias, and Nagui M. Rouphail. Effect of roundabout operations on pollutant emissions. *Transportation Research Part D: Transport and Environment*, 11(5):333–343, September 2006.

[4] Margarida C. Coelho, H. Christopher Frey, Nagui M. Rouphail, Haibo Zhai, and Luc Pelkmans. Assessing methods for comparing emissions from gasoline and diesel light-duty vehicles based on microscale measurements. *Transportation Research Part D: Transport and Environment*, 14(2):91–99, March 2009.

[5] P. Fernandes, M.C. Coelho, and N.M. Rouphail. Assessing the impact of closely-spaced intersections on traffic operations and pollutant emissions on a corridor level. *Transportation Research Part D: Transport and Environment*, 54:304–320, July 2017.

[6] José M. López-Martínez, Felipe Jiménez, F. Javier Páez-Ayuso, M. Nuria Flores-Holgado, Angélica N. Arenas, Blanca Arenas-Ramirez, and Francisco Aparicio-Izquierdo. Modelling the fuel consumption and pollutant emissions of the urban bus fleet of the city of Madrid. *Transportation Research Part D: Transport and Environment*, 52:112–127, May 2017.