Supplement of Geosci. Model Dev., 17, 6465–6487, 2024
https://doi.org/10.5194/gmd-17-6465-2024-supplement


Geoscientific Model Development
Open Access

*Supplement of*

# Description and validation of Vehicular Emissions from Road Traffic (VERT) 1.0, an R-based framework for estimating road transport emissions from traffic flows

**Giorgio Veratti et al.**

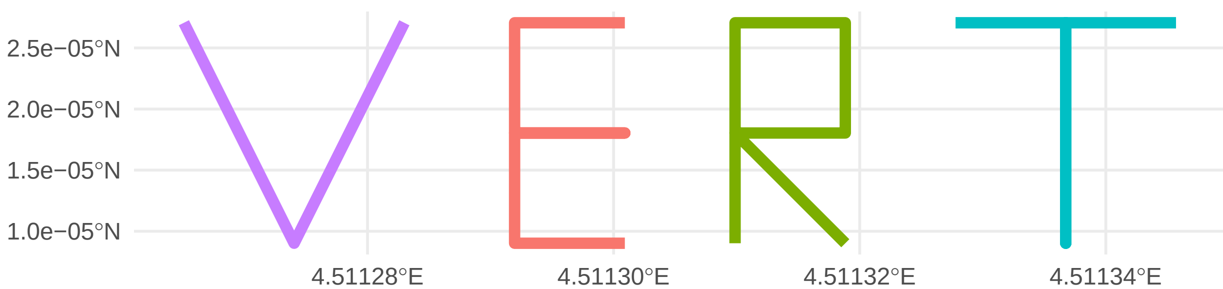
*Correspondence to:* Giorgio Veratti (giorgio.veratti@unimore.it)

# S1 Vignette documentation for emissions estimation using VERT

```r
# Install package from source
install.packages("path/to/file/vert_1.0.tar.gz", repos = NULL, type="source")
```

```r
suppressMessages(library(sf))
suppressMessages(library(ggplot2))
suppressMessages(library(dplyr))
suppressMessages(library(vert))

# Generate sample data to simulate traffic flows within a fictitious road network
# Create LINESTRING geometries
v_coords <- matrix(c(1, 3, 2, 1, 3, 3), ncol = 2, byrow = TRUE)
e_coords <- matrix(c(5, 1, 4, 1, 4, 2, 5, 2, 4, 2, 4, 3, 5, 3),
                   ncol = 2, byrow = TRUE)
r_coords <- matrix(c(6, 1, 6, 3, 7, 3, 7, 2, 6, 2, 7, 1), ncol = 2, byrow = TRUE)
t_coords <- matrix(c(8, 3, 9, 3, 9, 1, 9, 3, 10, 3), ncol = 2, byrow = TRUE)
v_linestring <- st_linestring(v_coords)
e_linestring <- st_linestring(e_coords)
r_linestring <- st_linestring(r_coords)
t_linestring <- st_linestring(t_coords)

# Create a data frame
letters_df <- data.frame(letter = c("V", "E", "R", "T"),
                         geometry = st_sfc(v_linestring,
                                           e_linestring,
                                           r_linestring,
                                           t_linestring),
                         stringsAsFactors = FALSE)

# Convert the data frame to an sf object and add necessary data
road_network <- road_network_2023ef <- st_sf(data = letters_df) %>%
    ## Assign CRS
    st_set_crs("+proj=utm +zone=32 +datum=WGS84 +units=m +no_defs") %>%
    ## Compute the road length for each segment, in kilometre.
    mutate(road.length = as.numeric(st_length(geometry)) / 1000,
           ## Let's create some dummy data.
           ## Number of passenger cars travveling for each road segment.
           traffic.flux.pc = c(100, 75, 50, 30),
           ## Number of mopeds/motorcycle travveling for each road segment.
           traffic.flux.Lcat = c(10, 40, 10, 5),
           ## Number of Light Commercial vehicles travveling for each road segment.
           traffic.flux.lcv = c(15, 15, 30, 10),
           ## Number of Heavy commercial vehicles travveling for each road seg.
           traffic.flux.hcv = c(2, 7, 5, 10),
           ## Reference speed for passenger cars in km/h.
```

```r
          traffic.speed.pc = 70,
          ## Reference speed for mopeds/motorcycle in km/h.
          traffic.speed.Lcat = 45,
          ## Reference speed for Light Commercial vehicles in km/h.
          traffic.speed.lcv = 60,
          ## Reference speed for Heavy Commercial vehicles in km/h.
          traffic.speed.hcv = 50,
          ## Type of traffic
          traffic.type = c("Urban Peak", "Urban Off Peak", "Rural", "Highway"),
          ## Road slope
          road.slope = c(-0.02, 0, 0.02, 0.04),
          ## Load of Heavy Commercial vehicles.
          hdv.load = c(0, 0.5, 0.5, 1),
          ## Grams per square meter of silt loading on the road surface.
          ## This information is exclusively required for the resuspension
          ## calculation conducted using the EPA methodology.
          sL = c(0.2, 0.3, 0.4, 0.5))


# Let's plot the road network
ggplot() +
    geom_sf(data = road_network, aes(color = letter), size = 2) +
    theme_minimal() +
    theme(legend.position = "none")
```




```r
# After preparing the road network with traffic data, the next step is to set up
# the fleet composition data frame.
# This involves defining four distinct traffic fluxes for Passenger cars,
# mopeds/motorcycles, Light Commercial vehicles, and Heavy Commercial vehicles.
# To streamline this process, refer to the "fleet_example_4fluxes" data frame
# included in the package:
View(fleet_example_4fluxes)

# It's important to note that the order of each class is predefined. The columns
# "perc_urban," "perc_rural," and "perc_highway" represent the percentage
# breakdown of the fleet composition for Urban Peak/Urban Off Peak, Rural and
# Highway road types, respectively.
# Given that we are working with four distinct traffic fluxes, the sum of each
# column must be 4. Specifically, rows 1 to 325 pertain to Passenger cars and
# should sum to 1. Rows 326 to 371 are allocated to mopeds/motorcycles and must
# also sum to 1. Rows 372 to 459 correspond to Light Commercial vehicles and
# should total 1. Finally, rows 460 to 666 are designated for Heavy Commercial
# vehicles and must sum to 1.
# The "year" column denotes the reference year for the vehicle's sold year.
```

```r
# This information helps estimate the approximate age of a certain category and
# subsequently the distance it might have traveled. Users can adjust the years
# based on their preferences. Feel free to leave other columns unchanged.

# Let's compute the exhaust emissions for some pollutants using the traffic
# data previously created and the fleet composition available in the package.
for(pol in c("CO", "VOC", "SO2", "NH3", "N2O")){

                         ## Use here your fleet composition.
    road_network <- main(fleet = fleet_example_4fluxes,
                         ## This is your reference road netowk.
                         traffic.df = road_network,
                         ## Number of traffic classes,
                         ## in this case PC, Lcat, LCV and HCV.
                         n.traffic.flux = 4,
                         ## Emission factors reference year,
                         ## two options are available: 2020 & 2023
                         emep.eea.year = 2020,
                         ## Pollutant of interest
                         pollutant = pol,
                         ## Emission type
                         emis.type = "exhaust",
                         ## The mileage share attributed to urban conditions (km).
                         Surb = 0.804,
                         ## The mean trip distance (km).
                         L.trip = 2.5,
                         ## Ambient temperature in Celsius degree (°C).
                         Ta = 8.5,
                         ## The simulation year. it is used to estimate the
                         ## distance travelled by each vehicle category.
                         reference.year = 2023,
                         ## Sulfur content in diesel blend (ppm).
                         sulphur.diesel = 10,
                         ## Sulfur content in petrol blend (ppm).
                         sulphur.petrol = 10,
                         ## Sulfur content in lubricant oil (ppm).
                         sulphur.lube = 1000,
                         ## "single.core" or "multi.core" computation
                         computation = "single.core",
                         ## Write or not a log file
                         log.file = F)

}

# Take a look at the new columns
road_network
```

```
## Simple feature collection with 4 features and 19 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: 1 ymin: 1 xmax: 10 ymax: 3
## CRS:           +proj=utm +zone=32 +datum=WGS84 +units=m +no_defs
##   letter                       geometry road.length traffic.flux.pc
## 1      V     LINESTRING (1 3, 2 1, 3 3) 0.004472136             100
```

```
## 2      E LINESTRING (5 1, 4 1, 4 2, ... 0.006000000             75
## 3      R LINESTRING (6 1, 6 3, 7 3, ... 0.006414214             50
## 4      T LINESTRING (8 3, 9 3, 9 1, ... 0.006000000             30
##   traffic.flux.Lcat traffic.flux.lcv traffic.flux.hcv traffic.speed.pc
## 1                10               15                2               70
## 2                40               15                7               70
## 3                10               30                5               70
## 4                 5               10               10               70
##   traffic.speed.Lcat traffic.speed.lcv traffic.speed.hcv   traffic.type
## 1                 45                60                50     Urban Peak
## 2                 45                60                50 Urban Off Peak
## 3                 45                60                50          Rural
## 4                 45                60                50        Highway
##   road.slope hdv.load  sL      emis_CO     emis_VOC     emis_SO2     emis_NH3
## 1      -0.02      0.0 0.2 0.0007331939 1.914952e-04 7.242136e-07 7.427194e-06
## 2       0.00      0.5 0.3 0.0016243768 5.325562e-04 1.190059e-06 9.211000e-06
## 3       0.02      0.5 0.4 0.0007577965 1.590683e-04 1.087168e-06 8.824058e-06
## 4       0.04      1.0 0.5 0.0002595472 6.674373e-05 1.143903e-06 5.018285e-06
##       emis_N2O
## 1 8.356665e-06
## 2 9.726834e-06
## 3 3.530245e-06
## 4 2.241127e-06
```

```r
# Let's test the multi core computation for other pollutants.
# Please note that the 'parallel' package is required.
for(pol in c("NOx", "CH4", "PM Exhaust")){
                         ## Use here your fleet composition.
    road_network <- main(fleet = fleet_example_4fluxes,
                         ## This is your reference road netowk.
                         traffic.df = road_network,
                         ## Number of traffic classes,
                         ## in this case PC, Lcat, LCV and HCV.
                         n.traffic.flux = 4,
                         ## Emission factors reference year,
                         ## two options are available: 2020 & 2023
                         emep.eea.year = 2020,
                         ## Pollutant of interest
                         pollutant = pol,
                         ## Emission type
                         emis.type = "exhaust",
                         ## The mileage share attributed to urban conditions (km).
                         Surb = 0.804,
                         ## The mean trip distance (km).
                         L.trip = 2.5,
                         ## Ambient temperature in Celsius degree (°C).
                         Ta = 8.5,
                         ## The simulation year. it is used to estimate the distance
                         ## travelled by each vehicle category.
                         reference.year = 2023,
                         ## Sulfur content in diesel blend (ppm).
                         sulphur.diesel = 10,
                         ## Sulfur content in petrol blend (ppm).
                         sulphur.petrol = 10,
```

```r
                            ## Sulfur content in lubricant oil (ppm).
                            sulphur.lube = 1000,
                            ## "single.core" or "multi.core" computation
                            computation = "multi.core",
                            ## Number of cores
                            n.core = 4)

}

# Take a look at the new columns
road_network
```

```
## Simple feature collection with 4 features and 22 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: 1 ymin: 1 xmax: 10 ymax: 3
## CRS:           +proj=utm +zone=32 +datum=WGS84 +units=m +no_defs
##   letter                      geometry road.length traffic.flux.pc
## 1      V     LINESTRING (1 3, 2 1, 3 3) 0.004472136            100
## 2      E LINESTRING (5 1, 4 1, 4 2, ... 0.006000000             75
## 3      R LINESTRING (6 1, 6 3, 7 3, ... 0.006414214             50
## 4      T LINESTRING (8 3, 9 3, 9 1, ... 0.006000000             30
##   traffic.flux.Lcat traffic.flux.lcv traffic.flux.hcv traffic.speed.pc
## 1                10               15                2               70
## 2                40               15                7               70
## 3                10               30                5               70
## 4                 5               10               10               70
##   traffic.speed.Lcat traffic.speed.lcv traffic.speed.hcv   traffic.type
## 1                 45                60                50     Urban Peak
## 2                 45                60                50 Urban Off Peak
## 3                 45                60                50          Rural
## 4                 45                60                50        Highway
##   road.slope hdv.load  sL      emis_CO     emis_VOC     emis_SO2     emis_NH3
## 1      -0.02      0.0 0.2 0.0007331939 1.914952e-04 7.242136e-07 7.427194e-06
## 2       0.00      0.5 0.3 0.0016243768 5.325562e-04 1.190059e-06 9.211000e-06
## 3       0.02      0.5 0.4 0.0007577965 1.590683e-04 1.087168e-06 8.824058e-06
## 4       0.04      1.0 0.5 0.0002595472 6.674373e-05 1.143903e-06 5.018285e-06
##       emis_N2O     emis_NOx     emis_CH4 emis_PM.Exhaust
## 1 8.356665e-06 0.0002872645 9.049042e-06    5.988979e-06
## 2 9.726834e-06 0.0004639886 2.306441e-05    1.236062e-05
## 3 3.530245e-06 0.0005434518 7.171920e-06    6.855393e-06
## 4 2.241127e-06 0.0006820063 2.895485e-06    1.097837e-05
```

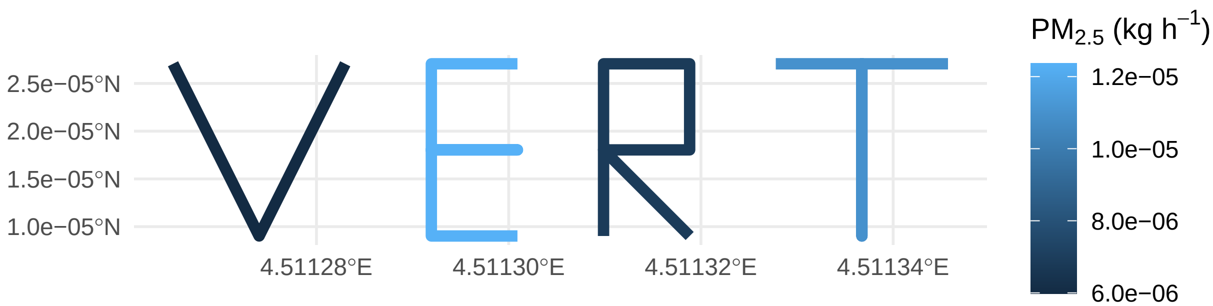
```r
# Let's examine the output generated using the updated emission factors from 2023.
for(pol in c("NOx", "CH4", "PM Exhaust")){
                                 ## Use here your fleet composition.
    road_network_2023ef <- main(fleet = fleet_example_4fluxes,
                                 ## This is your reference road netowk.
                                 traffic.df = road_network,
                                 ## Is the number of traffic classes,
                                 ## in this case PC, Lcat, LCV and HCV.
                                 n.traffic.flux = 4,
                                 ## Emission factors reference year,
                                 ## two are options available: 2020 & 2023
```

```
                                emep.eea.year = 2023,
                                ## Pollutant of interest
                                pollutant = pol,
                                ## Emission type
                                emis.type = "exhaust",
                                ## The mileage share attributed to urban
                                ## conditions (km).
                                Surb = 0.804,
                                ## The mean trip distance (km).
                                L.trip = 2.5,
                                ## Ambient temperature in Celsius degree (°C).
                                Ta = 8.5,
                                ## The simulation year. it is used to estimate
                                ## the distance travelled by each
                                ## vehicle category.
                                reference.year = 2023,
                                ## Sulfur content in diesel blend (ppm).
                                sulphur.diesel = 10,
                                ## Sulfur content in petrol blend (ppm).
                                sulphur.petrol = 10,
                                ## Sulfur content in lubricant oil (ppm).
                                sulphur.lube = 1000,
                                ## "single.core" or "multi.core" computation
                                computation = "multi.core",
                                ## Number of cores
                                n.core = 4)


}

# Take a look at the road_network_2023ef
road_network_2023ef
```

```
## Simple feature collection with 4 features and 23 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: 1 ymin: 1 xmax: 10 ymax: 3
## CRS:           +proj=utm +zone=32 +datum=WGS84 +units=m +no_defs
##   letter                       geometry road.length traffic.flux.pc
## 1      V     LINESTRING (1 3, 2 1, 3 3) 0.004472136             100
## 2      E LINESTRING (5 1, 4 1, 4 2, ... 0.006000000              75
## 3      R LINESTRING (6 1, 6 3, 7 3, ... 0.006414214              50
## 4      T LINESTRING (8 3, 9 3, 9 1, ... 0.006000000              30
##   traffic.flux.Lcat traffic.flux.lcv traffic.flux.hcv traffic.speed.pc
## 1                10               15                2               70
## 2                40               15                7               70
## 3                10               30                5               70
## 4                 5               10               10               70
##   traffic.speed.Lcat traffic.speed.lcv traffic.speed.hcv   traffic.type
## 1                 45                60                50     Urban Peak
## 2                 45                60                50 Urban Off Peak
## 3                 45                60                50          Rural
## 4                 45                60                50        Highway
##   road.slope hdv.load  sL      emis_CO     emis_VOC     emis_SO2     emis_NH3
```

```
## 1      -0.02      0.0 0.2 0.0007331939 1.914952e-04 7.242136e-07 7.427194e-06
## 2       0.00      0.5 0.3 0.0016243768 5.325562e-04 1.190059e-06 9.211000e-06
## 3       0.02      0.5 0.4 0.0007577965 1.590683e-04 1.087168e-06 8.824058e-06
## 4       0.04      1.0 0.5 0.0002595472 6.674373e-05 1.143903e-06 5.018285e-06
##       emis_N2O     emis_NOx     emis_CH4 emis_PM.Exhaust emis_PM.Exhaust
## 1 8.356665e-06 0.0002872645 9.049042e-06    5.988979e-06    5.250522e-06
## 2 9.726834e-06 0.0004639886 2.306441e-05    1.236062e-05    1.123473e-05
## 3 3.530245e-06 0.0005434518 7.171920e-06    6.855393e-06    6.240712e-06
## 4 2.241127e-06 0.0006820063 2.895485e-06    1.097837e-05    1.071503e-05
```

```
# Let's plot the PM2.5 emissions generated with VERT
ggplot() +
    geom_sf(data = road_network, aes(color = emis_PM.Exhaust), size = 2) +
    theme_minimal() +
    labs(color = expression(paste(PM[2.5], " ", "(", "kg ", h^{-1}, ")")))
```




```
# Running Losses evaparotive emissions
for(pol in c("VOC")){

    road_network <- main(fleet = fleet_example_4fluxes,
                  traffic.df = road_network,
                  n.traffic.flux = 4,
                  pollutant = pol,
                  emis.type = "evaporative-running_losses",
                  wear.source = NA,
                  Surb = 0.804,
                  L.trip = 12.5,
                  perc.wet.d = 0.2,
                  Ta = 8.5,
                  reference.year = 2020,
                  sulphur.diesel = 10,
                  sulphur.petrol = 10,
                  sulphur.lube = 1000,
                  computation = "multi.core",
                  n.core = 4)

}

# Resuspension emissions using the EPA formula
for(pol in c("PM10")){

    road_network <- main(fleet = fleet_example_4fluxes,
                  traffic.df = road_network,
                  n.traffic.flux = 4,
                  pollutant = pol,
```

```r
                  emis.type = "resuspension-EPA",
                  wear.source = NA,
                  Surb = 0.804,
                  L.trip = 12.5,
                  perc.wet.d = 0.2,
                  Ta = 8.5,
                  reference.year = 2020,
                  sulphur.diesel = 10,
                  sulphur.petrol = 10,
                  sulphur.lube = 1000,
                  computation = "multi.core",
                  n.core = 4)

}

# Resuspension emissions based on custom EF
for(pol in c("PM10")){

    road_network <- main(fleet = fleet_example_4fluxes,
                  traffic.df = road_network,
                  n.traffic.flux = 4,
                  pollutant = pol,
                  emis.type = "resuspension-customEF",
                  wear.source = NA,
                  Surb = 0.804,
                  L.trip = 12.5,
                  perc.wet.d = 0.2,
                  Ta = 8.5,
                  reference.year = 2020,
                  sulphur.diesel = 10,
                  sulphur.petrol = 10,
                  sulphur.lube = 1000,
                  resusp.EF.pc = 12.5,
                  resusp.EF.Lcat = 1.1,
                  resusp.EF.lcv = 45,
                  resusp.EF.hcv = 250,
                  computation = "multi.core",
                  n.core = 4)

}

# Non-exhaust emissions
for(sour in c("brake.wear", "surface.wear", "tyre.wear")){
  for(pol in c("TSP", "PM10", "PM2.5")){

    road_network <- main(fleet = fleet_example_4fluxes,
                  traffic.df = road_network,
                  n.traffic.flux = 4,
                  emep.eea.year = 2020,
                  pollutant = pol,
                  emis.type = "PM.wear",
                  wear.source = sour,
                  Surb = 0.804,
```

```r
                  L.trip = 2.5,
                  perc.wet.d = 0.2,
                  Ta = 10,
                  reference.year = 2017,
                  sulphur.diesel = 10,
                  sulphur.petrol = 10,
                  sulphur.lube = 1000,
                  computation = "multi.core",
                  n.core = 4)

   }
}

# Let's examine the output
road_network
```

```
## Simple feature collection with 4 features and 34 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: 1 ymin: 1 xmax: 10 ymax: 3
## CRS:           +proj=utm +zone=32 +datum=WGS84 +units=m +no_defs
##   letter                       geometry road.length traffic.flux.pc
## 1      V     LINESTRING (1 3, 2 1, 3 3) 0.004472136             100
## 2      E LINESTRING (5 1, 4 1, 4 2, ... 0.006000000              75
## 3      R LINESTRING (6 1, 6 3, 7 3, ... 0.006414214              50
## 4      T LINESTRING (8 3, 9 3, 9 1, ... 0.006000000              30
##   traffic.flux.Lcat traffic.flux.lcv traffic.flux.hcv traffic.speed.pc
## 1                10               15                2               70
## 2                40               15                7               70
## 3                10               30                5               70
## 4                 5               10               10               70
##   traffic.speed.Lcat traffic.speed.lcv traffic.speed.hcv   traffic.type
## 1                 45                60                50     Urban Peak
## 2                 45                60                50 Urban Off Peak
## 3                 45                60                50          Rural
## 4                 45                60                50        Highway
##   road.slope hdv.load  sL      emis_CO     emis_VOC     emis_SO2     emis_NH3
## 1      -0.02      0.0 0.2 0.0007331939 1.914952e-04 7.242136e-07 7.427194e-06
## 2       0.00      0.5 0.3 0.0016243768 5.325562e-04 1.190059e-06 9.211000e-06
## 3       0.02      0.5 0.4 0.0007577965 1.590683e-04 1.087168e-06 8.824058e-06
## 4       0.04      1.0 0.5 0.0002595472 6.674373e-05 1.143903e-06 5.018285e-06
##       emis_N2O     emis_NOx     emis_CH4 emis_PM.Exhaust emis_runningLosses_VOC
## 1 8.356665e-06 0.0002872645 9.049042e-06    5.988979e-06           3.869214e-06
## 2 9.726834e-06 0.0004639886 2.306441e-05    1.236062e-05           6.843480e-06
## 3 3.530245e-06 0.0005434518 7.171920e-06    6.855393e-06           9.797648e-06
## 4 2.241127e-06 0.0006820063 2.895485e-06    1.097837e-05           4.561893e-07
##   emis_PM10_resusp_EPA emis_PM10_resusp_customEF emis_brake.wear_TSP
## 1         0.0009468551              1.089412e-05        4.330576e-06
## 2         0.0018184606              2.043900e-05        7.441182e-06
## 3         0.0039884163              2.075640e-05        6.463377e-06
## 4         0.0073204470              1.998300e-05        5.661865e-06
##   emis_brake.wear_PM10 emis_brake.wear_PM2.5 emis_surface.wear_TSP
## 1         4.243965e-06          1.688925e-06          8.679822e-06
## 2         7.292358e-06          2.902061e-06          1.282481e-05
```

```
## 3         6.334109e-06          2.520717e-06          1.054411e-05
## 4         5.548628e-06          2.208127e-06          8.351601e-06
##   emis_surface.wear_PM10 emis_surface.wear_PM2.5 emis_tyre.wear_TSP
## 1           4.339911e-06            2.343552e-06       7.098311e-06
## 2           6.412405e-06            3.462699e-06       1.004305e-05
## 3           5.272057e-06            2.846911e-06       9.334321e-06
## 4           4.175801e-06            2.254932e-06       6.652993e-06
##   emis_tyre.wear_PM10 emis_tyre.wear_PM2.5
## 1        4.258986e-06         2.981290e-06
## 2        6.025830e-06         4.218081e-06
## 3        5.600593e-06         3.920415e-06
## 4        3.991796e-06         2.794257e-06
```

```
# Let's test the emission factors from 2023
for(sour in c("brake.wear", "surface.wear", "tyre.wear")){
  for(pol in c("TSP", "PM10", "PM2.5")){

    road_network_2023ef <- main(fleet = fleet_example_4fluxes,
                   traffic.df = road_network,
                   n.traffic.flux = 4,
                   emep.eea.year = 2023,
                   pollutant = pol,
                   emis.type = "PM.wear",
                   wear.source = sour,
                   Surb = 0.804,
                   L.trip = 2.5,
                   perc.wet.d = 0.2,
                   Ta = 10,
                   reference.year = 2017,
                   sulphur.diesel = 10,
                   sulphur.petrol = 10,
                   sulphur.lube = 1000,
                   computation = "multi.core",
                   n.core = 4)

   }
}

# Let's examine the output
road_network_2023ef
```

```
## Simple feature collection with 4 features and 35 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: 1 ymin: 1 xmax: 10 ymax: 3
## CRS:           +proj=utm +zone=32 +datum=WGS84 +units=m +no_defs
##   letter                       geometry road.length traffic.flux.pc
## 1      V     LINESTRING (1 3, 2 1, 3 3) 0.004472136             100
## 2      E LINESTRING (5 1, 4 1, 4 2, ... 0.006000000              75
## 3      R LINESTRING (6 1, 6 3, 7 3, ... 0.006414214              50
## 4      T LINESTRING (8 3, 9 3, 9 1, ... 0.006000000              30
##   traffic.flux.Lcat traffic.flux.lcv traffic.flux.hcv traffic.speed.pc
## 1                10               15                2               70
## 2                40               15                7               70
## 3                10               30                5               70
```

```
## 4                 5              10              10              70
##   traffic.speed.Lcat traffic.speed.lcv traffic.speed.hcv   traffic.type
## 1                 45                60                50     Urban Peak
## 2                 45                60                50 Urban Off Peak
## 3                 45                60                50          Rural
## 4                 45                60                50        Highway
##   road.slope hdv.load  sL      emis_CO     emis_VOC     emis_SO2     emis_NH3
## 1      -0.02      0.0 0.2 0.0007331939 1.914952e-04 7.242136e-07 7.427194e-06
## 2       0.00      0.5 0.3 0.0016243768 5.325562e-04 1.190059e-06 9.211000e-06
## 3       0.02      0.5 0.4 0.0007577965 1.590683e-04 1.087168e-06 8.824058e-06
## 4       0.04      1.0 0.5 0.0002595472 6.674373e-05 1.143903e-06 5.018285e-06
##       emis_N2O     emis_NOx     emis_CH4 emis_PM.Exhaust emis_runningLosses_VOC
## 1 8.356665e-06 0.0002872645 9.049042e-06    5.988979e-06           3.869214e-06
## 2 9.726834e-06 0.0004639886 2.306441e-05    1.236062e-05           6.843480e-06
## 3 3.530245e-06 0.0005434518 7.171920e-06    6.855393e-06           9.797648e-06
## 4 2.241127e-06 0.0006820063 2.895485e-06    1.097837e-05           4.561893e-07
##   emis_PM10_resusp_EPA emis_PM10_resusp_customEF emis_brake.wear_TSP
## 1         0.0009468551              1.089412e-05        4.330576e-06
## 2         0.0018184606              2.043900e-05        7.441182e-06
## 3         0.0039884163              2.075640e-05        6.463377e-06
## 4         0.0073204470              1.998300e-05        5.661865e-06
##   emis_brake.wear_PM10 emis_brake.wear_PM2.5 emis_surface.wear_TSP
## 1         4.243965e-06          1.688925e-06          8.679822e-06
## 2         7.292358e-06          2.902061e-06          1.282481e-05
## 3         6.334109e-06          2.520717e-06          1.054411e-05
## 4         5.548628e-06          2.208127e-06          8.351601e-06
##   emis_surface.wear_PM10 emis_surface.wear_PM2.5 emis_tyre.wear_TSP
## 1           4.339911e-06            2.343552e-06       7.098311e-06
## 2           6.412405e-06            3.462699e-06       1.004305e-05
## 3           5.272057e-06            2.846911e-06       9.334321e-06
## 4           4.175801e-06            2.254932e-06       6.652993e-06
##   emis_tyre.wear_PM10 emis_tyre.wear_PM2.5 emis_tyre.wear_PM2.5
## 1        4.258986e-06         2.981290e-06         2.869393e-06
## 2        6.025830e-06         4.218081e-06         4.083091e-06
## 3        5.600593e-06         3.920415e-06         3.696518e-06
## 4        3.991796e-06         2.794257e-06         2.756307e-06
```

```r
# Now let's suppose we want to export NOx emissions in a format
# suitable for GRAL, the GRAz Lagrangian model.
# Supposing that the traffic data provided to VERT is representative
# of an hourly interval, we convert emissions from kg/h to kg/(km*h)
# and then we suppose that different road have different modulations,
# i.e. belong to different source group in the GRAL environment:
network4GRAL <- road_network %>%
     mutate(emis_NOx4GRAL = emis_NOx / road.length,
            sg = ifelse(traffic.type == "Urban Peak", 1,
                 ifelse(traffic.type == "Urban Off Peak", 2,
                 ifelse(traffic.type == "Rural", 3, 4))),
            road_width = ifelse(traffic.type == "Urban Peak", 5,
                 ifelse(traffic.type == "Urban Off Peak", 5,
                 ifelse(traffic.type == "Rural", 7, 8))))


# Export the emissions
```

```r
emis2GRAL(road_network = network4GRAL,
          road_name = "letter",
          source_group = "sg",
          road_width = "road_width",
          road_emis = "emis_NOx4GRAL",
          share_pm2.5 = 0,
          share_pm10 = 0,
          pm_diam = 0,
          pm_density = 0,
          dry_dep_vel_pm2.5 = 0,
          dry_dep_vel_pm10 = 0,
          dry_dep_vel_pm30 = 0,
          mode = 0,
          dir_out = "~/Desktop/")
```