Analyse de la concentration de CO2 dans l'atmosphère depuis 1958

François Févotte

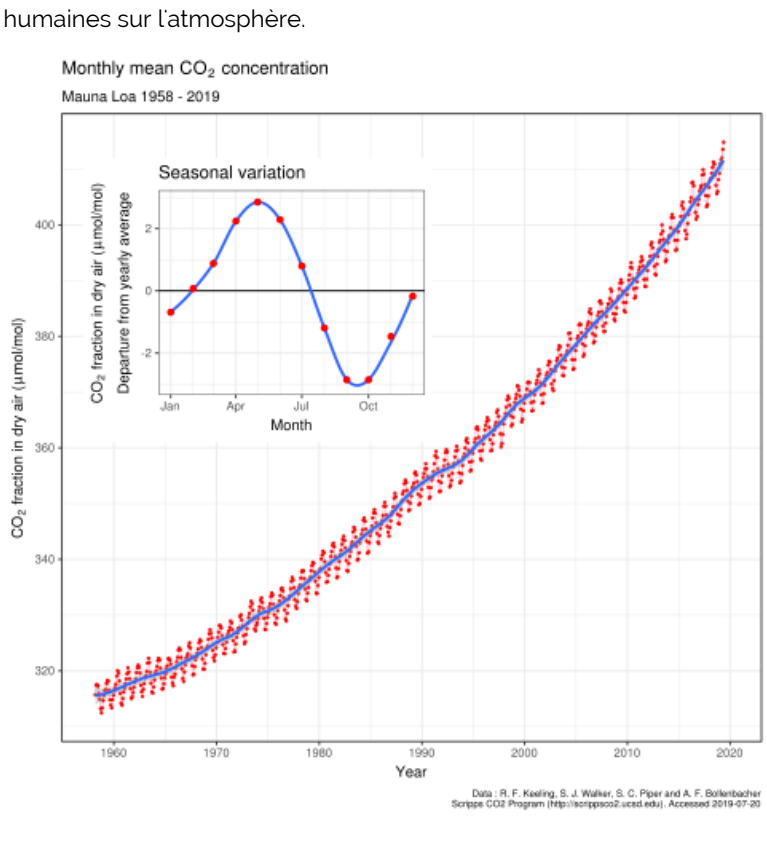
avril 2020

Ce document a été réalisé à l'occasion du MOOC "Recherche Reproductible". Il a donc plus vocation à fournir des résultats *reproductibles* que des résultats *intéressants*. N'étant pas un scientifique des données, il est tout à fait possible que l'analyse réalisée ici soit caduque ; je plaide pour l'indulgence du lecteur !

Ce document a été automatiquement généré à partir du fichier `exercice.jmd`, que le lecteur intéressé pourra consulter directement. Une annexe détaille toutes les étapes nécessaires afin de reproduire l'analyse et re-générer intégralement le présent document. Cette annexe, nommée `repro.html`, devrait se trouver dans le même dépôt git et même répertoire que le présent document.

Résumé:

Depuis 1958, la concentration atmosphérique en CO2 est mesurée régulièrement à l'observatoire de Mauna Loa, à Hawaï. Ces mesures, initiées par Charles D. Keeling, ont été les parmi les premières à mettre en évidence l'accroissement rapide de la concentration en CO2 dans l'atmosphère, un constat qui a par la suite permis d'attirer l'attention de la communauté scientifique (puis par la suite du grand public) sur l'impact des activités humaines sur l'atmosphère.



La courbe de Keeling fait apparaître une augmentation de la concentration atmosphérique de l'ordre de 1 à 2 ppm/an [monroe2015]. Elle fait aussi apparaître une variation cyclique d'environ 5 ppm chaque année, qui rend l'analyse de la tendance systématique haussière plus délicate.

Nous nous proposons ici de refaire pas à pas l'étude de ce jeu de données. Nous séparons la composante cyclique de la composante tendancielle, afin de caractériser cette dernière. Les paramètres de notre modèle correspondent à un taux d'augmentation du CO2 atmosphérique de 0.8 ppm/an en 1959, et 1.5 ppm/an en 2015 (cette dernière valeur étant un peu sous-estimée par rapport aux 2.25 ppm/an annoncés dans [monroe2015]).

Notre modèle est construit en utilisant les données de 1959 à 2014. Il est validé par comparaison aux données de 2015 à nos jours. Une extrapolation sur la période 2020-2025 est aussi réalisée afin de permettre une comparaison à de futures mesures.

1 - Gestion des dépendances

1.1 - Environnement

Nous utilisons Julia dans sa version 1.4.0, sur une architecture matérielle de type x86 (64 bits)

```
using InteractiveUtils
versioninfo()
```

```
Julia Version 1.4.0
Commit b8e9a9ecc6 (2020-03-21 16:36 UTC)
Platform Info:
  OS: Linux (x86_64-pc-linux-gnu)
  CPU: Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz
  WORD_SIZE: 64
  LIBM: libopenlibm
  LLVM: libLLVM-8.0.1 (ORCJIT, skylake)
Environment:
  JULIA_PROJECT = @.
```

Il nous faut maintenant charger l'environnement logiciel de notre étude, c'est à dire toutes les bibliothèques sur lesquelles elle s'appuie, dans les bonnes versions. La liste des dépendances directes est contenue dans le fichier `Project.toml`, et complétée par le fichier `Manifest.toml` (qui liste toutes les dépendances directes et indirectes, accompagnées de leurs numéros de version précis).

Toutes ces informations permettent au gestionnaire de paquets de re-créer un environnement logiciel identique à celui qui a été utilisé pour développer cette analyse.

```
using Pkg
Pkg.activate(@__DIR__)
Pkg.instantiate()
Pkg.status()
```

```
Project Exercice v0.1.0
Status `~/tmp/MOOC-RR/module3/exo3/Project.toml`
  [336ed68f] CSV v0.6.1
  [a93c6f00] DataFrames v0.20.2
  [82cc6244] DataInterpolations v2.0.0
  [38e38edf] GLM v1.3.9
  [cd3eb016] HTTP v0.8.13
  [9b87118b] PackageCompiler v1.1.1
  [91a5bcdd] Plots v0.29.9
  [44d3d7a6] Weave v0.9.4
```

1.2 - Chargement des dépendances

Tant que nous y sommes, profitons en pour charger dès maintenant les paquets dont nous aurons besoin.

```
import HTTP
import CSV
import DataInterpolations; const DI=DataInterpolations
using GLM
using Printf
using Dates
using DataFrames
using Statistics
using Plots; plotly()
```

2 - Données d'entrée

Nos données d'entrée proviennent du programme Scripps CO2. Nous fondons l'analyse sur le jeu de données contenant des observations hebdomadaires.

2.1 - Téléchargement

Le jeu de données est téléchargé une seule fois ; c'est une copie locale qui sert à réaliser l'analyse. Ceci permet de garantir la reproductibilité des données utilisées pour l'analyse, qui seront stockées dans git aux côtés du présent document.

Il est toutefois possible de forcer le téléchargement en positionnant la variable `force_download=true`. Ceci permet notamment d'actualiser le jeu de données.

```
const data_url = "https://scrippsco2.ucsd.edu/assets/data/atmospheric/stations/in_situ_co2/weekly
const data_file = "weekly_in_situ_co2_mlo.csv"
const force_download = false
```

```
if force_download || !isfile(data_file)
    println("Downloading data:")
    println("  url  = $data_url")
    println("  file = $data_file")

    open(data_file, "w") do f
        req = HTTP.request(:GET, data_url)
        @assert req.status == 200 "Error while downloading"
        write(f, req.body)
    end

    open("download.stamp", "w") do f
        write(f, string(today()))
    end
else
    println("Using local data:")
    println("  file = $data_file")
    println("  downloaded = ", readline("download.stamp"))
end
```

```
Using local data:
  file = weekly_in_situ_co2_mlo.csv
  downloaded = 2020-04-10
```

2.2 - Lecture

Les données d'entrée sont stockées au format CSV, et contiennent 44 lignes d'informations préliminaires que nous reproduisons ici, et qui seront sautées lors de la lecture des données.

```
skip = 44
open(data_file) do f
    for _ in 1:skip; println(readline(f)); end
end
```

```
"-------------------------------------------------------------------------------------------"
" Atmospheric CO2 concentrations (ppm) derived from in situ air measurements                "
" at Mauna Loa, Observatory, Hawaii: Latitude 19.5°N Longitude 155.6°W Elevation 3397m      "
"                                                                                           "
" Source: R. F. Keeling, S. J. Walker, S. C. Piper and A. F. Bollenbacher                   "
" Scripps CO2 Program ( http://scrippsco2.ucsd.edu )                                        "
" Scripps Institution of Oceanography (SIO)                                                 "
" University of California                                                                  "
" La Jolla, California USA 92093-0244                                                       "
"                                                                                           "
" Status of data and correspondence:                                                        "
"                                                                                           "
" These data are subject to revision based on recalibration of standard gases. Questions    "
" about the data should be directed to Dr. Ralph Keeling (rkeeling@ucsd.edu), Stephen Walker"
" (sjwalker@ucsd.edu) and Stephen Piper (scpiper@ucsd.edu), Scripps CO2 Program.            "
"                                                                                           "
" Baseline data in this file through 06-Feb-2020 from archive dated 06-Feb-2020 08:55:31    "
"                                                                                           "
"-------------------------------------------------------------------------------------------"
"                                                                                           "
" Please cite as:                                                                           "
"                                                                                           "
" C. D. Keeling, S. C. Piper, R. B. Bacastow, M. Wahlen, T. P. Whorf, M. Heimann, and       "
" H. A. Meijer, Exchanges of atmospheric CO2 and 13CO2 with the terrestrial biosphere and   "
" oceans from 1978 to 2000.  I. Global aspects, SIO Reference Series, No. 01-06, Scripps    "
" Institution of Oceanography, San Diego, 88 pages, 2001.                                   "
"                                                                                           "
" If it is necessary to cite a peer-reviewed article, please cite as:                       "
"                                                                                           "
" C. D. Keeling, S. C. Piper, R. B. Bacastow, M. Wahlen, T. P. Whorf, M. Heimann, and       "
" H. A. Meijer, Atmospheric CO2 and 13CO2 exchange with the terrestrial biosphere and       "
" oceans from 1978 to 2000: observations and carbon cycle implications, pages 83-113,       "
" in "A History of Atmospheric CO2 and its effects on Plants, Animals, and Ecosystems",     "
" editors, Ehleringer, J.R., T. E. Cerling, M. D. Dearing, Springer Verlag,                 "
" New York, 2005.                                                                           "
"                                                                                           "
"-------------------------------------------------------------------------------------------"
"                                                                                           "
"                                                                           "
" The data file below contains 2 columns indicaing the date and CO2         "
" concentrations in micro-mol CO2 per mole (ppm), reported on the 2008A     "
" SIO manometric mole fraction scale.  These weekly values have been        "
" adjusted to 12:00 hours at middle day of each weekly period as            "
" indicated by the date in the first column.                                "
```

Le fichier est structuré en deux colonnes :

- `date` : date de la mesure
- `co2` : concentration en CO2 (en ppm molaires)

```
data_raw = CSV.read(data_file; skipto=skip+1, header=[:date, :co2])
```

L'examen des premières et dernières lignes de données révèle qu'elles couvrent la période de fin mars 1958 jusqu'à nos jours.

```
info(data_raw)
```

3156 rows × 2 columns

	date	co2
	Dates.Date	Float64
1	1958-03-29	316.190000
2	1958-04-05	317.310000
3	1958-04-12	317.690000
...		
3154	2020-01-11	413.390000
3155	2020-01-25	413.360000
3156	2020-02-01	413.990000

2.3 - Vérification des données manquantes

Les relevés étant hebdomadaires, l'écart entre deux dates successives du jeu de données devrait être de 7 jours. Un point manquant provoque un écart de 14 jours, ce qui devrait être rattrapable dans le reste de l'analyse ; au delà, il faudra se poser des questions sur le traitement à apporter.

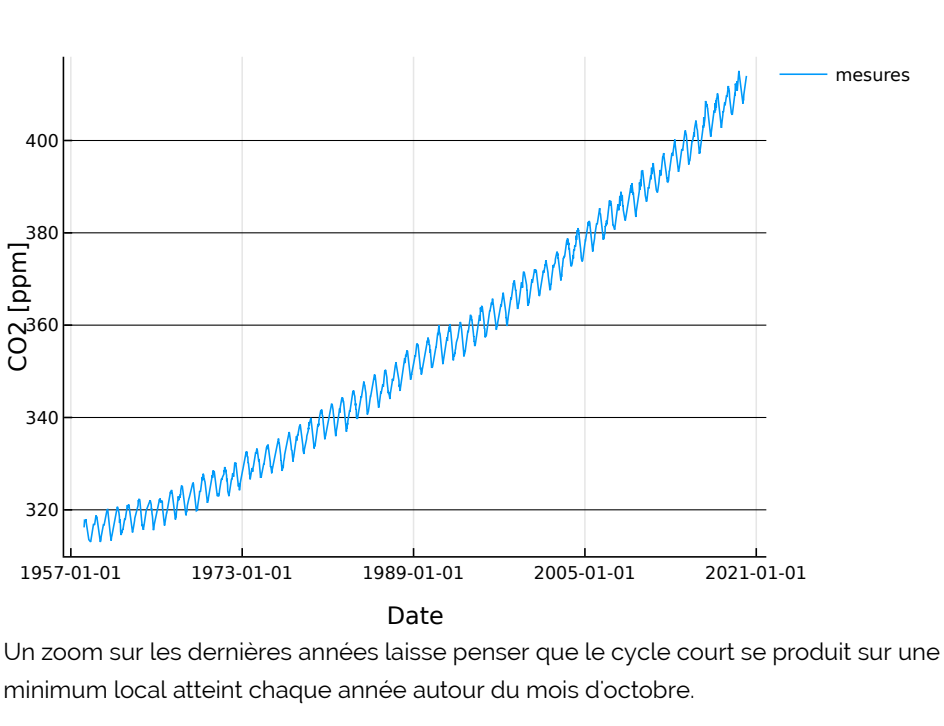
```
dates = data_raw.date
for i in 2:length(dates)
    if dates[i]-dates[i-1] > 14Days
        println("Missing data: ",
                dates[i-1], " - ", dates[i],
                "  (", dates[i]-dates[i-1], ")")
    end
end
```

```
Missing data: 1958-05-24 - 1958-07-05  (42 days)
Missing data: 1958-09-06 - 1958-11-08  (63 days)
Missing data: 1962-08-18 - 1962-09-15  (28 days)
Missing data: 1964-01-18 - 1964-05-30  (133 days)
Missing data: 1964-06-06 - 1964-06-27  (21 days)
Missing data: 1966-07-09 - 1966-08-06  (28 days)
Missing data: 1967-01-14 - 1967-02-04  (21 days)
Missing data: 1984-03-24 - 1984-04-28  (35 days)
Missing data: 2003-10-04 - 2003-10-25  (21 days)
Missing data: 2005-02-19 - 2005-03-26  (35 days)
Missing data: 2006-02-04 - 2006-02-25  (21 days)
Missing data: 2012-09-29 - 2012-10-20  (21 days)
```

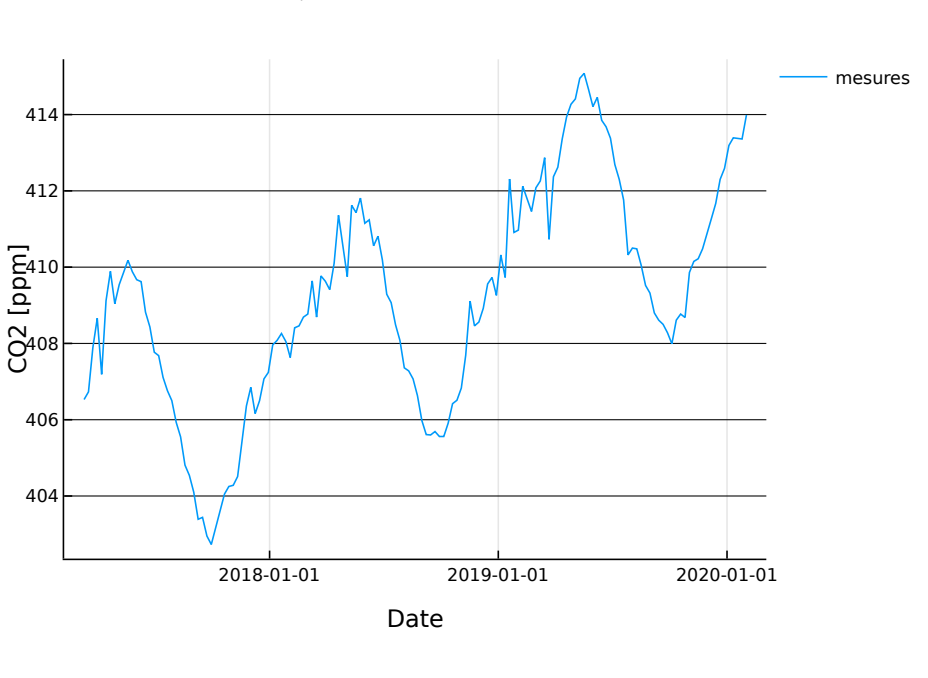
Il y a 12 périodes durant lesquelles les données sont manquantes, dont une en particulier ayant duré 19 semaines en 1964. Le traitement devra en tenir compte.

2.4 - Aperçu global des données

Une visualisation de l'ensemble des données semble montrer une augmentation tendancielle de la concentration en CO2, à laquelle se superpose une oscillation à plus haute fréquence.



Un zoom sur les dernières années laisse penser que le cycle court se produit sur une période annuelle, avec un minimum local atteint chaque année autour du mois d'octobre.



3 - Analyse

Dans cette analyse, nous allons tenter de séparer ces deux composantes : composante tendancielle "lisse" et composante oscillante de période annuelle. Plus précisément, en notant C la concentration en CO2 et t le temps, nous cherchons une approximation des mesures sous la forme :

$$C(t) \approx \theta(t) + \phi(t),$$

où $\phi(t)$ est une fonction périodique, de période 1 an, donnant la *forme* de la variation de la concentration en CO2 sur des échelles de temps courtes. $\theta(t)$ est une fonction très régulière, idéalement un polynôme d'ordre bas, donnant la *tendance* de la variation de concentration en CO2 en temps long.

La démarche que nous suivons est globalement la suivante : le jeu de données va être découpé en périodes annuelles. Chacune de ces périodes annuelles sera traitée (indépendamment des autres) afin d'en extraire une composante lisse, et une composante périodique.

Si notre hypothèse est correcte, les composantes périodiques de chaque année devraient être relativement comparables les unes aux autres, et pouvoir être approchées par leur moyenne : ce comportement annuel moyen constituera notre composante oscillante $\phi(t)$.

On pourra ensuite obtenir la composante lisse tendancielle en éliminant la composante oscillatoire moyenne du signal d'origine :

$$\theta(t) = C(t) - \phi(t).$$

La périodicité des données (hebdomadaire) n'étant que peu adaptée à un découpage annuel, nous allons commencer par interpoler les données à une maille journalière. Ceci nous permettra de découper le jeu de données en années.

3.1 - Travaux sur les dates

Il est plus simple d'interpoler entre deux nombres qu'entre deux dates. Dans la suite, nous adopterons une convention selon laquelle chaque date peut être représentée par le nombre de jours qui la sépare de la première mesure :

```
date2num(d::Date) = Dates.value(d - data_raw.date[1])
num2date(n::Int)  = data_raw.date[1] + n * Days
```

Par exemple, pour les premières mesures :

3 rows × 2 columns

	date	date_num
	Dates…	Int64
1	1958-03-29	0
2	1958-04-05	7
3	1958-04-12	14

Afin de comparer des données année par année, nous allons aussi enrichir les données avec de nouvelles représentations de la date : une date peut être décomposée comme un couple `(year, day)` dans lequel

- `year` représente l'année
- `day` représente l'indice du jour dans l'année (entre 0 et 365).

Dans ce formalisme, le 1er janvier 2020 est représenté par le couple `(year=2020, day=0)`. Le 31 décembre 1983 est représenté par le couple `(year=1983, day=364)`.

```
dayinyear(date::Date) = Dates.value(date - Date(year(date)))
dayinyear(num::Int)   = dayinyear(num2date(num))

let d = Date("1983-12-31")
  year(d), dayinyear(d)
end
```

```
(1983, 364)
```

Enfin, prévenons dès maintenant que tout le code de l'analyse fonctionne en présence d'années bissextiles, mais rien n'a été fait pour les traiter à part : l'impact, de l'ordre de 1/365 une année sur 4, a été jugé négligeable a priori.

3.2 - Interpolation à la maille journalière

On construit un interpolateur linéaire basé sur les mesures de CO2 en fonction de la "date numérique". C'est le paquet Julia `DataInterpolations` qui se charge d'effectuer le gros du travail.

```
interp = DI.LinearInterpolation(data_raw.co2, date2num.(data_raw.date));
```

Nous allons profiter de la construction des données interpolées pour gérer le problème des données manquantes : nous n'interpolerons aucune donnée dans les "trous" de 3 semaines ou plus. La construction de ce nouveau jeu de données interpolées est aussi l'occasion d'enrichir les formats de représentation des dates. Nous avons maintenant 5 colonnes dans notre jeu de données interpolé :

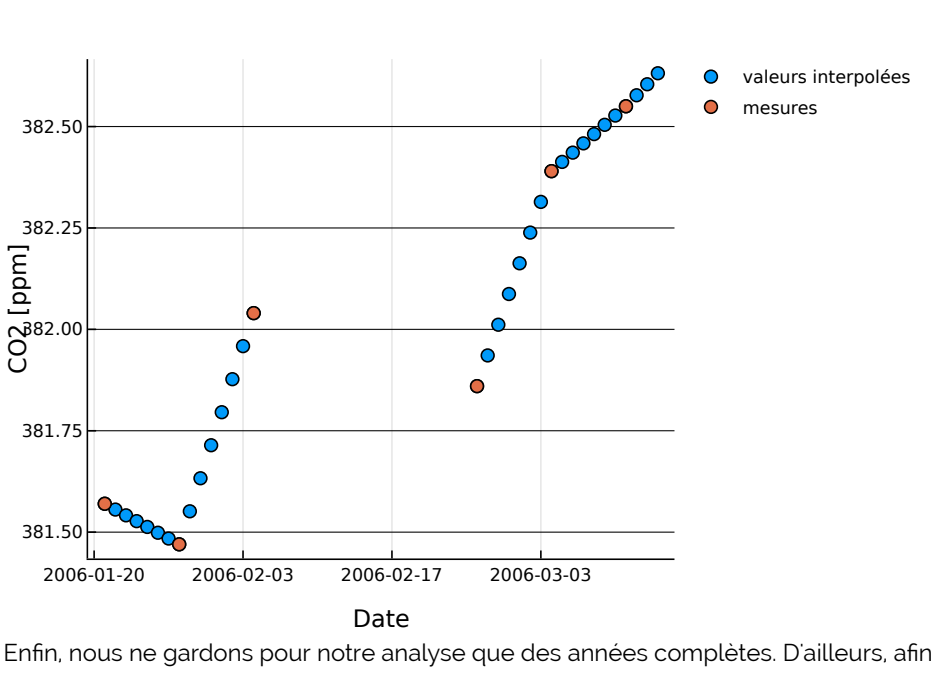
- `date` : date identifiant le jour de la mesure (ou de la valeur interpolée)
- `co2` : valeur de la mesure de CO2 (ou de l'interpolation)
- `date_num` : date convertie en nombre
- `year` : composante de la date identifiant l'année
- `day` : composante de la date identifiant le jour dans l'année (entre 0 et 365)

```
data_interp = DataFrame(date_num=Int[], co2=Float64[], date=Date[], year=Int[], day=Int[])
for i in 2:length(dates)
    if dates[i]-dates[i-1] > 14Days
        # pas d'interpolation : seule la date de gauche est incluse
        range = dates[i-1] => dates[i-1]
    else
        # Interpolation entre le début de la période et
        # la fin (exclue car traitée en tant que début de
        # la prochaine période)
        range = dates[i-1] => dates[i]- 1Days
    end

    # Pour chaque jour dans la période considérée,
    # on ajoute une nouvelle ligne de données en interpolant
    for date_num in date2num(first(range)):date2num(last(range))
        date = num2date(date_num)
        push!(data_interp, (date_num = date_num,
                            date = date,
                            year = year(date),
                            day  = dayinyear(date),
                            co2  = interp(date_num)))
    end
end

info(data_interp)
```

22132 rows × 5 columns

	date_num	co2	date	year	day
	Int64	Float64	Dates.Date	Int64	Int64
1	0	316.190000	1958-03-29	1958	87
2	1	316.350000	1958-03-30	1958	88
3	2	316.510000	1958-03-31	1958	89
...					
22130	22586	413.720000	2020-01-29	2020	28
22131	22587	413.810000	2020-01-30	2020	29
22132	22588	413.900000	2020-01-31	2020	30

En zoomant sur les données interpolées autour de l'une des périodes de données manquantes, on observe bien le résultat attendu : une interpolation linéaire journalière lorsque les données sont disponibles, mais aucune interpolation lorsque les données sont manquantes.



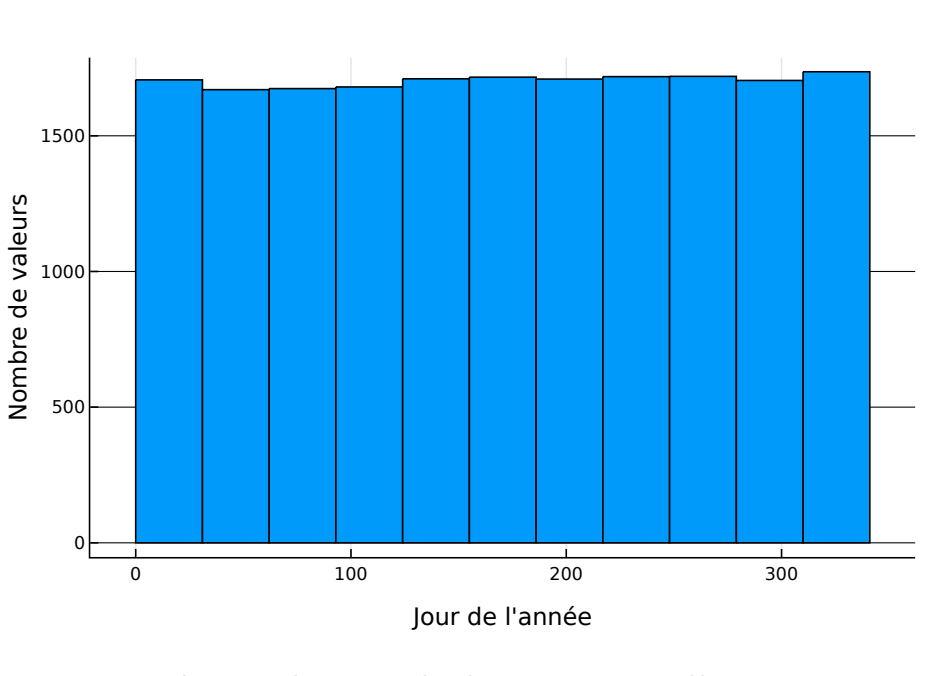
Enfin, nous ne gardons pour notre analyse que des années complètes. D'ailleurs, afin de tester la validité de nos résultats, nous n'allons réaliser l'analyse que sur une fraction des données ; nous garderons les 5 dernières années comme données de test. Les données sur lesquelles portera l'analyse couvrent donc la période 1959-2014.

```
firstyear = minimum(data_interp.year)+1 # Première année incomplète
lastyear  = maximum(data_interp.year)-6 # Dernière année incomplète + 5 années de test
idx = (data_interp.year .>=firstyear) .& (data_interp.year.<=lastyear)
data = data_interp[idx, :];

info(data)
```

20100 rows × 5 columns

	date_num	co2	date	year	day
	Int64	Float64	Dates.Date	Int64	Int64
1	278	315.231429	1959-01-01	1959	0
2	279	315.235714	1959-01-02	1959	1
3	280	315.240000	1959-01-03	1959	2
...					
20098	20729	399.270000	2014-12-29	2014	362
20099	20730	399.400000	2014-12-30	2014	363
20100	20731	399.530000	2014-12-31	2014	364

Sur ces années complètes, la composante `day` de la date devrait être équirépartie entre 0 et 365, ce qui est globalement le cas. Les données manquantes n'ont donc pas d'impact significatif de ce point de vue là.



3.3 - Analyse des variations annuelles

Pour chaque année, on commence par tenter d'extraire la composante oscillante de la mesure. Si l'on note $C_a(d)$ la concentration en CO2 le jour numéro d de l'année a, on cherche à écrire :

$$\forall d \in 0 \ldots 365, \quad C_a(d) = \underbrace{\theta_a(d)}_{\text{tendance locale}} + \underbrace{\phi_a(d)}_{\text{forme locale}}$$

ce qui correspond à une version locale (pour l'année a) de l'expression globale cherchée pour la variation de concentration en CO2 : θ_a et ϕ_a donnent respectivement la tendance et la forme de la concentration pour l'année a.

Cherchant une tendance d'ordre aussi bas que possible, nous supposons que θ_a peut être approchée par un modèle affine à cette échelle de temps courts :

$$\theta_a(d) = \alpha_a + \beta_a\, d.$$

Pour que la forme ϕ_a soit périodique, il faut que cette fonction prenne la même valeur en début d'année qu'en fin d'année : $\phi(0) = \phi(365)$. Afin de définir la constante α de manière unique, on fixe de plus :

$$\phi(0) = \phi(365) = 0.$$

On obtient donc

$$\alpha_a = C_a(0),$$

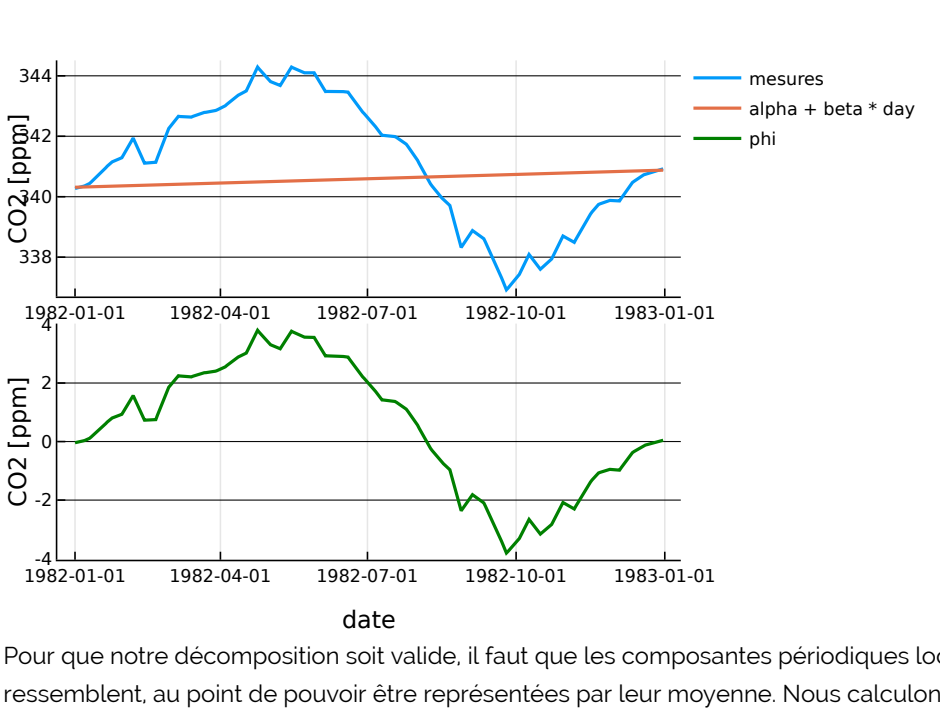
$$\beta_a = \frac{C_a(365) - C_a(0)}{365},$$

$$\phi_a(d) = C_a(d) - \alpha_a - \beta_a d.$$

En pratique, plutôt que des valeurs ponctuelles $C_a(0)$ et $C_a(365)$, on prend plutôt des valeurs (notées C_0 et C_1 dans le code) moyennées sur les 7 premiers et 7 derniers jours de l'année.

```
data.phi   = copy(data.co2)
data.alpha = zero(data.co2)
data.beta  = zero(data.co2)
for year in unique(data.year)
    idx = (data.year .== year)
    C₀ = data.co2[idx .& (data.day .< 7)   ] |> mean
    C₁ = data.co2[idx .& (data.day .> 358)] |> mean
    α = C₀
    β = (C₁ - C₀) / 365

    data.alpha[idx] .= α
    data.beta[idx]  .= β
    data.phi[idx]   .-= α .+ β*data.day[idx]
end
```

Examinons par exemple l'effet de ce traitement sur les données interpolées de l'année 1982. On voit, sur la figure du haut, les mesures brutes ainsi que la tendance locale (affine). Sur la figure du bas, la composante périodique locale vérifie bien les contraintes demandées, avec ses valeurs nulles aux bords.



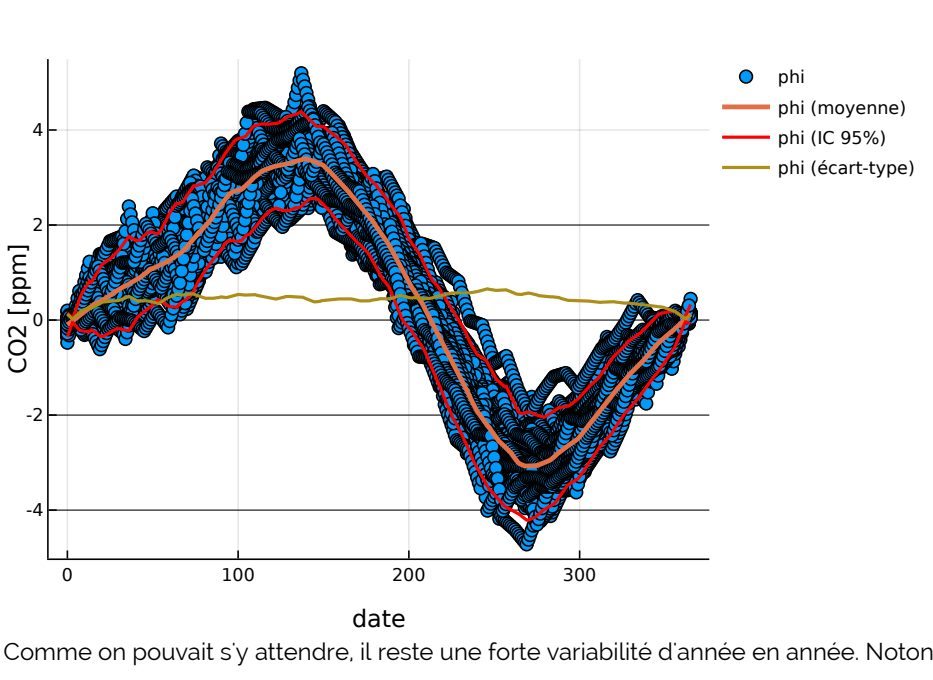
Pour que notre décomposition soit valide, il faut que les composantes périodiques locales de chaque année se ressemblent, au point de pouvoir être représentées par leur moyenne. Nous calculons donc cette moyenne (et l'écart-type associé) pour toutes les années du jeu de données d'étude.

```
avg = by(data, :day, :phi=>mean, :phi=>std)
info(avg)
```

366 rows × 3 columns

	day	phi_mean	phi_std
	Int64	Float64	Float64
1	0	-0.102212	0.125698
2	1	-0.073884	0.088030
3	2	-0.043949	0.048068
...			
364	363	0.058941	0.055141
365	364	0.091555	0.077335
366	365	0.126676	0.100662

Et nous traçons l'ensemble des composantes oscillantes locales aux côtés de cette moyenne.



Comme on pouvait s'y attendre, il reste une forte variabilité d'année en année. Notons toutefois que l'écart-type, bien qu'important, reste un ordre de grandeur en dessous des valeurs extrêmes, ce qui permet d'espérer que ce profil moyen est suffisamment représentatif pour permettre d'extraire la composante tendancielle lisse des données.

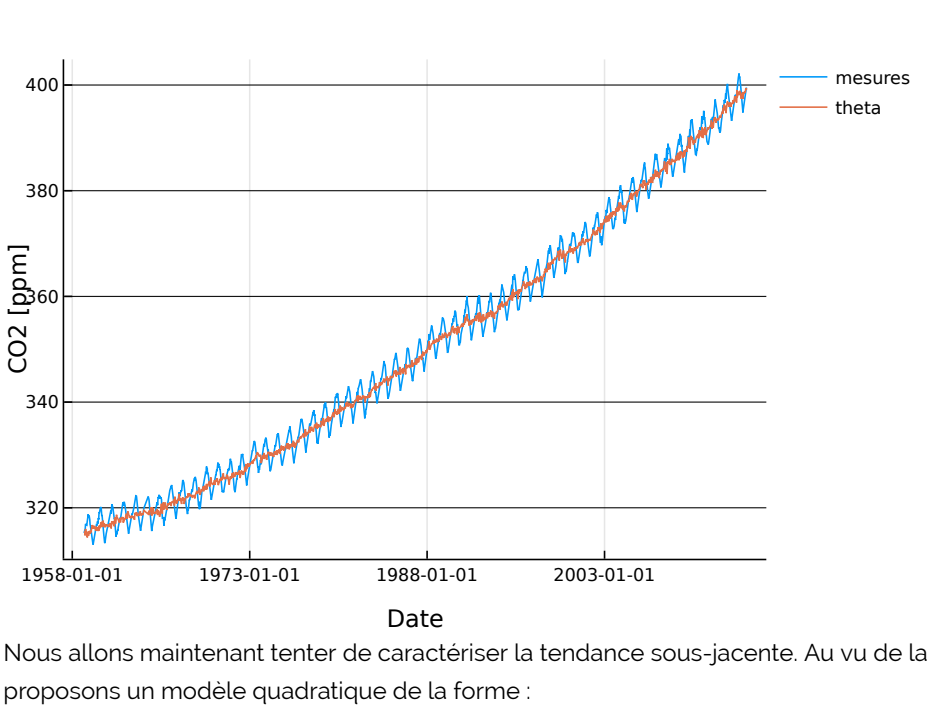
3.4 - Analyse des variations tendancielles

Nous sommes maintenant prêts à extraire la composante tendancielle des mesures. Il suffit pour cela de retrancher aux données brutes la composante oscillante moyenne : d'après notre modèle, on a en effet

$$\theta(t) = C(t) - \phi(t).$$

```
data = join(data, avg, on=:day)
data.theta = data.co2 .- data.phi_mean;
```

Même s'il reste des oscillations locales, nous constatons tout de même que la composante tendancielle est devenue suffisamment lisse pour récupérer une forme de monotonie.



Nous allons maintenant tenter de caractériser la tendance sous-jacente. Au vu de la courbe (convexe), nous proposons un modèle quadratique de la forme :

$$\theta(t) = \alpha + \beta t + \gamma t^2$$

Le paquet Julia `GLM` permet de *fitter* des modèles linéaires (ou modèles linéaires généralisés). Nous l'utilisons ici pour estimer les paramètres de notre modèle. Afin de ne pas considérer un nombre artificiellement élevé de points de mesure, l'estimation est réalisée sur un sous-échantillonnage du jeu de données, comprenant un point (moyen) par an, ce qui correspond à l'échelle de temps que nous n'avons pas encore capturé dans la composante oscillante.

```
sample = by(data, :year, theta = :theta=>mean, date_num = :date_num=>mean)
model  = GLM.lm(@formula(theta ~ date_num + date_num^2), sample)
```

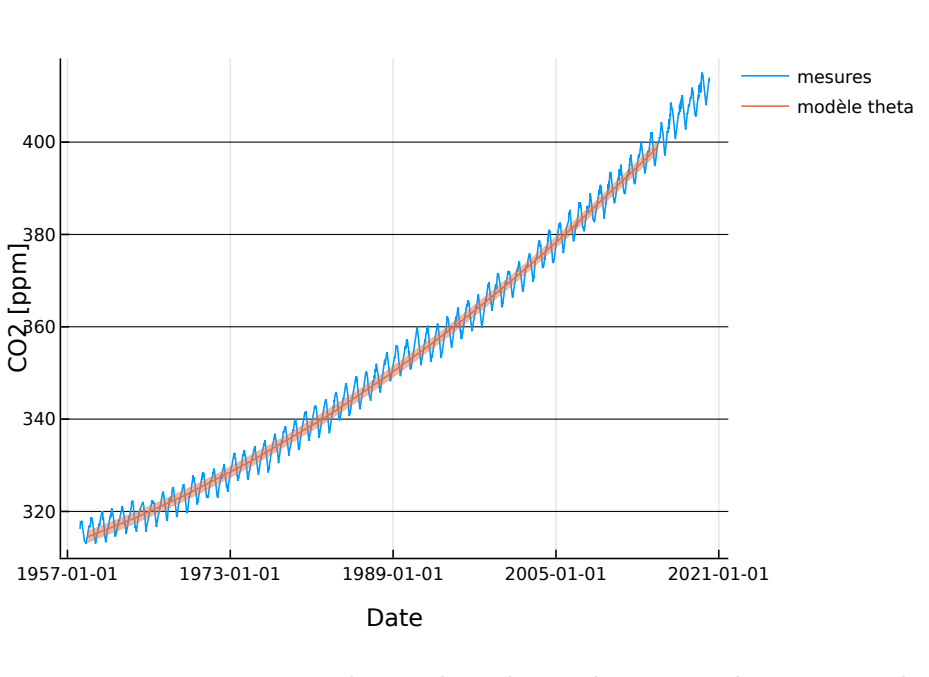
```
StatsModels.TableRegressionModel{GLM.LinearModel{GLM.LmResp{Array{Float64,1}},GLM.DensePredChol{

theta ~ 1 + date_num + :(date_num ^ 2)

Coefficients:
──────────────────────────────────────────────────────────────────────────────────────
                  Estimate   Std. Error    t value  Pr(>|t|)     Lower 95%    Upper 95%
──────────────────────────────────────────────────────────────────────────────────────
(Intercept)   313.862        0.263289    1192.08      <1e-99   313.334       314.39
date_num        0.00221635   5.77649e-5    38.3685    <1e-39     0.00210049    0.00233221
date_num ^ 2    9.10059e-8   2.66606e-9    34.135     <1e-37     8.56585e-8    9.63533e-8
──────────────────────────────────────────────────────────────────────────────────────
```

Le modèle obtenu semble correspondre assez bien aux données, avec une incertitude de l'ordre de quelques pourcents sur l'estimation des paramètres α et β.

```
# Paramètres du modèle et intervalle de confiance à 95%
α,  β,  γ  = coef(model)
α1, β1, γ1 = coef(model)-2*stderror(model)
α2, β2, γ2 = coef(model)+2*stderror(model)

# Taux d'accroissement
τ(d) = (β + γ * date2num(d)) * 365

τ1 = τ(Date("1959"))
τ2 = τ(Date("2015"))
```

Ce modèle caractérise en particulier une hausse tendancielle de la concentration de CO2 atmosphérique de l'ordre de τ_1 = 0.82 ppm/an en 1959 et τ_2 = 1.50 ppm/an en 2015. Si la valeur de 1959 correspond assez bien au taux de 0.75 ppm/an déterminé dans [monroe2015], la valeur que nous trouvons pour 2015 est sous-évaluée de plus de 30% par rapport aux 2.25 ppm/an calculés par Monroe.

Par ailleurs, s'il est clair que la tendance est à l'augmentation, on voit toutefois que l'incertitude sur les paramètres n'est pas complètement négligeable. L'incertitude sur β est en particulier de nature à engendrer une perte de prédictibilité du modèle en temps long.



3.5 - Reconstruction du signal complet et prédiction

Nous avons maintenant tous les éléments nécessaires afin de reconstruire l'évolution de la concentration en CO2 selon notre modèle :

$$C(t) = \theta(t) + \phi(t).$$

Les 5 dernières années du jeu de données brut n'ont pas servi à calibrer notre modèle ; nous allons les utiliser comme données de test. Nous extrapolons aussi les valeurs des 5 prochaines années. Pour toute cette période, nous reconstruisons les valeurs de concentration en CO2 tous les 10 jours.

```
date_num = date2num(Date(lastyear)):10:date2num(today()+5Years)
prediction = DataFrame(date_num=date_num,
                       date=num2date.(date_num),
                       day=dayinyear.(date_num))
info(prediction)
```

413 rows × 3 columns

	date_num	date	day
	Int64	Dates.Date	Int64
1	20367	2014-01-01	0
2	20377	2014-01-11	10
3	20387	2014-01-21	20
...			
411	24467	2025-03-24	82
412	24477	2025-04-03	92
413	24487	2025-04-13	102

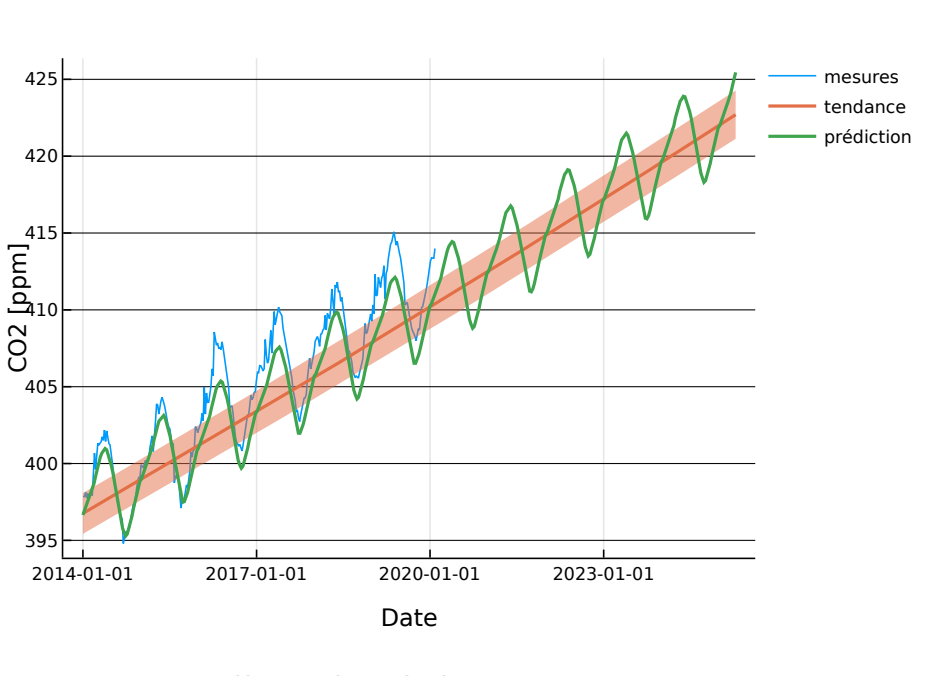
Le modèle `GLM` précédemment calibré est utilisé pour prédire $\theta(t)$. On y ajoute la fonction de forme annuelle $\phi(t)$ par périodicité.

```
pred = predict(model, prediction, interval=:prediction, level=0.95)
prediction.theta  = pred.prediction
prediction.theta1 = pred.prediction .- pred.lower
prediction.theta2 = pred.upper .- pred.prediction

prediction = join(prediction, avg, on=:day)

prediction.co2 = prediction.theta .+ prediction.phi_mean
info(prediction)
```

413 rows × 9 columns

	date_num	date	day	theta	theta1	theta2	phi_mean	phi_std	co2
	Int64	Dates.Date	Int64	Float64?	Float64	Float64	Float64	Float64	Float64
1	20367	2014-01-01	0	396.753169	1.330117	1.330117	-0.102212	0.125698	396.650956
2	20377	2014-01-11	10	396.812412	1.330441	1.330441	0.198187	0.220453	397.010599
3	20387	2014-01-21	20	396.871673	1.330766	1.330766	0.403495	0.374229	397.275168
...									
411	24467	2025-03-24	82	422.568862	1.571011	1.571011	2.023296	0.459201	424.592158
412	24477	2025-04-03	92	422.635568	1.571895	1.571895	2.538372	0.467285	425.173939
413	24487	2025-04-13	102	422.702291	1.572782	1.572782	2.751356	0.541977	425.453647

On voit que la forme annuelle semble bien reproduite sur les 5 premières années, pour lesquelles il est possible de comparer les prédictions avec les mesures réelles. En revanche, la tendance ne colle que sur les deux premières années d'extrapolation ; on observe un décalage significatif et croissant ensuite.



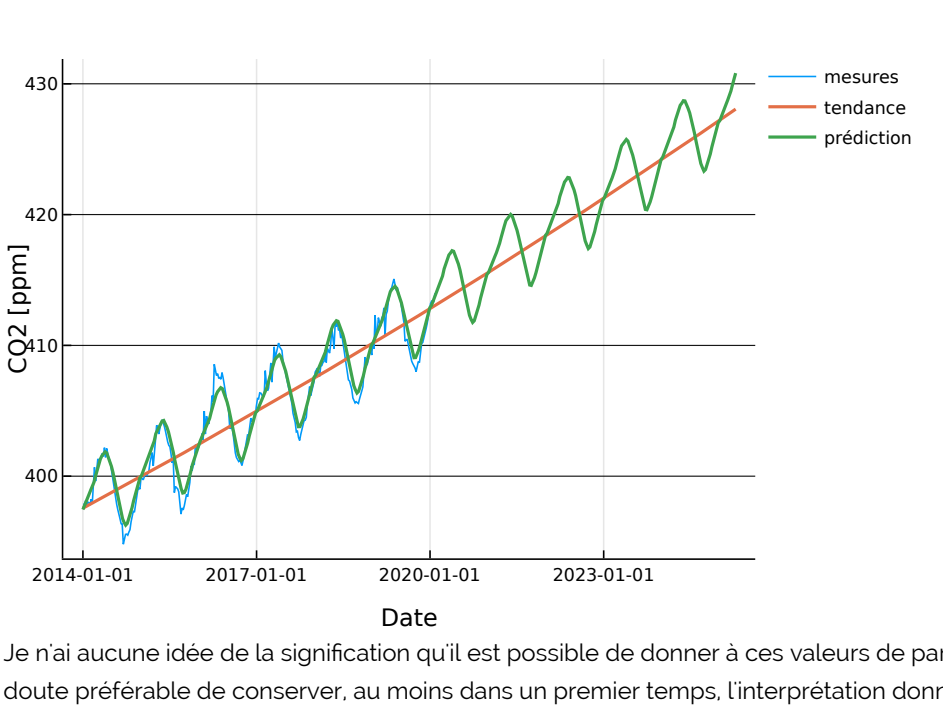
3.6 - Pour aller plus loin

En utilisant des techniques de calage de modèle plus complexes, il est possible de mieux caractériser la tendance en temps longs :

```
model2 = glm(@formula(theta ~ date_num + date_num^2), sample,
             InverseGaussian(), InverseSquareLink())
```

```
StatsModels.TableRegressionModel{GLM.GeneralizedLinearModel{GLM.GlmResp{Array{Float64,1},Distribu

theta ~ 1 + date_num + :(date_num ^ 2)

Coefficients:
───────────────────────────────────────────────────────────────────────────────────────
                  Estimate    Std. Error    z value  Pr(>|z|)      Lower 95%     Upper 95%
───────────────────────────────────────────────────────────────────────────────────────
(Intercept)     1.02097e-5   1.68812e-8    604.796      <1e-99    1.01766e-5    1.02428e-5
date_num       -1.73581e-10  3.46014e-12  -50.166      <1e-99   -1.80363e-10  -1.668e-10
date_num ^ 2   -8.3699e-16   1.52956e-16   -5.47208    <1e-7    -1.13678e-15  -5.37201e-16
───────────────────────────────────────────────────────────────────────────────────────
```

En reprenant l'analyse précédente, ce nouveau modèle donne les prédictions suivantes, qui collent quasi-parfaitement aux mesures dans la période de test :



Je n'ai aucune idée de la signification qu'il est possible de donner à ces valeurs de paramètres, aussi est-il sans doute préférable de conserver, au moins dans un premier temps, l'interprétation donnée par le modèle quadratique simple (quoi que donnant des prédictions plus éloignées des données).

4 - Conclusions

Dans cette étude, nous avons tenté de proposer une analyse (reproductible) de la courbe de Keeling, qui suit les variations de la concentration atmosphérique en CO2 depuis 1959 à Hawaï. La courbe est décomposée en une composante tendancielle à laquelle se superposent des variations périodiques annuelles. La coomposante tendancielle montre clairement un comportement d'augmentation sur le long terme. Une caractérisation de cette tendance sous forme de modèle quadratique fait apparaître des taux d'accroissement de l'ordre de 0.8 ppm/an en 1959, qui s'accélèrent pour monter à environ 1.5 ppm/an en 2015. Les taux d'accroissement trouvés ici peuvent être comparés aux taux de 0.75 ppm/an (1959) et 2.25 ppm/an (2015) trouvés dans la littérature [monroe2015]. Il serait intéressant de pousser la comparaison entre ces deux travaux plus avant, afin de comprendre l'origine des écarts significatifs (de l'ordre de 30%) sur les taux d'accroissements de 2015.

Par ailleurs, nous avons dans cette étude proposé une validation du modèle utilisant les 5 dernières années de mesures comme données de test. Une extrapolation sur les 5 prochaines années (2020-2025) est aussi proposée. Il convient de noter ici que l'utilisation de techniques poussées de calage de paramètre semble de prime abord donner des estimations plus prédictives. Une autre perspective de ce travail pourrait être d'interpréter et valider l'utilisation de telles techniques dans ce contexte.

Published from exercice.jmd using Weave.jl on 2020-04-18.