

Aborder la question environnementale de manière transversale dans des cours d'informatique

Pablo de Oliveira Castro (pablo.oliveira@uvsq.fr)

2024/06/21, Atelier WID, GS ISN Paris-Saclay

Université de Versailles, Université Paris-Saclay, Li-PaRAD

2019 - Je prends conscience de l'urgence climatique

- Réoriente une partie de mes enseignements / recherches autour de la question environnementale
- Sujet nouveau pour moi, complexe, et interdisciplinaire
- Pas à pas → J'inclus des points de vue sur le sujet dans les cours existants

Plan

- Cours M1, calcul numérique pour modéliser le climat
- Cours L3, optimisation de l'énergie consommée
- Conclusion et critiques

- **Cours en M1 Calcul Haute Performance Simulation, GS SIS**
 - T. Dufaud, P. de Oliveira Castro
 - 30 étudiants, 10 semaines - 1.5h CM + 3h TD
 - Introduire les algorithmes pour la simulation numérique

- **Syllabus**
 - Calcul en arithmétique flottante et erreur numérique
 - Algèbre linéaire dense
 - Algèbre linéaire creuse
 - Méthodes itératives
 - **Intégration numérique** ← modèle climatique

Séquence pédagogique

- Cours

- Introduction à l'intégration numérique
- Schéma d'Euler explicite et implicite
- Analyse d'erreur et convergence

- TD (séance 1)

- Preuve de la convergence de la méthode d'Euler
- Implémentation du modèle SimClimat en C

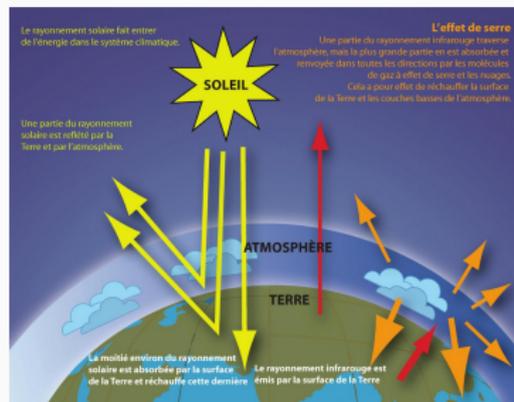
- TD (séance 2)

- Validation du modèle et discussion
- Erreur numérique vs. erreur de troncature

<https://github.com/pablooliveira/cours-calcul-numerique>

Modèle SimClimat (simplifié)

- Modèle pédagogique développé par C. Risi (LMD)
- Simplifié avec l'aide d'Alexis Tantet (LMD) dans le cadre de l'atelier d'écologie politique francilien (Écopolien)
- **Radiatif**: bilan énergétique des rayonnements entrants et sortants
- **Global**: nous considérons la terre comme un seul point
- **Forçages** : CO₂, albedo, soleil
- **Rétroactions** : vapeur d'eau



GIEC rapport WP1, 2007, p. 104

Mise en équation

$$P_{in} = (1 - \alpha) \frac{S_0}{4} \quad \text{rayonnement solaire (constant)}$$

$$P_{out} = (1 - G(c_{ges}, T)) \sigma T^4 \quad \text{infrarouge (Stefan-Boltzmann)}$$

- une partie du rayonnement infrarouge est conservé par l'effet de serre
- G est fonction de la concentration de GES et de la température
- Forçage CO_2 entropique et modèle linéaire pour la vapeur d'eau.

$$dT = (P_{in} - P_{out}) \frac{dt}{C}$$

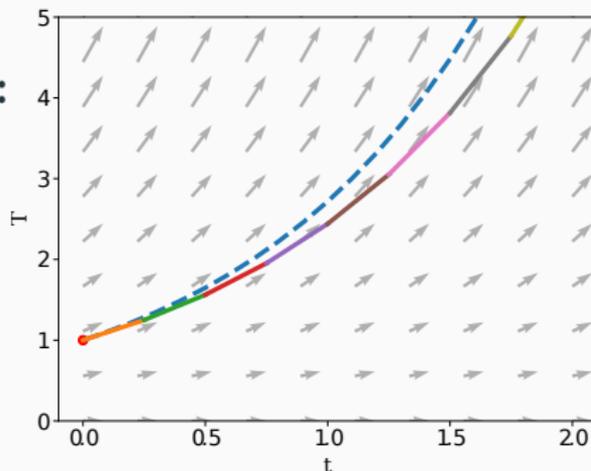
C modélise l'inertie de changement de température

Intégration numérique

Euler explicite

$$T_{n+1} = T_n + \Delta t \frac{dT}{dt}$$

```
def euler(dT, T0, t0, h, n):  
    T, t = T0, t0  
    for i in range(n):  
        T = T + h*dT(T, t)  
        t = t + h  
    return T
```



Travailler avec les rapports du GIEC

- Initier les étudiants à la **littérature scientifique**.
- Rapport du WP1 du GIEC AR5 est intimidant (> 2000 pages) ... mais il comporte des FAQ très accessibles.
- La FAQ 12.1 aborde les questions de validité et incertitude des modèles numériques.
- À la lumière de cette FAQ, les étudiants font une **analyse critique de leur modèle**.

Frequently Asked Questions

FAQ 12.1 | Why Are So Many Models and Scenarios Used to Project Climate Change?

Future climate is partly determined by the magnitude of future emissions of greenhouse gases, aerosols and other natural and man-made forcings. These forcings are external to the climate system, but modify how it behaves. Future climate is shaped by the Earth's response to those forcings, along with internal variability inherent in the climate system. A range of assumptions about the magnitude and pace of future emissions helps scientists develop

Comparaison aux modèles du GIEC AR5

- Modèle conservateur dans les fourchettes bases du GIEC
- Très simplifié :
 - global
 - modèle linéaire pour la pression de vapeur
 - albedo constant
- Boucles de rétroaction non-modélisées (fonte des glaces, incendies, etc.)

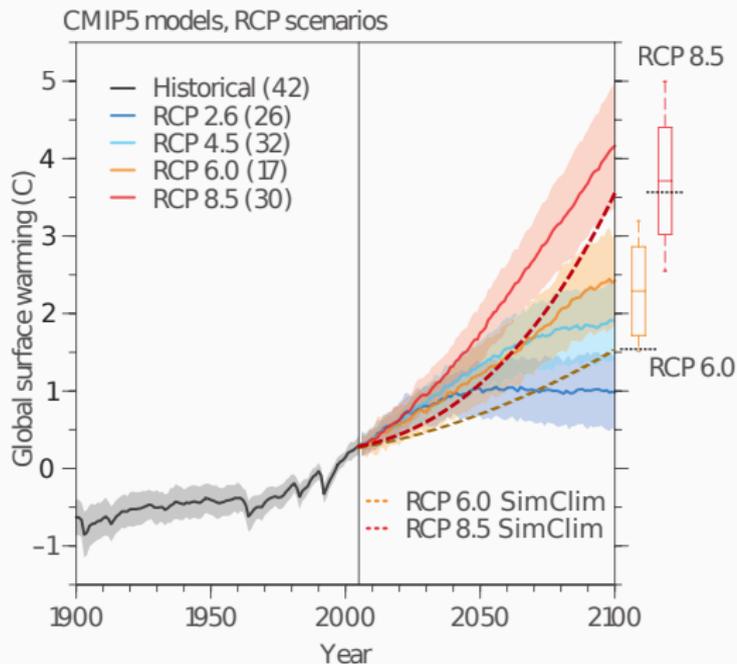


Figure originale : Knutti & Sedláček, Nature Climate Change (2013)

- Sujet **motivant** pour les étudiants
- Permet d'explorer les questions de validité, d'erreur et d'incertitude centrales en simulation
- Faut-il prolonger la discussion autour de l'urgence climatique ?
 - Actuellement, 15 minutes de discussion dans un **mode horizontal** sont proposées
 - Selon les promotions, difficultés pour s'emparer de ce format

- **Cours en L3 Informatique à l'UVSQ**
 - E. Frenoux, S. Lopes, P. de Oliveira Castro, Y. Strozecki
 - 100-120 étudiants, 12 semaines - 1.5h CM + 3h TD
- **Objectifs :**
 - Programmation système - apprentissage du langage Rust
 - Gestion efficace et correcte des ressources : **mémoire**, temps, **énergie**
 - Sensibilisation à l'impact environnementale du numérique
 - Optimisation du coût énergétique des calculateurs

Syllabus

- Langage Rust et outils de développement
- Garanties du modèle mémoire (emprunts, propriété)
- Extensions et POO
- Gestion avancée de la mémoire
- Éco-informatique (E. Frenoux)
- Consommation d'énergie des calculateurs
- Optimisation de code pour l'énergie

Comment mesurer l'énergie d'un calcul ?

- **Watt-mètre**

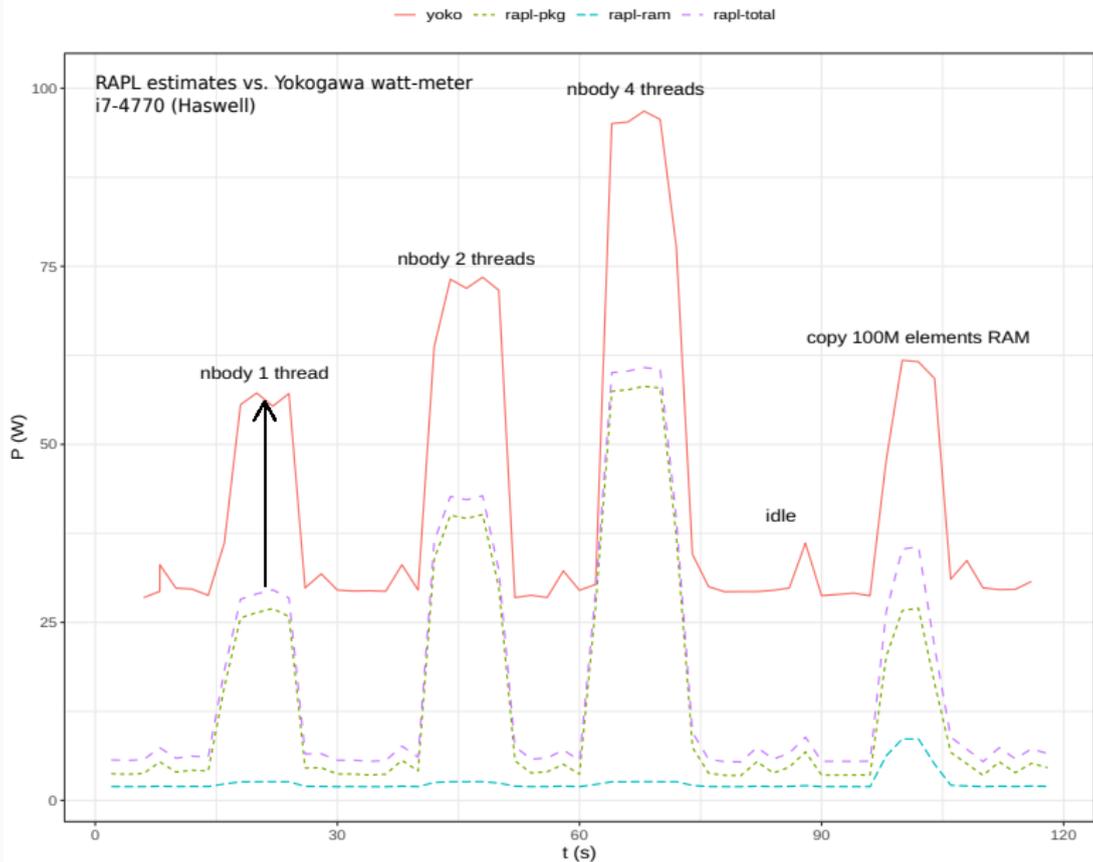
- mesure précise de la consommation totale
- résolution d'échantillonnage lente (environ 1 seconde)
- difficile à utiliser en TD

- **Compteurs d'énergie intégrés** (rapl, nvml)

- facilement accessibles (linux-perf) et taux d'échantillonnage élevé
- ne capturent pas la consommation totale du système
- utilisent des modèles d'estimation de puissance
- Outils standardisés pour Windows / OSX ?



Calibrage : RAPL vs. Yokogawa Watt-mètre



TD: Multiplication de Matrices

```
pub fn multiply(a: &Matrix, b: &Matrix) -> Matrix {
    assert!(a.n == b.n);
    let n = a.n;
    let mut c = Matrix::zero(n);

    for i in 0..n {
        for j in 0..n {
            for k in 0..n {
                c[(i, j)] += a[(i, k)] * b[(k, j)];
            }
        }
    }

    c
}
```

https://github.com/uvsq-pef/td_mult_mat/tree/solution

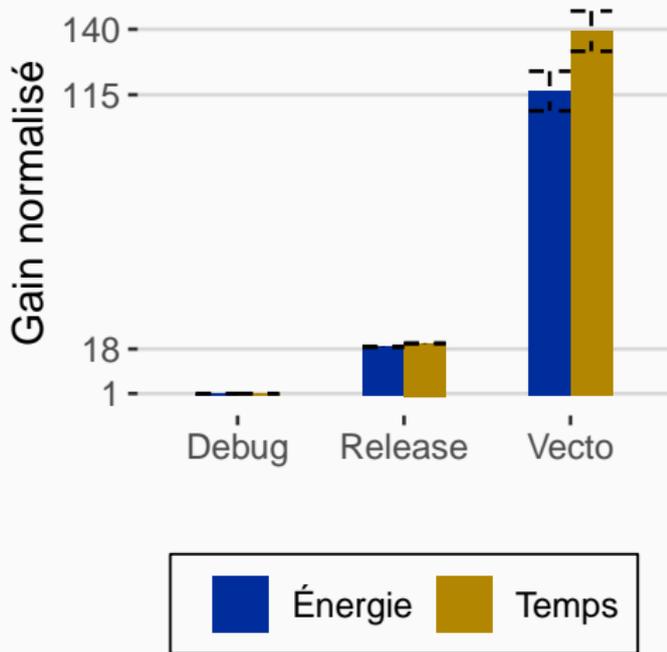
Vectorisation

```
for i in 0..n {  
  for k in 0..n {  
    for j in 0..n {  
      c[(i, j)] += a[(i, k)] * b[(k, j)];  
    }  
  }  
}
```

- Intervertir les boucles `k` et `j`
- Permet la **vectorisation** (plus de stride)
- Instructions SIMD très efficaces (`mulpd`, `vfmaddpd`)
- Ici le compilateur Rust ne le fait pas automatiquement

Impact des optimisations du compilateur

- Release = optimisations du compilateur
- Importance de la **vectorisation**
- **Même ordre de grandeur** pour les gains d'énergie et temps
- Discussion sur l'impact du langage / modèle d'exécution (interprété, compilé, JIT)



Taille : 1024 x 1024, 1M éléments

i7-9850H @ 2.60GHz 6 cœurs, 12 Mo cache L3

Problème de localité pour les grandes matrices

- Pour n grand, les matrices ne tiennent pas en cache
- Pas de réutilisation des données
 - accès mémoire distants répétés
 - augmentation du temps et de l'énergie

→ Multiplication par blocs

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

- Taille de bloc pour que A_{ij} et B_{ij} tiennent en cache.

$$\begin{bmatrix} C_1 \\ \vdots \\ C_t \end{bmatrix} = \begin{bmatrix} A_1 \\ \vdots \\ A_t \end{bmatrix} \times B$$

- On découpe les matrices C et A en tranches que l'on répartit sur un groupe de threads. (bibliothèque Rayon)

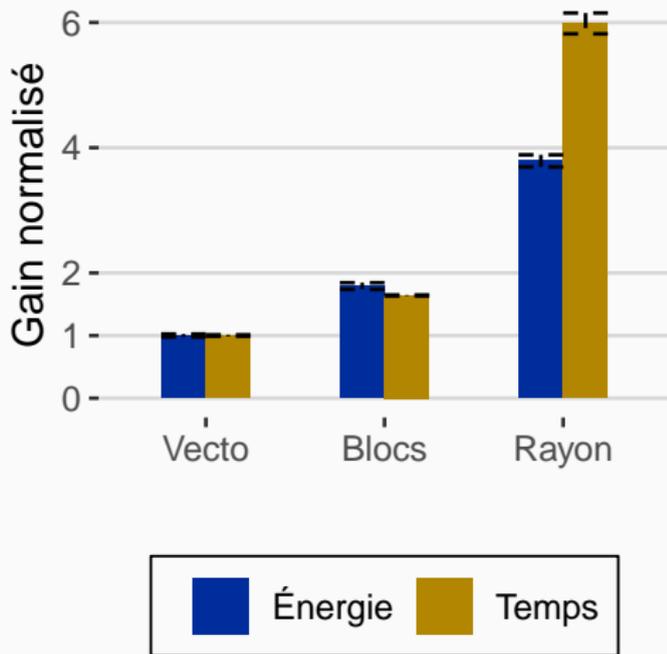
```
let cc = c.values.par_chunks_mut(n*Matrix::BLOCK);
```

```
let ca = a.values.par_chunks(n*Matrix::BLOCK);
```

```
cc.zip(ca).for_each(|(cb, ab)|  
    multiply_block(cb, ab, &b.values, ...))
```

Différentes implémentations

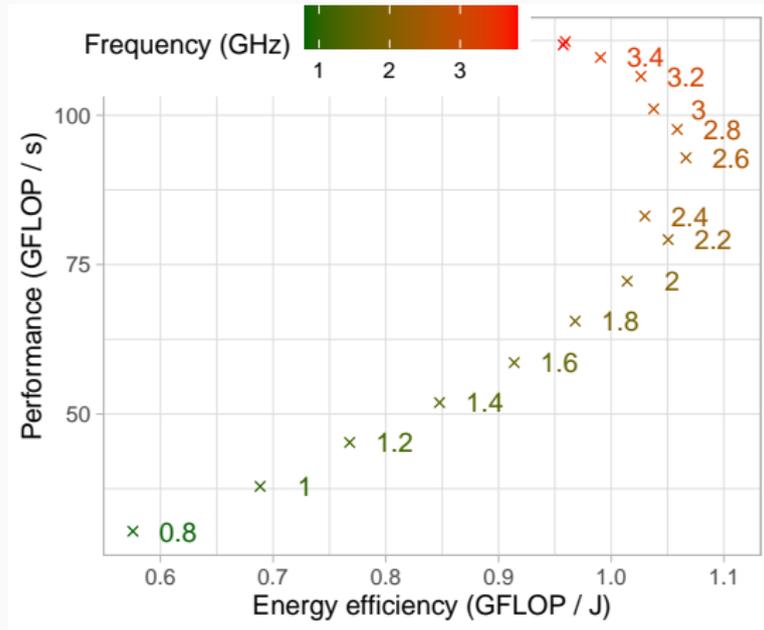
- Blocs : optimisation de la localité
- Rayon + Blocs : Parallélisation sur 6 cœurs
 - Découplage des gains d'énergie et de temps
 - Plus de cœurs allumés (+ rapide / + énergivore)
 - Ici, un parallélisme de 6 permet le meilleur gain d'énergie.



Taille : 4608 x 4608, 21M éléments

i7-9850H @ 2.60GHz 6 cœurs, 12 Mo cache L3

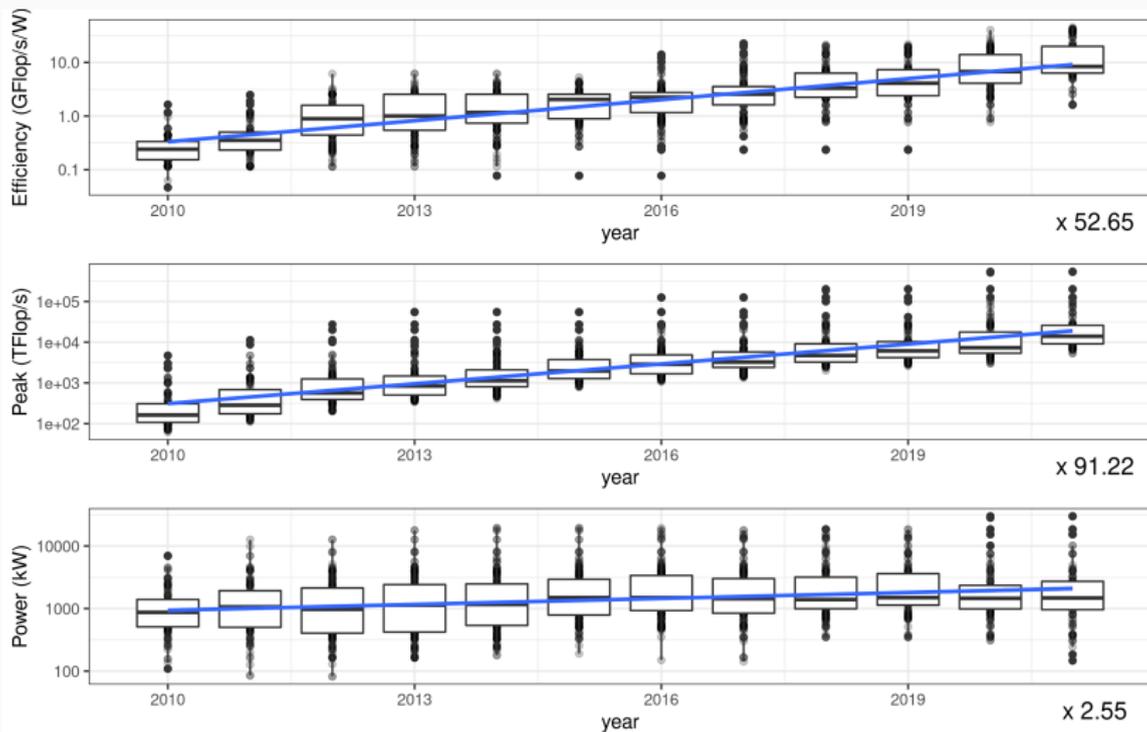
Compromis énergie / temps à trouver



- Modèle: RAPL + 40W
- Factorisation LU (Intel MKL)

- « Race to idle » n'est pas toujours optimal
- Baisser à 2.6 GHz permet d'économiser de l'énergie

Analyse du TOP-100 des systèmes HPC



Augmentation de l'efficacité mais aussi de la consommation totale

Effets rebond

- En 1865, Jevons montre que les améliorations des machines à vapeur se traduisent par une augmentation de la consommation de charbon.
- Dans le domaine du calcul intensif, les gains d'efficacité contribuent à la demande croissante en calcul.
 - augmentation nette de la consommation totale d'énergie.
- Effets de rebond pour les centres de données [Masanet, 2020]
 - Augmentation de 6% de la consommation d'énergie entre 2010 et 2018
(augmentation de 255% du nombre de nœuds).
- **Effets de rebond indirects** : les avancées en calcul peuvent contribuer à l'accélération d'autres domaines.

Besoin d'une discussion interdisciplinaire

- L'IA / HPC peut contribuer au développement durable
... **mais son coût énergétique doit être réduit**
- **Efficacité :**
 - Améliorer le matériel et le **logiciel**
 - Utiliser des modèles plus petits / une précision plus faible... soumis à des effets de rebond
- **Frugalité en informatique**
 - Équilibrer le coût de calcul par rapport aux résultats de chaque tâche
 - Choisir le modèle de taille appropriée
 - Évaluer l'impact environnemental

Questions interdisciplinaires et éthiques ?

Les deux séquences ouvrent des questions qui **dépassent l'informatique**:

- L'urgence climatique → questions existentielles, sociales, économiques, politiques, etc.
- Les limites des solutions techniques face à l'effet rebond

Pistes pour aborder ces questions

- **Co-construction** : Se placer en non-spécialiste et explorer ces controverses avec les étudiants ?
- **Interdisciplinarité** : Faire intervenir des collègues philosophes, sociologues, historiens, etc. ?