7th March 2023

Approximate Computing (AxC) for Green Software

J.Sallou@tudelft.nl

ORGANISATION of the lecture

OPEN CONVERSATION

I prefer the conversational mode, as I find it more lively, engaging, and interesting for everyone.

So, if you have a question regarding the content I am talking about, or feel that you need more clarification, please interrupt me.

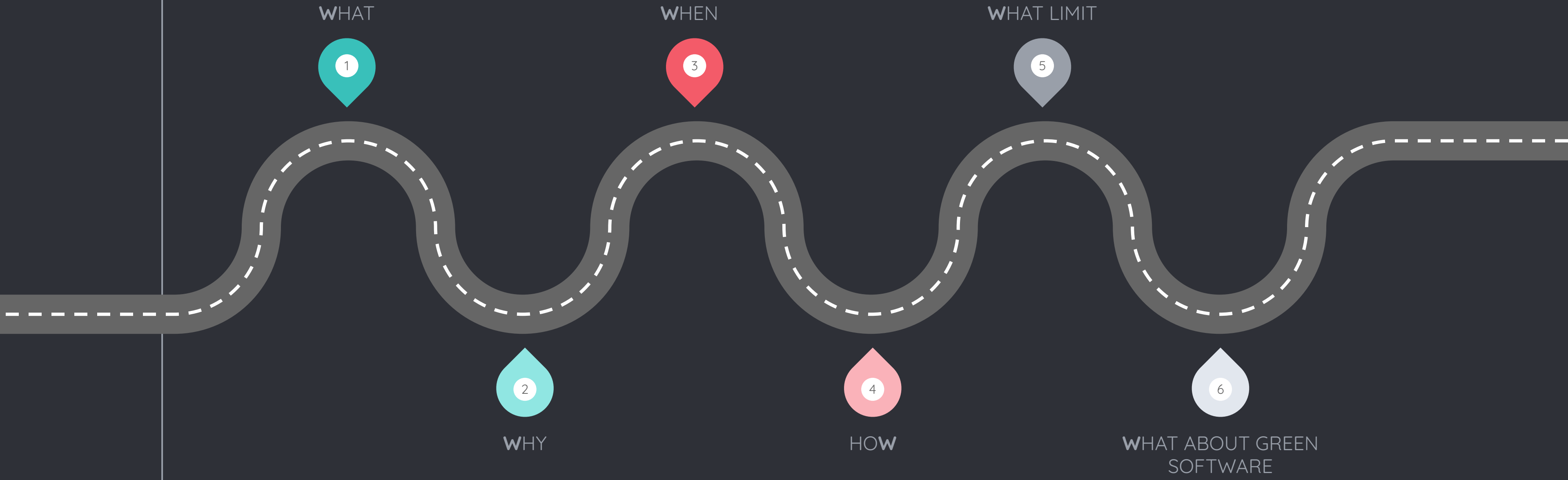
BREAK MIDWAY

Hello!

I AM JUNE

I studied Agronomy, Bioinformatics and Software Engineering. This led me to bring together environmental sciences and software engineering in my research projects. I now aim to make (AI-based) software more (environmentally) sustainable.

Roadmap: The W-journey



1

What is Approximate Computing (AxC)?

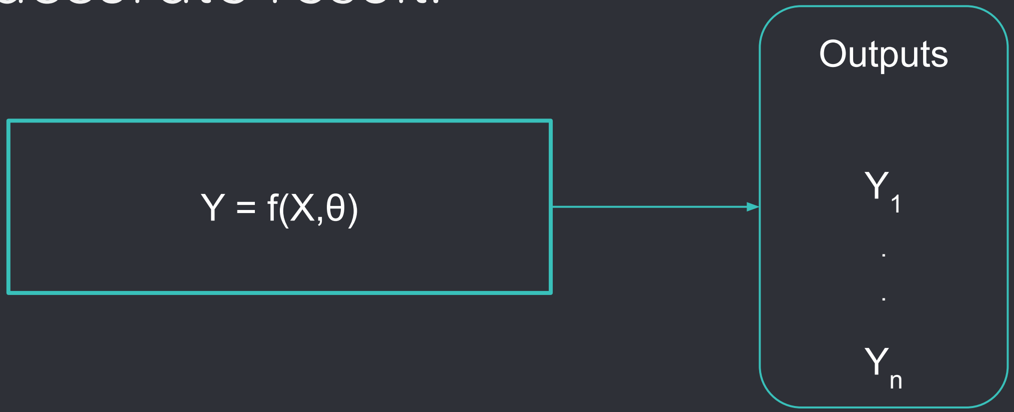
WHAT is AxC?

Producing a less accurate result rather than the reference accurate result.

WHAT is AxC?

Producing a less accurate result rather than the reference accurate result.

Reference



WHAT is AxC?

Producing a less accurate result rather than the reference accurate result.

Reference

$$Y = f(X,\theta)$$

Approximate

$$\hat{Y} = \hat{f}(\hat{X},\theta)$$

Outputs

Y_1

.

.

Y_n

WHAT is AxC?

Producing a less accurate result rather than the reference accurate result.

Reference

$$Y = f(X, \theta)$$

Approximate

$$\hat{Y} = \hat{f}(\hat{X}, \theta)$$

Outputs

$\hat{Y}_1$

.

.

$\hat{Y}_n$

2

What for?

Why doing AxC? What is the goal?

WHAT for?

Approximate Computing is based on the principle of a **trade-off** between accuracy and performance (e.g., execution time, memory, or energy consumption).

3

Why is it relevant?

Why is it relevant?

Discrete World 📈📊

Complexity 🤯😵

The Race for ~~More~~ 🏃

4

When to apply AxC?

Whenever the accuracy is not the priority. AxC is only relevant when considering the purpose of the execution.

WHEN?

- Image, video, signal processing
- Artificial Intelligence
- Telecommunications
- Mobile computing
- Etc

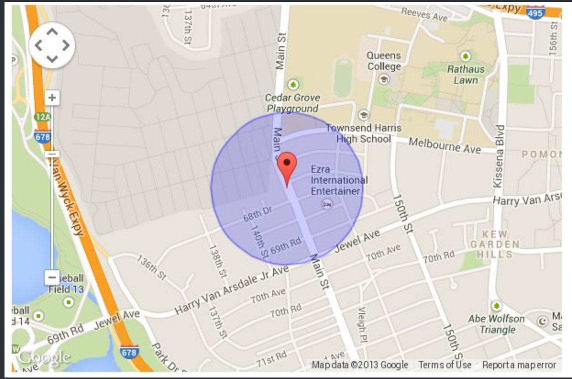
WHEN?

Netflix/Youtube/Twitch

- According to bandwidth of the network connexion, the fps (frame per second) rate is adapted.

WHEN?

Google Maps



- Approximation in localizing feature.

WHEN?

- What about health applications?
- Self-driving cars?

5

How to apply AxC?

How?

Data

Circuits

Voltage

Loop perforation

Algorithm

memoization

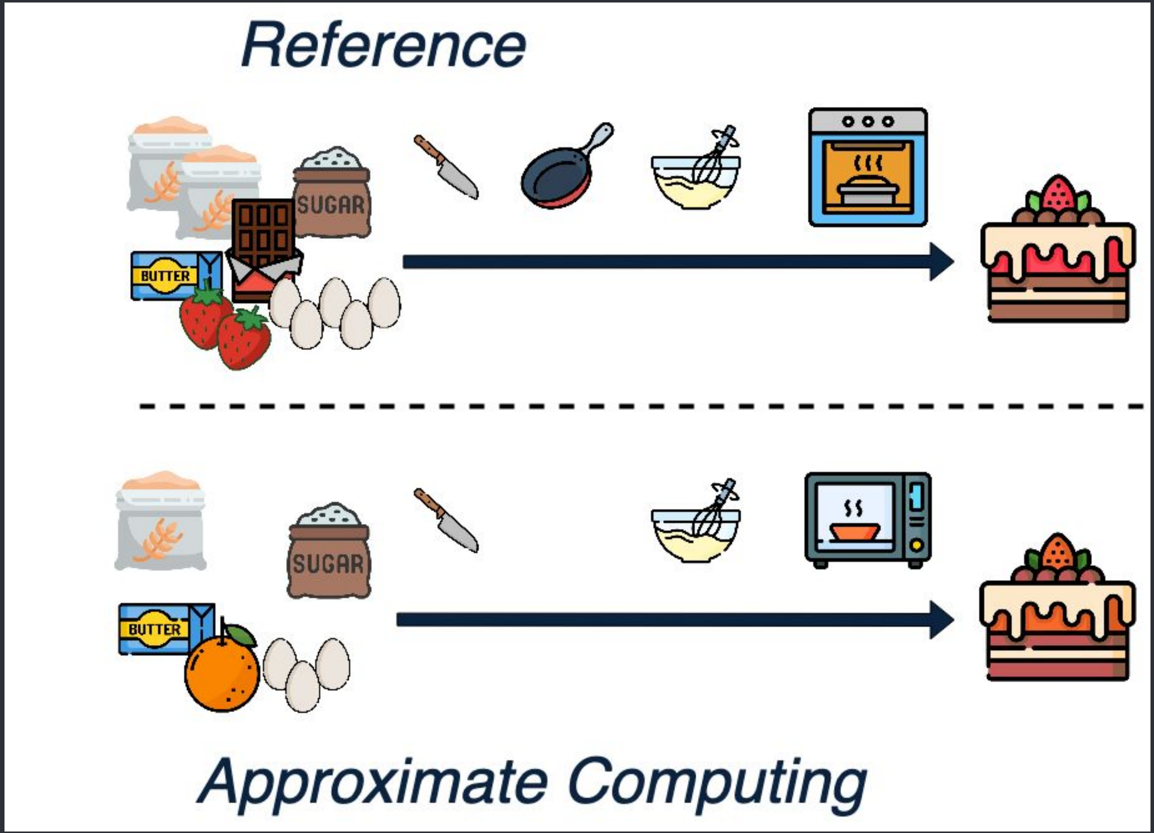
AxC Techniques

Hardware Level

Software Level

AxC for Software

AxC for Software



Algorithm-level AxC

Computation Skipping
- Fine-Grained
- Coarse-Grained

Computation Replacement

Computation Skipping

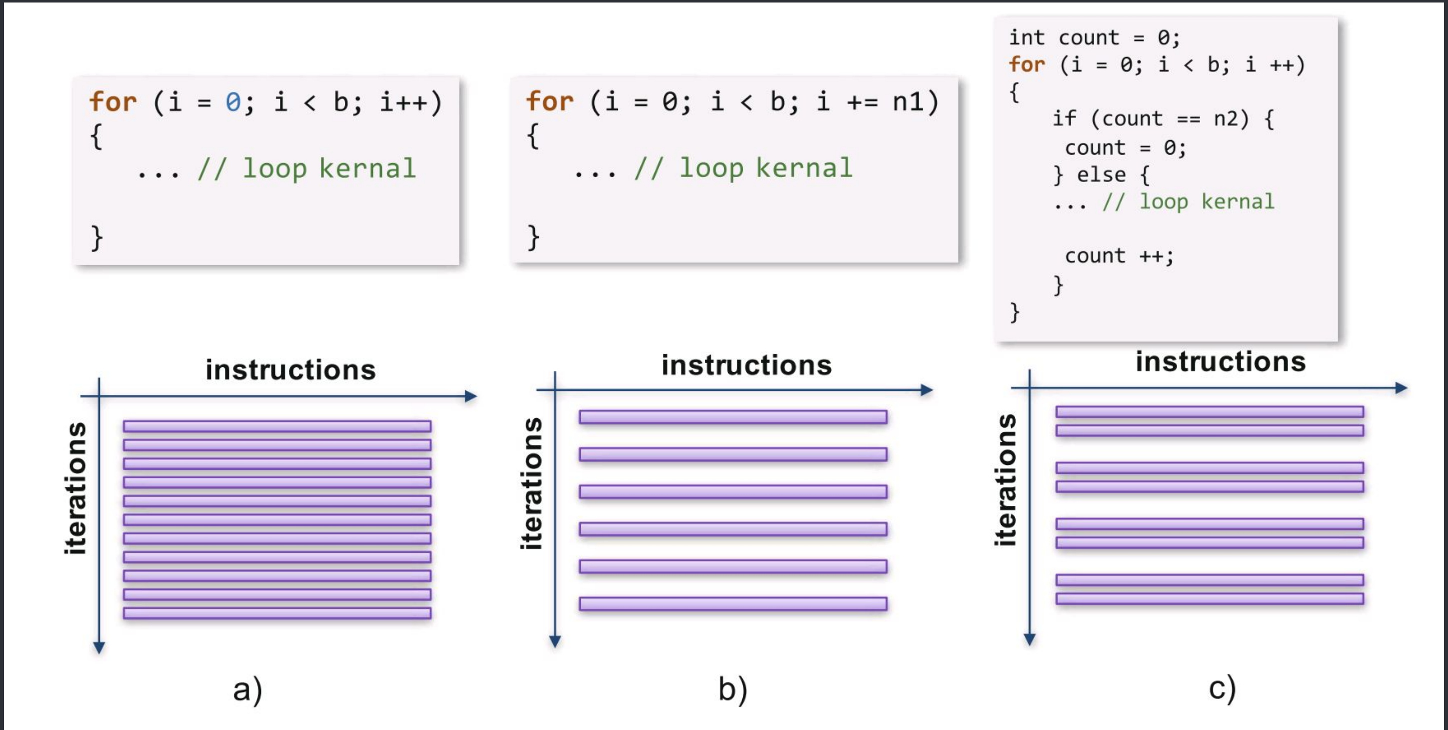
Applications:

- Search Space Enumeration
- Monte Carlo Simulation
- Iterative Refinement
- Data Structure Update

Loop perforation

- Data-oriented applications
- Reduce loop complexity

Loop perforation



rate= 1

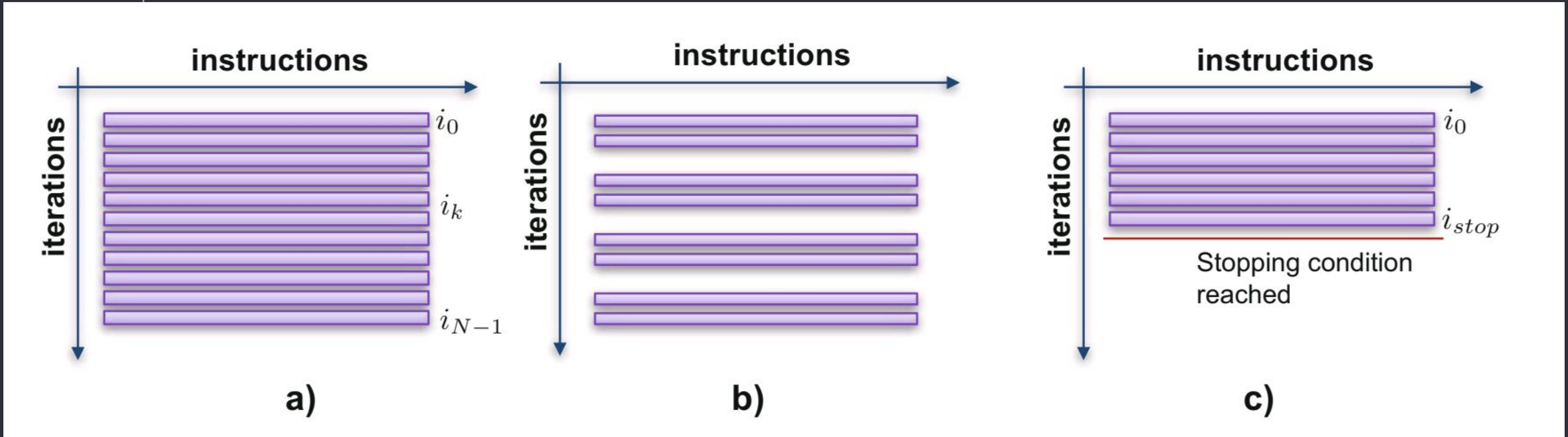
rate= 1 - $1/n_1$

rate= $1/n_2$

Early termination / Loop truncation

- Ending of computation before its end
- Convergent computation

Early termination / Loop truncation



$Y_0 = C$

For $i=1$, $i<\mathbf{i_{stop}}$, $i+=1$;

$\quad Y_i = f(X_i, Y_{i-1})$

Others

- Loop unrolling (+ interpolation)
- Loop tiling (interpolation = copy nearby output)
- …

Task Skipping

- Skip an entire task (block of code)
- E.g., Adaptation of processing load of hardware capabilities (best effort computing)

Computation Replacement

- Algorithm Selection
- Parameter Adjustment
- Memoization
- Neural Network Approximation
- Mathematical Function Approximation

Algorithm Selection

There are several versions of the same processing task / code block, with different costs / accuracy.

Adaptation according to context, runtime environment, etc.

```
def compute_2_plus_2():
        Return 2+(1-1)+12-10

def compute_2_plus_2_simplified():
        Return 2+2
```

Parameter Adjustment

-> Hyperparameter tuning.
During the design of software, the application parameters are optimised. For each parameter, the best value for accuracy is selected.

Tuning those parameters to best match the performance goal.

Memoization

Saving the results in a Look-Up Table.

The computation will not have to be performed again.
Just checking the result in the table.

Inputs	Outputs
X_1	Y_1
..	..
X_n	Y_n

Neural Network Approximation

- Data-driven approach
- Replace a complex processing by a neural network
- Neural network have been studied a lot. Specific hardware to run them exist. Parallelism is possible.

- Requires a lot of data
- Resource-demanding

Training Dataset	
Inputs	Outputs
X_1	Y_1
..	..
X_n	Y_n

Mathematical Function Approximation

A.k.a ‘Model Order Reduction’

- Needs a great deal of (numerical) expertise
- Time-consuming

$$\boxed{Y = f(X,\theta)}$$

$$\downarrow$$

$$\boxed{\hat{Y} = \hat{f}(X,\theta)}$$

6

What is the limit?

(How to define the sky?)

WHAT is the limit?

WHEN TO STOP?

- Validation Metric
- Validation Criterion
- Execution Purpose
- Domain experts

$\hat{E}_i = \hat{Y}_i - Y_i$

$\hat{E}_i < \text{threshold}$

WHAT is the limit?

Data processing domain	Quality evaluation function
Digital signal processing	Signal to noise ratio
	Mean squared error
	Relative difference
Image processing	Peak signal to noise ratio
	SSIM
	Mean squared error
	Pixel difference
Image segmentation & recognition	Ratio of misclustered points
	Mean centroid distance
	Top-1–top-5 classification
Video coding	Bjøntegaard delta peak signal to noise ratio
	Bjøntegaard delta bit rate
Digital communications	Bit error rate
Web search	Number of correct results in top 25 results

WHAT is the limit?

Prediction of AI models :

- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1 score = 2 * (Precision * Recall) / (Precision + Recall)
- Mean Squared Error
- ...

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2.$$

Case Study

Scientific Computing

Case Study: Scientific Computing

Scientific simulation software is elaborated by scientists to understand real-world phenomena. Such software is **complex** and **long to execute**. Hence, software is not interactive, and not usable to support the decision making of stakeholders impacting activities involved in climate change.

Loop Aggregation for Approximate Scientific Computing

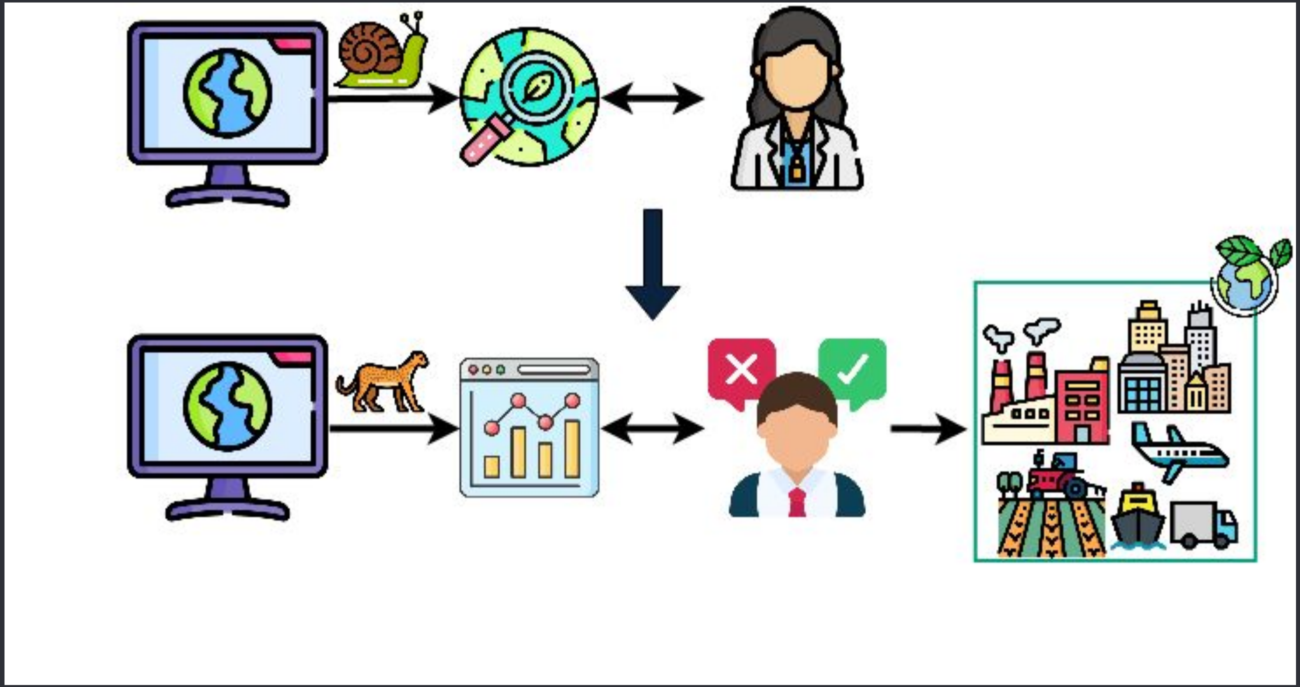
June Sallou[1,2(✉)], Alexandre Gauvain[2], Johann Bourcier[1(✉)], Benoit Combemale[3(✉)], and Jean-Raynald de Dreuzy[2]

[1] Univ Rennes, Inria, CNRS, IRISA, Rennes, France
{june.benvegnu-sallou,johann.bourcier}@irisa.fr
[2] Geosciences Rennes, OSUR, Rennes, France
{alexandre.gauvain,jean-raynald.de-dreuzy}@univ-rennes1.fr
[3] Inria and University of Toulouse Jean Jaurès, Toulouse, France
benoit.combemale@inria.fr

Abstract. Trading off some accuracy for better performances in scientific computing is an appealing approach to ease the exploration of various alternatives on complex simulation models. Existing approaches involve the application of either time-consuming model reduction techniques or resource-demanding statistical approaches. Such requirements prevent any opportunistic model exploration, e.g., exploring various scenarios on environmental models. This limits the ability to analyse new models for scientists, to support trade-off analysis for decision-makers and to empower the general public towards informed environmental intelligence. In this paper, we present a new approximate computing technique, aka. `loop aggregation`, which consists in automatically reducing the main loop of a simulation model by aggregating the corresponding spatial or temporal data. We apply this approximate scientific computing approach on a geophysical model of a hydraulic simulation with various input data. The experimentation demonstrates the ability to drastically decrease the simulation time while preserving acceptable results with a minimal set-up. We obtain a median speed-up of 95.13% and up to 99.78% across all the 23 case studies.

Keywords: Approximate computing · Trade-off · Computational science

Case Study: Context



Case Study: AxC

- (Hydrogeological) simulation model = iterative computation over n days
- Inputs are recharge rates per day (i.e., quantities of water to enter the soil). => X
- Outputs are the level of the underground water. => Y

$Y_0 = C$
For $i=1, i<N+1, i+=1;$
$\quad Y_i = f(X_i, Y_{i-1})$

?

Case Study: AxC

- **Perforation of the processing**

$Y_0 = C$

For i=1, i<N+1, i+=1;

$Y_i = f(X_i, Y_{i-1})$

$Y_0 = C$

For i=1, i<N+1, i+=**p**;

$Y_i = f(X_i, Y_{i-1})$

Case Study: AxC

- Perforation of the processing
- **Interpolation of results for the skipped iterations**

$Y_0 = C$

For $i=1$, $i<N+1$, $i+=1$;

$Y_i = f(X_i, Y_{i-1})$

$Y_0 = C$

For $i=1$, $i<N+1$, $i+=p$;

$Y_i = f(X_i, Y_{i-1})$

$\mathbf{Y_{i-p+1}, .., Y_{i-1} = Interpolate(Y_{i-p}, Y_i)}$

Case Study: AxC

- Perforation of the processing
- Interpolation of results for the skipped iterations
- **Aggregation of inputs**

$Y_0 = C$
For i=1, i<N+1, i+=1;
  $Y_i = f(X_i, Y_{i-1})$

$Y_0 = C$
For i=1, i<N+1, i+=p;
  $\mathbf{Y_i = Aggregate(Y_{i-p+1}, Y_i)}$
  $Y_i = f(X_i, Y_{i-1})$
  $Y_{i-p+1}, \ldots, Y_{i-1} = Interpolate(Y_{i-p}, Y_i)$

Case Study: AxC => Loop Aggregation

1. Aggregation of inputs
2. Perforation of the processing
3. Interpolation of results for the skipped iterations

$Y_0 = C$
For i=1, i<N+1, i+=1;
$\quad Y_i = f(X_i, Y_{i-1})$

$Y_0 = C$
For i=1, i<N+1, i+=**p**;
$\quad \mathbf{Y_i = Aggregate(Y_{i-p+1}, Y_i)}$
$\quad Y_i = f(X_i, Y_{i-1})$
$\quad \mathbf{Y_{i-p+1},.., Y_{i-1} = Interpolate(Y_{i-p}, Y_i)}$

Case Study: Aggregation

Stress Period Number	SP Duration (days)	Recharge (m/day)
0	1	0.00000
1	1	0.00001
2	1	**0.00000**
3	1	0.00012
4	1	**0.00159**
...	...	...

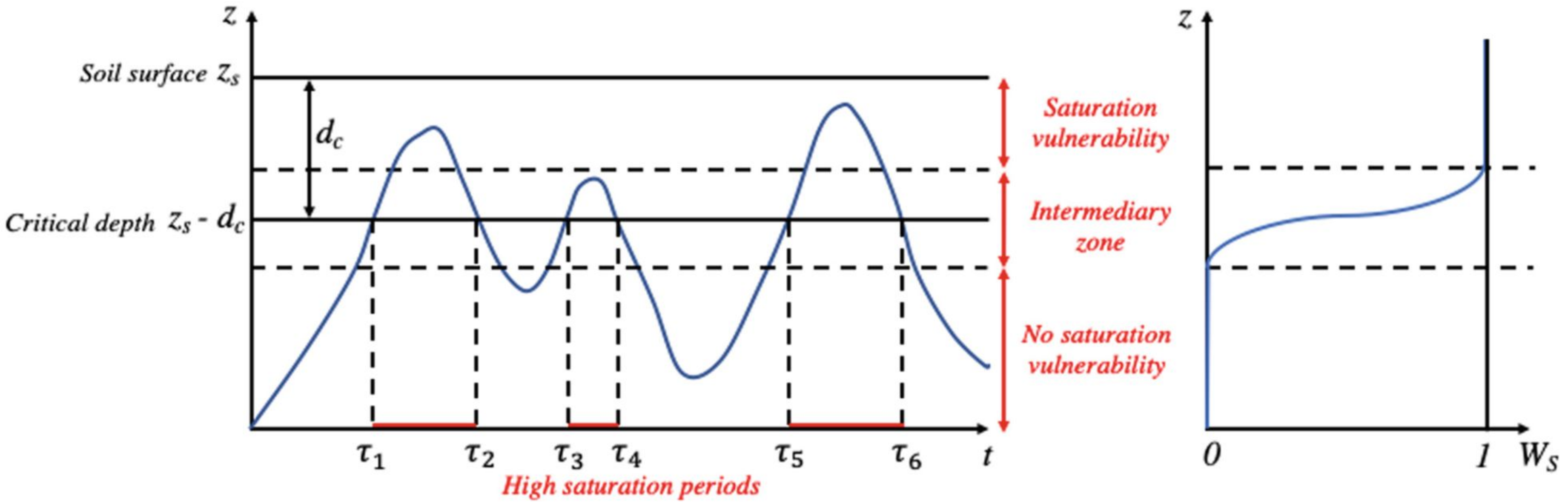
p = 2

0.00000

0.00159

Stress Period Number	SP Duration (days)	Recharge (m/day)
0	1	0.00000
2	**2**	**0.00000**
4	**2**	**0.000159**
...	...	...

Case Study: Validation Metric

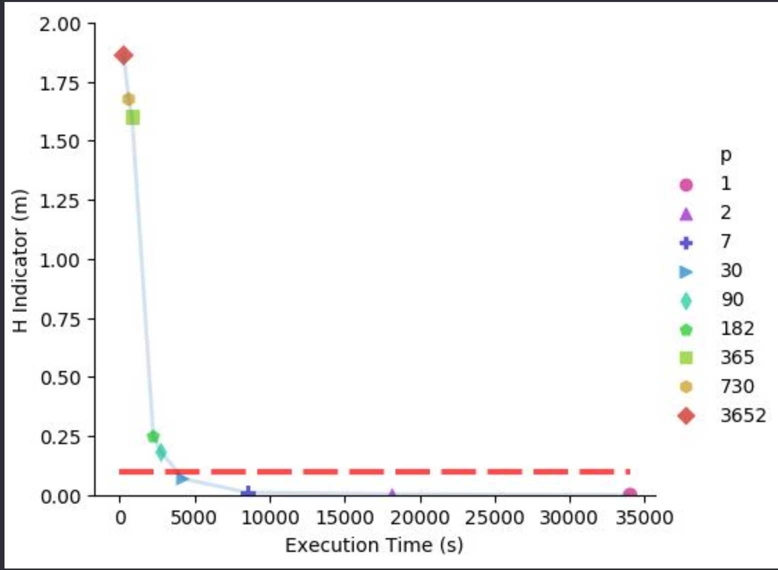


Case Study: Validation Metric (H<0.1m)

$$W_s(h) = \begin{cases} 0 & \text{if } h < z_s - \left(d_c + \frac{\Delta d_c}{2}\right) \\ sin\left(\frac{\pi}{2}\frac{h-\left(z_s-\left(d_c+\frac{\Delta d_c}{2}\right)\right)}{\Delta d_c}\right) & \text{if } z_s - \left(d_c - \frac{\Delta d_c}{2}\right) \leq h \leq z_s - \left(d_c + \frac{\Delta d_c}{2}\right) \\ 1 & \text{if } h > z_s - \left(d_c - \frac{\Delta d_c}{2}\right) \end{cases} \tag{1}$$

$$\|\Delta h\|_2 = \sqrt{\frac{\sum_t \sum_x max\left[W_s\left(h_R\left(x,t\right)\right), W_s\left(h_A\left(x,t\right)\right)\right] * \left(h_R\left(x,t\right) - h_A\left(x,t\right)\right)^2}{\sum_t \sum_x max\left[W_s\left(h_R\left(x,t\right)\right), W_s\left(h_A\left(x,t\right)\right)\right]}} \tag{2}$$

Case Study: Results



p	Time (s)	Speed-up (%)	H Ind. (m)
1	34032	0	0
2	18173	46.6	3.62E-03
7	8519	74.97	9.99E-03
30	4256	87.49	6.81E-02
90	2761	91.89	1.80E-01
182	2238	93.42	2.47E-01
365	820	97.59	1.6
730	597	98.25	1.68
3652	255	99.25	1.86

We apply AxC to speed up the simulation execution. We adapt the software (data & algorithm) by reducing the number of iterations of the simulation. The experimentation shows a median speed-up of **95.13%**.

Case Study: Comments? Critical thinking?

Case Study: Comments? Critical thinking?

Replication of time measurements

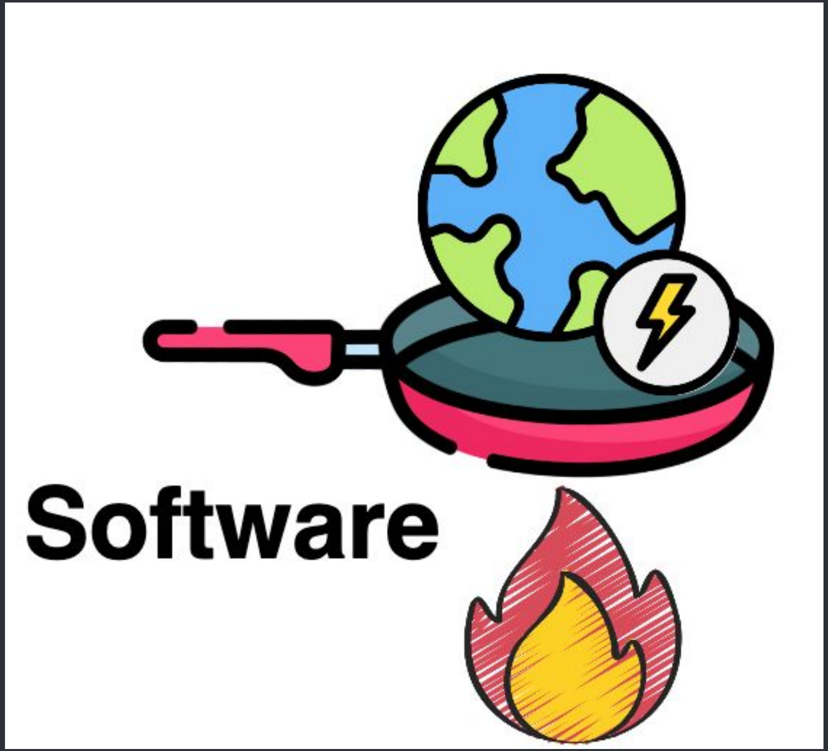
p	Number of replicates	Mean (s)	Median (s)	Standard Deviation (s)	RSE (%)
1	30	3.57E+04	3.66E+04	3.01E+03	8.42
365	30	1.02E+03	9.68E+02	1.71E+02	16.77
3652	30	2.07E+02	2.00E+02	2.76E+01	13.33

7

What about Green Software?

AxC for Green Software

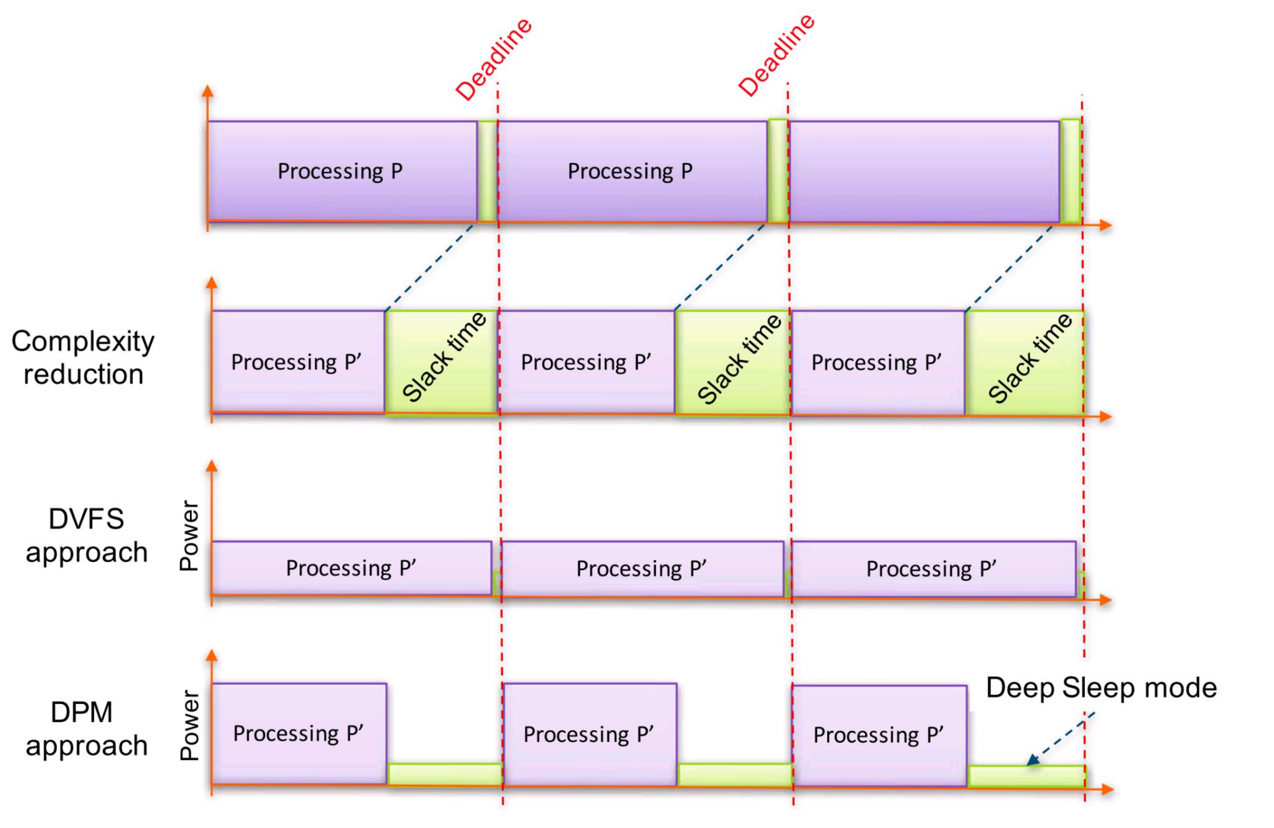
SOFTWARE is (H)EATING the world



PERFORMANCE = ENERGY EFFICIENCY

The goal is to make software greener by trading off accuracy for better energy efficiency.

AxC for Green Software



Dynamic Voltage and Frequency Scaling

Dynamic Power Management

Case Study

Green AI

Case Study: Green AI

With the growing availability of large-scale datasets, and the popularization of affordable storage and computational capabilities, the energy consumed by AI is becoming a growing concern.

Data-Centric Green AI
An Exploratory Empirical Study

Roberto Verdecchia*, Luís Cruz†, June Sallou‡, Michelle Lin§, James Wickenden¶, Estelle Hotellier‖
*Vrije Universiteit Amsterdam, The Netherlands - r.verdecchia@vu.nl
†Delft University of Technology, The Netherlands - l.cruz@tudelft.nl
‡Univ Rennes, France - june.benvegnu-sallou@irisa.fr
§McGill University, Canada - michelle.lin2@mail.mcgill.ca
¶University of Bristol, United Kingdom - jw17943@bristol.ac.uk
‖Inria, France - estelle.hotellier@inria.fr

***Abstract*—With the growing availability of large-scale datasets, and the popularization of affordable storage and computational capabilities, the energy consumed by AI is becoming a growing concern. To address this issue, in recent years, studies have focused on demonstrating how AI energy efficiency can be improved by tuning the model training strategy. Nevertheless, how modifications applied to datasets can impact the energy consumption of AI is still an open question.**

To fill this gap, in this exploratory study, we evaluate if data-centric approaches can be utilized to improve AI energy efficiency. To achieve our goal, we conduct an empirical experiment, executed by considering 6 different AI algorithms, a dataset comprising 5,574 data points, and two dataset modifications (number of data points and number of features).

Our results show evidence that, by exclusively conducting modifications on datasets, energy consumption can be drastically reduced (up to 92.16%), often at the cost of a negligible or even absent accuracy decline. As additional introductory results, we demonstrate how, by exclusively changing the algorithm used, energy savings up to two orders of magnitude can be achieved.

In conclusion, this exploratory investigation empirically demonstrates the importance of applying data-centric techniques to improve AI energy efficiency. Our results call for a research agenda that focuses on data-centric techniques, to further enable and democratize Green AI.

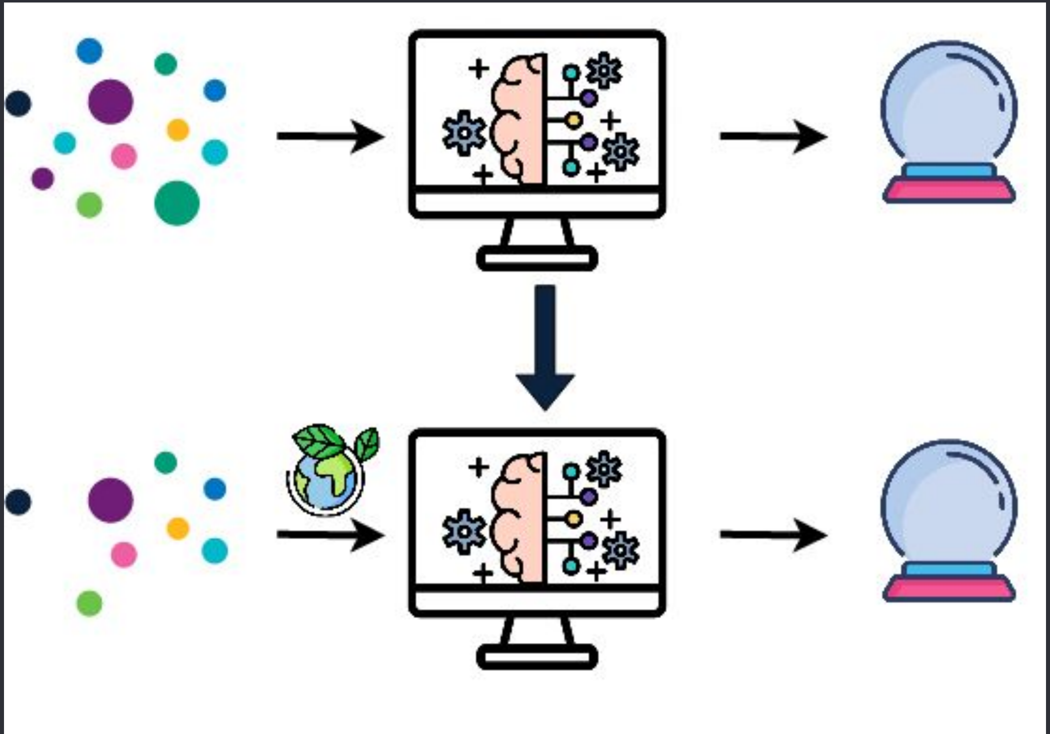
***Index Terms*—Energy Efficiency, Artificial Intelligence, Green AI, Data-centric, Empirical Experiment**

been estimated to consume the energy equivalent of a trans-American flight [5]. Hence, a new subfield is emerging to make the development and application of AI technologies environmentally sustainable: *Green AI* [6].

On a related note, the current research practice of collecting massive amounts of data is not necessarily yielding better results. Being able to collect high-quality data is more important than collecting big data – a trend coined as *Data-centric AI*[1]. Instead of creating learning techniques that squeeze every bit of performance, data-centric AI focuses on leveraging systematic, reliable, and efficient practices to collect high-quality data.

Therefore, in this study, we conduct an exploratory empirical study on the intersection of Green AI and Data-centric AI. We investigate the potential impact of modifying datasets to improve the energy consumption of training AI models. In particular, we focus on machine learning, the branch of AI that deals with the automatic generation of models based on sample data – machine learning and AI are used interchangeably throughout this paper. In addition to investigating the energy impact of dataset modifications, we also analyze the inherent trade-offs between energy consumption and performance when reducing the size of the dataset – either in the number of

Case Study: Green AI



Case Study: Green AI

We apply AC to investigate the potential impact of modifying datasets to improve the energy consumption of training AI models. Our results show evidence that energy consumption can be drastically reduced (up to **92.16%**).

8 Take-Away Messages

AxC in Brief

Trade-Off

Accuracy vs Performance

Context

Execution purpose

Diversity of applications

Different levels, various domains of applications.

Validation Metric

Involvement of domain experts. With respect to execution purpose.

Good-Enough is great for sustainability!

References

- Bosio, Alberto, Daniel Ménard, and Olivier Sentieys, eds. *Approximate Computing Techniques: From Component-to Application-Level*. Springer Nature, 2022.
- Mittal, Sparsh. "A survey of techniques for approximate computing." ACM Computing Surveys (CSUR) 48.4 (2016): 1-33.
- Xu, Qiang, Todd Mytkowicz, and Nam Sung Kim. "Approximate computing: A survey." IEEE Design & Test 33.1 (2015): 8-22.
- June Sallou. On reliability and flexibility of scientific software in environmental science : towards a systematic approach to support decision-making. Software Engineering [cs.SE]. Université Rennes 1, 2022. English. NNT : 2022REN1S020 . tel-03854849
- Sallou, June, et al. "Loop aggregation for approximate scientific computing." Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part II 20. Springer International Publishing, 2020.
- R. Verdecchia, L. Cruz, J. Sallou, M. Lin, J. Wickenden and E. Hotellier, "Data-Centric Green AI An Exploratory Empirical Study," 2022 International Conference on ICT for Sustainability (ICT4S), Plovdiv, Bulgaria, 2022, pp. 35-45, doi: 10.1109/ICT4S55073.2022.00015.

Thanks! Please, give me your feedback.

Well Done! You've reached the end of the lecture. :)

You can contact me at

J.Sallou@tudelft.nl