Loop Aggregation for Approximate Scientific Computing

June Sallou, Alexandre Gauvain, Johann Bourcier, Benoit Combemale, Jean-Raynald de Dreuzy

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, France
{june.benvegnu-sallou, johann.bourcier}@irisa.fr
[2] Geosciences Rennes, OSUR, France
{jean-raynald.de-dreuzy,alexandre.gauvain}@univ-rennes1.fr
[3] Inria & University of Toulouse Jean Jaurès, 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.

1 Introduction

There is a long-standing history on numerical analysis to provide a better accuracy in scientific computing. Research activities in the past decades result in efficient solvers to accurately simulate ever more complex models. For example, complex climate change models can nowadays be simulated to elaborate global warming scenarios and their consequences. However, such simulations come with the price of large computing and memory resources. While extremely useful to elaborate accurate results on precise scenarios, the required resources limit the possible interactions with the models. This prevents any *interactive*, *live* and *customized* manipulation of such models, e.g., 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 context, trading some accuracy for better performance in scientific computing is an appealing approach to facilitate the exploration of various alternatives on complex simulation models. This has been initially explored through model reduction techniques [11,1], which require specific knowledge on the model and highly-qualified analysis. Alternatively, statistical approaches are explored to automatically infer a new (surrogate) model from a large and well-qualified set of input/output simulations [9]. Although these approaches are mostly automatic and potentially accurate and efficient, they usually require a large, possibly controlled, set-up before providing acceptable results from the inferred model.

Model reduction and statistical techniques are potentially very efficient, but all require an important initial set-up (either time-consuming or resource-demanding) that prevents any *opportunistic* model exploration, e.g., exploration of various scenarios on environmental models.

In this paper, we present a new approximate computing (AC) technique, called `loop aggregation`. According to the main variable of interest, we automatically reduce the main loop of the simulation model by aggregating to a given degree the corresponding spatial or temporal data. This aggregation can be either applied as a pre-processing of the input data or by model transformation. For example, in the case of an a posteriori study of the causes of soil drying-up with all the necessary data available, it is worthwhile to use the pre-processing implementation. Or, in the case of a crisis situation and the continuous monitoring of a sudden and dangerous flooding episode, the model transformation will be able to manage the available on-going data. We apply `loop aggregation` on a geophysical model of a hydraulic simulation with various input data from different sites and climate series. The geophysical model concerns the groundwater flow in coastal areas where sea level rise changes the distribution of saturation inland and potentially generates risks of floodings even when submersion is properly managed. Experiments are performed on 23 sites to investigate the potential reduction of computational time without significant modification of the assessment of groundwater-issued vulnerability. Results show a median speed-up of 95.13%, demonstrating the ability to drastically reduce the simulation time while preserving acceptable results with a minimal set-up. We discuss the capacities and limitations of the approach in the perspectives of further generalisation.

To follow up, Section 2 presents the simulation model of interest and motivates the needs for improved interactivity. After an introduction of the background related to AC in Section 3, Section 4 details the overall approach and the rationales for a specific AC technique in the context of scientific computing. Section 5 gives the details of experimenting the proposed approach on the simulation model of interest and Section 6 describes the evaluation conducted to validate the approach. Section 7 presents related works on statistical approaches, model reduction techniques and AC. Section 8 concludes the paper and gives the perspectives related to this work.

2 Motivating Example

Hydrologists are working to determine the impact of the sea level rise on coastal aquifers, on increased saturation levels and associated consequences on inland vulnerability. Between the current state of the aquifers and the predicted sea level rise and climate scenarios, hydrological models are expected to provide predictions.

Those models are based on the three-dimensional software Modflow [8] considered to be an international standard for simulating and predicting groundwater movements. It is based on the Darcy's law and conservation principles to represent the groundwater flows. Groundwater flows are essentially modelled by a diffusion equation with Dirichlet boundary conditions when the groundwater level reaches the surface. The resulting parabolic partial differential equation is discretised with a finite difference method and integrated with classical implicit temporal schemes [8]. The quantity of interest to assess the groundwater-issued vulnerability is derived from the depth to the groundwater level. When groundwater levels rise to some tens of centimeters to the surface, vulnerability becomes difficult to mitigate. Modflow requires both the geological and geographical settings of the studied site (inputs illustrated as *Geology* and *Land Use* in Fig. 1) and the meteorological forcing term (represented as the *Weather* input in Fig. 1) driving the infiltration and the recharge to the aquifer. This configuration does not change over the simulation period. The meteorological forcing term comes from climate scenarios available on the next century with the estimation of the different elements of the hydrological balance taken here as the input (recharge) to the aquifer. The groundwater flow model provides over the simulation period the location of the groundwater surface, more generally called water table.

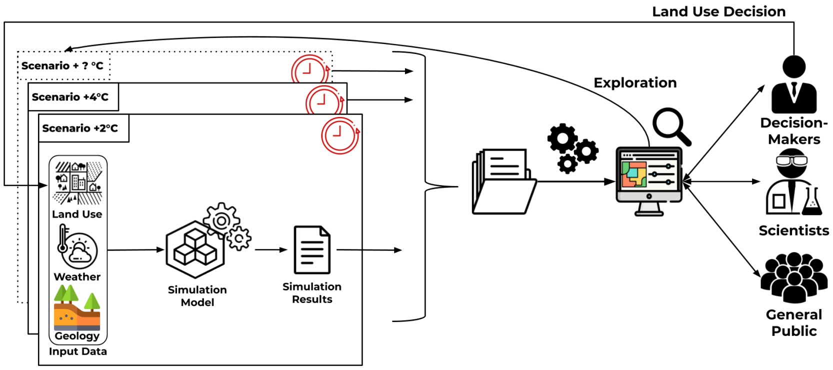


Figure 1: Exploration of several climate scenarios simulations for various stakeholders.

As shown in Fig. 1, although the model was created by and for hydrologists, the simulation results can also be of interest to other users. Indeed, with growing awareness of climate change, general public including decision-makers increasingly ask to investigate by themselves the effect of climate change on property and land use planning. Overall, people want to explore the different future climate scenarios, and associated simulations, in an interactive way to make informed decisions or to understand their impact.

Since Modflow is a complex model, its execution can last more than a day. This simulation time is multiplied by the number of scenarios to explore which prevent effective and interactive exploration of the predictions. Making the model run faster would enable such exploration, but the predictions obtained must remain scientifically acceptable to respect the main trends and avoid any significant bias.

Thus, there is a need for finding a trade-off between accuracy and performance. A solution is to simplify the model. However, hydrologists and/or decision-makers may not have the expertise to make this trade-off through model reduction. This raises several scientific questions : (i) Can we make Modflow run faster while maintaining acceptable predictions? (ii) Can we do so without any expertise in hydrology or model simplification and any time-consuming/resource-demanding set-up? (iii) More generally, how to achieve it for scientific simulation models? In this paper, we propose to investigate AC for scientific simulation models to provide relevant trade-offs between accuracy and performance.

3 Introducing acceptable approximation into models

The trade-off between accuracy and performance is a well-known concern in software engineering research. Approximate computing (AC) is one way to address this concern automatically from a computational point of view. It relies on the difference between the accuracy required by the developer or the user and the accuracy given by the execution of the software [7]. It introduces approximation into the program while producing acceptable outputs with respect to the purpose of the application thanks to an approximation strategy.

Loop perforation is an approximation strategy which assumes that iterations of loops in a program take time to be computed when not all of them would be necessary to achieve a result similar to that obtained from all of the iterations. Within the software code, the appropriate loops are modified so that only a subset of the iterations is realised. In practice, for a loop incremented by 1 at each iteration, applying the loop perforation technique would mean changing the increment from 1 to p so that only every p-th iteration is performed [12].

Thus, the number of iterations is reduced, fewer calculations are performed and a gain in performance is obtained. The choice of p is made according to the acceptability constraints made on the software outputs. Loop perforation is a technique that achieves the trade-off between accuracy and performance without knowledge of the application domain.

Another AC strategy, approximate loop unrolling [10], proposes, in addition to skipping iterations, to interpolate the results of non-computed iterations thanks to an interpolation function. It enables to better preserve accuracy.

Models, such as scientific simulation models whose consideration is producing an output within an acceptable precision range and are being by definition only an approximation of reality, can naturally benefit from a trade-off between accuracy and performance through approximate computing.

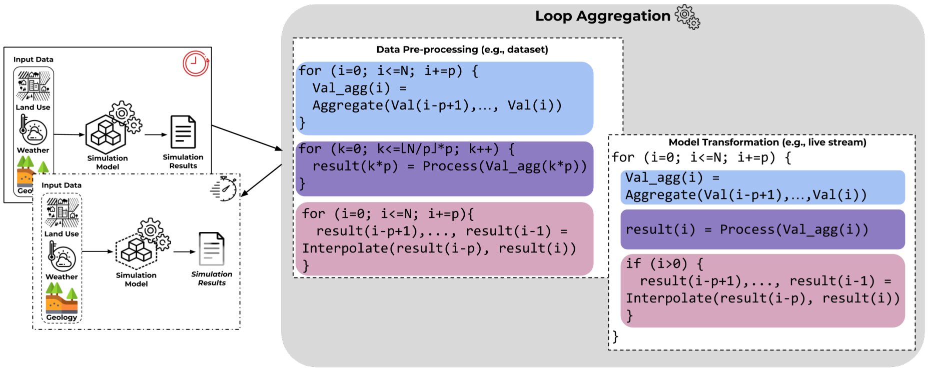
For scientific simulation models, the techniques of loop perforation or approximate unrolling cannot be straightforwardly applied. Removing iterations from a simulation leads to different case studies and non-comparable simulations. There is the exception of convergent simulations for which it is possible to safely remove iterations whose results do not provide more information. However without any information about the convergent criteria, it is not safe to assume that scientific simulations with different numbers of iteration can be comparable. In the case of our motivating example, Modflow, fully removing some iterations corresponding to time steps would alter the duration of the simulation period. Even more, meteorologic data are highly variable and cannot be easily inferred between time steps. The recharge rate (ie. quantity of water to enter the aquifer per time unit) varies on a daily basis. Removing iterations would change the climate scenario. Thus, the model of Modflow undergoing loop perforation or loop unrolling would not be comparable to the initial model.

4 Approach

To handle the issue of non-comparable simulations, we reduce the number of computations while ensuring comparable conditions (eg. same duration of the simulation period for Modflow). We introduce a new AC strategy adapted to scientific models, the `loop aggregation` technique (depicted in Fig. 2). This technique is similar to loop perforation and loop unrolling since it skips some loop iterations but adds specific stages to keep the results and simulation consistent with the baseline. It acts on the main loop of the simulator which is the loop iterating at the highest level on all the input data and enclosing all the processing of those data. The `loop aggregation` technique consists in three stages:

- *aggregation* (highlighted in blue): the input values of the main variable of interest are aggregated through an aggregation function. In the case of Modflow, the aggregate function consists in combining the recharge rate per day.
- *processing* (highlighted in violet): the operations within the loop are only performed on aggregated values.
- *interpolation* (highlighted in pink): the intermediate results are retrieved through an interpolation function.

As shown in Fig. 2, the simulation context guides the type of the `loop aggregation` implementation : data pre-processing or model transformation. Both implementations are equivalent as they reflect the `loop aggregation` approach. The difference is that the three stages are not carried out at the same moment. When all the input data are available before the model is run, a black-box implementation relies on the separation of the three stages (Data pre-processing). The aggregation stage acts as a pre-processing before the model execution, hence the name of the strategy. The values of the input data are aggregated according to an aggregation factor, p. The processing stage is carried out when the model runs with the model remaining as is. The interpolation stage is then performed as post-processing after the model execution. With a simulation context of dynamic data flows (ie. stream data), the model transformation strategy is used. It is then necessary to perform the three stages of approximation dynamically



```
for (i=0; i<=N; i+=p) {
  Val_agg(i) =
  Aggregate(Val(i-p+1),…, Val(i))
}
for (k=0; k<=⌊N/p⌋*p; k++) {
  result(k*p) = Process(Val_agg(k*p))
}
for (i=0; i<=N; i+=p){
  result(i-p+1),..., result(i-1) =
Interpolate(result(i-p), result(i))
}
```

```
for (i=0; i<=N; i+=p) {
  Val_agg(i) =
  Aggregate(Val(i-p+1),…,Val(i))
  result(i) = Process(Val_agg(i))
  if (i>0) {
    result(i-p+1),..., result(i-1) =
  Interpolate(result(i-p), result(i))
  }
}
```

Figure 2: The `Loop Aggregation` technique.

to take data into account when they are retrieved. This implies accessing the model source code and adopting a white-box approach. All the stages and the modifications are made within the main loop of the model and the value of the iteration step is replaced by p, the value of the aggregation factor.

In essence, the `loop aggregation` technique adds an aggregation stage to the approximation process described in approximate loop unrolling to enable AC with scientific models. It enables a black-box implementation with separate stages running at different times when all data are available or a white-box implementation with a model transformation when dealing with dynamic data. The number of computations is reduced by the use of the p-factor and the approximate simulation is still comparable to the reference one. In theory, the technique can be applied to all scientific models with a main loop iterating over temporal or spatial data. There is no need for specific knowledge about the application domain of the model except for information about the use of the model.

5 Experimenting Loop Aggregation on Modflow

5.1 The case study of Modflow

The case study of our motivating example is based on the prediction of groundwater movements in a watershed near Lestre in Normandy in France to assess the risk of increased saturation at this site. The prediction period for the simulation is 42 years and is represented by 15340 stress periods (ie. time steps) whose duration is set to correspond to a simulated day. The parameters of the model have been set by hydrologists. Executing Modflow with those inputs and non-aggregated data constitutes the reference simulation.

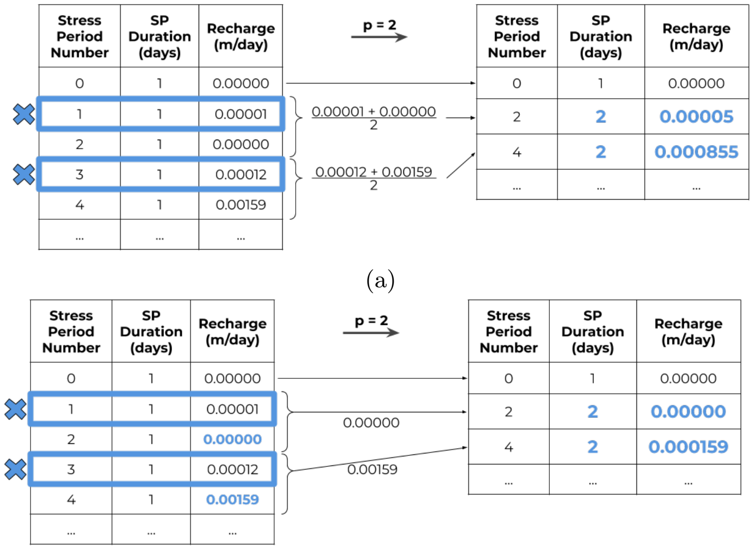
5.2 Approximating the model with the `Loop Aggregation` approach

We derive the approximate simulations using our `loop aggregation` technique with different aggregation p-factors to assess the variation of the simulation execution time.

In the case of Modflow, the main loop iterating over the stress periods, the computation reduction is done by removing some of them. The values of the recharge rate are aggregated as it is the variable of interest. As all the input data are available upstream, we perform our `loop aggregation` approach on those data following the Data Pre-processing implementation. We experiment two strategies for the aggregation stage (detailed hereafter) and linear regression for results interpolation.

Strategy with the mean as aggregation function of the recharge. The recharge data are aggregated for p being equal to 2, 7, 30, 90, 182, 365, 730 and 3652, corresponding to stress period durations of 2 days, 7 days, 1 month, 3 months, 6 months, 1 year, 2 years and 10 years. Those values of p were chosen to represent meaningful periods for hydraulic and meteorological events.

The mean is chosen as the aggregation function. The choice has been made according to the use of the recharge rate and according to the advice of experts to maintain the overall flux balance. To ensure comparable approximate simulations, the aggregation has also to impact the values of the stress period duration. Indeed, the simulation period must represent a span of 42 years. The stress period duration of the aggregated stress periods is thus changed into the value of p. For instance, with p=2, the inputs are modified as shown in Fig. 3a.



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
...	...	...

Stress Period Number	SP Duration (days)	Recharge (m/day)
0	1	0.00000
2	2	0.00005
4	2	0.000855
...	...	...

(a)

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
...	...	...

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

(b)

Figure 3: Aggregation strategies with p=2. a : mean as the aggregation function of the recharge. b : p-th value assigned as the aggregated value of recharge.

Strategy with assigning the p-th value as the aggregated value of recharge. The aggregation is again carried out for p being equal to 2, 7, 30, 90, 182, 365, 730 and 3652. The stress period duration is changed into the value of p and the recharge value of the p-th stress period is assigned as the mean recharge value for the corresponding aggregated stress periods. The aggregation is carried out as presented in Fig. 3b for p=2.

5.3 Conditions of the experimentation

Modflow is run as a fortran written executable with compiled code for computing the groundwater flows in the aquifer. A wrapping software layer written in Python by hydrologists is used to configure Modflow and format the simulation inputs. The version of Modflow used is MODFLOW-NWT-SWR1, the U.S. Geological Survey modular finite-difference groundwater-flow model with Newton formulation and with the version number 1.1.4 released on 04/01/2018 associated with the SWR1 which version number is 1.04.0 released on 09/15/2016.

The experimentation is done on a single node with a Intel(R) Xeon(R) CPU E5-2650 v4 processor with 2.20GHz. Each simulation is run on a single core with 2 threads and 8GB RAM. The model is embedded inside a Docker image deployed on a virtual machine for each simulation. The virtual machine is a Alpine Linux 3.4.3 amd64. Through the Docker image, the memory available for each simulation is limited to 2Gb. These measures are taken to limit variations in the experimentation environment.

6 Evaluation

In this section, we validate our ability to apply the `loop aggregation` on our motivating scenario presented in Section 2. The goal is to answer the following research questions :

RQ1 : Is `loop aggregation` able to perform substantial performance increase while maintaining meaningful results for experts ?

RQ2 : Is the `loop aggregation` technique able to produce relevant trade-offs for various input data such as climate scenarios and geographical sites?

6.1 Acceptation Criterion

Domain experts have established an approximation indicator called acceptance criterion, which represents a threshold under which the indicator value should remain for the approximated results to be considered acceptable.

Simulation approximations are defined on the quantities of interest of the models. Considering the issues of coastal saturation, the relevant quantities derive from the proximity of the aquifer to the surface. When the top of the aquifer (water table) approaches the soil surface at a distance smaller than d_c, water resources, soil humidity, flooding risks and other human activities are impacted. The characteristic distance d_c depends on the type of human activity (eg. agriculture or cites). It also depends on local choices of collectivities.

However, the transition is not sharp. Rather, it is a transition zone from not vulnerable conditions when the aquifer is deep enough to vulnerable close to the surface (Fig. 4). The width of the transition zone will be noted Δd_c and be taken as a linear function of d_c with $\mathit{\Delta} d_c(\boldsymbol{x})=\alpha d_c(\boldsymbol{x})$ where α is the proportionality factor and $\boldsymbol{x}$ is the position. With d_c typically of the order of 30 cm and α equal to 1/3, the $\mathit{\Delta} d_c$ the transition width is of the order of 10 cm.

Approximations on the quantities of interest will thus be weighted according to their proximity to the surface with the function $W_s(h)$, presented in Equation (1), where h is the piezometric level and z_s is the altitude of the soil surface.

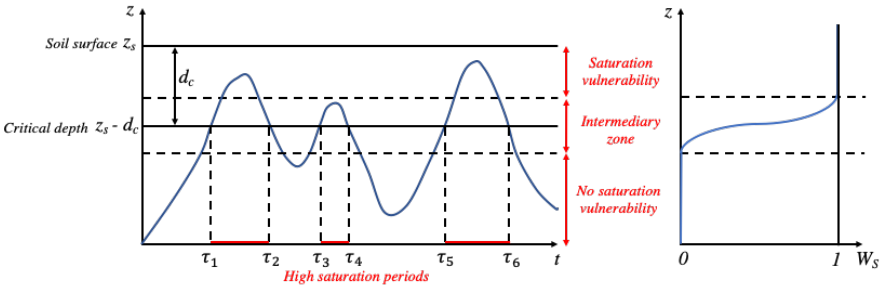


Figure 4: Vulnerability zone and the associated representation of Ws.

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

The H indicator $\|\Delta h\|_2$ on the saturation level is defined by Eq. 2. The threshold for the H indicator is set by hydrologists to 0.1 m, meaning that any approximation of the water table depth within a margin of 10 cm is acceptable. Variables issued by the reference and approximate simulations are indexed by the letters R and A respectively.

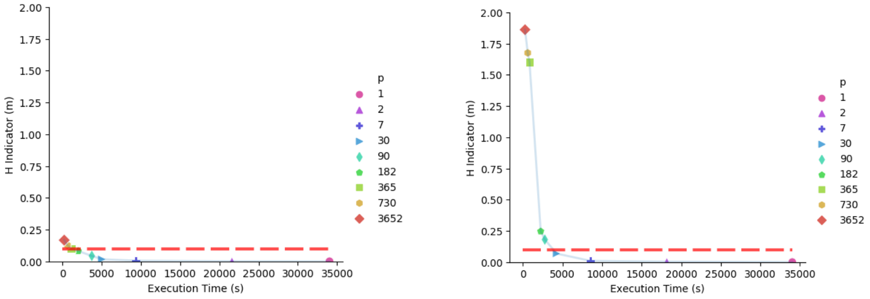
$$\|\Delta h\|_2=\sqrt{\frac{\sum_t\sum_x max[W_s(h_R(x,t)),W_s(h_A(x,t))]*(h_R(x,t)-h_A(x,t))^2}{\sum_t\sum_x max[W_s(h_R(x,t)),W_s(h_A(x,t))]}} \tag{2}$$

6.2 [RQ1] Performance increase with loop aggregation

To answer **RQ1**, we assess the performance increase and the acceptation criterion when applying `loop aggregation` on the Modflow model (Section 2). We use two aggregation strategies with the same inputs (site, ie. Lestre, and recharge series). The reference simulation is run in 34032 seconds, ie. 9 hours, 27 minutes and 12 seconds.

Experiments with the strategy of the mean as the aggregation function. We observe in Fig. 5a&c that the most approximated the simulation is, the fastest it is. It follows the rational idea that, for a dominantly linear model, the duration of the execution is directly linked to the number of iterations in the loop.With respect to the acceptance criterion, the approximated simulations performed in our experiment with a period of less than one year (p=365) are considered to produce acceptable outputs. Within these acceptable outputs, the shortest execution time (1149 seconds or 19 minutes and 9.0 seconds) is obtained with the simulation of one year stress periods. The execution time is reduced by more than 29 times, a speed-up of more than 96.6%.

Test with the strategy of the *p*-th value assigned to the aggregated value. `Loop aggregation` shows again a performance gain (Fig. 5b&d). We observe that the H indicator is higher for the same aggregation rate than for the previous strategy which is consistent with the fact that we introduce more approximation here (ie. the recharge values of the aggregated iterations are not taken into account). The best speed-up is 87.49% with p=30 (Fig. 5d).



(a) (b)

p	Time (s)	Speed-up (%)	H Ind. (m)
1	34032	0	0
2	21607	36.51	1.13E-03
7	9331	72.58	5.51E-03
30	5058	85.14	1.65E-02
90	3759	88.95	4.34E-02
182	2026	94.05	8.16E-02
365	1149	96.62	9.97E-02
730	647	98.1	1.27E-01
3652	241	99.29	1.67E-01

(c)

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

(d)

Figure 5: Evolution of H indicator and speed-up according to p for the mean aggregation function strategy (a and c) and for the strategy with the p-th value as the aggregated value (b and d). The red dashed line represents the value of the acceptation criterion. c,d : H Ind. = H Indicator.

Stability of the execution time across simulations. To assess the stability of the execution time obtained for the simulations, we run 30 replicates of the reference simulation and 30 replicates of the simulation with p=365 and p=3652. We use here the approximation with the mean aggregation function. The summary of the results is shown in Table 1. The execution times are stable enough to back the conclusion of substantial performance increase, ie. the standard deviations and relative standard errors are significantly lower than the speed-up.

To answer **RQ1**, the `loop aggregation` provides substantial performance increase while preserving accepted results for the Modflow hydraulic simulator.

p	Number of	Mean	Median	Standard Deviation	RSE
	replicates	(s)	(s)	(s)	(%)
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

Table 1: Variability of time across replicate simulations. RSE = Relative Standard Error.

6.3 Approach robustness

To answer **RQ2**, we experiment `loop aggregation` with other inputs such as climate scenarios (i.e. recharge series) or geographic sites. In these experiments, we use the mean aggregation strategy for the following case studies.

Another climate scenario In this experiment, we use another climate scenario while the rest of the experiment inputs remain the same as in inprevious section. Again, `loop aggregation` leads to a substantial speed up 84.84% (p=90), while remaining within the acceptation criterion.

Replication on other geographical sites. We conduct the same experimentation on 22 other geographical sites. The cumulative execution time amounts to 24 days, 15 hours, 44 minutes and 27 seconds. The speed-up between the reference simulation and the fastest acceptable approximation is illustrated in Fig. 6 across the sites.

Empirically, we find that `loop aggregation` enables an acceptable approximate simulation for all sites. The gains are not homogeneous but they are substantial. The mean and median speed-up are 91.93% and 95.13% with a minimum of 72.26% (Doville) and a maximum of 99.78% (Graye-sur-Mer).

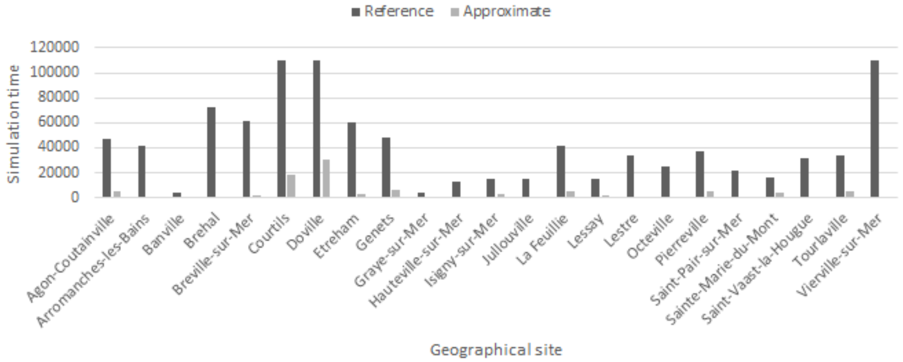


Figure 6: Speed-up across different geographical sites.

To answer **RQ2**, the `loop aggregation` technique provides appealing speed-up while maintaining acceptable results with various inputs (ie. recharge series and geographical sites).

6.4 Threats to validity

Although we empirically validate that the `loop aggregation` approach gives conclusive results for the Modflow scientific simulator in several scenarios, some internal and external limitations remain. Our approach is not analytical but empirical. Our technique may not be the only answer to find trade-off between performance and acceptability with a minimal set-up. The experimentation was carried out on a specific environment. Care should be taken to ensure that the conclusion can be made with other environments. Moreover, while the indicator used to determine the acceptability of the approximated results were given by experts, it may not meet the expectations of other Modflow experts. To mitigate these limitations, the experimentations have been carried out with various inputs (several sites, another recharge series) and a second aggregation strategy. Regarding external limitations, our implementation of `loop aggregation` is limited to a single scientific simulation model based on a differential equation and to aggregating temporal data. Following works are needed to apply the technique with an aggregation on spatial data as well as including other scientific simulation models (eg. other forms of equations).

7 Related Work

Other scientific approaches address the problem of providing trade-off between accuracy and performance. The common goals of these works is to create a surrogate model that can provide an acceptable solution while using less resources. We can classify all these various works in three different categories : the statistical approaches, model reduction approaches and AC approaches.

7.1 Statistical approaches

We gather under the umbrella of statistical approaches all techniques that consist in finding correlation between a set of inputs and outputs of the model. It therefore encompasses most machine learning and regression techniques. These black-box techniques do not impose access to the inner model, but require both access to a large number of inputs/outputs of the model and a training period. At the end of the process, the original model is replaced by the learned model.

In [5], authors present an approach to efficiently explore architectural design spaces through the replacement of simulator by a learned artificial neural network. The artificial neural network is trained through sample inputs which were obtained by repeated execution of the initial simulator.

In [4], authors describe an approach which builds a surrogate model of a mobile network simulator. This surrogate model is then used to guide an optimisation technique. In [13], authors elaborate on a technique which leverage Kriging models to be used for global multidisciplinary design optimisation.

7.2 Model Reduction approaches

Model reduction approaches[6] refer to a set of techniques that aim to reduce the complexity of the simulation model used to represent the natural phenomenon.

The design of the reduced model is done manually and requires both a deep understanding of the natural phenomenon and model reduction techniques. At the end of the process, the original model is replaced by the reduced model.

In [6], authors review the various model reduction techniques used in fluid dynamics systems. In [2], authors present a tailored used of space and time decomposition to transform the original model into several reduced models.

7.3 Approximate Computing approaches

AC approaches[7] refer to a set of techniques that aim to balance accuracy of computing with performance. These techniques leverage the initial software or model and automatically reduce its computation time by performing systematic approximations. These techniques are automatic and systematic and keep the structure of the original model.

In [12], authors present one of the major technique of AC : **loop perforation**. This technique, which is presented in more details in Section 3, consists in reducing the required computing for a software loop by performing only a subset of its iteration. In [10], authors present an **approximate loop unrolling** technique for trading performance of loop over precision. They applied their technique, on computer vision library, music synthesizer library, text search engine and a machine learning library.

Our loop approximation technique is inspired by the approximate loop unrolling technique but as mentioned in Section 4, applying this approach directly on a scientific computation model would radically change the case study and therefore it is not suitable.

Other works have been proposed to reduce the computation of loop iterating over matrices. In [3], authors present an approach, called randomized numerical linear algebra, which intends to remove randomly (according to a distribution law) some data from a matrix to accelerate its processing. As mentioned before, removing data is not a suitable approach since it will completely change the subject of study.

8 Conclusion and Future Work

In this paper, we propose `loop aggregation`, an approximate scientific computing technique, that enables to automatically and systematically reduce the main loop of a simulation model by aggregating the corresponding spatial or temporal data. It can either be implemented as a black-box approximation with a data pre-processing or as a white-box model transformation. Our experimentation on a hydraulic simulator shows a median 95.13% speed-up of the simulation time while preserving acceptable results for all the 23 use cases. The approach is supported with a minimal set-up as opposed to time-consuming model reduction and resource-demanding statistical techniques. The flexibility provided ensures that users can explore the simulations according to their specific constraints.

These results encourage further validation of the `loop aggregation` approach on other scientific simulation models with other forms of equations or other simulation contexts. The technique does not replace the other statistical or model reduction approaches applied to scientific models but rather complements them. Indeed, thanks to its minimal set-up, it can be used during a first approximation phase to generate input/output pairs that can later be used for more efficient statistical approaches as well as allowing a first exploration of the model to better understand it for a possible model reduction approach later on.

References

1. Chaturantabut, S., Sorensen, D.C.: Nonlinear Model Reduction via Discrete Empirical Interpolation. SIAM Journal on Scientific Computing (Sep 2010)
2. Chinesta, F., Ladeveze, P., Cueto, E.: A short review on model order reduction based on proper generalized decomposition. Archives of Computational Methods in Engineering **18**(4), 395 (2011)
3. Drineas, P., Mahoney, M.: RandNLA. Commun. ACM (May 2016), https://dl.acm.org/doi/10.1145/2842602
4. Efstathiou, D., McBurney, P., Zschaler, S., Bourcier, J.: Surrogate-assisted optimisation of composite applications in mobile ad hoc networks. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. pp. 1239–1246. ACM (2014)
5. İpek, E., McKee, S.A., Caruana, R., de Supinski, B.R., Schulz, M.: Efficiently exploring architectural design spaces via predictive modeling. In: Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems. p. 195–206. ASPLOS XII, Association for Computing Machinery, New York, NY, USA (2006)
6. Lassila, T., Manzoni, A., Quarteroni, A., Rozza, G.: Model order reduction in fluid dynamics: challenges and perspectives. In: Reduced Order Methods for modeling and computational reduction, pp. 235–273. Springer (2014)
7. Mittal, S.: A survey of techniques for approximate computing. ACM Comput. Surv. **48**(4), 62:1–62:33 (Mar 2016). https://doi.org/10.1145/2893356
8. Niswonger, R., Panday, S., Ibaraki, M.: MODFLOW-NWT, A Newton Formulation for MODFLOW-2005: U.S. Geological Survey Techniques and Methods 6–A37 (2011), https://pubs.usgs.gov/tm/tm6a37, [Online; accessed 4. Dec. 2019]
9. Razavi, S., Tolson, B.A., Burn, D.H.: Review of surrogate modeling in water resources. Water Resour. Res. **48**(7) (Jul 2012). https://doi.org/10.1029/2011WR011527
10. Rodriguez-Cancio, M., Combemale, B., Benoit, B.: Approximate Loop Unrolling. In: CF 2019 - ACM International Conference on Computing Frontiers. pp. 94–105. ACM, Alghero, Sardinia, Italy (2019). https://doi.org/10.1145/3310273.3323841
11. Rowley, C.W., Dawson, S.T.M.: Model Reduction for Flow Analysis and Control. Annu. Rev. Fluid Mech. **49**(1), 387–417 (Jan 2017). https://doi.org/10.1146/annurev-fluid-010816-060042
12. Sidiroglou-Douskos, S., Misailovic, S., Hoffmann, H., Rinard, M.: Managing performance vs. accuracy trade-offs with loop perforation. In: Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering. p. 124–134. ESEC/FSE '11, Association for Computing Machinery, New York, NY, USA (2011). https://doi.org/10.1145/2025113.2025133
13. Simpson, T., Mauery, T., Korte, J., Mistree, F.: Kriging models for global approximation in simulation-based multidisciplinary design optimization. AIAA Journal **39**(12), 2233–2241 (12 2001). https://doi.org/10.2514/2.1234