Accuracy of Mixed Precision Computation in Large Coupled Biological Systems

Aquilina Al Khoury, Samuel Bernard, Jonathan Rouzaud-Cornabas

Accuracy of Mixed Precision Computation in Large Coupled Biological Systems*

Aquilina Al Khoury[1,2,3], Samuel Bernard[1,3], and Jonathan Rouzaud-Cornabas[1,2]

[1] Inria Grenoble Rhône-Alpes, France
aquilina.al-khoury@inria.fr
[2] Univ. Lyon, INSA Lyon, Inria, CNRS, LIRIS, France
[3] Institut Camille Jordan, Univ. Lyon, CNRS UMR5208, Villeurbanne F-69622, France

Abstract. Thanks to the advancement of knowledge and technologies, we want to simulate larger and larger biological systems. But classical methods must be rethink to be able to cope with such system. At the same time, mainly riding the trends of AI/ML, a novel approach is to use methods mixing different arithmetic precision. Indeed, such method improves arithmetic intensity while saving memory, energy and computational time. However, controlling the errors induced by the mix of different precision remains the major issue. In this paper, we are present a new method allowing the use of mixed precision to solve ordinary differential systems of equations. We evaluate our method against large biological systems. We show that with such systems, we can reduce the arithmetic precision of a part of the biological model will retain the same numerical precision.

Keywords: Mixed Precision · ODEs · Biological Systems.

1 Introduction

1.1 Large Coupled Biological Systems

Systems biology considers biological structures as complex, heterogeneous systems that cannot be reduced to the sum of their parts; *interactions* between basic elements play a pivotal role in biological function [19]. To represent the diversity in the nature of these interactions, mathematical and computational models often take the form of large, heterogeneous coupled systems of basic biological units, with scales ranging from molecules [14], to cells [13] and organisms [27]. Such models take the form of systems of N strongly coupled equations. The state of each unit depends on the other $N-1$ units; the evaluation of the coupling terms, therefore, has a computational complexity in $O(N^2)$, which quickly becomes prohibitive for large values of N.

We have a thorough knowledge and mastery of 3 systems exhibiting these characteristics:

* Supported by the Inria Exploratory Action ExODE.

Neuroscience Modeling in the form of ODEs is used for the simulation of neurons and their synapses. For example, the model [8, 7] of synaptic plasticity has 27 equations per synapse and 15 per astrocytes [9, 8]. A neuron is composed of at least a hundred synapses (several thousand on average). For the moment, the modeling is limited to a small set of neurons and synapses (around one thousand equations), but the goal is to go to several thousands of neurons and synapses (millions of equations). Indeed, a microscopic neural network is composed of at least 1,000 neurons, its simulation requires the ability to solve ODE systems of several million equations. More realistic neural networks are even more larger: 1 million for bee or 86 billion neurons for human.

Cell-Cell Interaction The mechanical and biochemical interaction between cells is important and leads to emergent phenomena in the spatiotemporal organization of tissues, such as synchronization of rhythms, and regulation of cell proliferation or differentiation. These models can be conceived as a set of N (variable over time) cells coupled to each other. Each cell is represented by an ODE system in d dimensions, of the order of a few tens, for a system of total size dN. In the study of a model with heterogeneous cell populations, the total number of cells to be simulated can easily go to several tens of thousands, or even several hundreds of thousands, making simulations prohibitive. For example, the number of cells during an immune response can go from 10^2 to 10^5. Moreover, in some cases, the individual-centered model can be seen as a discretization of a MacKean-Vlasov type transport PDE, for example, and converges when $N \to \infty$. It is therefore interesting to be able to scale up in terms of the number of cells to simulate.

Gene Regulation Networks Aevol platform is an *in-silico* experimental evolution software in the community [24]. An extension of Aevol, R-Aevol [29], adds the modeling of gene regulatory networks. These networks are modeled as ODE systems composed of several hundred equations per simulated organism. For the moment, R-Aevol cannot be used simply because of its high computational cost. However, in order to produce new knowledge in biology, it is necessary in the medium term to simulate more organisms, but also more complex organisms.

In practice, computational models with nonlinear coupling terms are limited to a few hundred or thousand equations, unless simplifying assumptions are made to compute the coupling terms faster. Mean field assumptions for instance reduce the coupling complexity to $O(N)$ [1]. Numerical approaches, such as fast transforms, make use of coupling approximation algorithms that can be computed in $O(N \log(N))$, in a spirit similar to the fast Fourier transform [17]. These numerical approaches are powerful (*e.g.* for the N-body problem [26]), but are difficult to implement for arbitrary coupling terms. Mean field assumptions are not always biological warranted. It is not clear how large numerical systems *need* to be, but recent advances in single-cell omics have highlighted the phenotypic and genotypic diversity of healthy and cancer tissues [11]. The case can be made for scaling up system sizes by a factor 10 to 1000.

A defining feature of biological systems is their capacity to function in presence of *noise* [20], a property often called *robustness*. Here make the working hypothesis that the relevant features are robust to noise to explore the possibility to use low numerical accuracy in numerical simulations of large coupled systems.

1.2 Arithmetic and mixed precision

Computers use a finite set of bits to approximate real numbers, usually 32 or 64 bits (single precision (SP) and double precision (DP)). The storage and arithmetic of SP and DP numbers have been integrated into the silicon of computing units for decades. More recently, mainly due to the high computational need of artificial intelligence (AI) and machine learning (ML) algorithms, half precision types have become more popular. Another approach of handling real number is the usage of arbitrary-precision arithmetic where precision is limited only by the available memory. Nonetheless, it exhibits very poor performance and can not be used for large computation. Matlab has a toolbox to perform arbitrary-precision arithmetic: Variable precision arithmetic (VPA). In this paper, we will use it as great truth for arithmetic precision but only for small to medium size systems as for larger one it becomes prohibitively expensive.

For a long time, computing units are capable of calculating with different precision (SP and DP): it is called multi-precision computing. But it is limited to relying on a unique precision for each part of an application. For few years, computing units are also capable of mixing different precision within a single operation. The goal is to increase computational efficiency without losing too much accuracy.

Several studies have looked at mixed precision and multiple precision computing. Buttari et al. [5, 4] looked at solutions of linear systems using iterative refinement of mixed precision (SP/DP). Kouya [21] extended these studies by adding a mixed double-multiple precision method. These methods perform most arithmetic operations in SP, then post-process these solutions by refining them in DP, direct [31, 10, 30] or iterative [3, 28]

As previously stated, driven by AI/ML exponential grow, hardware manufacturers have also incorporated mixed precision in their architectures (Intel VNNI, Intel Nervana, NVidia TensorCore, Google TPU). They combine the use of half precision (16 bits) and SP (32 bits) (and DP (64 bits)) in order to reduce memory usage and increase computational density. If the hardware has been developed for classical deep learning, the use of mixed precision was also considered for linear algebra [23, 18] and in linear problems that arise from the discretization of partial differential equations [15].

1.3 Motivation

In this paper, we propose a new method to evaluate very large systems of the form $H(X) = F(X) + G(X)$, $x \in \mathbb{R}^{dN}$, as would arise from the discretization of a system of ODEs: $X_{k+1} = H(X_k)$. N is the number of biological units, and d

is the number of dynamical variables in each unit. The function F is the part of H with complexity $O(N)$, and G has complexity $O(N^2)$. F can be thought as the intrinsic dynamics and G as the interaction/coupling dynamics. Systems with cell-cell interaction, such as neural networks or bacterial colonies are examples [27, 9]. We assume that there is no (easy) way to reduce the complexity of G, instead using mixed precision during evaluation of G: the interaction terms are evaluated in reduced precision, while G itself retains working precision. This approach is supported by two observations: 1) biological interactions are inherently noisy, and 2) interactions should obey the law of large numbers. As N becomes large, evaluation errors (and noise) on coupling terms should decrease like $1/\sqrt{N}$.

Let us consider a vector $x \in \mathbb{R}^N$, and, for instance, the coupling functions

$$G_i(x) = \frac{1}{N} \sum_{j=1}^{N} g_j(x_j, x_i), \text{ for } i = 1, ..., N. \tag{1}$$

The coupling term G_i is the average of all interactions from other to unit i. The terms g_j are evaluated at a reduced precision with error ϵ_j. If we assume that the errors are independent and identically distributed with finite variance, we can apply the central limit theorem.

Theorem 1. *If X_1, X_2,...,X_N are N random samples drawn from a population with overall mean μ and finite variance σ^2, and if $\overline{X}_N$ is the sample mean, then the limiting form of the distribution*

$$Z_N = \sqrt{N} \left(\frac{\overline{X}_N - \mu}{\sigma} \right)$$

is the standard normal distribution ($N(0; 1)$)

The cumulative error on G_i, $\frac{1}{N} \sum_{j=1}^{N} \epsilon_j$, would then approach a normal distribution with standard deviation $\sigma/\sqrt{N}$. If, in addition, the errors are not biased, i.e. the mean of ϵ_j is 0, the cumulative error will decrease like $1/\sqrt{N}$.

We apply the method to tow test systems: linearly coupled harmonic oscillators, which offer analytical tractability, and a coupled cell cycle model entrained by coupled circadian clock. We show theoretically and with numerical simulations that accuracy of $H(X)$ increase with $\sqrt{N}$ over relevant range of N. We test how the mixed precision method can be used in practice on a system of ODEs, using existing solvers.

2 Methods

We consider a dynamical system (discrete or continuous) with N coupled units, each described by d dynamical variables. The state of the system is given by a

vector $X \in \mathbb{R}^{dN}$. Denote by $X_i \in \mathbb{R}^d$ the state of the i-th unit. The dynamics (or update rules) is specified by a function $H : \mathbb{R}^{dN} \to \mathbb{R}^{dN}$,

$$H(X) = F(X) + G(X), \tag{2}$$

where

$$F(X) = \big(F_1(X_1), F_2(X_2), ..., F_N(X_N)\big)^T, \tag{3}$$

$$G(X) = \big(G_1(X), G_2(X), ..., G_N(X)\big)^T. \tag{4}$$

The function F describes the intrinsic dynamics of each unit and has complexity of order $O(N)$, while G describes the interactions between units and has a complexity of order $O(N^2)$.

2.1 Selected systems

We evaluate our novel method on different types of systems in order to extend the validity of our reasoning. Al our systems are presented on a large scale to introduce the problem of a balance between precision and error control.

Coupled linear harmonic oscillators We consider N single linear harmonic oscillators is described by a pair of ODEs

$$\frac{dx_i}{dt} = y_i, \tag{5}$$

$$\frac{dx_i}{dt} = -x_i,\ i = 1, ..., N. \tag{6}$$

We add a coupling term $\frac{1}{N}\sum_{j=1}^{n}(x_j - x_i)$ to the right-hand-side of the equation (5). This term is added to introduce $O(N^2)$ complexity. The choice of this system is not random, indeed we wish to treat a linear system while that offers an analytical solution. Let $X = \big(x_1, x_2, ..., x_N, y_1, y_2, ..., y_N\big)^T \in \mathbb{R}^{2N}$. The ODE system for the coupled harmonic oscillators can be written in matrix form. If I_N is the $N \times N$ identity matrix and $\mathbf{1}_N$ is the $N \times N$ identity matrix unit matrix, then

$$\frac{dX}{dt} = BX + CX, \tag{7}$$

where

$$B = \left(\begin{array}{c|c} 0 & I_N \\ \hline -I_N & 0 \end{array}\right), \text{ and } C = \left(\begin{array}{c|c} \frac{1}{N}\mathbf{1}_N - I_N & 0 \\ \hline 0 & 0 \end{array}\right). \tag{8}$$

The right-hand-side of equation (7) has the form $H(X) = F(X)+G(X)$ with $F(X) = BX$ and $G(X) = CX$. The coupling function G can be evaluated in linear instead of quadratic time, but this system aims at presenting a theoretical framework with an exact solution. The $2N \times 2N$ matrix $A = B + C$ possesses the eigenvalues $\lambda_{1,2} = \pm i$ and $N - 1$ repeated $\lambda = (-1 \pm i\sqrt{3})/2$. Therefore the equilibrium solution $X = 0$ is a stable center, which can be difficult for numerical solvers to handle properly.

Cell cycle driven by coupled circadian clocks The choice of the linear system was based on theoretical purposes. Here we look at a more realistic model that present computational challenges, especially in terms of precision and the margin of errors.

The circadian clock is a smooth oscillator with a period of approximately 24h, while the cell cycle is often viewed as a relaxation oscillator (with a fast-slow dynamics). The interaction between these two oscillators has been established experimentally. We consider here a simplified version of a model previously developed [12]. The circadian clock is modeled by a two-dimensional Goodwin model, and the cell cycle is modeled by a FitzHugh-Nagumo model. The Goodwin+Fitzhugh-Nagumo model is denoted G2FHN. The Goodwin model describes a negative feedback oscillators in cellular system, such as circadian rhythms or enzymatic regulation (*e.g.* lactose in bacteria). The Fitzugh-Nagumo was originally developed to describe the electro-physiology of neurons. We use it here to describe the balance between different protein complexes driving the sharp transition to mitosis during the cell cycle. The equations for the intracellular circadian clock (2D Goodwin model) are

$$\frac{dx}{dt} = k_0\theta^h \frac{ax^2 + K\bar{C}}{\theta^h + y^h} - k_1 x, \tag{9}$$

$$\frac{dy}{dt} = k_2(x - y). \tag{10}$$

The equations for the cell cycle (FitzHugh-Nagumo excitable loop) are

$$\frac{dv}{dt} = v - \frac{v^3}{3} - w + v_{\text{on}}(x), \tag{11}$$

$$\frac{dw}{dt} = \epsilon(v + b - cw). \tag{12}$$

The degradation rate of x, k_1, is a random parameter following a normal distribution with mean $\bar{k}_1$ and standard deviation σ_1. The parameter $\theta^h = k_1/(k_0 - k_1)$. The term

$$v_{\text{on}}(x) = I_0 \frac{x^2}{k_3^2 + x^2}$$

is responsible for driving the cell cycle by the circadian clock in each cell. Parameter values are $K = 4.0$, $k_0 = 2.0$, $\bar{k}_1 = 0.339278$, $\sigma_1 = 0.090909$, $k_2 = 0.144832$, $h = 4$, $K_3 = 2.0$, $a = 2.0$, $\epsilon = 0.228249$, $b = 0.7$, $c = 0.8$, $I_0 = 0.5669$. The system of these four ODEs is repeated for N cells. Circadian clock coupling from

cell j to cell i is given by the term $\arctan(x_j - x_i)$, and then averaged over all the cells. For cell i, $i = 1, ..., N$,

$$\bar{C}_i = \frac{1}{N} \sum_{j=1}^{N} \arctan(x_j - x_i). \tag{13}$$

Denoting $X_i = (x_i, y_i, v_i, z_i)^T \in \mathbb{R}^4$, and $X = (X_1, ..., X_N)^T \in \mathbb{R}^{4N}$, the G2FHN model can be expressed as

$$\frac{dX_i}{dt} = F_i(X_i) + G_i(X), \tag{14}$$

with

$$G_i(X) = k_0 \theta^h \frac{a x_i^2 + K\bar{C}_i}{\theta^h + y_i^h}, \tag{15}$$

and F_i given by the rest of equations (9-12).

2.2 Mixed-precision evaluation of the coupling term

These errors are independent of the machine error and are produced because of the rough diagrams on which the numerical methods are based. We will not elaborate much on this notion, since we seek through our study to highlight machine precision and its optimization.

Machine precision, or machine epsilon, denoted ε_m is defined as the different between 1.0 and the smallest encodable floating point number larger than 1. (Matlab definition). In single precision, $\varepsilon_m \simeq 10^{-7}$, while in double the precision ε_m is almost 10^{-15}. Indeed, machine precision depends on the size of the mantissa, it does not depend directly on the number of encoding bits. Single precision has a mantissa of 23 bits. To get the number of digits from binary to decimal, we can divide by 3.3 or so $(\log(10)/\log(2))$, so $23/3.3 = 7$. In double precision, the mantissa has 52 bits, which is more than twice single precision: $52/3.3 = 15$.

We tend to believe that these details are largely sufficient, in this case we will have totally ignored the iterative aspect of numerical schemes and methods causing propagation and especially an accumulation of rounding errors.

The following example highlights the effect of iterations and serves for a better understanding of the explained concepts. Let Y be the result of N successive multiplications, then the relative error is equal to $\dfrac{\delta Y}{Y}$ and varies as $\sqrt{N}\varepsilon_m$. For a more concrete framework, after 100 million multiplications per second for 24 hours with a program representing the reals in a simple precision ($\varepsilon_m \simeq 10^{-7}$, we have the rounding error equal to :

$$\frac{\delta Y}{Y} = 10^{-7}\sqrt{10^8 \times 3600 \times 24} \simeq 0.29$$

which is almost equal to 30%. This result allows us to conclude that a numerical computation requiring a lot of computational time must be done in a DP environment, in order to limit the rounding errors as much as possible.

But what we have just deduced poses a strong contradiction with what we have proposed. Since according to our method we seek to decrease the precision where the code requires a large calculation time and increase this precision elsewhere, whereas following our last reasoning a DP (at most than double if possible) is more than useful for reducing the produced errors and keeping good results.

However, an obvious question arises : how do we always ensure a good error control through our method ?

2.3 Experimental design

Tests will be carried out the coupled harmonic oscillators and the G2FHN model. All tests are done with MATLAB R2020b and its symbolic math package. To do so, each system is evaluated with the following precision: variable precision arithmetics (VPA) with 64 decimal digits of precision, double precision (DP), mixed precision double/single (DPSP), and single precision SP. Evaluations of the functions H at random vector X drawn from uniform distribution in $[0, 1]$ are performed for different system sizes $N = 10^n$, $n = 1, ..., 7$, with 10 runs for each value of N. Additional tests were made in half precision (Matlab `half`). For the mixed precision scheme, the functions F were evaluated in high precision, and G in mixed precision: functions g_i in equation (1) are evaluated in reduced precision, but their averages are accumulated in high precision. For DPSP,

```
G(i) = 1/N*sum(g(single(x-x(i))),'double');
```

For VPA, we limited the tests to $N \leq 10^4$ (G2FHN) and to $N \leq 10^5$ (harmonic oscillator), due to computational time required by the symbolic library used by the VPA. Then, we calculate the error produced between the different precision schemes. For $N > 10^4$, we restrict the number of computed coefficients of H to 10^4, to limit computation times. All results are reported as mean and standard deviations of the error over all runs. ODE simulations of the G2FHN model are carried out in DP and mixed DPSP for $N = 10^1, 10^2, ..., 10^4$, for $t \in [0, 50]$ and mean error on the variable x between DP and DPSP is computed. Sample trajectories for $N = 100$ and $t \in [0, 1000]$ are also computed.

3 Results and discussion

3.1 Mixed precision theoretical convergence

Let $X, \hat{X}$ vectors in high and reduced precision, and ΔX the matrix of differences, with $\Delta_{ij} X = X_j - X_i$ for $i, j = 1, ..., N$. The components G_i of the

coupling functions for the two systems considered here can be expressed as $G_i(\Delta X, X_i)$. The mixed precision error is

$$\begin{aligned} G_i(\Delta X, X_i) - G_i(\Delta \hat{X}, X_i) &= \frac{1}{N}\sum_{j=1}^{N} g_i(\Delta_{ij} X, X_i) - \frac{1}{N}\sum_{j=1}^{N}\big(g(\Delta_{ij}\hat{X}, X_i) \\ &= \frac{1}{N}\sum_{j=1}^{N} g_i(\Delta_{ij} X, X_i), -g(\Delta_{ij}\hat{X}, X_i)\big), \\ &= \frac{1}{N}\sum_{j=1}^{N} \frac{\partial g_i(0, X_i)}{\partial \Delta X}(\Delta_{ij} X - \Delta_{ij}\hat{X}) + O((\Delta X - \Delta\hat{X})^2). \end{aligned}$$

The dominant error term is proportional to $\Delta_{ij}X - \Delta_{ij}\hat{X} = X_j - \hat{X}_j - (X_i - \hat{X}_i) \simeq 2\varepsilon_m$. And at this moment the Theorem 1 intervenes to tell us that if the dominant error term has zero mean, the error on G_i follows asymptotically a normal distribution $N\Big(0, 2\varepsilon_m/\sqrt{N}\Big)$.

So, this affirms us that as N gets larger the errors produced will be compensated.

$$G_i(\Delta X, X_i) - G_i(\Delta \hat{X}, X_i) = C_i O\Big(\frac{1}{\sqrt{N}}\Big),$$

where the constant

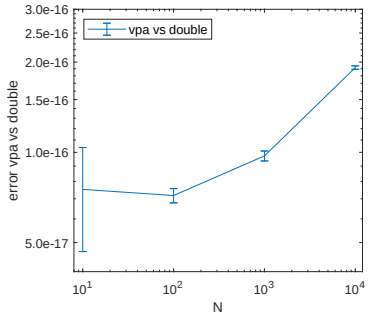
$$C = 2\frac{\partial g_i(0, X_i)}{\partial \Delta X}.$$

3.2 Numerical simulations

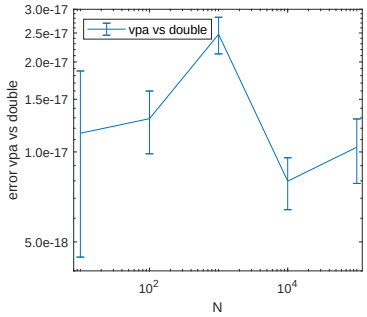
Fig. 1.A shows us that the mean error induced by the evaluation of the G2FHN system in VPA then in a DP is between 7.1765×10^{-17} and 1.9172×10^{-16}, while that of the harmonic oscillator varies between 3.1866×10^{-18} and 2.4748×10^{-17}. Then as shown in Fig. 1a and Fig. 1b, we notice that despite the variability of the induced error it is always of the order of a DP. Following this observation and by considering the precision in VPA (with 64 digits in our case) as a reference for the comparison, we can use the DP while making sure of its capacity to offer a good evaluation of the systems. On the other hand, we observe the tendency of the error to grow with the number N, as we see in the Fig. 1a this error rises quickly and enormously starting 10^3 cells. This is not very reassuring especially that this error increase may not stabilize and then explode with a very large N.

The last observation concerning the mean error of the systems DP affirms our interest to seek an alternative to this precision. We study then the results of our mixed precision DP-SP. We note that following the results offered in Fig. 1a and Fig. 1b we consider the double precision as a reference in our comparisons.

The mean error of the G2FHN system evaluated in double then in mixed precision DP-SP (blue curve of Fig. 2a) is between 3.869×10^{-11} and $4.2039 \times$

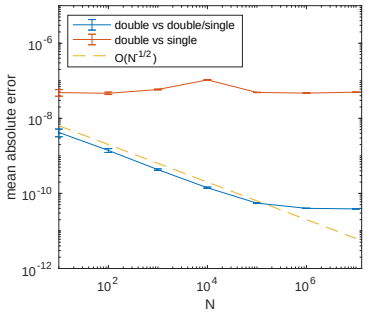


(a) Mean error evaluation ± standard deviation of the G2FHN system in VPA (with 64 digits) then in DP. Results are presented with a log/log scale for N = 10, 10^2, 10^3, 10^4 and with ten repetitions for each value of N.

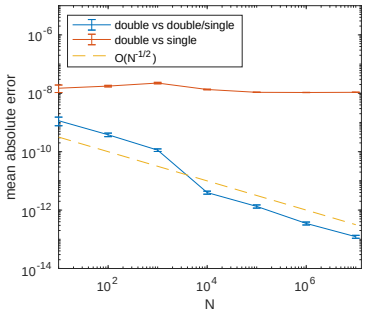


(b) Mean error evaluation ± standard deviation of the harmonic oscillator system in VPA (with 64 digits) then in DP. Results are presented with a log/log scale for N = 10, 10^2, 10^3, 10^4, 10^5 and with ten repetitions for each value of N.

Fig. 1. VPA vs DP in the cases of the G2FHN system and the harmonic oscillator



(a) Mean error evaluation ± standard deviation of the G2FHN system for DP vs DP-SP (blue curve) and DP vs SP (red curve). Results are presented with a log/log scale for N = 10, 10^2, 10^3, 10^4, 10^5, 10^6 and 10^7, with ten repetitions for each value of N



(b) Mean error evaluation ± standard deviation of the harmonic oscillator system for DP vs DP-SP (blue curve) and DP vs SP (red curve). Results are presented with a log/log scale for N = 10, 10^2, 10^3, 10^4, 10^5, 10^6 and 10^7, with ten repetitions for each value of N

Fig. 2. Comparison between DP, DP-SP and SP for the G2FHN and the harmonic oscillator. Errors in DPSP decrease with $1/\sqrt{N}$ (orange dashed lines).

10^{-9}, while that of the same system evaluated in DP then in SP (blue curve of Fig. 2a) varies between 4.6383×10^{-8} and 1.0566×10^{-7}. For the harmonic oscillator (shown in Fig. 2b), this error extends from 1.2258×10^{-13} to 1.1500×10^{-9} for an evaluation in DP then in DP-SP (blue curve of Fig. 2b) and between 1.0644×10^{-8} and 2.2408×10^{-8} for DP vs SP (red curve of Fig. 2b). The decreasing aspect of the two blue curves reveals a remarkable reduction of this error which reaches the order of 10^{11} for the G2FHN as shown on Fig. 2a and 10^{-13} for the harmonic oscillator as shown on Fig. 2b. We are then in the order of a double precision. The obtained results for the mixed precision DP-SP method are very interesting, especially that they affirm that the CLT applies well, then by increasing the number N the induced errors are compensated. Much more than that, the average error not only decreases with the increase of N but better, it stabilizes (see blue curve of Fig. 2a). Moreover, by studying the standard deviation of the mean error for DP vs DP-SP , we can see that it remains acceptable and even negligible. These interpretations of the above figures are very important, not only because they validate our reasoning, but also highlight the efficiency of our diagram of precision especially for large scale systems.

3.3 Extension to Half precision

Following these good results and given the usefulness of the G2FHN system as being a great example of a non linear biological system, on a large scale and with a coupling term, we push then our tests to apply our scheme of precision but with a mixture of single and half precision (noted SP-HP).

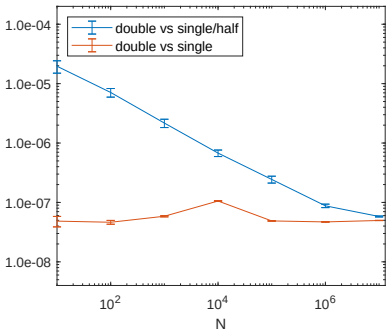


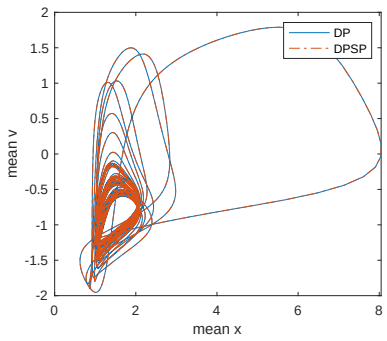
Fig. 3. Mean error evaluation ± standard deviation of the G2FHN system for DP vs SP-HP (blue curve) and DP vs SP (red curve). Results are presented with a log/log scale for N = 10, 10^2, 10^3, 10^4, 10^5, 10^6 and 10^7, with ten runs for each value of N

According to Fig.3, the mean error of the evaluation of G2FHN in double then in mixed precision SP-HP varies between 5.7991×10^{-8} and 1.9630×10^{-5} (blue

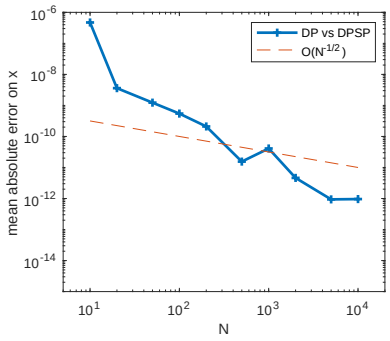
curve on Fig.3). We can see that using this method for small systems is not such a good idea, on the other hand what is captivating is the fascinating reduction of the error for large systems, which is a great result, given the possibility of taking advantage of the material offered by machine learning and increasing the speed of the calculation thanks to GPUs, without worrying about the error induced by such a reduced precision (the half precision). We note that this scheme can also be applied with a mixed precision DP-HP.

3.4 Applications

To test whether using mixed precision could be viable in numerical ODE solvers, we compared ODE simulations of the G2FHN model in DP and DPSP. We chose parameter values in order to make trajectories between DP and DPSP as different as possible (Fig. 4a). We then ran simulations for N ranging from 10 to 10^4. The mean absolute error on the first component x of the system decreases with N (Fig. 4b). This suggests that precision can be adapted to system size.



(a) Mean trajectory of the G2FHN system in the (x, v) phase space in DP (blue) and DPSP (orange). Time ranges from 0 to 1000.



(b) Mean error on x over time with respect to N, for times t between 0 and 50. The error decreases faster than $1/sqrtN$ (orange dashed line).

Fig. 4. ODE simulations of the G2FHN model. Parameters as described, except $K = 10.0$ and $I_0 = 0.64$.

3.5 Related Works

As already mentioned at the beginning of this paper, several studies have been made within the framework of mixed precision. The impact of accuracy on computational time and stability has been studied in the context of ordinary differential equations [25], for high order methods and relatively small system sizes [22].

In the articles of Kouya [21] and [2], we can clearly see the efficiency of the mixed precision. However these two studies are based on the use of the iterative refinement method, therefore the correction of the induced error is made following an increase in calculation operations (an increase in the number of iterations). In addition, the application of this method was limited to an ambiance of linear equations systems. In their article, [2] explain the limits of their method and the conditions for its success. For a direct method in the case of a sparse linear system, we see the exigence of an iterative refinement procedure which converges in a small number of steps and that the cost of each iteration is low compared to the cost of factorization of the system. If the cost of each iteration is too high, then a low number of iterations will result in a performance loss compared to a full double precision solver. For the iterative method, certain conditions must be applied to the "preconditioner" term in order to obtain the desired results.

On the flip side, through our method we end up with the same results concerning the efficiency of the mixed precision and even better sometimes as regards to the reduction of the produced error without even worrying about all the limits presented in other articles. We note that in our scheme of precision, we do not increase the number of operations. Moreover our method is not limited to linear systems of equations, but also applies to non linear cases with a very large scale and a complicated coupling term resulting in the chaotic aspect of the system, as in the case of the G2FHN. All this is done through a very simple method, efficient and supported by a relevant mathematical theorem.

4 Conclusion

To conclude, we propose and validate the great behavior of the induced error of our mixed precision scheme DP-SP while preserving the double precision. This method is justified by a mathematical reasoning which affirms its convergence with an average error of the order of $\frac{\epsilon}{\sqrt{N}}$, but also verified by various numerical tests which show the compensation of the error with the increase of the system size.

As we have already seen, this method is characterized by its simplicity, its efficiency and above all its vast field of application, especially in biology with large and complicated systems. By the way, following all these mentioned advantages we note that through this article the study of the precision was done by considering the rounding error, whereas we know well that this is not the only error involved in optimizing accuracy.

This encourages us to deal with approximation errors, in order to obtain a solver and a numerical scheme compatible with our mixed precision method, so we can be able to offer an optimal precision for large scale systems in future works. In order to do so, we will use existing tools (PROMISE [16] and VerifTracer [6]) to evaluate the numerical quality of our code and quantify the magnitude of floating point related errors. Nonetheless, one of our goal is to improve performance (execution time) of ODE solver. Thus we will do a thorough performance evaluation of our method on the different proposed biological

systems. To conclude, we will assess how our method can benefit from next generation computing platform. Especially, we will work on porting our method to take into account silicon based mixed precision implementations that were tailored for IA/ML.

Acknowledgements

This work has been supported by Inria Exploratory Action ExODE (Scaling the solving of Ordinary Differential Equation for Computational Biology).

References

1. Acebrón, J.A., Bonilla, L.L., Vicente, C.J.P., Ritort, F., Spigler, R.: The kuramoto model: A simple paradigm for synchronization phenomena. Reviews of modern physics **77**(1), 137 (2005)
2. Baboulin, M., Buttari, A., Dongarra, J., Kurzak, J., Langou, J., Langou, J., Luszczek, P., Tomov, S.: Accelerating scientific computations with mixed precision algorithms. Computer Physics Communications **180**(12), 2526–2533 (2009)
3. Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., Van der Vorst, H.: Templates for the solution of linear systems: building blocks for iterative methods. SIAM (1994)
4. Buttari, A., Dongarra, J., Kurzak, J., Luszczek, P., Tomov, S.: Using mixed precision for sparse matrix computations to enhance the performance while achieving 64-bit accuracy. ACM Transactions on Mathematical Software (TOMS) **34**(4), 1–22 (2008)
5. Buttari, A., Dongarra, J., Langou, J., Langou, J., Luszczek, P., Kurzak, J.: Mixed precision iterative refinement techniques for the solution of dense linear systems. The International Journal of High Performance Computing Applications **21**(4), 457–466 (2007)
6. Chatelain, Y., De Oliveira Castro, P., Petit, E., Defour, D., Bieder, J., Torrent, M.: Veritracer: Context-enriched tracer for floating-point arithmetic analysis. In: 2018 IEEE 25th Symposium on Computer Arithmetic (ARITH). pp. 61–68 (2018). https://doi.org/10.1109/ARITH.2018.8464687
7. Cui, Y., Prokin, I., Xu, H., Delord, B., Genet, S., Venance, L., Berry, H.: Endocannabinoid dynamics gate spike-timing dependent depression and potentiation. Elife **5**, e13185 (2016)
8. Cui, Y., Yang, Y., Ni, Z., Dong, Y., Cai, G., Foncelle, A., Ma, S., Sang, K., Tang, S., Li, Y., et al.: Astroglial kir4. 1 in the lateral habenula drives neuronal bursts in depression. Nature **554**(7692), 323 (2018)
9. De Pittà, M., Volman, V., Berry, H., Ben-Jacob, E.: A tale of two stories: Astrocyte regulation of synaptic depression and facilitation. PLOS Computational Biology **7**(12), 1–18 (12 2011). https://doi.org/10.1371/journal.pcbi.1002293, https://doi.org/10.1371/journal.pcbi.1002293
10. Demmel, J.W.: Applied numerical linear algebra. SIAM (1997)
11. Efremova, M., Teichmann, S.A.: Computational methods for single-cell omics across modalities. Nature methods **17**(1), 14–17 (2020)
12. El Cheikh, R., Bernard, S., El Khatib, N.: A multiscale modelling approach for the regulation of the cell cycle by the circadian clock. J Theor Biol **426**, 117–125 (2017)

13. Gallaher, J.A., Enriquez-Navas, P.M., Luddy, K.A., Gatenby, R.A., Anderson, A.R.: Spatial heterogeneity and evolutionary dynamics modulate time to recurrence in continuous and adaptive cancer therapies. Cancer research **78**(8), 2127–2139 (2018)
14. Glaser, J., Nguyen, T.D., Anderson, J.A., Lui, P., Spiga, F., Millan, J.A., Morse, D.C., Glotzer, S.C.: Strong scaling of general-purpose molecular dynamics simulations on gpus. Computer Physics Communications **192**, 97–107 (2015)
15. Göddeke, D., Strzodka, R., Turek, S.: Performance and accuracy of hardware-oriented native-, emulated-and mixed-precision solvers in fem simulations. International Journal of Parallel, Emergent and Distributed Systems **22**(4), 221–256 (2007)
16. Graillat, S., Jézéquel, F., Picot, R., Févotte, F., Lathuilière, B.: Auto-tuning for floating-point precision with discrete stochastic arithmetic. Journal of Computational Science **36**, 101017 (2019). https://doi.org/https://doi.org/10.1016/j.jocs.2019.07.004, https://www.sciencedirect.com/science/article/pii/S1877750318309475
17. Greengard, L., Strain, J.: The fast gauss transform. SIAM Journal on Scientific and Statistical Computing **12**(1), 79–94 (1991)
18. Haidar, A., Tomov, S., Dongarra, J., Higham, N.J.: Harnessing gpu tensor cores for fast fp16 arithmetic to speed up mixed-precision iterative refinement solvers. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis. pp. 47:1–47:11. SC '18, IEEE Press, Piscataway, NJ, USA (2018), http://dl.acm.org/citation.cfm?id=3291656.3291719
19. Kirk, P.D.W., Babtie, A.C., Stumpf, M.P.H.: Systems biology (un)certainties. Science **350**(6259), 386–388 (2015). https://doi.org/10.1126/science.aac9505, https://science.sciencemag.org/content/350/6259/386
20. Kitano, H.: Biological robustness. Nature Reviews Genetics **5**(11), 826–837 (2004)
21. Kouya, T.: Mixed precision iterative refinement for solving linear systems of equations using ieee754 double precision arithemtic and multiple precision arithmetic and its applicaton to fully implicit runge-kutta method
22. Kouya, T.: Practical implementation of high-order multiple precision fully implicit runge-kutta methods with step size control using embedded formula. arXiv preprint arXiv:1306.2392 (2013)
23. Li, X.S., Demmel, J.W., Bailey, D.H., Henry, G., Hida, Y., Iskandar, J., Kahan, W., Kang, S.Y., Kapur, A., Martin, M.C., Thompson, B.J., Tung, T., Yoo, D.J.: Design, implementation and testing of extended and mixed precision blas. ACM Trans. Math. Softw. **28**(2), 152–205 (Jun 2002). https://doi.org/10.1145/567806.567808, http://doi.acm.org/10.1145/567806.567808
24. Liard, V.F., Parsons, D.P., Rouzaud-Cornabas, J., Beslon, G.: The Complexity Ratchet: Stronger than selection, weaker than robustness. In: ALIFE 2018 - the 2018 conference on artificial Life. pp. 1–8. Tokyo, Japan (Jul 2018). https://doi.org/10.1162/isal_a_00051, https://hal.archives-ouvertes.fr/hal-01882628
25. Murray, L.: Gpu acceleration of runge-kutta integrators. IEEE Transactions on Parallel and Distributed Systems **23**(1), 94–101 (Jan 2012). https://doi.org/10.1109/TPDS.2011.61
26. Rein, H., Liu, S.F.: Rebound: an open-source multi-purpose n-body code for collisional dynamics. Astronomy & Astrophysics **537**, A128 (2012)
27. Rocabert, C., Knibbe, C., Consuegra, J., Schneider, D., Beslon, G.: Beware batch culture: Seasonality and niche construction predicted to favor bacterial adaptive diversification. PLoS computational biology **13**(3), e1005459 (2017)

28. Saad, Y.: Iterative methods for sparse linear systems. SIAM (2003)
29. Vadée-Le-Brun, Y., Rouzaud-Cornabas, J., Beslon, G.: In Silico Experimental Evolution suggests a complex intertwining of selection, robustness and drift in the evolution of genetic networks complexity. In: Artificial Life. Proceedings of the Artificial Life Conference 2016, MIT Press, Cancun, Mexico (Jul 2016), https://hal.archives-ouvertes.fr/hal-01375645
30. Wilkinson, J.H.: Rounding errors in algebraic processes. Courier Corporation (1994)
31. Ypma, T.J.: Historical development of the newton–raphson method. SIAM review **37**(4), 531–551 (1995)