МАТЕМАТИКА И МАТЕМАТИЧЕСКО ОБРАЗОВАНИЕ, 2004
MATHEMATICS AND EDUCATION IN MATHEMATICS, 2004
Proceedings of the Thirty Third Spring Conference of
the Union of Bulgarian Mathematicians
Borovets, April 1–4, 2004

FIRST PARALLEL IMPLEMENTATION OF THE CESTAC METHOD WITH SELF VALIDATION*

Jean-Luc Lamotte, David Martins

With the CESTAC method, which allows to validate numerical computation, one needs to run the code several times (in practice 3 times). The CADNA library (Control of Accuracy and Debugging for Numerical Applications) proposes a sequential and synchronous implementation of the CESTAC method. By using CADNA the computation time is multiplied by a factor which varies from 5 to 8 without the self validation of the method and by a factor around 15 with the self validation. These ratios are too important. This paper presents the first results obtained with a parallel implementation of the CESTAC method which includes the self validation.

1. The CESTAC Method. The CESTAC Method (Contrôle et Estimation Stochastique des Arrondis de Calculs) has been proposed by M. La Porte et J. Vignes in 1974 [4]. It is a probabilistic method which allows to estimate the roundoff errors propagation due to the finite representation of the real numbers. The main idea is to run the same program with different roundoff errors propagations. Therefore each variable is represented by a set of values. Their common digits are called the exact significant digits, the rest is due to the roundoff errors propagation.

1.1. *The random arithmetic.* On a computer, when an operation between two floating point numbers is done, in many cases, the result P can not be represented as a floating point number. The FPU must round off the result to the first upper floating point number P^+ or the first lower floating point number P^-. The IEEE arithmetic defined 4 rules to choose P^- or P^+: rounding to the nearest, toward zero, toward $+\infty$ and toward $-\infty$. The random arithmetic proposes to use another rounding mode. If the result P is not a floating point number, the value P^- or P^+ is chosen with a probability $\frac{1}{2}$.

1.2. *The CESTAC method.* N executions of a same program using the random arithmetic allow to obtain N samples of all the variables. The theory about the CESTAC method [1,2,5] shows that we can choose as the value of any variable R the mean value $\overline{R}$ of the N samples and we can estimate its number of exact significant digits $C_{\overline{R}}$ with the formula 2.

$$\overline{R} = \frac{1}{N}.\sum_{i=1}^{N} R_i \tag{1}$$

*2000 Mathematics Subject Classification: 65G50, 65G20, 65Y05.
Key words: CESTAC methd, numerical validation method, parallel implementation.

$$C_{\overline{R}} = \log_{10}\left(\frac{\sqrt{N}.\left|\overline{R}\right|}{s.\tau_\beta}\right) \quad \text{and} \quad s^2 = \frac{1}{N-1}.\sum_{i=1}^{N}\left(R_i - \overline{R}\right)^2 \tag{2}$$

$C_{\overline{R}}$ corresponds to the number of decimal digits in common between the computed result $\overline{R}$ and the mathematical result r.

Practically the best implementation is done with $N = 3$ and $\tau_\beta = 4.303$

1.3. *Conditions for a correct use.* A theoretically study has been done by J.-M. Chesneaux and published in [1, 2]. In this paper, we recall the main results. First, to apply correctly the CESTAC method, it is necessary to perform a dynamical control of all the multiplications and all the divisions. If a multiplication between two numbers which have no exact significant digit, or a division with a divisor which have no exact significant digit is performed, then the CESTAC method is invalidated. These two verifications are called **the self validation**.

Secondly, the 3 executions of a program with the CESTAC method must follow the same way. It is not possible that different instructions are executed for the various samples after the same test.

These two previous points yield necessarily the synchronization of the 3 executions which allows at any time to access all the samples of a variable for:

- the evaluation of the tests,
- the computation of the number of exact significant digits,
- the self validation (test of multiplications and divisions),
- the display of the results.

1.4. *The CADNA software.* The CADNA (Control And Debugging of Numerical Applications) software is a sequential and synchronous implementation of the CESTAC method. It defines new types (the stochastic types) and all the operations on the real numbers have been overloaded. All information about CADNA can be found on the website: `http://www.lip6.fr/cadna`.

1.5. *Performance of the sequential implementation.* To compare the different implementations, we will use two programs :

Matrix multiplication multiplies two square matrix with the classical algorithm.

Gaussian method solves of a linear system with the gaussian elimination method with total pivoting.

These two applications have been run on Pentium III 450 Mhz processor under the Linux system. The table 1 shows the run times in seconds obtained using the IEEE arithmetic and the stochastic arithmetic implemented by the CADNA library with and without the self validation. The over-cost due to the numerical validation is important (from 6 to 7) and the self validation multiplies it by a factor 2.

A parallel implementation can be a response to reduce the over-cost of the CADNA library.

Application	IEEE time	CADNA with self validation time	CADNA with self validation ratio	CADNA without self validation time	CADNA without self validation ratio
Gauss.	13.5	194	14.4	97	7.18
Multi.	14.1	139	9.8	83.3	5.91

Table 1. Computation time obtained with the standard IEEE arithmetic, the CADNA software with and without self-validation

2. Parallel implementation.

2.1. *Without self validation.* The parallel software architecture is based on 3 processes which run synchronously. All together, they form an entity named computation box. Each process computes one sample of all the variables. Some functions need communications between the 3 processes to take the same decisions. There are listed here:

- the comparison functions,
- the absolute value function,
- the conversion function from a stochastic type to an IEEE type,
- the computation of the number of exact significant digits of a value.

The communication system hold in our implementation is based on a distributed exchange of the values between all the processes (see figure 1). Therefore, the processes can all compute the same function and can obtain the same result.

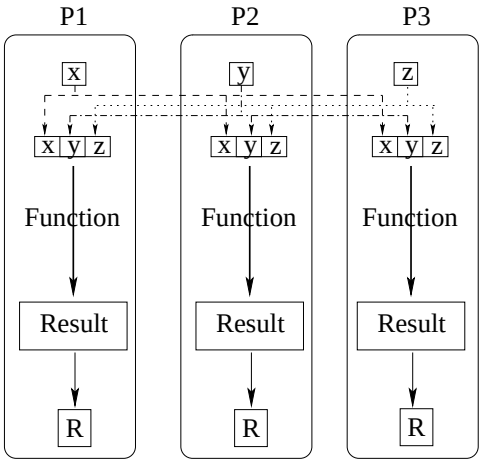

Fig. 1. The decision system between our 3 processes

The performance of this implementation without self validation has been presented in the SCAN2002 conference [3].

Without self-validation, the over-cost of the CADNA library is reduced from 7 with the sequential implementation to 3.5 with our parallel implementation with 3 processes.

The time ratio is therefore around 2. If we decide to use the self validation with this parallel implementation, the performances become disastrous because each multiplication and each division require an exchange between the 3 processes.

2.2. *With self-validation.* The new idea for the parallel implementation is to desynchronize the computation and the self validation. In the sequential implementation of the CESTAC method, the tests of the operands of the multiplication and the division are performed during the operation. A special function is called if a numerical problem is found. Therefore, with a debugger, it is possible to stop the program on this special function and to make a true numerical debugging.

With the parallel implementation, the desynchronization of the operations and of their validation tests should improve the performance in run time, but has an inconvenience: it will not be possible to use a debugger to find the numerical problem with the parallel implementation of CADNA library. However, in the two cases, at the end of run, the check-up of the numerical problems is listed. If the user wants a low run time, he can use the parallel version with self validation, if he wants a numerical debugging of this code, he will use the sequential version.

The new architecture is based on the previous architecture in which a validation box of several processes is added. When a multiplication or a division is computed, the operands are stored in a buffer on each processor. When the 3 buffers are full (this happens at the same time on the 3 processes because they run the same code), the processes send their buffer to the validation box. The validation box is a set of processors which receive the buffers of operators and of operands. They only test if the operation is valid or not. In the last case, a instability counter is incremented. The figure 2 shows the system architecture with 5 processes P_i. At the end of the run, a check-up of instabilities is given.

3. The performance. To obtain efficient performance, our parallel system must be run with 3 processes for the computation box and P_{vb} processes for the validation box. The value of P_{vb} is now difficult to choose because it depends on the number of multiplications and divisions in the numerical program. If vb is too small the validation box will be a bottleneck which will slow down the computation box. If P_{vb} is too important, most of the processes will not have computation to perform. The efficiency will become low.

This architecture poses two questions: what will be the size of the buffers? There is no precise answer to this question. The efficient size depends on the performances of the network (latency and bandwidth). We have decided to use a minimum of two buffers per computation process. When the first one is full, it is sent to the validation box with a non-blocking function which allows to continue the computation. When the second buffer is full, the sending of the first one must be ended, etc ...

The table 2 presents the run times in seconds obtained with a program of square matrix multiplication in function of the buffer size and of the number of processes for the validation box for the parallel version. The computations have been done on the IBM Power 4 Regatta named *Zahir* in the IDRIS. Center*

Before the beginning of the discussion about the performances of our parallel implementation, it is interesting to note that with a resolution program of linear system with

*(http://www.idris.fr)

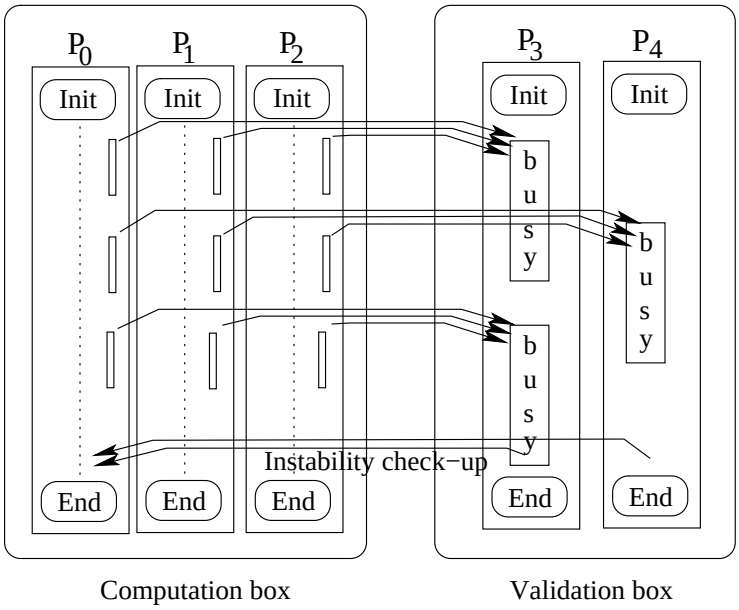

Fig. 2. Architecture system: this example implements 3 processes for the computation box and 2 processes for the validation box

	Matrix size : 2000 × 2000 CADNA Sequential without self validation computation time : 4019 Number of processes for the Validation box		
	2	3	4
Buffers size	Time (Speed-Up)	Time (Speed-Up)	Time (Speed-Up)
1000	2700,39 (1,48)	2644,91 (1,52)	2560,7 (1,60)
2000	**2160 (1,86)**	2295,08 (1,75)	2300,15 (1,74)
3000	2301,09 (1,74)	2361,67 (1,70)	2391,05 (1,68)
4000	3734 (1,07)	2325,16 (1,72)	2208,56 (1,81)

Table. 2. Performance measurement obtained with the matrix multiplication program (time are in seconds and speed-up related to the sequential version of CADNA)

the gaussian elimination method with total pivoting, similar results have been obtained. The optimal speed-up is reached with 2 buffers and 2 processes for the validation box. The values obtained 1.72 et 1.70 respectively with a system of size 1000 and 2000, the other values being clearly less good.

3.1. *Discussion.* The choice of our example is important. The matrix multiplication program is very interesting because the main instruction of the program is: $c_{ij} = c_{ij} + a_{ik} * a_{kj}$. This program contains a lot of multiplications, exactly one multiplication for one addition. This program will be a good test because it will generate a lot of operations to validate.

The evaluation of our validation box separated from the computation box is satisfactory. The aim (obtaining the same computation time with and without self validation) is nearly reached for the resolution of linear system and almost reached for the multiplication program with two validation processes and two buffers (size: 2000 elements). These performances are near those obtained without self-validation.

The difference between the ratio 1.7 for the resolution of the linear system and the ratio 1.9 for the matrix multiplication is normal. The matrix multiplication does not need any exchange. It only uses addition and multiplication operators. In the opposite, the total pivoting in the gaussian elimination method needs calls to the test function and to the absolute value function which generate a lot of exchanges.

4. Conclusion The aim to obtain nearly the same run time between the parallel implementation with and without self validation is reached. The next step in our study is to validate the computation time on other parallel systems and to find automatically the optimal size of the buffer and the optimal processes number for the validation box.

Acknowledgments. The author thanks IDRIS-CNRS for the support during the use of their IBM Power 4 Regatta.

REFERENCES

[1] J.-M. Chesneaux. Study of the computing accuracy by using probabilistic approach. Contribution to computer arithmetic and self-validating numerical methods. (Ed. C. Ulrich), Publisher J.C Baltzer, 1990. 19–30.
[2] J.-M. Chesneaux. L'Arithmtique Stochastique et le Logiciel CADNA. Habilitation à diriger des recherches, Université Pierre et Marie Curie, Novembre 1995.
[3] J.-L. Lamotte. Parallelization of the CESTAC Method on shared memory and distributed memory computers. Proceeding of the SCAN2002 conference, Paris, 2002, 124.
[4] J. Vignes, M. La Porte. Error analysis in computing. *Information Processing*, **74**, 1974.
[5] J. Vignes. Discrete stochastic arithmetic for validating results of numerical software. *Numerical Algorithms*, 2004, to appear.

Laboratory LIP6, University Paris 6
4 place Jussieu
75 252 Paris cedex 5
France
e-mail: Jean-Luc.Lamotte@lip6.fr

ПЪРВА ПАРАЛЕЛНА РЕАЛИЗАЦИЯ НА CESTAC МЕТОДА С ВАЛИДАЦИЯ

Ж.-Л. Ламот, Д. Мартинс

С метода CESTAC, който позволява верифицирани изчисления, програмният код трябва да бъде изпълняван няколко пъти (на практика 3). Библиотеката CADNA предлага последователна и синхронна реализация на CESTAC метода. Използвайки CADNA времето за изпълнение се умножава с фактор 5–8 без валидация на метода и с фактор около 15 ако има валидация. Статията представя първите резултати получени от паралелната реализация на CESTAC метода с валидация.