Role of precision in meteorological computing: a study using the NMITLI Varsha GCM

T. N. VENKATESH* and U. N. SINHA

Flosolver Unit, National Aerospace Laboratories,
PB 1779, Bangalore, 560017, India
**E-mail: tnv@flosolver.nal.res.in*

An issue in large scale computing of unsteady phenomena is the role played by round-off errors. These errors can accumulate in long integrations leading to major differences in the computed quantities such as rainfall. Computations with the same code at different precisions (single, double and arbitrary precision) have been made to assess the role of round-off. For the study, Varsha2C and Varsha2C-MP, C and multi-precision versions of the spectral, hydrostatic Varsha GCM have been used. Varsha2C-MP uses the ARPREC library for computing using an arbitrary number of digits.

We find that even for sequential runs, round-off causes solutions to differ beyond 12 days when two different processors are used. With the same processors, the results of parallel runs made using a different number of processors differ significantly after 10 days, due to the round-off errors in the global summation operations. Use of double precision and changing the communication strategy reduces the differences by about 10 %. With the use of multi-precision, repeatable results, both across different processor types and different number of processors, can be obtained for up to 30 days of integration, provided 64 or more digits are used for the computing. For the Lorenz system of equations, however, precision has no significant effect on the results. Calculations using up to 2048 digits show that given the same initial perturbation, the time at which two solutions diverge (by order 1) does not depend on the precision. These results suggest that, for long integrations of meteorological models, far more careful computing is required than is generally appreciated.

Keywords: Round-off errors, multi-precision computing, Varsha GCM

1. Introduction

The field of meteorological computing is complex and has many dimensions to it. To obtain realistic and meaningful results one has to ensure that the physical model adopted, and the simplifying assumptions, numerical approximations made are reasonable; the algorithms and the computer code used are correct; and finally that the machine does give the intended output. While there is an extensive literature on modelling and numerics, there is relatively less work on the code and machine dependent parts even though it is known to practitioners that some results are sensitive to the operating system and even the compiler optimizations used. With the use of massively parallel architectures and heterogeneous environments (in grid computing), the issue of reproducible results across different processor types and varying number of processors assumes even greater importance. The growth of round-off errors is not usually a problem for short (3 days) to medium range (7-10 days) integrations, but has to be addressed for longer (30 day, seasonal or climate) integrations. Since one is trying to isolate the effect of some small parameter, like variations in CO_2 concentration, it is important to ensure that there is no numerical noise clouding the results.

We approach this problem by looking at what effect using a higher number of digits has on the results of 30 day simulations using the Varsha GCM of the National Aerospace Laboratories (NAL) Bangalore. The quantity of interest is the monsoon rainfall over India. The C version of the Varsha GCM was modified to enable multi-precision computing for this study. We find that the precision used has a significant impact on the computations beyond ten days of integration.

The organization of this paper is as follows. A brief description of the Varsha GCM and its different versions is given in section 2. The main results with the GCM computations are presented in section 3 and results with the simple Lorenz model in 4. The discussion and conclusions are in section 5.

2. Varsha GCM

The Varsha GCM code version 1.0. is a hydrostatic spectral general circulation model developed at the National Aerospace Laboratories (NAL) as part of the Government of India's NMITLI project. The code has its roots in the GCM T-80 code of National Centre for Medium Range Weather Forecasts (NCMRWF), India. The NCMRWF code was in turn based on NCEP spectral model [1]. This code was ported to a UNIX platform from the CRAY specific vectorized code and parallelized for a distributed memory (MPI type library) by NAL [2] in 1993. It was subsequently re-engineered [3] using FORTRAN 90 and, as part of the NMITLI project, new radiation and boundary layer modules were added [4]. The model can be run at different spectral truncation as well as physical grid resolutions. The number of vertical layers is 18 and the physical parameterizations include the Kuo-Anthes cumulus scheme. The shortwave radiation is computed as described in Sinha et al 1994 [2], while there are two options for the long wave radiation computation: (i) The Fels-Schwarzkoff scheme and (ii) a new scheme devised for Varsha based on Varghese et al [5]. For the boundary layer the options are (i) the Monin-Obukhov scaling along with a gustiness parameter, and (ii) a new boundary layer scheme based on the scaling arguments of Rao and Narasimha [6].

In addition to Varsha 1.0, Varsha2C, a new code written completely in ANSI C, has been developed. This version has the added feature of being able to change from single to double precision run with the change of a compile time flag. Another version the Varsha2C-MP, which can calculate quantities to an arbitrary number of digits, is described in more detail in the next section.

2.1. *Multi-precision version of Varsha GCM*

```
#include <stdio.h>
#include <stdlib.h>
#include "mpi.h"
#ifdef MULTI_PRECISION
  #include "mp_real.h"
#endif

void agcm(int *,int *);
int main(int argc, char *argv[])
{
  int numprocs,myid;

  MPI_Init(&argc,&argv);
  MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
  MPI_Comm_rank(MPI_COMM_WORLD,&myid);

#ifdef MULTI_PRECISION
  mp::mp_init(128);
#endif
  agcm(&numprocs,&myid);
#ifdef MULTI_PRECISION
  MPI_Barrier(MPI_COMM_WORLD);
  mp::mp_finalize();
#endif
  MPI_Barrier(MPI_COMM_WORLD);
  MPI_Finalize();
  return 0;
}
```

```
//  precision is controlled here
#ifdef MULTI_PRECISION
    #include "mp_real.h"

    typedef mp_real real;
#else
  #ifdef DOUBLE_PRECISION
    typedef double real;
  #else
    typedef float real;
  #endif
#endif
```

Fig. 1. Segments of the Varsha2C-MP code showing the changes required to use the multi-precision data type *mp_real* and control the number of digits.

Varsha2C-MP, the multi-precision version of the Varsha GCM, uses the ARPREC multi-precision library to calculate quantities to an arbitrary number of digits. The ARPREC package can perform arbitrary-precision arithmetic [7]. It is written in C++ and is portable across many platforms. A new type call *mp_real* can be defined and all the usual arithmetic operations performed on variables of this type. The number of digits used for computation can be set at run time depending on the memory available. The main advantage is that existing codes can be converted to use this data type with minimal changes.

Since the ARPREC package can be linked to C++ programs, some minor changes were required to cast the Varsha2C into a C++ compatible code. Code segments showing the declaration of the *mp_real* data type and the function call to set the number of digits are shown in Figure 1. While ARPREC and similar multi-precision packages have been used for studies in number theory and vortex sheet computations, to the knowledge of the authors, Varsha2C-MP is the first code of this complexity (spectral atmospheric GCM with over 25 thousand lines of code) to be software-engineered for arbitrary precision computing.

3. Results

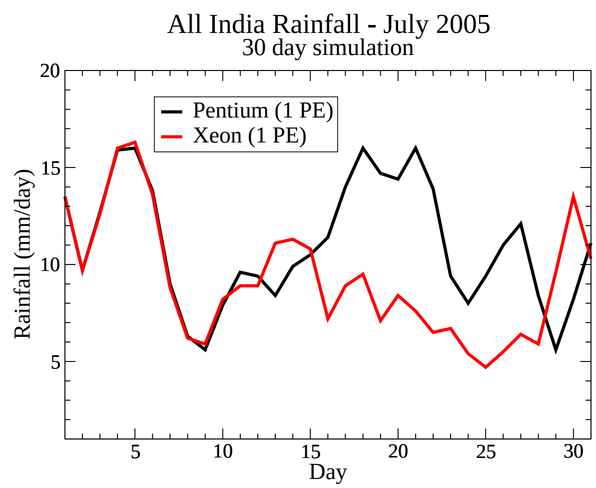

Fig. 2. Comparison of the all India rainfall computed with sequential runs of the Varsha 1.0 code on Intel Pentium and Xeon processors.

There are two reasons for using rainfall as a variable for studying the sensitivity of the computation to the precision used. It is first of all an important quantity during the monsoon in India [8]. Secondly rainfall parameterizations usually have conditional statements, table lookups (for the moist adiabat processes), making the computed rainfall a very sensitive quantity. The All India rainfall, which is the sum over all the grid-points within the country, further makes the variable even more sensitive to small shifts in the spatial patterns. Other quantities like wind velocities and temperature are smoother, and differences are seen only after a longer period of integration.

All the results reported here were started with the initial conditions for 1 July 2005. This month was characterized by two peaks in the rainfall over India. The first peak, was due to heavy rainfall over Gujarat in the first week and the second peak in the fourth week due to very heavy rainfall over Maharastra, in particular Mumbai. One month integrations were carried out. For the single and double precision runs, the horizontal resolution was 120 spectral modes and 80 km physical grid.

All the computations were performed on the Flosolver series of parallel computers [9,10]. The Flosolver Mk6 which uses Intel Pentium processors and the Mk6-X (with Intel Xeon processors) were used. The compilers used were the Intel Fortran compiler for the Varsha GCM 1.0 and the GNU gcc for Varsha2C and Varsha2C-MP.

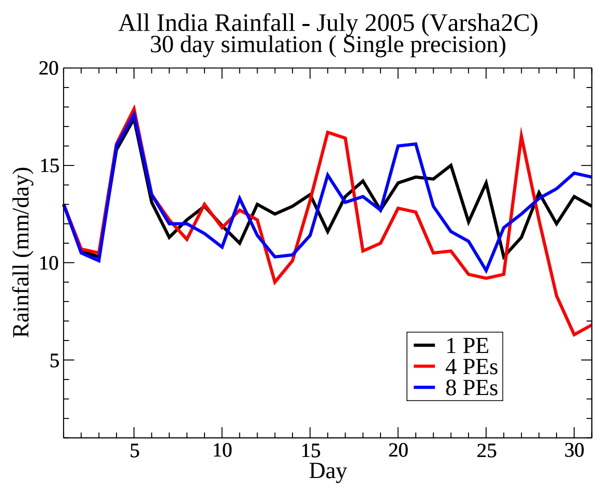

Fig. 3. Comparison of the all India rainfall computed using the parallel Varsha2C code at single precision on Flosolver MK6-X, when the number of processors is varied.

First the lack of repeatability even with a sequential code is shown. In fig. 2, the all India rainfall computed on Intel Pentium and Xeon processors with the Varsha 1.0 code is shown. One can see that the solutions diverge after around 12 days.

The effect of using a different number of processors is seen from Fig. 3. With the Varsha2C code run on one, four and eight processors, the difference in results may be seen from around day 8. The reason for this behaviour is due to the presence of round-off errors during the calculation of the global sums from the partial sums computed on each processor. The global sums are required during the transformation of prognostic variable tendencies from physical to spectral domain. This is confirmed by reduction of such deviations when a higher number of digits is used (results are presented in the following sections).

3.1. *Effect of double precision and ordered summation*

The effect of using double precision is seen from Fig. 4. One can see that there are differences in results starting from around day 10. As compared to the single precision runs, there is an improvement, but not very much.

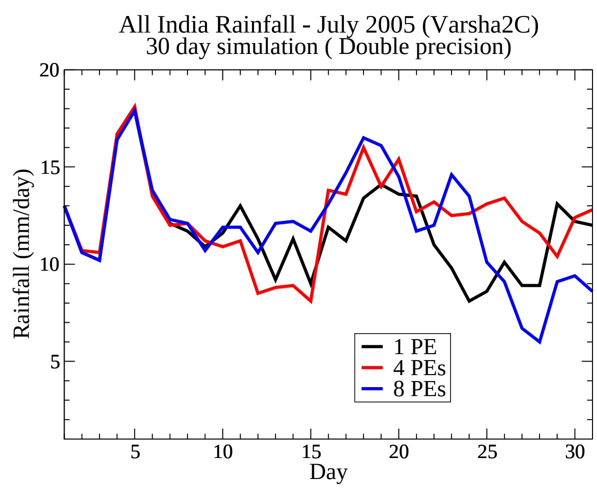

Fig. 4. Comparison of the all India rainfall computed using the parallel Varsha2C code at double precision on Flosolver MK6-X, when the number of processors is varied.

A major portion of the communication in the Varsha GCM is taken by global sums and global maxima (MPI_ALLREDUCE calls). Various optimizations are possible for these global operations, depending on the machine architecture. However, these optimizations come at the expense of changing the order of summations, which can change the results. The role of various strategies of summation has been studied by He and Ding [11]. To reduce such errors, the communication calls were changed as follows.

The global summations were carried out by transferring the partial sums

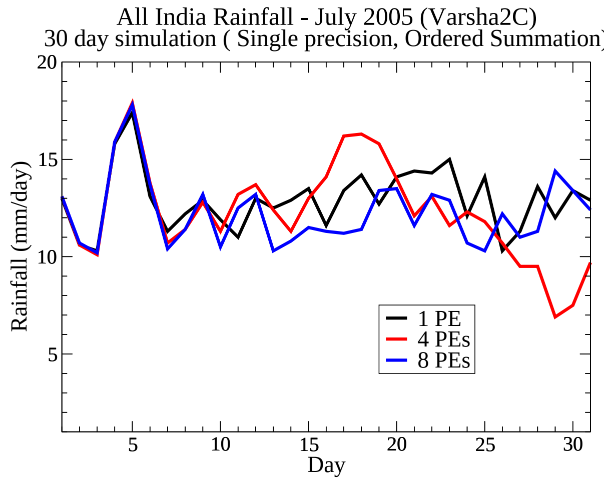

Fig. 5. Comparison of the all India rainfall computed using the parallel Varsha2C code at single precision, with order of summation maintained, on Flosolver MK6-X, when the number of processors is varied.

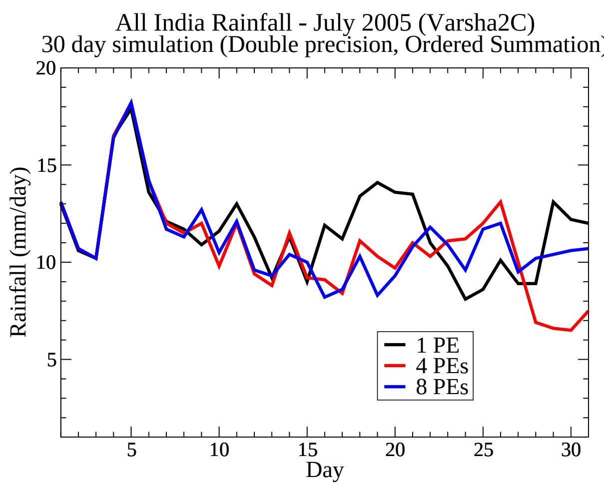

Fig. 6. Comparison of the all India rainfall computed using the parallel Varsha2C code at double precision, with order of summation maintained, on Flosolver MK6-X, when the number of processors is varied.

to one processor, performing the summation and then broadcasting the result to all the processors. This is of course an inefficient method but reduces the errors introduced due to changing the order of summation. The results using this strategy are shown in figures 5 and 6. The spread between the computations on a different number of processors reduces, but still there are significant differences after around 12 days.

3.2. *Multi-precision*

We now report results using the Varsha2C-MP code. All the runs were started with the initial conditions for 1 July 2005. For the multi-precision runs, the horizontal resolution was 60 spectral modes and 150 km physical grid. 32, 64 and 128 digits were used. This lower resolution was due to limitations of the memory and the need to get the results in a reasonable amount of time. The 128 digit runs for example took around 10 days of real time.

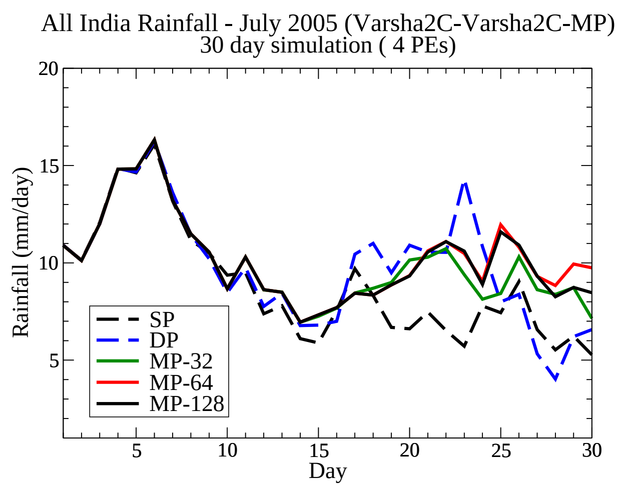

Fig. 7. Variation of the all India rainfall computed at different precisions, using the parallel Varsha2C-MP code and four processors.

The global summations were carried by transferring the partial sums to one processor, performing the summation and then broadcasting the result to all the processors, since the library does not handle operations with the *mp_real* data-type at present.

The effect of precision for a fixed number of processors (four) is shown in Fig. 7. While the single and double precision curves vary, there is convergence for 64 and 128 digits.

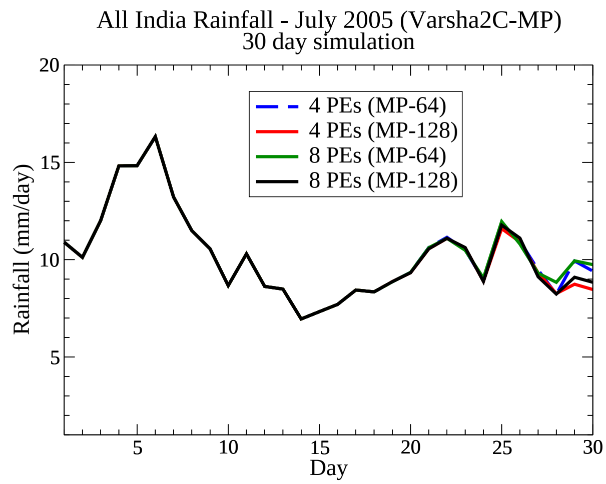

Fig. 8. Variation of the all India rainfall computed using different numbers of processors and different precision levels (64 and 128 digits) with the parallel Varsha2C-MP code.

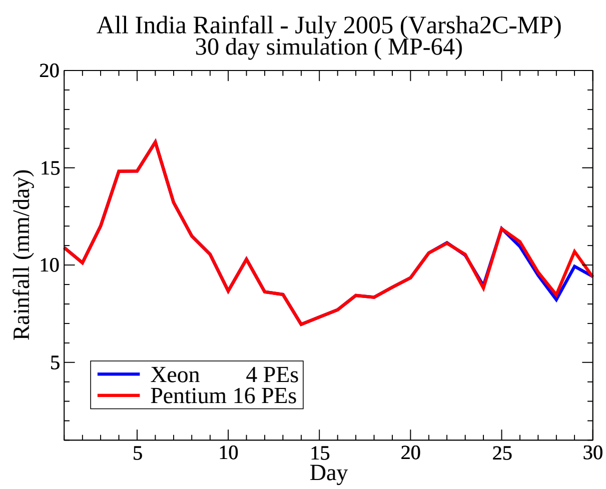

Fig. 9. Comparison of the all India rainfall computed at a precision of 64 digits, on four Xeon processors and sixteen Pentium processors.

The dependence on the number of processors for precision levels of 64 and 128 digits is shown in figure 8. One can see that repeatable results are obtained in contrast to the single and double precision runs. There are only minor differences (around 5 %) in the computed rainfall beyond 25 days.

In figure 9, a comparison of the rainfall, computed using 64 digits, on four Xeon processors and sixteen Pentium processors, is made. The agreement between the results, made on different processor types and different number of processors is in contrast to the divergence for sequential runs using single precision (Fig. 2).

3.3. *Spatial patterns*

Contours of rainfall for different precisions are shown in figures 10, 11 and 12. After ten days, there are only very minor differences between the double-precision and higher precision runs. After twenty days of integration, differences between DP and MP-128 (taken as reference) are evident. By thirty days, MP-32 is different significantly. There are some differences between MP-64 and MP-128 runs also.

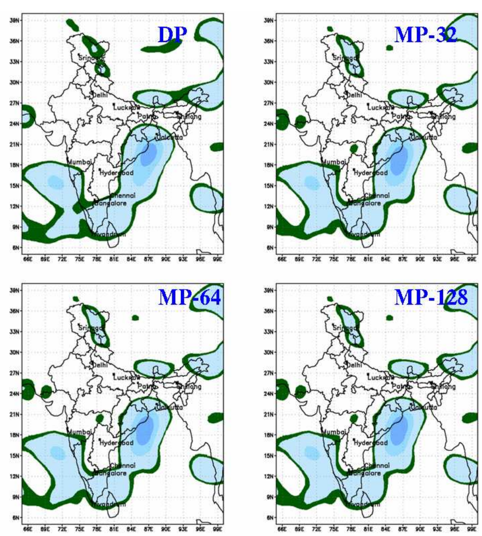

Fig. 10. Rainfall contours after ten days of integration with the Varsha2C-MP code at different precisions. DP : Double precision, MP-32 : 32 digits, MP-64 : 64 digits and MP-128 : 128 digits.

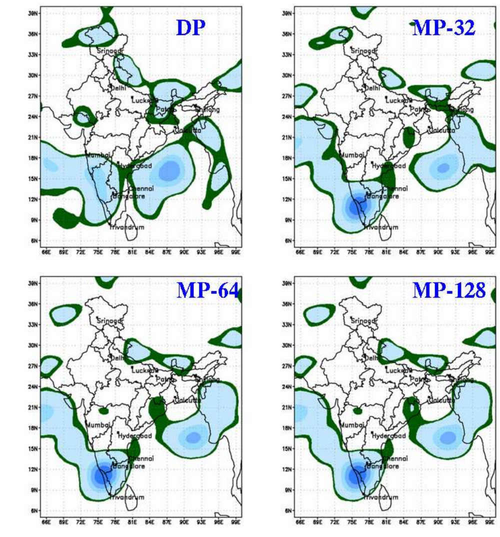

Fig. 11. Rainfall contours after twenty days of integration with the Varsha2C-MP code at different precisions. DP : Double precision, MP-32 : 32 digits, MP-64 : 64 digits and MP-128 : 128 digits.

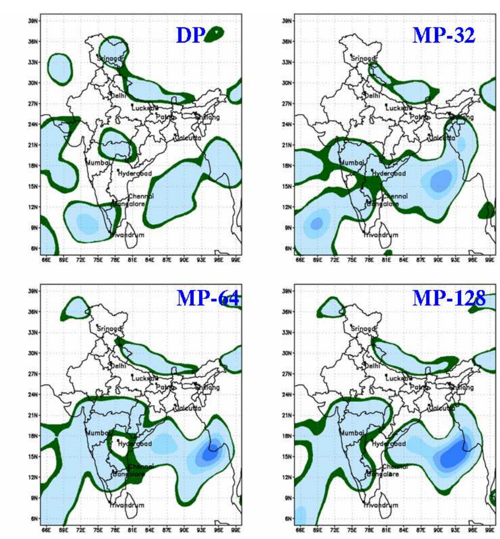

Fig. 12. Rainfall contours after thirty days of integration with the Varsha2C-MP code at different precisions. DP : Double precision, MP-32 : 32 digits, MP-64 : 64 digits and MP-128 : 128 digits.

4. Tests with the Lorenz model

The Lorenz system of equations [12] is very well known and led to the rapid growth of the field of chaos. The "butterfly effect", where in a small change in the initial conditions can lead to order one changes in the solution after some time, has had a major influence on how both the scientific community and the general public view weather and climate. The Lorenz system has also been used widely for a number of studies, including some of the recent innovative works in data assimilation [13]. The motivation for the study was the non-repeatability of numerical results on the computer, when a model run was restarted. The differences between the original computation and the one following a restart was due to the truncation of the number of digits when the results were stored. This small perturbation in the initial conditions (for the run started from the stored values) grew rapidly and made the results differ from the original single run.

In the previous section, we have shown that repeatable results can be obtained using a higher number of digits. The deviation due to round-off is clearly isolated. It is natural to ask what effect precision has when there is a perturbation of the initial conditions (as would be done in ensemble simulations). Such simulations with the Varsha GCM are computationally expensive and will be carried out at a later date. Meanwhile, tests with the Lorenz model have been done and are described in this section.

The Lorenz system of three coupled ODEs are

$$\begin{aligned} \dot{x} &= \mathbf{s}(y - x) \\ \dot{y} &= \mathbf{r}x - y - xz \\ \dot{z} &= xy - \mathbf{b}z \end{aligned} \tag{1}$$

.

Here x, y and z are the dependent variables and $\mathbf{s}$, $\mathbf{r}$ and $\mathbf{b}$ are parameters. The usual values of these parameters are $\mathbf{s} = 10$, $\mathbf{r} = 28$ and $\mathbf{b} = 8/3$. The method of solution is to start with some initial values of x, y, z and integrate the coupled ordinary differential equations using a time integration scheme.

The fourth order Runge-Kutta scheme has been used here. If one considers x, y and z to be the co-ordinates of a point in three dimensions, the trajectory is of interest. This is a function of the initial conditions and the parameters $\mathbf{s}$, $\mathbf{r}$ and $\mathbf{b}$. For most initial conditions, after a certain time, the trajectory lies in a region of space which is independent of the initial conditions. This set is called an attractor.

It is useful to see how the trajectories of two initial conditions which

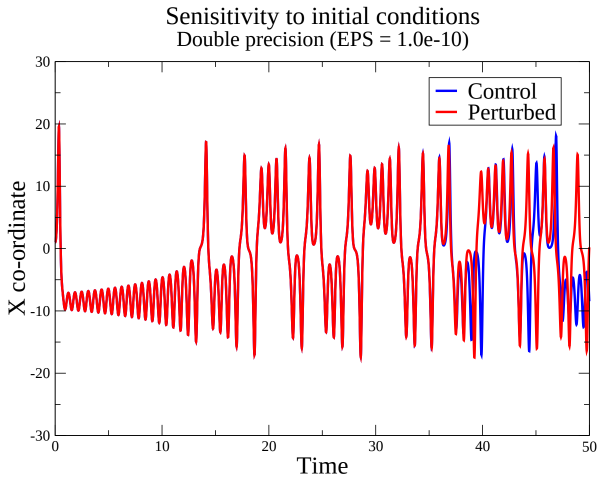

Fig. 13. Sensitivity of the solution to small changes in the initial condition. The values of x co-ordinate as function of time is plotted for the control and perturbed ($x(0) = x(0)_{control} + \delta$) run. The double precision results are shown here.

differ by a small amount behave with time. The sensitivity of the solution to small changes in the initial condition, with calculation made at double-precision, is shown in figure 13. For the control run the initial values of x, y and z were 1,1,1 respectively. For the perturbed run, $x(0) = x(0)_{control} + \delta$ and δ was set to 10^{-10} for most of the studies. One can see from the figure that the solutions diverge after around $t = 32$.

Similar computations were made doubling the number of digits successively. Up to 2048 digits were used. For calculation with higher precision, the results are very similar to the double-precision results. The times at which solutions (control and perturbed runs) diverge by an amount ϵ (1.0×10^{-2}) was found to be 31.89 for calculation at all the precision levels (from double precision upto 2048 digits). It is to be noted that the the time step is 0.01. This is not to say that precision has no effect on the solution. From table 1 one can see that there are changes in the values of the solution computed with double precision as compared to the 2048 digit computation at $t = 30$ and and $t = 40$. The nature of the solution (where chaos starts) does not change with the precision. If δ is decreased, the time at which there is significant departure increases.

Table 1.

Values of x at different times for runs with different precisions

	20.000	30.000	40.000
DP	13.47290574333027	9.866860933476248	-8.743318752850055
MP-16	13.47290574339619	9.866859443550267	-8.534596463420804
MP-32	13.47290574339619	9.866859443550123	-8.534596443596334
MP-64	13.47290574339619	9.866859443550123	-8.534596443596334
MP-128	13.47290574339619	9.866859443550123	-8.534596443596334
MP-512	13.47290574339619	9.866859443550123	-8.534596443596334
MP-1024	13.47290574339619	9.866859443550123	-8.534596443596334
MP-2048	13.47290574339619	9.866859443550123	-8.534596443596334

5. Discussion and conclusions

A complete atmospheric GCM, Varsha2C-MP, with the capability of performing calculations with any desired number of digits (within memory and CPU time limitations) has been developed at NAL. This code is a useful tool to study the effects that round-off errors can have on the computations and can be used for integrations where high precision is required.

Computations with the Varsha2C-MP code clearly show that by using a higher number of digits, the spread of round-off errors can be minimized. It takes 64-digits precision for one month simulations to be done with repeatable results both across different processor types and number of processors.

The implications of this study are as follows: Round off errors cannot be ignored in long-term GCM simulations (such as seasonal runs) as they can cause significant differences in the computed atmospheric fields, especially rainfall depending on processor type and number of processors used in a parallel machine. The deviations in the numerical solution due to round-off errors, while significant, are not like the deviations seen in chaotic systems. For the simple Lorenz model, precision has a minor effect but does not change the nature of the solution.

For complex systems one must be careful to distinguish between intrinsic chaotic behaviour of the system and deviations due to round-off errors. This study has demonstrated that multi-precision computing provides a means of distiguishing between the two.

Acknowledgments

This work was carried out under the Government of India's NMITLI project. The financial support and technical inputs provided by the NMITLI cell, CSIR, is gratefully acknowledged. The authors acknowledge the encouragement of Dr. A. R. Upadhya, Director NAL. It is a pleasure to acknowledge Prof. R. Narasimha for his support and advice through out this programme and critical suggestions regarding the manuscript. The authors further wish to thank all the colleagues at the Flosolver Unit, NAL who have helped in various ways.

References

1. E. Kalnay, J. Sela, K. Campana, B. K. Basu, M. Schwarzkopf, P. Long, M. Kaplan and J. Alpert J, National Centers for Environmental Prediction, Washington DC, USA, (1988).
2. U. N. Sinha, V. R. Sarasamma, Rajalakshmy S., K. R. Subramanian, P. V. R. Bharadwaj, C. S. Chandrashekar, T. N. Venkatesh, R. Sunder, B. K. Basu, S. Gadgil, and A. Raju, *Current Science*, **67**, No. 3, 178–184, (1994).
3. R. S. Nanjundiah and Sinha, U. N., *Current Science*, **76**, 1114–1116 (1999).
4. U. N. Sinha, T. N. Venkatesh, V. Y. Mudkavi, A. S. Vasudeva Murthy, R. S. Nanjundiah, V. R. Sarasamma, Rajalakshmy S., K. Bhagyalakshmi, A. K. Verma, C. K. Sreelekha, and K. L. Resmi, **NAL PD FS 0514**, National Aerospace Laboratories, Bangalore, India (2005).
5. S. Varghese, A. S. Vasudeva Murthy and R. Narasimha, *Journal of the Atmospheric Sciences*, **60**, 2869–2886, (2003).
6. K. G. Rao, and R. Narasimha, 2006, *Journal of Fluid Mechanics*, **547**, 115-135, (2006).
7. `http://crd.lbl.gov/~dhbailey/dhbpapers/arprec.pdf`
8. U. N. Sinha, T. N. Venkatesh, V. Y. Mudkavi, A. S. Vasudeva Murthy, R. S. Nanjundiah, V. R. Sarasamma, Rajalakshmy S., K. Bhagyalakshmi, A. K. Verma, C. K. Sreelekha, and K. L. Resmi, **NAL PD FS0608**, April (2006).
9. U. N. Sinha, M. D. Deshpande and V. R. Sarasamma, *Supercomputer*, **6**, No. 4, (1989).
10. T. N. Venkatesh, U. N. Sinha and R. S. Nanjundiah, in "Developments in Terracomputing", World Scientific, Proceedings of the Ninth ECMWF Workshop on the Use of High Performance Computing in Meteorology held at Reading, England during November 2000. 62 – 72, (2001).
11. Y. He and C. H. Q. Ding, Proceeding of the ninth ECMWF workshop on the use of HPC in meteorology, World Scientific, 296 – 317, (2001).
12. E. N. Lorenz, *Journal of the Atmospheric Sciences*, **20**, 130-141, (1963).
13. E. Kalnay, M. Pena, S.-C. Yang, and M. Cai, Proc. ECMWF Seminar on Predictability, Reading, U.K. (2002).