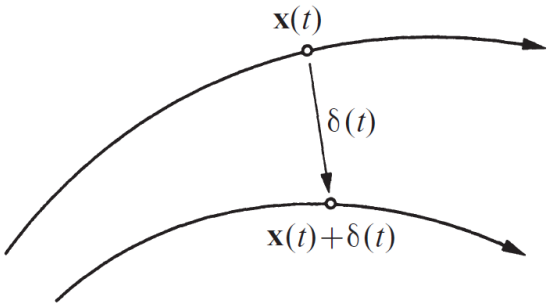
# PARALLEL COMPUTING OF LONG TERM RELIABLE TRAJECTORIES FOR THE LORENZ ATTRACTOR

Ivan Hristov, Radoslava Hristova

*EuroCC conference: "High-performance computing for the benefit of researchers and society", Sofia 21-22 November 2022*

# Sensitive dependence on initial conditions




$$\delta(t) \sim \delta(0)e^{\lambda t}$$

$\lambda > 0$ is the Lyapunov exponent.
Predictability horizon (Lyapunov time) T is defined by:

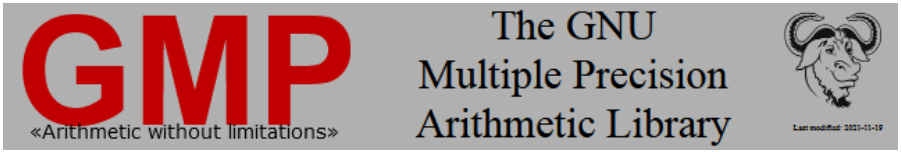
$$T = \frac{1}{\lambda}\ln(\frac{tol}{\varepsilon})$$

where $tol$ is our tolerance and $\varepsilon$ is the round-off unit (precision).

# What do we need to compute the longest term reliable trajectory?

We need:

1. A multiple-precision floating point arithmetic.




2. For efficiency we need a class of numerical methods allowing arbitrary high order of accuracy.

Breaking the limits: The Taylor series method

R. Barrio [a,*], M. Rodríguez [a], A. Abad [b], F. Blesa [c]

[a] GME, Depto. Matemática Aplicada and IUMA, Universidad de Zaragoza, E-50009 Zaragoza, Spain
[b] GME, Depto. Física Teórica and IUMA, Universidad de Zaragoza, E-50009 Zaragoza, Spain
[c] GME, Depto. Física Aplicada, Universidad de Zaragoza, E-50009 Zaragoza, Spain

3. Parallelization of the algorithm

MPI+OpenMP parallel technologies, MPIGMP library

SCIENCE CHINA
Physics, Mechanics & Astronomy

• Article •

February 2014 Vol. 57 No. 2: 330–335
doi: 10.1007/s11433-013-5375-z

# On the mathematically reliable long-term simulation of chaotic solutions of Lorenz equation in the interval [0,10000]

LIAO ShiJun[1,2,3*] & WANG PengFei[4]

[1]*State Key Laboratory of Ocean Engineering, Shanghai 200240, China;*
[2]*School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China;*
[3]*Nonlinear Analysis and Applied Mathematics Research Group (NAAM), King Abdulaziz University (KAU), Jeddah, Saudi Arabia;*
[4]*State Key Laboratory of Atmospheric Sciences and Geophysical Fluid Dynamics, Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100029, China*

Received June 27, 2013; accepted August 26, 2013; published online January 1, 2014

Using 1200 CPUs of the National Supercomputer TH-A1 and a parallel integral algorithm based on the 3500th-order Taylor expansion and the 4180-digit multiple precision data, we have done a reliable simulation of chaotic solution of Lorenz equation in a rather long interval $0 \leqslant t \leqslant 10000$ LTU (Lorenz time unit). Such a kind of mathematically reliable chaotic simulation has never been reported. It provides us a numerical benchmark for mathematically reliable long-term prediction of chaos. Besides, it also proposes a safe method for mathematically reliable simulations of chaos in a finite but long enough interval. In addition, our very fine simulations suggest that such a kind of mathematically reliable long-term prediction of chaotic solution might have no physical meanings, because the inherent physical micro-level uncertainty due to thermal fluctuation might quickly transfer into macroscopic uncertainty so that trajectories for a long enough time would be essentially uncertain in physics.

**chaos, reliable simulation, uncertainty propagation**

**PACS number(s):** 05.45.Pq, 05.40.Ca

# The model problem

We consider as a model problem the paradigmatic Lorenz system [1], which is derived from a simple model of atmospheric convection. Historically the Lorenz system is the first example of a system with a chaotic attractor.

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= Rx - y - xz \\ \frac{dz}{dt} &= xy - bz, \end{aligned} \tag{1}$$

We consider the standard Salztman's parameter values:

$R = 28$, $\sigma = 10$, $b = 8/3$.

[1] *Lorenz, Edward N. "Deterministic nonperiodic flow."Journal of the atmospheric sciences 20.2 (1963): 130-141.*

# The model problem

For these parameters the system has a chaotic attractor with Lyapunov exponent $\boldsymbol{\lambda \sim 0.906}$. The Lorenz system has three unstable fixed points. One of them is the origin $\boldsymbol{P_0 = (0, 0, 0)}$, and the other two are symmetric: $\boldsymbol{P_+}$ and $\boldsymbol{P_-}$, with coordinates:
$\boldsymbol{(\pm\sqrt{b(R-1)}, \pm\sqrt{b(R-1)}, R{-}1) \sim (\pm 8.485, \pm 8.485, 27)}$

*R. Barrio et al. / Computer Physics Communications 194 (2015) 76–83*

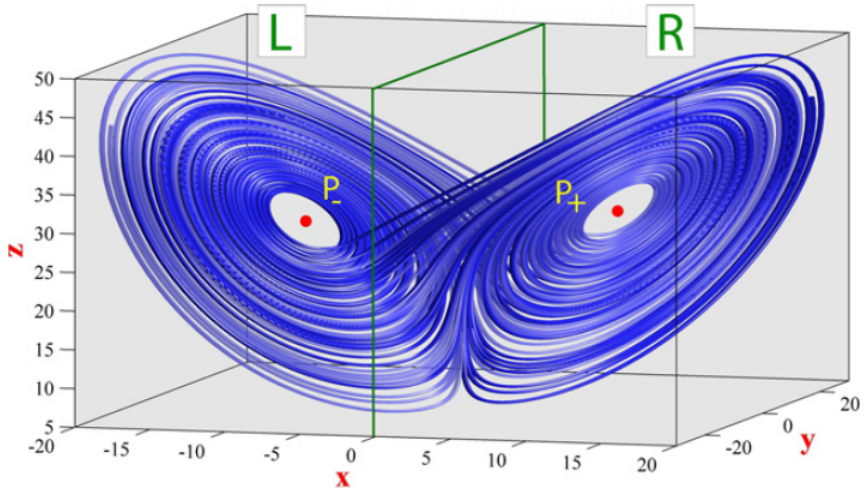



**Fig. 1.** The Lorenz attractor and the symbolic notation.

# Taylor series method

Generally we want to solve numerically the initial value problem

$$\boldsymbol{U'(t) = f(U(t)), t \in [0, T]}$$
$$\boldsymbol{U(0) = U_0}$$

with the multiple-precision Taylor series method. We assume that f is analytic on its domain of definition and that $\boldsymbol{U(t)}$ is defined in $\boldsymbol{[0, T]}$. The N-th order Taylor series method for unknown vector $\boldsymbol{U(t)}$ is given by the expression:

$$\boldsymbol{U(t+\tau) = \sum_{i=0}^{N} U^{[i]} \tau^i, \;\; U^{[i]} = \frac{1}{i!} \frac{dU^{(i)}(t)}{dt^i},}$$

where $\boldsymbol{U^{[i]}}$ are the so called normalized derivatives. We use an adaptive step-size strategy. The stepsize $\boldsymbol{\tau}$ is determined by the last two terms of the Taylor expansions:

$$\boldsymbol{\tau = \frac{e^{-0.7/(N-1)}}{e^2} \min \left\{ \left( \frac{1}{\|\mathrm{U^{[N-1]}}\|_\infty} \right)^{\frac{1}{N-1}}, \left( \frac{1}{\|\mathrm{U^{[N]}}\|_\infty} \right)^{\frac{1}{N}} \right\}}$$

## Computing the Taylor coefficients (the normalized derivatives)

From equation (1) we have

$$\begin{aligned} x_1 &= \sigma(y_0 - x_0), \\ y_1 &= Rx_0 - y_0 - x_0 z_0, \\ z_1 &= x_0 y_0 - b z_0. \end{aligned}$$

By applying Leibniz rule we obtain the following procedure for computing the normalized derivatives $x_i, y_i, z_i$ for $i = 0, ..., N-1$.

$$\begin{aligned} x_{i+1} &= \frac{1}{i+1}\sigma(y_i - x_i), \\ y_{i+1} &= \frac{1}{i+1}(Rx_i - y_i - \sum_{j=0}^{i} x_{i-j} z_j), \\ z_{i+1} &= \frac{1}{i+1}(\sum_{j=0}^{i} x_{i-j} y_j - b z_i). \end{aligned} \tag{2}$$

# Pseudocode of Taylor series method for the Lorenz system



```
while (time < T)
{
   //! Computing derivatives - O(N^2)
   for (i = 0; i<N; i++)
   {
      s1=0.0;
      s2=0.0;
      for (j=0; j<=i; j++)
      {
         s1=s1+x[i-j]*z[j];
         s2=s2+x[i-j]*y[j];
      }
      x[i+1] = Sigma*(y[i]-x[i])/(i+1);
      y[i+1] = (R*x[i]-y[i]-s1)/(i+1);
      z[i+1] = (s2-b*z[i])/(i+1);
   }
   //! One step with Horner's rule - O(N)
   h1=x[N];
   h2=y[N];
   h3=z[N];
   for (i=N-1; i>=0; i--)
   {
      h1=h1*tau+x[i];
      h2=h2*tau+y[i];
      h3=h3*tau+z[i];
   }
   x[0]=h1;
   y[0]=h2;
   z[0]=h3;

   time+=tau;
}
```

# MPIGMP library

To explain to MPI how to package and unpackage GMP multiple precision types, one needs good knowledge of both underlying representation of these types and MPI. **Tomonori Kouya** done an excellent work by creating additional libraries for MPI programs which want to use MPFR and GMP multiple precision libraries. We rely on the tiny MPIGMP library of Tomonori Kouya.

# Computational resources

The large computations for the reference solution in the time interval [0,11000] and the presented results for the performance are from **Nestum** Cluster, Sofia, Bulgaria.

**Nestum** Cluster is part of the SofiaTech park science laboratory complex, Sofia, Bulgaria. Nestum is a homogeneous HPC cluster based on two socket nodes. Each node consists of 2 x Intel(R) Xeon(R) Processor E5-2698v3 (Haswell) with 32 cores at 2.3 GHz.

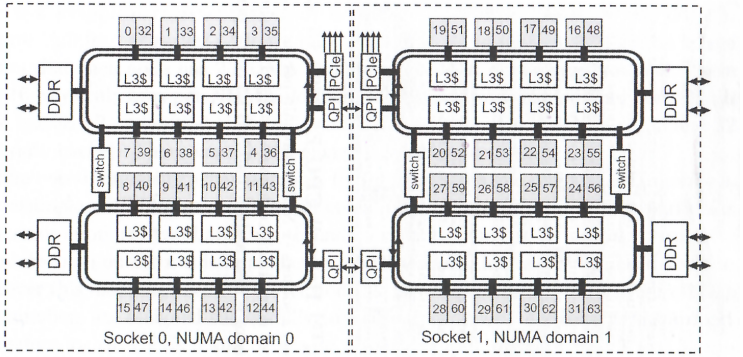
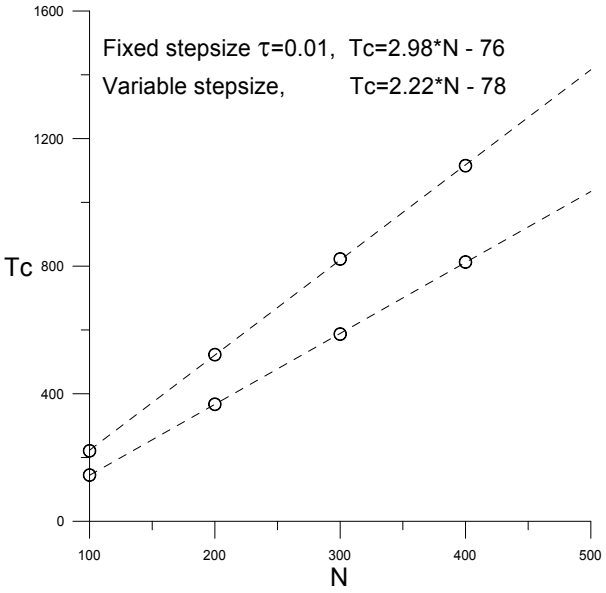



Figure 12.4: **A typical NUMA server-node with logical CPU numbers** – A server with two 16-coreIntel® Xeon™ E5-2698 v3 CPUs showing how the operating system maps logical CPU numbers onto hardware threads.

picture from: T. G. Mattson, Y. He, and A. E. Koniges. "The OpenMP common core."(2019).

# Estimation of the needed precision and order of the method




$\boldsymbol{T_c}$ is the critical predictable time - a kind of practical Lyapunov time.
$\boldsymbol{N}$ - the order of the Taylor series method
$\boldsymbol{K}$ - the number of the exact decimal digits of the precision
$\boldsymbol{T_c} = 2.55K - 81$
$\boldsymbol{\ln(10)/\lambda = \ln(10)/0.906 = 2.54148 \approx 2.55}$

Back
Close

# Results

• We computed a reference solution in the rather long time interval [0,11000]! for $\boldsymbol{x(0) = -15.8, y(0) = -17.48, z(0) = 35.64}$ (the same initial conditions as in Liao and Wang work in order to check their benchmark table). This is $\approx$ $\boldsymbol{40\%}$ larger problem then those for [0,10000]. The reference solution at $\boldsymbol{t = 11000}$ with 60 correct digits:

x= 6.106292690556899719177820030953700552671858850539708627355 08

y=-3.337953509287124281739749781445523608142105426985124626407 48

z= 34.160347153258364886745033471071226184091330735824261000528 5

• We performed two large computations with 256 CPU cores (8 nodes in Nestum). The first computation is with 4566 decimal digits of precision and 5240-th order method. The second computation is for verification - with 4778 decimal digits of precision and 5490-th order The first computation lasted $\approx$ 9 days and 18 hours and the second $\approx$ 11 days and 7 hours.

# Results

- By estimating the time needed for the same accuracy and with fixed stepsize 0.01, we conclude that by applying variable stepsize strategy we have $\mathbf{2.1x}$ speedup.

- Although the work per step in the case of variable stepsize increases by $\approx \mathbf{80\%}(\mathbf{(2.98/2.22)^2 = 1.80})$, the average stepsize is $\approx 0.034$ and thus the overall work is $\approx \mathbf{53\%}$ from the work in the case of fixed stepsize 0.01. Also the parallel efficiency increases from 55.5% up to 63.6% for the first computation and from 56.2% up to 64.3% for the second. The overall speedup with 256 cores for the first computation is 163, for the second - 165.

## Conclusions

World record for the longest term reliable trajectory for the paradigmatic Lorenz system is achieved on **Nestum** cluster. Using variable stepsize strategy not only makes the Taylor series algorithm more robust and decreases the computational work for a given accuracy, but also gives a higher parallel efficiency. Parallelized version of multiple precision Taylor series algorithm should be used with a variable stepsize strategy as a better alternative of the fixed stepsize one.

## Acknowledgments

We thank for the opportunity to use the computational resources of the **Nestum** cluster, Sofia, Bulgaria!

THANK YOU FOR YOUR ATTENTION!