## 1 Problem (2 points/answer)

Edward N. Lorenz 11 was a meteorologist working on modelling the atmosphere in the early days of computational physics when processors exhibited less computational power than today's microcontrollers. The Lorenz equation is called a chaotic system, chaos being defined as the exponential growth of initial errors in the system description, or due to different simulation conditions such as timesteps. Despite being chaotic, the system remains bounded withing a finite state space: if we are to consider $x$ as a temperature, $y$ as a pressure and $z$ as a wind speed, these quantities remain bounded within reasonable values under most simulation circumstances. Atmospheric condition ("weather") prevision is considered chaotic since the quantities are known to remain within bounded values but their actual value at a given time is difficult to assess, with an error growing exponentially over simulation time.

The simplified set of equations (Fig. 22 representing the state of the atmosphere behaviour as three variables $(x, y, z)$ is summarized [2] as

$$
\left\{\begin{array}{l}
\frac{d x}{d t}=-\sigma x+\sigma y \\
\frac{d y}{d t}=R x-y-x z \\
\frac{d z}{d t}=-B z+x y
\end{array}\right.
$$

with constants $\sigma=10, B=\frac{8}{3}$ and $R=28$. Remember that a differential equation $\frac{d x}{d t}=a$ is solved by computing the small variation $d x$ during a small duration $d t$ as $a \times d t$ and updating the variable $x$ with its evolution $x \leftarrow x+d x$.

1. In order to first become familiar with the processing method, implement in C using floating point numbers the solver for identifying the location of a ball dropped from the top of a building following the free fall equation $\frac{d^{2} x}{d t^{2}}=-g$ with $g=10 \mathrm{~m} \cdot \mathrm{~s}^{-2}$ the constant acceleration of gravity, or in other words $v=\frac{d x}{d t}$ and $\frac{d v}{d t}=-g$ with $v$ the ball velocity and $x$ its position. What is the position of the ball, initially static before being dropped, after a 10 s fall computed with a $d t=10^{-3}$ s step?
2. How many iterations did you need to reach that result? When printing the solution of the calculation, also provide the theoretical result $x=\frac{1}{2} g t^{2}$ : the.$^{2}$ function is provided by the pow() function of the mathematical library: how do you compile the program to link with this library providing the squaring function?
3. Repeat the calculation with $d t=10^{-2}$ s: how does the result compare with the previous conclusion?
4. Repeat the calculation with $d t=10^{-1} \mathrm{~s}$ : how does the result compare with the previous result? Implement the same calculation in fixed point number representation and compare with this last floating point result: how do they compare?

Now that we are convinced we can numerically solve a differential equation, we wish to repeat Lorenz's calculation and solve his set of differential equations to identify $(x, y, z)$ as a function of time. We first consider the floating point implementation to benefit from modern processor features, before considering the fixed point implementation best suited for embedded systems not fitted with a hardware floating point unit.
5. Using floating point numbers in C, solve the first 50 seconds of Lorenz's set of equations for a time step $d t=10^{-3} \mathrm{~s}$ and initial conditions $(x, y, z)=(0.1,0 ., 0$.$) . Plot the evolution of x$ and $z$ as a function of time.
6. Repeat with $d t=10^{-2}$ s: how does the result compare with the previous computation?
7. Repeat with $d t=10^{-1} \mathrm{~s}$ : how does the result compare with the previous computation?
8. Implement the Lorenz equation solver as a fixed point representation keeping three relevant decimals. How does the result compare with the simulation of question 4 ?

James Gleick [3] tells the story of E. Lorenz's discovery as
"One day in the winter of 1961, wanting to examine one sequence at greater length, Lorenz took a shortcut. Instead of starting the whole run over, he started midway through. To give the machine its initial conditions, he typed the numbers straight from
the earlier printout. Then he walked down the hall to get away from the noise and drink a cup of coffee. When he returned an hour later, he saw something unexpected, something that planted a seed for a new science.

This new run should have exactly duplicated the old. Lorenz had copied the numbers into the machine himself. The program had not changed. Yet as he stared at the new printout, Lorenz saw his weather diverging so rapidly from the pattern of the last run that, within just a few months, all resemblance had disappeared. He looked at one set of numbers, then back at the other. He might as well have chosen two random weathers out of a hat. His first thought was that another vacuum tube had gone bad.

Suddenly he realized the truth. There had been no malfunction. The problem lay in the numbers he had typed. In the computer's memory, six decimal places were stored: .506127. On the printout, to save space, just three appeared: .506. Lorenz had entered the shorter, rounded-off numbers, assuming that the difference - one part in a thousand - was inconsequential.
[...]
... in Lorenz's particular system of equations, small errors proved catastrophic "
9. The stability of Lorenz's attractor - the set of values the variables $x, y$ and $z$ might reach whatever the initial conditions and initial errors, is best visualized by plotting one variable against the other instead of their time dependence: solve the Lorenz equation usin fixed point numbers keeping two decimals and plot $z$ against $x$. How does the chart compare with the same plot using the solutions found in the previous question?
merical integration. In certain cases all except three of small superposed perturbations $x_{0}(\tau), y_{0}(\tau), z_{0}(\tau)$. Such the dependent variables eventually tended to zero, and these three variables underwent irregular, apparently nonperiodic fluctuations.
These same solutions would have been obtained if the eries had at the start been truncated to include a total three terms. Accordingly, in this study we shall let $a\left(1+a^{2}\right)^{-1} \kappa^{-1} \psi=X \sqrt{2} \sin \left(\pi a H^{-1} x\right) \sin \left(\pi H^{-1} z\right), \quad$ (23) $\pi R_{e}^{-1} R_{a} \Delta T^{-1} \theta=Y \sqrt{2} \cos \left(\pi a H^{-1} x\right) \sin \left(\pi H^{-\frac{1}{z}}\right)$
$-Z \sin \left(2 \pi H^{-1} 2\right), \quad$ (24)
where $X, Y$, and $Z$ are functions of time alone When pressions (23) and (24) are substituted into (17) and 18), and trigonometric terms other than those occurring in (23) and (24) are omitted, we obtain the equations

| $X$ | $=-\sigma X+\sigma Y$, |
| ---: | :--- |
| $Y$ | $=-X Z+r X-Y$, |
| $Z$ | $=X Y$ |

Here a dot denotes a derivative with respect to the dimensionless time $\tau=\pi^{2} H^{-2}\left(1+a^{2}\right) \kappa t$, while $\sigma=\kappa^{-1} \nu$ is he Prandll number, $r=R_{c}^{-1} R_{a}$, and $b=4\left(1+a^{2}\right)^{-1}$.
ized equations

(29)

Since the coefficients in (29) vary with time, unless the basic state $X, Y, Z$ is a steady-state solution of (25)-(27), a general solution of (29) is not feasible. However, the variation of the volume $V_{0}$ of a small displaced in accordance with (25)-(27), is region is by the diagonal sum of the matrix , is determined specifically

## $V_{0}=-(\sigma+b+1) V_{0}$

This is perhaps most readily seen by visualizing the motion in phase space as the flow of a fluid, whose divergence is

$$
\frac{\partial X^{*}}{\partial X}+\frac{\partial Y^{*}}{\partial Y}+\frac{\partial Z}{\partial Z}=-(\sigma+b+1) .
$$

## 7. Numerical integration of the convection equations

To obtain numerical solutions of the convection equations, we must choose numerical values for the constants. Following Saltzman (1962), we shall let $\sigma=10$ and $a^{2}=\frac{1}{2}$, so that $b=8 / 3$. The critical Rayleigh number for instability of steady convection then occurs when $r=470 / 19=24.74$.
We shall choose the slightly supercritical value $r=28$. The states of steady convection are then represented by the points $(6 \sqrt{2}, 6 \sqrt{2}, 27)$ and $(-6 \sqrt{2},-6 \sqrt{2}, 27)$ in phase space, while the state of no convection corresponds to the origin. (0.non).

Figure 2: Excerpts from the original manuscript by E.N. Lorenz introducing (left) the non-linear set of coupled equations as well as the linearization only valid locally (Eq. 29), and (right) the parameters leading to chaotic behaviour.

## 2 Questions (1 point/answer)

10. A program running on a STM32 displays messages over the 9600 baud asynchronous serial port in 8 N 1 format: the C software for communicating is printf ( $" \% 3.3 f \backslash r \backslash n ", M \_P I$ ) for displaying $\pi$ with a maximum of 3 digits in the integer part and 3 digits in the fractional part. How long does the communication last? Justifiy (the constant M_PI is provided by the mathematical library in math.h).
11. Compiling an executable output.elf requires three C-language source codes a.c, b.c and c.c. Provide the Makefile for generating output.elf that will only recompile a single source code when modified in the text editor in order to regenerate the executable, and will avoid re-compiling the objects resulting from the untouched source files.
12. The following screenshot was captured on an SPI bus configured to program a 28 -bit Direct Digtal Synthesizer (DDS) AD9834 (top chart) clocked at a reference frequency of 70 MHz


Identify which color amongst the three bottom curves matches which signal amongst CS\#, MOSI and CLOCK. What is the SPI configuration set for this communication (CPHA, CPOL or MODE)? Justify.
13. On the previous chart, what was the word sent to the DDS over the SPI bus? Justify.
14. Considering that the Frequency Tuning Word is transmitted as two 14 -bit words separated by two most-significant-bits indicating which register is being programmed, does the output frequency of the DDS (top signal and curson) match your expectaction? Justify.
15. Considering bit indexes start at 0 , select bits 6 to 14 of a 16 -bit word and set bit 12 of the initial integer to 1 .
16. How was the integer of the previous quesiton declared in the C language?

Negative grades if copies from Google/web search irrelevant to the questions are provided as answers.

## References

[1] E.N. Lorenz, Deterministic nonperiodic flow, J. of Atmospheric Sciences 20(2) 130-141 (1963) at https://journals. ametsoc.org/view/journals/atsc/20/2/1520-0469_1963_020_0130_dnf_2_0_co_2.xml
[2] H.O. Peitgen, H Jürgens, D. Saupe, \& M.J. Feigenbaum, Chaos and fractals: new frontiers of science, Springer New York (1992)
[3] J. Gleick, Chaos: Making a New Science, Random House UK (1997)

## Answers

1. A deterministic equation whose output only depends through polynomial laws on constants and time leads to stable solutions, in this case the free fall of a ball always converges to the $500 \pm 5 \mathrm{~m}$ solution irrelevant on the time step $d t$. Such behaviour led Descartes in the 17th century to consider nature as deterministic and its evolution to be predictable assuming initial conditions to be known accurately enough
2. using the pow) () function involves the mathematical library libm linked to the executable by gcc with the -lm flag. The floating point implementation is
\#include<stdio.h>
\#include<math.h> // pow() requires compiling with -lm
int main()
\{float $\mathrm{g}=-10 ., \mathrm{v}=0 ., \mathrm{x}=0$.;
float $\mathrm{dt}=0.001, \mathrm{t}, \mathrm{dx}, \mathrm{dv} ; / /$ change $t$ to 0.01 or 0.1
int n;
for ( $\mathrm{t}=0 . ; \mathrm{t}<=10 . ; \mathrm{t}+=\mathrm{dt}$ )
$\{\mathrm{dv}=\mathrm{g} * \mathrm{dt} ; \mathrm{dx}=\mathrm{v} * \mathrm{dt}$;
$\mathrm{v}=\mathrm{v}+\mathrm{dv}$; $\mathrm{x}=\mathrm{x}+\mathrm{dx}$;
$\mathrm{n}++$;
\}
printf("\%d: \%f \%f $\left.{ }^{n} ", n, x, 0.5 * g * \operatorname{pow}(t, 2)\right)$;
\}
3. all results consistently converge to 500 m whatever the timestep, with better convergence with smaller timesteps but more computational power needed to reach the solution.
4. The fixed point implementation of the same algorithm replaces the arithmetic operations "+" and most importantly "*" with addfix() and mulfix() described below, and scaling all parameters including the acceleration $g$ with the scaling factor indicating the location of the decimal in the integer representation.
5. Lorenz equation is solved with
\#include <stdio.h>
int main()
\{double $x=0.1, y=0 ., z=0 .$, sigma $, R, B, d x, d y, d z ;$
double dt=.001, t;
int k;
sigma $=10$.;
$\mathrm{B}=8 . / 3$.;
$\mathrm{R}=470$./19.;
for $\quad(\mathrm{t}=0 ; \mathrm{t}<50 . ; \mathrm{t}+=\mathrm{dt})$

$$
\{d x=\operatorname{sigma} *(y-x) * d t ;
$$

$\mathrm{dy}=(\mathrm{x} *(\mathrm{R}-\mathrm{z})-\mathrm{y}) * \mathrm{dt}$;
$\mathrm{dz}=(\mathrm{x} * \mathrm{y}-\mathrm{B} * \mathrm{z}) * \mathrm{dt}$;
$\mathrm{x}=\mathrm{x}+\mathrm{dx}$;
$\mathrm{y}=\mathrm{y}+\mathrm{dy}$;
$\mathrm{z}=\mathrm{z}+\mathrm{dz}$;
printf("\%f \%f \%f $\backslash n ", x, y, z) ;$
\}
\}
6. The $d t=10^{-2}$ s leads to a completely different behaviour of the variables as a function of time when computed with $d t=10^{-3} \mathrm{~s}$ as shown in the figure below, even though their value remains constrained withing the same boundaries:



7. With too large a timestep, e.g. $d t=0.1 \mathrm{~s}$, the solution diverges to infinity and can no longer be represented as a floating point number, hence the display of a solution as NaN or Not a Number.
8. Using the fixed point library
\#include "fixed.h"

```
long addfix(long in1,long in2) {return(in1+in2);}
long mulfix(long in1,long in2)
{long long tmp;
    tmp=(long long)in1*(long long)in2;
    tmp/=SCALE;
    return((int)tmp);
}
long divfix(long in1, long in2)
{long long tmp=(long long)in 1 *SCALE;
    if (in2!=0)
        tmp/=(long long)in2;
    else tmp=0;
    return((long)tmp);
}
```

we implement the solution to the Lorenz equation as fixed point calculation
\#include <stdio.h>
\#include "fixed.h"

```
int main()
{int x=(int)(0.1*SCALE), y=0,z=0, sigma,R,B,dx,dy,dz;
    int dt=(int)(.01*SCALE), t;
    int k;
    sigma=(10*SCALE);
    B}=(\mathrm{ int) ((8./3.) *SCALE);
    R=(int)(470./19.*SCALE);
    printf("%d %d %d\n",x,dt,B);
    for ( }\textrm{t}=0;\textrm{t}<(50*\mathrm{ SCALE); t+=dt)
        {dx=mulfix(sigma, mulfix(addfix (y,-x),dt));
            dy=mulfix(addfix(mulfix(x, addfix(R,-z)),-y),dt);
            dz=mulfix(addfix(mulfix (x,y), -mulfix (B, z)),dt);
            x=addfix (x,dx);
            y=addfix(y,dy);
            z=addfix(z,dz);
            printf("%d %d %d\n",x,y,z);
        }
}
```

9. By plotting one variable as a function of the other (here $z$ as a function of $x$ ), similar initial conditions lead to divergent behaviour still contrainted within bounded values of each variable along a structure known as a strange attractor in the phase space $(x, y, z)$ :

10. 9600 bauds at 8 N 1 is $10.4 \mathrm{~ms} /$ symbol and the string represents $\pi$ with three decimals or 5 digits including the decimal separator followed by two symbols for the line feed and carriage return so a total of 7 symbols or 72.9 ms .
11. The makefile is
all: output.elf
output.elf: a.o b.o c.o
gcc -o output.elf a.o b.o c.o
a.o: a.c
gcc -c a.c
b.o: b.c
gcc -c b.c
c.o: c.c gcc -c c.c
12. green is chip select, blue is the periodic clock, red is the data. The rest state of the clock is high so CPOL=1, and the first clock transition exhibits the stable state of data so $\mathrm{CPHA}=0$.
13. The word is $0 \times 40404040$ but the endianness could have been opposite.
14. After removing the two pairs of unwanted bits, the freqency tuning word becomes the concatenation of $0 x 0040$ (12 least significant bits) and 0x010 ( 12 most signifcant bits) or 0x0100040 which is converted by the DDS to 0x0100040/2^28*70e6 or 273450 Hz , matching the frequency displayed on the oscilloscope screenshot.
15. short $s ; s \mid=(1 \ll 12) ; s=(s \gg 6) \& 0 x 0 x 1 f f$
16. see above
