1.– We want to evaluate with an absolute error of less than $1, cm^2$ the area of a square enclosure. To do this, one side of the enclosure is measured and the area is calculated by squaring it. It is known "a priori" that the enclosure is approximately $1\, m^2$. How accurate should the ruler used in the measurement be? Would it be possible to perform this calculation correctly on a computer using simple precision?

2.– We want to draw on a plotter the curve $y = f(x)$ for values of x included in the interval $[a, b]$. For this purpose the following subroutine is created:

```
SUBROUTINE CURVA(A,B,N)
IMPLICIT REAL*4 (A-H,O-Z)
IMPLICIT INTEGER*4 (I-N)
DELTA=(B-A)/FLOAT(N)
X=A
Y=F(A)
CALL MOVE(X,Y)
DO I=1,N
   X=A+FLOAT(I)*DELTA
   Y=F(X)
   CALL DRAW(X,Y)
ENDDO
RETURN
END
```

where subroutine `MOVE(X,Y)` moves the pen without drawing to the (x, y) coordinate point, subroutine `DRAW(X,Y)` moves the pen by drawing a straight line from the previous position to the (x, y) point, and `F(X)` is a function of type `REAL*4 FUNCTION` that is written separately. According to the FORTRAN compiler manual, for variables of type `REAL*4`, $m = 24$ bits are allocated for storing the mantissa (sign included).

a) Explain very briefly how the subroutine works..

b) Reasonably find —in first approximation— a maximum value of N beyond which the results are not improved, because the precision of the computer does not allow discriminating between two consecutive values of x.

c) Reasonably find —in first approximation— a maximum value of N beyond which the results are not improved, because the precision of the computer does not allow discriminating between two consecutive values of $f(x)$.

d) Considering also that the plotter interprets the (x, y) coordinates in centimeters, and that the maximum resolution of the pen is 0.1 millimeters, what is the maximum value of N to be used in practice?

NOTAE: Simplifications deemed reasonable may be made, since we speak of "first approximation", as long as they are justified..

3.– Present several pathological examples of numerical operations in which it is demonstrated that in a digital computer:

- **a)** "Similar" numbers must not be subtracted.
- **b)** There must be no division by "small" numbers.
- **c)** It is preferable to add the smallest numbers first.
- **d)** The order of the factors alters the product.

Indicate what is the theoretical justification of these assertions. Show the examples in decimal system using a calculator.

4.– In a particular situation it is necessary to evaluate the following function:

$$f(x) = 1 - (1 - x)(1 + x) = x^2$$

for values of x such that $|x| << 1$.

The computer used uses m bits for the storage of the mantissa (including the sign) in floating point, and that rounds by approximation.

- **a)** Should $f(x)$ be calculated as $1-(1-x)(1+x)$ or as x^2? (In the latter case the calculation is done by multiplying the variable x by itself).
- **b)** Establish the relative error bounds in the two cases.
- **c)** Which operation is best in terms of the values of x ?
- **d)** Present an example numerical of the convenience of using one or the other method when calculations are performed manually with the help of three significant decimal digits.

5.– The n first powers $\varphi^1, \varphi^2, \ldots, \varphi^n$ of the golden ratio $\varphi = (-1+\sqrt{5})/2$ can be obtained without performing any multiplication, since the following relation is fulfilled:

$$\varphi^i = \varphi^{i-2} - \varphi^{i-1}.$$

Show that this way of performing calculations is numerically unstable, while repeated multiplication is numerically stable.

6.– We want to approximate e^x for various values of x in the interval $[0, 10]$ by summing the first n terms of the Taylor series. The calculation will be performed on a computer using 24 bits for the floating-point mantissa (including the sign). Carry out a simple study (without taking into account the effect of the propagation of the rounding error) that establishes a maximum value of the number of terms to be considered, above which no appreciable improvement in the approximation can be seen.

7.– In a calculus process it is required to repeatedly evaluate the function $f(x) = \sqrt{1+x} - 1$. It is known that for small values of x , there exists, among others, the asymptotic approximation to the above function $f(x) \approx f_0(x) = x/2$. The calculations will be performed on a digital computer. Could the results obtained using the asymptotic approximation f_0 be better than the results of the calculations obtained using the function $f(x)$ itself? If yes: For what range of values of x?; what would this range be if simple precision (24 bits for the mantissa, including the sign) is used? In the affirmative or negative case, present several numerical examples that corroborate the result, using the decimal base and performing floating point operations with three digits.

8.– In a computational process it is necessary to evaluate the function $f(x) = e^{-1/x}$ for various values of $x \in [0.05, 0.10]$ with a relative error $r_f \leq 10^{-8}$. If the values of x are known with relative error $r_x \leq 10^{-5}$, and assuming we can increase the computer's accuracy as far as necessary, can a satisfactory result be obtained? If not, with what precision would it be necessary to know the data?

In either case, what type of precision—single or double—should be used if the calculations are performed on a digital computer? Assume that in single and double precision 24 and 53 bits, respectively, are used to store the mantissa (including the sign)..

9.– In a computational process it is necessary to evaluate several million times the function $f(x) = \ln(1 + x)$, always for positive ($x > 0$) and small ($x << 1$) values of x. The FORTRAN compiler function available is very accurate, but requires a relatively high computation time (on the order of the equivalent to several tens of elementary operations).

To reduce the computational time, the possibility of approximating the function using the first two terms of its Taylor development is evaluated, this simplification being admitted when the relative error is less (in absolute value) than one hundred times the machine storage error. The following subroutines are prepared:

```
SUBROUTINE EXACT(X,Y1)
REAL*4 X,Z,Y1
Z=1.+X
Y1=ALOG(Z)
RETURN
END
```

```
SUBROUTINE APPROX(X,Y2)
REAL*4 X,T,Y2
T=1.-X/2.
Y2=X*T
RETURN
END
```

Knowing that the computer works in floating point rounding by approximation and allocating $m = 24$ bits to the mantissa (including 1 bit for the sign), and that the values of `X` are exact (in the sense that their inherent error is zero, although in general they will be affected by the corresponding storage error):

- **a)** Analyze instruction by instruction the previous subroutines, obtaining the relative errors of the variables at each step.
- **b)** Obtain the relative error of the value `Y1` calculated by the subroutine `EXACT` with respect to the exact value of $f(x)$. Obtain a bound of the error and explain what happens when x tends to zero.
- **c)** Obtain the relative error of the value `Y2` calculated by the subroutine `APPROX` with respect to the exact value of $f(x)$. Obtain a bound of the error and explain what happens when x tends to zero.
- **d)** Discuss for which range of values of `X` the subroutine `APPROX` can be used instead of the subroutine `EXACT`.
- **e)** Is it possible that the `APPROX` subroutine provides more accurate results than the `EXACT` subroutine? If so, for what values of `X`?

Note: The Taylor expansion (with the Lagrangian remainder) of the function $f(x)$ is:

$$f(x) = -\sum_{i=1}^{n}(-1)^i \frac{x^i}{i} + R_n(x); \qquad R_n(x) = -\left(\frac{-1}{1+\xi}\right)^{n+1} \frac{x^{n+1}}{n+1}; \qquad \xi \in (0, x).$$

10.– A FORTRAN program includes the instruction `X=`n, where `X` is a real variable of type `REAL*4` (single precision) or of type `REAL*8` (double precision), and n is a positive integer constant of type `INTEGER*4`.

Determine what is the value of n at which storage errors can occur in single precision and double precision.

11.– Develop a FORTRAN program that finds the solution to the above problem by numerical experimentation.

12.– In a digital computer the following sum is obtained

$$S_n = \overbrace{a + a + \ldots + a}^{\text{“}n\text{” times}},$$

where a is any number whose inherent error is r_a^I. The operations are performed in floating point, allocating m bits to the mantissa (including the sign):

- **a)** Calculate the total error of the result obtained by performing the above operation. Indicate what part of the total error is due to the propagation of the inherent error of the data a and what part is due to the propagation of the storage errors of the intermediate operations.
- **b)** Analyze how the maximum error grows as the value of n increases. Discuss to what extent this maximum value can be considered exaggerated.
- **c)** Repeat the previous sections assuming that the calculation is performed in the form $S_n = n \cdot a$. Compare the results obtained in the two cases.
- **d)** Relate these theoretical conclusions with the set-up of problem 8 of the previous problems sheet and make a theoretical prediction of the PK from which the problems referred to in that exercise can be produced. If necessary, compare the theoretical prediction with the "experimental" results obtained by numerical simulation.

13.– An engineer usually programs in FORTRAN and uses a certain digital computer. The engineer wants to know what is the machine error r_M with which the calculations are performed, both for variables of type `REAL*4` (single precision) and for variables of type `REAL*8` (double precision).

The engineer does not have access to manuals or any documentation explaining how the data is stored and how many bits are allocated to mantissa storage in each case.

- **a)** Relate the machine error r_M with the value of the Machine Epsilon $\varepsilon > 0$, corresponding to the smallest positive real number such that $\mathcal{A}(1+\varepsilon) \neq \mathcal{A}(1)$, with $\mathcal{A}(x)$ being the stored value corresponding to x.
- **b)** To realize a FORTRAN program that allows to obtain experimentally the value of the Machine Epsilon ε for the two types of precision (single and double) on the computer on which the program is run.
- **c)** Test the program on a computer and compare the results with theoretical predictions. with the theoretical predictions. Explain the discrepancies, if any.
