Chapter 1: Introduction to Fortran Programming

1. Introduction

The **Fortran (Formula Translation)** language was created in **1957** by **IBM**[1], under the direction of **John Backus**,

to facilitate scientific and technical computations. It was one of the first high-level programming languages and remains a key tool in numerical computing today.

Fortran is particularly valued for its **performance in scientific computing**, its **optimized handling of arrays and matrices**, and its **efficiency in complex simulations**. Unlike modern languages such as **Python, C++, or Java**, which are more versatile, Fortran stands out for its **speed and efficiency in intensive numerical calculations**, making it widely used in **physics, meteorology, engineering, and numerical modeling**.

Over the years, Fortran has evolved to meet modern needs, with versions like **Fortran 90, Fortran 95, Fortran 2003, Fortran 2008, and Fortran 2018**, each introducing improvements in **modular programming, dynamic array handling, and parallel computing compatibility**.

1.2 Fortran compiler

To use Fortran, you need to install a **Fortran compiler**. Several options are available:

1.2.1 GNU Fortran (gfortran):

- Open-source compiler included in GCC.
- Compatible with Linux, Windows, and macOS.
- Installation:
  - **On Linux**: sudo apt install gfortran
  - **On Windows**: Using **MinGW** or **Cygwin**.

[1] IBM stands for International Business Machines Corporation. It is an American company founded in 1911, specializing in computers, software, technology services, and hardware.

1.2.2 Intel Fortran Compiler (ifort):

- High-performance compiler for scientific computing.
- Available for free in the **Intel oneAPI Toolkit**.

1.2.3 Silverfrost FTN95:

- A Windows-based compiler with strong debugging support.

1.2.4 LFortran:

- A modern, interactive compiler with Jupyter Notebook compatibility.

1.3 Environment and interface of Fortran

Fortran can be used with different **Integrated Development Environments (IDEs)** to facilitate coding and compilation:

a) **Code::Blocks**: Supports gfortran, simple and efficient.

b) **Visual Studio Code + Fortran Extension**: Lightweight and customizable.

c) **Eclipse + Photran Plugin**: Advanced IDE with full features.

d) **Intel Fortran Compiler + Visual Studio**: Recommended for intensive calculations.

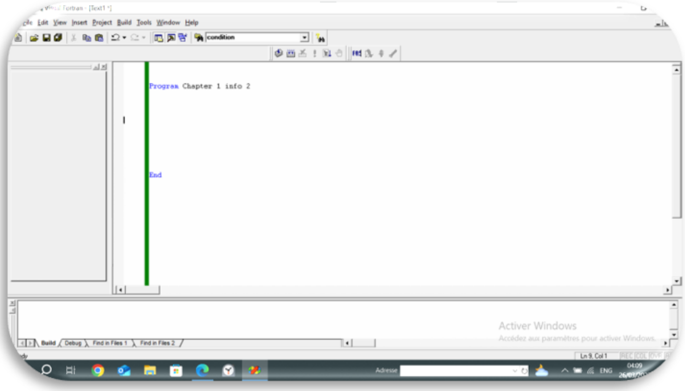
1.4 Fortran file : .for et .f90

Fortran uses different types of source files:

(a) .for **files:**

- Used in older versions (Fortran 77 and earlier).
- Code is written in **fixed-column format**.

Example:



Visual Fortran - [Text1 *]
File Edit View Insert Project Build Tools Window Help
condition
Build
Debug
Find in Files 1
Find in Files 2
Activer Windows
Accédez aux paramètres pour activer Windows.
Ln 9, Col 1
Adresse
ENG
04:09

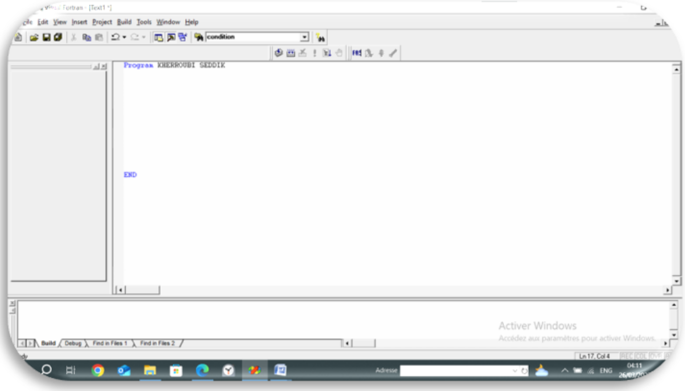
```
      Program Chapter 1 info 2



      End
```

(b).f90 **files**:

- Introduced with **Fortran 90**.
- Code is written in **free-form format** (no column restrictions).

- Supports modular programming and dynamic arrays.

Example:



```
Program KHERROUBI SEDDIK

END
```