Fortran vs. Python Performance: Modeling EDC Cracking

Rodrigo Silva Ferreira Dr. Helianildes Silva Ferreira

Introduction

This report compares the performance of a Fortran-based ODE solver with a pure Python implementation for modeling the thermal cracking of Ethylene Dichloride (EDC). The system of ordinary differential equations is solved using the Runge-Kutta-Gill method.

Reaction System

The cracking of EDC involves two sequential reactions:

$$\text{EDC} \xrightarrow{k_1} \text{MVC} + \text{HCl}, \quad \text{MVC} \xrightarrow{k_2} \text{HCl}$$

Expressed as:

$$\begin{aligned}
\frac{dy_1}{dz} &= -k_1 y_1 \\
\frac{dy_2}{dz} &= k_1 y_1 - k_2 y_2 \\
\frac{dy_3}{dz} &= k_1 y_1 + k_2 y_2
\end{aligned}$$

With initial condition:

$$y(0) = [1.0,\ 0.0,\ 0.0]$$

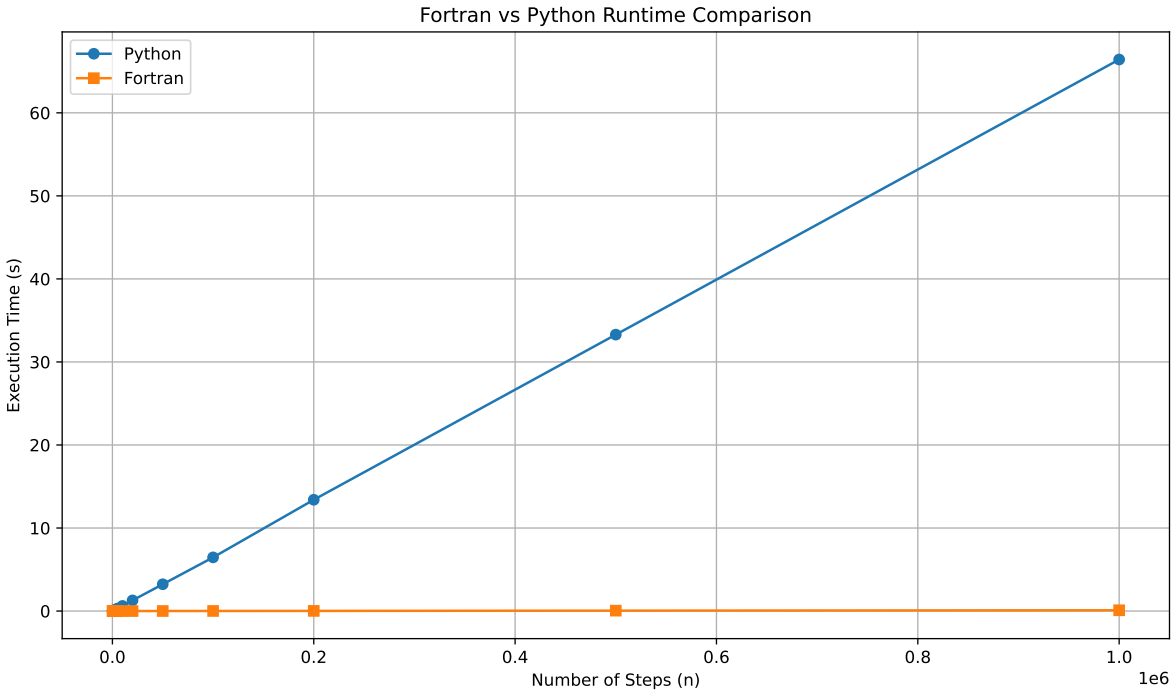
Methodology

- Fortran subroutine `rk4g_solver` compiled with `f2py`
- Python version implemented using NumPy
- Performance measured using `timeit`
- Results compared across multiple values of `n`

Results Table

	n	Fortran (s)	Python (s)	Speedup (x)
0	100	0.0000	0.0060	230.84
1	500	0.0000	0.0300	613.93
2	1000	0.0001	0.0600	570.25
3	2000	0.0002	0.1187	660.38
4	5000	0.0004	0.3021	680.40
5	10000	0.0010	0.6185	638.93
6	20000	0.0018	1.2838	722.32
7	50000	0.0045	3.2288	723.16
8	100000	0.0097	6.4652	669.15
9	200000	0.0193	13.4027	695.66
10	500000	0.0488	33.2895	682.82
11	1000000	0.0948	66.4226	700.39

Performance Plot



Conclusion

We observed significant speedups (up to 700x) when using the Fortran implementation over Python, particularly for large-scale simulations. This reinforces the importance of choosing the right tool for the problem at hand.