

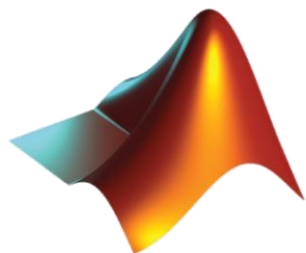
# Metodi del Calcolo Scientifico

Raffaele Cerizza 845512  
Giacomo Savazzi 845372  
Andrea Assirelli 820149

*22 Luglio 2022*

# Primo Progetto – Algebra Lineare Numerica

*Sistemi Lineari con Matrici Sparse e Definite Positive*



Linguaggio	Utilizzo	Istallazione	Documentazione	Manutenzione
Matlab	++	+	+	+
C++	+	-	-	+
R	+	+	+	+



*Microsoft Windows 10 Pro*



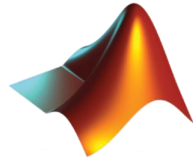
*Ubuntu 20.04.3 LTS*

*CPU: Intel(R) Xeon(R) Gold 6130 CPU*

*Physical Memory (RAM): 8.00GB*

*Disk: 154GB SSD*

*Architecture: x64*



```
% Inizio del calcolo del tempo di risoluzione del sistema lineare.  
tic;  
  
% Risoluzione del sistema lineare con metodo di Cholesky.  
x = A\b;  
  
% Termine del calcolo del tempo di risoluzione del sistema lineare.  
time = toc;  
  
% Disabilitazione di diary.  
diary off;  
  
% Calcolo dell'errore relativo (come norma Euclidea) fra la soluzione  
% calcolata e la soluzione esatta.  
error = norm(x-xe,2)/norm(xe,2);
```

- *Calcolo Velocità di Risoluzione*
- *Calcolo Errore Relativo*

- *Calcolo Memoria in Uso*

```
1 % Questa funzione usa il comando WHOS per individuare gli elementi  
2 % all'interno del BASE workspace. Dopodiché somma i byte di ciascun  
3 % elemento e restituisce la somma in MB.  
4 function [memory_in_use] = monitor_memory()  
5 elements_in_memory = evalin('base','whos');  
6 if size(elements_in_memory,1) > 0  
7  
8     ....for i = 1:size(elements_in_memory,1)  
9         ....array_of_elements(i) = elements_in_memory(i).bytes;  
10    ....end  
11    ....memory_in_use = sum(array_of_elements);  
12    ....memory_in_use = memory_in_use/1048576;  
13  
14 else  
15     ....memory_in_use = 0;  
16 end
```

```
47 print("-----")
48 ## Esecuzione della decomposizione di Cholesky pivot MMD
49 i = Sys.time()
50 R = trycatch(
51 {
52   chol.spam(matrix, pivot = "MMD")
53 },
54 error = function(e){
55   ## Errore sollevato se la matrice non è definitiva positiva. Se non fosse
56   ## stata simmetrica ci saremmo fermati prima
57   return (e)
58 }
59 )
60 chol_time = difftime(Sys.time(), i, units = "secs")
61 print(chol_time, digits = 4L)
62 print(paste("-> ", name, " - chol time pivot MMD: ", chol_time, sep = ""))
63 chol_size = object.size(R)
64 print(chol_size, units = "Mb", digits = 4L)
65 print(paste("-> ", name, " - chol size pivot MMD: ", chol_size, sep = ""))
66 ## Risoluzione sistema lineare pivot MMD
67 i = Sys.time()
68 x = solve.spam(R,b)
69 solve_time = difftime(Sys.time(), i, units = "secs")
70 print(solve_time, digits = 4L)
71 print(paste("-> ", name, " - solve time pivot MMD: ", solve_time, sep = ""))
72 print(paste(chol_time+solve_time, digits = 4L))
73 print(paste("-> ", name, " - tot time pivot MMD: ", solve_time+chol_time, sep = ""))
74 ## calcolo errore relativo
75 rel_error = norm(xe - x, type = "2") / norm(xe, type = "2")
76 print(rel_error, digits = 4L)
77 print(paste("-> ", name, " - relative error pivot MMD: ", rel_error, sep = ""))
```

- *Calcolo Velocità di Risoluzione*
- *Calcolo Errore Relativo*
- *Calcolo Memoria Utilizzata*



```
// Preparazione fattorizzazione matrice
solver.analyzePattern(resultMatrix);

// Fattorizzazione matrice
solver.factorize(resultMatrix);

// Memoria Finale Matrice
#ifdef _MSC_VER
GetProcessMemoryInfo(GetCurrentProcess(), &pmc, sizeof(pmc));
double finalMemory = pmc.WorkingSetSize / (1024.0 * 1024.0);
#elif defined(__GNUC__)
double finalMemory = getValue() / 1024;
#endif

// Calcolo memoria utilizzata Matrice fattorizzata
MatrixMemoryUsage = finalMemory - initialMemory;

// Calcolo soluzione
VectorXd x = solver.solve(b);

// Fine timer Cholesky
auto end = std::chrono::high_resolution_clock::now();
std::chrono::duration<double> duration = end - start;

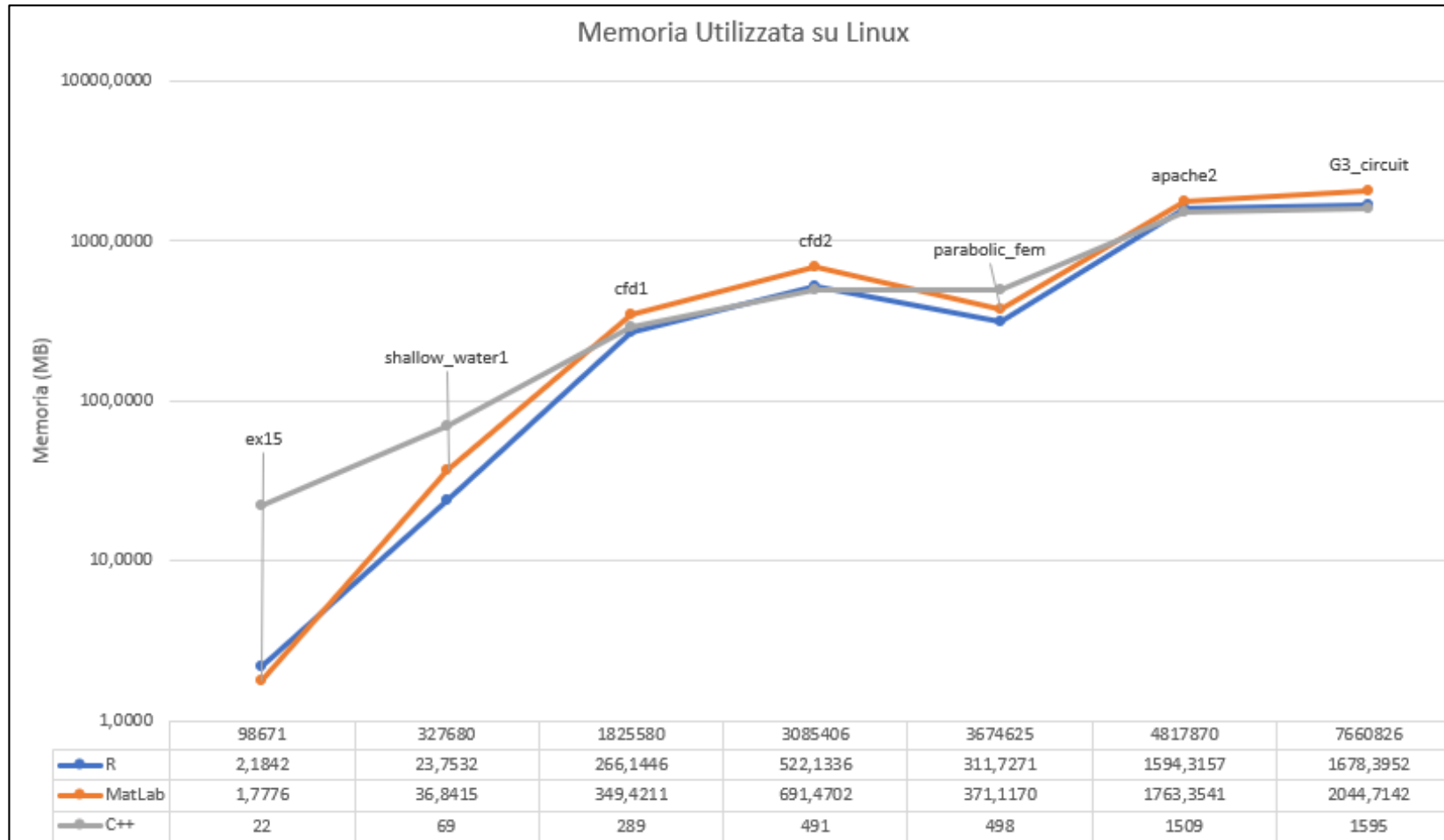
// Tempo esecuzione cholesky in secondi
ElapsedTime = duration.count();

// Calcolo errore relativo
RelativeError = (x - xe).norm() / xe.norm();

return true;
}
```

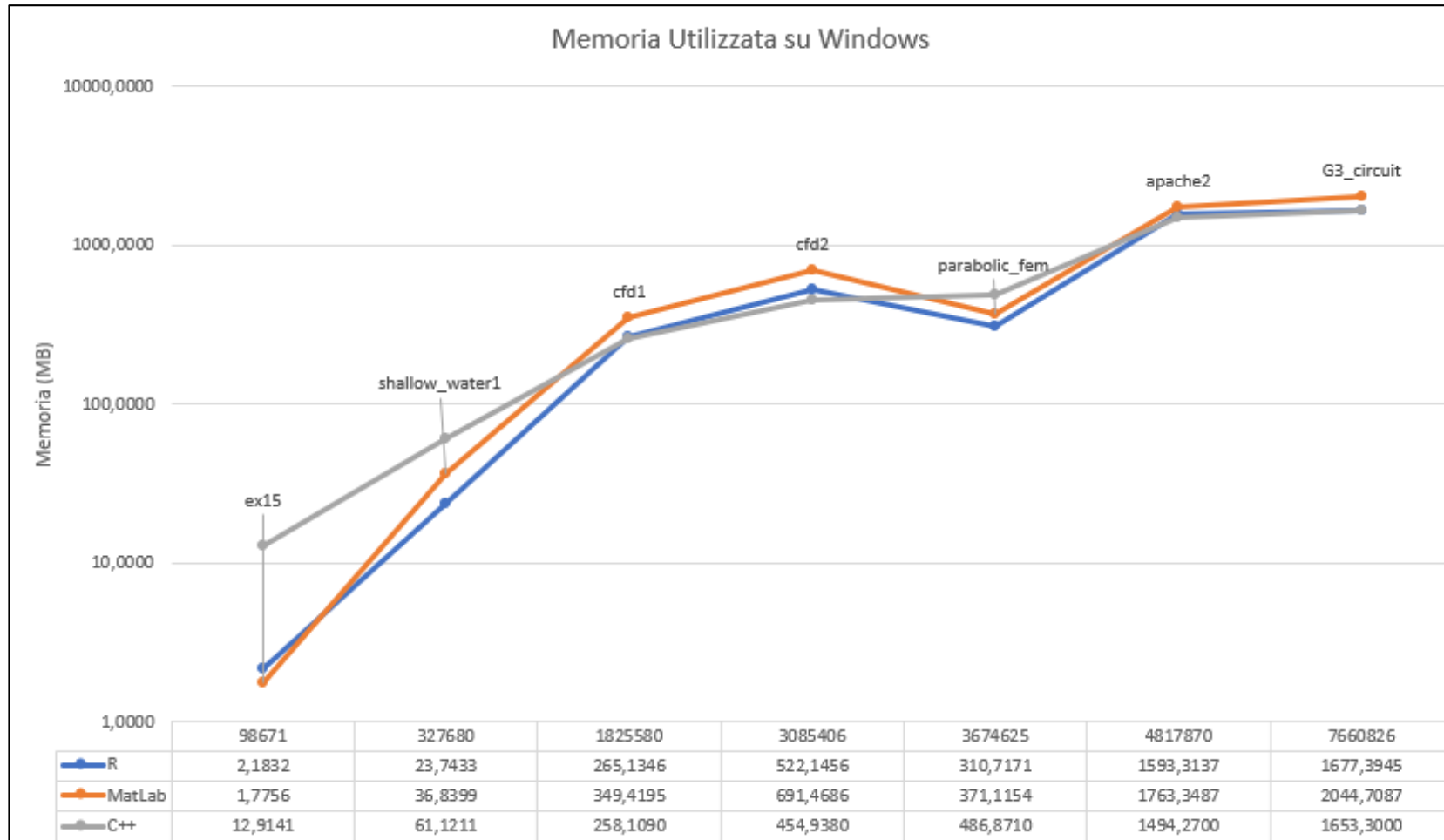
- *Calcolo Velocità di Risoluzione*
- *Calcolo Errore Relativo*
- *Calcolo Memoria Utilizzata*

## Risultati – Memoria Utilizzata su Linux

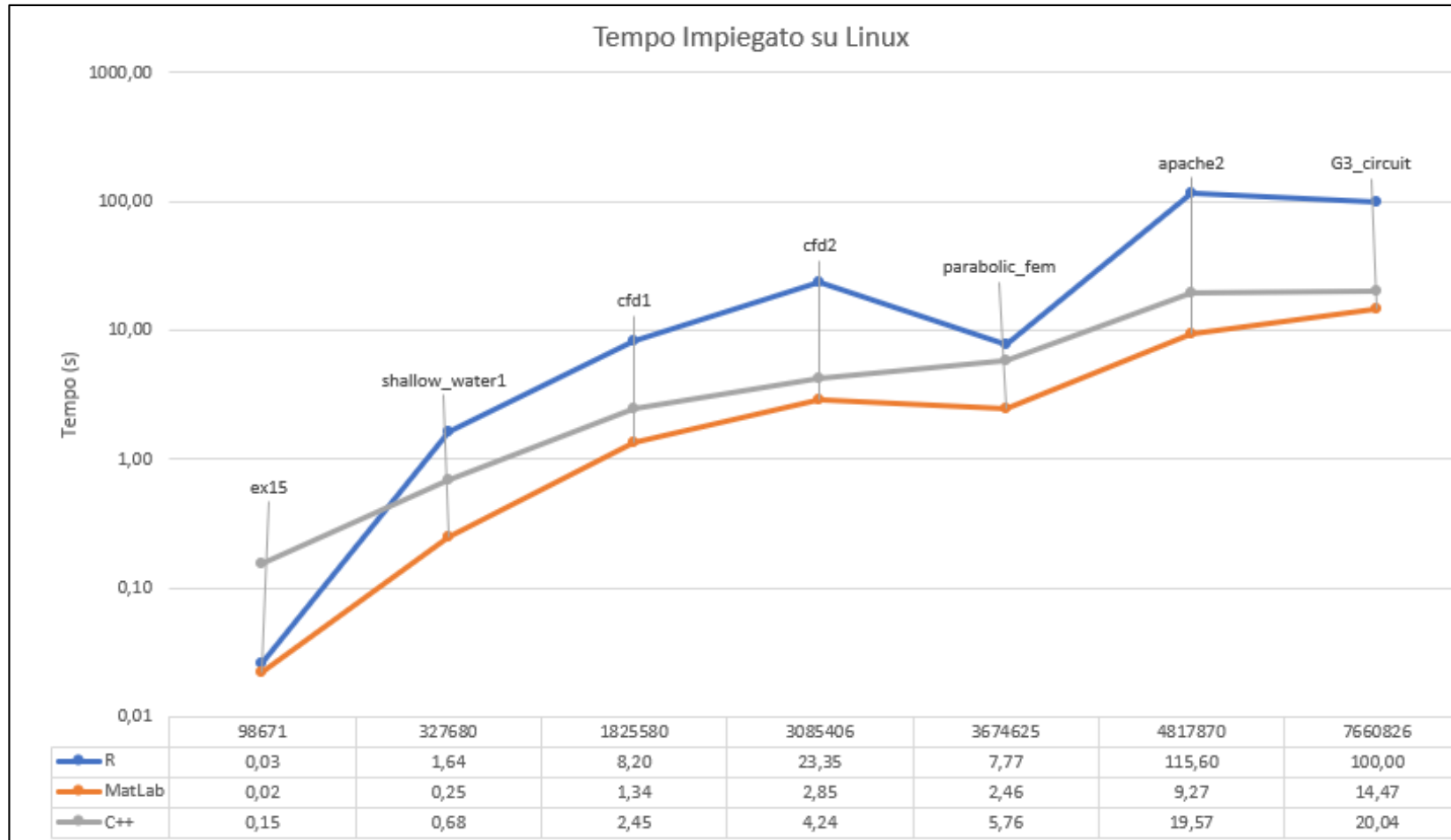




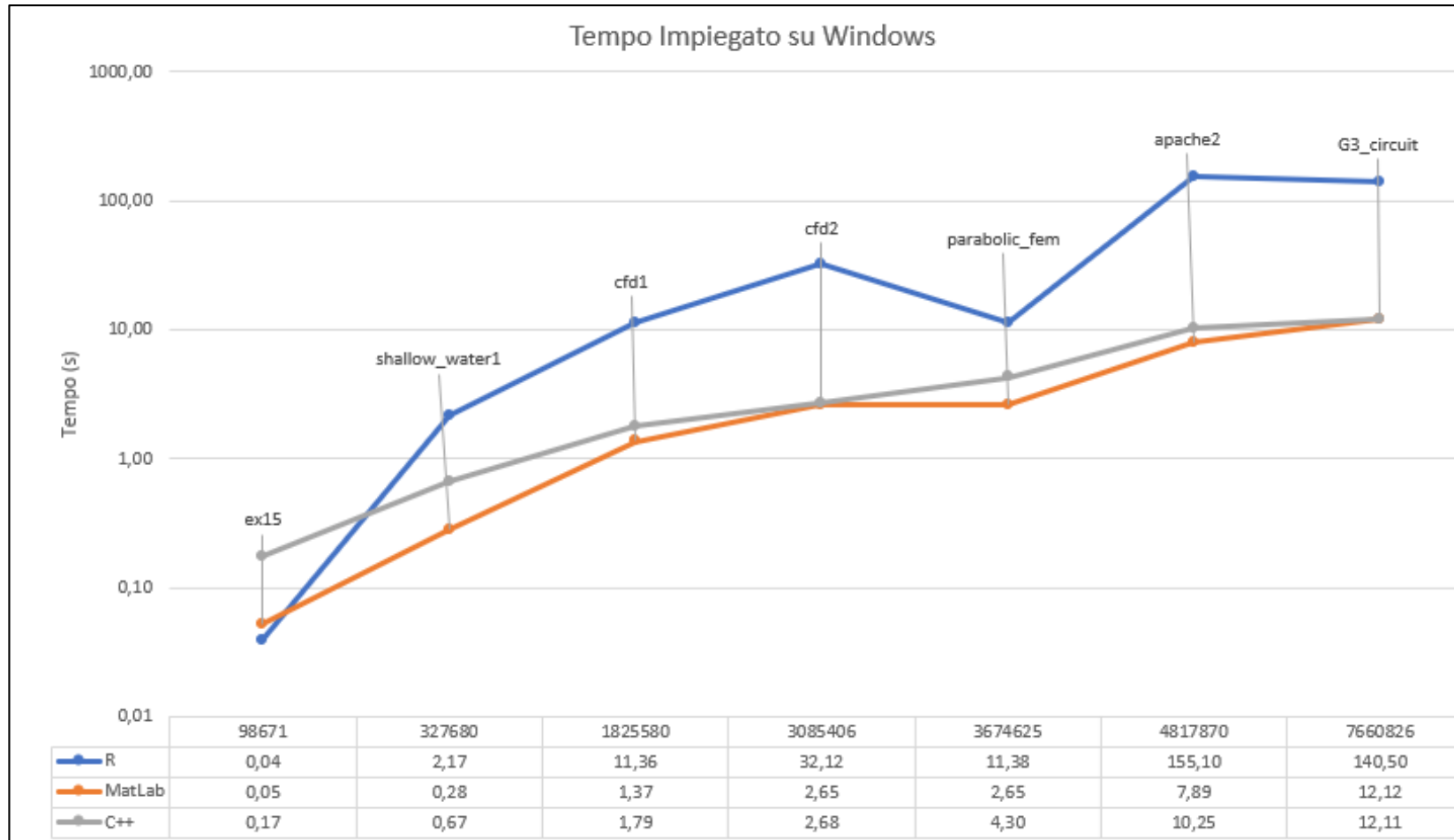
## Risultati – Memoria Utilizzata su Windows



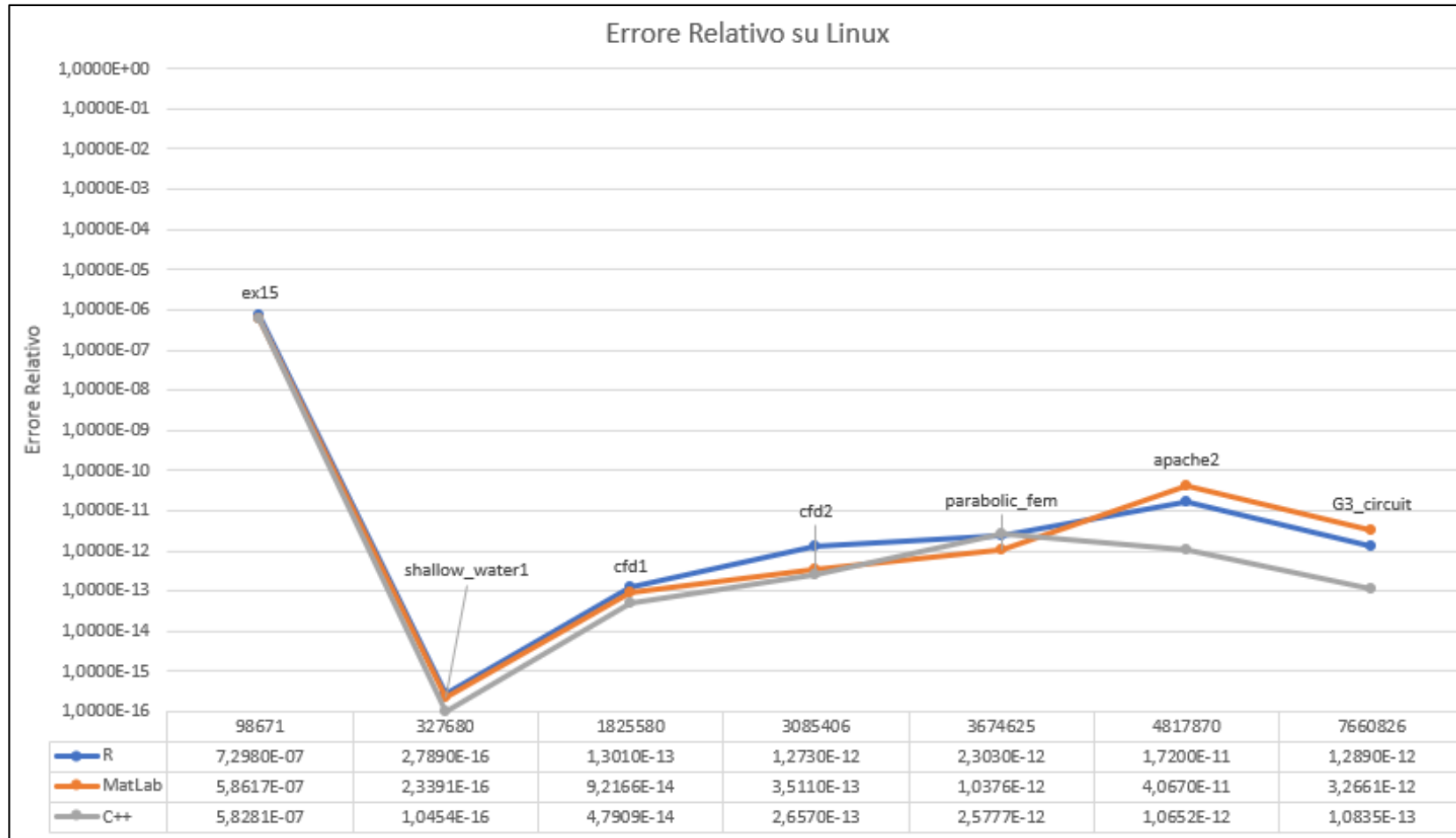
## Risultati – Tempo Impiegato su Linux



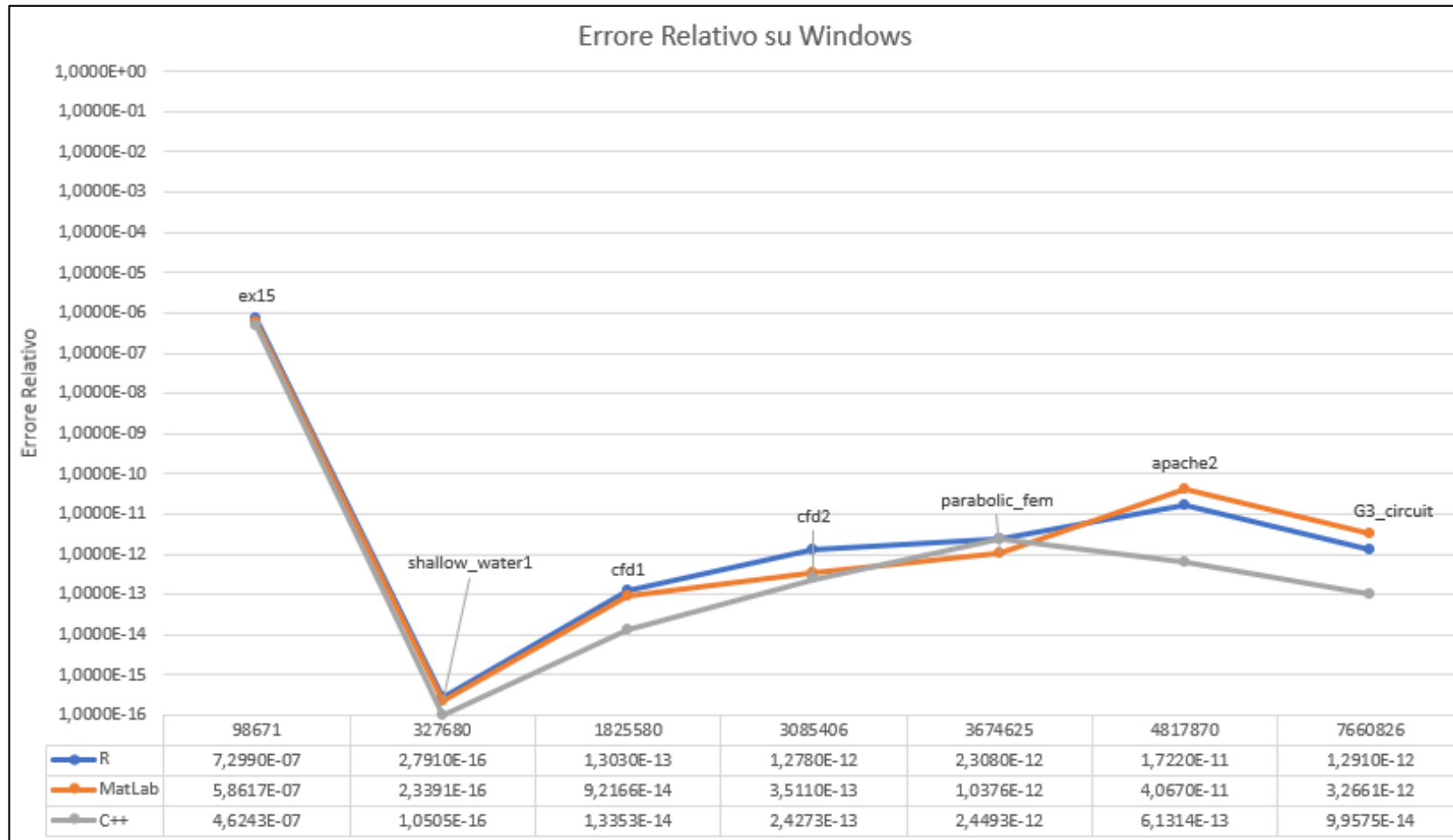
## Risultati – Tempo Impiegato su Windows



## Risultati – Errore Relativo su Linux



## Risultati – Errore Relativo su Windows



# Conclusioni



Linguaggio	Utilizzo	Istallazione	Documentazione	Manutenzione
Matlab	++	+	+	+
C++	+	-	-	+
R	+	+	+	+

Linguaggio	Memoria	Tempo	Errore
Matlab	+	++	+
C++	++	+	++
R	++	-	+

**GRAZIE PER  
L'ATTENZIONE**