

# ALGEBRA LINEARE NUMERICA - METODI DIRETTI PER MATRICI SPARSE

METODI DEL CALCOLO SCIENTIFICO - PROGETTO I

Mattia Pennati	793375
Francesco Prete	793389
Andrea Spreafico	793317

# INTRODUZIONE

- A livello aziendale e di ricerca, una delle sfide principali nella risoluzione di problemi dal costo computazionale elevato è la scelta dello strumento ottimale
- Dal punto di vista informatico la domanda diventa: software «*open source*» o software «*proprietario*»?
  - Esistono differenze rilevanti tra le due categorie?
  - Vale la pena pagare per un servizio? Se sì, quando?
- Lo scopo principale di questo progetto è quello di confrontare le prestazioni di software open source e proprietario, al fine di ottenere le informazioni necessarie per decidere consapevolmente quale sia la scelta migliore nella risoluzione di sistemi lineari per matrici sparse

# SISTEMI LINEARI

- Un sistema lineare è un sistema composto da più equazioni di primo grado che devono essere verificate tutte contemporaneamente
- Queste strutture sono molto ricorrenti in ambiti ingegneristici e fisici (termodinamica, elettromagnetismo..), ma anche informatici (teoria dei grafi, computer vision..)
- Trovare strumenti efficienti ed efficaci per la risoluzione di questo tipo di sistemi è quindi fondamentale

# MATRICE ASSOCIATA AL SISTEMA

- Nella letteratura sono quindi state trovate diverse strategie risolutive per questo tipo di problemi e, le principali, operano sulla matrice associata al sistema
- La matrice associata al sistema è quella matrice che si ricavava dalla giustapposizione di due matrici:
  - la matrice dei coefficienti
  - il vettore dei termini noti del sistema
- Noi ci concentreremo principalmente sulla matrice dei coefficienti che, come ogni altra matrice, può essere «densa» o «sparsa»

# MATRICI SPARSE

- Una matrice si dice sparsa se quasi tutti i suoi elementi sono uguali a zero
- Questa particolarità permette di memorizzare e gestire con metodi efficienti questo tipo di matrici, grazie a strutture dati che ne comprimono la quantità di spazio utilizzato per la memorizzazione
- Questo permette di utilizzare matrici sparse molto più grandi rispetto a quelle dense, creando però grossi problemi computazionali. Per questo motivo, per le finalità di questo lavoro, abbiamo deciso di utilizzare alcune matrici sparse di varie dimensioni relative a problemi reali

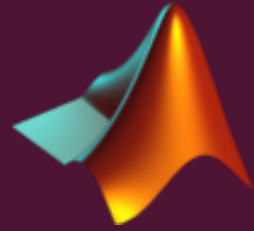
# DATASET [1]

- In particolare verranno considerate le seguenti matrici:

MATRICI NON DEFINITE POSITIVE		MATRICI DEFINITE POSITIVE	
Nome Matrice	# Righe	Nome Matrice	# Righe
ex19	12005	cf1	70656
water_tank	60740	shallow_water1	81920
torso1	116158	parabolic_fem	525825
PR02R	161070	G3_circuit	1585478

# SOFTWARE UTILIZZATI

- I software scelti per il confronto sono Matlab <sup>[2]</sup> e GNU Octave <sup>[3]</sup>
- Questa scelta è stata presa dopo aver confrontato alcuni software e librerie che implementano diverse strategie risolutive (Lu, Cholesky...) per i sistemi lineari basandoci sul lavoro di analisi comparativa svolto da T. Davis <sup>[4]</sup>



# MATLAB

- Matlab (**M**atrix **L**aboratory) è un ambiente di sviluppo per l'omonimo linguaggio, creato dalla MathWorks nel 1984 disponibile ad oggi per Windows, Mac OS, GNU/Linux e Unix
- Matlab è tuttora mantenuto, aggiornato e documentato dalla MathWorks che ne pubblica una nuova versione due volte all'anno
- Matlab implementa, per la risoluzione di sistemi lineari per matrici sparse (e non), la funzione «*mldivide*» <sup>[5]</sup> che sceglie il miglior risolutore in base alla matrice in input. In particolare:
  - La decomposizione LU (tramite la libreria UMFPACK <sup>[6]</sup>) per matrici non definite positive
  - La decomposizione di Cholesky (tramite la libreria CHOLMOD <sup>[7]</sup>) per matrici definite positive



# OSSERVAZIONI SULLA SCELTA DEI SOFTWARE

- Nella scelta di un software da confrontare a Matlab sono stati considerati i seguenti fattori:
  - Abilità del SW nel riconoscere le proprietà di una matrice in modo da poter scegliere e applicare strategie di risoluzione diverse
  - Abilità del SW di utilizzare le stesse strategie di fattorizzazione di Matlab (almeno LU e Choleski)
- Abbiamo considerato svariati software e librerie, ma sono stati esclusi quelli non conformi alle specifiche sopra riportate (ad esempio la libreria Scipy di Python, che permette di utilizzare solamente la fattorizzazione LU)
- Dopo vari confronti, abbiamo dedotto che le alternative migliori erano quelle «Matlab-like», ovvero ambienti di sviluppo integrati ingegneristico-matematici



# OCTAVE

- GNU Octave è un ambiente di sviluppo open source con un linguaggio di scripting simile a quello di Matlab, nato come strumento di calcolo per l'ingegneria chimica nel 1988
- Octave è stato sviluppato da John Eaton (che è anche il primo manutentore e colui che ne ha scritto la documentazione) e James Rawlings. Ad oggi è disponibile per Windows, Mac OS, GNU/Linux e Unix ed è possibile contribuire al progetto attraverso una repository pubblica su GitHub <sup>[8]</sup>
- Anche Octave implementa per la risoluzione di sistemi lineari «*mldivide*» <sup>[9]</sup> che utilizza le stesse librerie (e risolutori) dell'omonima funzione Matlab

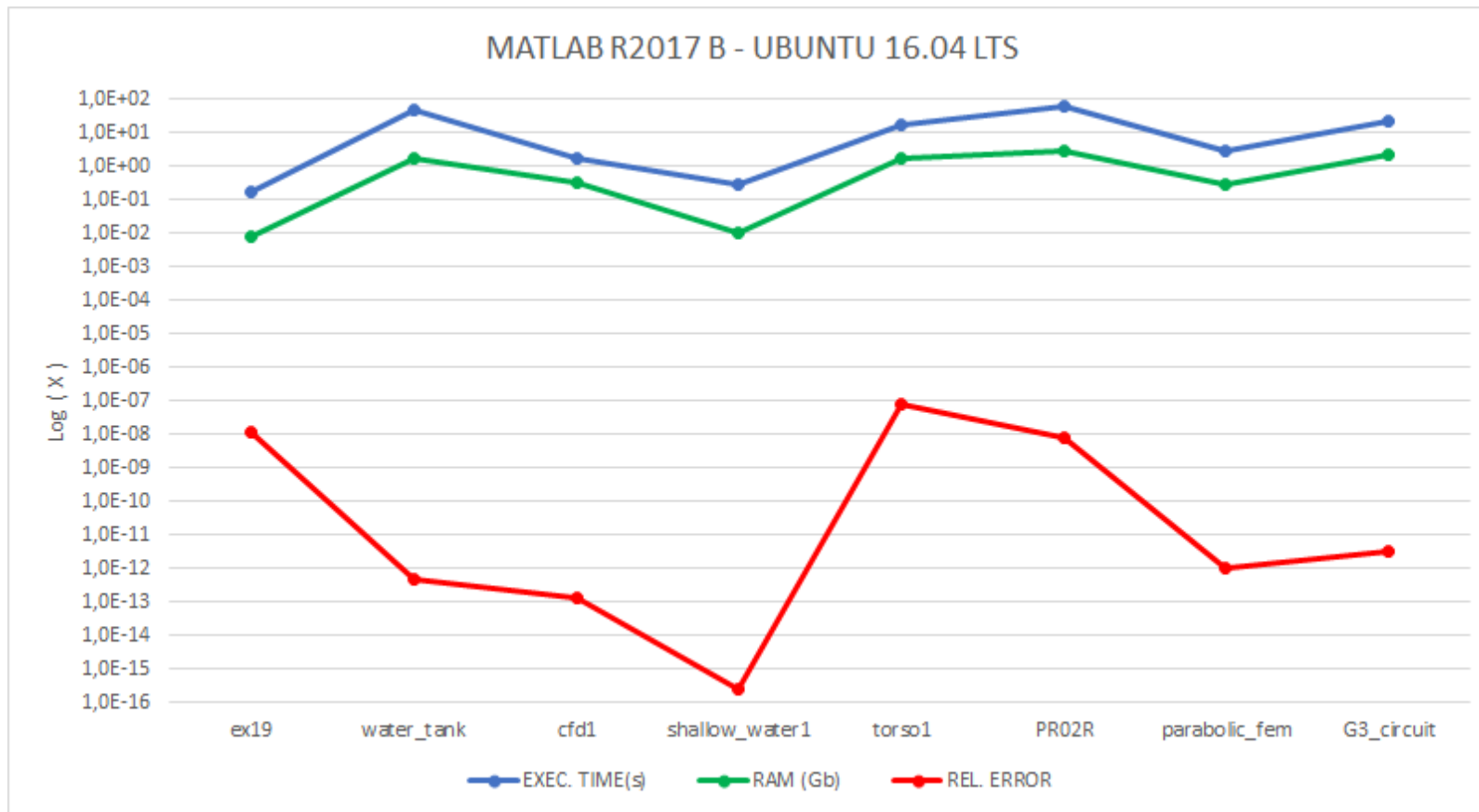
# CRITERI DI VALUTAZIONE

- I due software appena descritti sono stati testati in due diversi sistemi operativi (Ubuntu 16.04 LTS e Windows 10 Home) sullo stesso PC (8 GB di Ram, Intel Core i5 1.2 GHz)
- Per ogni matrice usata come test sono riportati:
  - Tempo di esecuzione della risoluzione del sistema lineare associato alla matrice (calcolati, in entrambi i casi con le apposite funzioni di profiling dei software)
  - Spazio occupato in RAM durante l'esecuzione dell'operazione (ovvero l'aumento dell'utilizzo della RAM durante l'esecuzione considerando la differenza tra il picco massimo e la RAM pre-operazione)
  - Errore relativo della soluzione ottenuta (calcolato come  $\frac{\|x - x_e\|_2}{\|x_e\|_2}$ )

# PRESENTAZIONE DEI RISULTATI

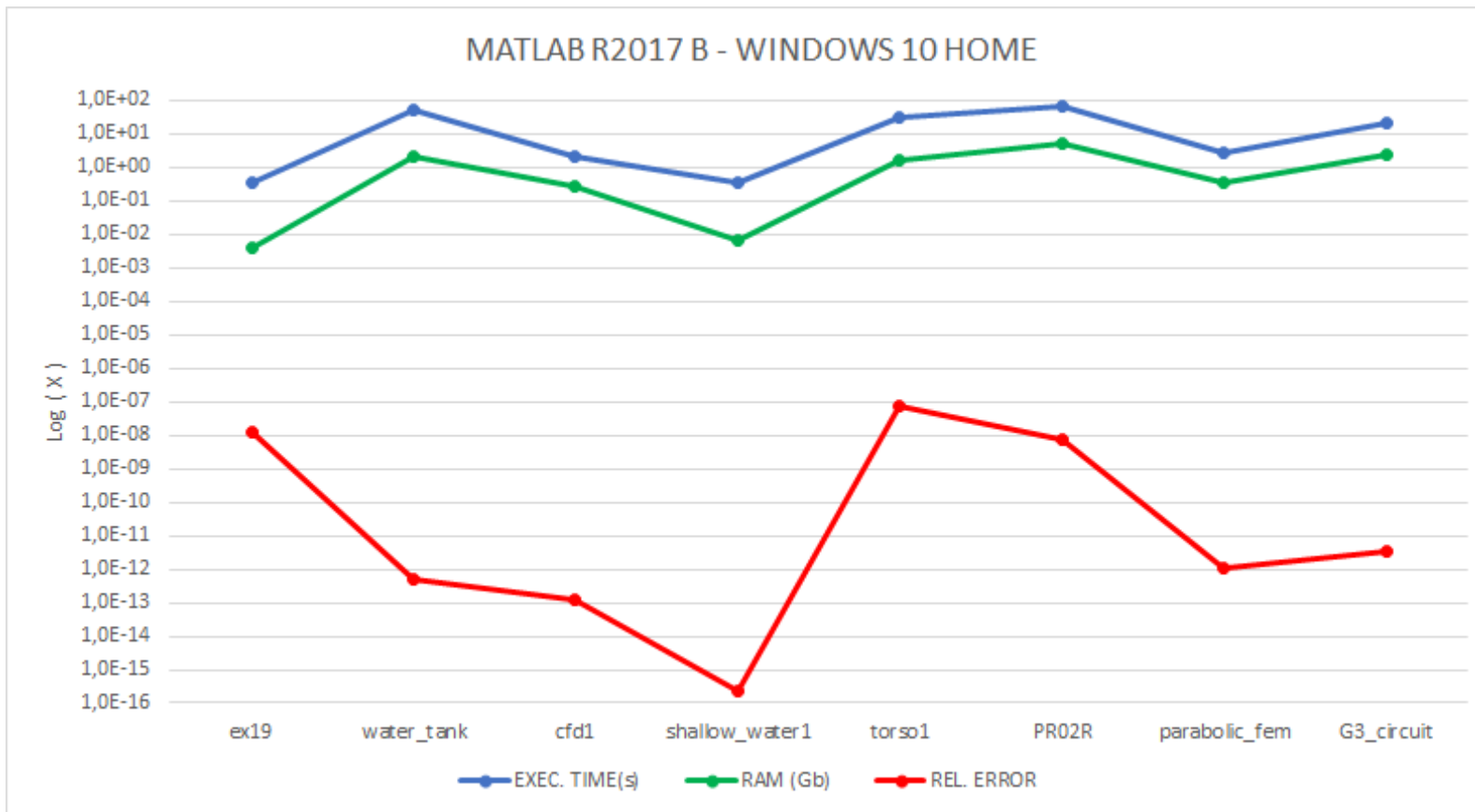
- Questi dati saranno riportati in 4 grafici in cui ognuno mostra i tre parametri (tempo di esecuzione, utilizzo di memoria ed errore relativo) per ogni configurazione software-sistema operativo
- Inoltre verranno riportati alcuni grafici comparativi che andranno ad analizzare ogni singolo parametro sulle diverse configurazioni

# RISULTATI MATLAB - UBUNTU



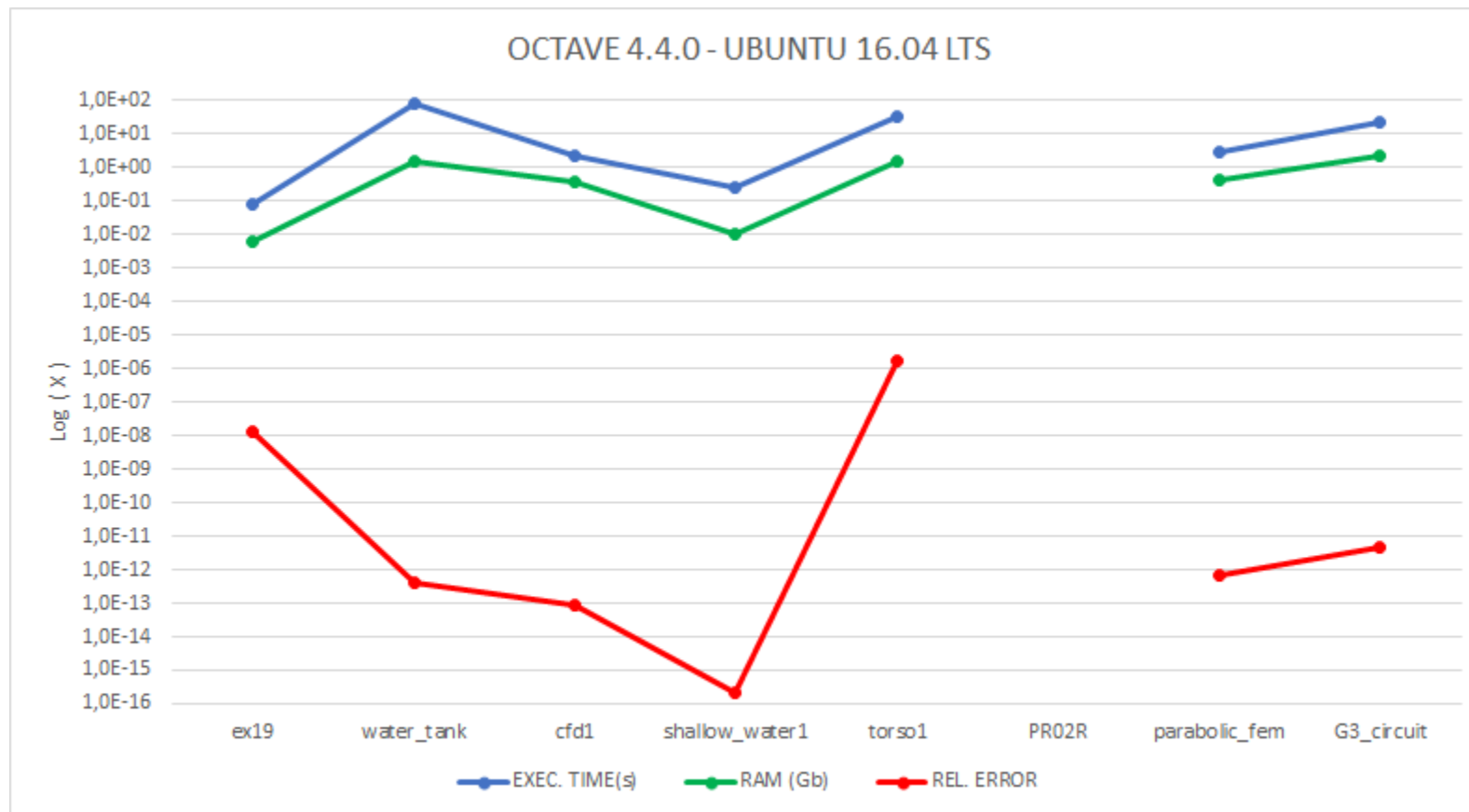
EXEC. TIME(s)	RAM (Gb)	REL. ERROR
0,1641	0,0077	1,21E-08
44,8851	1,617	4,81E-13
1,7704	0,3097	1,25E-13
0,2726	0,0104	2,38E-16
16,0361	1,771	7,79E-08
57,7414	2,7896	7,64E-09
2,713	0,2849	1,06E-12
21,5417	2,21144	3,23E-12

# RISULTATI MATLAB - WINDOWS



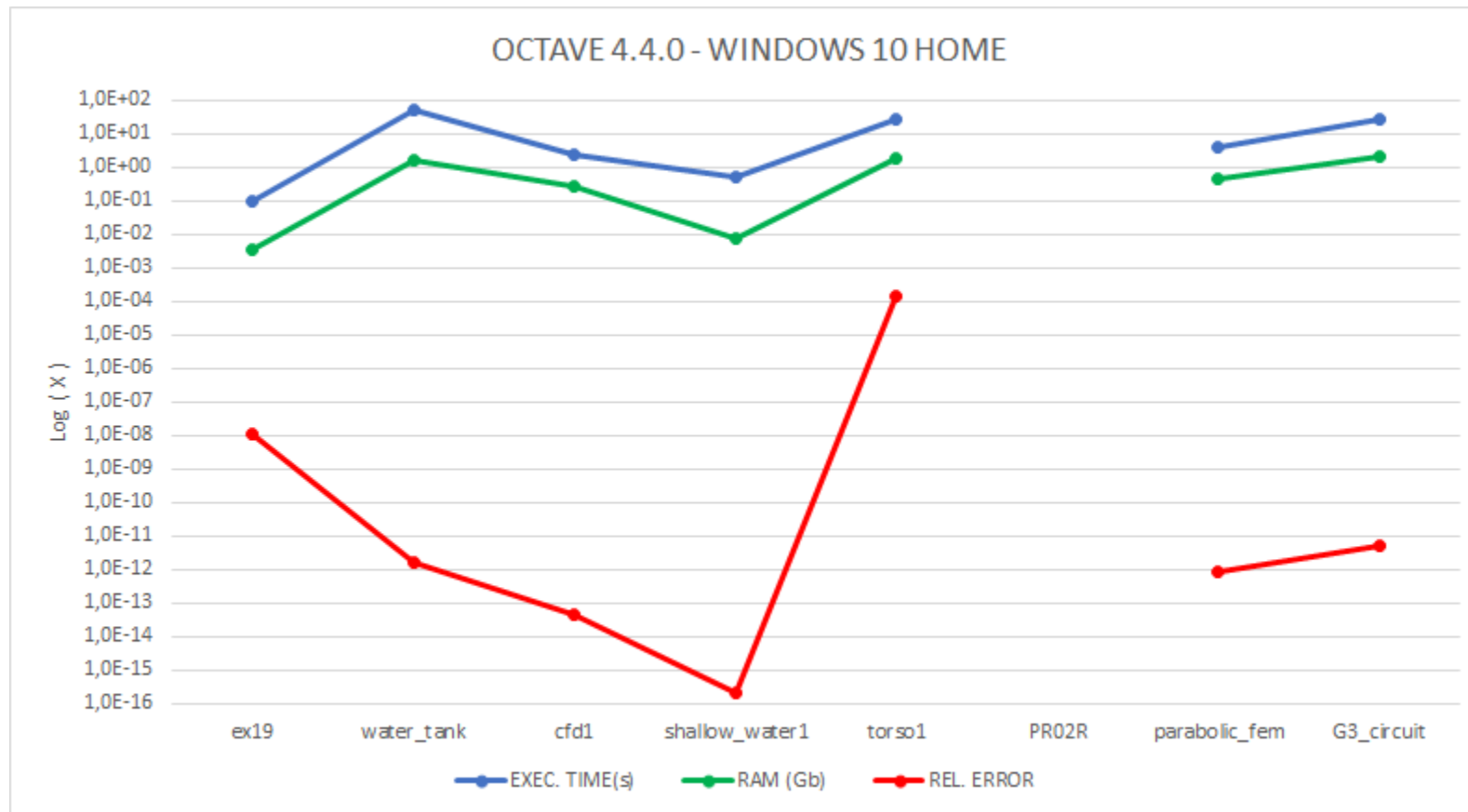
EXEC. TIME(s)	RAM (Gb)	REL. ERROR
0,3679	0,0041	1,21E-08
51,9891	2,133	4,81E-13
1,9481	0,2844	1,25E-13
0,3258	0,0064	2,38E-16
30,646	1,5958	7,79E-08
69,2541	5,0718	7,64E-09
2,6198	0,3634	1,06E-12
20,6289	2,291	3,23E-12

# RISULTATI OCTAVE - UBUNTU



EXEC. TIME(s)	RAM (Gb)	REL. ERROR
0,0743	0,0062	1,34E-08
74,554	1,452	4,09E-13
2,1882	0,3619	8,33E-14
0,2631	0,0108	2,02E-16
32,9366	1,4861	1,79E-06
NC***	NC***	NC***
2,89	0,4019	6,98E-13
23,3488	2,0975	4,89E-12

# RISULTATI OCTAVE - WINDOWS

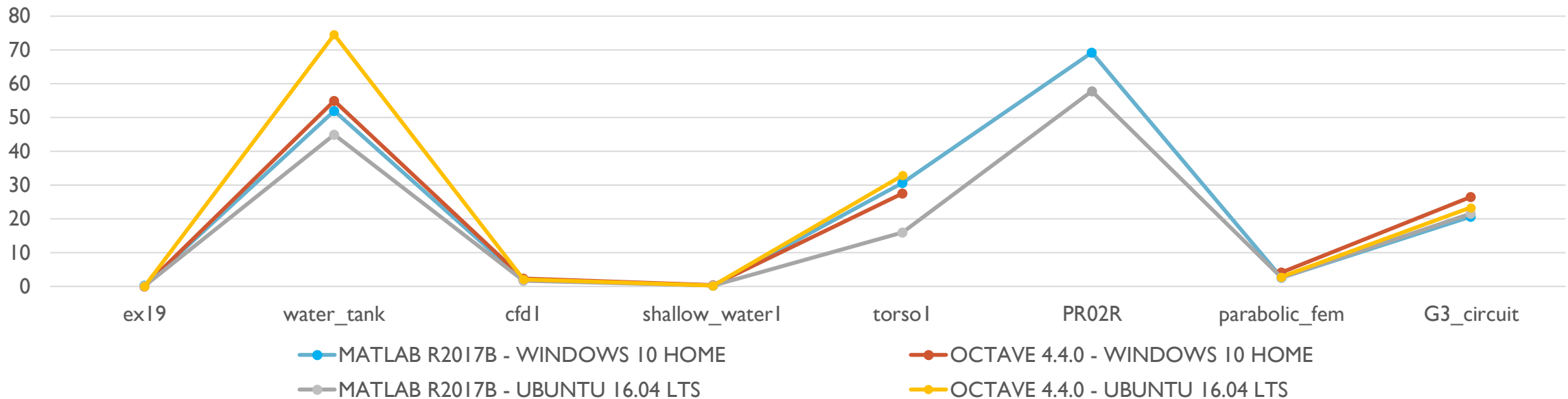


EXEC. TIME(s)	RAM (Gb)	REL. ERROR
0,1033	0,0037	1,16E-08
54,985	1,7222	1,50E-12
2,3576	0,2849	4,42E-14
0,4863	0,0072	2,05E-16
27,5227	1,817	1,47E-04
NC***	NC***	NC***
4,1672	0,4582	8,24E-13
26,4748	2,2278	4,91E-12



# GRAFICO COMPARATIVO – EXECUTION TIME

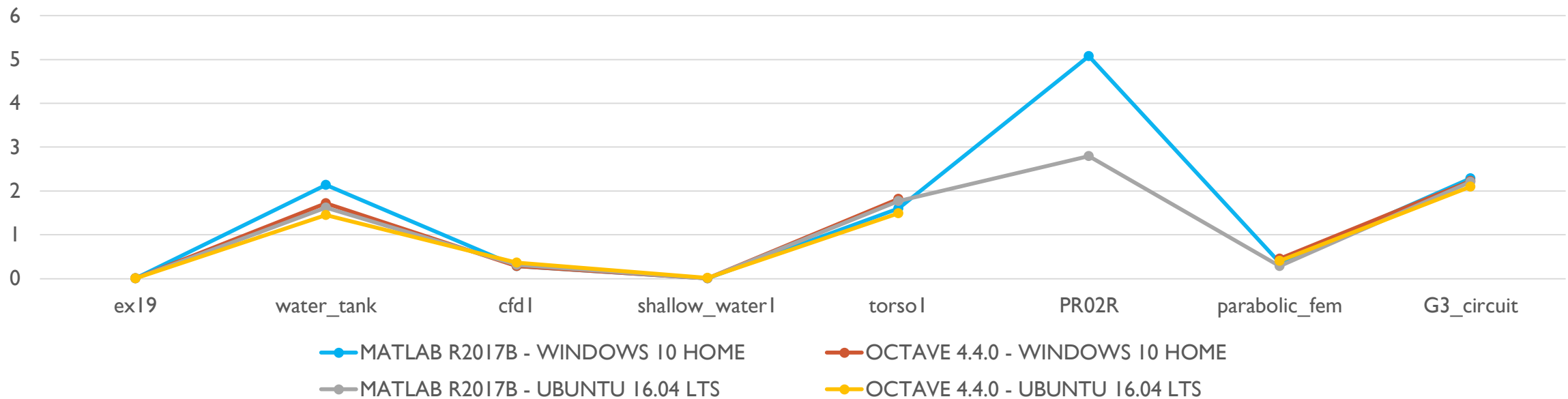
EXECUTION TIME (seconds)



Matrix Name	ex19	water_tank	Cfd1	shallow_water1	torso1	PR02R	parabolic_fem	G3_circuit
MATLAB R2017B - WINDOWS 10 HOME	0,3679	51,9891	1,9481	0,3258	30,646	69,2541	2,6198	20,6289
OCTAVE 4.4.0 - WINDOWS 10 HOME	0,1033	54,985	2,3576	0,4863	27,5227	NC***	4,1672	26,4748
MATLAB R2017B - UBUNTU 16.04 LTS	0,1641	44,8851	1,7704	0,2726	16,0361	57,7414	2,713	21,5417
OCTAVE 4.4.0 - UBUNTU 16.04 LTS	0,0743	74,554	2,1882	0,2631	32,9366	NC***	2,89	23,3488

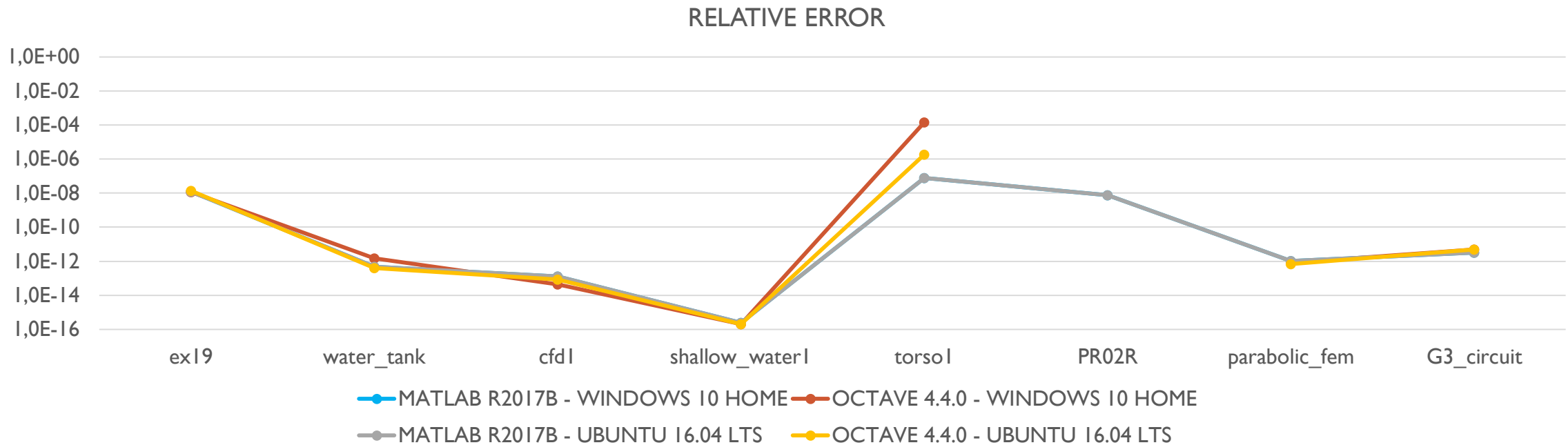
# GRAFICO COMPARATIVO – MEMORY USAGE

MEMORY USAGE (GB)













Matrix Name	ex19	water_tank	cfd1	shallow_water1	torso1	PR02R	parabolic_fem	G3_circuit
MATLAB R2017B - WINDOWS 10 HOME	0,0041	2,133	0,2844	0,0064	1,5958	5,0718	0,3634	2,291
OCTAVE 4.4.0 - WINDOWS 10 HOME	0,0037	1,7222	0,2849	0,0072	1,817	NC***	0,4582	2,2278
MATLAB R2017B - UBUNTU 16.04 LTS	0,0077	1,617	0,3097	0,0104	1,771	2,7896	0,2849	2,21144
OCTAVE 4.4.0 - UBUNTU 16.04 LTS	0,0062	1,452	0,3619	0,0108	1,4861	NC***	0,4019	2,0975

# GRAFICO COMPARATIVO – RELATIVE ERROR







Matrix Name	ex19	water_tank	cfd1	shallow_water1	torso1	PR02R	parabolic_fem	G3_circuit
MATLAB R2017B - WINDOWS 10 HOME	1,213E-08	4,8123E-13	1,2527E-13	2,381E-16	7,7926E-08	7,6393E-09	1,0572E-12	3,2311E-12
OCTAVE 4.4.0 - WINDOWS 10 HOME	1,15926E-08	1,5004E-12	4,41776E-14	2,05012E-16	0,000147214	NC***	8,23914E-13	4,90661E-12
MATLAB R2017B - UBUNTU 16.04 LTS	1,213E-08	4,8123E-13	1,2527E-13	2,381E-16	7,7926E-08	7,6393E-09	1,0572E-12	3,2311E-12
OCTAVE 4.4.0 - UBUNTU 16.04 LTS	1,34063E-08	4,08668E-13	8,33296E-14	2,01554E-16	1,78544E-06	NC***	6,97774E-13	4,89426E-12

# ANALISI DEI RISULTATI – OSSERVAZIONI GENERALI

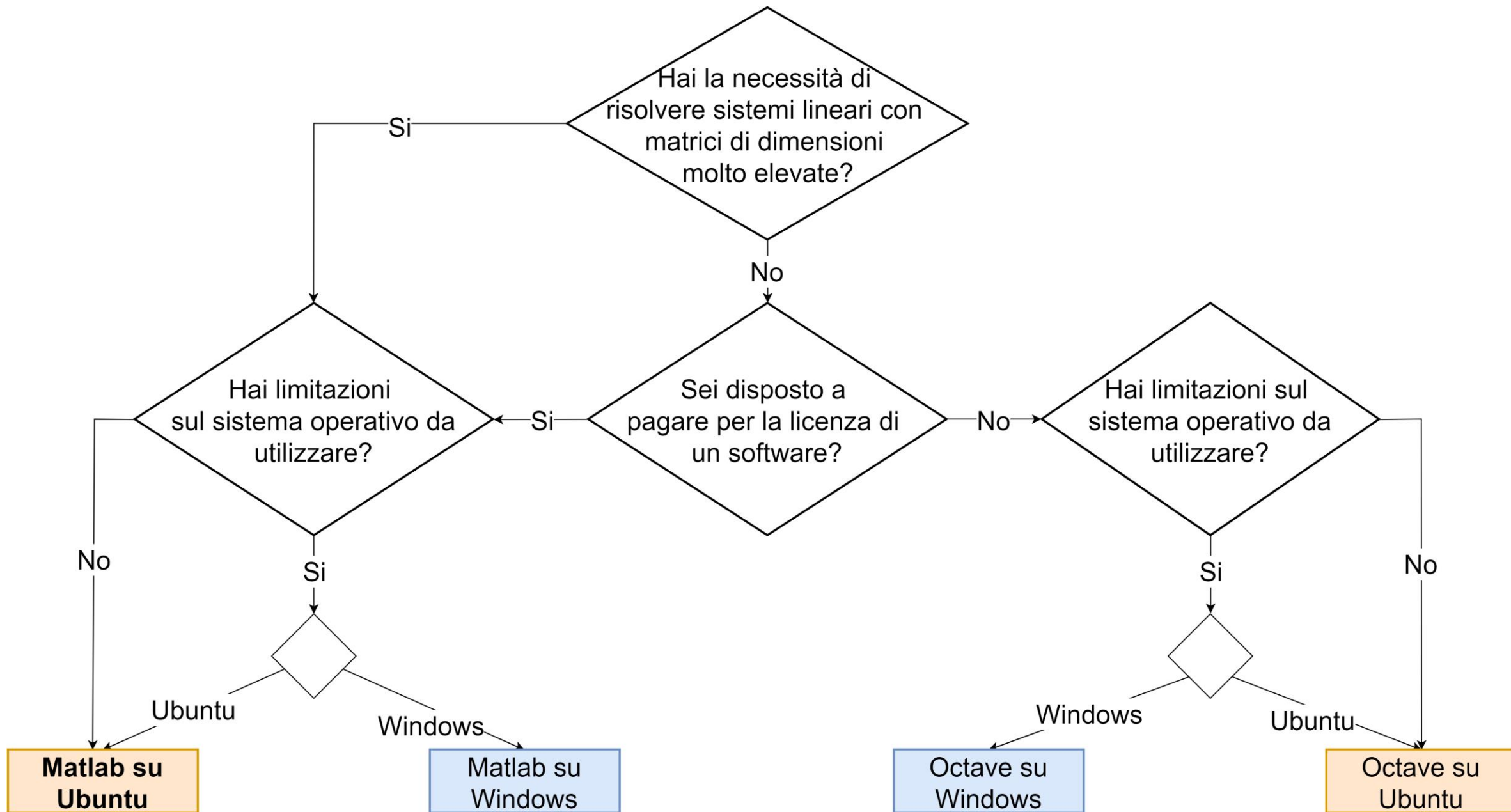
	Matlab	Octave
Software gratuito?		
La documentazione è esaustiva?		
La documentazione è ben strutturata?		
Tutorial ed esempi online?		
Tutorial ed esempi online sufficienti?		

# ANALISI DEI RISULTATI – SW & OS

Matlab			Octave	
Windows	Ubuntu		Windows	Ubuntu
		Il software permette di calcolare tutte le matrici senza problemi?		
2	1	Miglior tempo di esecuzione?	3	4
3	2	Miglior utilizzo della memoria?	3	1
1		Errore relativo?	3	2

# CONCLUSIONI

- Tra le configurazioni analizzate, Matlab su Ubuntu risulta quella con le performance migliori. Matlab, inoltre, garantisce una miglior esperienza d'uso grazie ad un'ottima documentazione e la presenza di una grandissima quantità di guide e tutorial online. In conclusione, Matlab offre un servizio per il quale vale la pena pagare
- Per quanto riguarda problemi con matrici di dimensioni ridotte, ovvero quelle che necessitano di una mole computazionale limitata, Octave rappresenta un'alternativa con performance che si discostano di poco da quelle di Matlab (riguardo tutti i parametri). Se, però, le dimensioni della matrice aumentano, Octave non è in grado di terminare la computazione, rendendo Matlab l'unica soluzione
- La scelta del software (e più in generale della configurazione) dipende quindi dal tipo di problema da affrontare e dalla disponibilità economica



# BIBLIOGRAFIA - I

- [1] Timothy A. Davis, Yifan Hu, “The university of Florida sparse matrix collection”, ACM Trans. Math. Softw. 38, 1, Article 1, 25 pages (2011)
- [2] MATLAB and Statistics Toolbox Release 2017b The MathWorks, Inc., Natick, Massachusetts, United States, <https://it.mathworks.com/help/matlab/>
- [3] John W. Eaton, David Bateman, Søren Hauberg, Rik Wehbring (2017) GNU Octave version 4.4.0 manual: a high-level interactive language for numerical computations, <https://www.gnu.org/software/octave/doc/v4.4.0/>
- [4] Davis, Timothy A. Direct methods for sparse linear systems. Vol. 2, pp. 141-143, Siam, 2006
- [5] <https://it.mathworks.com/help/matlab/ref/mldivide.html>, 29/05/2018



## BIBLIOGRAFIA - 2

- [6] Davis Timothy A,. "Algorithm 832". *ACM Transactions on Mathematical Software*. **30** (2), pp.196–199 (2004)
- [7] Chen Yanqing, et al. "Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate." *ACM Transactions on Mathematical Software (TOMS)* 35.3 (2008): 22
- [8] <https://github.com/NexMirror/Octave>, 29/05/2018
- [9] <https://octave.org/doc/v4.0.0/Sparse-Linear-Algebra.html>, 29/05/2018