Implementation of Buckingham's Pi theorem using Python

Item Type Journal Article
Author Pankaj Dumka
Author Rishika Chauhan
Author Ayush Singh
Author Gaurav Singh
Author Dhananjay Mishra
URL https://www.sciencedirect.com/science/article/pii/S0965997822001363
Volume 173
Pages 103232
Publication Advances in Engineering Software
ISSN 0965-9978
Date 2022-11-01
Journal Abbr Advances in Engineering Software
DOI 10.1016/j.advengsoft.2022.103232
Accessed 7/17/2023, 8:32:59 PM
Library Catalog ScienceDirect
Language en
Abstract Buckingham's Pi theorem plays an important role in engineering, applied mathematics, and physics for dimensional analysis. From the given variables, it will be utilised to evaluate the set of dimensionless parameters. It indicates that the validity of the physical law is independent of the specific unit system, and it can be expressed as an identity incorporating only dimensionless variables associated with the law. A python-based function has been developed and reported in this manuscript, which can be utilised to evaluate Pi terms (commonly called π terms) for any fluid flow problem. Smaller moderation in the written function can make it capable of solving any fundamental dimensions. Different fluid mechanics problems are utilised to test and validate the reported function, five of which are presented in this manuscript along with code. Obtained results are in good agreement with the theoretically obtained results.
Date Added 7/17/2023, 8:32:59 PM
Modified 7/11/2025, 3:17:34 AM

Tags:

Buckingham's Pi theorem, BuPi, Dimensional analysis, Fluid mechanics, nrv, Physical similarity analysis, Python programming, RV,rv, SI

Attachments

- ScienceDirect Full Text PDF
- ScienceDirect Snapshot

Computer performance analysis and the Pi Theorem

Item Type Journal Article
Author Robert W. Numrich
URL https://doi.org/10.1007/s00450-010-0147-8
Volume 29
Issue 1
Pages 45–71
Publication Computer Science - Research and Development
ISSN 1865-2034
Date February 1, 2014
Journal Abbr Comput. Sci.
DOI 10.1007/s00450-010-0147-8
Accessed 3/4/2023, 4:11:31 PM
Library Catalog February 2014
Abstract This paper applies the Pi Theorem of dimensional analysis to a representative set of examples from computer performance analysis. It is a survey paper that takes a different look at problems involving latency, bandwidth, cache-miss ratios, and the efficiency of parallel numerical algorithms. The Pi Theorem is the fundamental tool of dimensional analysis, and it applies to problems in computer performance analysis just as well as it does to problems in other sciences. Applying it requires the definition of a system of measurement appropriate for computer performance analysis with a consistent set of units and dimensions. Then a straightforward recipe for each specific problem reduces the number of independent variables to a smaller number of dimensionless parameters. Two machines with the same values of these parameters are self-similar and behave the same way. Self-similarity relationships emphasize how machines are the same rather than how they are different. The Pi Theorem is simple to state and simple to prove, using purely algebraic methods, but the results that follow from it are often surprising and not simple at all. The results are often unexpected but they almost always reveal something new about the problem at hand.
Date Added 3/4/2023, 4:11:31 PM
Modified 7/8/2025, 10:54:25 PM

Tags:

Dimensional analysis, Bandwidth, Cache behavior, Computational force, Computational intensity, Latency, Parallel algorithms, Performance analysis, Performance metrics, Pi Theorem, Scaling

Notes:

Cited By :2

Attachments

- Numrich - 2014 - Computer performance analysis and the Pi Theorem.pdf
- ResearchGate Link
- Snapshot

Self-similarity of parallel machines

Item Type	Journal Article
Author	Robert W. Numrich
Author	Michael A. Heroux
URL	https://www.sciencedirect.com/science/article/pii/S0167819110001444
Volume	37
Issue	2
Pages	69-84
Publication	Parallel Computing
ISSN	0167-8191
Date	2011-02-01
Journal Abbr	Parallel Computing
DOI	10.1016/j.parco.2010.11.003
Abstract	Self-similarity is a property of physical systems that describes how to scale parameters such that dissimilar systems appear to be similar. Computer systems are self-similar if certain ratios of computational forces, also known as computational intensities, are equal. Two machines with different computational power, different network bandwidth and different inter-processor latency behave the same way if they have the same ratios of forces. For the parallel conjugate gradient algorithm studied in this paper, two machines are self-similar if and only if the ratio of one force describing latency effects to another force describing bandwidth effects is the same for both machines. For the two machines studied in this paper, this ratio, which we call the mixing coefficient, is invariant as problem size and processor count change. The two machines have the same mixing coefficient and belong to the same equivalence class.
Date Added	7/8/2025, 7:09:34 PM
Modified	7/14/2025, 11:04:35 PM

Tags:

Benchmark analysis, Computational force, Computational intensity, Dimensional analysis, Equivalence class, Mixing coefficient, Parallel algorithms, Scaling, Self-similarity

Attachments

- ScienceDirect Full Text PDF
- ScienceDirect Snapshot

Computational forces in the SAGE benchmark

Item Type	Journal Article
Author	Robert W. Numrich
URL	https://www.sciencedirect.com/science/article/pii/S074373150800213X
Volume	69
Issue	3
Pages	315-325
Publication	Journal of Parallel and Distributed Computing
ISSN	0743-7315
Date	2009-03-01
Journal Abbr	Journal of Parallel and Distributed Computing
DOI	10.1016/j.jpdc.2008.12.001
Accessed	2/4/2024, 12:13:35 AM
Library Catalog	ScienceDirect
Abstract	Dimensional analysis applied to a complicated timing formula for the SAGE benchmark yields new insight into the limits to scalability. A single surface, defined by two curvilinear coordinates, describes the parallel efficiency of the benchmark. Each machine, as a function of the number of processors, follows its own path on the surface determined by dimensionless ratios of hardware forces to software forces. Two machines with the same ratios follow the same path and are self-similar, even though the numerical value of each individual force may be different. For this benchmark, latency effects are unimportant relative to bandwidth effects because of the slab decomposition used to distribute the problem across processors. To a good first-order approximation, a single force ratio describes the efficiency as a function of the number of processors. A simpler model, with a single dimensionless exponent, describes the first-order behavior of the computational power as a function of the number of processors.
Date Added	2/4/2024, 12:13:35 AM
Modified	2/4/2024, 2:12:20 AM

Tags:

Dimensional analysis, Computational force, Computational intensity, Parallel algorithms, Scaling, Benchmark analysis, Self-similarity

Attachments

- Numrich - 2009 - Computational forces in the SAGE benchmark.pdf
- ScienceDirect Snapshot

Computational forces in the Linpack benchmark

Item Type	Journal Article
Author	Robert W. Numrich
URL	https://www.sciencedirect.com/science/article/pii/S074373150800035X
Volume	68
Issue	9
Pages	1283-1290
Publication	Journal of Parallel and Distributed Computing
ISSN	0743-7315
Date	2008-09-01
Journal Abbr	Journal of Parallel and Distributed Computing
DOI	10.1016/j.jpdc.2008.02.008
Accessed	6/2/2023, 12:24:32 AM
Library Catalog	ScienceDirect
Language	en
Abstract	Dimensional analysis reduces a complicated ten-parameter formula for the execution time of the Linpack benchmark to a simpler two-parameter formula. These two parameters are ratios of software forces and hardware forces that determine a self-similarity surface. Machines move along paths on this surface as the problem size and the number of processors change. Two machines scale the same way, they move along the same path, if they have the same hardware forces. To design efficient algorithms, the programmer must produce software forces large enough to overcome the hardware forces. Modern machines have larger hardware forces than older machines and are harder to program.
Date Added	6/2/2023, 12:24:32 AM
Modified	2/4/2024, 12:16:20 AM

Tags:

Dimensional analysis, Scalability, Computational force, Computational intensity, Performance analysis, Linpack benchmark, Parallel numerical algorithms

Attachments

- ScienceDirect Full Text PDF
- ScienceDirect Snapshot

Dimensional Analysis Applied to a Parallel QR Algorithm

Item Type	Conference Paper
Author	Robert W. Numrich
Editor	Roman Wyrzykowski
Editor	Jack Dongarra
Editor	Konrad Karczewski

Editor	Jerzy Wasniewski
Place	Berlin, Heidelberg
Publisher	Springer
Pages	148-155
ISBN	978-3-540-68111-3
Date	2008
DOI	10.1007/978-3-540-68111-3_16
Library Catalog	Springer Link
Language	en

Abstract We apply dimensional analysis to a formula for execution time for a QR algorithm from a paper by Henry and van de Geijn. We define a single efficiency surface that reduces performance analysis for this algorithm to an exercise in differential geometry. As the problem size and the number of processors change, different machines move along different paths on the surface determined by two computational forces specific to each machine. We show that computational force, also called computational intensity, is a unifying concept for understanding the performance of parallel numerical algorithms.

Proceedings Title	Parallel Processing and Applied Mathematics
Date Added	7/13/2024, 8:54:20 PM
Modified	7/13/2024, 8:54:20 PM

Tags:

Scalability, computational force, computational intensity, dimensional analysis, parallel numerical algorithms, performance analysis

Attachments

- Full Text PDF