matlab remainder

matlab remainder

In MATLAB, the concept of the "remainder" is fundamental in various mathematical and computational applications, especially when dealing with modular arithmetic, cyclic processes, and partitioning data. The remainder operation in MATLAB allows users to find the leftover part of a division between two numbers, which is crucial in algorithms that require modular calculations, such as cryptography, signal processing, and numerical methods. Understanding how MATLAB handles the remainder and related functions can significantly improve the robustness and correctness of your mathematical computations. This article provides an in-depth exploration of the MATLAB remainder concept, including its functions, differences from similar operations, practical applications, and common pitfalls.

Understanding the Remainder in MATLAB

What Is the Remainder?

The remainder of a division is the amount left over after dividing one number by another. For example, dividing 17 by 5 yields a quotient of 3 with a remainder of 2 because:

17 ÷ 5 = 3 (quotient) with a leftover of 2 (remainder)

In mathematical terms, for two numbers a and b, the remainder r satisfies:

$$a = b \times q + r$$

where q is the quotient (integer division result), and r is the remainder.

In MATLAB, the remainder is typically computed for scalar values, vectors, or matrices, depending on the function used, with specific behaviors and edge cases.

Key MATLAB Functions for Remainder Calculation

MATLAB provides several functions related to remainders and modular arithmetic, each serving different purposes:

- **rem**: Computes the remainder after division, with sign considerations based on the dividend.
- **mod**: Computes the modulus, always returning a non-negative result for positive divisor.

- **factor**: Factors integers into prime factors, indirectly related to remainders in some algorithms.

- **dividend** and **divisor**: Not MATLAB functions, but conceptual components in division operations.

Each of these functions has distinct behaviors, especially when dealing with negative numbers or matrices.

MATLAB Remainder Functions in Detail

1. rem Function

The rem function computes the remainder after division, with the remainder having the same sign as the dividend:

```matlab
r = rem(a, b);
```

Behavior:

- If \(a\) and \(b\) are scalars, vectors, or matrices of compatible sizes, rem returns the element-wise remainder.
- The sign of the result is the same as the sign of \(a\).

Example:

```matlab
rem(17, 5) % Returns 2
rem(-17, 5) % Returns -2
rem(17, -5) % Returns 2
rem(-17, -5) % Returns -2
```

Use Case:

- When you need to preserve the sign of the dividend in modular operations.

2. mod Function

The mod function computes the modulus, which always returns a non-negative result when the divisor is positive:

```matlab
r = mod(a, b);
```

Behavior:

- The sign of the result depends on the divisor; for a positive divisor, the result is always in [0, b).
- It is particularly useful for wrapping indices, cyclic calculations, and periodic functions.

Example:

```matlab
mod(17, 5) % Returns 2
mod(-17, 5) % Returns 3
mod(17, -5) % Returns -3
mod(-17, -5) % Returns -2
```

Use Case:

- When you need a positive remainder, especially in modular arithmetic and circular data structures.

3. Comparing rem and mod

Aspect	rem	mod
Sign of result	Same as dividend (a)	Same as divisor (b)
Typical use	Sign-sensitive calculations	Non-negative remainders, cyclic indexing
Mathematical basis	Remainder of division with sign consideration	Modular arithmetic, always positive for positive divisor

Example illustrating difference:

```matlab
a = -13;
b = 5;

remResult = rem(a, b); % -13 mod 5 = -3
modResult = mod(a, b); % -13 mod 5 = 2
```

Practical Applications of Remainder Calculations in MATLAB

1. Modular Arithmetic in Cryptography

Cryptography algorithms often rely on modular arithmetic for key generation, encryption, and decryption processes. MATLAB’s mod function is essential in implementing these algorithms, such as RSA:

```matlab
ciphertext = mod(plaintext^e, n);
```

Here, mod ensures the ciphertext stays within a specific numeric range, maintaining the cyclic properties necessary for encryption.

2. Index Wrapping and Circular Data Structures

In data processing, especially with images, signals, or time series, wrapping around indices when they exceed bounds is common:

```matlab
index = mod(currentIndex - 1, totalElements) + 1;
```

This ensures indices cycle correctly, avoiding out-of-bounds errors.

3. Signal Processing and Periodic Functions

Repetition and phase calculations often involve remainders:

```matlab
phaseShift = mod(phase, 2pi);
```

This keeps phase angles within a principal range, such as [0, 2π).

Edge Cases and Common Pitfalls

Handling Negative Numbers

One common mistake is choosing between rem and mod when negative numbers are involved. Remember:

- Use rem if the sign of the remainder should match the dividend.

- Use mod if you want a non-negative result, especially with positive divisors.

Division by Zero

Attempting to perform remainder operations with zero as the divisor results in errors:

```matlab
rem(10, 0) % Error: Division by zero
```

Always validate the divisor before performing such operations.

Matrix and Vector Inputs

Operations involving matrices and vectors are element-wise:

```matlab
a = [10, -10];
b = 3;
rem(a, b) % Returns [1, -1]
```

Ensure dimensions are compatible to avoid broadcasting issues.

Advanced Topics Related to Remainder in MATLAB

1. Remainder and Integer Division

While MATLAB does not have a dedicated integer division operator, functions like idivide facilitate integer division and remainders:

```matlab
[q, r] = idivide(a, b, 'fix'); % Returns quotient and remainder
```

This is useful for applications requiring exact integer division.

2. Remainder in Looping and Algorithm Design

Loop constructs often rely on remainders:

```matlab
for i = 1:N
if rem(i, 2) == 0
% Even index operations
end
end
```

Efficient algorithms often leverage these operations for performance.

3. Remainder in Polynomial and Numerical Computations

In polynomial division or root-finding, remainders determine divisibility and factorization, highlighting the importance of understanding these functions' behavior.

Summary and Best Practices

- Always choose the appropriate function (rem vs mod) based on the sign behavior required.
- Validate input data, especially divisors, to avoid division errors.
- Use mod for cyclic calculations, array indexing, and non-negative remainders.
- Use rem when the sign of the remainder should match the dividend, such as in certain numerical algorithms.
- Be aware of how MATLAB handles element-wise operations with vectors and matrices.

By mastering these functions and understanding their differences and applications, MATLAB users can perform modular arithmetic accurately and efficiently, enabling robust implementations across various scientific and engineering domains.

This comprehensive understanding of MATLAB’s remainder functions equips you with the knowledge to handle a wide array of computational tasks involving division remainders, ensuring precise and predictable results in your programs.

Frequently Asked Questions

What is the MATLAB 'remainder' function and how does it differ from 'mod'?

The MATLAB 'remainder' function computes the remainder after division, similar to 'mod', but differs in its handling of negative numbers. 'remainder' returns the signed remainder, maintaining the sign of the dividend, whereas 'mod' returns a non-negative result when the divisor is positive.

How do I use the 'remainder' function in MATLAB?

You can use the 'remainder' function by specifying two inputs: 'remainder(a, b)'. For example, 'remainder(10, 3)' returns 1, the remainder of 10 divided by 3.

What is the output of 'remainder(-7, 3)' in MATLAB?

The output of 'remainder(-7, 3)' is -1, because 'remainder' maintains the sign of the dividend (-7), giving a signed remainder.

Can 'remainder' be used with matrices in MATLAB?

Yes, 'remainder' can be applied element-wise to matrices and arrays. When used with matrices, it computes the remainder for each corresponding element.

What are common applications of the 'remainder' function in MATLAB?

The 'remainder' function is often used in signal processing, cyclic calculations, and algorithms requiring modular arithmetic, especially when the sign of the dividend matters.

How does 'remainder' handle division by zero in MATLAB?

Dividing by zero using 'remainder' results in a warning and outputs NaN or Inf, depending on the inputs, similar to other arithmetic operations in MATLAB.

Is there a difference between 'remainder' and 'mod' when dealing with negative numbers?

Yes. 'remainder' returns a result with the same sign as the dividend, while 'mod' always returns a non-negative result when the divisor is positive. This affects calculations involving negative operands.

How can I compare the outputs of 'remainder' and 'mod' in MATLAB?

You can compare outputs by executing both functions with the same inputs. For example, 'remainder(-7, 3)' yields -1, whereas 'mod(-7, 3)' yields 2, illustrating their different handling of negatives.

Are there any limitations or considerations when using

'remainder' in MATLAB?

Yes, 'remainder' is primarily designed for situations where the sign of the dividend should be preserved. It may not be suitable for all modular arithmetic applications, especially when non-negative results are needed. Always consider the sign behavior based on your application's requirements.

Additional Resources

Matlab Remainder: A Comprehensive Guide to Understanding and Using the Modulo Operation in MATLAB

When working with numerical data, algorithms, or simulations in MATLAB, understanding how to compute remainders and modular arithmetic is essential. The Matlab remainder function plays a critical role in such calculations, providing a way to determine the leftover value after division. Whether you're dealing with periodic functions, index wrapping, or implementing algorithms that require modular arithmetic, mastering the concept of the Matlab remainder is vital for efficient and accurate programming.

What Is the Matlab Remainder?

The matlab remainder refers to the operation that calculates the leftover part of a division between two numbers. In MATLAB, this operation is primarily performed using the `rem()` function, which returns the remainder after division.

For example:

```matlab
r = rem(10, 3); % r will be 1
```

Here, `10` divided by `3` gives a quotient of `3` with a remainder of `1`, which is what `rem()` returns.

Difference Between `rem()` and `mod()`

While both functions compute remainders, they do so differently, especially regarding the sign of the result when dealing with negative numbers:

Function	Sign of Result with Negative Inputs	Description
`rem()`	Same sign as numerator	Computes the remainder based on the division's actual quotient.
`mod()`	Always non-negative (for positive divisor)	Computes the modulus, which is always non-negative if divisor is positive.

Understanding this distinction is crucial when selecting the appropriate function for your application.

Deep Dive into the `rem()` Function

Syntax and Basic Usage

```matlab
r = rem(a, b)
```

- `a`: Dividend (numerator)
- `b`: Divisor (denominator)
- `r`: Remainder after division

Key Characteristics

- The result `r` has the same sign as `a`.
- The magnitude of `r` is less than the magnitude of `b`.
- Suitable when the sign of the remainder should match the numerator, such as in certain mathematical or physical simulations.

Exploring the Behavior of `rem()` with Negative Numbers

Understanding how `rem()` behaves with negative inputs is crucial. Consider the following examples:

```matlab
rem(10, 3) % Returns 1
rem(-10, 3) % Returns -1
rem(10, -3) % Returns 1
rem(-10, -3) % Returns -1
```

In these cases, the sign of the remainder matches the numerator, regardless of the divisor's sign.

Implications for Programming

- When working with periodic data or wrapping indices, the sign of the remainder can affect logic.
- For positive-only remainders, `mod()` may be more appropriate.

When to Use `rem()` vs. `mod()`

Choosing between `rem()` and `mod()` depends on your specific needs:

| Use Case | Recommended Function | Explanation |

|------------|----------------------|--------------|
| Sign of the remainder should match numerator | `rem()` | Maintains the sign of the dividend, useful in certain mathematical contexts. |
| Remainder should be non-negative | `mod()` | Always produces a non-negative result when divisor is positive, ideal for indexing and periodic calculations. |

Example: Modular Indexing

Suppose you are wrapping indices in an array:

```matlab
index = rem(index, arrayLength);
```

If `index` is negative, `rem()` may produce a negative result, which is invalid for array indexing. Instead, use `mod()`:

```matlab
index = mod(index, arrayLength);
```

This ensures `index` is always within the valid range `[0, arrayLength-1]`.

Practical Applications of the Matlab Remainder

The `rem()` function finds applications across various fields:

1. Signal Processing

- Calculating phase shifts.
- Wrapping time or frequency data.

2. Numerical Algorithms

- Implementing cyclic buffers.
- Index wrapping in arrays during iterative processes.

3. Mathematical Computations

- Solving congruences.
- Simulating periodic phenomena.

4. Computer Graphics

- Tiling textures.
- Repeating patterns.

Step-by-Step Example: Using `rem()` in a MATLAB Script

Suppose you want to simulate wrapping an index around an array:

```matlab
array = [10, 20, 30, 40, 50];
arrayLength = length(array);
indices = -7:7; % Generate indices from -7 to 7

wrappedIndices = rem(indices, arrayLength) + 1; % MATLAB arrays are 1-based

disp('Wrapped indices:')
disp(wrappedIndices)

% Accessing array elements with wrapped indices
values = array(wrappedIndices);
disp('Corresponding array values:')
disp(values)
```

This script:

- Creates a range of indices, including negative ones.
- Wraps indices using `rem()` to ensure they are within the valid range.
- Accesses array elements safely, demonstrating practical use.

Common Pitfalls and Tips

- Negative Divisors: Be cautious with negative divisors, as behavior varies. Use `mod()` if you prefer non-negative results.
- Division by Zero: Both `rem()` and `mod()` will throw an error if the divisor is zero.
- Data Types: Ensure input data types are compatible; floating-point inputs are accepted but beware of floating-point inaccuracies.

Summary and Best Practices

- Use `rem()` when the remainder should have the same sign as the numerator.
- Use `mod()` for non-negative remainders, especially in array indexing or cyclic calculations.
- Always consider the sign implications in your specific application.
- For wrapping indices, prefer `mod()` to avoid negative indices.

Final Thoughts

Mastering the Matlab remainder operation enables you to write more robust, efficient, and mathematically sound code. Whether calculating remainders for algorithmic logic, data wrapping, or periodic functions, understanding the nuances between `rem()` and `mod()` provides you with the flexibility needed to handle diverse computational challenges effectively.

By integrating these concepts into your MATLAB programming toolkit, you'll be better equipped to solve complex problems involving modular arithmetic, periodicity, and data wrapping with confidence and precision.

Matlab Remainder

matlab remainder: Signals, Systems, Transforms, and Digital Signal Processing with MATLAB Michael Corinthios, 2018-09-03 Signals, Systems, Transforms, and Digital Signal Processing with MATLAB® has as its principal objective simplification without compromise of rigor. Graphics, called by the author, the language of scientists and engineers, physical interpretation of subtle mathematical concepts, and a gradual transition from basic to more advanced topics are meant to be among the important contributions of this book. After illustrating the analysis of a function through a step-by-step addition of harmonics, the book deals with Fourier and Laplace transforms. It then covers discrete time signals and systems, the z-transform, continuous- and discrete-time filters, active and passive filters, lattice filters, and continuous- and discrete-time state space models. The author goes on to discuss the Fourier transform of sequences, the discrete Fourier transform, and the fast Fourier transform, followed by Fourier-, Laplace, and z-related transforms, including Walsh–Hadamard, generalized Walsh, Hilbert, discrete cosine, Hartley, Hankel, Mellin, fractional Fourier, and wavelet. He also surveys the architecture and design of digital signal processors, computer architecture, logic design of sequential circuits, and random signals. He concludes with simplifying and demystifying the vital subject of distribution theory. Drawing on much of the author's own research work, this book expands the domains of existence of the most important transforms and thus opens the door to a new world of applications using novel, powerful mathematical tools.

matlab remainder: MATLAB For Dummies Jim Sizemore, John Paul Mueller, 2014-10-20 Plot graphs, solve equations, and write code in a flash! If you work in a STEM field, chances are you'll be using MATLAB on a daily basis. MATLAB is a popular and powerful computational tool and this book provides everything you need to start manipulating and plotting your data. MATLAB has rapidly become the premier data tool, and MATLAB For Dummies is a comprehensive guide to the fundamentals. MATLAB For Dummies guides you through this complex computational language from installation to visualization to automation. Learn MATLAB's language fundamentals including syntax, operators, and data types Understand how to use the most important window in MATLAB – the Command Window Get the basics of linear algebra to get up and running with vectors, matrices, and hyperspace Automate your work with programming scripts and functions Plot graphs in 2D and 3D to visualize your data Includes a handy guide for MATLAB's functions and plotting routines MATLAB is an essential part of the analysis arsenal and MATLAB For Dummies provides clear, thorough guidance to get the most out of your data.

matlab remainder: Matlab Dorothy C. Attaway, 2016-07-23 MATLAB: A Practical Introduction to Programming and Problem Solving, Fourth Edition, winner of a 2017 Textbook Excellence Award (Texty), has been updated to reflect the functionality of the current version of MATLAB, including the new H2 Graphics system. It features new and revised end-of-chapter exercises, more engineering applications to help the reader learn this software tool in context, and a new section on object-oriented programming in MATLAB. MATLAB has become the standard software tool for solving scientific and engineering problems due to its powerful built-in functions and its ability to program. Assuming no knowledge of programming, this book guides the reader through both programming and built-in functions to easily exploit MATLAB's extensive capabilities for tackling engineering problems. The book starts with programming concepts, such as variables, assignments, and selection statements, moves on to loops, and then solves problems using both the programming concept and the power of MATLAB. In-depth coverage is given to input/output, a topic fundamental to many engineering applications. - Winner of a 2017 Textbook Excellence Award (Texty) from the Textbook and Academic Authors Association - Presents programming concepts and MATLAB built-in functions side-by-side - Offers a systematic, step-by-step approach, building on concepts throughout the book and facilitating easier learning - Includes sections on common pitfalls and programming guidelines to direct students toward best practices - Combines basic programming concepts, built-in functions, and advanced topics for problem solving with MATLAB to make this book uniquely suitable for a wide range of courses teaching or using MATLAB across the curriculum

matlab remainder: Numerical Methods with Chemical Engineering Applications Kevin D. Dorfman, Prodromos Daoutidis, 2017-01-11 This undergraduate textbook integrates the teaching of numerical methods and programming with problems from core chemical engineering subjects.

matlab remainder: Digital Dice Paul J. Nahin, 2013-03-24 Some probability problems are so difficult that they stump the smartest mathematicians. But even the hardest of these problems can often be solved with a computer and a Monte Carlo simulation, in which a random-number generator simulates a physical process, such as a million rolls of a pair of dice. This is what Digital Dice is all about: how to get numerical answers to difficult probability problems without having to solve complicated mathematical equations. Popular-math writer Paul Nahin challenges readers to solve twenty-one difficult but fun problems, from determining the odds of coin-flipping games to figuring out the behavior of elevators. Problems build from relatively easy (deciding whether a dishwasher who breaks most of the dishes at a restaurant during a given week is clumsy or just the victim of randomness) to the very difficult (tackling branching processes of the kind that had to be solved by Manhattan Project mathematician Stanislaw Ulam). In his characteristic style, Nahin brings the problems to life with interesting and odd historical anecdotes. Readers learn, for example, not just how to determine the optimal stopping point in any selection process but that astronomer Johannes Kepler selected his second wife by interviewing eleven women. The book shows readers how to write elementary computer codes using any common programming language, and provides solutions and line-by-line walk-throughs of a MATLAB code for each problem. Digital Dice will appeal to anyone who enjoys popular math or computer science. In a new preface, Nahin wittily addresses some of the responses he received to the first edition.

matlab remainder: Languages and Compilers for Parallel Computing Keith Cooper, John Mellor-Crummey, Vivek Sarkar, 2011-02-24 This book constitutes the thoroughly refereed post-proceedings of the 23rd International Workshop on Languages and Compilers for Parallel Computing, LCPC 2010, held in Houston, TX, USA, in October 2010. The 18 revised full papers presented were carefully reviewed and selected from 47 submissions. The scope of the workshop spans foundational results and practical experience, and targets all classes of parallel platforms including concurrent, multithreaded, multicore, accelerated, multiprocessor, and cluster systems

matlab remainder: Real-Time Digital Signal Processing from MATLAB® to C with the TMS320C6x DSPs, Second Edition Thad B. Welch, Cameron H.G. Wright, Michael G. Morrow, 2011-12-22 From the Foreword: ...There are many good textbooks today to teach digital signal processing, but most of them are content to teach the theory, and perhaps some MATLAB® simulations. This book has taken a bold step forward. It not only presents the theory, it reinforces it with simulations, and then it shows us how to actually use the results in real-time applications. This last step is not a trivial step, and that is why so many books, and courses, present only theory and simulations. With the combined expertise of the three authors of this text...the reader can step into the real-time world of applications with a text that presents an accessible path... —Delores M. Etter, Texas Instruments Distinguished Chair in Electrical Engineering and Executive Director, Caruth

Institute for Engineering Education, Southern Methodist University, Dallas, Texas, USA Mastering practical application of real-time digital signal processing (DSP) remains one of the most challenging and time-consuming pursuits in the field. It is even more difficult without a resource to bridge the gap between theory and practice. Filling that void, Real-Time Digital Signal Processing from MATLAB® to C with the TMS320C6x DSPs, Second Edition is organized in three sections that cover enduring fundamentals and present practical projects and invaluable appendices. This updated edition gives readers hands-on experience in real-time DSP using a practical, step-by-step framework that also incorporates demonstrations, exercises, and problems, coupled with brief overviews of applicable theory and MATLAB® application. Engineers, educators, and students rely on this book for precise, simplified instruction on use of real-time DSP applications. The book's software supports the latest high-performance hardware, including the powerful, inexpensive, and versatile OMAP-L138 Experimenter Kit and other development boards. Incorporating readers' valuable feedback and suggestions, this installment covers additional topics (such as PN sequences) and more advanced real-time DSP projects (including higher-order digital communications projects), making it even more valuable as a learning tool.

matlab remainder: Signal Processing Techniques for Communication K.C. Raveendranathan, 2024-10-30 The reference text discusses signal processing tools and techniques used for the design, testing, and deployment of communication systems. It further explores software simulation and modeling tools like MATLAB, GNU Octave, Mathematica, and Python for modeling, simulation, and detailed analysis leading to comprehensive insights into communication systems. The book explains topics such as source coding, pulse demodulation systems, and the principle of sampling and aliasing. This book: Discusses modern techniques including analog and digital filter design, and modulation principles including quadrature amplitude modulation, and differential phase shift keying. Covers filter design using MATLAB, system simulation using Simulink, signal processing toolbox, linear time-invariant systems, and non-linear time-variant systems. Explains important pulse keying techniques including Gaussian minimum shift keying and quadrature phase shift keying. Presents signal processing tools and techniques for communication systems design, modeling, simulation, and deployment. Illustrates topics such as software-defined radio (SDR) systems, spectrum sensing, and automated modulation sensing. The text is primarily written for senior undergraduates, graduate students, and academic researchers in the fields of electrical engineering, electronics and communication engineering, computer science, and engineering.

matlab remainder: Real-Time Digital Signal Processing from MATLAB to C with the TMS320C6x DSPs Thad B. Welch, Cameron H.G. Wright, Michael G. Morrow, 2016-12-19 This updated edition gives readers hands-on experience in real-time DSP using a practical, step-by-step framework that also incorporates demonstrations, exercises, and problems, coupled with brief overviews of applicable theory and MATLAB applications. Organized in three sections that cover enduring fundamentals and present practical projects and invaluable appendices, this new edition provides support for the most recent and powerful of the inexpensive DSP development boards currently available from Texas Instruments: the OMAP-L138 LCDK. It includes two new real-time DSP projects, as well as three new appendices: an introduction to the Code Generation tools available with MATLAB, a guide on how to turn the LCDK into a portable battery-operated device, and a comparison of the three DSP boards directly supported by this edition.

matlab remainder: A Guide to MATLAB Brian R. Hunt, Ronald L. Lipsman, Jonathan M. Rosenberg, Kevin R. Coombes, John E. Osborn, Garrett J. Stuck, 2006-06-08 This is a short, focused introduction to MATLAB, a comprehensive software system for mathematical and technical computing. It contains concise explanations of essential MATLAB commands, as well as easily understood instructions for using MATLAB's programming features, graphical capabilities, simulation models, and rich desktop interface. Written for MATLAB 7, it can also be used with earlier (and later) versions of MATLAB. This book teaches how to graph functions, solve equations, manipulate images, and much more. It contains explicit instructions for using MATLAB's companion software, Simulink, which allows graphical models to be built for dynamical systems. MATLAB's new publish feature is discussed, which allows mathematical computations to be combined with text and graphics, to produce polished, integrated, interactive documents. For the beginner it explains everything needed to start using MATLAB, while experienced users making the switch to MATLAB 7 from an earlier version will also find much useful information here.

matlab remainder: An Introduction to High-performance Scientific Computing , 1996 Designed for undergraduates, An Introduction to High-Performance Scientific Computing assumes a basic knowledge of numerical computation and proficiency in Fortran or C programming and can be used in any science, computer science, applied mathematics, or engineering department or by practicing scientists and engineers, especially those associated with one of the national laboratories or supercomputer centers. This text evolved from a new curriculum in scientific computing that was developed to teach undergraduate science and engineering majors how to use high-performance computing systems (supercomputers) in scientific and engineering applications. Designed for undergraduates, An Introduction to High-Performance Scientific Computing assumes a basic knowledge of numerical computation and proficiency in Fortran or C programming and can be used in any science, computer science, applied mathematics, or engineering department or by practicing scientists and engineers, especially those associated with one of the national laboratories or supercomputer centers. The authors begin with a survey of scientific computing and then provide a review of background (numerical analysis, IEEE arithmetic, Unix, Fortran) and tools (elements of MATLAB, IDL, AVS). Next, full coverage is given to scientific visualization and to the architectures (scientific workstations and vector and parallel supercomputers) and performance evaluation needed to solve large-scale problems. The concluding section on applications includes three problems (molecular dynamics, advection, and computerized tomography) that illustrate the challenge of solving problems on a variety of computer architectures as well as the suitability of a particular architecture to solving a particular problem. Finally, since this can only be a hands-on course with extensive programming and experimentation with a variety of architectures and programming paradigms, the authors have provided a laboratory manual and supporting software via anonymous ftp. Scientific and Engineering Computation series

matlab remainder: Computer Programming and Numerical Methods Mr. Rohit Manglik, 2023-06-23 Introduces programming and numerical methods for solving engineering problems, emphasizing computational efficiency.

matlab remainder: Stochastic Approaches for Systems Biology Mukhtar Ullah, Olaf Wolkenhauer, 2011-07-12 This textbook focuses on stochastic analysis in systems biology containing both the theory and application. While the authors provide a review of probability and random variables, subsequent notions of biochemical reaction systems and the relevant concepts of probability theory are introduced side by side. This leads to an intuitive and easy-to-follow presentation of stochastic framework for modeling subcellular biochemical systems. In particular, the authors make an effort to show how the notion of propensity, the chemical master equation and the stochastic simulation algorithm arise as consequences of the Markov property. The text contains many illustrations, examples and exercises to illustrate the ideas and methods that are introduced. Matlab code is also provided where appropriate. Additionally, the cell cycle is introduced as a more complex case study. Senior undergraduate and graduate students in mathematics and physics as well as researchers working in the area of systems biology, bioinformatics and related areas will find this text useful.

matlab remainder: *BASIC ELECTRICAL AND ELECTRONICS ENGINEERING* Dr. Jaikaran Singh,Prof. Vikas Pandey, 2020-01-01 This book provides an overview of the basics of electrical and electronic engineering that are required at the undergraduate level. Efforts have been taken to keep the complexity level of the subject to bare minimum so that the students of non-electrical/electronics can easily understand the basics. It offers an unparalleled exposure to the entire gamut of topics such as Electricity Fundamentals, Network Theory, Electro-magnetism, Electrical Machines, Transformers, Measuring Instruments, Power Systems, Semiconductor Devices, Digital Electronics and Integrated Circuits.

matlab remainder: *Discrete and Continuous Fourier Transforms* Eleanor Chu, 2008-03-19 Long employed in electrical engineering, the discrete Fourier transform (DFT) is now applied in a range of fields through the use of digital computers and fast Fourier transform (FFT) algorithms. But to correctly interpret DFT results, it is essential to understand the core and tools of Fourier analysis. Discrete and Continuous Fourier Transform

matlab remainder: Channel Coding Techniques for Wireless Communications K. Deergha Rao, 2019-11-22 This book discusses the latest channel coding techniques, MIMO systems, and 5G channel coding evolution. It provides a comprehensive overview of channel coding, covering modern techniques such as turbo codes, low-density parity-check (LDPC) codes, space–time coding, polar codes, LT codes, and Raptor codes as well as the traditional codes such as cyclic codes, BCH, RS codes, and convolutional codes. It also explores MIMO communications, which is an effective method for high-speed or high-reliability wireless communications. It also examines the evolution of 5G channel coding techniques. Each of the 13 chapters features numerous illustrative examples for easy understanding of the coding techniques, and MATLAB-based programs are integrated in the text to enhance readers' grasp of the underlying theories. Further, PC-based MATLAB m-files for illustrative examples are included for students and researchers involved in advanced and current concepts of coding theory.

matlab remainder: Web-Based Control and Robotics Education Spyros G. Tzafestas, 2009-07-31 For the things we have to learn before we can do them, we learn by doing them. Aristotle Teaching should be such that what is offered is perceived as a valuable gift and not as a hard duty. Albert Einstein The second most important job in the world, second only to being a good parent, is being a good teacher. S.G. Ellis The fast technological changes and the resulting shifts of market conditions require the development and use of educational methodologies and opportunities with moderate economic demands. Currently, there is an increasing number of edu- tional institutes that respond to this challenge through the creation and adoption of distance education programs in which the teachers and students are separated by physical distance. It has been verified in many cases that, with the proper methods and tools, teaching and learning at a distance can be as effective as traditional fa- to-face instruction. Today, distance education is primarily performed through the Internet, which is the biggest and most powerful computer network of the World, and the World Wide Web (WWW), which is an effective front-end to the Internet and allows the Internet users to uniformly access a large repertory of resources (text, data, images, sound, video, etc.) available on the Internet.

matlab remainder: Electromechanical Systems, Electric Machines, and Applied Mechatronics Sergey Edward Lyshevski, 2018-02-06 Recent trends in engineering show increased emphasis on integrated analysis, design, and control of advanced electromechanical systems, and their scope continues to expand. Mechatronics-a breakthrough concept-has evolved to attack, integrate, and solve a variety of emerging problems in engineering, and there appears to be no end to its application. It has become essential for all engineers to understand its basic theoretical standpoints and practical applications. Electromechanical Systems, Electric Machines, and Applied Mechatronics presents a unique combination of traditional engineering topics and the latest technologies, integrated to stimulate new advances in the analysis and design of state-of-the-art electromechanical systems. With a focus on numerical and analytical methods, the author develops the rigorous theory of electromechanical systems and helps build problem-solving skills. He also stresses simulation as a critical aspect of developing and prototyping advanced systems. He uses the MATLABTM environment for his examples and includes a MATLABTM diskette with the book, thus providing a solid introduction to this standard engineering tool. Readable, interesting, and accessible, Electromechanical Systems, Electric Machines, and Applied Mechatronics develops a thorough understanding of the integrated perspectives in the design and analysis of electromechanical systems. It covers the basic concepts in mechatronics, and with numerous worked examples, prepares the reader to use the results in engineering practice. Readers who master this book will know what they are doing, why they are doing it, and how to do it.

matlab remainder: Parallel Computing: On the Road to Exascale G.R. Joubert, H. Leather, M. Parsons, 2016-04-28 As predicted by Gordon E. Moore in 1965, the performance of computer processors increased at an exponential rate. Nevertheless, the increases in computing speeds of single processor machines were eventually curtailed by physical constraints. This led to the development of parallel computing, and whilst progress has been made in this field, the complexities of parallel algorithm design, the deficiencies of the available software development tools and the complexity of scheduling tasks over thousands and even millions of processing nodes represent a major challenge to the construction and use of more powerful parallel systems. This book presents the proceedings of the biennial International Conference on Parallel Computing (ParCo2015), held in Edinburgh, Scotland, in September 2015. Topics covered include computer architecture and performance, programming models and methods, as well as applications. The book also includes two invited talks and a number of mini-symposia. Exascale computing holds enormous promise in terms of increasing scientific knowledge acquisition and thus contributing to the future well-being and prosperity of mankind. A number of innovative approaches to the development and use of future high-performance and high-throughput systems are to be found in this book, which will be of interest to all those whose work involves the handling and processing of large amounts of data.

matlab remainder: Design and Analysis of Control Systems Arthur G.O. Mutambara, 2017-12-14 Written to inspire and cultivate the ability to design and analyze feasible control algorithms for a wide range of engineering applications, this comprehensive text covers the theoretical and practical principles involved in the design and analysis of control systems. From the development of the mathematical models for dynamic systems, the author shows how they are used to obtain system response and facilitate control, then addresses advanced topics, such as digital control systems, adaptive and robust control, and nonlinear control systems.

Related to matlab remainder

rem - Remainder after division - MATLAB - MathWorks Find the remainder after division for a set of integers including both positive and negative values. Note that nonzero results have the same sign as the dividend

mod - Remainder after division (modulo operation) - MATLAB Find the remainder after division for a set of integers including both positive and negative values. Note that nonzero results are always positive if the divisor is positive

rem - Remainder after division - MATLAB - MathWorks Find the remainder after division in case the dividend is a rational number, and the divisor is an integer. Find the remainder after division for these numbers

Difference between mod and rem functions - MathWorks The MATLAB documentation states that "The concept of remainder after division is not uniquely defined, and the two functions mod and rem each compute a different variation.

Arithmetic Operations - MATLAB & Simulink - MathWorks MATLAB supports 1-, 2-, 4-, and 8-byte storage for integer data. If you use the smallest integer type that accommodates your data, you can save memory and program execution time

Check usage of remainder and reciprocal operations - MathWorks Description This check identifies uses of remainder and reciprocal operations that cause non-finite results. This check performs SLDV analysis on the model. Running this check triggers an

mod - Symbolic modulus after division - MATLAB - MathWorks Description m = mod(a,b) finds the modulus after division. To find the remainder, use rem. If a is a polynomial expression, then mod(a,b) returns the unevaluated modulus of the polynomial

mod - □□□□ □□□ (□□□ □□) - MATLAB - MathWorks MATLAB®□ backgroundPool□ □□□ □□□□□□□ □□□ □□□□□ Parallel Computing Toolbox™□ ThreadPool□ □□□ □□ □□ □□□ □□ □ □□□□

How to get the remainder of a division in Simulink? - MathWorks How to get the remainder of a division in Learn more about divide, simulink, remainder, rem

Least-squares deconvolution and polynomial division - MATLAB This MATLAB function deconvolves a vector h out of a vector y using polynomial long division, and returns the quotient x and remainder r such that y = conv (x,h) + r

rem - Remainder after division - MATLAB - MathWorks Find the remainder after division for a set of integers including both positive and negative values. Note that nonzero results have the same sign as the dividend

mod - Remainder after division (modulo operation) - MATLAB Find the remainder after division for a set of integers including both positive and negative values. Note that nonzero results are always positive if the divisor is positive

rem - Remainder after division - MATLAB - MathWorks Find the remainder after division in case the dividend is a rational number, and the divisor is an integer. Find the remainder after division for these numbers

Difference between mod and rem functions - MathWorks The MATLAB documentation states that "The concept of remainder after division is not uniquely defined, and the two functions mod and rem each compute a different variation.

Arithmetic Operations - MATLAB & Simulink - MathWorks MATLAB supports 1-, 2-, 4-, and 8-byte storage for integer data. If you use the smallest integer type that accommodates your data, you can save memory and program execution time

Check usage of remainder and reciprocal operations - MathWorks Description This check identifies uses of remainder and reciprocal operations that cause non-finite results. This check performs SLDV analysis on the model. Running this check triggers an

mod - Symbolic modulus after division - MATLAB - MathWorks Description m = mod(a,b) finds the modulus after division. To find the remainder, use rem. If a is a polynomial expression, then mod(a,b) returns the unevaluated modulus of the polynomial

mod - □□□□ □□□ (□□□ □□) - MATLAB - MathWorks MATLAB®□ backgroundPool□ □□□ □□□□□□□□ □□□ □□□□□ Parallel Computing Toolbox™□ ThreadPool□ □□□ □□ □□ □□□ □□ □ □□□□

How to get the remainder of a division in Simulink? - MathWorks How to get the remainder of a division in Learn more about divide, simulink, remainder, rem

Least-squares deconvolution and polynomial division - MATLAB This MATLAB function deconvolves a vector h out of a vector y using polynomial long division, and returns the quotient x and remainder r such that y = conv (x,h) + r

rem - Remainder after division - MATLAB - MathWorks Find the remainder after division for a set of integers including both positive and negative values. Note that nonzero results have the same sign as the dividend

mod - Remainder after division (modulo operation) - MATLAB Find the remainder after division for a set of integers including both positive and negative values. Note that nonzero results are always positive if the divisor is positive

rem - Remainder after division - MATLAB - MathWorks Find the remainder after division in case the dividend is a rational number, and the divisor is an integer. Find the remainder after division for these numbers

Difference between mod and rem functions - MathWorks The MATLAB documentation states that "The concept of remainder after division is not uniquely defined, and the two functions mod and rem each compute a different variation.

Arithmetic Operations - MATLAB & Simulink - MathWorks MATLAB supports 1-, 2-, 4-, and 8-byte storage for integer data. If you use the smallest integer type that accommodates your data, you can save memory and program execution time

Check usage of remainder and reciprocal operations - MathWorks Description This check identifies uses of remainder and reciprocal operations that cause non-finite results. This check performs SLDV analysis on the model. Running this check triggers an

mod - Symbolic modulus after division - MATLAB - MathWorks Description m = mod(a,b) finds the modulus after division. To find the remainder, use rem. If a is a polynomial expression, then mod(a,b) returns the unevaluated modulus of the polynomial

mod - □□□□ □□□ (□□□ □□) - MATLAB - MathWorks MATLAB®□ backgroundPool□ □□□ □□□□□□□ □□□ □□□□□ Parallel Computing Toolbox™□ ThreadPool□ □□□ □□ □□ □□□ □□ □ □□□□
How to get the remainder of a division in Simulink? - MathWorks How to get the remainder of a division in Learn more about divide, simulink, remainder, rem
Least-squares deconvolution and polynomial division - MATLAB This MATLAB function deconvolves a vector h out of a vector y using polynomial long division, and returns the quotient x and remainder r such that y = conv (x,h) + r
rem - Remainder after division - MATLAB - MathWorks Find the remainder after division for a set of integers including both positive and negative values. Note that nonzero results have the same sign as the dividend
mod - Remainder after division (modulo operation) - MATLAB Find the remainder after division for a set of integers including both positive and negative values. Note that nonzero results are always positive if the divisor is positive
rem - Remainder after division - MATLAB - MathWorks Find the remainder after division in case the dividend is a rational number, and the divisor is an integer. Find the remainder after division for these numbers
Difference between mod and rem functions - MathWorks The MATLAB documentation states that "The concept of remainder after division is not uniquely defined, and the two functions mod and rem each compute a different variation.
Arithmetic Operations - MATLAB & Simulink - MathWorks MATLAB supports 1-, 2-, 4-, and 8-byte storage for integer data. If you use the smallest integer type that accommodates your data, you can save memory and program execution time
Check usage of remainder and reciprocal operations - MathWorks Description This check identifies uses of remainder and reciprocal operations that cause non-finite results. This check performs SLDV analysis on the model. Running this check triggers an
mod - Symbolic modulus after division - MATLAB - MathWorks Description m = mod(a,b) finds the modulus after division. To find the remainder, use rem. If a is a polynomial expression, then mod(a,b) returns the unevaluated modulus of the polynomial
mod - □□□□ □□□ (□□□ □□) - MATLAB - MathWorks MATLAB®□ backgroundPool□ □□□ □□□□□□□ □□□ □□□□□ Parallel Computing Toolbox™□ ThreadPool□ □□□ □□ □□ □□□ □□ □ □□□□
How to get the remainder of a division in Simulink? - MathWorks How to get the remainder of a division in Learn more about divide, simulink, remainder, rem
Least-squares deconvolution and polynomial division - MATLAB This MATLAB function deconvolves a vector h out of a vector y using polynomial long division, and returns the quotient x and remainder r such that y = conv (x,h) + r
rem - Remainder after division - MATLAB - MathWorks Find the remainder after division for a set of integers including both positive and negative values. Note that nonzero results have the same sign as the dividend
mod - Remainder after division (modulo operation) - MATLAB Find the remainder after division for a set of integers including both positive and negative values. Note that nonzero results are always positive if the divisor is positive
rem - Remainder after division - MATLAB - MathWorks Find the remainder after division in case the dividend is a rational number, and the divisor is an integer. Find the remainder after division for these numbers
Difference between mod and rem functions - MathWorks The MATLAB documentation states that "The concept of remainder after division is not uniquely defined, and the two functions mod and rem each compute a different variation.
Arithmetic Operations - MATLAB & Simulink - MathWorks MATLAB supports 1-, 2-, 4-, and 8-byte storage for integer data. If you use the smallest integer type that accommodates your data, you can save memory and program execution time
Check usage of remainder and reciprocal operations - MathWorks Description This check

identifies uses of remainder and reciprocal operations that cause non-finite results. This check performs SLDV analysis on the model. Running this check triggers an
mod - Symbolic modulus after division - MATLAB - MathWorks Description m = mod(a,b) finds the modulus after division. To find the remainder, use rem. If a is a polynomial expression, then mod(a,b) returns the unevaluated modulus of the polynomial
mod - □□□□ □□□ (□□□ □□) - MATLAB - MathWorks MATLAB®□ backgroundPool□ □□□ □□□□□□□ □□□ □□□□□ Parallel Computing Toolbox™□ ThreadPool□ □□□ □□ □□ □□□ □□ □ □□□□
How to get the remainder of a division in Simulink? - MathWorks How to get the remainder of a division in Learn more about divide, simulink, remainder, rem
Least-squares deconvolution and polynomial division - MATLAB This MATLAB function deconvolves a vector h out of a vector y using polynomial long division, and returns the quotient x and remainder r such that y = conv (x,h) + r

Related Articles

- quitclaim deed oklahoma
- blank birth certificate template
- guia deportiva pdf

Back to Home