|  |
| --- |
| Matrices |
| Objectives | | **06 Jan 2025** | |
| * review linear algebra concepts and terminology * set up and numerically solve linear systems in Julia * create and manipulate matrices in Julia   1. transposes   2. sub-blocks   3. sparse matrices * recognize an orthogonal matrix and describe its properties * compute coordinates using a basis, in particular an orthogonal basis * compute lengths of vectors and find angles of vectors in higher dimensions | | | |
| Linear Algebra Review | | |  |
| Linear algebra provides a way of compactly representing and operating on sets of linear equations. Stanford has a 26-page online linear algebra for machine learning review guide that I recommend. <https://cs229.stanford.edu/section/cs229-linalg.pdf>  the following data and screenshots are taken from the stanford.edu pdf | | | |
| Basic Concept | represent and operate on sets of linear equations  starting with:    convert to the form:    to arrive at: | | |
| Notation | : a matrix with m rows and m columns  : a vector with n entries  : the transpose of x  : value ‘A’ at the ith row and the jth column | | |

|  |  |  |
| --- | --- | --- |
| Matrix Multiplication | |  |
| Definition | for the **matrix** **product** to exist, the number of **columns** in **A** must **equal** the number of **rows** in **B** | |
| Dot Product  (Inner Product) | **dot** **products** are special cases of matrix multiplication where   * **output** is a **scalar** | |
| Outer Product | * **output** is an 𝑚×𝑛 **matrix** | |
| Matrix-Vector Multiplication | * ‘**y’** is a **linear** **combination** of the **columns** of ‘A’, where the **coefficients** of the linear combination are given by the **entries** **of** ‘**x’** | |
| Matrix-Matrix Multiplication | **where**, | |
| Properties of Matrix Multiplication |  | |
| Operations and Properties | |  |
| Diagonal Matrix |  | |
| Identity Matrix | a **special** **case** of the **diagonal** **matrix** where all non-zero values = **1** | |
| Transpose | **properties** of transpose: | |

|  |  |
| --- | --- |
| Symmetric Matrices | any square matrix can be represented as a sum of a symmetric matrix and anti-symmetric matrix where the first matrix is symmetric and the second is antisymmetric    the set of all symmetric matrices of size *n* is denoted as |
| Trace | **properties** of **trace**: |
| Norms |  |
| Linear Dependance |  |
| Inverse |  |
| Orthogonal Matrices |  |
| Span |  |
| Range |  |
| Nullspace |  |
| Determinant | these **three** **properties** lend to these **properties**: |
| Quadratic Form | **implies**: |

|  |  |  |  |
| --- | --- | --- | --- |
| Eigenvalues and Eigenvectors |  | | |
| The Gradient |  | | |
| The Hessian |  | | |
| Gradients of the Determinant |  | | |
| The Lagrangian |  | | |
| Async Materials | |  | |
| Matrices | | | |
| Linear Transformation |  | | |
| Gaussian Elimination |  | | |
| Solution with Inverse |  | | |
| Numerics | | | |
| Numerical Software |  | | |
| Matrix Operations | | | |
| Transpose |  | | |
| Transpose and Product |  | | |
| Row and Column Vectors |  | | |
| Dot Product |  | | |
| Sub-Blocks |  | | |
| Products of Block Matrices |  | | |
| Example |  | | |
| Angles and Orthogonal Matrices | | |
| Length and Angles |  | | |
| Rotation in 2-D |  | | |
| Angles and Orthogonal Matrices |  | | |
| 2-D Geometric View |  | | |

|  |  |
| --- | --- |
| Coordinates | |
| Definition |  | |
| Coordinate Transform |  | |
| Subspaces |  | |
| Subspaces and Coordinates |  | |
| Finding the Coordinate |  | |

|  |  |
| --- | --- |
| Live Session Notes | 06 Jan 2026 |

|  |  |  |
| --- | --- | --- |
| Instructor | Joseph Slagel (Tanner) | |
| Email | [slagel@unc.edu](mailto:slagel@unc.edu) | |
| Website | <https://shemesh.larc.nasa.gov/people/jts/> << NASA!! | |
| Office Hours | Friday at 12:00 pm | |
| What to expect | |  |

|  |
| --- |
| * “a really fun class” * Julia programming language * assignments due on Sunday |

|  |  |
| --- | --- |
| Julia Programming Language |  |

|  |  |
| --- | --- |
| What is Julia? | * an open-source, multi-platform, high-level, high-performance   programming language for technical computing.   * an LLVM-based JIT compiler that allows it to match the performance   of languages such as C and FORTRAN without the hassle of low-level code   * dynamically typed, provides multiple dispatches, and is designed for   parallelism and distributed computation.   * many built-in mathematical functions, including special functions   (e.g. Gamma) and supports complex numbers right out of the box.   * generates code automagically thanks to Lisp-inspired macros. |
| Julia Commands | |
| Accessing Help |  |
| Comments |  |
| Information About Objects |  |
| Using Packages |  |
| The Working Directory |  |
| Arithmetic Operators |  |
| Assignment Operators |  |
| Numeric Comparison Operators |  |
| Logical Operators |  |
| Other Operators |  |
| Creating Vectors |  |
| Vector Functions |  |
| Selecting Vector Elements |  |
| Characters and Strings |  |
| Combining and Splitting Strings |  |
| Finding and Mutating Strings |  |
| Defining DataFrames |  |
| Manipulating DataFrames |  |