Chapter 1
Introduction to Linear Algebra

ABSTRACT

This chapter introduces widely used concepts about linear algebra in computer science, as well as information about the standard libraries that gather kernels for linear algebra operations, such as the basic linear algebra subprograms (BLAS) and the linear algebra package (LAPACK). The creation and evolution of these libraries is historically contextualized to help the reader understand their relevance and utility. Moreover, dense and sparse linear algebra are explained. The authors describe the levels of the BLAS library, the motivation behind the hierarchical structure of the BLAS library, and its connection with the LAPACK library. The authors also provide a detailed introduction on some of the most used and popular dense linear algebra kernels or routines, such as GEMM (matrix-matrix multiplication), TRSM (triangular solver), GETRF (LU factorization), and GESV (LU solve). Finally, the authors focus on the most important sparse linear algebra routines and the motivation behind the discussed approaches.

INTRODUCTION

A wide variety of engineering and scientific applications (Vogel et al., 2009) (McCraney et al., 2020) (Plimpton, 1995) rely on linear algebra to address fundamental problems such as computing eigenvalues or solving systems of linear equations. These problems are present in fields such as automatic control, chemistry simulations, or aerodynamic calculations. At the same time, solving these types of problems frequently implies the use of basic linear

DOI: 10.4018/978-1-7998-7082-1.ch001

algebra operations, e. g. the matrix-matrix multiplication. In this context, the Basic Linear Algebra Subprograms (BLAS) specification defines a set of low-level routines to perform common Dense Linear Algebra (DLA) operations. Moreover, the interface of each routine included in the library and their functionalities are specified. The majority of nowadays companies', researchers' and engineers works rely on applications that compute linear algebra operations. The main reason behind this is the fact that the data used by those applications is stored in memory quantitatively and occupying positions of memory in such a way that, organized in an appropriate manner, forms lists of elements that can be understood as vectors or matrices (commonly known in Computer Science as arrays).

In this Chapter the authors will first provide a summary of the main linear algebra definitions required to understand the basics of the operations that form current engineering applications. Then BLAS and LAPACK libraries will be presented, as they implementations of the linear algebra operations are widely used all over the engineering environments. Lately, some insights will be provided about sparse linear algebra in Computer Science.

INTRODUCTION TO BASIC LINEAR ALGEBRA

Naturally, each of the linear algebra operations involved in today's engineering applications have different computational requirements and complexities, and it is necessary to understand that before proceeding to analyze how the most common linear algebra operations in computer science can be optimized through task-parallelism.

In this section the authors present the basic linear algebra terms and operations to which they will refer in later chapters. Note that the authors will base all the definitions in Real numbers (R).

Basic Terminology

As it has already been mentioned, in most engineering applications it is common to see the data stored in memory forming structures that can be understood as arrays.

Definition 1. (Array) An array is a structure formed by a collection of elements organized in such a way that they can be indexed, this is, each of the elements can be associated to an index that represents its position in the array with respect to the rest of the elements.

Related Content

Python's Role in Predicting Type 2 Diabetes Using Insulin DNA Sequence
Aswathi Sasidharanand N. Arulkumar (2023). *Advanced Applications of Python Data Structures and Algorithms (pp. 266-278).*
www.irma-international.org/chapter/pythons-role-in-predicting-type-2-diabetes-using-insulin-dna-sequence/326089

Computer Architectures and Programming Models: How to Exploit Parallelism
(2023). *Developing Linear Algebra Codes on Modern Processors: Emerging Research and Opportunities (pp. 26-46).*
www.irma-international.org/chapter/computer-architectures-and-programming-models/313453

Introduction
(2021). *MATLAB® With Applications in Mechanics and Tribology (pp. 1-7).*
www.irma-international.org/chapter/introduction/276280

Hierarchical Matrices: A Context in Which New Features Are Vital
(2023). *Developing Linear Algebra Codes on Modern Processors: Emerging Research and Opportunities (pp. 187-227).*
www.irma-international.org/chapter/hierarchical-matrices/313458

The ODE- and BVP-Solvers With M&T Applications
(2021). *MATLAB® With Applications in Mechanics and Tribology (pp. 166-190).*
www.irma-international.org/chapter/the-ode--and-bvp-solvers-with-mt-applications/276285