

# Preface

This book is intended to serve as a practical guide for the development and implementation of numerical algorithms on Graphics Processing Units (GPUs). The book assumes that the reader is familiar with the mathematical context and has a good working knowledge of GPU architecture and its programming sufficient to translate specialized mathematical algorithms and pseudo-codes presented in the book into a fully functional CUDA or OpenCL software. In case the reader is not familiar with the GPU programming, the reader is directed to other sources, such as NVIDIA's *CUDA Parallel Computing Platform* website, for low-level programming details, tools, and techniques prior to reading this book.

## Book Focus

The main focus of this book is on the efficient implementation of numerical methods on GPUs. The book chapters are written by the leaders in the field working for many years on the development and implementation of computationally intensive numerical algorithms for solving scientific computing and engineering problems.

It is widely understood and accepted that modern scientific discovery in all of the disciplines requires extensive computations. It is also the case that modern engineering heavily utilizes advanced computational models and tools. At the heart of many such computations are libraries of mathematical codes for solving systems of linear equations, computing solutions of differential equations, finding integrals and function values, transforming time series, etc. These libraries have been developed over several decades and have been constantly updated to track the ever changing architecture and capabilities of computing hardware. With the introduction of GPUs, many of the existing numerical libraries are currently undergoing another phase of transformation in order to continue serving the computational science and engineering community by providing the required level of performance. Simultaneously, new numerical methods are under development to take advantage of the revolutionary architecture of GPUs. In either case, the developers of such

numerical codes face the challenge of extracting parallelism present in numerical methods and expressing it in the form that can be successfully utilized by the massively parallel GPU architecture. This frequently requires reformulating the original algorithmic structure of the code, tuning its performance, and developing and validating entirely new algorithms that can take advantage of the new hardware. It is my hope that this book will serve as a reference implementation and will provide the guidance for the developers of such codes by presenting a collective experience from many recent successful efforts.

## **Audience and Organization**

This book targets practitioners working on the implementation of numerical codes on GPUs, researchers and software developers attempting to extend existing numerical libraries to GPUs, and readers interested in all aspects of GPU programming. It especially targets community of computational scientists from disciplines known to make use of linear algebra, differential equations, Monte Carlo methods, and Fourier transform.

The book is organized in four parts, each covering a particular set of numerical methods. First part is dedicated to the solution of linear algebra problems, ranging from the matrix–matrix multiplication, to the solution of systems of linear equations, to the computation of eigenvalues. Several chapters in this part address the problem of computing on a very large number of small matrixes. The final chapter also addresses the sparse matrix–vector product problem.

Second part is dedicated to the solution of differential equations and problems based on the space discretization of differential equations. Methods such as finite elements, finite difference, and successive over-relaxation with the applications to problem domains such as flow and wave propagation and solution of Maxwell's equations are presented. One chapter also addresses the challenge of integrating a large number of independent ordinary differential equations.

Third part is dedicated to the use of Monte Carlo methods for numerical integration. Monte Carlo techniques are well suited for GPU implementation and their use is widening. The part also includes chapters about random number generation on GPUs as a necessary first step in Monte Carlo methods.

The final part consists of two chapters dedicated to the efficient implementation of Fourier transform and one chapter discussing N-body simulations.

## **Acknowledgments**

This book consists of contributed chapters provided by the experts in various fields involved with numerical computations on GPUs. I would like to thank all of the contributing authors whose work appears in this edition. I am also thankful to the Directors of the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign for the support and encouragement.

Urbana, IL, USA

Volodymyr Kindratenko

Numerical Computations with GPUs

Kindratenko, V. (Ed.)

2014, X, 405 p. 107 illus., 49 illus. in color., Hardcover

ISBN: 978-3-319-06547-2