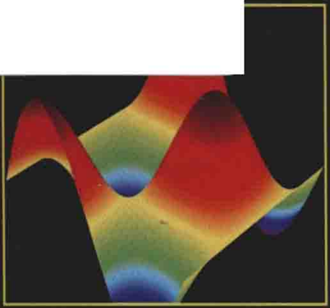
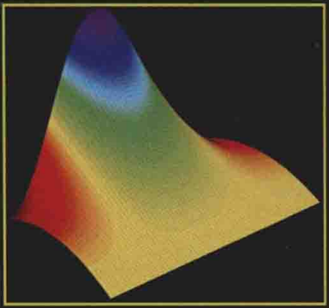
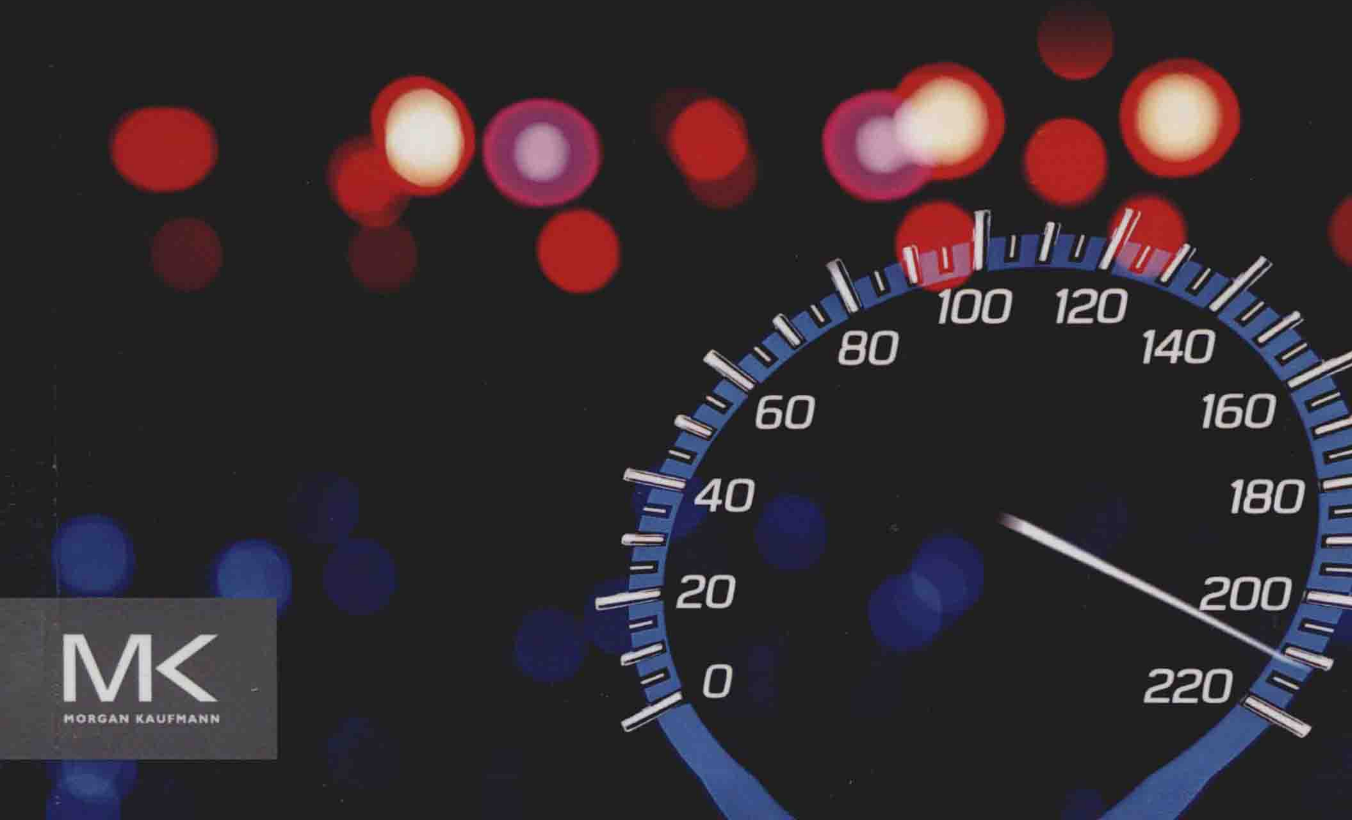
GPU Programming in MATLAB





Nikolaos Ploskas and Nikolaos Samaras



MK
MORGAN KAUFMANN

GPU Programming in MATLAB

Nikolaos Ploskas

Nikolaos Samaras

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO
Morgan Kaufmann is an imprint of Elsevier

MK
MORGAN KAUFMANN

Morgan Kaufmann is an imprint of Elsevier
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States

Notices
Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

ISBN: 978-0-12-805132-0

For information on all Morgan Kaufmann publications
visit our website at https://www.elsevier.com/

Publisher: Todd Green
Acquisition Editor: Todd Green
Editorial Project Manager: Lindsay Lawrence
Production Project Manager: Priya Kumaraguruparan
Cover Designer: Mark Rogers

Typeset by SPi Global, India

GPU Programming in MATLAB

To my family
– Nikolaos Ploskas

To my son, Stathis
– Nikolaos Samaras

About the Authors

Nikolaos Ploskas is a Postdoctoral Researcher at the Department of Chemical Engineering, Carnegie Mellon University, USA. He received his Bachelor of Science degree, Master's degree, and Ph.D. in Computer Systems from the Department of Applied Informatics of the University of Macedonia, Greece. His primary research interests are in

Operations research,
Mathematical programming,
Linear programming,
Parallel programming,
GPU programming,
Decision support systems.

Dr. Ploskas has participated in several international and national research projects. He is the author or co-author of writings in more than 40 publications, including high-impact journals and book chapters, and conference publications. He has also served as a reviewer for many scientific journals. He received an honorary award from HELORS (Hellenic Operations Research Society) for the best doctoral dissertation in operations research (2014).

Nikolaos Samaras is a Professor at the Department of Applied Informatics, School of Information Sciences, University of Macedonia, Greece. Professor Samaras's current research interests are at the interface between computer science and operations research, which apply to a variety of engineering and scientific systems:

Linear/Non Linear optimization: theory, algorithms, and software
Network optimization: theory, algorithms, and software
Scientific computing: HPC, and GPU-programming

He has served on the editorial board of the *Operations Research: An International Journal*, and as a reviewer in many scientific journals. He has also held numerous positions within HELORS (Hellenic Operations Research Society). He was awarded with the Thomson ISI/ASIS&T Citation Analysis Research Grant (2005).

Dr. Samaras has published more than 35 journal papers in high-impact journals, including *Computational Optimization and Applications, Computers and Operations Research, European Journal of Operational Research, Annals of Operations Research, Journal of Artificial Intelligence Research, Discrete Optimization, Applied Mathematics and Computation, International Journal of Computer Mathematics, Electronics Letters, Computer Applications in Engineering Education, Journal of Computational Science, and Applied Thermal Engineering*. He has also published more than 85 conference papers.

Foreword

This book represents an important addition to the library of professional MATLAB reference texts. Whereas most other MATLAB-related texts typically focus on a specific engineering domain, this book targets general MATLAB users, who are already familiar with MATLAB and wish to improve their program's speed using multicore and GPU parallelization. Until recently, parallelization was employed by supercomputers and were outside the reach of the regular MATLAB user. But with multiple CPU cores and powerful GPU cards ubiquitous in modern computers, parallelization is now available to anyone, and it would seem a waste not to use all this available power for our compute-intensive MATLAB programs. Unfortunately, MATLAB users have few resources explaining the fine details about how *exactly* to make their MATLAB programs run on the GPU. MATLAB's internal documentation, good as it may be, may not be enough for professional development. I believe that this book successfully fills this gap. A detailed discussion of GPU programming in MATLAB is presented, starting with a general overview, continuing with a discussion about how to employ easy-to-use gpuArrays, all the way to the detailed intricacies of compiling CUDA kernels and integrating GPU code into MEX-files. The reader therefore benefits from a discussion at various levels of increasing complexity. Multiple usage examples are presented to enable users in different engineering disciplines to understand the material, including a discussion about real-world limitations such as memory or bandwidth. MATLAB error messages, which are sometimes difficult to understand and overcome, are explained alongside suggested solutions/workarounds. Multiple tips and best practices are suggested throughout the book. While this book does not cover other aspects of MATLAB performance tuning in any great detail, its discussion of GPU programming for MATLAB is very detailed and quite up-to-date. With GPU programming becoming commonplace, such a dedicated, detailed and highly readable book about this subject is a welcome addition. This textbook should be on the bookself of any MATLAB programmer who plans to employ GPU parallelization.

Yair Altman

"Accelerating MATLAB Performance," http://UndocumentedMatlab.com

Preface

MATLAB is a high-level language for technical computing. It is widely used as a rapid prototyping tool in many scientific areas. Many researchers and companies use MATLAB to solve computationally intensive problems and run their codes faster. MATLAB provides the Parallel Computing Toolbox that allows users to solve their computationally intensive problems using multicore processors, computer clusters, and GPUs.

With the advances made in hardware, GPUs have gained a lot of popularity in the past decade and have been widely applied to computationally intensive applications. There are currently two major models for programming on GPUs: CUDA and OpenCL. CUDA is more mature and stable. In order to access the CUDA architecture, a programmer can write codes in C/C++ using CUDA C or Fortran using the PGI's CUDA Fortran, among others.

This book, however, takes another approach. This book is intended for students, scientists, and engineers who develop or maintain applications in MATLAB and would like to accelerate their codes using GPU programming without losing the many benefits that MATLAB offers. The readers of this book likely have some or a lot of experience with MATLAB coding, but they are not familiar with parallel architectures.

The main aim of this book is to help readers implement their MATLAB applications on GPUs in order to take advantage of their hardware and accelerate their codes. This book includes examples for every concept that is introduced in order to help its readers apply the knowledge to their applications. We preferred to follow a tutorial rather than a case study approach when writing this book because MATLAB's users have different backgrounds. Hence, the examples presented in this book aim to focus the interest of the readers on the techniques used to implement an application on a GPU and not on a specific application domain. The examples provided are common problems in many scientific areas such as image processing, signal processing, optimization, communications systems, statistics, etc.

MATLAB's documentation for GPU computing is very helpful, but the information is not available in one location and important implementation issues on GPU programming are not discussed thoroughly. Various functions and toolboxes have been created since MATLAB introduced GPU support in 2010, so information is scattered. The aim of this book is to fill this gap. In addition, we provide many real-world examples in various scientific areas in order to demonstrate MATLAB's GPU capabilities. Readers with some experience of CUDA C/C++ programming will also be able to obtain more advanced knowledge by utilizing CUDA C/C++ code in MATLAB or by profiling and optimizing their GPU applications.

The main emphasis of this book is addressed on two fronts:

- The features that MATLAB inherently provides for GPU programming. This part is divided into three parts:
    1. GPU-enabled MATLAB built-in functions that require the existence of the Parallel Computing Toolbox.
    2. Element-wise operations for GPUs that do not require the existence of the Parallel Computing Toolbox.
    3. GPU-enabled MATLAB functions found in several toolboxes other than Parallel Computing Toolbox, including Communications System Toolbox, Image Processing Toolbox, Neural Network Toolbox, Phased Array System Toolbox, Signal Processing Toolbox, and Statistics and Machine Learning Toolbox.
- Linking MATLAB with CUDA C/C++ codes either when MATLAB cannot execute an existing piece of code on GPUs or when the user wants to use highly optimized CUDA-accelerated libraries.

The main target groups of this book are:

- Undergraduate and postgraduate students who take a course on GPU programming and want to use MATLAB to exploit the parallelism in their applications.
- Scientists who develop or maintain applications in MATLAB and would like to accelerate their codes using GPUs without losing the many benefits that MATLAB offers.
- Engineers who want to accelerate their computationally intensive applications in MATLAB without the need to rewrite them in another language, such as CUDA C/C++ or CUDA Fortran.

We are thankful to MathWorks for providing us an academic license for MATLAB through their MathWorks Book Program. We also thank NVIDIA for hardware donations in the context of the NVIDIA Academic Partnership program. Special thanks to Ioannis Athanasiadis for providing ideas and implementing examples for GPU-enabled functions of MATLAB toolboxes. Finally, we thank our families for their love and support over many years.

Nikolaos Ploskas
Nikolaos Samaras

Contents