

# Contents

Acknowledgments . . . . .	xvii
About the Authors . . . . .	xix
<b>Introduction . . . . .</b>	<b>1</b>
What Is CUDA? . . . . .	1
What Does “Need-to-Know” Mean for Learning CUDA? . . . . .	2
What Is Meant by “for Engineers”? . . . . .	3
What Do You Need to Get Started with CUDA? . . . . .	4
How Is This Book Structured? . . . . .	4
Conventions Used in This Book . . . . .	8
Code Used in This Book . . . . .	8
User’s Guide . . . . .	9
Historical Context . . . . .	10
References . . . . .	12
<b>Chapter 1: First Steps . . . . .</b>	<b>13</b>
Running CUDA Samples . . . . .	13
CUDA Samples under Windows . . . . .	14
CUDA Samples under Linux . . . . .	17
Estimating “Acceleration” . . . . .	17

Running Our Own Serial Apps . . . . .	19
dist_v1 . . . . .	19
dist_v2 . . . . .	20
Summary . . . . .	22
Suggested Projects . . . . .	23
<b>Chapter 2: CUDA Essentials . . . . .</b>	<b>25</b>
CUDA's Model for Parallelism . . . . .	25
Need-to-Know CUDA API and C Language Extensions . . . . .	28
Summary . . . . .	31
Suggested Projects . . . . .	31
References . . . . .	31
<b>Chapter 3: From Loops to Grids . . . . .</b>	<b>33</b>
Parallelizing dist_v1 . . . . .	33
Executing dist_v1_cuda . . . . .	37
Parallelizing dist_v2 . . . . .	38
Standard Workflow . . . . .	42
Simplified Workflow . . . . .	43
Unified Memory and Managed Arrays . . . . .	43
Distance App with cudaMallocManaged() . . . . .	44
Summary . . . . .	47
Suggested Projects . . . . .	48
References . . . . .	48
<b>Chapter 4: 2D Grids and Interactive Graphics . . . . .</b>	<b>49</b>
Launching 2D Computational Grids . . . . .	50
Syntax for 2D Kernel Launch . . . . .	51

Defining 2D Kernels . . . . .	52
dist_2d . . . . .	53
Live Display via Graphics Interop . . . . .	56
Application: Stability . . . . .	66
Running the Stability Visualizer . . . . .	73
Summary . . . . .	76
Suggested Projects . . . . .	76
References . . . . .	77
<b>Chapter 5: Stencils and Shared Memory . . . . .</b>	<b>79</b>
Thread Interdependence . . . . .	80
Computing Derivatives on a 1D Grid . . . . .	81
Implementing dd_1d_global . . . . .	82
Implementing dd_1d_shared . . . . .	85
Solving Laplace's Equation in 2D: heat_2d . . . . .	88
Sharpening Edges in an Image: sharpen . . . . .	102
Summary . . . . .	117
Suggested Projects . . . . .	118
References . . . . .	119
<b>Chapter 6: Reduction and Atomic Functions . . . . .</b>	<b>121</b>
Threads Interacting Globally . . . . .	121
Implementing parallel_dot . . . . .	123
Computing Integral Properties: centroid_2d . . . . .	130
Summary . . . . .	138
Suggested Projects . . . . .	138
References . . . . .	138

<b>Chapter 7: Interacting with 3D Data</b> . . . . .	<b>141</b>
Launching 3D Computational Grids: <code>dist_3d</code> . . . . .	144
Viewing and Interacting with 3D Data: <code>vis_3d</code> . . . . .	146
Slicing . . . . .	149
Volume Rendering . . . . .	153
Raycasting . . . . .	154
Creating the <code>vis_3d</code> App . . . . .	156
Summary . . . . .	171
Suggested Projects . . . . .	171
References . . . . .	171
<b>Chapter 8: Using CUDA Libraries</b> . . . . .	<b>173</b>
Custom versus Off-the-Shelf . . . . .	173
Thrust . . . . .	175
Computing Norms with <code>inner_product()</code> . . . . .	176
Computing Distances with <code>transform()</code> . . . . .	180
Estimating Pi with <code>generate()</code> , <code>transform()</code> , and <code>reduce()</code> . . . . .	185
cuRAND . . . . .	190
NPP . . . . .	193
<code>sharpen_npp</code> . . . . .	194
More Image Processing with NPP . . . . .	198
Linear Algebra Using cuSOLVER and cuBLAS . . . . .	201
cuDNN . . . . .	207
ArrayFire . . . . .	207
Summary . . . . .	207
Suggested Projects . . . . .	208
References . . . . .	209

<b>Chapter 9: Exploring the CUDA Ecosystem . . . . .</b>	<b>211</b>
The Go-To List of Primary Sources . . . . .	211
CUDA Zone . . . . .	211
Other Primary Web Sources . . . . .	212
Online Courses . . . . .	213
CUDA Books . . . . .	214
Further Sources . . . . .	217
CUDA Samples . . . . .	217
CUDA Languages and Libraries . . . . .	217
More CUDA Books . . . . .	217
Summary . . . . .	218
Suggested Projects . . . . .	219
<b>Appendix A: Hardware Setup . . . . .</b>	<b>221</b>
Checking for an NVIDIA GPU: Windows . . . . .	221
Checking for an NVIDIA GPU: OS X . . . . .	222
Checking for an NVIDIA GPU: Linux . . . . .	223
Determining Compute Capability . . . . .	223
Upgrading Compute Capability . . . . .	225
Mac or Notebook Computer with a CUDA-Enabled GPU . . . . .	225
Desktop Computer . . . . .	226
<b>Appendix B: Software Setup . . . . .</b>	<b>229</b>
Windows Setup . . . . .	229
Creating a Restore Point . . . . .	230
Installing the IDE . . . . .	230
Installing the CUDA Toolkit . . . . .	230
Initial Test Run . . . . .	235

OS X Setup . . . . .	238
Downloading and Installing the CUDA Toolkit . . . . .	239
Linux Setup . . . . .	240
Preparing the System Software for CUDA Installation . . . . .	240
Downloading and Installing the CUDA Toolkit . . . . .	240
Installing Samples to the User Directory . . . . .	241
Initial Test Run . . . . .	242
<b>Appendix C: Need-to-Know C Programming . . . . .</b>	<b>245</b>
Characterization of C . . . . .	245
C Language Basics . . . . .	246
Data Types, Declarations, and Assignments . . . . .	248
Defining Functions . . . . .	250
Building Apps: Create, Compile, Run, Debug . . . . .	251
Building Apps in Windows . . . . .	252
Building Apps in Linux . . . . .	258
Arrays, Memory Allocation, and Pointers . . . . .	262
Control Statements: <code>for</code> , <code>if</code> . . . . .	263
The <code>for</code> Loop . . . . .	264
The <code>if</code> Statement . . . . .	265
Other Control Statements . . . . .	267
Sample C Programs . . . . .	267
<code>dist_v1</code> . . . . .	267
<code>dist_v2</code> . . . . .	271
<code>dist_v2</code> with Dynamic Memory . . . . .	275
References . . . . .	277

<b>Appendix D: CUDA Practicalities: Timing, Profiling, Error Handling, and Debugging . . . . .</b>	<b>279</b>
Execution Timing and Profiling . . . . .	279
Standard C Timing Methods . . . . .	280
CUDA Events . . . . .	282
Profiling with NVIDIA Visual Profiler . . . . .	284
Profiling in Nsight Visual Studio . . . . .	288
Error Handling . . . . .	292
Debugging in Windows . . . . .	298
Debugging in Linux . . . . .	305
CUDA-MEMCHECK . . . . .	308
Using Visual Studio Property Pages . . . . .	309
References . . . . .	312
Index . . . . .	313

Thank you to the many colleagues here at the University of Washington—Seattle Department of Mechanical Engineering who contributed via discussions ranging from big-picture perspective down to the finest technical details. That list includes but is not limited to Mark Genter, Di Zheng, and Ben Weiss (who helped create several of the figures and also provided us with some lifesaving software to support logical tags and automated code formatting). We would also like to thank Mechanical Engineering Department Chair Per Reinholt for his approval for us to offer the class that helped to inspire the creation of much of the book's content. Additional thanks go to our colleagues David Haynor of University of Washington Department of Radiology and William Ledoux of the Seattle VA Hospital, whose research initiatives continue to motivate meaningful journeys into CUDA territory.

We wish to say a special thank you to the good folks at NVIDIA, including CEO Jensen Huang who not only had, but also acted on, a vision of what could be accomplished by enhancing access to GPU-based parallel computing; Chandie Chey, Academic Programs Manager; Kimberly Powell, Director of Higher Education and Healthcare Industries; Jon Sapozhnik and Bob Cravetta, helpful and inspirational CUDA gurus; and last, but definitely not least, Jay White, Director