Preface

Adults, like children, learn by playing. With this insight in mind, I wrote this book as a non-exhaustive and non-overambitius text that aims at introducing computational geodynamics to young students, either undergraduates or beginning graduates. The goal, more than trying to cover a topic that is way too vast for a book to contain it, is to show some of the fundamental strategies available with a strong focus on practical, playful, easy-to-use techniques.

Python is now a standard tool for programming in science and even more in the industry, where the explosion of the Big Data, mining, and artificial intelligence research fields has convinced many professionals to learn this simple and powerful language. While I write, Python, combined with its visualization and numerical libraries, is replacing MATLAB in many scientific and technological contexts, because it offers the same capabilities but with the great advantage of being open source. Open projects attract more developers which, in the long term, always overcomes closed/commercial tools if the user base is sufficiently large. Many developers are in this moment turning existing C and Fortran libraries into Python, among them the ones that allow parallel computing, programming NVIDIA GPUs, running machine learning algorithms, and many more. Although Numerical Python is not the only fast growing scientific/technological language, as others such as Ruby and Scala are emerging as well, Python is the present state of the art in scientific computing.

Geodynamics, the study of the evolution of our planet and similarly of the other solid planets, is a fast evolving field. It constantly englobes new research topics ranging from the dynamics of the Earth's core, to the convection of its slowly deforming solid mantle, from the study of the dynamics of the Geoid (the *gravitational* shape of the Earth) to the global resonance of the entire planet to the stimulus of a large Earthquake on its surface, from the dynamics of the two-phase gas-magma system in a volcano conduit to the estimation of the risks related to small and giant eruptions, from the dramatic formation of our planet to the increase of its mineralogical variety during its evolution, from the detection of the chemical spectrum of the atmosphere of some exoplanets modelling their interior dynamics.

Given that most of these systems are at the same time complex and inaccessible to direct measurements, numerical modeling plays a key a role to their understanding. I have been teaching numerical modeling applied to geosciences, and geodynamics in particular, in my past academic positions at the University of Sydney, Seoul National University, and presently at the University of Louisiana at Lafayette. In my courses I have used numerous books that covered several aspects of geodynamics, however many of the key progresses in the past 30 years have been driven or confirmed by well crafted numerical experiments. I have, therefore, decided to write a textbook that focuses on the practice of designing these numerical experiments, organized as a hands-on manual, almost like one of these online tutorial that programmers use every day, but with a greater structure, designed to teach undergraduate and graduate students to create their own numerical models.

In my courses, I verified that the insight that students gain by designing and writing a code is often greater than the understanding gained by the solution itself. To create a software that simulates the Earth's interior behavior is like to teach geodynamics to a machine, and requires a very clear understanding of the underlying physics behind geological phenomena. This book combines the material taught in three different occasions. One is the introduction to computation to honor sophomore students of general physics at the University of Louisiana at Lafayette. The second is a course of computational geophysics taught at the same university to graduate students during the academic years 2013/2014 and 2014/2015. Finally, I added advanced topics such as the one taught at a summer school on large-scale Boundary Element Method (BEM) at the University of Brescia, Italy, in 2011.

The book is structured in four parts. In the first one, I introduce in three chapters a (i) bird's eye view of the computational capabilities of Python, (ii) the visualization tools available in Python, and (iii) how to use the powerful Numerical Python libraries and other numerical tools to embed C, OpenMP, MPI capabilities to our Python programs.

In the second part, I illustrate few examples of how to use Numerical Python to solve typical problems of a standard general physics course. These problems are normally solved only using analytical tools, while here I illustrate how to approach them numerically, and at the same time familiarize with momentum and energy equations, the main ones that are solved in geodynamic modeling. The goal of this part is to introduce the less expert reader to using Python, solve numerically some simple problems, and teach at the same time how to calculate standard physics quantities such as momentum, work, power, dissipative energy, and the action. In this part, we will for the first time learn how to numerically calculate derivatives and integrals, how to monitor the accuracy of a numerical solution, and how to visualize the results of a computational model.

In the third part of the book, I introduce the main laws of continuum mechanics that every geodynamicist needs to know, diffusion and momentum equations in a continuum context, and how to use Finite Differences, and the advanced *Particle in Cell (PIC)* technique to solve them. The goal of this part is to describe how to write compact and elegant but still very fast routines that allow implementing a sophisticate and advanced code such as *PIC* in simple and straightforward manner. Applications to solving the equations that describe strongly viscous flow, porous media flow, elasticity, both in a linear and nonlinear setup are introduced in a simplified and introductory manner. Also some important issues such as numerical stability/instability are shown using examples.

The fourth and last part of the book covers a small set of more advanced topics such as numerical approaches that do not require the discretization of the space, but are built from the summation of fundamental solutions. This will give the reader the opportunity to familiarize with tree-based codes, that are at the core of many modern numerical techniques and can efficiently solve the many body interaction problem that applies to many aspects of geosciences and beyond, such as planetary science. A specific application to the calculation of the dynamics of a multiphase fluid is finally introduced and applied to the motion of gas bubbles in a magmatic system.

I believe that this book still lacks much material that I would like to add. In particular more details on the solution techniques for the non-linear Navier-Stokes equation, lagrangian multiplier method, detailed implementations of the tree fast multipole method, and many more examples of how to parallelize and benchmark these codes. I have however to stop here and plan these additions for a second edition of this book. I hope that the very incomplete material of this first edition will anyway result useful to some young students who approach the fascinating field of computational geodynamics for the first time.

Who Should Read this Book

This book has been written for ambitious undergraduate college students and for graduate students in geophysics/geodynamics. I eliminated most of the calculus based derivations, while I focus on in implementations and examples, with the idea that it is the ‘practice’ that makes the ‘scientist’, and that ‘creativity’ is the product of ‘perseverance’. This book is, therefore, designed for readers of every background who desire to learn about the behavior of our planet by explicitly modeling it.

This book is also addressed to more advanced practitioners who have been modelling geodynamics using different programming languages than Python and aim at trying new numerical techniques. In many ways Python is the best language for prototyping scientific implementations, and also for running some high performance simulations. I remember that when I begun my Ph.D. in Geodynamics, I dreamed about learning everything, mantle convection, earthquakes, seismic wave propagation, oceanography, glacial rebound, exoplanets, surface processes, and so on, but quickly I realized that to write an efficient, robust, and reliable software for modeling is a huge work. I certainly do not affirm that now that with the scientific environment of Python, and the emerging new tools that come from Machine Learning, this is today possible, but certainly presently a numerical modeler can aim at a much broader and multidisciplinary project than it was possible 15 years ago.

How Should this Book be Read

This volume is structured with increasing difficulty, starting from a level accessible to freshman college student. Through the book the level rises until reaching some topics chosen from the present-day geodynamics research. The goal of this book is to offer to the reader the key instruments not only to create general and powerful computational tools, but also a clear understanding of the difficulty of implementing them.

The reader is expected to test all the examples proposed and try to do as many exercises as possible. Real learning is achieved only by writing a software by our own, and this is much easier to achieve in Python compared to standard scientific languages such as C and Fortran.

The reader with little background in computational sciences will find easier to study the book in the same order as the material is presented, but the expert programmer can safely skip the first chapters where the main tools for achieving high performance with Python are introduced. Only from Chap. 7 the book starts to build on past shown examples, therefore sequential reading is recommended.

Great software for Geodynamics already exist today, such as *Access*, *Underworld*, and Terra. Still it is possible today, by using smart programming to quickly but gradually guide a student to building a geodynamic modeling software. The goal of this book is not to push the student toward competing against large projects, but to prepare them to understand how they work in order to work within these projects, or to develop new modules for running techniques never implemented elsewhere. In other word the idea of this book is to invite students to learn by experimenting in freedom.

Lafayette, Louisiana, USA Gabriele Morra
December 2016

http://www.springer.com/978-3-319-55680-2

Pythonic Geodynamics
Implementations for Fast Computing
Morra, G.
2018, XVI, 227 p. 46 illus., 40 illus. in color., Hardcover
ISBN: 978-3-319-55680-2