Part V

Advanced Methods of Computational Gasdynamics

Logically, all things are created by a combination of simpler, less capable components.
Dogbert in Dilbert *by Scott Adams*

This part of the book concerns solution-sensitive methods. After possibly accounting for the wind direction, the first-generation methods studied so far mindlessly treat every part of the solution the same, regardless of how the solution behaves. The solution-sensitive methods studied in Part V combine a range of first-generation methods, varying the exact blend from place to place based on solution features such as shocks. For example, a solution-sensitive method might use Roe's first-order upwind method at shocks and the Lax–Wendroff method in smooth regions. Most solution-sensitive methods decide what to do based on solution gradients or flux gradients; large gradients indicate shocks or other features with the potential to cause trouble in the polynomial interpolations underlying most numerical methods.

In the literature, solution-sensitive methods are commonly called *high-resolution* or *TVD* methods. Less common terms include *hybrid*, *adaptive*, *self-adjusting*, *averaged*, *reconstructed*, *essentially nonoscillatory*, *combination*, *corrected*, and *limited* methods. No one book could describe the huge number of solution-sensitive methods suggested in the research literature. Part V includes methods based partly on historical significance and influence, partly on how well they exemplify general principles and notations, partly on current popularity, and partly on the author's personal tastes.

Many solution-sensitive methods explicitly enforce nonlinear stability conditions, such as the upwind range condition or the TVD condition. Stability conditions reduce spurious oscillations near shocks, but they also typically impose clipping errors at solution extrema. Although combination methods have trade-offs, these are far less severe than for the fixed methods studied in Part IV. In particular, far from trading stability with accuracy at extrema, fixed methods trade stability with accuracy throughout the *entire* solution. Fixed methods either do well at shocks or well in smooth regions, but not both simultaneously, which stems from the identical behavior found in the underlying interpolation polynomials, as discussed in Chapters 8, 9, and 10.

Part V tries to combine the best features of logical and chronological ordering, to give both a necessary sense of history and of the logical connections between apparently disparate methods. As a first cut, the methods in Part V are broken down logically into either *flux-averaged methods* or *solution-averaged* methods, more commonly known as *reconstruction–evolution* methods. In an alternative but equivalent terminology, the methods in Part V are divided into *flux-reconstructed* and *solution-reconstructed* methods. As a second cut, the flux-averaged methods are divided logically into flux-limited, flux-corrected, and self-adjusting hybrid methods. Each of these four threads – flux limited, flux corrected, self-adjusting hybrid, and reconstruction–evolution – has its own chapter. While Part V is divided logically into chapters, each chapter is divided chronologically into sections. In other words, each chapter orders its methods according to their birth dates.

The first section in each chapter of Part V describes a seminal *second-generation* method. Subsequent sections describe *third-generation* methods inspired by the second-generation method. Designed in the 1970s, second-generation methods have limited practical potential.

For example, many second-generation methods only apply to the linear advection equation. Those developed for the Euler equations have relatively primitive features, especially with regard to modeling multiwave families. The period from 1979 to 1983 saw several important advances in the first-generation methods underlying the second-generation methods, including flux vector splitting, real and approximate Riemann solvers, and entropy fixes, as described in Chapter 18. From 1979 to 1987, and more slowly since then, researchers formed the third-generation methods by combining the advances in the first-generation methods with the combination approaches developed earlier for second-generation methods.

As explained previously, Part V separates flux-averaged methods into three species – flux limited, self-adjusting hybrid, and flux corrected. However, these divisions mask the fact that, on the most fundamental level, all three approaches are the same. The differences arise not in the basic ideas and philosophies, but in the details, the traditions, the jargon, and the notations. Thus flux-averaged methods are like triplets separated at birth: They have much of the same basic genetics, but environmental factors have conspired to make them increasingly different from each other over the years. For linear systems of equations, there is no fundamental distinction between flux averaging, in any form, and solution averaging. In other words, if $f(u) = au$ where $a = const.$, it does not matter whether you average to find u and then find $f(u)$ or whether you average to find $f(u)$ directly, since the two types of averages differ only by the constant factor a. However, for nonlinear equations with nonlinear fluxes, flux averaging and solution averaging are distinct, with only a few notable exceptions.

Part V describes numerical methods using a modular approach, first covering scalar conservation laws and then the Euler equations. In a few cases, for clarity or brevity, the description starts with the linear advection equation or ends with scalar conservation laws. To save space, Part V only tests numerical methods for scalar conservation laws.

Part V describes over a dozen distinct numerical methods. The "best" method depends on your application, efficiency demands, accuracy demands, complexity tolerance, and taste. Unfortunately, the difficulties of full-scale numerical testing and the limits of theoretical analysis do not allow for complete and definitive descriptions of the trade-offs between specific methods. When you realize that even problems as simple as integration and differentiation have spawned a number of peacefully coexisting numerical methods, as seen in Chapter 10, it should come as no surprise that something as complicated as a system of partial differential or integral equations should suffer a similar fate.

Although this book cannot offer the reader a "best" method, it can offer a personal perspective on the current popularity of the various categories of methods. Remember that Part V divides methods into four species – flux limited, flux corrected, self-adjusting hybrids, and reconstruction–evolution. Of these four, flux-limited and "flux limited–like" methods currently appear to have the upper hand. Flux-corrected methods reached maturity first, but they have not received the same level of attention and continuing development in recent times as other sorts of methods and have accordingly fallen somewhat behind the state of the art. However, certain communities still use flux-corrected methods almost exclusively. Outside of these communities, however, the most widely used flux-corrected methods strongly resemble traditional flux-limited methods. Self-adjusting hybrids were the most popular sort of methods in the 1980s, but these have more recently lost ground to flux-limited and "flux limited–like" methods. Last but not least, reconstruction–evolution methods are the Rolls Royce of numerical methods – elegant, elaborate, rigorous, prestigious, but too expensive for most people. The only exceptions are, again, those reconstruction–evolution methods that strongly resemble traditional flux-limited methods.

The reader may find it enlightening to see the course taken by one of the leading independent commercial software companies in this field. FLUENT corporation offers a computational gasdynamics package called RAMPANT. In its original incarnation, written in the late 1980s, RAMPANT used Jameson's self-adjusting hybrid method, described in Section 22.3. However, the most recent version of RAMPANT uses a modified version of the Anderson–Thomas–Van Leer method described in Subsection 23.3.1, which is a finite-volume version of the Chakravarthy–Osher flux-limited method seen in Subsection 20.3.1, which is, in turn, a semidiscrete version of Sweby's flux-limited method seen in Section 20.2. In conversation, Dr. Wayne Smith, who played a leading role in developing both the original and new versions of RAMPANT, gives two reasons for the change in the core method: First, the flux-limited method does a better job at reducing spurious overshoots and oscillations; second, the flux-limited method seems to mate better with the viscous terms and the other sorts of important real-life effects that routinely appear in practical applications. Like most commercial codes, RAMPANT incorporates a plethora of features, many of them heuristic, which adapt it to complicated geometries, and to many types of compressible gas flows, with maximum efficiency for either steady or unsteady flows. Thus, in the end, the core method is not necessarily as important as all of the implementation details, efficiencies, and generalizations layered on top. This book is devoted to explaining the core methods, since this is the first and most vital step towards understanding other aspects of numerical methods, as explained starting way back in Chapter 1.

This part of the book requires extra effort and patience, not because it involves especially difficult concepts or details, but because it combines on so many concepts and details at once, which often requires complicated notation and a flawless command of the rest of the book. Despite every effort to improve the presentation, some of the sections seen in Part V still require the cool skill of a matador combined with the calm steadfastness of a saint. However, having said this, if you have assiduously studied the earlier chapters, you might be surprised at how quickly you master most of the remaining material, especially if you remember to rest occasionally and to reduce things into simple modular components and concepts.