Low-Storage Time Integrators with Improved Accuracy and Stability Properties

David I. Ketcheson

Articles you may be interested in

Symplectic Runge-Kutta-Nyström Methods with Phase-Lag Order Six and Infinity
AIP Conf. Proc. **1281**, 694 (2010); 10.1063/1.3498573

Multirate Implicit-Explicit Time Integration Schemes in Atmospheric Modelling
AIP Conf. Proc. **1281**, 1831 (2010); 10.1063/1.3498252

Exponential Methods for the Time Integration of Schrödinger Equation
AIP Conf. Proc. **1281**, 1821 (2010); 10.1063/1.3498247

Large-Scale Multiple Scattering Analysis Using Fast Multipole BEM in Time-Domain
AIP Conf. Proc. **1233**, 1196 (2010); 10.1063/1.3452072

Parareal and Spectral Deferred Corrections
AIP Conf. Proc. **1048**, 388 (2008); 10.1063/1.2990941

Low-Storage Time Integrators with Improved Accuracy and Stability Properties

David I. Ketcheson

King Abdullah University of Science and Technology, Box 4700, Thuwal, Saudi Arabia, 23955-6900.
(david.ketcheson@kaust.edu.sa)

Solution of partial differential equations by the method of lines requires the integration of large numbers of ordinary differential equations (ODEs). In such computations, storage requirements are typically one of the main considerations, especially if a high order ODE solver is required. Recently, in [1], a new class of Runge-Kutta methods was proposed that requires only two storage locations per ODE, even when an error estimate is needed. This class generalizes and extends the previously-existing classes of low-storage Runge-Kutta methods. In this talk, we review this new class of low-storage Runge-Kutta methods and present newly developed methods that have particularly good error constants and/or stability regions. We also present new methods that maintain the linear stability properties of well-known full storage methods, like the method of Fehlberg, or of Dormand & Prince, while offering a low-storage implementation.

The two previously-known classes of low-storage methods were developed by Williamson [2] and by van der Houwen & Wray [3], respectively. During the last decade, both classes of methods have been further developed, to include methods optimized for special properties and methods with embedded error estimators. Existing classes of low-storage embedded methods require an extra (third) memory register to provide the error estimate. If a desired error tolerance is exceeded in a given step, it may be necessary to restart that step. In this case, another additional register is necessary for storing the previous step solution. In some applications where no error estimate is used, restarting may still be required based on some other condition (such as a CFL condition) that is checked after each step. In this case, again, existing low-storage methods require the use of an extra register.

The new methods developed in [1] are capable of providing a solution and an error estimate with only two registers, or a solution, error estimate, and restart capability with only three registers; for this reason we refer to them as *minimum-storage* methods. In [1], a few reasonably good fourth-order methods of these types were presented. The focus of this talk is the presentation of new methods of third to fifth order with even better properties. These methods compare favorably with those developed by Kennedy et. al. [4], in the most extensive previous work on low-storage methods. Namely, the new methods possess smaller error constants, larger stability regions, and/or less demanding memory requirements.

Changing the time integrator in a production method-of-lines PDE solving code is typically one of the simplest numerically significant modifications that can be made. Hence these new methods should quickly yield a high payoff for codes that operate on memory-limited supercomputers. In order to encourage practitioners to switch to minimum-storage methods, it may be helpful to provide integrators with properties similar to those currently in use. To this end, we develop new methods whose stability functions coincide with those of important existing full-storage methods, like Fehlberg's, or Dormand & Prince. We demonstrate through computational examples that these new methods behave similarly to the methods they are modeled on, while providing the additional benefit of low storage.

REFERENCES

1. D. Ketcheson, *Journal of Computational Physics* **229**, 1763 – 1773 (2010).
2. J. H. Williamson, *Journal of Computational Physics* **35**, 48–56 (1980).
3. P. van der Houwen, *Numerische Mathematik* **20**, 149–164 (1972).
4. C. A. Kennedy, M. H. Carpenter, and R. M. Lewis, *Applied Numerical Mathematics* **35**, 177–219 (2000).