Advanced Time Integration for Magnetic Fusion

D. R. Reynolds[1], R. Samtaney[2], A. Sandu[3], M. Tokman[4], C. S. Woodward[5]

[1]Southern Methodist University, [2]KAUST, [3]Virginia Tech, [4]University of California, Merced (mtokman@ucmerced.edu), [5]Lawrence Livermore National Laboratory

Introduction. Modeling fusion devices gives rise to a multitude of problems that require solution of large systems of evolutionary equations. In applications ranging from modeling the core of a fusion device using magnetohydrodynamics (MHD) to describing the behavior of plasma edge with Vlasov-Maxwell type equations, the ability to accurately and efficiently compute the evolution of a given system in time is a central task. Given the complexity and large scale of such systems of equations, solving them numerically in experimentally relevant parameter regimes over spatial and temporal scales of interest presents a difficult computational challenge. Recent advances in the field of numerical time integration and software can potentially bring significant computational savings to problems in fusion modeling. Achieving computational breakthroughs in this field, however, requires close collaboration between the fusion and numerical time integration communities. Such collaboration is necessary to create tailor-made integrators for problems in fusion and develop next generation codes that will include advanced time stepping techniques, adaptive strategies, runtime error monitoring tools, and scalable algorithms for exascale computing platforms.

Time integration in magnetic fusion codes. Production fusion codes (here we refer to codes utilized for simulating tokamak physics in three physical dimensions and exclude other workhorse codes such as TRANSP) may be broadly categorized into two types: (a) those dealing with macroscopic tokamak physics usually modeled as fluid (different flavors of magnetohydrodynamics) codes and (b) those dealing with turbulent transport and plasma micro-turbulence, usually modeled with gyrokinetic models. The former include codes such as M3D and its successor M3D-C1 [28] and NIMROD [46], both of which have the capability to solve single-fluid resistive MHD and also have two-fluid MHD capability (two-fluid MHD typically involves high frequency dispersive waves, and generally requires implicit treatment). The original M3D code utilized a semi-implicit time integration process, treating the fast compressive waves implicitly, and the remaining shear Alfven wave modes and slower modes explicitly. The time integration schemes in nonlinear MHD codes (M3D-C1 and NIMROD) may be characterized as linearly implicit or semi-implicit. These linearizations or semi-implicit decompositions are typically formed through problem-specific, physics-based decompositions, with a goal of generating subsystems that are more amenable to scalable algebraic solvers. Within both M3D-C1 and NIMROD, these techniques rely on first specifying a fixed-step "θ-method" for time discretization of each equation, along with physically-motivated selection of individual terms to treat implicitly within the solver. However, while these time discretization approaches excel at rendering scalable algebraic solvers, they typically lack rigorous mathematical theory regarding overall convergence or stability for the coupled systems, and do not admit *a posteriori* estimates of temporal error for use in adaptivity or uncertainty quantification.

The microturbulence codes are further split into those which utilize a "continuum" approach (essentially a misnomer) in which the velocity space is discretized (e.g. GYRO) or particle (PIC) codes in which Lagrangian particles essentially sample the velocity space (e.g XGC and GTC). Within the major particle codes, a large system of ODEs must be evolved, which include highly non-trivial global force calculations for particle accelerations. Due to the lack of severe time step stability restrictions in these systems, these codes typically use fixed-step explicit Runge-Kutta or "leapfrog" methods for time integration, and spend significant effort on scalable solvers for the inter-particle force calculations. Furthermore, due to differences in particle speeds, such codes may include time step subcycling for faster-moving particles.

Advances in time integration. There have been significant advances made in the field of time integration over the past several decades. New classes of time integration schemes have been introduced, existing methods have been refined and a range of algorithmic improvements and software [23, 24, 41, 50, 51] have been developed to enable accurate, stable and efficient implementation of these methods on a variety of architectures.

Numerous improvements have been introduced in the field of implicit integration with many new schemes constructed. For instance, implicit methods can now be constructed with optimized error constants for a given problem or specifically designed to reduce the computational time of the linear solves embedded within an implicit integrator (e.g.,[39, 48, 49]). A new class of exponential integrators have recently gained interest as an alternative to implicit methods [25]. Like the implicit methods, exponential techniques possess good stability properties but offer computational advantages compared to the implicit schemes particularly for problems where it is difficult to construct an efficient preconditioner. There have been a number of innovations in the way implicit or exponential integrators can be coupled with Krylov solvers and other algorithms to derive efficient time integration techniques (e.g. Rosenbrock-Krylov [49] and exponential-Krylov [48] methods). More traditional approaches such as the fully implicit Newton-Krylov methods with "physics-based" preconditioners, the FAS (full approximation scheme) multigrid methods [36] and exponential integrators have also shown promise for MHD [7, 8, 11–13, 32, 47], but have not yet been used in production fusion codes.

While many of the fusion codes employ the technique of applying either different time integration methods (e.g. explicit, implicit, etc) to different physical terms within a single differential equation or using a technique where each of these terms is advanced separately in time, often such multimethod or splitting approaches are used in an *ad hoc* manner without rigorous error estimators or analysis that ensure the accuracy and stability of the overall time integrator. Considerable progress has been made recently on constructing multimethods that use different strategies for solving different components of the system [6, 45, 52, 53]. New analytical results on splitting time integrators [14, 26, 42] allow for better understanding of the error incurred in using either a term or dimensional splitting as well as construction of new arbitrarily high-order in time splitting approaches (e.g. new Hamiltonian splitting methods for Vlasov-Maxwell systems [38]). Asynchronous multirate time stepping takes different time steps for different components to achieve a global target accuracy [2, 3, 43]. Multirate versions of many time stepping schemes have been proposed, including linear multistep [1, 19, 29], extrapolation [10, 17], Runge-Kutta [9, 20, 22, 30, 31], Rosenbrock-Wanner [4, 21], waveform relaxation [15, 44], Galerkin [33–35], and combined multiscale [16] approaches.

A promising new direction is development of time integrators particularly adapted to massively parallel architectures. Initial steps are being taken in development of parallel in time algorithms [5, 27, 37, 40]. Error estimators can be used to improve algorithmic resiliency [18].

Conclusions. A wide range of new computational methods have been developed in the field of time integration. Some of these new techniques can be added to existing fusion simulation codes with relative ease, others require more research and fundamental changes to the currently employed algorithms and implementations. However, thorough understanding of the computational challenges of a given problem is necessary to build an efficient time integrator. Constructing, analyzing and implementing state-of-the-art time integration techniques that can deliver computational breakthroughs for the fusion simulations seems to be only possible if there is a close collaboration between the physicists and the numerical analysts working on the problem. Such collaborations should be encouraged and developed through funding mechanisms as well as joint workshops.

References

[1] E. Constantinescu A. Sandu. Multirate explicit Adams methods for time integration of conservation laws. *Journal of Scientific Computing*, 38:229–249, 2009. 10.1007/s10915-008-9235-3.

[2] J.F. Andrus. Numerical solution for ordinary differential equations separated into subsystems. *SIAM Journal on Numerical Analysis*, 16(4):605–611, 1979.

[3] J.F. Andrus. Stability of a multi-rate method for numerical integration of ODEs. *Computers & Mathematics with applications*, 25(2):3–14, 1993.

[4] A. Bartel and M. Günther. A multirate W-method for electrical networks in state-space formulation. *Journal of Computational and Applied Mathematics*, 147(2):411–425, 2002.

[5] K. Burrage. Parallel methods for initial value problems. *Applied Numerical Mathematics*, 11(1–3):5 – 25, 1993.

[6] A. Cardone, Z. Jackiewicz, A. Sandu, and H. Zhang. Extrapolated IMEX Runge-Kutta methods. *Mathematical Modelling and Analysis*, 19(2):18–43, 2014.

[7] L. Chacón. An optimal, parallel, fully implicit Newton–Krylov solver for three-dimensional viscoresistive magnetohydrodynamics. *Physics of Plasmas (1994-present)*, 15(5):056103, 2008.

[8] L. Chacón and D. A. Knoll. A 2D high-β Hall MHD implicit nonlinear solver. *Journal of Computational Physics*, 188(2):573–592, 2003.

[9] E.M. Constantinescu and A. Sandu. Multirate timestepping methods for hyperbolic conservation laws. *Journal on Scientific Computing*, 33(3):239–278, 2007.

[10] E.M. Constantinescu and A. Sandu. Extrapolated implicit-explicit time stepping. *SIAM Journal on Scientific Computing*, 31(6):4452–4477, 2010.

[11] R. Samtaney D. R. Reynolds and H.C. Tiedeman. A fully implicit Newton–Krylov–Schwarz method for tokamak magnetohydrodynamics: Jacobian construction and preconditioner formulation. *Computational Science & Discovery*, 5(1):014003, 2012.

[12] R. Samtaney D. R. Reynolds and C.S. Woodward. A fully implicit numerical method for single-fluid resistive magnetohydrodynamics. *Journal of Computational Physics*, 219(1):144–162, 2006.

[13] R. Samtaney D. R. Reynolds and C.S. Woodward. Operator-based preconditioning of stiff hyperbolic systems. *SIAM Journal on Scientific Computing*, 32(1):150–170, 2010.

[14] G. Wanner E. Hairer, C. Lubich. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations.* Springer, Berlin, 2 edition, 2006.

[15] A.E. Ruehli E. Lelarasmee and A.L. Sangiovanni-Vincentelli. The waveform relaxation method for time-domain analysis of large scale integrated circuits. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 1(3):131–145, 1982.

[16] B. Engquist and R. Tsai. Heterogeneous multiscale methods for stiff ordinary differential equations. *Mathematics of Computation*, 74:1707–1742, 2005.

[17] C. Engstler and C. Lubich. Multirate extrapolation methods for differential equations with different time scales. *Computing*, 58(2):173–185, 1997.

[18] W. H. Enright and D.J. Higham. Parallel defect control. *BIT Numerical Mathematics*, 31(4):647–663, 1991.

[19] C.W. Gear and D.R. Wells. Multirate linear multistep methods. *BIT*, 24:484–502, 1984.

[20] M. Günther, A. Kværnø, and P. Rentrop. Multirate partitioned Runge-Kutta methods. *BIT Numerical Mathematics*, 41(3):504–514, 2001. 0006-3835.

[21] M. Günther and P. Rentrop. Multirate ROW-methods and latency of electric circuits. *Applied Numerical Mathematics*, 13(1-3):83–102, 1993.

[22] M. Günther and A. Sandu. Multirate GARK methods. *Numerische Mathematik*, Conditionally accepted., 2015.

[23] A.C. Hindmarsh. LSODE and LSODI, two new initial value ordinary differential equation solvers. *SIGNUM Newsl.*, 15(4):10–11, 1980.

[24] A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, and C.S. Woodward. SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. *ACM Trans. Math. Soft.*, 31(3):363–396, 2005.

[25] M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numer.*, pages 209–286, 2010.

[26] W. Hundsdorfer and J. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer, Berlin, 2003.

[27] G. Turinici J. L. Lions, Y. Maday. A parallel in time discretization of PDEs. *C.R. Acad. Sci. Paris*, 332:661–668, 2001.

[28] S.C. Jardin, N. Ferraro, X. Luo, J. Chen, J. Breslau, K.E. Jansen, and M.S. Shephard. The M3D-c^1 approach to simulating 3D 2-fluid magnetohydrodynamics in magnetic fusion experiments. *J. Phys.: Conf. Ser.*, 125:012044, 2008.

[29] T. Kato and T. Kataoka. Circuit analysis by a new multirate method. *Electrical Engineering in Japan*, 126(4):55–62, 1999.

[30] A. Kværnø. Stability of multirate Runge-Kutta schemes. *International Journal of Differential Equations and Applications*, 1(1):97–105, 2000.

[31] A. Kværnø and P. Rentrop. Low order multirate Runge-Kutta methods in electric circuit simulation, 1999.

[32] J.M. Finn L. Chacón, D. A. Knoll. An implicit, nonlinear reduced resistive MHD solver. *Journal of Computational Physics*, 178(1):15–36, 2002.

[33] A. Logg. Multi-adaptive Galerkin methods for ODEs I. *SIAM Journal on Scientific Computing*, 24(6):1879–1902, 2003.

[34] A. Logg. Multi-adaptive Galerkin methods for ODEs II: Implementation and applications. *SIAM Journal on Scientific Computing*, 25(4):1119–1141, 2003.

[35] A. Logg. Multi-adaptive Galerkin methods for ODEs III: *A priori* estimates. *SIAM Journal on Numerical Analysis*, 43(6):2624–2646, 2006.

[36] R. Samtaney M. F. Adams and A. Brandt. Toward textbook multigrid efficiency for fully implicit resistive magnetohydrodynamics. *Journal of Computational Physics*, 229(18):6208–6219, 2010.

[37] Y. Maday and G. Turinici. The parareal in time iterative solver: a further direction to parallel implementation. In TimothyJ. Barth, Michael Griebel, DavidE. Keyes, RistoM. Nieminen, Dirk Roose, Tamar Schlick, Ralf Kornhuber, Ronald Hoppe, Jacques Périaux, Olivier Pironneau, Olof Widlund, and Jinchao Xu, editors, *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 441–448. Springer Berlin Heidelberg, 2005.

[38] E. Faou N. Crouseilles, L. Einkemmer. A Hamiltonian splitting for the Vlasov-Maxwell system. *JCP*, 283:224–240, 2015.

[39] H. Podhaisky R. Weiner, B.A. Schmitt. Rowmap–a ROW-code with Krylov techniques for large stiff ODEs,. *Appl. Numer. Math.*, 25:303–319, 1997.

[40] V. Rao and A. Sandu. An adjoint-based scalable algorithm for time-parallel integration. *Journal of Computational Science*, 5(2):76 – 84, 2014. Empowering Science through Computing + BioInspired Computing.

[41] D. R. Reynolds, D. J. Gardner, A. C.Hindmarsh, C. S. Woodward, and J. M. Sexton. User documentation for ARKode v1.0.1. Technical Report UCRL-SM-668082, Lawrence Livermore National Laboratory, 2015.

[42] G.R.W. Quispel R.I. McLachlan. Splitting methods. *Acta Numer.*, 11:341–434, 2002.

[43] J.R. Rice. Split Runge-Kutta methods for simultaneous equations. *Journal of Research of the National Institute of Standards and Technology*, 64(B):151–170, 1960.

[44] J. Sand and K. Burrage. A Jacobi waveform relaxation method for ODEs. *SIAM Journal on Scientific Computing*, 20(2):534–552, 1998.

[45] A. Sandu and M. Günther. A generalized-structure approach to additive Runge-Kutta methods. *SIAM Journal on Numerical Analysis*, 53(1):17–42, 2015.

[46] C.R. Sovinec, A.H. Glasser, T.A. Gianakon, D.C. Barnes, R.A. Nebel, S.E. Kruger, S.J. Plimpton, A. Tarditi, M.S. Chu, and the NIMROD Team. Nonlinear magnetohydrodynamics with high-order finite elements. *J. Comp. Phys.*, 195:355, 2004.

[47] M. Tokman and P. M. Bellan. Three?dimensional model of the structure and evolution of coronal mass ejections. *Astrophys. J.*, 567:1202–210, 2002.

[48] P. Tranquilli and A. Sandu. Exponential-Krylov methods for ordinary differential equations. *Journal of Computational Physics*, 278:31–46, 2014.

[49] P. Tranquilli and A. Sandu. Rosenbrock-Krylov methods for large systems of differential equations. *SIAM Journal on Scientific Computing*, 36(3):A1313–A1338, 2014.

[50] C. S. Woodward, A. C. Hindmarsh, D. R. Reynolds, R. Serban, and L. E. Banks. SUNDIALS (SUite of Nonlinear and DIfferential/ALgebraic Solvers). http://www.llnl.gov/casc/sundials.

[51] H. Zhang and A. Sandu. FATODE: A library for forward, adjoint and tangent linear integration of stiff systems. *SIAM Journal on Scientific Computing*, 36(5):C504–C523, 2014.

[52] H. Zhang, A. Sandu, and S. Blaise. Partitioned and implicit-explicit general linear methods for ordinary differential equations. *Journal of Scientific Computing*, 61(1):119–144, 2014.

[53] E. Zharovsky, A. Sandu, and H. Zhang. A class of IMEX two-step Runge-Kutta methods. *SIAM Journal on Numerical Analysis*, 53(1):321–341, 2015.