L

Large-Scale Computing for Molecular Dynamics Simulation

Aiichiro Nakano, Rajiv K. Kalia, Ken-ichi Nomura, and Priya Vashishta
Department of Computer Science, Department of Physics and Astronomy, and Department of Chemical Engineering and Materials Science, University of Southern California, Los Angeles, CA, USA

Mathematics Subject Classification

65Y05; 70F10; 81V55

Synonyms

High performance computing for atomistic simulation

Short Definition

Large-scale computing for molecular dynamics simulation combines advanced computing hardware and efficient algorithms for atomistic simulation to study material properties and processes encompassing large spatiotemporal scales.

Description

Material properties and processes are often dictated by complex dynamics of a large number of atoms. To understand atomistic mechanisms that govern macroscopic material behavior, large-scale molecular dynamics (MD) simulations [1] involving multibillion atoms are performed on parallel supercomputers consisting of over 10^5 processors [2]. In addition, special-purpose computers are built to enable long-time MD simulations extending millisecond time scales (or 10^{12} time steps using a time discretization unit of 10^{-15} s) [3] (for extending the time scale, see also ▸ Transition Pathways, Rare Events and Related Questions). Key enabling technologies for such large spatiotemporal-scale MD simulations are efficient algorithms to reduce the computational complexity and parallel-computing techniques to map these algorithms onto parallel computers.

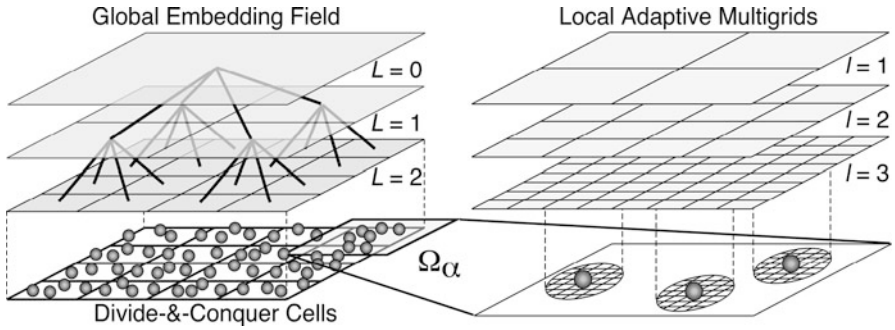
Linear-Scaling Molecular-Dynamics Simulation Algorithms

The MD approach (see also ▸ Applications to Real Size Biological Systems) follows the time evolution of the positions, $\mathbf{r}^N = \{\mathbf{r}_i | i = 1,\ldots,N\}$, of N atoms by solving coupled ordinary differential equations [1]:

$$m_i \frac{d^2}{dt^2}\mathbf{r}_i = -\frac{\partial}{\partial \mathbf{r}_i} E\left(\mathbf{r}^N\right), \tag{1}$$

where t is the time, and $\mathbf{r}_i$ and m_i are the position and mass of the i-th atom, respectively. Atomic force law is mathematically encoded in the interatomic potential energy $E(\mathbf{r}^N)$, and key to large-scale MD simulations is, foremost, linear-scaling algorithms that

B. Engquist (ed.), *Encyclopedia of Applied and Computational Mathematics*,
DOI 10.1007/978-3-540-70529-1



Large-Scale Computing for Molecular Dynamics Simulation, Fig. 1 Schematic of an embedded divide-and-conquer algorithm [2]. (*Left*) The physical space is subdivided into spatially localized cells, with local atoms constituting subproblems (*bottom*), which are embedded in a global field (*shaded*) solved with a tree-based algorithm. (*Right*) To solve the subproblem in domain Ω_α in the divide-and-conquer density functional theory algorithm, coarse multigrids (*gray*) are used to accelerate iterative solutions on the original real-space grid (corresponding to the grid refinement level, $l = 3$). The *bottom* panel shows fine grids adaptively generated near the atoms (*spheres*) to accurately operate the ionic pseudopotentials on the electronic wave functions

compute $E(\mathbf{r}^N)$ in $O(N)$ time. This algorithmic and mathematical challenge is often addressed based on data-locality principles. An example is embedded divide-and-conquer (EDC) algorithms, in which the physical system is divided into spatially localized computational cells and these cells are embedded in a global mean field that is computed efficiently with tree-based algorithms (Fig. 1) [2].

There exist a hierarchy of MD simulation methods with varying accuracy and computational complexity. In classical MD simulation, $E(\mathbf{r}^N)$ is often an analytic function $E_{\mathrm{MD}}(\{\mathbf{r}_{ij}\},\{\mathbf{r}_{ijk}\},\{\mathbf{r}_{ijkl}\})$ of atomic pair, $\mathbf{r}_{ij}$, triplet, $\mathbf{r}_{ijk}$, and quadruplet, $\mathbf{r}_{ijkl}$, positions, where the hardest computation is the evaluation of the long-range electrostatic interaction between all atomic pairs. The fast multipole method (FMM) algorithm reduces the $O(N^2)$ computational complexity of the resulting N-body problem to $O(N)$ [4]. In the FMM, the physical system is recursively divided into subsystems to form an octree data structure, and the electrostatic field is computed recursively on the octree with $O(N)$ operations, while maintaining spatial locality at each recursion level. In addition to computing the electrostatic potential and forces, the FMM can be used to compute atomistic stress tensor components based on a complex charge method [5]. Furthermore, a space-time multiresolution MD approach [2] utilizes temporal locality through multiple time stepping, which uses different force-update schedules for different force components [6, 7]. Specifically, forces from neighbor atoms are computed at every MD step, whereas forces from farther atoms are updated less frequently.

To simulate the breakage and formation of chemical bonds with moderate computational costs, various reactive molecular dynamics (RMD) simulation methods have been developed [2]. In RMD, the interatomic potential energy $E_{\mathrm{RMD}}(\mathbf{r}^N,\{q_i\},\{B_{ij}\})$ typically depends on the atomic charges $\{q_i \mid i = 1,\ldots,N\}$ and the chemical bond orders B_{ij} between atomic pairs (i, j), which change dynamically adapting to the local environment to describe chemical reactions. To describe charge transfer, RMD uses a charge equilibration scheme, in which atomic charges are determined at every MD step to minimize the electrostatic energy with the charge-neutrality constraint. This variable N-charge problem amounts to solving a dense linear system of equations, which requires $O(N^3)$ operations. A fast RMD algorithm uses FMM to perform the required matrix-vector multiplications with $O(N)$ operations [2]. It further utilizes the temporal locality of the solutions to reduce the amortized computational cost averaged over simulation steps to $O(N)$. To accelerate the convergence, a multilevel preconditioned conjugate-gradient (MPCG) method splits the Coulomb-interaction matrix into short- and long-range parts and uses the sparse short-range matrix as a preconditioner [8]. The extensive use of the sparse preconditioner enhances the data locality and thereby improves the computational efficiency.

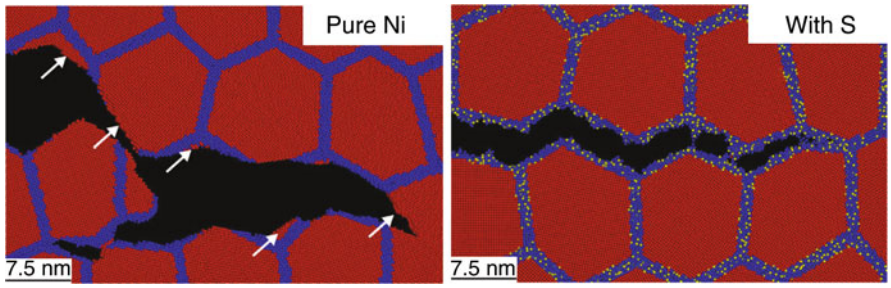
In quantum molecular dynamics (QMD) simulation, the interatomic potential energy is computed quantum mechanically [9]. One approach to approximately solve the resulting exponentially complex quantum N-body problem is density functional theory (DFT, see ▸ Density Functional Theory), which reduces the complexity to $O(N^3)$ by solving M one-electron problems self-consistently instead of one M-electron problem (the number of electrons M is on the order of N). The DFT problem can be formulated as a minimization of the energy functional $E_{\mathrm{QMD}}(\mathbf{r}^N, \psi^M)$ with respect to electronic wave functions (or Kohn-Sham orbitals), $\psi^M(\mathbf{r}) = \{\psi_n(\mathbf{r}) \mid n = 1,\ldots,M\}$ subject to orthonormality constraints (see ▸ Fast Methods for Large Eigenvalues Problems for Chemistry and ▸ Numerical Analysis of Eigenproblems for Electronic Structure Calculations). Various linear-scaling DFT algorithms have been proposed [10, 11] based on a data locality principle called quantum nearsightedness [12] (see ▸ Linear Scaling Methods). Among them, divide-and-conquer density functional theory (DC-DFT) [13] is highly scalable beyond 10^5 processors [2]. In the DC-DFT algorithm, the physical space is a union of overlapping domains, $\Omega = \Sigma_\alpha \Omega_\alpha$ (Fig. 1), and physical properties are computed as linear combinations of domain properties that in turn are computed from local electronic wave functions. For DFT calculation within each domain, one implementation uses a real-space approach based on adaptive multigrids [2] (see ▸ Finite Difference Methods). Similar data-locality and divide-and-conquer concepts have been applied to design $O(N)$ algorithms for high-accuracy QM methods [14], including the fragment molecular orbital method [15]. A major advantage of the EDC simulation algorithms is the ease of codifying error management. The EDC algorithms often have a well-defined set of localization parameters, with which the computational cost and the accuracy are controlled. For example, the total energy computed with the DC-DFT algorithm converges rapidly as a function of its localization parameter (i.e., the depth of the buffer layer to augment each domain for avoiding artificial boundary effects). The DC-DFT-based QMD algorithm has also overcome the energy drift problem, which plagues most $O(N)$ DFT-based QMD algorithms, especially with large basis sets ($>10^4$ unknowns per electron, necessary for the transferability of accuracy) [2].

Scalable Parallel Computing

To perform large-scale MD simulations, it is necessary to decompose the computation in the $O(N)$ MD algorithms to subtasks and map them onto parallel computers [1]. A parallel computer in general consists of a number of compute nodes interconnected via a communication network [16]. Within each node, multicore processors, each consisting of simpler processors called cores, share common memory [17]. There are several schemes for mapping MD algorithms onto parallel computers [1]. For large granularity (i.e., the number of atoms per processor, $N/P > 10^2$), spatial decomposition is optimal, where each processor is assigned a spatial subsystem and is responsible for the computation of the forces on the atoms within its spatial subsystem. For finer granularity ($N/P \sim 1$), on the other hand, force decomposition (i.e., force computations are divided among processors) and other hybrid decomposition schemes become more efficient [18–20]. Parallelization schemes also include load-balancing capability [21]. For irregular data structures, the number of atoms assigned to each processor varies significantly, and this load imbalance degrades the parallel efficiency. Load balancing can be stated as an optimization problem, in which we minimize the load-imbalance cost as well as the size and the number of messages.

Parallel efficiency is defined as the speedup achieved using P processors over one processor, divided by P. Parallel efficiency over 0.9 has been achieved on a cluster of multicore compute nodes with $P > 10^5$ combining a hierarchy of parallelization schemes [22], including:

1. Internode parallelization based on message passing [23], in which independent processes (i.e., running programs) on different nodes exchange messages over a network.
2. Intra-node (inter-core), multithreading parallelization [24] on multicore central processing units (CPUs) as well as on hardware accelerators such as graphics processing units (GPUs) [25], in which multiple threads (i.e., processes sharing certain hardware resources such as memory) run concurrently on multiple cores within each compute node.
3. Intra-core, single-instruction multiple data (SIMD) parallelization [16, 26], in which a single instruction



Large-Scale Computing for Molecular Dynamics Simulation, Fig. 2 Close-ups of fracture simulations for nanocrystalline nickel without and with amorphous sulfide grain-boundary phases, where *red*, *blue* and *yellow* colors represent nickel atoms inside grains (>0.5 nm from grain boundaries), nickel atoms within 0.5 nm from grain boundaries, and sulfur atoms, respectively. The figure shows a transition from ductile, transgranular tearing (*left*) to brittle, intergranular cleavage (*right*). *White arrows* point to transgranular fracture surfaces

executes on multiple operands concurrently in a vector processing unit within each core.

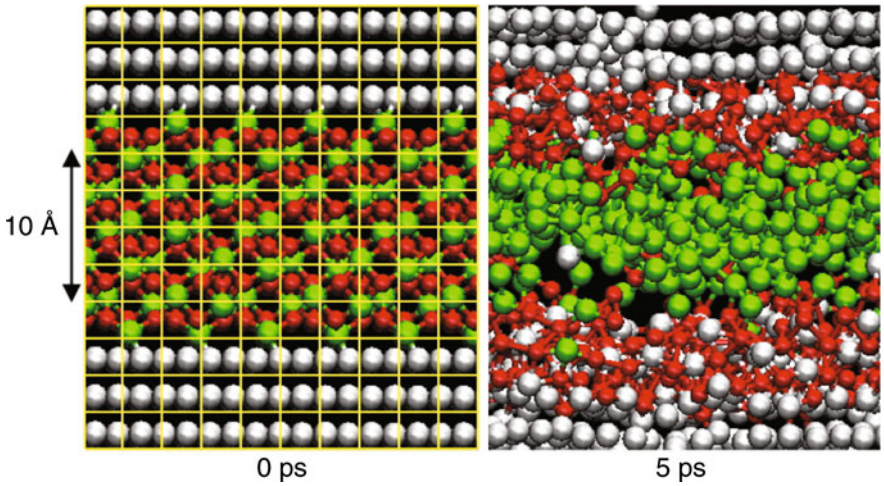
A number of software packages have been developed for parallel MD simulations. Widely available packages for MD include Amber (http://ambermd.org), Desmond (http://www.schrodinger.com/products/14/3), DL_POLY (http://www.cse.scitech.ac.uk/ccg/software/DL_POLY), Gromacs (http://www.gromacs.org), and NAMD (http://www.ks.uiuc.edu/Research/namd). Parallel implementations of MD and RMD are found in LAMMPS (http://lammps.sandia.gov). DFT-based QMD packages include CP2K (http://cp2k.berlios.de), Quantum ESPRESSO (http://www.quantum-espresso.org), SIESTA (http://www.icmab.es/siesta), and VASP (http://cms.mpi.univie.ac.at/vasp), along with those specialized on linear-scaling DFT approaches such as Conquest (http://hamlin.phys.ucl.ac.uk/NewCQWeb/bin/view), ONETEP (http://www.tcm.phy.cam.ac.uk/onetep), and OpenMX (http://www.openmx-square.org). Finally, quantum-chemical approaches to QMD are implemented in, e.g., GAMESS (http://www.msg.ameslab.gov/gamess), Gaussian (http://www.gaussian.com), and NWChem (http://www.nwchem-sw.org).

Large-Scale Molecular Dynamics Applications

Using scalable parallel MD algorithms, computational scientists have performed MD simulations involving billion-to-trillion atoms on massively parallel supercomputers consisting of over 10^5 processors to study various material processes such as instability at fluid interfaces and shock-wave propagation [27, 28].

The largest RMD simulations include 48 million-atom simulation of solute segregation-induced embrittlement of metal [29]. This simulation answers a fundamental question encompassing chemistry, mechanics, and materials science: How a minute amount of impurities segregated to grain boundaries of a material essentially alters its fracture behavior. A prime example of such grain-boundary mechano-chemistry is sulfur segregation-induced embrittlement of nickel, which is an important problem for the design of the next-generation nuclear reactors to address the global energy problem. Experiments have demonstrated an essential role of sulfur segregation-induced grain boundary amorphization on the embrittlement, but the central question remains unsolved: Why does amorphization cause embrittlement? The RMD simulation (Fig. 2) establishes the missing link between sulfur-induced intergranular amorphization and embrittlement [29]. The simulation results reveal that an order-of-magnitude reduction of grain-boundary shear strength due to amorphization, combined with tensile-strength reduction, allows the crack tip to always find an easy propagation path. This mechanism explains all experimental observations and elucidates the experimentally found link between grain-boundary amorphization and embrittlement.

While large-scale electronic structure calculations involving over 10^4 atoms have been re-



Large-Scale Computing for Molecular Dynamics Simulation, Fig. 3 Snapshots of the atomic configuration during DC-DFT-based QMD simulation of thermite reaction, where *green*, *red*, and *gray* spheres show the positions of Fe, O and Al atoms, respectively. *Yellow* meshes at time 0 ps show the nonoverlapping cores used by the DC-DFT algorithm

ported (see ▸ Large-Scale Electronic Structure and Nanoscience Calculations), QMD simulations extending a long trajectory are usually limited to thousands of atoms. Examples of systems studied by large QMD simulations include metals under extreme conditions [30], reaction of nanoenergetic materials [31], and ionic conductivity in batteries [32]. Chemical reactions in energetic materials with nanometer-scale microstructures (or nanoenergetic materials) are very different from those in conventional energetic materials. For example, in conventional thermite materials made of aluminum and iron oxide, the combustion front propagates at a speed of ∼cm/s. In nanothermites of aluminum nanoparticles embedded in iron oxide, the combustion speed is accelerated to ∼km/s. Such rapid reactions cannot be explained by conventional diffusion-based mechanisms. DC-DFT-based QMD simulation has been performed to study electronic processes during thermite reaction [31]. Here, the reactants are Al and Fe_2O_3, and the products are Al_2O_3 and Fe (Fig. 3). The simulation results reveal a concerted metal-oxygen flip mechanism that enhances mass diffusion and reaction rate at the metal/oxide interface. This mechanism leads to novel two-stage reactions, which explain experimental observation in thermite nanowire arrays.

Conclusions

Large-scale MD simulations to encompass large spatiotemporal scales are enabled with scalable algorithmic and parallel-computing techniques based on spatiotemporal data-locality principles. The spatiotemporal scale covered by MD simulation on a sustained petaflops computer (which can operate 10^{15} floating-point operations per second) per day is estimated as $NT \sim 2$ (e.g., $N = 2$ billion atoms for $T = 1$ ns) [22], which continues to increase on emerging computing architectures.

References

1. Rapaport, D.C.: The art of molecular dynamics simulation. 2nd edn. Cambridge University Press, Cambridge (2004)
2. Nakano, A., et al.: De novo ultrascale atomistic simulations on high-end parallel supercomputers. Int. J. High Perform. Comput. Appl. **22**, 113 (2008)
3. Shaw, D.E., et al.: Anton, a special-purpose machine for molecular dynamics simulation. Commun. ACM. **51**, 91 (2008)
4. Greengard, L., Rokhlin, V.: A fast algorithm for particle simulations. J. Comput. Phys. **73**, 325 (1987)
5. Ogata S., et al.: Scalable and portable implementation of the fast multipole method on parallel computers. Comput. Phys. Commun. **153**, 445 (2003)
6. Martyna, G.J., et al.: Explicit reversible integrators for extended systems dynamics. Mol. Phys. **87**, 1117 (1996)
7. Schlick, T., et al.: Algorithmic challenges in computational molecular biophysics. J. Comput. Phys. **151**, 9 (1999)
8. Nakano, A.: Parallel multilevel preconditioned conjugate-gradient approach to variable-charge molecular dynamics. Comput. Phys. Commun. **104**, 59 (1997)
9. Car, R., Parrinello, M.: Unified approach for molecular dynamics and density functional theory. Phys. Rev. Lett. **55**, 2471 (1985)
10. Goedecker, S.: Linear scaling electronic structure methods. Rev. Mod. Phys. **71**, 1085 (1999)

11. Bowler, D.R., et al.: Introductory remarks: Linear scaling methods - Preface. J. Phys. Condens. Matter **20**, 290301 (2008)
12. Kohn, W.: Density functional and density matrix method scaling linearly with the number of atoms. Phys. Rev. Lett. **76**, 3168 (1996)
13. Yang, W.: Direct calculation of electron-density in density-functional theory. Phys. Rev. Lett. **66**, 1438 (1991)
14. Goedecker, S., Scuseria, G.E.: Linear scaling electronic structure methods in chemistry and physics. Comput. Sci. Eng. **5**, 14 (2003)
15. Kitaura, K., et al.: Fragment molecular orbital method: an approximate computational method for large molecules. Chem. Phys. Lett. **313**, 701 (1999)
16. Grama, A., et al.: Introduction to parallel computing. 2nd edn. Addison Wesley, Harlow (2003)
17. Asanovic, K., et al.: The landscape of parallel computing research: A view from Berkeley. University of California, Berkeley (2006)
18. Plimpton, S.J.: Fast parallel algorithms for short-range molecular dynamics. J. Comput. Phys. **117**, 1 (1995)
19. Kale, L., et al.: NAMD2: greater scalability for parallel molecular dynamics. J. Comput. Phys. **151**, 283 (1999)
20. Shaw, D.E.: A fast, scalable method for the parallel evaluation of distance-limited pairwise particle interactions. J. Comput. Chem. **26**, 1318 (2005)
21. Devine, K.D., et al.: New challenges in dynamic load balancing. Appl. Num. Math. **52**, 133 (2005)
22. Nomura, K., et al.: A metascalable computing framework for large spatiotemporal-scale atomistic simulations. Proceedings of International Parallel and Distributed Processing Symposium IPDPS 2009, IEEE, Rome (2009)
23. Gropp, W., Lusk, E., Skjellum, A.: Using MPI. 2nd edn. MIT, Cambridge (1999)
24. Chapman, B., Jost, G., van der Pas, R.: Using OpenMP. MIT, Cambrige (2007)
25. Phillips, J.C., Stone, J.E.: Probing biomolecular machines with graphics processors. Commun. ACM. **52**, 34 (2009)
26. Peng, L., et al.: Exploiting hierarchical parallelisms for molecular dynamics simulation on multicore clusters. J. Supercomput. **57**, 20 (2011)
27. Glosli, J.N., et al.: Extending stability beyond CPU millennium: a micron-scale atomistic simulation of Kelvin-Helmholtz instability. Proceedings of Supercomputing (SC07), ACM, New York (2007)
28. Germann, T.C., Kadau, K.: Trillion-atom molecular dynamics becomes a reality. Int. J. Mod. Phys. C. **19**, 1315 (2008)
29. Chen, H.P., et al.: Embrittlement of metal by solute segregation-induced amorphization. Phys. Rev. Lett. **104**, 155502 (2010)
30. Gygi, F., et al.: Large-scale first-principles molecular dynamics simulations on the BlueGene/L platform using the Qbox code. Proceedings of Supercomputing 2005 (SC05), ACM, Washington, DC (2005)
31. Shimojo, F., et al.: Enhanced reactivity of nanoenergetic materials: A first-principles molecular dynamics study based on divide-and-conquer density functional theory. Appl. Phys. Lett. **95**, 043114 (2009)
32. Ikeshoji, T., et al.: Fast-ionic conductivity of Li^+ in $LiBH_4$. Phys. Rev. B. **83**, 144301 (2011)

Large-Scale Electronic Structure and Nanoscience Calculations

Juan C. Meza[1] and Chao Yang[2]
[1]School of Natural Sciences, University of California, Merced, CA, USA
[2]Computational Research Division, MS-50F, Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Synonyms

Electronic structure; Kohn-Sham equations; Nanoscience

Definition

The electronic structure of an atomic or molecular system can yield insights into many of the electrical, optical, and mechanical properties of materials. Real-world problems, such as nanostructures, are difficult to study, however, as many algorithms do not scale well with system size requiring new techniques better suited to large systems.

Overview

The electronic structure of a system can be described by the solution of a quantum many-body problem described by the Schrödinger equation: $H\Psi = \Psi E$, where H is a many-body Hamiltonian operator that describes the kinetic energy and the Coulomb interaction between electron–electron and electron–nucleus pairs, Ψ is a many-body wavefunction, and E is the total energy level of the system.

One popular approach for solving these types of problems relies on reformulating the original problem in terms of a different basic variable, the charge density, and using single-particle wavefunctions to replace the many-body wavefunctions. This approach is known as *Kohn-Sham density functional theory* (DFT) and can be viewed as a search for the minimizer of a certain functional of the charge density.

From a mathematical viewpoint, it can be shown that the first order necessary optimality condition (Euler-Lagrange equation) for minimizing the Kohn-Sham energy yields the following set of nonlinear eigenvalue equations (known as the Kohn-Sham equations): $H(\rho)\psi_i = \epsilon_i \psi_i,\ i = 1, 2, \ldots, n_e$, where $H(\rho) = -\Delta + V_{\text{ion}} + V_H(\rho) + V_{xc}(\rho)$, V_{ion}, V_H, and V_{xc} are the ionic, electron–electron (Hartree), and exchange-correlation potentials, and n_e is the number of electrons. Here $\rho(r)$ is the *electron charge density* defined by $\rho(r) = \sum_{i=1}^{n_e} |\psi_i(r)|^2$.

Although these equations contain far fewer degrees of freedom compared to the many-body Schrödinger equation, they are more difficult in terms of their mathematical structures. The most popular method to solve the Kohn-Sham equations is the *Self-Consistent Field* (SCF) iteration. The computational complexity of most of the existing algorithms is $\mathcal{O}(n_e^3)$, which can limit their applicability to large nanoscience problems. We will describe briefly some of the general strategies one may use to reduce the overall complexity of these algorithms and where the challenges lie in doing this.

The Kohn-Sham Map and the SCF Iteration

A useful concept for analyzing algorithms applied to large-scale Kohn-Sham problems is the following alternative definition of the charge density:

$$\rho = \text{diag}\left[\hat{X} g_\beta(\hat{\Lambda} - \mu)\hat{X}^*\right] = \text{diag}\left[g_\beta(H(\rho) - \mu)\right], \tag{1}$$

where $\hat{X} \in \mathbb{C}^{n\times n}$ contains the full set of eigenvectors of a discretized Kohn-Sham Hamiltonian, $\hat{\Lambda}$ is a diagonal matrix containing the corresponding eigenvalues of the Hamiltonian, $g_\beta(\lambda)$ is the Fermi-Dirac function:

$$g_\beta(\lambda, \mu) = \frac{2}{1 + \exp(\beta(\lambda - \mu))} = 1 - \tanh\left(\frac{\beta}{2}(\lambda - \mu)\right), \tag{2}$$

where β is a parameter chosen in advance and proportional to the inverse of the temperature, and μ is the chemical potential, which is chosen so that $\text{trace}\left[g_\beta(H(\rho) - \mu I)\right] = n_e$. At zero temperature, $\beta = \infty$ and (2) reduces to a step function that drops from 1 to 0 at μ.

Equation 1 defines a self-consistent map from ρ to itself. This map is sometimes referred to as the *Kohn-Sham map*. Because the Jacobian of this map is difficult to compute or invert, a practical approach for finding the fixed point of the Kohn-Sham map is to apply a Broyden type Quasi-Newton algorithm to solve (1) iteratively. This is generally known as a SCF iteration. The convergence of a SCF iteration depends largely on the choice of an effective Broyden updating scheme for approximating the Jacobian at each iteration. Such a scheme is known as *charge mixing* in the physics literature.

The dominant cost of a SCF iteration is the evaluation of the Kohn-Sham map, that is, the right hand side of (1). The most widely used technique for performing such an evaluation is to partially diagonalize $H(\rho)$ and compute its n_e smallest eigenvalues and the corresponding eigenvectors. For large-scale problems, the eigenvalue problem is often solved by an iterative method such as a Lanczos or Davidson algorithm.

An alternative approach is to treat the eigenvalue problem as a constrained minimization problem and apply an iterative minimization algorithm such as the locally optimal block preconditioned conjugate gradient (LOBPCG) algorithm [6] to minimize the trace of X^*HX subject to the orthonormality constraint $X^*X = I$. Because an effective preconditioner can be used in this approach, it is often more efficient than a Lanczos-based algorithm.

Both the Lanczos and the LOBPCG algorithms require performing orthogonalization among at least n_e basis vectors, which for large n_e incurs a cost of $\mathcal{O}(n_e^3)$. To reduce the frequency of orthogonalization, one may apply a simple subspace iteration to $p^{(k)}(H)$, where $p^{(k)}(\lambda)$ is a polynomial constructed at the kth SCF iteration to amplify the spectral components associated with the desired eigenvalues of H while filtering out the unwanted components. Although this algorithm may use approximately the same number of matrix-vector multiplications as that used in a Lanczos, Davidson, or LOBPCG algorithm, the basis orthogonalization cost is much lower (but not completely eliminated) for large n_e, as is shown in [20].

A recently developed method [9] for evaluating the Kohn-Sham map without resorting to performing a spectral decomposition of H relies on using a rational approximation to $g_\beta(\lambda - \mu)$ to compute the diagonal

entries of $g_\beta(H-\mu I)$ directly. The rational approximation to $g_\beta(\lambda-\mu)$ has the form:

$$g_\beta(\lambda-\mu) \approx \sum_{j=1}^{n_p} \mathrm{Im}\left[\frac{\omega_j}{\lambda - z_j}\right],$$

where z_j and ω_j are carefully chosen poles and weighting factors that minimize the approximation error. The number of poles required (n_p) is typically less than a hundred. Although computing $g_\beta(H-\mu I)$ would require us to compute $(H - z_i I)^{-1}$, which is likely to be completely dense, for n_p complex poles z_i, a significant amount of savings can be achieved if we only need the diagonal elements of $g_\beta(H-\mu I)$. Instead of computing the entire matrix $(H - z_i I)^{-1}$, one only needs to compute its diagonal. This task can be accomplished by using a special algorithm which we refer to as *selected inversion* [10, 11]. The complexity of selected inversion is $\mathcal{O}(n_e)$ for quasi-1D problems (e.g., nanotubes and nanowires), $\mathcal{O}(n_e^{3/2})$ for quasi-2D problems (e.g., graphene), and $\mathcal{O}(n_e^2)$ for general 3D problems.

Solving the Kohn-Sham Problem by Constrained Minimization

The Kohn-Sham problem can also be solved by minimizing the Kohn-Sham total energy directly. In this case, we seek to find

$$\min_{X^*X=I_{n_e}} E_{\mathrm{tot}}(X) \equiv \mathrm{trace}[X^*(\frac{1}{2}L + \hat{V}_{\mathrm{ion}})X] + \frac{1}{2}\rho^T L^\dagger \rho + \rho^T \epsilon_{xc}(\rho), \quad (3)$$

where $L \in \mathbb{R}^{n\times n}$ and $V_{\mathrm{ion}} \in \mathbb{R}^{n\times n}$ are matrix representations of finite dimensional approximations to the Laplacian and the ionic potential operator respectively. The matrix $L^\dagger$ is either the inverse or the pseudoinverse of L depending on the boundary condition imposed in the continuous model, and $X \in \mathbb{C}^{n\times n_e}$ contains approximate single-particle wavefunctions as its columns.

This approach has been attempted by several researchers [8, 14]. Most of the proposed methods treat the minimization of the total energy and constraint satisfaction separately. A more efficient direct constrained minimization (DCM) algorithm was proposed in [17, 18]. In this algorithm, the search direction and the step length are determined simultaneously from a subspace that consists of the existing wave functions $X^{(i)}$, the gradient of the Lagrangian, and the search direction produced in the previous iteration. A special strategy is employed to minimize the total energy within the search space, while maintaining the orthonormality constrained required for $X^{(i+1)}$. Solving the subspace minimization problem is equivalent to solving a nonlinear eigenvalue problem of a much smaller dimension.

Linearly Scaling Algorithms

Most of the algorithms discussed above can be implemented efficiently on modern high-performance parallel computers. However, for large nanoscience problems that consist of more than tens of thousands of atoms, many of these existing algorithms are still quite demanding in terms of computational resources. In recent years, there has been a growing level of interest in developing linearly scaling methods [1, 2, 4, 5, 12, 13, 16, 19] for electronic structure calculations. For insulators and semiconductors, the computational complexity of these algorithms indeed scales linearly with respect to n_e or the number of atoms. However, it is rather challenging to develop a linearly scaling algorithm for metallic systems for reasons that we will give below. In general, a linear scaling algorithm should meet the following criteria:

- The complexity for evaluating the Kohn-Sham map must be $\mathcal{O}(n_e)$.
- The total number of SCF iterations must be relatively small compared to n_e.

While most of the existing research efforts focus exclusively on the first criterion, we believe the second criterion is equally important.

All existing linearly scaling algorithms exploit the locality property of the single-particle wavefunctions (orbitals) or density matrices to reduce the complexity of the charge density (Kohn-Sham map) evaluation. The locality property has its roots in the "nearsightedness" principle first suggested by Kohn [7] and further investigated in [15]. In mathematical terms, the locality property implies that the invariant subspace spanned by the smallest n_e eigenvectors can be represented by a set of basis vectors that have local nonzero support (i.e., each basis vector has a relatively small number

of nonzero elements.), or the density matrix $D = g_\beta(H(\rho) - \mu I)$ is diagonally dominant, and the off-diagonal entries of the matrix decay rapidly to zero away from the diagonal. As a result, there are three main classes of linearly scaling methods.

In the first class of methods, one relaxes the orthonormality constraint of the single-particle wavefunctions but requires them to have localized nonzero support. As a result, the Kohn-Sham map can be evaluated by solving a sparse generalized eigenvalue problem. An iterative method such as the localized subspace iterations (LSI) [3] can be used to compute the desired invariant subspace. Because each basis vector of the invariant subspace is forced to be sparse, the matrix-vector multiplication used in such an algorithm can be evaluated efficiently with a complexity of $\mathcal{O}(n_e)$. More importantly, because such an algorithm does not perform basis reorthogonalization, it does not incur the $\mathcal{O}(n_e^3)$ cost of conventional eigensolvers.

The second class of methods employs a *divide-and-conquer* principle originally suggested in [19] to divide the problem into several subproblems defined on smaller subregions of the material domain. From a mathematical viewpoint, these are domain decomposition methods. A similar approach is used in the recently developed linear-scaling three-dimensional fragment (LS3DF) method [16]. These methods require local solutions to be patched together in a nontrivial way to preserve the total charge and to eliminate charge transfer between different regions.

The third class of linearly scaling methods relies on using either polynomial or rational approximations of $D = g_\beta(H - \mu I)$ and truncation techniques that ignore small off-diagonal entries in D to reduce the complexity of the Kohn-Sham map evaluation to $\mathcal{O}(n_e)$. It is important to note that the number of terms used in the polynomial or rational approximation to $g_\beta(H - \mu I)$ must be small enough in order to achieve linear scaling. For insulators and semiconductors in which the gap between the occupied and unoccupied states is relatively large, this is generally not difficult to achieve. For metallic systems that have no band gap, one may need a polynomial of very high degree to approximate $g_\beta(H - \mu I)$ with sufficient accuracy. It is possible to accurately approximate $g_\beta(H - \mu I)$ using recently developed pole expansion techniques [9] with less than 100 terms even when the band gap is very small. However, since the off-diagonal elements of D decay slowly to zero for metallic systems, the evaluation of the Kohn-Sham map cannot be performed in $\mathcal{O}(n_e)$ without losing accuracy at low temperature.

Linearly scaling algorithms can also be designed to minimize the total energy directly. To achieve linear scaling, the total energy minimization problem is reformulated as an unconstrained minimization problem. Instead of imposing the orthonormality constraint of the single-particle wavefunctions, we require them to have localized support. Such localized orbitals allow the objective and gradient calculations to be performed with $\mathcal{O}(n_e)$ complexity. The original version of orbital minimization methods uses direct truncations of the orbitals. They are known to suffer from the possibility of being trapped at a local minimizer [4]. The presence of a large number of local minimizers in this approach is partially due to the fact that direct truncation tends to destroy the invariance property inherent in the Kohn-Sham DFT model, and introduces many local minima in the Kohn-Sham energy landscape. This problem can be fixed by applying a localization procedure prior to truncation.

Cross-References

- Density Functional Theory
- Hartree–Fock Type Methods
- Schrödinger Equation for Chemistry
- Self-Consistent Field (SCF) Algorithms

References

1. Barrault, M., Cances, E., Hager, W.W., Bris, C.L.: Multilevel domain decomposition for electronic structure calculations. J. Comput. Phys. **222**, 86–109 (2006)
2. Galli, G.: Linear scaling methods for electronic structure calculations and quantum molecular dynamics simulations. Curr. Opin. Solid State Mater. Sci. **1**, 864–874 (1996)
3. Garcia-Cervera, C., Lu, J., Xuan, Y., Weinan, E.: A linear scaling subspace iteration algorithm with optimally localized non-orthogonal wave functions for Kohn-Sham density functional theory. Phys. Rev. B **79**(11), 115110 (2009)
4. Goedecker, S.: Linear scaling electronic structure methods. Rev. Mod. Phys. **71**(4), 1085–1123 (1999)
5. Kim, J., Mauri, F., Galli, G.: Total energy global optimizations using non-orthogonal localized orbitals. Phys. Rev. B **52**(3), 1640–1648 (1995)
6. Knyazev, A.: Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method. SIAM J. Sci. Comput. **22**(2), 517–541 (2001)
7. Kohn, W.: Density functional and density matrix method scaling linearly with the number of atoms. Phys. Rev. Lett. **76**(17), 3168–3171 (1996)

8. Kresse, G., Furthmüller, J.: Efficiency of ab initio total energy calculations for metals and semiconductors using a plane-wave basis set. Comput. Mater. Sci. **6**, 15–50 (1996)
9. Lin, L., Lu, J., Ying, L., Weinan, E.: Pole-based approximation of the Fermi-Dirac function. Chin. Ann. Math. **30B**, 729 (2009)
10. Lin, L., Yang, C., Lu, J., Ying, L., Weinan, E.: A fast parallel algorithm for selected inversion of structured sparse matrices with application to 2D electronic structure calculations. SIAM J. Sci. Comput. **33**, 1329–1351 (2011)
11. Lin, L., Yang, C., Meza, J.C., Lu, J., Ying, L., Weinan, E.: Selinv – an algorithm for selected inversion of a sparse symmetric matrix. ACM Trans. Math. Softw. **37**, 40:1–40:19 (2011)
12. Mauri, F., Galli, G.: Electronic-structure calculation and molecular dynamics simulations with linear system-size scaling. Phys. Rev. B **50**(7), 4316–4326 (1994)
13. Ordejón, P., Drabold, D.A., Grumbach, M.P., Martin, R.M.: Unconstrained minimization approach for electronic computations that scales linearly with system size. Phys. Rev. B **48**(19), 14646–14649 (1993)
14. Payne, M.C., Teter, M.P., Allen, D.C., Arias, T.A., Joannopoulos, J.D.: Iterative minimization techniques for ab initio total energy calculation: molecular dynamics and conjugate gradients. Rev. Mod. Phys. **64**(4), 1045–1097 (1992)
15. Prodan, E., Kohn, W.: Nearsightedness of electronic matter. PNAS **102**(33), 11635–11638 (2005)
16. Wang, L.W., Zhao, Z., Meza, J.: Linear-scaling three-dimensional fragment method for large-scale electronic structure calculations. Phys. Rev. B **29**, 165113–165117 (2008)
17. Yang, C., Meza, J.C., Wang, L.W.: A constrained optimization algorithm for total energy minimization in electronic structure calculation. J. Comput. Phys. **217**, 709–721 (2006)
18. Yang, C., Meza, J.C., Wang, L.W.: A trust region direct constrained minimization algorithm for the Kohn-Sham equation. SIAM J. Sci. Comput. **29**(5), 1854–1875 (2007)
19. Yang, W.: A local projection method for the linear combination of atomic orbital implementation of density-functional theory. J. Chem. Phys. **94**(2), 1208–1214 (1991)
20. Zhou, Y., Saad, Y., Tiago, M.L., Chelikowsky, J.R.: Self-consistent field calculations using Chebyshev-filtered subspace iteration. J. Comput. Phys. **219**, 172–184 (2006)

Lattice Boltzmann Methods

Paul Dellar[1] and Li-Shi Luo[2,3]
[1]OCIAM, Mathematical Institute, Oxford, UK
[2]Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA, USA
[3]Beijing Computational Science Research Center, Beijing, China

Mathematics Subject Classification

82B40; 76D05; 76M99; 35Q20; 35Q30; 35Q82

Synonyms

Lattice Boltzmann Method (LBM)

Short Definition

The lattice Boltzmann method (LBM) is a family of methods derived from kinetic equations for computational fluid dynamics, chiefly used for near-incompressible flows of Newtonian fluids.

Description

The primary focus of computational fluid dynamics (CFD) is the solution of the nonlinear Navier–Stokes–Fourier (NSF) equations that describe mass, momentum, and energy transport in a fluid. For the most common case of incompressible flow, these reduce to the Navier–Stokes (NS) equations for momentum transport alone, as supplemented by an elliptic equation to determine the pressure. The NSF equations may be derived from the Boltzmann equation of kinetic theory, with transport coefficients calculated from the underlying interatomic interactions. The lattice Boltzmann method (LBM) is distinguished by being a discretization of the Boltzmann equation, rather than a *direct* discretization of the NS equations.

Kinetic Theory and the Boltzmann Equation

Kinetic theory describes a dilute monatomic gas through a distribution function $f(\boldsymbol{x}, \boldsymbol{\xi}, t)$ for the number density of particles at position $\boldsymbol{x}$ moving with velocity $\boldsymbol{\xi}$ at time t. The distribution function evolves according to the Boltzmann equation [2, 6]

$$\partial_t f + \boldsymbol{\xi} \cdot \boldsymbol{\nabla} f = \mathcal{C}[f, f]. \tag{1}$$

The quadratic integral operator $\mathcal{C}[f, f]$ represents binary collisions between pairs of particles. The first few moments of f with respect to particle velocity $\boldsymbol{\xi}$ give hydrodynamic quantities: the fluid density ρ, velocity $\boldsymbol{u}$, momentum flux $\boldsymbol{\Pi}$, and energy flux $\mathbf{Q}$,

$$\rho = \int f \,\mathrm{d}\boldsymbol{\xi}, \quad \rho\boldsymbol{u} = \int \boldsymbol{\xi}\, f \,\mathrm{d}\boldsymbol{\xi},$$
$$\boldsymbol{\Pi} = \int \boldsymbol{\xi}\boldsymbol{\xi}\, f \,\mathrm{d}\boldsymbol{\xi}, \quad \mathbf{Q} = \int \boldsymbol{\xi}\boldsymbol{\xi}\boldsymbol{\xi}\, f \,\mathrm{d}\boldsymbol{\xi}, \qquad (2)$$

in convenient units with the particle mass scaled to unity. Collisions conserve mass, momentum, and energy, while relaxing f towards a Maxwell–Boltzmann distribution

$$f^{(0)} = \rho(2\pi\theta)^{-3/2} \exp\left(-\|\boldsymbol{u} - \boldsymbol{\xi}\|^2/(2\theta)\right). \qquad (3)$$

These together imply conservation of the temperature θ, given by $\mathrm{Tr}\,\boldsymbol{\Pi} = 3\rho\theta + \rho\|\boldsymbol{u}\|^2$ in energy units for which $\sqrt{\theta}$ is the Newtonian or isothermal sound speed.

Hydrodynamics describes near-equilibrium solutions, $f \approx f^{(0)}$, for which a linearized collision operator is sufficient. A popular model is the Bhatnagar–Gross–Krook (BGK) form [1]

$$\partial_t f + \boldsymbol{\xi}\cdot\nabla f = -\frac{1}{\tau}\left[f - f^{(0)}\right] \qquad (4)$$

that relaxes f towards an equilibrium distribution $f^{(0)}$ with the same $\rho, \boldsymbol{u}, \theta$ as f. This satisfies all the requirements necessary for deriving the NSF equations, but the Prandtl number is fixed at unity. The more general Gross–Jackson model [7] allows the specification of any finite number of relaxation times in place of the above single relaxation time τ.

Moments of the Boltzmann equation (1) give an infinite hierarchy of evolution equations for the moments of f. The first few are

$$\partial_t\rho + \nabla\cdot(\rho\boldsymbol{u}) = 0, \quad \partial_t(\rho\boldsymbol{u}) + \nabla\cdot\boldsymbol{\Pi} = 0,$$
$$\partial_t\boldsymbol{\Pi} + \nabla\cdot\mathbf{Q} = -\frac{1}{\tau}\left(\widetilde{\boldsymbol{\Pi}} - \widetilde{\boldsymbol{\Pi}}^{(0)}\right). \qquad (5)$$

Each evolution equation involves the divergence of the next higher moment. The first two right-hand sides vanish because collisions conserve microscopic mass and momentum. The right-hand side of the third equation arises from the traceless part $\widetilde{\boldsymbol{\Pi}}$ of the momentum flux being an eigenfunction of the BGK collision operator and an eigenfunction of the linearized Boltzmann collision operator for Maxwell molecules. The latter property holds to a good approximation for other interatomic potentials [2].

Temperature fluctuations are $\mathcal{O}(\mathrm{Ma}^2)$ when the Mach number $\mathrm{Ma} = \|\boldsymbol{u}\|/\sqrt{\theta}$ is small. It is then convenient to impose a constant temperature θ_0 when evaluating $f^{(0)}$. This takes the place of an independent energy evolution equation, and the last of (5) then holds with $\boldsymbol{\Pi}$ rather than the traceless part $\widetilde{\boldsymbol{\Pi}}$ on the right-hand side. A temperature evolution equation may be reintroduced under the Boussinesq approximation using a second distribution function [5, 10].

Derivation of the Hydrodynamic Equations

The NSF equations describe solutions of the Boltzmann equation that vary slowly on macroscopic timescales $\tau_0 \gg \tau$, where τ_0 may be a fluid eddy turnover time. The ratio $\epsilon = \tau/\tau_0$ may be identified with the Knudsen number Kn. The modern Chapman–Enskog expansion [2] seeks solutions of (1) or (4) through a multiple-scale expansion of both the distribution function and the time derivative:

$$f = \sum_{n=0}^{\infty} \epsilon^n f^{(n)}, \quad \partial_t = \sum_{n=0}^{\infty} \epsilon^n \partial_{t_n}. \qquad (6)$$

This expansion of f implies corresponding expansions of the moments:

$$\rho^{(n)} = \int f^{(n)}\mathrm{d}\boldsymbol{\xi}, \ \rho\boldsymbol{u}^{(n)} = \int \boldsymbol{\xi}\, f^{(n)}\mathrm{d}\boldsymbol{\xi},$$
$$\boldsymbol{\Pi}^{(n)} = \int \boldsymbol{\xi}\boldsymbol{\xi}\, f^{(n)}\mathrm{d}\boldsymbol{\xi}, \ \mathbf{Q}^{(n)} = \int \boldsymbol{\xi}\boldsymbol{\xi}\boldsymbol{\xi}\, f^{(n)}\mathrm{d}\boldsymbol{\xi}. \qquad (7)$$

The expansion of ∂_t prevents the overall expansion from becoming disordered after long times $t \sim \tau_0/\epsilon$, but requires additional solvability conditions, namely, that $\rho^{(n)} = 0$, $\boldsymbol{u}^{(n)} = 0$ for $n \geq 1$. Equivalently, one may expand the non-conserved moments $\boldsymbol{\Pi} = \boldsymbol{\Pi}^{(0)} + \epsilon\boldsymbol{\Pi}^{(1)} + \cdots$, $\mathbf{Q} = \mathbf{Q}^{(0)} + \epsilon\mathbf{Q}^{(1)} + \cdots$, while leaving the conserved moments ρ and $\boldsymbol{u}$ unexpanded.

Evaluating (5) at leading order gives the compressible Euler equations

$$\partial_{\tau_0}\rho + \nabla\cdot(\rho\boldsymbol{u}) = 0, \quad \partial_{\tau_0}(\rho\boldsymbol{u}) + \nabla\cdot\boldsymbol{\Pi}^{(0)} = 0. \qquad (8)$$

The inviscid momentum flux $\boldsymbol{\Pi}^{(0)} = \theta\rho\mathsf{I} + \rho\boldsymbol{u}\boldsymbol{u}$, with I the identity tensor, is given by the second moment of $f^{(0)}$. Evaluating the last of (5) at leading order gives

$$\partial_{\tau_0}\boldsymbol{\Pi}^{(0)} + \nabla\cdot\mathbf{Q}^{(0)} = -\frac{1}{\tau_0}\boldsymbol{\Pi}^{(1)}, \qquad (9)$$

where $\mathbf{Q}^{(0)}$ is known from $f^{(0)}$, and we evaluate $\partial_{\tau_0}\mathbf{\Pi}^{(0)}$ using the Euler equations (8). After some manipulation, $\epsilon\mathbf{\Pi}^{(1)} = -\tau\rho\theta\left[(\nabla\boldsymbol{u}) + (\nabla\boldsymbol{u})^{\mathsf{T}}\right] = -\tau\rho\theta\mathbf{S}$ becomes the NS viscous stress for an isothermal fluid with dynamic viscosity $\mu = \tau\rho\theta$. The multiple-scale expansion may be avoided by taking $\mathrm{Ma} = \mathcal{O}(\epsilon)$. This so-called diffusive scaling removes the separation of timescales by bringing the viscous term $\boldsymbol{u}\cdot\nabla\boldsymbol{u}$ into balance with $(\mu/\rho)\nabla^2\boldsymbol{u}$, pushing the $\partial_{\tau_0}\mathbf{\Pi}^{(0)}$ term in (9) to higher order [11].

Discrete Kinetic Theory

Discrete kinetic theory preserves the above structure that leads to the NS equations, but restricts the particle velocity to a finite set, $\boldsymbol{\xi} \in \{\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_{N-1}\}$. The previous integral moments become sums over a finite set $f_i(\boldsymbol{x}, t)$, one for each $\boldsymbol{\xi}_i$:

$$\rho = \textstyle\sum_i f_i,\ \ \rho\boldsymbol{u} = \textstyle\sum_i \boldsymbol{\xi}_i f_i,$$
$$\mathbf{\Pi} = \textstyle\sum_i \boldsymbol{\xi}_i\boldsymbol{\xi}_i f_i,\ \ \mathbf{Q} = \textstyle\sum_i \boldsymbol{\xi}_i\boldsymbol{\xi}_i\boldsymbol{\xi}_i f_i. \qquad (10)$$

The discrete analogue of the linearized Boltzmann equation is

$$\partial_t f_i + \boldsymbol{\xi}_i\cdot\nabla f_i = -\textstyle\sum_j \Omega_{ij}\left(f_j - f_j^{(0)}\right), \qquad (11)$$

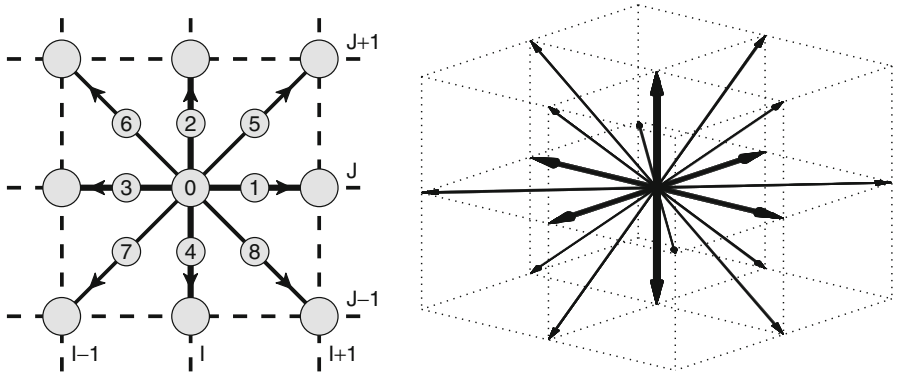
where Ω_{ij} is a constant $N \times N$ matrix giving a general linear collision operator. This linear, constant-coefficient hyperbolic system is readily discretized, as described below.

The aim now is to choose the velocity set $\{\boldsymbol{\xi}_i\}$, the equilibria $f_j^{(0)}(\rho, \boldsymbol{u})$, and the collision matrix Ω_{ij} so that the moment equations obtained from (11) coincide with the system (5) obtained previously from (1). The continuous Maxwell–Boltzmann equilibrium $f^{(0)}$ emerged from properties of Boltzmann's collision operator $C[f, f]$, but the $f_j^{(0)}(\rho, \boldsymbol{u})$ in (11) must be supplied explicitly. The discrete moments $\mathbf{\Pi}^{(0)}$ and $\mathbf{Q}^{(0)}$ should remain unchanged from continuous kinetic theory, at least to $\mathcal{O}(\mathrm{Ma}^2)$.

The discrete collision operator should conserve mass and momentum, and $\mathbf{\Pi}$ should be an eigenfunction. The simplest choice $\Omega_{ij} = \tau^{-1}\delta_{ij}$ gives the BGK collision operator (4), but more general choices improve numerical stability [3] and treatment of boundary conditions. The most common equilibria are the quadratic polynomials [9, 14]

$$f_j^{(0)}(\rho, \boldsymbol{u}) = w_j\rho\left(1 + 3\,\boldsymbol{u}\cdot\boldsymbol{\xi}_j + \frac{9}{2}(\boldsymbol{u}\cdot\boldsymbol{\xi}_j)^2 - \frac{3}{2}\|\boldsymbol{u}\|^2\right), \qquad (12)$$

with weights $w_0 = 4/9$, $w_{1,\ldots,4} = 1/9$, and $w_{6,\ldots,9} = 1/36$ for the D2Q9 lattice shown in Fig. 1. The particle velocities $\boldsymbol{\xi}_i$ are scaled so that $\xi_{ix},\ \xi_{iy} \in \{-1, 0, 1\}$ and $\theta = 1/3$. The $\boldsymbol{\xi}_i$ thus form an integer lattice. The above $f_j^{(0)}$ may be derived from a low Mach number expansion of the Maxwell–Boltzmann distribution, or as a moment expansion in the first few of Grad's [6] tensor Hermite polynomials $1, \boldsymbol{\xi}_i, \boldsymbol{\xi}_i\boldsymbol{\xi}_i - \theta\mathsf{I}$. The w_i and $\boldsymbol{\xi}_i$ are the weights and quadrature nodes for a Gauss–Hermite quadrature that holds exactly for polynomials of degree 5 or less. The ρ, $\boldsymbol{u}$, $\mathbf{\Pi}^{(0)}$ moments of the discrete and continuous equilibria thus coincide exactly [9], while the $\mathbf{Q}^{(0)}$ moment differs by an $\mathcal{O}(\mathrm{Ma}^3)$ term $\rho\boldsymbol{uuu}$.



Lattice Boltzmann Methods, Fig. 1 D2Q9 and D3Q19 lattices. The velocities $\boldsymbol{\xi}_i$ are scaled so that $\xi_{i\alpha} \in \{-1, 0, 1\}$ for $\alpha \in \{x, y, z\}$

Space–Time Discretization

For each i, we may write the left-hand side of (11) as a total derivative $\mathrm{d}f_i/\mathrm{d}s$ along the characteristic $(\boldsymbol{x},t) = (\boldsymbol{x}_0 + \boldsymbol{\xi}_i s, t + s)$ parametrized by s. Integrating (11) along this characteristic for a timestep Δt gives [10]

$$f_i(\boldsymbol{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) - f_i(\boldsymbol{x},t) = -\in \tau_0^{\Delta t} \sum_j \Omega_{ij} \left[f_j - f_j^{(0)} \right](\boldsymbol{x} + \boldsymbol{\xi}_i s, t + s)\, \mathrm{d}s. \tag{13}$$

Approximating the remaining integral by the trapezoidal rule gives

$$f_i(\boldsymbol{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) - f_i(\boldsymbol{x},t) = -\frac{1}{2}\Delta t \sum_j \Omega_{ij} \Big\{ f_j(\boldsymbol{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) + f_j(\boldsymbol{x},t) - f_j^{(0)}(\boldsymbol{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) - f_j^{(0)}(\boldsymbol{x},t) \Big\} + \mathcal{O}\left((\Delta t/\tau)^3 \right). \tag{14}$$

Neglecting the error term, and collecting all terms evaluated at $t + \Delta t$ to define

$$\overline{f}_i(\boldsymbol{x},t) = f_i(\boldsymbol{x},t) + \frac{1}{2}\Delta t \sum_j \Omega_{ij} \left(f_j - f_j^{(0)} \right), \tag{15}$$

leads to an explicit scheme, the lattice Boltzmann equation (LBE), for the $\overline{f}_i$:

$$\overline{f}_i(\boldsymbol{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) = \overline{f}_i(\boldsymbol{x},t) - \Delta t \sum_j \overline{\Omega}_{ij} \left(\overline{f}_j(\boldsymbol{x},t) - f_j^{(0)}(\boldsymbol{x},t) \right), \tag{16}$$

with discrete collision matrix $\overline{\boldsymbol{\Omega}} = \left(\mathsf{I} + \frac{1}{2}\Delta t \boldsymbol{\Omega} \right)^{-1} \boldsymbol{\Omega}$. When $\boldsymbol{\Omega} = \tau^{-1}\mathsf{I}$ this transformation reduces to replacing τ with $\tau + \Delta t/2$. Taking moments of (15) gives the conserved moments $\rho = \sum_i \overline{f}_i$ and $\rho \boldsymbol{u} = \sum_i \boldsymbol{\xi}_i \overline{f}_i$, unaffected by the collision term that distinguishes $\overline{f}_i$ from f_i. We may thus evaluate the $f_i^{(0)}$ in (16). However, non-conserved moments such as $\boldsymbol{\Pi}$ must be found by inverting (15) for the f_i.

The errors involving Δt from the space–time discretization of (11) are in principle entirely independent of the $\mathcal{O}(\tau^2)$ error in the derivation of the NS equations. However, the above usage of the trapezoidal rule requires $\Delta t \ll \tau$ to justify neglecting the error in (14). The same restriction is needed in the reverse derivation of partial differential equations from (16) using Taylor expansions in Δt [11]. However, the algorithm (16) successfully captures *slowly varying* hydrodynamic behavior on macroscopic timescales $\tau_0 \gg \Delta t$ even when $\Delta t \gg \tau$. The ratio $\Delta t/\tau$ may be identified with the grid-scale Reynolds number $\mathrm{Re}_{\mathrm{grid}} = \|\boldsymbol{u}\| \Delta x/\nu$, with $\Delta x = \Delta t$ in standard LB units. Stability for $\mathrm{Re}_{\mathrm{grid}} \gg 1$ is essential for applying the LBM to turbulent flows. Stable 2D simulations have been demonstrated [3] with $\mathrm{Re}_{\mathrm{grid}} \gtrsim 100$ and a collision matrix Ω_{ij} that suppresses the oscillations with period $2\Delta t$ that arise in the non-conserved moments when $\mathrm{Re}_{\mathrm{grid}} > 1$.

These successes do not imply that the LBE correctly captures *arbitrary* solutions of the discrete Boltzmann equation evolving on the collisional timescale τ, such as kinetic initial and boundary (Knudsen) layers [2, 6]. The LBE reproduces just enough of the true Boltzmann equation to capture the isothermal NS equations. It does not capture Burnett and higher order corrections relevant for rarefied flows at finite Knudsen numbers, and it does not capture Knudsen boundary layers.

Wider Applications

The core lattice Boltzmann algorithm described above has been extended into many wider applications: large eddy simulations of turbulent flows, multiphase flows, and soft condensed matter systems such as colloids, suspensions, gels, and polymer solutions [4, 12]. The LBM is commonly characterized as a second-order accurate scheme at fixed Mach number. However, the spatial derivatives on the left-hand side of (11) are treated exactly in deriving (13). The only approximation lies in the treatment of the collision integral. Comparisons with pseudo-spectral simulations for the statistics of turbulent flows show comparable accuracy when the LBM grid is roughly twice as fine as the pseudo-spectral collocation grid [13].

The nonequilibrium momentum flux $\boldsymbol{\Pi}^{(1)}$ is proportional to the local strain rate $\mathbf{S} = (\nabla \boldsymbol{u}) + (\nabla \boldsymbol{u})^{\mathsf{T}}$ under the Chapman–Enskog expansion, so $\mathbf{S}$ may be computed locally from $(\boldsymbol{\Pi} - \boldsymbol{\Pi}^{(0)})$ at each grid point with no spatial differentiation [15]. Adjusting the local collision time τ to depend on $\mathbf{S}$ extends the LBM to large eddy simulations using the Smagorinsky turbulence model, with an effective eddy viscosity $\mu_{\mathrm{turb}} \propto ||\mathbf{S}||$,

and to further generalized Newtonian fluids whose viscosities are functions of $||\mathbf{S}||$.

The straightforward implementation of boundary conditions by reflecting particles from solid boundaries makes the LBM attractive for simulating pore-scale flows in porous media and particle-scale flows of suspensions. The Brownian thermal fluctuations omitted in the Boltzmann equation, but relevant for colloids, may be restored by adding random noise to the non-conserved moments during collisions [4].

There are many LB formulations for multiphase and multicomponent flows [8]. They are essentially diffuse interface capturing schemes that use interactions between neighboring grid points to mimic the inter-particle interactions responsible for interfacial phenomena.

References

1. Bhatnagar, P.L., Gross, E.P., Krook, M.: A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component system. Phys. Rev. **94**, 511–525 (1954)
2. Cercignani, C.: The Boltzmann Equation and its Applications. Springer, New York (1988)
3. Dellar, P.J.: Incompressible limits of lattice Boltzmann equations using multiple relaxation times. J. Comput. Phys. **190**, 351–370 (2003)
4. Dünweg, B., Ladd, A.J.C.: Lattice Boltzmann simulations of soft matter systems. Adv. Polym. Sci. **221**, 1–78 (2009)
5. Eggels, J.G.M., Somers, J.A.: Numerical-simulation of free convective flow using the lattice-Boltzmann scheme. Int. J. Heat Fluid Flow **16**, 357–364 (1995)
6. Grad, H.: Principles of the kinetic theory of gases. In: Flügge, S. (ed.) Thermodynamik der Gase. Handbuch der Physik, vol. 12, pp. 205–294. Springer, Berlin (1958)
7. Gross, E.P., Jackson, E.A.: Kinetic models and the linearized Boltzmann equation. Phys. Fluids **2**, 432–441 (1959)
8. Gunstensen, A.K., Rothman, D.H., Zaleski, S., Zanetti, G.: Lattice Boltzmann model of immiscible fluids. Phys. Rev. A **43**, 4320–4327 (1991)
9. He, X., Luo, L.S.: Theory of the lattice Boltzmann method: from the Boltzmann equation to the lattice Boltzmann equation. Phys. Rev. E **56**, 6811–6817 (1997)
10. He, X., Chen, S., Doolen, G.D.: A novel thermal model of the lattice Boltzmann method in incompressible limit. J. Comput. Phys. **146**, 282–300 (1998)
11. Junk, M., Klar, A., Luo, L.S.: Asymptotic analysis of the lattice Boltzmann equation. J. Comput. Phys. **210**, 676–704 (2005)
12. Ladd, A.J.C.: Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. J. Fluid Mech. **271**, 285–309 (1994)
13. Peng, Y., Liao, W., Luo, L.S., Wang, L.P.: Comparison of the lattice Boltzmann and pseudo-spectral methods for decaying turbulence: low-order statistics. Comput. Fluids **39**, 568–591 (2010)
14. Qian, Y.H., d'Humières, D., Lallemand, P.: Lattice BGK models for the Navier–Stokes equation. Europhys. Lett. **17**, 479–484 (1992)
15. Somers, J.A.: Direct simulation of fluid flow with cellular automata and the lattice-Boltzmann equation. Appl. Sci. Res. **51**, 127–133 (1993)

Least Squares Calculations

Åke Björck
Department of Mathematics, Linköping University, Linköping, Sweden

Introduction

A computational problem of primary importance in science and engineering is to fit a mathematical model to given observations. The influence of errors in the observations can be reduced by using a greater number of measurements than the number of unknown parameters. Least squares estimation was first used by Gauss in astronomical calculations more than two centuries ago. It has since been a standard approach in applications areas that include geodetic surveys, photogrammetry, signal processing, system identification, and control theory. Recent technological developments have made it possible to generate and treat problems involving very large data sets.

As an example, consider a model described by a scalar function $f(x,t)$, where $x \in \mathbf{R}^n$ is an unknown parameter vector to be determined from measurements $b_i = f(x,t_i) + e_i, i = 1,\ldots,m (m > n)$, where e_i are errors. In the simplest case $f(x,t_i)$ is linear in x:

$$f(x,t) = \sum_{j=1}^{n} x_j \phi_j(t), \tag{1}$$

where $\phi_j(t)$ are known basis functions. Then the measurements form an overdetermined system of linear equations $Ax = b$, where $A \in \mathbf{R}^{m\times n}$ is a matrix with elements $a_{ij} = \phi_j(t_i)$.

It is important that the basis function $\phi_j(t)$ are chosen carefully. Suppose that $f(x,t)$ is to be modeled by a polynomial of degree n. If the basis functions

are chosen as the monomials t^j, then A will be a Vandermonde matrix. Such matrices are notoriously ill conditioned and this can lead to an inaccurate solution.

The Least Squares Principle

In the standard Gauss–Markov linear model, it is assumed that a linear relation $Ax = y$ holds, where $A \in \mathbf{R}^{m\times n}$ is a known matrix of full column rank, x is a parameter vector to be determined, and $y \in \mathbf{R}^m$ a constant but unknown vector. The vector $b = f + e$ is a vector of observations and e a random error vector. It is assumed that e has zero mean and covariance matrix $\sigma^2 I$, where σ^2 is an unknown constant.

Theorem 1 (***The Gauss–Markov Theorem***) *In the linear Gauss–Markov model, the best linear unbiased estimator of x is the least square estimate $\hat{x}$ that minimizes the sum of squares*

$$S(x) = \|r(x)\|_2^2 = \sum_{i=1}^{m} r_i^2,$$

where $r(x) = b - Ax$ is the residual vector. A necessary condition for a minimum is that the gradient vector $\partial S/\partial x$ is zero. This condition gives $A^T(b - Ax) = 0$, i.e., $r(x) \perp \mathcal{R}(A)$, the range of A. It follows that $\hat{x}$ satisfies the normal equations $A^T Ax = A^T b$. The best linear unbiased estimator of any linear functional $c^T x$ is $c^T \hat{x}$.

The covariance matrix of the estimate $\hat{x}$ is $\mathcal{V}(\hat{x}) = \sigma^2(A^T A)^{-1}$. The residual vector $\hat{r} = b - A\hat{x}$ is uncorrelated with $\hat{x}$ and an unbiased estimate of σ^2 is given by $s^2 = \|\hat{r}\|_2^2/(m-n)$.

In the complex case $A \in \mathbf{C}^{m\times n}$, $b \in \mathbf{C}^m$, the complex scalar product has to be used in Gauss–Markov theorem. The least squares estimate minimizes $\|r\|_2^2 = r^H r$, where r^H denotes the complex conjugate transpose of r. The normal equations are $A^H Ax = A^H b$. This has applications, e.g., in complex stochastic processes.

It is easy to generalize the Gauss–Markov theorem to the case where the error e has a symmetric positive definite covariance matrix $\sigma^2 V$. The least squares estimate then satisfies the generalized normal equations

$$A^T V^{-1} Ax = A^T V^{-1} b. \tag{2}$$

The covariance matrix of the least squares estimate $\hat{x}$ is $\mathcal{V}(\hat{x}) = \sigma^2(A^T V^{-1} A)^{-1}$ and an unbiased estimate of σ^2 is given by $s^2 = \hat{r}^T V^{-1}\hat{r}/(m-n)$. In the special case of weighted least squares, the covariance matrix is $V = D^{-2}$, $D = \mathrm{diag}(d_1, \ldots, d_m)$. After a diagonal scaling this is equivalent to the scaled standard problem $\min_x \|Db - (DA)x)\|_2$.

Calculating Least Squares Estimates

Comprehensive discussions of methods for solving least squares problems are found in [6] and [1]. In the following we write the algebraic linear least squares problems in the form $\min_x \|Ax - b\|_2$.

The singular value decomposition (SVD) is a powerful tool both for analyzing and solving the linear least squares problem. The SVD of $A \in \mathbf{R}^{m\times n}$ of $\mathrm{rank}(A) = n$ is

$$A = U\Sigma V^T = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 \\ 0 \end{pmatrix} V^T = U_1 \Sigma_1 V^T, \tag{3}$$

where $\Sigma_1 = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$. Here $\sigma_1 \geq \sigma_2 \geq \cdots, \geq \sigma_n \geq 0$ are the singular values of A and the matrices $U = (u_1, u_2, \ldots, u_m)$ and $V = (v_1, v_2, \ldots, v_n)$ are square orthogonal matrices, whose columns are the left and right singular vectors of A. If $\sigma_n > 0$ the least squares solution equals

$$x = V\Sigma_1^{-1}(U_1^T b) = \sum_{i=1}^{n} \frac{c_i}{\sigma_i} v_i, \quad c_i = u_i^T b \tag{4}$$

If A has small singular values, then small perturbations in b can give rise to large perturbations in x. The ratio $\kappa(A) = \sigma_1/\sigma_n$ is the condition number of A. The condition number of the least squares solution x can be shown to depend also on the ratio $\|r\|_2/\sigma_n\|x\|_2$ and equals [1] $\kappa(x) = \kappa(A)\left(1 + \dfrac{\|r\|_2}{\sigma_n\|x\|_2}\right)$. The second term will dominate if $\|r\|_2 > \sigma_n\|x\|_2$.

Because of the high cost of computing and modifying the SVD, using the expansion (4) is not always justified. Simpler and cheaper alternative methods are available.

The Method of Normal Equations

If $A \in \mathbf{R}^{m\times n}$ has full column rank, the solution can be obtained from the normal equations. The symmetric

matrix $A^TA \in \mathbf{R}^{n\times n}$ is first formed. Then the Cholesky factorization $A^TA = R^TR$ is computed, where R is an upper triangular matrix with positive diagonal elements. These operations require $mn^2+n^3/3$ floating point operations (flops). For a right-hand side b, the least squares solution is obtained by computing $d = A^Tb \in \mathbf{R}^n$ and solving two triangular systems $R^Tz = d$ and $Rx = z$. The residual matrix is $r = b - Ax$. This requires $2n(2m + n)$ flops.

The estimated covariance matrix of x is

$$V_x = s^2(R^TR)^{-1} = s^2SS^T, \quad s^2 = r^Tr/(m-n), \quad S = R^{-1}. \tag{5}$$

The estimated variance of any linear functional $\phi = f^Tx$ is

$$V_\phi = s^2f^TSS^Tf = s^2v^Tv, \quad R^Tv = f. \tag{6}$$

and can be computed without forming V_x. Setting $f = e_i$ gives the variance of the component x_i. The components of the normalized residual $\tilde{r} = \frac{1}{s}\,\mathrm{diag}(V_x)^{-1}\hat{r}$ should be uniformly distributed random variables. This can be used to detect and identify bad observations.

QR Factorizations and Bidiagonal Decomposition

The method of normal equations is efficient and sufficiently accurate for many problems. However, forming the normal equations squares the condition number of the problem. This can be seen by using the SVD to show that $A^TA = V\Sigma U^TU\Sigma V^T = V\Sigma_1^2V^T$ and hence $\kappa(A^TA) = \kappa^2(A)$. Methods using orthogonal transformations preserve the condition number and should be preferred unless the problem is known to be well conditioned. The QR factorization of the matrix $A \in \mathbf{R}^{m\times n}$ of full column rank is

$$A = Q\begin{pmatrix} R \\ 0 \end{pmatrix} = Q_1R, \tag{7}$$

where $Q = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \in \mathbf{R}^{m\times m}$ is orthogonal and $R \in \mathbf{R}^{n\times n}$ upper triangular. It can be computed in $2(mn^2 - n^3/3)$ flops using Householder transformations. The matrix Q is then implicitly represented as $Q = P_1P_2\cdots P_n$ where $P_i = I - 2v_iv_i^T$, $\|v_i\|_2 = 1$. Only the Householder vectors v_i need to be stored and saved. The least squares solution and the residual vector are then obtained in about $8mn-3n^2$ flops from

$$Q^Tb = P_n\cdots P_2P_1b = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, \quad Rx = c_1,$$
$$r = P_1P_2\cdots P_n\begin{pmatrix} 0 \\ c_2 \end{pmatrix}. \tag{8}$$

Using orthogonality it follows that $\|r\|_2 = \|c_2\|_2$. If the diagonal elements in the triangular factor R are chosen to be positive, then R is uniquely determined and mathematically (not numerically) the same as the Cholesky factor from the normal equations. Thus, the expression (5) for the estimated covariance matrix is valid.

It is recommended that column pivoting is performed in the QR factorization. This will yield a QR factorization of $A\Pi$ for some permutation matrix Π. The standard strategy is to choose at each step $k = 1, \ldots, n$, the column that maximizes the diagonal element r_{kk} in R. Then the sequence $r_{11} \geq r_{22} \geq \cdots \geq r_{nn} > 0$ is nonincreasing, and the ratio r_{11}/r_{nn} is often used as a rough approximation of $\kappa(A)$.

A rectangular matrix $A \in \mathbf{R}^{m\times n}$, $m > n$ can be transformed further to lower (or upper) bidiagonal form by a sequence of two-sided orthogonal transformations

$$U^TAV = \begin{pmatrix} B \\ 0 \end{pmatrix}, \quad B = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_n \\ & & & \beta_{n+1} \end{pmatrix} \tag{9}$$

where $U = (u_1, u_2, \ldots, u_m)$, $V = (v_1, v_2, \ldots, v_n)$. This orthogonal decomposition requires $4(mn^2-n^3/3)$ flops, which is twice as much as the QR factorization. It is essentially unique once the first column $u_1 = Ue_1$ has been chosen. It is convenient to take $u_1 = b/\beta_1$, $\beta_1 = \|b\|_2$. Then $U^Tb = \beta_1e_1$ and setting $x = Vy$, we have

$$U^T(b - Ax) = \begin{pmatrix} \beta_1e_1 - By \\ 0 \end{pmatrix}.$$

The least squares solution can be computed in $O(n)$ flops by solving the bidiagonal least squares problem $\min_y \|By - \beta_1e_1\|_2$. The upper bidiagonal form makes the algorithm closely related to the iterative LSQR algorithm in [7]. Also, with this choice of u_1

the decomposition will terminate early with a core subproblem if an entry α_i or β_i is zero ([8]].

Rank-Deficient Problems

Rank deficiency in least squares problems can arise in different ways. In statistics one often has a large set of variables, called the factors, that are used to control, explain, or predict other variables. The set of factors correspond to the columns of a matrix $A = (a_1, a_2, \ldots, a_n)$. If these are highly collinear, then the approximate rank of A is less than n and the least squares solution is not unique. Often the rank of A is not known in advance, but needs to be determined as part of the solution process.

In the rank-deficient case one can seek the least squares solution of minimum norm, i.e., solve the problem

$$\min_{x \in S} \|x\|_2, S = \{x \in \mathbf{R}^n \mid \|b - Ax\|_2 = \min\}. \quad (10)$$

This problem covers as special cases both overdetermined and underdetermined linear systems. The solution is always unique and called the pseudoinverse solution. It is characterized by $x \in \mathcal{R}(A^T)$ and can be obtained from the SVD of A as follows. If $\text{rank}(A) = r < n$, then $\sigma_j = 0$, $j > r$, and

$$x = A^{\dagger} b = V_1 \Sigma_1^{-1} (U_1^T b) = \sum_{i=1}^{r} \frac{c_i}{\sigma_i} v_i, \quad c_i = u_i^T b, \quad (11)$$

i.e., it is obtained simply by excluding terms corresponding to zero singular values in the expansion (4). The matrix $A^{\dagger} = V_1 \Sigma_1^{-1} U_1^T$ is called the pseudoinverse of A.

In some applications, e.g., in signal processing, one has to solve a sequence of problems where the rank may change. For such problems methods that use a pivoted QR factorization have the advantage over the SVD in that these factorizations can be efficiently updated; see [4]. One useful variant is the URV decomposition, which has the form

$$A\Pi = URV^T = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}. \quad (12)$$

Here R_{11} is upper triangular and the entries of R_{12} and R_{22} have small magnitudes. The orthogonal matrices U_1 and V_2 approximate the range and null space of A, respectively.

Large-Scale Problems

Many applications lead to least squares problems where A is large and sparse or structured. In the QR factorization of a sparse matrix, the factor Q will often be almost full. This is related to the fact that $Q = AR^{-1}$ and even if R is sparse R^{-1} will have no zero elements. Therefore, computing the factor Q explicitly for a sparse matrix should be avoided. A QR algorithm for banded matrices which processes rows or block of rows sequentially is given in [6, Chap. 27]. An excellent source book on factorization of matrices with more irregular sparsity is [3].

An efficient iterative method for solving large sparse least squares problems is the Krylov subspace method LSQR (see [7]). It uses a Lanczos process to generate the vectors v_i, u_{i+1}, $i = 1, 2, \ldots$ and the columns of the matrix B in (9). LSQR only requires one matrix-vector product with A and A^T per iteration step. If A is rank deficient, LSQR converges to the pseudoinverse solution.

Regularization of Least Squares Problems

In discrete approximations to inverse problems, the singular values σ_i of A cluster at zero. If the *exact* right-hand side b is contaminated by white noise, this will affect *all coefficients* c_i in the SVD expansion (4) more or less equally. Any attempt to solve such a problem without restriction on x will lead to a meaningless solution.

Truncated SVD and Partial Least Squares

If the SVD of A is available, then regularization can be achieved simply by including in the SVD expansion only terms for which $\sigma_1 > \text{tol}$, for some tolerance tol only. An often more efficient alternative is to use partial least squares (PLS). Like truncated SVD it computes a sequence of approximate least squares solutions by orthogonal projections onto lower dimensional subspaces. PLS can be implemented through a partial reduction of A to lower bidiagonal form. It is used extensively in chemometrics, where it was introduced in [10]. The connection to the bidiagonal decomposition is exhibited in [2].

Tikhonov Regularization

Tikhonov regularization is another much used method. In this a penalty is imposed on the 2-norm of $\|x\|_2$ of the solution. Given $A \in \mathbf{R}^{m\times n}$ a regularized least squares problem $\min_x \left[\|Ax-b\|_2^2+\mu^2\|x\|_2^2\right]$ is solved, where the parameter μ governs the balance between a small residual and a smooth solution. In statistics Tikhonov regularization is known as "ridge regression." The solution $x(\mu) = (A^TA + \mu^2 I)^{-1}A^Tb$ can be computed by Cholesky factorization. In terms of the SVD expansion, it is $x(\mu) = \sum_{i=1}^{n} \frac{c_i\sigma_i}{\sigma_i^2+\mu^2}v_i$. Methods using QR factorization, which avoid forming the cross-product matrix A^TA, can also be used [1]. The optimal value of μ depends on the noise level in the data. The choice of μ is often a major difficulty in the solution process and often an ad hoc method is used; see [5].

In the LASSO (Least Absolute Shrinkage and Selection) method a constraint involving the one norm $\|x\|_1$ is used instead. The resulting problem can be solved using convex optimization methods. LASSO tends to give solutions with fewer nonzero coefficients than Tikhonov regularization; see [9]. This property is fundamental for its use in compressed sensing.

References

1. Björck, Å.: Numerical Methods For Least Squares Problems, pp. xvii+408. SIAM, Philadelphia (1996). ISBN 0-89871-360-9
2. Bro, R., Eldén, L.: PLS works. Chemometrics **23**, 69–71 (2009)
3. Davis, T.A.: Direct Methods for Sparse Linear Systems, Fundamental of Algorithms, vol. 2. SIAM, Philadelphia (2006)
4. Fierro, R.D., Hansen, P.C., Hansen, P.S.K.: UTV tools: Matlab templates for rank-revealing UTV decompositions. Numer. Algorithms **20**, 165–194 (1999)
5. Hansen, P.C.: Rank-Deficient and Discrete Ill-Posed Problems. In: Numerical Aspects of Linear Inversion, pp. x+224. SIAM, Philadelphia (1998). ISBN 978-0-898716-26-9
6. Lawson, C.L., Hanson, R.J.: Solving Least Squares Problems, pp. xii+337. Prentice-Hall, Englewood Cliffs (1974). Revised republication by SIAM, Philadelphia (1995). ISBN 0-89871-356-0
7. Paige, C.C., Saunders, M.A.: LSQR. An algorithm for sparse linear equations and sparse least squares. ACM Trans. Math. Softw. **8**, 43–71 (1982)
8. Paige, C.C., Strakoš, Z.: Core problems in linear algebraic systems. SIAM J. Matrix Anal. Appl. **27**(2), 861–875 (2006)
9. Tibshirani, R.: Regression shrinkage and selection via the LASSO. R. Stat. Soc. B. **58**(1), 267–288 (1996)
10. Wold, S., Ruhe, A., Wold, H., Dunn, W.J.: The collinearity problem in linear regression, the partial least squares (pls) approach to generalized inverses. SIAM J. Sci. Stat. Comput. **5**, 735–743 (1984)

Least Squares Finite Element Methods

Pavel Bochev[1] and Max Gunzburger[2]
[1]Computational Mathematics, Sandia National Laboratories, Albuquerque, NM, USA
[2]Department of Scientific Computing, Florida State University, Tallahassee, FL, USA

The root cause for the remarkable success of early finite element methods (FEMs) is their intrinsic connection with Rayleigh-Ritz principles. Yet, many partial differential equations (PDEs) are not associated with unconstrained minimization principles and give rise to less favorable settings for FEMs. Accordingly, there have been many efforts to develop FEMs for such PDEs that share some, if not all, of the attractive mathematical and algorithmic properties of the Rayleigh-Ritz setting. Least-squares principles achieve this by abandoning the naturally occurring variational principle in favor of an artificial, external energy-type principle. Residual minimization in suitable Hilbert spaces defines this principle. The resulting least-squares finite element methods (LSFEMs) consistently recover almost all of the advantages of the Rayleigh-Ritz setting over a wide range of problems, and with some additional effort, they can often create a completely analogous variational environment for FEMs.

A more detailed presentation of least-squares finite element methods is given in [1].

Abstract LSFEM theory Consider the abstract PDE problem

$$\textit{find } u \in X \quad \textit{such that} \quad \mathcal{L}u = f \quad \textit{in } Y, \qquad (1)$$

where X and Y are Hilbert spaces, $\mathcal{L} : X \mapsto Y$ is a bounded linear operator, and $f \in Y$ is given data.

Sandia National Laboratories is a multiprogram laboratory operated by the Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the US Department of Energy's National Nuclear Sec- urity Administration under contract DE-AC04-94AL85000.

Assume (1) to be well posed so that there exist positive constants α and β such that

$$\beta\|u\|_X \le \|\mathcal{L}u\|_Y \le \alpha\|u\|_X \qquad \forall\, u \in X. \tag{2}$$

The *energy balance* (2) is the starting point in the development of LSFEMs. It gives rise to the unconstrained minimization problem, i.e., the *least-squares principle* (LSP):

$$\{J, X\} \;\rightarrow\; \left\{ \min_{u\in X} J(u; f),\;\; J(u; f) = \|\mathcal{L}u - f\|_Y^2 \right\}, \tag{3}$$

where $J(u, f)$ is the *residual energy* functional. From (2), it follows that $J(\cdot;\cdot)$ is *norm equivalent*:

$$\beta^2\|u\|_X^2 \le J(u; 0) \le \alpha^2\|u\|_X^2 \qquad \forall\, u \in X. \tag{4}$$

Norm equivalence (4) and the Lax-Milgram Lemma imply that the Euler-Lagrange equation of (3)

$$\begin{aligned} &\text{find } u \in X \quad \text{such that} \quad \big(\mathcal{L}v, \mathcal{L}u\big)_Y \equiv Q(u, w) \\ &\qquad = F(w) \equiv \big(\mathcal{L}v, f\big)_Y \quad \forall\; w \in X \end{aligned} \tag{5}$$

is well posed because $Q(u, w)$ is an equivalent inner product on $X \times X$. The unique solution of (5), resp. (3), coincides with the solution of (1).

We define an LSFEM by restricting (3) to a family of finite element subspaces $X^h \subset X$, $h \to 0$. The LSFEM approximation $u^h \in X^h$ to the solution $u \in X$ of (1) or (3) is the solution of the unconstrained minimization problem

$$\{J, X^h\} \rightarrow \left\{ \min_{u^h\in X^h} J(u^h; f),\;\; J(u; f)=\|\mathcal{L}u^h-f\|_Y^2 \right\}. \tag{6}$$

To compute u^h, we solve the Euler-Lagrange equation corresponding to (6):

$$\begin{aligned} &\textit{find } u^h \in X^h \textit{such that } Q(u^h, w^h) \\ &\qquad = F(w^h) \forall\, w^h \in X^h. \end{aligned} \tag{7}$$

Let $\{\phi_j^h\}_{j=1}^N$ denote a basis for X^h so that $u^h = \sum_{j=1}^N u_j^h \phi_j^h$. Then, problem (7) is equivalent to the linear system of algebraic equations

$$\mathbb{Q}^h \vec{u}^h = \vec{f}^h \tag{8}$$

for the unknown vector $\vec{u}^h$, where $\mathbb{Q}_{ij}^h = \big(\mathcal{L}\phi_j^h, \mathcal{L}\phi_i^h\big)_Y$ and $\vec{f}_i^h = \big(\mathcal{L}\phi_i, f\big)_Y$.

Theorem 1 *Assume that* (2)*, or equivalently,* (4)*, holds and that* $X^h \subset X$*. Then:*

- *The bilinear form* $Q(\cdot,\cdot)$ *is continuous, symmetric, and strongly coercive.*
- *The linear functional* $F(\cdot)$ *is continuous.*
- *The problem* (5) *has a unique solution* $u \in X$ *that is also the unique solution of* (3)*.*
- *The problem* (7) *has a unique solution* $u^h \in X^h$ *that is also the unique solution of* (6)*.*
- *The LSFEM approximation* u^h *is optimally accurate with respect to solution norm* $\|\cdot\|_X$*. for which* (1) *is well posed, i.e., for some constant* $C > 0$

$$\|u - u^h\|_X \le C \inf_{v^h\in X^h} \|u - v^h\|_X \tag{9}$$

- *The matrix* $\mathbb{Q}^h$ *of* (8) *is symmetric and positive definite.* □

Theorem 1 only assumes that (1) *is well posed and that* X^h *is conforming*. It does not require $\mathcal{L}$ to be positive self-adjoint as it would have to be in the Rayleigh-Ritz setting, nor does it impose any compatibility conditions on X^h that are typical of other FEMs. Despite the generality allowed for in (1), *the LSFEM based on* (6) *recovers* all *the desirable features possessed by finite element methods in the Rayleigh-Ritz setting*. This is what makes LSFEMs intriguing and attractive.

Practical LSFEM Intuitively, a "practical" LSFEM has coding complexity and conditioning comparable to that of other FEMs for the same PDE. The LSP $\{J, X\}$ in (3) recreates a true Rayleigh-Ritz setting for (1), yet the LSFEM $\{J, X^h\}$ in (6) may be impractical. Thus, sometimes it is necessary to replace $\{J, X\}$ by a practical discrete alternative $\{J^h, X^h\}$. Two opposing forces affect the construction of $\{J^h, X^h\}$: a desire to keep the resulting LSFEM simple, efficient, and practical and a desire to recreate the true Rayleigh-Ritz setting. The latter requires J^h to be as close as possible to the "ideal" norm-equivalent setting in (3).

The transformation of $J(\cdot,\cdot)$ into a discrete functional $J^h(\cdot,\cdot)$ illustrates the interplay between these issues. To this end, it is illuminating to write the energy balance (2) in the form

$$C_1\|\mathcal{S}_X u\|_0 \le \|\mathcal{S}_Y \circ \mathcal{L}u\|_0 \le C_2\|\mathcal{S}_X u\|_0, \tag{10}$$

where $\mathcal{S}_X, \mathcal{S}_Y$ are norm-generating operators for X, Y, respectively, with $L^2(\Omega)$ acting as a pivot space. At the least, practicality requires that the basis of X^h can be constructed with no more difficulty than for Galerkin FEM for the same PDE. To secure this property, we ask that the domain $D(\mathcal{S}_X)$ of $\mathcal{S}_X$ contains "practical" discrete subspaces. Transformation of (1) into an *equivalent first-order system* PDE achieves this. Then, practicality of the "ideal" LSFEM (6) depends solely on the effort required to compute $\mathcal{S}_Y \circ \mathcal{L}u^h$. If this effort is deemed reasonable, the original energy norm $|||u||| = \|\mathcal{S}_Y \circ \mathcal{L}u\|_0$ can be retained and the transition process is complete. Otherwise, we proceed to replace the composite operator $\mathcal{S}_Y \circ \mathcal{L}$ by a computable discrete approximation $\mathcal{S}_Y^h \circ \mathcal{L}^h$. We may need a *projection* operator π^h that maps the data f to the domain of $\mathcal{S}_Y^h$. The conversion process and the key properties of the resulting LSFEM can be encoded by the *transition* diagram

$$\begin{array}{ccccc} J(u;f) = \|\mathcal{S}_Y \circ (\mathcal{L}u - f)\|_0^2 & \to & |||u||| \\ \downarrow & & \downarrow & & \downarrow \\ J^h(u^h;f) = \|\mathcal{S}_Y^h \circ (\mathcal{L}^h u^h - \pi^h f)\|_0 & \to & |||u^h|||_h \end{array} \tag{11}$$

and the companion *norm-equivalence* diagram

$$\begin{array}{ccccc} C_1\|u\|_X & \le & |||u||| & \le & C_2\|u\|_X \\ \downarrow & & \downarrow & & \downarrow \\ C_1(h)\|u^h\|_X & \le & |||u^h|||_h & \le & C_2(h)\|u^h\|_X. \end{array} \tag{12}$$

Because $\mathcal{L}$ defines the problem being solved, the choice of $\mathcal{L}^h$ governs the accuracy of the LSFEM. The goal here is to make J^h as close as possible to J for the exact solution of (1). On the other hand, $\mathcal{S}_Y$ defines the energy balance of (1), i.e., the proper scaling between data and solution. As a result, the main objective in the choice of $\mathcal{S}_Y^h$ is to ensure that the scaling induced by J^h is as close as possible to (2), i.e., to "bind" the LSFEM to the energy balance of the PDE.

Taxonomy of LSFEMs Assuming that X^h is practical, restriction of $\{J, X\}$ to X^h transforms (3) into the *compliant* LSFEM $\{J, X^h\}$ in (6). Apart from this "ideal" LSFEM which reproduces the classical Rayleigh-Ritz principle, there are two other kinds of LSFEMs that gradually drift away from this setting, primarily by *simplifying the approximations* of the norm-generating operator $\mathcal{S}_Y$. Mesh-independent $C_1(h)$ and $C_2(h)$ in (12) characterize the *norm-equivalent* class, which retains virtually all attractive properties of the Rayleigh-Ritz setting, including identical convergence rates and matrix condition numbers. A mesh-dependent norm-equivalence (12) distinguishes the *quasi-norm-equivalent* class, which admits the broadest range of LSFEMs, but can give problems with higher condition numbers.

Examples We use the Poisson equation for which $\mathcal{L} = -\Delta$ to illustrate different classes of LSFEMs. One energy balance (2) for this equation corresponds to $X = H^2(\Omega) \cap H_0^1(\Omega)$ and $Y = L^2(\Omega)$:

$$\alpha\|u\|_2 \le \|\Delta u\|_0 \le \beta\|u\|_2.$$

The associated LSP

$$\{J, X\} \to \left\{ \min_{u \in X} J(u;f),\ J(u;f) = \|\Delta u - f\|_0^2 \right\}$$

leads to impractical LSFEMs because finite element subspaces of $H^2(\Omega)$ are not easy to construct.

Transformation of $-\Delta u = f$ into the equivalent first-order system

$$\nabla \cdot \mathbf{q} = f \quad \text{and} \quad \nabla u + \mathbf{q} = 0 \tag{13}$$

can solve this problem. The spaces $X = H_0^1(\Omega) \times [L^2(\Omega)]^d$, $Y = H^{-1}(\Omega) \times [L^2(\Omega)]^d$ have practical finite element subspaces and provide the energy balance

$$\begin{aligned} \alpha(\|u\|_1 + \|\mathbf{q}\|_0) &\le \|\nabla \cdot \mathbf{q}\|_{-1} + \|\nabla u + \mathbf{q}\|_0 \\ &\le \beta(\|u\|_1 + \|\mathbf{q}\|_0). \end{aligned}$$

This energy balance gives rise to the *minus-one norm* LSP

$$\begin{aligned} \{J, X\} \to \Big\{ &\min_{(u,\mathbf{q}) \in X} J(u, \mathbf{q}; f),\quad J(u, \mathbf{q}; f) \\ &= \|\nabla \cdot \mathbf{q} - f\|_{-1}^2 + \|\nabla u + \mathbf{q}\|_0^2 \Big\}. \end{aligned} \tag{14}$$

However, (14) is still impractical because the norm-generating operator $\mathcal{S}_{H^{-1}} = (-\Delta)^{-1/2}$ is not computable in general. The simple approximation $\mathcal{S}_{H^{-1}}^h = h\mathbf{I}$ yields the *weighted* LSFEM

$$\{J^h, X^h\} \to \left\{ \min_{(u^h,\mathbf{q}^h)\in X^h} J^h(u^h, \mathbf{q}^h; f), \quad J^h(u^h, \mathbf{q}^h; f) = h^2\|\nabla\cdot\mathbf{q}^h - f\|_0^2 + \|\nabla u^h + \mathbf{q}^{\mathbf{h}}\|_0^2 \right\} \tag{15}$$

which is quasi-norm equivalent. The more accurate approximation $\mathcal{S}^h_{H^{-1}} = h\mathbf{I} + \mathbf{K}^{h^{1/2}}$, where $\mathbf{K}^h$ is a spectrally equivalent preconditioner for $-\Delta$ gives the *discrete minus-one norm* LSFEM

$$\{J^h, X^h\} \to \left\{ \min_{(u^h,\mathbf{q}^h)\in X^h} J^h(u^h, \mathbf{q}^h; f), \quad J^h(u^h, \mathbf{q}^h; f) = \|\nabla\cdot\mathbf{q}^h - f\|_{-h}^2 + \|\nabla u^h + \mathbf{q}^h\|_0^2 \right\} \tag{16}$$

which is norm equivalent.

The first-order system (13) also has the energy balance

$$\alpha(\|u\|_1 + \|\mathbf{q}\|_{\mathrm{div}}) \le \|\nabla\cdot\mathbf{q}\|_0 + \|\nabla u + \mathbf{q}\|_0 \le \beta(\|u\|_1 + \|\mathbf{q}\|_{\mathrm{div}})$$

which corresponds to $X = H_0^1(\Omega)\times H(\mathrm{div},\Omega)$ and $Y = L^2(\Omega)\times[L^2(\Omega)]^d$. The associated LSP

$$\{J, X\} \to \left\{ \min_{(u,\mathbf{q})\in X} J(u,\mathbf{q}; f), \quad J(u,\mathbf{q}; f) = \|\nabla\cdot\mathbf{q} - f\|_0^2 + \|\nabla u + \mathbf{q}\|_0^2 \right\} \tag{17}$$

is practical. Approximation of the scalar u by standard nodal elements and of the vector $\mathbf{q}$ by div-conforming elements, such as Raviart-Thomas, BDM, or BDFM, yields a compliant LSFEM which under some conditions has the exact same local conservation property as the mixed Galerkin method for (13).

Reference

1. Bochev, P., Gunzburger, M.: Least Squares Finite Element Methods. Springer, Berlin (2009)

Levin Quadrature

Sheehan Olver
School of Mathematics and Statistics, The University of Sydney, Sydney, NSW, Australia

Mathematics Subject Classification

65D30; 41A60

Synonyms

Levin rule; Levin-type method

Short Definition

Levin quadrature is a method for computing highly oscillatory integrals that does not use moments.

Description

Levin quadrature is a method for calculating integrals of the form

$$I[f] = \int_a^b f(x)e^{i\omega g(x)}dx,$$

where f and g are suitably smooth functions, $i = \sqrt{-1}$, and ω is a large real number.

If u satisfies the differential equation

$$u'(x) + i\omega g'(x)u(x) = f(x), \tag{1}$$

then

$$I[f] = u(b)e^{i\omega g(b)} - u(a)e^{i\omega g(a)}.$$

In Levin quadrature we represent

$$u \approx \sum_{k=1}^{n} c_k \psi_k(x)$$

for some basis $\psi_1(x), \ldots, \psi_n(x)$, typically a polynomial basis such as monomials $\psi_k(x) = x^{k-1}$ or Chebyshev polynomials $\psi_k(x) = T_{k-1}(x)$. The coefficients $c_1, \ldots, c_n$ are determined by solving (1) using a collocation method: for a sequence of points $x_1, \ldots, x_n$ (such as Chebyshev points), solve the linear system

$$\sum_{k=1}^{n} c_k(\psi_k'(x_1) + i\omega g'(x_1)\psi_k(x_1)) = f(x_1), \ldots,$$

$$\sum_{k=1}^{n} c_k(\psi_k'(x_n) + i\omega g'(x_n)\psi_k(x_n)) = f(x_n).$$

We then have the approximation

$$I[f] \approx Q[f] = \sum_{k=1}^{n} c_k[\psi_k(b)e^{i\omega g(b)} - \psi_k(a)e^{i\omega g(a)}].$$

When $g'(x) \neq 0$ for $x \in (a, b)$, a and b are including as collocation points and f is differentiable with bounded variation, then the error of approximating $I[f]$ by $Q[f]$ decays like $O(\omega^{-2})$. If f is $m+1$ times differentiable and m collocation points are clustered like $O(\omega^{-1})$ near each endpoint, or if m derivatives at the endpoints are used in the collocation system, then the error decay improves to $O(\omega^{-m-2})$ [4].

The approach can be generalized to multivariate oscillatory integrals

$$I[f] = \int_{\Omega} f(\mathbf{x})e^{i\omega g(\mathbf{x})} d\mathbf{x},$$

where $\Omega \subset \mathbb{R}^d$, $\mathbf{x} \in \mathbb{R}^d$ and $f, g : \mathbb{R}^d \to \mathbb{R}$. On rectangular domains $\Omega = [a, b] \times [c, d]$, this consists of solving the PDE [1]

$$u_{xy} + i\omega g_y u_x + i\omega g_x u_y + (i\omega g_{xy} - \omega^2 g_x g_y)u = f$$

using collocation and approximating

$$I[f] \approx u(b, d)e^{i\omega g(b,d)} - u(a, d)e^{i\omega g(a,d)}$$
$$-u(b, c)e^{i\omega g(b,c)} + u(a, c)e^{i\omega g(a,c)}.$$

For other domains, the dimension of the integral can be reduced by solving the PDE

$$\nabla \cdot \mathbf{u} + i\omega \nabla g \cdot \mathbf{u} = f,$$

where $\mathbf{u} : \mathbb{C}^d \to \mathbb{C}^d$, so that

$$I[f] = \int_{\partial\Omega} e^{i\omega g} \mathbf{u} \cdot d\mathbf{s}.$$

Iterating the procedure reduces the integral to a univariate integral, at which point standard Levin quadrature is applicable [5].

Levin quadrature can be generalized to other oscillators which satisfy a linear differential equation, such as Bessel functions or Airy functions. We refer the reader to [2, 3, 6, 7].

References

1. Levin, D.: Procedure for computing one- and two-dimensional integrals of functions with rapid irregular oscillations. Math. Comp. **38**, 531–538 (1982)
2. Levin, D.: Fast integration of rapidly oscillatory functions. J. Comput. Appl. Math. **67**, 95–101 (1996)
3. Levin, D.: Analysis of a collocation method for integrating rapidly oscillatory functions. J. Comput. Appl. Math. **78**, 131–138 (1997)
4. Olver, S.: Moment-free numerical integration of highly oscillatory functions. IMA J. Num. Anal. **26**, 213–227 (2006)
5. Olver, S.: On the quadrature of multivariate highly oscillatory integrals over non-polytope domains. Numer. Math. **103**, 643–665 (2006)
6. Olver, S.: Numerical approximation of vector-valued highly oscillatory integrals. BIT **47**, 637–655 (2007)
7. Xiang, S.: Numerical analysis of a fast integration method for highly oscillatory functions. BIT **47**, 469–482 (2007)

Lie Group Integrators

Hans Z. Munthe-Kaas
Department of Mathematics, University of Bergen, Bergen, Norway

Synopsis

Lie group integrators (LGIs) are numerical time integration methods for differential equations evolving on smooth manifolds, where the time-stepping is computed from a Lie group acting on the domain. LGIs are constructed from basic mathematical operations in Lie algebras, Lie groups, and group actions. An extensive survey is found in [12].

Classical integrators (Runge-Kutta and multistep methods) can be understood as special cases of Lie group integrators, where the Euclidean space $\mathbb{R}^n$ acts upon itself by translation; thus in each time step, the solution is updated by adding an update vector, e.g., Euler method for $\dot{y}(t) = f(y(t))$, for $y, f(y) \in \mathbb{R}^n$ steps forwards from t to $t + h$ as

$$y_{n+1} = y_n + hf(y_n).$$

Consider instead a differential equation evolving on the surface of a sphere, $\dot{z}(t) = v(z) \times z(t)$, where $z, v \in \mathbb{R}^3$ and $\times$ denotes the vector product. Let $\hat{v}$ denote the *hat map*, a skew-symmetric matrix given as

$$\hat{v} := \begin{pmatrix} 0 & -v(3) & v(2) \\ v(3) & 0 & -v(1) \\ -v(2) & v(1) & 0 \end{pmatrix}, \qquad (1)$$

we can write the equation as $\dot{z}(t) = \widehat{v(z)}z(t)$. By freezing $\hat{v}$ at z_n, we obtain a step of the *exponential Euler method* as

$$z_{n+1} = \exp(h\hat{v}(z_n))z_n.$$

Here $\exp(h\hat{v}(z_n))$ is the matrix exponential of a skew-symmetric matrix. This is an orthogonal matrix which acts on the vector z_n as a rotation, and hence z_{n+1} sits exactly on the sphere. This is the simplest (nonclassical) example of a Lie group integrator.

In the cases where the Lie groups are matrix groups, LGIs are numerical integrators based on matrix commutators and matrix exponentials and are thus related to exponential integrators. The general framework of LGI may also be applied in very general situations where Lie group actions are given in terms of differential equations. The performance of LGIs depends on how efficiently the basic operations can be computed and how well the Lie group action approximates the dynamics of the system to be solved. In many cases, a good choice of action leads to small local errors, and a higher cost per step can be compensated by the possibility of taking longer time steps, compared to classical integrators.

Lie group methods are by construction preserving the structure of the underlying manifold M. Since all operations are intrinsic, it is not possible to drift off M. Furthermore, these methods are equivariant with respect to the group action, e.g., in the example of the sphere, the methods will not impose any particular coordinate system or orientation on the domain, and all points in the domain are treated equivalently.

Building Blocks

Applications of LGI generally involve the following steps:

1. Choose a Lie group and Lie group action which can be computed fast and which captures some essential features of the problem to be solved. This is similar to the task of finding a preconditioner in iterative solution of linear algebraic equations.
2. Identify the Lie algebra, commutator, and exponential map of the Lie group action.
3. Write the differential equation in terms of the infinitesimal Lie algebra action, as in (2) below.
4. Choose a Lie group integrator, plug in all building blocks, and solve the problem.

We briefly review the definition of these objects and illustrate by examples below. A *group* is a set G with an identity element $e \in G$ and associative group product $a, b \mapsto ab$ such that every $a \in G$ has a multiplicative inverse $a^{-1}a = aa^{-1} = e$. A left *group action* of G on a set M is a map $\cdot : G \times M \to M$ such that $e \cdot p = p$ and $(ab) \cdot p = a \cdot (b \cdot p)$ for all $a, b \in G$ and $p \in M$. A *Lie group* is a group G which also has the structure

of a smooth differentiable manifold such that the map $a, b \mapsto a^{-1}b$ is smooth. If M also is a manifold, then a smooth group action is called a *Lie group action*.

The *Lie algebra* $\mathfrak{g}$ of a Lie group G is the tangent space of G at the identity e, i.e., $\mathfrak{g}$ is the vector space obtained by taking the derivative at $t = 0$ of all smooth curves $\gamma(t) \in G$ such that $\gamma(0) = e$:

$$\mathfrak{g} = \{V = \dot{\gamma}(0) \,:\, \gamma(t) \in G, \gamma(0) = e\} \equiv T_e G.$$

By differentiation, we define the *infinitesimal Lie algebra action* $\cdot : \mathfrak{g} \times M \to TM$ which for $V \in \mathfrak{g}$ and $p \in M$ produces a tangent $V \cdot p \in T_p M$ as

$$V \cdot p = \left.\frac{\partial}{\partial t}\right|_{t=0} (\gamma(t) \cdot p) \in T_p M, \quad \text{where } V = \dot{\gamma}(0).$$

The *exponential map* $\exp: \mathfrak{g} \to G$ is the $t = 1$ flow of the infinitesimal action; more precisely, we define $\exp(V) \in G$ as $\exp(V) := y(1)$, where $y(t) \in G$ is the solution of the initial value problem

$$\dot{y}(t) = V \cdot y(t), \quad y(0) = e.$$

The final operation we need in order to define a Lie group method is the *commutator* or *Lie bracket*, a bilinear map $[\cdot,\cdot]: \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ defined for $V, W \in \mathfrak{g}$ as

$$[V, W] = \left.\frac{\partial^2}{\partial s \partial t}\right|_{s=t=0} \exp(sV)\exp(tW)\exp(-sV).$$

The commutator measures infinitesimally the extent to which two flows $\exp(sV)$ and $\exp(tW)$ fail to commute. We denote ad_V the linear operator $W \mapsto [V, W]: \mathfrak{g} \to \mathfrak{g}$.

In the important case where G is a matrix Lie group, the exponential is the matrix exponential and the commutator is the matrix commutator $[V, W] = VW - WV$. If G acts on a vector space M by matrix multiplication $a \cdot p = ap$, then also the infinitesimal Lie algebra action $V \cdot p = Vp$ is given by matrix multiplication.

Definition

Given a smooth manifold M and a Lie group G with Lie algebra $\mathfrak{g}$ acting on M. Consider a differential equation for $y(t) \in M$ written in terms of the infinitesimal action as

$$\dot{y}(t) = f(t, y) \cdot y, \quad y(0) = y_0, \tag{2}$$

for a given function $f: \mathbb{R} \times M \to \mathfrak{g}$. A *Lie group integrator* is a numerical time-stepping procedure for (2) which is based on intrinsic Lie group operations, such as exponentials, commutators, and the group action on M.

Methods (Examples)

Lie Euler: $y_{n+1} = \exp(hf(t_n, y_n)) \cdot y_n$.
Lie midpoint:

$$\begin{aligned} K &= hf\,(t_n + h/2, \exp(K/2) \cdot y_n) \\ y_{n+1} &= \exp(K) \cdot y_n \end{aligned}$$

Lie RK4: There are several similar ways of turning the classical RK4 method into a 4 order Lie group integrator [16,18]. The following version requires only two commutators:

$$\begin{aligned} K_1 &= hf(t_n, y_n) \\ K_2 &= hf(t_n/2, \exp(K_1/2) \cdot y_n) \\ K_3 &= hf\,(t_n + h/2, \exp(K_2/2 - [K_1, K_2]/8) \cdot y_n) \\ K_4 &= hf(t_n + h/2, \exp(K_3) \cdot y_n) \\ y_{n+1} &= \exp\,(K_1/6 + K_2/3 + K_3/3 + K_4/6 \\ &\qquad -[K_1, K_2]/3 - [K_1, K_4]/12) \cdot y_n \end{aligned}$$

RKMK methods: This is a general procedure to turn any classical Runge-Kutta method into a Lie group integrator of the same order. Given the coefficients $a_{j,\ell}, b_j, c_j$ of an s-stage and pth order RK method, a single step $y(t_n) \approx y_n \mapsto y_{n+1} \approx y(t_n + h)$ is given as

$$\left.\begin{aligned} U_j &= \sum_{\ell=1}^{s} a_{j,\ell} K_\ell \\ F_j &= hf(t_n + c_j h, \exp(U_j) \cdot y_n) \\ K_j &= d\exp_{U_j}^{-1}(F_j) \end{aligned}\right\} j = 1, \ldots, s$$

$$y_{n+1} = \exp\left(\sum_{\ell=1}^{s} b_\ell K_\ell\right) \cdot y_n,$$

where $d\exp^{-1}_{U_j}(F_j) = F_j - \frac{1}{2}[U_j, F_j] + \frac{1}{12}[U_j,[U_j,F_j]] - \frac{1}{720}\mathrm{ad}^4_{U_j}F_j + \cdots = \sum_{j=0}^{p}\frac{B_j}{j!}\mathrm{ad}^j_{U_j}F_j$ is the inverse of the Darboux derivative of the exponential map, truncated to the order of the method and B_j are the Bernoulli numbers [12, 17].

Crouch-Grossman and commutator-free methods: Commutators pose a problem in the application of Lie group integrators to stiff equations, since the commutator often increases the stiffness of the equations dramatically. Crouch-Grossman [6, 19] and more generally commutator-free methods [5] avoid commutators by doing basic time-stepping using a composition of exponentials. An example of such a method is CF4 [5]:

$$\begin{aligned}
K_1 &= hf(t_n, y_n)\\
K_2 &= hf(t_n/2, \exp(K_1/2)\cdot y_n)\\
K_3 &= hf\ (t_n + h/2, \exp(K_2/2)\cdot y_n)\\
K_4 &= hf(t_n + h/2, \exp(K_1/2)\cdot\\
&\quad \exp(K_3 - K_1/2)\cdot y_n)\\
y_{n+1} &= \exp\left(K_1/4 + K_2/6 + K_3/6 - K_4/12\right)\cdot\\
&\quad \exp\left(K_2/6 + K_3/6 + K_4/4 - K_1/12\right)\cdot y_n
\end{aligned}$$

Magnus methods: In the case where $f(t,y) = f(t)$ is a function of time alone, then (2) is called an equation of *Lie type*. Specialized numerical methods have been developed for such problems [1, 10]. Explicit Magnus methods can achieve order 2p using only p function evaluations, and they are also easily designed to be time symmetric.

Lie Group Actions (Examples)

Rotational problems: Consider a differential equation $\dot{y}(t) = v(y(t)) \times y(t)$, where $y, v \in \mathbb{R}^2$ and $||y(0)|| = 1$. Since $||y(t)|| = 1$ for all t, we can take M to be the surface of the unit sphere. Let $G = SO(3)$ be the special orthogonal group, consisting of all orthogonal matrices with determinant 1. Let $\gamma(t) \in G$ be a curve such that $\gamma(0) = e$. By differentiating $\gamma(t)^T\gamma(t) = e$, we find that $\dot{\gamma}(0)^T + \gamma(0) = 0$, thus $\mathfrak{g} = \mathfrak{so}(3)$, the set of all skew-symmetric 3×3 matrices. The infinitesimal Lie algebra action is left multiplication with a skew matrix, the commutator is the matrix commutator, and the exponential map is the matrix exponential. Written in terms of the infinitesimal Lie algebra action, the differential equation becomes $\dot{y} = \widehat{v(y)}y$, and we may apply any Lie group integrator. Note that for low-dimensional rotational problems, all basic operations can be computed fast using Rodrigues-type formulas [12].

Isospectral action: Isospectral differential equations are matrix-valued equations where the eigenvalues are first integrals (invariants of motion). Consider $M = \mathbb{R}^{n\times n}$ and the action of $G = SO(n)$ on M by similarity transforms, i.e., for $a \in G$ and $y \in M$, we define $a \cdot y = aya^T$. By differentiation, of the action we find the infinitesimal action for $V \in \mathfrak{g} = \mathfrak{so}(n)$ as $V \cdot y = Vy - yV$; thus for this action, (2) becomes

$$\dot{y}(t) = f(t,y)\cdot y = f(t,y)y - yf(t,y),$$

where $f : \mathbb{R}\times M \to \mathfrak{g}$. See [2, 12] for more details.

Affine action: Let $G = Gl(n) \rtimes \mathbb{R}^n$ be the *affine linear group*, consisting of all pairs a, b where $a \in \mathbb{R}^{n\times n}$ is an invertible matrix and $b \in \mathbb{R}^n$ is a vector. The *affine action* of G on $M = \mathbb{R}^n$ is $(a,b)\cdot y = ay + b$. The Lie algebra of G is $\mathfrak{g} = \mathfrak{gl}(n) \rtimes \mathbb{R}^n$, i.e., $\mathfrak{g}$ consists of all pairs (V,b) where $V \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. The infinitesimal action is given as $(V,b)\cdot y = Vy + b$. This action is useful for differential equations of the form $\dot{y}(t) = L(t)y + N(y)$, where $L(t)$ is a stiff linear part and N is a nonstiff nonlinear part. Such equations are cast in the form (2) by choosing $f(t,y) = (L(t), N(y))$. Applications of Lie group integrators to such problems are closely related to exponential integrators. For stiff equations it is important to use a commutator-free Lie group method.

Coadjoint action: Many problems of computational mechanics are naturally formulated as Lie-Poisson systems, evolving on coadjoint orbits of the dual of a Lie algebra [14]. Lie group integrators based on the coadjoint action of a Lie group on the dual of its Lie algebra are discussed in [7].

Classical integrators as Lie group integrators: The simplest of all group actions is when $G = M = \mathbb{R}^n$, with vector addition as group operation and group action. From the definitions, we find that in this case $\mathfrak{g} = \mathbb{R}^n$, the commutator is 0, and the exponential map is the identity map from $\mathbb{R}^n$ to itself. The infinitesimal Lie algebra action becomes $V \cdot y = V$; thus, (2) reduces to $\dot{y}(t) = f(t,y)$, where $f(t,y) \in \mathbb{R}^n$. We see that classical integration methods are special cases of

Lie group integrators, and all the examples of methods above reduce to well-known Runge-Kutta methods.

Implementation Issues

For efficient implementation of LGI, it is important to employ fast algorithms for computing commutators and exponentials. A significant volume of research has been devoted to this. Important techniques involve replacing the exponential map with other coordinate maps on Lie groups [13, 20]. For special groups, there exist specialized algorithms for computing matrix exponentials [4, 21]. Time reversible LGI is discussed in [22], but these are all implicit methods and thus costly. Optimization of the number of commutators and exponentials has been considered in [3, 18].

References

1. Blanes, S., Casas, F., Oteo, J., Ros, J.: The Magnus expansion and some of its applications. Phys. Rep. **470**(5–6), 151–238 (2009)
2. Calvo, M., Iserles, A., Zanna, A.: Numerical solution of isospectral flows. Math. Comput. **66**(220), 1461–1486 (1997)
3. Casas, F., Owren, B.: Cost efficient Lie group integrators in the RKMK class. BIT Numer. Math. **43**(4), 723–742 (2003)
4. Celledoni, E., Iserles, A.: Methods for the approximation of the matrix exponential in a Lie-algebraic setting. IMA J. Numer. Anal. **21**(2), 463 (2001)
5. Celledoni, E., Marthinsen, A., Owren, B.: Commutator-free Lie group methods. Future Gen. Comput. Syst. **19**(3), 341–352 (2003)
6. Crouch, P., Grossman, R.: Numerical integration of ordinary differential equations on manifolds. J. Nonlinear Sci. **3**(1), 1–33 (1993)
7. Engø, K., Faltinsen, S.: Numerical integration of Lie-Poisson systems while preserving coadjoint orbits and energy. SIAM J. Numer. Anal. **39**, 128–145 (2002)
8. Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations, vol 31. Springer, Berlin/New York (2006)
9. Hochbruck, M., Ostermann, A.: Exponential integrators. Acta Numer. **19**, 209–286 (2010)
10. Iserles, A., Nørsett, S.: On the solution of linear differential equations in Lie groups. Philos. Trans. A **357**(1754), 983 (1999)
11. Iserles, A., Zanna, A.: Efficient computation of the matrix exponential by generalized polar decompositions. SIAM J. Numer. Anal. **42**, 2218–2256 (2005)
12. Iserles, A., Munthe-Kaas, H., Nørsett, S., Zanna, A.: Lie-group methods. Acta Numer. **9**(1), 215–365 (2000)
13. Krogstad, S., Munthe-Kaas, H., Zanna, A.: Generalized polar coordinates on Lie groups and numerical integrators. Numer. Math. **114**(1), 161–187 (2009)
14. Marsden, J., Raţiu, T.: Introduction to Mechanics and Symmetry: A Basic Exposition of Classical Mechanical Systems. Springer, New York (1999)
15. Munthe-Kaas, H.: Lie-Butcher theory for Runge-Kutta methods. BIT Numer. Math. **35**(4), 572–587 (1995)
16. Munthe-Kaas, H.: Runge-Kutta methods on Lie groups. BIT Numer. Math. **38**(1), 92–111 (1998)
17. Munthe-Kaas, H.: High order Runge-Kutta methods on manifolds. Appl. Numer. Math. **29**(1), 115–127 (1999)
18. Munthe-Kaas, H., Owren, B.: Computations in a free Lie algebra. Philos. Trans. R. Soc. Lond. Ser. A **357**(1754), 957 (1999)
19. Owren, B., Marthinsen, A.: Runge-Kutta methods adapted to manifolds and based on rigid frames. BIT Numer. Math. **39**(1), 116–142 (1999)
20. Owren, B., Marthinsen, A.: Integration methods based on canonical coordinates of the second kind. Numer. Math. **87**(4), 763–790 (2001)
21. Zanna, A., Munthe-Kaas, H.: Generalized polar decompositions for the approximation of the matrix exponential. SIAM J. Matrix Anal. Appl. **23**, 840 (2002)
22. Zanna, A., Engø, K., Munthe-Kaas, H.: Adjoint and selfadjoint Lie-group methods. BIT Numer. Math. **41**(2), 395–421 (2001)

Linear Elastostatics

Tarek I. Zohdi
Department of Mechanical Engineering, University of California, Berkeley, CA, USA

Notation

Throughout this work, boldface symbols denote vectors or tensors. For the inner product of two vectors (first-order tensors), $\mathbf{u}$ and $\mathbf{v}$, we have $\mathbf{u} \cdot \mathbf{v} = u_i v_i = u_1 v_1 + u_2 v_2 + u_3 v_3$ in three dimensions, where Cartesian basis and Einstein index summation notation are used. In this introduction, for clarity of presentation, *we will ignore the difference between second-order tensors and matrices.* Furthermore, we exclusively employ a Cartesian basis. Accordingly, if we consider the second-order tensor $\mathbf{A} = A_{ik}\, \mathbf{e}_i \otimes \mathbf{e}_k$, then a first-order contraction (inner product) of two second-order tensors $\mathbf{A} \cdot \mathbf{B}$ is defined by the matrix product $[\mathbf{A}][\mathbf{B}]$, with components of $A_{ij}B_{jk} = C_{ik}$. It is clear that the range of the inner index j must be the same for $[\mathbf{A}]$ and $[\mathbf{B}]$. For three dimensions, we have $i, j = 1, 2, 3$. The second-order inner product of two tensors or matrices is defined as $\mathbf{A} : \mathbf{B} = A_{ij}B_{ij} = tr([\mathbf{A}]^T [\mathbf{B}])$.

Kinematics of Deformations

The term deformation refers to a change in the shape of a continuum between a reference configuration and current configuration. In the reference configuration, a representative particle of a continuum occupies a point P in space and has the position vector (Fig. 1)

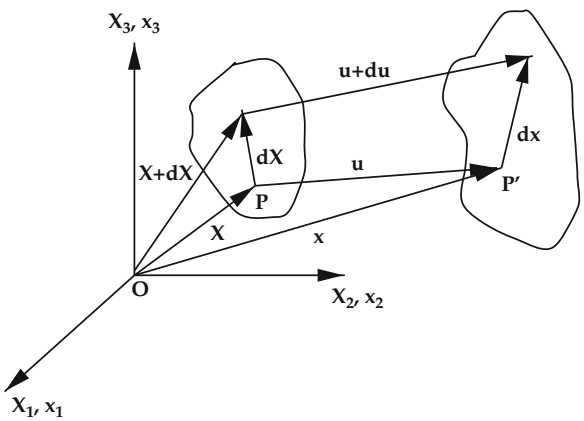
$$\mathbf{X} = X_1\mathbf{e}_1 + X_2\mathbf{e}_2 + X_3\mathbf{e}_3, \tag{1}$$

where $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ is a Cartesian reference triad and X_1, X_2, X_3 (with center O) can be thought of as labels for a material point. Sometimes the coordinates or labels (X_1, X_2, X_3) are called the referential or material coordinates. In the current configuration, the particle originally located at point P (at time $t = 0$) is located at point P' and can be also expressed in terms of another position vector $\mathbf{x}$, with coordinates (x_1, x_2, x_3). These are called the current coordinates. In this framework, the displacement is $\mathbf{u} = \mathbf{x} - \mathbf{X}$ for a point originally at $\mathbf{X}$ and with final coordinates $\mathbf{x}$.

When a continuum undergoes deformation (or flow), its points move along various paths in space. This motion may be expressed as a function of $\mathbf{X}$ and t as (Frequently, analysts consider the referential configuration to be fixed in time, thus, $\mathbf{X} \neq \mathbf{X}(t)$.)

$$\mathbf{x}(\mathbf{X}, t) = \mathbf{u}(\mathbf{X}, t) + \mathbf{X}(t)\,, \tag{2}$$

which gives the present location of a point at time t, written in terms of the referential coordinates X_1, X_2, X_3. The previous position vector may be interpreted as a mapping of the initial configuration onto the current configuration. In classical approaches, it is assumed that such a mapping is one to one and continuous, with continuous partial derivatives to whatever order is required. The description of motion or deformation expressed previously is known as the Lagrangian formulation. Alternatively, if the independent variables are the coordinates $\mathbf{x}$ and time t, then $\mathbf{x}(x_1, x_2, x_3, t) = \mathbf{u}(x_1, x_2, x_3, t) + \mathbf{X}(x_1, x_2, x_3, t)$, and the formulation is denoted as Eulerian (Fig. 1).



Linear Elastostatics, Fig. 1 Different descriptions of a deforming body

Deformation of Line Elements

Partial differentiation of the displacement vector $\mathbf{u} = \mathbf{x} - \mathbf{X}$, with respect to $\mathbf{X}$, produces the following displacement gradient:

$$\nabla_X \mathbf{u} = \mathbf{F} - \mathbf{1}, \tag{3}$$

where

$$\mathbf{F} \overset{\text{def}}{=} \nabla_X \mathbf{x} \overset{\text{def}}{=} \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} & \frac{\partial x_1}{\partial X_3} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} & \frac{\partial x_2}{\partial X_3} \\ \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_2} & \frac{\partial x_3}{\partial X_3} \end{bmatrix}. \tag{4}$$

$\mathbf{F}$ is known as the material deformation gradient.

Now, consider the length of a differential element in the reference configuration $d\mathbf{X}$ and $d\mathbf{x}$ in the current configuration, $d\mathbf{x} = \nabla_X \mathbf{x} \cdot d\mathbf{X} = \mathbf{F} \cdot d\mathbf{X}$. Taking the difference in the squared magnitudes of these elements yields

$$\begin{aligned} d\mathbf{x} \cdot d\mathbf{x} - d\mathbf{X} \cdot d\mathbf{X} &= (\nabla_X \mathbf{x} \cdot d\mathbf{X}) \cdot (\nabla_X \mathbf{x} \cdot d\mathbf{X}) \\ &\quad - d\mathbf{X} \cdot d\mathbf{X} \\ &= d\mathbf{X} \cdot (\mathbf{F}^T \cdot \mathbf{F} - \mathbf{1}) \cdot d\mathbf{X} \\ &\overset{\text{def}}{=} 2\, d\mathbf{X} \cdot \mathbf{E} \cdot d\mathbf{X}. \end{aligned} \tag{5}$$

Equation (5) defines the so-called strain tensor:

$$\begin{aligned} \mathbf{E} &\overset{\text{def}}{=} \frac{1}{2}(\mathbf{F}^T \cdot \mathbf{F} - \mathbf{1}) \\ &= \frac{1}{2}[\nabla_X \mathbf{u} + (\nabla_X \mathbf{u})^T + (\nabla_X \mathbf{u})^T \cdot \nabla_X \mathbf{u}]. \end{aligned} \tag{6}$$

Remark 1 It should be clear that $d\mathbf{x}$ can be reinterpreted as the result of a mapping $\mathbf{F} \cdot d\mathbf{X} \rightarrow d\mathbf{x}$ or a change in configuration (reference to current). One may develop so-called Eulerian formulations, employing the current configuration coordinates to generate Eulerian strain tensor measures. An important quantity is the Jacobian of the deformation gradient, $J \stackrel{\text{def}}{=} det\,\mathbf{F}$, which relates differential volumes in the reference configuration ($d\omega_0$) to differential volumes in the current configuration ($d\omega$) via $d\omega = J\, d\omega_0$. The Jacobian of the deformation gradient must remain positive; otherwise, we obtain physically impossible "negative" volumes. For more details, we refer the reader to the texts of Malvern [3], Gurtin [2], and Chandrasekharaiah and Debnath [1].

Equilibrium/Kinetics of Solid Continua

The balance of linear momentum in the deformed (current) configuration is

$$\underbrace{\int_{\partial\omega} \mathbf{t}\, da}_{\text{surface forces}} + \underbrace{\int_{\omega} \rho\mathbf{b}\, d\omega}_{\text{body forces}} = \underbrace{\frac{d}{dt}\int_{\omega} \rho\dot{\mathbf{u}}\, d\omega}_{\text{inertial forces}}, \tag{7}$$

where $\omega \subset \Omega$ is an arbitrary portion of the continuum, with boundary $\partial\omega$, ρ is the material density, $\mathbf{b}$ is the body force per unit mass, and $\dot{\mathbf{u}}$ is the time derivative of the displacement. The force densities, $\mathbf{t}$, are commonly referred to as "surface forces" or tractions.

Postulates on Volume and Surface Quantities

Now, consider a tetrahedron in equilibrium, as shown in Fig. 2, where a balance of forces yields

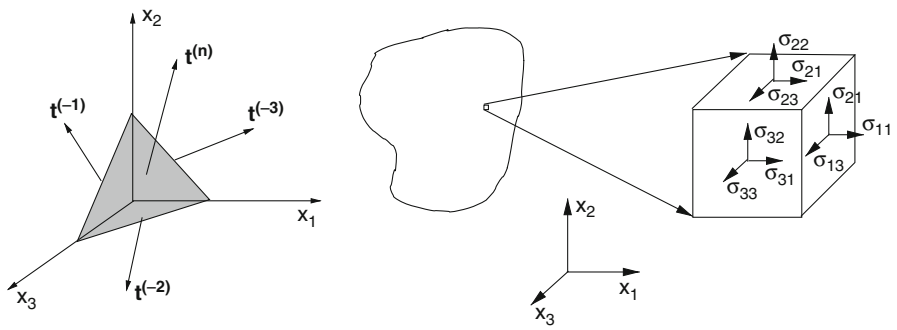
$$\mathbf{t}^{(n)}\Delta A^{(n)} + \mathbf{t}^{(-1)}\Delta A^{(1)} + \mathbf{t}^{(-2)}\Delta A^{(2)} + \mathbf{t}^{(-3)}\Delta A^{(3)} + \rho\mathbf{b}\Delta V = \rho\Delta V\ddot{\mathbf{u}}, \tag{8}$$

where $\Delta A^{(n)}$ is the surface area of the face of the tetrahedron with normal $\mathbf{n}$ and ΔV is the tetrahedron volume. As the distance (h) between the tetrahedron base (located at (0,0,0)) and the surface center goes to zero ($h \rightarrow 0$), we have $\Delta A^{(n)} \rightarrow 0 \Rightarrow \frac{\Delta V}{\Delta A^{(n)}} \rightarrow 0$. Geometrically, we have $\frac{\Delta A^{(i)}}{\Delta A^{(n)}} = \cos(x_i, x_n) \stackrel{\text{def}}{=} n_i$, and therefore $\mathbf{t}^{(n)} + \mathbf{t}^{(-1)}\cos(x_1, x_n) + \mathbf{t}^{(-2)}\cos(x_2, x_n) + \mathbf{t}^{(-3)}\cos(x_3, x_n) = \mathbf{0}$. It is clear that forces on the surface areas could be decomposed into three linearly independent components. It is convenient to introduce the concept of stress at a point, representing the surface forces there, pictorially represented by a cube surrounding a point. The fundamental issue that must be resolved is the characterization of these surface forces. We can represent the surface force density vector, the so-called traction, on a surface by the component representation:

$$\mathbf{t}^{(i)} \stackrel{\text{def}}{=} \begin{Bmatrix} \sigma_{i1} \\ \sigma_{i2} \\ \sigma_{i3} \end{Bmatrix}, \tag{9}$$

where the second index represents the direction of the component and the first index represents components of the normal to corresponding coordinate plane. Henceforth, we will drop the superscript notation of $\mathbf{t}^{(n)}$, where it is implicit that $\mathbf{t} \stackrel{\text{def}}{=} \mathbf{t}^{(n)} = \boldsymbol{\sigma}^T \cdot \mathbf{n}$, where

$$\boldsymbol{\sigma} \stackrel{\text{def}}{=} \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}, \tag{10}$$



Linear Elastostatics, Fig. 2 (*Left*) Cauchy tetrahedron: a "sectioned point" and (*Right*) stress at a point.

or explicitly ($\mathbf{t}^{(1)} = -\mathbf{t}^{(-1)}$, $\mathbf{t}^{(2)} = -\mathbf{t}^{(-2)}$, $\mathbf{t}^{(3)} = -\mathbf{t}^{(-3)}$)

$$\begin{aligned}\mathbf{t} &= \mathbf{t}^{(1)}n_1 + \mathbf{t}^{(2)}n_2 + \mathbf{t}^{(3)}n_3 = \boldsymbol{\sigma}^T \cdot \mathbf{n} \\ &= \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}^T \begin{Bmatrix} n_1 \\ n_2 \\ n_3 \end{Bmatrix}, \end{aligned} \tag{11}$$

where $\boldsymbol{\sigma}$ is the so-called Cauchy stress tensor.

Remark 2 In the absence of couple stresses, a balance of angular momentum implies a symmetry of stress, $\boldsymbol{\sigma} = \boldsymbol{\sigma}^T$, and thus the difference in notations becomes immaterial. Explicitly, starting with an angular momentum balance, under the assumptions that no infinitesimal "micro-moments" or so-called couple stresses exist, then it can be shown that the stress tensor must be symmetric, i.e., $\int_{\partial\omega} \mathbf{x}\times\mathbf{t}\,da + \int_\omega \mathbf{x}\times\rho\mathbf{b}\,d\omega = \frac{d}{dt}\int_\omega \mathbf{x}\times\rho\dot{\mathbf{u}}\,d\omega$; that is, $\boldsymbol{\sigma}^T = \boldsymbol{\sigma}$. It is somewhat easier to consider a differential element, such as in Fig. 2, and to simply sum moments about the center. Doing this, one immediately obtains $\sigma_{12} = \sigma_{21}, \sigma_{23} = \sigma_{32}$, and $\sigma_{13} = \sigma_{31}$. Consequently, $\mathbf{t} = \boldsymbol{\sigma}\cdot\mathbf{n} = \boldsymbol{\sigma}^T\cdot\mathbf{n}$.

Balance Law Formulations

Substitution of (11) into (7) yields ($\omega \subset \Omega$)

$$\underbrace{\int_{\partial\omega} \boldsymbol{\sigma}\cdot\mathbf{n}\,da}_{\text{surface forces}} + \underbrace{\int_\omega \rho\mathbf{b}\,d\omega}_{\text{body forces}} = \underbrace{\frac{d}{dt}\int_\omega \rho\dot{\mathbf{u}}\,d\omega}_{\text{inertial forces}}. \tag{12}$$

A relationship can be determined between the densities in the current and reference configurations, $\int_\omega \rho\,d\omega = \int_{\omega_0}\rho J\,d\omega_0 = \int_{\omega_0}\rho_0\,d\omega_0$. Therefore, the Jacobian can also be interpreted as the ratio of material densities at a point. Since the volume is arbitrary, we can assume that $\rho J = \rho_0$ holds at every point in the body. Therefore, we may write $\frac{d}{dt}(\rho_0) = \frac{d}{dt}(\rho J) = 0$, when the system is mass conservative over time. This leads to writing the last term in (12) as $\frac{d}{dt}\int_\omega \rho\dot{\mathbf{u}}\,d\omega = \int_{\omega_0}\frac{d(\rho J)}{dt}\dot{\mathbf{u}}\,d\omega_0 + \int_{\omega_0}\rho\ddot{\mathbf{u}}J\,d\omega_0 = \int_\omega \rho\ddot{\mathbf{u}}\,d\omega$. From Gauss's divergence theorem and an implicit assumption that $\boldsymbol{\sigma}$ is differentiable, we have $\int_\omega (\nabla_x\cdot\boldsymbol{\sigma} + \rho\mathbf{b} - \rho\ddot{\mathbf{u}})\,d\omega = \mathbf{0}$. If the volume is argued as being arbitrary, then the integrand must be equal to zero at every point, yielding

$$\nabla_x\cdot\boldsymbol{\sigma} + \rho\mathbf{b} = \rho\ddot{\mathbf{u}}. \tag{13}$$

The First Law of Thermodynamics: An Energy Balance

The interconversions of mechanical, thermal, and chemical energy in a system are governed by the first law of thermodynamics, which states that the time rate of change of the total energy, $\mathcal{K} + \mathcal{I}$, is equal to the mechanical power, $\mathcal{P}$, and the net heat supplied, $\mathcal{H} + \mathcal{Q}$, i.e., $\frac{d}{dt}(\mathcal{K} + \mathcal{I}) = \mathcal{P} + \mathcal{H} + \mathcal{Q}$. Here the kinetic energy of a subvolume of material contained in Ω, denoted ω, is $\mathcal{K} \overset{\text{def}}{=} \int_\omega \frac{1}{2}\rho\dot{\mathbf{u}}\cdot\dot{\mathbf{u}}\,d\omega$; the power (rate of work) of the external forces acting on ω is given by $\mathcal{P} \overset{\text{def}}{=} \int_\omega \rho\mathbf{b}\cdot\dot{\mathbf{u}}\,d\omega + \int_{\partial\omega}\boldsymbol{\sigma}\cdot\mathbf{n}\cdot\dot{\mathbf{u}}\,da$; the heat flow into the volume by conduction is $\mathcal{Q} \overset{\text{def}}{=} -\int_{\partial\omega}\mathbf{q}\cdot\mathbf{n}\,da = -\int_\omega \nabla_x\cdot\mathbf{q}\,d\omega$, $\mathbf{q}$ being the heat flux; the heat generated due to sources, *such as chemical reactions*, is $\mathcal{H} \overset{\text{def}}{=} \int_\omega \rho z\,d\omega$, where z is the reaction source rate per unit mass; and the internal energy is $\mathcal{I} \overset{\text{def}}{=} \int_\omega \rho\mathrm{w}\,d\omega$, w being the internal energy per unit mass. Differentiating the kinetic energy yields

$$\begin{aligned}\frac{d\mathcal{K}}{dt} &= \frac{d}{dt}\int_\omega \frac{1}{2}\rho\dot{\mathbf{u}}\cdot\dot{\mathbf{u}}\,d\omega \\ &= \int_{\omega_0}\frac{d}{dt}\frac{1}{2}(\rho J\dot{\mathbf{u}}\cdot\dot{\mathbf{u}})\,d\omega_0 \\ &= \int_{\omega_0}\left(\frac{d}{dt}\rho_0\right)\frac{1}{2}\dot{\mathbf{u}}\cdot\dot{\mathbf{u}}\,d\omega_0 \\ &\quad + \int_\omega \rho\frac{d}{dt}\frac{1}{2}(\dot{\mathbf{u}}\cdot\dot{\mathbf{u}})\,d\omega \\ &= \int_\omega \rho\dot{\mathbf{u}}\cdot\ddot{\mathbf{u}}\,d\omega, \end{aligned} \tag{14}$$

where we have assumed that the mass in the system is constant. We also have

$$\begin{aligned}\frac{d\mathcal{I}}{dt} &= \frac{d}{dt}\int_\omega \rho\mathrm{w}\,d\omega = \frac{d}{dt}\int_{\omega_0}\rho J\,\mathrm{w}\,d\omega_0 \\ &= \int_{\omega_0}\underbrace{\frac{d}{dt}(\rho_0)}_{=0}\mathrm{w}\,d\omega_0 + \int_\omega \rho\dot{\mathrm{w}}\,d\omega = \int_\omega \rho\dot{\mathrm{w}}\,d\omega. \end{aligned} \tag{15}$$

By using the divergence theorem, we obtain

$$\int_{\partial\omega} \boldsymbol{\sigma}\cdot\mathbf{n}\cdot\dot{\mathbf{u}}\,da = \int_{\omega} \nabla_x\cdot(\boldsymbol{\sigma}\cdot\dot{\mathbf{u}})\,d\omega = \int_{\omega}(\nabla_x\cdot\boldsymbol{\sigma})\cdot\dot{\mathbf{u}}\,d\omega + \int_{\omega}\boldsymbol{\sigma}:\nabla_x\dot{\mathbf{u}}\,d\omega. \tag{16}$$

Combining the results, and enforcing a balance of linear momentum, leads to

$$\int_{\omega}(\rho\dot{\mathrm{w}} + \dot{\mathbf{u}}\cdot(\rho\ddot{\mathbf{u}} - \nabla_x\cdot\boldsymbol{\sigma} - \rho\mathbf{b}) - \boldsymbol{\sigma}:\nabla_x\dot{\mathbf{u}} + \nabla_x\cdot\mathbf{q} - \rho z)\,d\omega = \int_{\omega}(\rho\dot{\mathrm{w}} - \boldsymbol{\sigma}:\nabla_x\dot{\mathbf{u}} + \nabla_x\cdot\mathbf{q} - \rho z)\,d\omega = 0. \tag{17}$$

Since the volume ω is arbitrary, the integrand must hold locally and we have

$$\rho\dot{\mathrm{w}} - \boldsymbol{\sigma}:\nabla_x\dot{\mathbf{u}} + \nabla_x\cdot\mathbf{q} - \rho z = 0. \tag{18}$$

When dealing with multifield problems, this equation is used extensively.

Linearly Elastic Constitutive Equations

We now discuss relationships between the stress and strain, so-called material laws or constitutive relations for linearly elastic cases (infinitesimal deformations).

The Infinitesimal Strain Case

In infinitesimal deformation theory, the displacement gradient components are considered small enough that higher-order terms like $(\nabla_X\mathbf{u})^T\cdot\nabla_X\mathbf{u}$ and $(\nabla_x\mathbf{u})^T\cdot\nabla_x\mathbf{u}$ can be neglected in the strain measure $\mathbf{E} = \frac{1}{2}(\nabla_X\mathbf{u} + (\nabla_X\mathbf{u})^T + (\nabla_X\mathbf{u})^T\cdot\nabla_X\mathbf{u})$, leading to $\mathbf{E}\approx\boldsymbol{\epsilon}\overset{\text{def}}{=}\frac{1}{2}[\nabla_X\mathbf{u} + (\nabla_X\mathbf{u})^T]$. If the displacement gradients are small compared with unity, $\boldsymbol{\epsilon}$ coincides closely to $\mathbf{E}$. If we assume that $\frac{\partial}{\partial\mathbf{X}}\approx\frac{\partial}{\partial\mathbf{x}}$, we may use $\mathbf{E}$ or $\boldsymbol{\epsilon}$ interchangeably. Usually $\boldsymbol{\epsilon}$ is the symbol used for infinitesimal strains. Furthermore, to avoid confusion, when using models employing the geometrically linear infinitesimal strain assumption, we use the symbol of ∇ with no $\mathbf{X}$ or $\mathbf{x}$ subscript. Hence, the infinitesimal strains are defined by

$$\boldsymbol{\epsilon} = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T). \tag{19}$$

Linear Elastic Constitutive Laws

If we neglect thermal effects, (18) implies $\rho\dot{\mathrm{w}} = \boldsymbol{\sigma}:\nabla_x\dot{\mathbf{u}}$ which, in the infinitesimal strain linearly elastic case, is $\rho\dot{\mathrm{w}} = \boldsymbol{\sigma}:\dot{\boldsymbol{\epsilon}}$. From the chain rule of differentiation, we have

$$\rho\dot{\mathrm{w}} = \rho\frac{\partial\mathrm{w}}{\partial\boldsymbol{\epsilon}}:\frac{d\boldsymbol{\epsilon}}{dt} = \boldsymbol{\sigma}:\dot{\boldsymbol{\epsilon}}\Rightarrow\boldsymbol{\sigma} = \rho\frac{\partial\mathrm{w}}{\partial\boldsymbol{\epsilon}}. \tag{20}$$

The starting point to develop a constitutive theory is to assume a stored elastic energy function exists, a function denoted $W\overset{\text{def}}{=}\rho\mathrm{w}$, which depends only on the mechanical deformation. The simplest function that fulfills $\boldsymbol{\sigma} = \rho\frac{\partial\mathrm{w}}{\partial\boldsymbol{\epsilon}}$ is $W = \frac{1}{2}\boldsymbol{\epsilon}:\mathbb{E}:\boldsymbol{\epsilon}$, where $\mathbb{E}$ is the fourth-rank elasticity tensor. Such a function satisfies the intuitive physical requirement that, for any small strain from an undeformed state, energy must be stored in the material. Alternatively, a small strain material law can be derived from $\boldsymbol{\sigma} = \frac{\partial W}{\partial\boldsymbol{\epsilon}}$ and $W\approx c_0 + \mathbf{c}_1:\boldsymbol{\epsilon} + \frac{1}{2}\boldsymbol{\epsilon}:\mathbb{E}:\boldsymbol{\epsilon} + \ldots$ which implies $\boldsymbol{\sigma}\approx\mathbf{c}_1 + \mathbb{E}:\boldsymbol{\epsilon} + \ldots$. We are free to set $c_0 = 0$ (it is arbitrary) in order to have zero strain energy at zero strain, and, furthermore, we assume that no stresses exist in the reference state ($\mathbf{c}_1 = \mathbf{0}$). With these assumptions, we obtain the familiar relation

$$\boldsymbol{\sigma} = \mathbb{E}:\boldsymbol{\epsilon}. \tag{21}$$

This is a linear relation between stresses and strains. The existence of a strictly positive stored energy function in the reference configuration implies that the linear elasticity tensor must have positive eigenvalues at every point in the body. Typically, different materials are classified according to the number of independent components in $\mathbb{E}$. In theory, $\mathbb{E}$ has 81 components, since it is a fourth-order tensor relating 9 components of stress to strain. However, the number of components can be reduced to 36 since the stress and strain tensors are symmetric. This is observed from the matrix representation (The symbol $[\cdot]$ is used to indicate the matrix notation equivalent to a tensor form, while $\{\cdot\}$ is used to indicate the vector representation.) of $\mathbb{E}$:

$$\underbrace{\begin{Bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{23}\\ \sigma_{31}\end{Bmatrix}}_{\stackrel{\text{def}}{=}\{\boldsymbol{\sigma}\}} = \underbrace{\begin{bmatrix} E_{1111} & E_{1122} & E_{1133} & E_{1112} & E_{1123} & E_{1113}\\ E_{2211} & E_{2222} & E_{2233} & E_{2212} & E_{2223} & E_{2213}\\ E_{3311} & E_{3322} & E_{3333} & E_{3312} & E_{3323} & E_{3313}\\ E_{1211} & E_{1222} & E_{1233} & E_{1212} & E_{1223} & E_{1213}\\ E_{2311} & E_{2322} & E_{2333} & E_{2312} & E_{2323} & E_{2313}\\ E_{1311} & E_{1322} & E_{1333} & E_{1312} & E_{1323} & E_{1313}\end{bmatrix}}_{\stackrel{\text{def}}{=}[\mathbb{E}]} \underbrace{\begin{Bmatrix}\epsilon_{11}\\ \epsilon_{22}\\ \epsilon_{33}\\ 2\epsilon_{12}\\ 2\epsilon_{23}\\ 2\epsilon_{31}\end{Bmatrix}}_{\stackrel{\text{def}}{=}\{\boldsymbol{\epsilon}\}}. \tag{22}$$

The existence of a scalar energy function forces $\mathbb{E}$ to be symmetric since the strains are symmetric; in other words, $W = \frac{1}{2}\boldsymbol{\epsilon} : \mathbb{E} : \boldsymbol{\epsilon} = \frac{1}{2}(\boldsymbol{\epsilon} : \mathbb{E} : \boldsymbol{\epsilon})^T = \frac{1}{2}\boldsymbol{\epsilon}^T : \mathbb{E}^T : \boldsymbol{\epsilon}^T = \frac{1}{2}\boldsymbol{\epsilon} : \mathbb{E}^T : \boldsymbol{\epsilon}$ which implies $\mathbb{E}^T = \mathbb{E}$. Consequently, $\mathbb{E}$ has only 21 independent components. The nonnegativity of W imposes the restriction that $\mathbb{E}$ remains positive definite. At this point, based on many factors that depend on the material microstructure, it can be shown that the components of $\mathbb{E}$ may be written in terms of anywhere between 21 and 2 independent parameters. Accordingly, for isotropic materials, we have two planes of symmetry and an infinite number of planes of directional independence (two free components), yielding

$$\mathbb{E} \stackrel{\text{def}}{=} \begin{bmatrix} \kappa+\frac{4}{3}\mu & \kappa-\frac{2}{3}\mu & \kappa-\frac{2}{3}\mu & 0 & 0 & 0\\ \kappa-\frac{2}{3}\mu & \kappa+\frac{4}{3}\mu & \kappa-\frac{2}{3}\mu & 0 & 0 & 0\\ \kappa-\frac{2}{3}\mu & \kappa-\frac{2}{3}\mu & \kappa+\frac{4}{3}\mu & 0 & 0 & 0\\ 0 & 0 & 0 & \mu & 0 & 0\\ 0 & 0 & 0 & 0 & \mu & 0\\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}. \tag{23}$$

In this case, we have

$$\mathbb{E} : \boldsymbol{\epsilon} = 3\kappa\frac{tr\boldsymbol{\epsilon}}{3}\mathbf{1} + 2\mu\boldsymbol{\epsilon}' \Rightarrow \boldsymbol{\epsilon} : \mathbb{E} : \boldsymbol{\epsilon} = 9\kappa\left(\frac{tr\boldsymbol{\epsilon}}{3}\right)^2 + 2\mu\boldsymbol{\epsilon}' : \boldsymbol{\epsilon}', \tag{24}$$

where $tr\boldsymbol{\epsilon} = \epsilon_{ii}$ and $\boldsymbol{\epsilon}' = \boldsymbol{\epsilon} - \frac{1}{3}(tr\boldsymbol{\epsilon})\mathbf{1}$ is the deviatoric strain. The eigenvalues of an isotropic elasticity tensor are $(3\kappa, 2\mu, 2\mu, \mu, \mu, \mu)$. Therefore, we must have $\kappa > 0$ and $\mu > 0$ to retain positive definiteness of $\mathbb{E}$. All of the material components of $\mathbb{E}$ may be spatially variable, as in the case of composite media.

Material Component Interpretation

There are a variety of ways to write isotropic constitutive laws, each time with a physically meaningful pair of material values.

Splitting the Strain

It is sometimes important to split infinitesimal strains into two physically meaningful parts:

$$\boldsymbol{\epsilon} = \frac{tr\boldsymbol{\epsilon}}{3}\mathbf{1} + \left(\boldsymbol{\epsilon} - \frac{tr\boldsymbol{\epsilon}}{3}\mathbf{1}\right). \tag{25}$$

An expansion of the Jacobian of the deformation gradient yields $J = det(\mathbf{1} + \nabla_X\mathbf{u}) \approx 1 + tr\nabla_X\mathbf{u} + \mathcal{O}(\nabla_X\mathbf{u}) = 1 + tr\boldsymbol{\epsilon} + \ldots$. Therefore, with infinitesimal strains, $(1 + tr\boldsymbol{\epsilon})d\omega_0 = d\omega$, and we can write $tr\boldsymbol{\epsilon} = \frac{d\omega - d\omega_0}{d\omega_0}$. Hence, $tr\boldsymbol{\epsilon}$ is associated with the *volumetric part of the deformation*. Furthermore, since $\boldsymbol{\epsilon}' \stackrel{\text{def}}{=} \boldsymbol{\epsilon} - \frac{tr\boldsymbol{\epsilon}}{3}\mathbf{1}$, the so-called strain deviator describes distortion in the material.

Infinitesimal Strain Material Laws

The stress $\boldsymbol{\sigma}$ can be split into two parts (dilatational and a deviatoric):

$$\boldsymbol{\sigma} = \frac{tr\boldsymbol{\sigma}}{3}\mathbf{1} + \left(\boldsymbol{\sigma} - \frac{tr\boldsymbol{\sigma}}{3}\mathbf{1}\right) \stackrel{\text{def}}{=} -p\mathbf{1} + \boldsymbol{\sigma}', \tag{26}$$

where we call the symbol p the hydrostatic pressure and $\boldsymbol{\sigma}'$ the stress deviator. With (24), we write

$$p = -3\kappa\left(\frac{tr\boldsymbol{\epsilon}}{3}\right) \qquad \text{and} \qquad \boldsymbol{\sigma}' = 2\mu\boldsymbol{\epsilon}'. \tag{27}$$

This is one form of Hooke's law. The resistance to change in the volume is measured by κ. We note that

$(\frac{tr\boldsymbol{\sigma}}{3}\mathbf{1})' = \mathbf{0}$, which indicates that this part of the stress produces no distortion.

Another fundamental form of Hooke's law is

$$\boldsymbol{\sigma} = \frac{E}{1+\nu}\left(\boldsymbol{\epsilon} + \frac{\nu}{1-2\nu}(tr\boldsymbol{\epsilon})\mathbf{1}\right), \tag{28}$$

and the inverse form

$$\boldsymbol{\epsilon} = \frac{1+\nu}{E}\boldsymbol{\sigma} - \frac{\nu}{E}(tr\boldsymbol{\sigma})\mathbf{1}. \tag{29}$$

To interpret the material values, consider an idealized uniaxial tension test (pulled in the x_1 direction inducing a uniform stress state) where $\sigma_{12} = \sigma_{13} = \sigma_{23} = 0$, which implies $\epsilon_{12} = \epsilon_{13} = \epsilon_{23} = 0$. Also, we have $\sigma_{22} = \sigma_{33} = 0$. Under these conditions, we have $\sigma_{11} = E\epsilon_{11}$ and $\epsilon_{22} = \epsilon_{33} = -\nu\epsilon_{11}$. Therefore, E, Young's modulus, is the ratio of the uniaxial stress to the corresponding strain component. The Poisson ratio, ν, is the ratio of the transverse strains to the uniaxial strain.

Another commonly used set of stress-strain forms is the Lamé relations:

$$\begin{aligned}\boldsymbol{\sigma} &= \lambda(tr\boldsymbol{\epsilon})\mathbf{1} + 2\mu\boldsymbol{\epsilon} \qquad \text{or}\\ \boldsymbol{\epsilon} &= -\frac{\lambda}{2\mu(3\lambda+2\mu)}(tr\boldsymbol{\sigma}\mathbf{1}) + \frac{\boldsymbol{\sigma}}{2\mu}.\end{aligned} \tag{30}$$

To interpret the material values, consider a homogeneous pressure test (uniform stress) where $\sigma_{12} = \sigma_{13} = \sigma_{23} = 0$, and where $\sigma_{11} = \sigma_{22} = \sigma_{33}$. Under these conditions, we have

$$\kappa = \lambda + \frac{2}{3}\mu = \frac{E}{3(1-2\nu)} \qquad \text{and} \qquad \mu = \frac{E}{2(1+\nu)}, \tag{31}$$

and consequently,

$$\frac{\kappa}{\mu} = \frac{2(1+\nu)}{3(1-2\nu)}. \tag{32}$$

We observe that $\frac{\kappa}{\mu} \to \infty$ implies $\nu \to \frac{1}{2}$ and $\frac{\kappa}{\mu} \to 0$ implies $\Rightarrow \nu \to -1$. Therefore, since both κ and μ must be positive and finite, this implies $-1 < \nu < 1/2$ and $0 < E < \infty$. For example, some polymeric foams exhibit $\nu < 0$, steels $\nu \approx 0.3$, and some forms of rubber have $\nu \to 1/2$. We note that λ *can be positive or negative*. For more details, see Malvern [3], Gurtin [2], and Chandrasekharaiah and Debnath [1].

References

1. Chandrasekharaiah, D.S., Debnath, L.: Continuum Mechanics. Academic, Boston (1994)
2. Gurtin, M.: An Introduction to Continuum Mechanics. Academic, New York (1981)
3. Malvern, L.: Introduction to the Mechanics of a Continuous Medium. Prentice Hall, Englewood Cliffs (1968)

Linear Programming

Robert J. Vanderbei
Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, USA

Mathematics Subject Classification

Primary 90C05; Secondary 49N15

Synonyms

Linear optimization (LP)

Short Definition

The *Linear Programming Problem* (LP) is the problem of maximizing or minimizing a linear function of one or more, and typically thousands of, variables subject to a similarly large number of equality and/or inequality constraints.

Description

Although Leonid Kantorovich [3] is generally credited with being the first to recognize the importance of linear programming as a tool for solving many practical operational problems, much credit goes to George Dantzig for independently coming to this realization a few years later (see [1, 2]). Originally, most applications arose out of military operations. However, it was quickly appreciated that important applications

appear in all areas of science, engineering, and business analytics.

A problem is said to be in *symmetric standard form* if all the constraints are inequalities and all of the variables are nonnegative:

$$\begin{aligned} &\text{maximize} && c^T x \\ &\text{subject to} && Ax \le b \\ & && x \ge 0. \end{aligned} \tag{1}$$

Here, A is an $m \times n$ matrix whose (i, j)-th element is $a_{i,j}$, b is an m-vector whose i-th element is b_i, and c is an n-vector whose j-th element is c_j. The linear function $c^T x$ is called the *objective function*. A particular choice of x is said to be *feasible* if it satisfies the constraints of the problem.

It is easy to convert any linear programming problem into an equivalent one in standard form. For example, any greater-than-or-equal-to constraint can be converted to a less-than-or-equal-to constraint by multiplying by minus one, any equality constraint can be replaced with a pair of inequality constraints, a minimization problem can be converted to maximization by negating the objective function, and every unconstrained variable can be replaced by a difference of two nonnegative variables.

Duality

Associated with every linear programming problem is a *dual problem*. The dual problem associated with (1) is

$$\begin{aligned} &\text{minimize} && b^T y \\ &\text{subject to} && A^T y \ge c \\ & && y \ge 0. \end{aligned} \tag{2}$$

Written in standard form, the dual problem is

$$\begin{aligned} -\text{maximize} \quad & -b^T y \\ \text{subject to} \quad & -A^T y \le -c \\ & y \ge 0 \quad . \end{aligned}$$

From this form we see that the dual of the dual is the primal. We also see that the dual problem is in some sense the *negative-transpose* of the primal problem.

The *weak duality theorem* states that, if x is feasible for the primal problem and y is feasible for the dual problem, then $c^T x \le b^T y$. The proof is trivial: $c^T x \le y^T A x \le y^T b$. The weak duality theorem is useful in that it provides a *certificate of optimality*: if x is feasible for the primal problem and y is feasible for the dual problem and $c^T x = b^T y$, then x is optimal for the primal problem and y is optimal for the dual problem.

There is also a *strong duality theorem*. It says that, if x is optimal for the primal problem, then there exists a y that is optimal for the dual problem and the two objective function values agree: $c^T x = b^T y$.

All algorithms for linear programming are based on simultaneously finding an optimal solution for both the primal and the dual problem (or showing that either that the primal problem is infeasible or unbounded). The value of the dual is that it proves that the primal solution is optimal.

Slack Variables and Complementarity

It is useful to introduce *slack variables* into the primal and dual problems so that all inequalities are simple nonnegativities:

Primal Problem:

$$\begin{aligned} &\text{maximize} && c^T x \\ &\text{subject to} && Ax + w = b \\ & && x, w \ge 0. \end{aligned}$$

Dual Problem:

$$\begin{aligned} &\text{minimize} && b^T y \\ &\text{subject to} && A^T y - z = c \\ & && y, z \ge 0. \end{aligned}$$

It is trivial to check that $(c + z)^T x = y^T A x = y^T (b - w)$. Hence, if x and w are feasible for the primal problem and y and z are feasible for the dual problem and $c^T x = b^T y$, then it follows that x is optimal for the primal, y is optimal for the dual and $z^T x + y^T w = 0$. Since all of the terms in these inner products are nonnegative, it follows that

$$z_j x_j = 0 \quad \text{for all } j \qquad \text{and} \qquad y_i w_i = 0 \quad \text{for all } i.$$

This condition is called *complementarity*.

Geometry

The feasible set is an n-dimensional *polytope* defined by the intersection of $n + m$ halfspaces where each halfspace is determined either by one of the m constraint inequalities, $Ax \le b$, or one of the n nonnegativity constraints on the variables, $x \ge 0$. Generally speaking, the vertices of this polytope

correspond to the intersection of n hyperplanes defined as the boundaries of a specific choice of n out of the $n + m$ halfspaces. Except in degenerate cases, the optimal solution to the LP occurs at one of the vertices.

Ignoring, momentarily, which side of a hyperplane is feasible and which is not, the $n + m$ hyperplanes generate up to $(n + m)!/n!m!$ possible vertices corresponding to the many ways that one can choose n hyperplanes from the $n + m$. Assuming that these points of intersection are disjoint one from the other, these points in n-space are called *basic solutions*. The intersections that lie on the feasible set itself are called *basic feasible solutions*.

Simplex Methods

Inspired by the geometric view of the problem, George Dantzig introduced a class of algorithms, called *simplex methods*, that start at the origin and repeatedly jump from one basic solution to an adjacent basic solution in a systematic manner such that eventually a basic feasible solution is found and then ultimately an optimal vertex is found.

With the slack variables defined, the problem has $n + m$ variables. As the slack variables w and the original variables x are treated the same by the simplex method, it is convenient to use a common notation:

$$x \longleftarrow \begin{bmatrix} x \\ w \end{bmatrix}.$$

A basic solution corresponds to choosing n of these variables to be set to zero. The m equations given by

$$Ax + w = b \tag{3}$$

can then be used to solve for the remaining m variables. Let $\mathcal{N}$ denote a particular choice of n of the $n + m$ indices and let $\mathcal{B}$ denote the complement of this set (so that $\mathcal{B} \cup \mathcal{N} = \{1, \ldots, n + m\}$). Let $x_{\mathcal{N}}$ denote the n-vector consisting of the variables x_j, $j \in \mathcal{N}$. These variables are called *nonbasic variables*. Let $x_{\mathcal{B}}$ denote the m-vector consisting of the rest of the variables. They are called *basic variables*. Initially, $x_{\mathcal{N}} = [x_1 \ \cdots \ x_n]^T$ and $x_{\mathcal{B}} = [x_{n+1} \ \cdots \ x_{n+m}]^T$ so that (3) can be rewritten as

$$x_{\mathcal{B}} = b - Ax_{\mathcal{N}}. \tag{4}$$

While doing jumps from one basic solution to another, this system of equations is rearranged so that the basic variables always remain on the left and the nonbasics appear on the right. Down the road, these equations become

$$x_{\mathcal{B}} = x^*_{\mathcal{B}} - B^{-1}Nx_{\mathcal{N}} \tag{5}$$

where B denotes the $m \times m$ invertible matrix consisting of the columns of the matrix $[A \ \ I]$ associated with the basic variables $\mathcal{B}$, N denotes those columns of that matrix associated with the nonbasic variables $\mathcal{N}$, and $x^*_{\mathcal{B}} = B^{-1}b$. Equation (5) is called a *primal dictionary* because it defines the primal basic variables in terms of the primal nonbasic variables. The process of updating equation (5) from one iteration to the next is called a *simplex pivot*.

Associated with each dictionary is a basic solution obtained by setting the nonbasic variables to zero and reading from the dictionary the values of the basic variables

$$x_{\mathcal{N}} = 0 \qquad \text{and} \qquad x_{\mathcal{B}} = x^*_{\mathcal{B}}.$$

In going from one iteration to the next, a single element of $\mathcal{N}$, say j^*, and a single element of $\mathcal{B}$, say i^*, are chosen and these two variables are swapped in these two sets. The variable x_{j^*} is called the *entering variable* and x_{i^*} is called the *leaving variable*.

In complete analogy with the primal problem, one can write down a *dual dictionary* and read off a dual basic solution. The initial primal/dual pair had a symmetry that we called the negative-transpose property. It turns out that this symmetry is preserved by the pivot operation. As a consequence, it follows that primal/dual complementarity holds in every primal/dual basic solution. Hence, a basic solution is optimal if and only if it is primal feasible and dual feasible.

Degeneracy and Cycling

Every variant of the simplex method chooses the entering and leaving variables at each iteration with the intention of improving some specific measure of a distance either from feasibility or optimality. If such a move does indeed make a strict improvement at every iteration, then it easily follows that the algorithm will find an optimal solution in a finite number of pivots because there are only a finite number of ways to partition the set $\{1, 2, \ldots, n + m\}$ into m basic and n nonbasic components. If the metric is always making

a strict improvement, then it can never return to a place it has been before. However, it can happen that a simplex pivot can make zero improvement in one or more iterations. Such pivots are called *degenerate pivots*. It is possible, although exceedingly rare, for simple variants of the simplex method to produce a sequence of degenerate pivots eventually returning to a basic solution already visited. If the algorithm chooses the entering and leaving variables according to a deterministic rule, then returning once implies returning infinitely often and the algorithm fails. This failure is called *cycling*. There are many safe-guards to prevent cycling, perhaps the simplest being to add a certain random aspect to the entering/leaving variable selection rules. All modern implementations of the simplex method have anti-cycling safeguards.

Empirical Average-Case Performance

Given the anti-cycling safeguards, it follows that the simplex method is a finite algorithm. But, how fast is it in practice? The answer is that, on average, most variants of the simplex method take roughly order $\min(n, m)$ pivots to find an optimal solution. Such average case performance is about the best that one could hope for and accounts for much of the practical usefulness of linear programming in solving important everyday problems.

Worst-Case Performance

One popular variant of the simplex method assumes that the initial primal dictionary is feasible and, at each iteration, selects for the entering variable the non-basic variable that provides the greatest rate of increase of the objective function and it then chooses the leaving variable so as to preserve primal feasibility. In 1972, Klee and Minty [6] constructed a simple family of LPs in which the n-th instance involved n variables and a feasible polytope that is topologically equivalent to an n-cube but for which the pivot rule described above takes short steps in directions of high rate of increase rather than huge steps in directions with a low rate of increase and in so doing visits all 2^n vertices of this distorted n-cube in 2^{n-1} pivots thus showing that this particular variant of the simplex method has *exponential complexity*. It is an open question whether or not there exists some variant of the simplex method whose worst-case performance is better than exponential.

Interior-Point Methods

For years it was unknown whether or not there existed an algorithm for linear programming that is guaranteed to solve problems in polynomial time. In 1979, Leonid Khachiyan [5] discovered the first such algorithm. But, in practice, his algorithm was much slower than the simplex method. In 1984, Narendra Karmarkar [4] developed a completely different polynomial time algorithm. It turns out that his algorithm and the many variants of it that have appeared over time are also highly competitive with the simplex method.

The class of algorithms inspired by Karmarkar's algorithm are called *interior-point algorithms*. Most implementations of algorithms of this type belong to a generalization of this class called *infeasible interior-point algorithms*. These algorithms are iterative algorithms that approach optimality only in the limit – that is, they are not finite algorithms. But, for any $\epsilon > 0$, they get within ϵ of optimality in polynomial time. The adjective "infeasible" points to the fact that these algorithms may, and often do, approach optimality from outside the feasible set. The adjective "interior" means that even though the iterates may be infeasible, it is required that all components of all primal and dual variables be strictly positive at every iteration.

Complexity

In the worst case, Karmarkar's algorithm requires on the order of $\sqrt{n}\log(1/\epsilon)$ iterations to get within ϵ of an optimal solution. But, an iteration of an interior-point method is more computationally intensive (order n^3) than an iteration of the simplex method (order n^2). Comparing arithmetic operations, one gets that interior-point methods require on the order of $n^{3.5}\log(1/\epsilon)$ arithmetic operations in the worst case, which is comparable to the average case performance of the simplex method.

References

1. Dantzig, G.: Programming in a linear structure. Econometrica **17**, 73–74 (1949)
2. Dantzig, G.: Maximization of a linear function of variables subject to linear inequalities. In: Koopmans, T. (ed.) Activity Analysis of Production and Allocation, pp. 339–347. Wiley, New York (1951)
3. Kantorovich, L.: A new method of solving some classes of extremal problems. Dokl. Akad. Sci. USSR **28**, 211–214 (1940)

4. Karmarkar, N.: A new polynomial time algorithm for linear programming. Combinatorica **4**, 373–395 (1984)
5. Khachian, L.: A polynomial algorithm in linear programming. Dokl. Acad. Nauk SSSR **244**, 191–194 (1979), in Russian. English Translation: Sov. Math. Dokl. **20**, 191–194
6. Klee, V., Minty, G.: How good is the simplex algorithm? In: Shisha, O. (ed.) Inequalities–III, pp. 159–175. Academic, New York (1972)

Linear Sampling

Michele Piana
Dipartimento di Matematica, Università di Genova, CNR – SPIN, Genova, Italy

Mathematics Subject Classification

34L25; 15A29

Synonyms

Linear sampling method; LSM

Glossary/Definition Terms

Direct scattering problem Problem of determining the total acoustic or electromagnetic field from the knowledge of the geometrical and physical properties of the scatterer.

Inverse scattering problem Problem of recovering the geometrical and physical properties of an inhomogeneity for the knowledge of the acoustic or electromagnetic scattered field.

Ill-posed problem In the sense of Hadamard, it is a problem whose solution does not exist unique or does not depend continuously on the data.

Far-field pattern In the asymptotic factorization of the far-field pattern, it is the term depending just on the observation angle.

Far-field operator Linear intergral operator whose intergral kernel is the far-field pattern.

Hankel function Complex function which is a linear combination of Bessel functions.

Far-field equation Linear integral equation relating the far-field operator with the far-field pattern of the field generated by a point source.

Wavenumber Real positive number given by the ratio between 2π and the wavelength of the incident wave.

Refractive index Complex-valued function where the real part is proportional to the electrical permittivity and the imaginary part is proportional to the electrical conductivity.

Herglotz wave function Wave function which is a weighted linear superposition of plane waves.

Tikhonov regularization Method for the solution of linear ill-posed problems based on the minimization of a convex functional with L^2 penalty term.

L^2 Hilbert space Linear space made of functions with bounded L^2 norm.

Maxwell equations The set of four equations describing classical electrodynamics.

Lippmann-Schwinger equation Integral equation at the basis of both classical and quantum scattering.

Poynting vector Vector field provided by the outer product between the electric and magnetic fields.

Short Definition

The linear sampling method (LSM) is a linear visualization method for solving nonlinear inverse scattering problems.

Description

Inverse Scattering Methods

Electromagnetic or acoustic scattering is a physical phenomenon whereby, in the presence of an inhomogeneity, an electromagnetic or acoustic incident wave is scattered and the total field at any point of the space is written as the sum of the original incident field and the scattered field. The direct scattering problem is the problem of determining this total field starting from the knowledge of the geometrical and physical properties of the scatterer. On the contrary, the inverse scattering problem is the problem of recovering information on the inhomogeneity from the knowledge of the scattered field. Solving inverse scattering problems is particularly challenging for two reasons. First, all inverse scattering problems significant in applications belong

to the class of the so-called ill-posed problems in the sense of Hadamard [1], and therefore, any reliable approach to their solution must face at some stage issues of uniqueness and numerical stability. Second, inverse scattering problems are often nonlinear, and there are physical conditions notably significant in the applied sciences where such nonlinearity is genuine and cannot be linearized by means of weak-scattering approximations.

Most computational approaches for the solution of inverse scattering problems can be divided into three families: (1) nonlinear optimization schemes, where the restoration is performed iteratively from an initial guess of the position and shape of the scatterer; (2) weak-scattering approximation methods, where a linear inverse problem is obtained by means of low- or high-frequency approximations; and (3) qualitative methods, which provide visualization of the inhomogeneity but are not able to reconstruct the point values of the scattering parameters. The linear sampling method (LSM) [2–4] is, historically, the first qualitative method, the most theoretically investigated, and the most experimentally tested. In this approach, a linear integral equation of the first kind is written for each point of a computational grid containing the scatterer, the integral kernel of such equation being the far-field pattern of the scattered field, and the right-hand side being an exactly known analytical function. This integral equation is approximately solved for each sampling point by means of a regularization method [5], and the object profile is recovered by exploiting the fact that the norm of this regularized solution blows up when the sampling point approaches the boundary from inside.

The main advantages of the linear sampling method are that it is fast, simple to implement, and not particularly demanding from a computational viewpoint. The method of course has also some disadvantages. The main one is that it only provides a visualization of the support of the scatterer and it is not possible to infer information about the point values of the refractive index.

Formulation of the Linear Sampling Method

As a test case, consider the two-dimensional scattering problem [4, 6] of determining $u = u(\cdot\,;\theta) \in C^2(\mathbb{R}^2 \setminus \partial D) \cap C^1(\mathbb{R}^2)$ such that

$$\begin{cases} \Delta u(x) + k^2\, n(x)\, u(x) = 0 & \text{for } x \in \mathbb{R}^2 \setminus \partial D \\ u(x) = e^{\mathrm{i}kx\cdot\hat{d}} + u^s(x) & \text{for } x \in \mathbb{R}^2 \\ \lim\limits_{r\to\infty}\left[\sqrt{r}\left(\dfrac{\partial u^s}{\partial r} - \mathrm{i}k u^s\right)\right] = 0, \end{cases} \tag{1}$$

where $D \subset \mathbb{R}^2$ is a C^2-domain, ∂D is its boundary, $\hat{d} = \hat{d}(\theta) = (\cos\theta, \sin\theta)$ is the incidence direction, and k is the wavenumber; $n(x)$ is the refractive index

$$n(x) := \frac{1}{\varepsilon_B}\left[\varepsilon(x) + \mathrm{i}\,\frac{\sigma(x)}{\omega}\right] \qquad \forall x \in \mathbb{R}^2, \tag{2}$$

where $\mathrm{i} = \sqrt{-1}$ and ω denote the angular frequency of the wave and $\varepsilon(x)$ and $\sigma(x)$ are the electrical permittivity and conductivity, respectively. We assume that $\varepsilon(x)$ is uniform in $\mathbb{R}^2 \setminus \bar{D}$ and equal to the background value $\varepsilon_B > 0$, while $\sigma = 0$ in the same region.

For each incidence direction $\hat{d}$, there exists a unique solution to problem (1) [6], and the corresponding scattered field $u^s = u^s(\cdot\,;\theta)$ has the following asymptotic behavior (holding uniformly in all directions $\hat{x} := x/|x|$):

$$u^s(x;\theta) = \frac{e^{\mathrm{i}kr}}{\sqrt{r}}\, u_\infty(\varphi;\theta) + O\left(r^{-3/2}\right) \quad \text{as } r = |x| \to \infty, \tag{3}$$

where (r, φ) are the polar coordinates of the observation point x and the function $u_\infty = u_\infty(\cdot\,;\theta) \in L^2[0, 2\pi]$ is known as the *far-field pattern* of the scattered field u^s.

Define the linear and compact *far-field operator* $F : L^2[0, 2\pi] \to L^2[0, 2\pi]$ corresponding to the inhomogeneous scattering problem (1) as

$$(Fg)(\varphi) := \int_0^{2\pi} u_\infty(\varphi, \theta) g(\theta)\, d\theta \quad \forall g \in L^2[0, 2\pi]. \tag{4}$$

The operator F is injective with dense range if k^2 is not a transmission eigenvalue [7].

Next consider the outgoing scalar field

$$\Phi(x, z) = \frac{\mathrm{i}}{4} H_0^{(1)}(k|x - z|) \qquad \forall x \neq z, \tag{5}$$

generated by a point source located at $z \in \mathbb{R}^2$, where $H_0^{(1)}(\cdot)$ denotes the Hankel function of the first kind and of order zero. The corresponding far-field pattern is given by

$$\Phi_\infty(\varphi, z) = \frac{e^{i\pi/4}}{\sqrt{8\pi k}} e^{-ik\hat{x}(\varphi)\cdot z},$$
$$\hat{x}(\varphi) := (\cos\varphi, \sin\varphi) \quad \forall \varphi \in [0, 2\pi]. \tag{6}$$

For each $z \in \mathbb{R}^2$, the *far-field equation* is defined as

$$(Fg_z)(\varphi) = \Phi_\infty(\varphi, z). \tag{7}$$

The linear sampling method is inspired by a *general theorem* [7], concerning the existence of ϵ-approximate solutions to the far-field equation and their qualitative behavior. According to this theorem, if $z \in D$, then for every $\epsilon > 0$, there exists a solution $g_z^\epsilon \in L^2[0, 2\pi]$ of the inequality

$$\left\| Fg_z^\epsilon - \Phi_\infty(\cdot, z) \right\|_{L^2[0,2\pi]} \leq \epsilon \tag{8}$$

such that for every $z^* \in \partial D$,

$$\lim_{z \to z^*} \left\| g_z^\epsilon \right\|_{L^2[0,2\pi]} = \infty \quad \text{and} \quad \lim_{z \to z^*} \left\| v_{g_z^\epsilon} \right\|_{L^2(D)} = \infty, \tag{9}$$

where $v_{g_z^\epsilon}$ is the Herglotz wave function with kernel g_z^ϵ. If $z \notin D$, the approximate solution remains unbounded.

On the basis of this theorem, the algorithm of the linear sampling method may be described as follows [3]. Consider a sampling grid that covers a region containing the scatterer. For each point z of the grid, compute a regularized solution $g_{\alpha^*(z)}$ of the (discretized) far-field equation (7) by applying Tikhonov regularization coupled with the generalized discrepancy principle [5]. The boundary of the scatterer is visualized as the set of grid points in which the (discretized) L^2-norm of $g_{\alpha^*(z)}$ becomes mostly large.

Computational Issues

The main drawback of this first formulation of the LSM is that the regularization algorithm for the solution of the far-field equation is applied point-wise, i.e., a different regularization parameter must be chosen for each sampling point z. A much more effective implementation is possible by formulating the method in a functional framework which is the direct sum of many L^2 spaces. The first step of this formulation is to observe that, in real experiments, the far-field pattern is measured for P observation angles $\{\varphi_i\}_{i=0}^{P-1}$ and Q incidence angles $\{\theta_j\}_{j=0}^{Q-1}$, i.e., for observation directions $\{\hat{x}_i = (\cos\varphi_i, \sin\varphi_i)\}_{i=0}^{P-1}$ and incidence directions $\{d_j = (\cos\theta_j, \sin\theta_j)\}_{j=0}^{Q-1}$. In the following, $P = Q = N$ and $\varphi_i = \theta_i \quad i = 0, N-1$. These values are placed into the *far-field matrix* $\mathbf{F}$, whose elements are defined as

$$\mathbf{F}_{ij} := u_\infty(\hat{x}_i, d_j). \tag{10}$$

In practical applications, the far-field matrix is affected by the measurement noise, and therefore, only a noisy version $\mathbf{F}_h$ of the far-field matrix is at disposal, such that

$$\mathbf{F}_h = \mathbf{F} + \mathbf{H}, \tag{11}$$

where $\mathbf{H}$ is the noise matrix with $\|\mathbf{H}\| \leq h$. Furthermore, for each $z = r(\cos\psi, \sin\psi) \in \mathcal{Z}$ containing the scatterer,

$$\boldsymbol{\Phi}_\infty(z) := \frac{e^{i\frac{\pi}{4}}}{\sqrt{8\pi k}} \left[e^{-ikr\cos(\varphi_0 - \psi)}, \ldots, e^{-ikr\cos(\varphi_{N-1} - \psi)} \right]^{\mathrm{T}}. \tag{12}$$

Therefore, the one-parameter family of linear integral equations (7) can be replaced by the one-parameter family of ill-conditioned square linear systems

$$\mathbf{F}_h \mathbf{g}(z) = \frac{N}{2\pi} \boldsymbol{\Phi}_\infty(z). \tag{13}$$

Then consider the direct sum of Hilbert spaces:

$$\left[L^2(\mathcal{Z})\right]^N := \underbrace{L^2(\mathcal{Z}) \oplus \cdots \oplus L^2(\mathcal{Z})}_{N \text{ times}}, \tag{14}$$

and define the linear operator $\mathsf{F}_h : \left[L^2(\mathcal{Z})\right]^N \to \left[L^2(\mathcal{Z})\right]^N$ such that

$$\left[\mathsf{F}_h \mathbf{g}(\cdot)\right](\cdot) := \left\{ \sum_{j=0}^{N-1} (\mathbf{F}_h)_{ij}\, g_j(\cdot) \right\}_{i=0}^{N-1}$$
$$\forall \mathbf{g}(\cdot) \in \left[L^2(\mathcal{Z})\right]^N, \tag{15}$$

where the $(\mathbf{F}_h)_{ij}$ are the elements of the noisy far-field matrix. This allows one to express the infinitely many algebraic systems (13) as the single functional equation in $[L^2(\mathcal{Z})]^N$

$$[\mathsf{F}_h \mathbf{g}(\cdot)](\cdot) = \frac{N}{2\pi} \boldsymbol{\Phi}_\infty(\cdot), \tag{16}$$

where $\boldsymbol{\Phi}_\infty(\cdot)$ is the element in $[L^2(\mathcal{Z})]^N$ trivially obtained from $\boldsymbol{\Phi}_\infty(z)$ simply regarding z as a variable on $\mathcal{Z}$ instead of a fixed point in $\mathbb{R}^2$. The regularization of this equation occurs in a way which is independent of z and therefore provides a single value of the regularization parameter (explicitly, the regularized solution of this equation can be computed by means of the singular system of the far-field matrix). With this no-sampling implementation of the LSM [8] and by means of a conventional personal computer, two-dimensional scatterers can be visualized in few seconds and complicated three-dimensional objects in a few minutes.

Physical Interpretation

The far-field equation at the basis of the LSM is not an equation of mathematical physics, in the sense that it cannot be derived as a consequence of general physical principles (as it happens, e.g., in the case of Maxwell equations or of the Lippmann-Schwinger equation). However, energy conservation can be utilized to explain the link between the approximate solution of the far-field equation described in the general theorem and the regularized solutions introduced in the LSM. In a local framework, $Fg_z^\epsilon - \Phi_\infty(\cdot, z)$ is the far-field pattern of the radiating field defined as

$$w_z^\epsilon(x) := \int_0^{2\pi} u^s(x,\theta) g_z^\epsilon(\theta) d\theta - \Phi(x,z) \quad \forall x \in \mathbb{R}^2 \setminus D. \tag{17}$$

The (time-averaged) Poynting vector field associated to this field and its flow lines are then considered. It is easy to show that if these flow lines go regularly from a neighborhood of the sampling point z up to infinity, then $\left\| g_z^\epsilon \right\|_{L^2[0,2\pi]}$ blows up when z approaches the boundary of the scatterer from inside and is unbounded when z is outside [9]. This holds, in particular, for Tikhonov-regularized solutions $g_{\alpha^*(z)}$ of the far-field equation, provided that the regularization parameter $\alpha^*(z)$ is chosen, as is always possible, in such a way that $\left\| Fg_{\alpha^*(z)} - \Phi_\infty(\cdot, z) \right\|_{L^2[0,2\pi]} \leq \epsilon$, for a nonvanishing (but small enough) ϵ. It must be pointed out that this interpretation is based on an a posteriori analysis: the performances of the LSM are related to the behavior of the flow lines of the Poynting vector, but such behavior is numerically observed and not theoretically predicted. To provide a rigorous mathematical justification of the LSM, it would be necessary to deduce the geometric properties of these flow lines a priori, i.e., starting from the knowledge of the scattering conditions.

Conclusions

The LSM represents an effective approach to inverse scattering problems. It provides fast visualizations of the scatterer's profile by requiring the solution of a functional equation (in its no-sampling implementation), and it does not need accurate initializations to work properly. Its main applications are concerned with nondestructive testing and medical imaging, in the case of nonlinear prototypal diagnostic procedures like microwave tomography. The intrinsic drawback of the LSM is the fact that it cannot recover point values of the physical parameters describing the scatterer. This limitation can be overcome by integrating the LSM with iterative schemes that are able to pointwise reconstruct these parameters (e.g., the electrical conductivity and permittivity in the case of electromagnetic scattering) and that, in order to work, need to be initialized by means of some approximate guess of the shape and dimension of the scatterer. In this hybrid approach [10], the linear sampling method can be utilized to obtain such initialization in a computationally effective way, and quantitative reconstructions are provided by the iterative inverse scattering scheme.

References

1. Hadamard, J.: Lectures on Cauchy's Problem in Linear Partial Differential Equations. Dover, New York (1923)
2. Colton, D., Kirsch, A.: A simple method for solving inverse scattering problems in the resonance region. Inverse Probl. **12**, 383–393 (1996)
3. Colton, D., Piana, M., Potthast, R.: A simple method using Morozov's discrepancy principle for solving inverse scattering problems. Inverse Probl. **13**, 1477–1493 (1997)
4. Colton, D., Haddar, H., Piana, M.: The linear sampling method in inverse electromagnetic scattering theory. Inverse Probl. **19**, S105–S137 (2003)
5. Tikhonov, A.N., Goncharski, A.V., Stepanov, V.V., Yagola, A.G.: Numerical Methods for the Solution of Ill-Posed Problems. Kluwer, Dordrecht (1995)
6. Colton, D., Kress, R.: Inverse Acoustic and Electromagnetic Scattering Theory. Springer, Berlin (1998)
7. Cakoni, F., Colton, D.: Qualitative Methods in Inverse Scattering Theory. Springer, Berlin (2006)

8. Aramini, R., Brignone, M., Piana, M.: The linear sampling method without sampling. Inverse Probl. **22**, 2237–2254 (2006)
9. Aramini, R., Caviglia, G., Massa, A., Piana, M.: The linear sampling method and energy conservation. Inverse Probl. **26**, 055004 (2010)
10. Brignone, M., Bozza, G., Randazzo, A., Piana, M., Pastorino, M.: A hybrid approach to 3d microwave imaging by using linear sampling and ant colony optimization. IEEE Trans. Ant. Prop. **56**, 3224–3232 (2008)

Linear Scaling Methods

Carlos J. García-Cervera
Mathematics Department, University of California, Santa Barbara, CA, USA

Definition

By linear scaling methods we understand numerical methodologies that provide an approximation to the solution of a given problem within a prescribed accuracy with computational cost that scales linearly with the number of degrees of freedom or variables in the system. Linear scaling methods play a significant role in large-scale scientific computing. However, it is often the case that even linear scaling algorithms are not computationally feasible for such large-scale problems, and sublinear scaling methods are required.

Linear scaling methods have a long history in numerical analysis, and the focus of this entry will be on linear scaling methods as they apply to computational chemistry and molecular modeling. We begin with a description of some of the linear scaling methodologies developed in the context of Kohn-Sham density functional theory (DFT). These algorithms focus on the computation of electronic structures. These provide the electronic density that can be used to obtain the interatomic forces via the Hellmann-Feynman theorem. The efficient evaluation of these forces requires fast summation techniques for particle interactions. More general linear and sublinear scaling methodologies that have been developed for multiscale modeling will be discussed as well.

Linear Scaling Methods in Kohn-Sham DFT

In Kohn-Sham DFT, the energy of a system on N_a atoms, with nuclei located at $\mathbf{R}_j$, $j = 1, \ldots, N_a$, and atomic charge Z_j, is written as [1]

$$E_{KS}[\rho; \mathbf{R}] = \frac{1}{2} \sum_{i=1}^{N} \int_{\mathbb{R}^3} |\nabla \psi_i|^2 \, d\mathbf{x} + F_H[\rho] + F_{XC}[\rho] + \int_{\Omega} V(\mathbf{x}) \rho(\mathbf{x}) \, d\mathbf{x} + V_{nn}. \tag{1}$$

The first term in (1) is the kinetic energy, and the other contributions to the energy are Hartree, exchange and correlation, external potential energies, and interionic interactions, respectively.

The Hartree energy describes the Coulombic interactions between electrons:

$$F_H[\rho] = \frac{1}{2} \int_{\Omega} \int_{\Omega} \frac{\rho(\mathbf{x}) \rho(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|} \, d\mathbf{x} \, d\mathbf{y}. \tag{2}$$

The exchange and correlation energy, $F_{XC}[\rho]$, introduces corrections to the energy that derive from using the noninteracting electron approximation for the kinetic and Hartree energies. Although the expression for the total energy in (1) is exact, $F_{XC}[\rho]$ remains unknown. A number of approximations have been developed [2], but for illustration purposes, we will adopt here the local density approximation (LDA) [1]: $F_{XC}[\rho] = \int \rho \varepsilon(\rho)$.

The last two terms in energy (1) are the effect of the external potential and the interatomic energy, respectively. In principle,

$$V(\mathbf{x}) = -\sum_{j=1}^{N_a} \frac{Z_j}{|\mathbf{x} - \mathbf{R}_j|}, \tag{3}$$

and

$$V_{nn} = \frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^{N_a} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|}. \tag{4}$$

However, a further reduction can be achieved by making use of pseudopotentials [3–6]: The core electrons and the nuclei are treated as a unit which interacts with the valence electrons through the pseudopotential $v(\mathbf{x})$.

In what follows ρ will be considered to be the density of the valence electrons only.

Minimizing the energy (1) under the orthogonality constraint for the orbitals leads to the Kohn-Sham equations, the system of nonlinear eigenvalue problems

$$\left(-\frac{1}{2}\Delta + V_{\text{eff}}[\rho]\mathbf{I}\right)\psi_i = \sum_{j=1}^{N}\lambda_{ij}\psi_j,$$

$$i = 1, 2, \ldots, N; \quad \rho = \sum_{i=1}^{N}|\psi_i|^2, \tag{5}$$

where $\mathbf{I}$ is the identity operator, V_{eff} is the variational derivative of the energy with respect to the density,

$$V_{\text{eff}}[\rho] = V(\mathbf{x}) + \int \frac{\rho(\mathbf{y})}{|\mathbf{x}-\mathbf{y}|}\,d\mathbf{y} + \varepsilon(\rho) + \rho\varepsilon'(\rho), \tag{6}$$

and λ_{ij} are Lagrange multipliers associated to the orthogonality constraints.

The traditional self-consistent approach [1] for the solution of this eigenvalue problem consists of two nested iterations: In the inner iteration, the orbitals $\{\psi_j\}_{j=1}^N$ are obtained by a process of diagonalization and orthogonalization; in the outer iteration, the electron density is updated until self-consistency is reached. The diagonalization and/or orthogonalization procedure scales typically as $O(N^3)$, which is prohibitively expensive for relatively small problems.

A number of new methodologies have been proposed for the solution of (6), which attempt to exploit the locality of the problem in order to reduce the computational complexity [7]. Locality, in quantum mechanics, refers to the property that a small disturbance in a molecule only has a local effect in the electron density, a phenomenon coined by W. Kohn as *nearsightedness* [8].

Localization

The localization properties of quantum systems are discussed in the entry ▸ Solid State Physics, Berry Phases and Related Issues, where representations in terms of Bloch and Wannier functions are described. Due to its localization properties, Wannier functions have often been used in the development of linear scaling methods for Kohn-Sham DFT.

One of the first implementations of Wannier functions in DFT codes was carried out by Marzari and Vanderbilt, who defined what are known as *maximally localized wannier functions* (MLWF) [9]. Given a family of Bloch functions $\{\psi_{n,\mathbf{k}}\}$ for $1 \le n \le N$, let $V_{\mathbf{k}} = \text{span}\,\{u_{n,\mathbf{k}}\}$, where $\mathbf{k} \in BZ$, the first Brillouin zone. For each space $V_{\mathbf{k}}$, we can construct another orthonormal basis via an orthonormal transformation $U^{\mathbf{k}}$. Given this family of bases of $V_{\mathbf{k}}$, we can construct corresponding family of Wannier functions. Marzari and Vanderbilt constructed an optimal set of Wannier functions by minimizing the spread of the Wannier functions associated to each family of orthonormal transformation $\{U_{\mathbf{k}}\}_{\mathbf{k}\in BZ}$, among all possible such transformations:

$$\{U_{\mathbf{k}}^*\} = \arg\min_U \sum_{n=1}^{N} < |x|^2 >_{n,U} - |< x >_{n,U}|^2. \tag{7}$$

This concept was generalized to the non-orthogonal case in [10]: Given a linear space $V = \text{span}\,\{\psi_j\}_{j=1}^N$ of dimension N, and a given smooth weight function $w \ge 0$, the optimally localized non-orthogonal wave function $\widetilde{\psi}$ is defined as

$$\widetilde{\psi} = \arg\min_{\phi\in V,\ \|\phi\|=1} \int_{\mathbb{R}^3} w(\mathbf{x})|\phi(\mathbf{x})|^2\,d\mathbf{x}, \tag{8}$$

where $w(\mathbf{x}) = |\mathbf{x}-\mathbf{x}_c|^{2p}$ and p is a positive integer (the maximally localized wannier function corresponds to the choice $p = 1$). In the context of the MLWFs, this would be equivalent to considering not only orthonormal transformations, but any automorphism of V. As a consequence, the admissible space is larger and therefore the non-orthogonal wave functions have better localization properties than orthogonal Wannier functions.

Linear Scaling Methods for Kohn-Sham DFT

The main approaches for Kohn-Sham DFT that have been proposed for linear scaling computations can be divided into the following categories:

1. Density matrix-based methods:
   (a) Fermi operator expansion
   (b) Density-matrix minimization
   (c) Optimal basis density-matrix minimization
2. Domain decomposition: divide and conquer

L

3. Localized orbital minimization
4. Localized subspace iteration

A description of some of these methodologies can be found in the entries ▸ Fast Methods for Large Eigenvalues Problems for Chemistry and ▸ Large-Scale Electronic Structure and Nanoscience Calculations. Further details about these methods can also be found in the recent book by Richard Martin [11] and the entry by Jean-Luc Fattebert. We will focus here on the localized subspace iteration.

Localized Subspace Iteration

The Kohn-Sham functional in the non-orthogonal formulation is invariant under automorphisms of the space spanned by the wave functions and the entry by Jean-Luc Fattebert. The advantage of this viewpoint is that the specific representation of the subspace is not relevant, and therefore one can choose a representation that is convenient. Linear scaling can be achieved by choosing a representation in terms of optimally localized non-orthogonal wave functions, as described in [12]. The algorithm is similar to the subspace iteration method of Zhou, Saad, Tiago, and Chelikowsky [13], but by avoiding diagonalization and orthogonalization, linear scaling is achieved.

To find the minimizing subspace, an initial subspace of dimension N is given and this space is successively improved by filtering out the components corresponding to the unoccupied states, that is eigenvalues above the Fermi energy. An efficient filter can be constructed using Chebyshev polynomials. After the filtering step, the locality of the representation needs to be reestablished and this is achieved with the algorithm presented in [10] and described earlier in the section entitled Localization.

An important component of the algorithm is the computation of the density, which involves the computation of $\mathbf{S}^{-1}$. A number of approaches that exploit the decay properties of the off-diagonal components of $\mathbf{S}$ and $\mathbf{S}^{-1}$ have appeared in the literature [14, 15].

Fast Summations Algorithms

In ab-initio molecular dynamics, interatomic forces are computed using Hellmann-Feynman's formula [16, 17] (see also the entry ▸ Large-Scale Computing for Molecular Dynamics Simulation).

To illustrate some of the fast summation techniques developed for evaluating interatomic interactions, consider a system of N particles at locations $\{\mathbf{R}_j\}_{j=1}^N$, with charges $\{Z_j\}_{j=1}^N$, interacting with each other via a potential of the form

$$\Phi(\mathbf{R}_j) = \sum_{\substack{i=1 \\ i\neq j}}^{N} \frac{Z_i}{|\mathbf{R}_i - \mathbf{R}_j|}. \tag{9}$$

Forces can be evaluated as

$$-\nabla\Phi(\mathbf{R}_j) = \sum_{\substack{i=1 \\ i\neq j}}^{N} Z_i \frac{\mathbf{R}_i - \mathbf{R}_j}{|\mathbf{R}_i - \mathbf{R}_j|^3}. \tag{10}$$

A direct computation of the summation for each particle scales as $O(N^2)$ and is therefore too costly for large-scale simulations. One of the first ideas for fast computations of summations of the form (9) was the treecode, introduced by Barnes and Hut [18]. The basic idea of the algorithm is to consider clusters of particles at different levels of spatial refinement, or scales, and to compute the interaction between clusters that are well separated by using an expansion in terms of multipoles. Interaction with particles which are nearby is computed by direct summation. By using a hierarchical decomposition of clusters, the algorithm achieves $O(N \log_2 N)$ complexity.

An algorithm with linear scaling, the *fast multipole method* (FMM), was introduced by Greengard and Rokhlin [19]. The algorithm consists of an *upward pass* and a *downward pass*. In the upward pass, multipole expansions are constructed at the finest level, and the multipole expansions are coarser levels at constructed by merging expansions from the next finer level. In the downward pass, the multipole expansions are converted into local expansions about the centers of each box, starting from the coarsest level. These expansions are used to construct the local expansions at increasingly finer levels. At the finest level, the expansions contain the contributions of all the sources that are well separated from the corresponding box and are evaluated at each target. Finally, the contributions from nearest neighbors are evaluated by direct summation.

From an algebraic point of view, there have been some generalizations of this algorithm that exploit the fact that interactions between clusters that are well

separated can be approximated well by low-rank matrices [20–22].

Linear Scaling in Multiscale Modeling

Linear scaling algorithms are of particular importance in atomistic computations, due to the large number of degrees of freedom involved. Even though these problems are formulated at the atomistic scale, we are typically interested in phenomena that occur at much larger scales. A number of algorithms and methodologies have been developed for specific multiscale problems in which one takes advantage of how the different scales interact with each other [23, 24].

One of the first attempts to develop a general methodology for multiscale problems was carried out by Achi Brandt as a generalization of the multigrid idea (see [25] for a review).

The multigrid method was originally developed as an efficient way to solve the algebraic equations resulting from the discretization of partial differential equations (PDEs) [26, 27]. The main ingredients of the multigrid method are:

1. A *restriction* operator that transfers information from a fine grid to a coarse grid
2. A *relaxation* or *smoothing* scheme at each level that improves the current approximation to the solution
3. An *interpolation* operator that transfers information from a coarse grid to a fine grid

The speed of convergence of the multigrid method depends on the interplay between the relaxation and interpolation operators and relies on the ability of the interpolation procedure to approximate the corresponding approximation after relaxation. It has been shown in a number of cases that the algorithm achieves linear scaling [28].

The generalization of the multigrid method to multiscale problems introduced by Achi Brandt proceeds by constructing a description of the problem at different physical scales. As the original multigrid, it consists of an equilibration scheme on each scale and interscale operators that transfer information from fine to coarse scales and from coarse to fine scales. By doing this, large-scale changes in the system can be effectively computed using a coarse grid, and the information gathered from the coarse scales provide large-scale corrections for the solutions on finer scales. The goal of these algorithms is to produce a macroscopic numerical description of the system in situations where a closed-form differential equation is not available or even appropriate. The computational cost of these procedures depends on the ability to express the equations at the coarser levels in terms of the coarse variables and not in terms of finer-level variables. To achieve this, Brandt combined the ideas of multigrid with renormalization techniques in order to efficiently obtain a description of the system on coarser levels. Applications to fluid dynamics, optimal control, Monte Carlo, and image processing among others were also discussed in [25].

For crystalline solids, Chen and Ming developed an efficient multigrid strategy for molecular mechanics at zero temperature that does not require the use of renormalization techniques [29]. The main idea in their approach is to use a Cauchy-Born (CB) elasticity model [30] as a coarse grid operator. This is used within a cascadic multigrid method to provide an elastically deformed state at every grid level that can be used as an initial guess for the molecular mechanics model. To illustrate the approach in [29], consider a nested sequence of triangulations $\mathcal{T}_0 \subset \mathcal{T}_1 \subset \ldots \mathcal{T}_L \subset \Omega$. The associated finite element spaces X_i are also nested: $X_0 \subset X_1 \subset \cdots \subset X_L$. The multigrid approach proceeds as follows:

- Initialization: Let $\mathbf{v}_0 = 0$ be the initial guess. Minimize the CB elasticity problem discretized on $\mathcal{T}_0$ to obtain $\mathbf{u}_0$.
- For $i = 1, \ldots, L$:
  - Interpolate $\mathbf{v}_i = I_{i-1}^i \mathbf{u}_{i-1}$, where $I_{i-1}^i : X_{i-1} \to X_i$ is the interpolation operator.
  - Use $\mathbf{v}_i$ as initial guess to minimize the CB problem discretized on $\mathcal{T}_i$.
- At the finest level L, construct the initial atomic locations by $\mathbf{y}_{CB} = \mathbf{x} + \mathbf{v}_L(\mathbf{x})$ and solve the molecular mechanics problem using $\mathbf{y}_{CB}$ as initial guess.

This method seems to bypass many local minima and keeps the original physically relevant minimum, and appears to be insensitive to the initial conditions and parameters of the nonlinear solvers. The method possesses optimal computational complexity for homogeneous deformations.

Sublinear Scaling Algorithms

For large-scale problems, even linear scaling algorithms might not be computationally feasible. In such

cases, it is necessary to resort to sublinear scaling methods, that is, algorithms whose complexity scales sublinearly with the size of the system. In fact, from an algorithmic viewpoint, one of the main purposes of multiscale modeling is to develop sub-linear scaling algorithms and some general methodologies, such as the heterogeneous multiscale method, have been developed for this purpose [31].

In the case of crystalline solids, an example of a sublinear scaling algorithm is the quasicontinuum (QC) method [32], developed to study systems in which a plastic deformation only occurs on a vanishingly small part of the whole sample. In the original QC method, representative atoms (rep-atoms) are introduced to reduce the number of degrees of freedom in regions where the atomic displacement is smooth; in those regions, the energy is approximated by using a simplified summation rule based on the Cauchy-Born hypothesis. The methodology has been extended to the context of orbital-free DFT [33, 34] (see the entry ▸ Atomistic to Continuum Coupling).

A different approach based on asymptotic analysis was presented in [35, 36]. Algorithms in the context of both orbital-free DFT and Kohn-Sham DFT were presented. The leading order in the asymptotics corresponds to the cauhy-born rule, but the asymptotic analysis also provides a systematic approach to improve the accuracy of the model. The main idea is to divide the localized orbitals of the electrons into two sets: one set associated with the atoms in the region where the deformation of the material is smooth (smooth region) and an other associated with the atoms around the defects (non-smooth region). The orbitals associated with atoms in the smooth region can be approximated accurately using asymptotic analysis, and the results can then be used to find the orbitals in the non-smooth region using a formulation of Kohn-Sham DFT for an embedded system.

References

1. Kohn, W., Sham, L.J.: Self-consistent equations including exchange and correlation effects. Phys. Rev. **140**(4A), 1133–1138 (1965)
2. Parr, R.G., Yang, W.: Density-Functional Theory of Atoms and Molecules. International Series of Monographs on Chemistry. Oxford University Press, New York (1989)
3. Goodwin, L., Needs, R.J., Heine, V.: A pseudopotential total energy study of impurity-promoted intergranular embrittlement. J. Phys. Condens. Matter **2**, 351–365 (1990)
4. Vanderbilt, D.: Soft self-consistent pseudopotentials in a generalized eigenvalue formalism. Phys. Rev. B **41**, 7892–7895 (1990)
5. Troullier, N., Martins, J.L.: Efficient pseudopotentials for plane-wave calculations. Phys. Rev. B **43**(3), 1993–2006 (1991)
6. Laasonen, K., Car, R., Lee, C., Vanderbilt, D.: Implementation of ultrasoft pseudopotentials in ab initio molecular dynamics. Phys. Rev. B **43**, 6796–6799 (1991)
7. Goedecker, S.: Linear scaling electronic structure methods. Rev. Mod. Phys. **71**(4), 1085–1123 (1999)
8. Kohn, W.: Density functional and density matrix method scaling linearly with the number of atoms. Phys. Rev. Lett. **76**(17), 3168–3171 (1996)
9. Marzari, N., Vanderbilt, D.: Maximally localized generalized Wannier functions for composite energy bands. Phys. Rev. B **56**(20), 12847–12865 (1997)
10. E, W., Li, T., Lu, J.: Localized basis of eigensubspaces and operator compressions. PNAS. **105**(23), 7907–7912 (2008)
11. Martin, R.M.: Electronic Structure: Basic theory and practical methods. Cambridge University Press, Cambridge (2005)
12. García-Cervera, C.J., Lu, J., Xuan, Y., Weinan, E.: A linear scaling subspace iteration algorithm with optimally localized non-orthogonal wave functions for kohn-sham density functional theory. Phys. Rev. B **79**(11), 115110 (2009)
13. Zhou, Y., Saad, Y., Tiago, M.L., Chelikowsky, J.R.: Self-consistent-field calculations using Chebyshev-filtered subspace iteration. J. Comput. Phys. **219**(1), 172–184 (2006)
14. Yang, W.: Electron density as the basic variable: a divide-and-conquer approach to the ab initio computation of large molecules. J. Mol. Struct. Theochem **255**, 461–479 (1992)
15. Jansík, B., Host, S., Jorgensen, P., Olsen, J., Helgaker, T.: Linear-scaling symmetric square-root decomposition of the overlap matrix. J. Chem. Phys. **126**(12), 124104 (2007)
16. Hellmann, H.: Einführung in die Quantenchemie. Deuticke, Leipzig (1937)
17. Feynman, R.P.: Forces in molecules. Phys. Rev. **56**(4), 340–343 (1939)
18. Barnes, J., Hut, P.: A hierarchical $O(N \log(N))$ force calculation algorithm. Nature **324**, 446–449 (1986)
19. Greengard, L., Rokhlin, V.: A fast algorithm for particle simulations. J. Comput. Phys. **73**, 325–348 (1987)
20. Greengard, L., Rokhlin, V.: A new version of the fast multipole method for the Laplace equation in three dimensions. In: Acta Numerica 1997. Acta Numerica, vol. 6, pp. 229–269. Cambridge University Press, Cambridge (1997)
21. Hackbusch, W.: A sparse matrix arithmetic based on $\mathcal{H}$-matrices. I. Introduction to $\mathcal{H}$-matrices. Computing **62**(2), 89–108 (1999)
22. Hackbusch, W., Khoromskij, B., Sauter, S.A.: On $\mathcal{H}^2$-matrices. In: Lectures on Applied Mathematics (Munich, 1999), pp. 9–29. Springer, Berlin (2000)
23. Pavliotis, G., Stuart, A.: Multiscale Methods: Averaging and Homogenization. Texts in Applied Mathematics. Springer, New York (2008)

24. Weinan, E.: Principles of Multiscale Modeling. Cambridge University Press, Cambridge/New York (2011)
25. Brandt, A.: Multiscale Scientific Computation: Review. In: Barth, T.J., Chan, T.F., Haimes, R. (eds.) Multiscale and Multiresolution Methods: Theory and Applications, pp. 3–96. Springer, Berlin/New York (2002)
26. Brandt, A.: Multi-level adaptive solutions to boundary-value problems. Math. Comp. **31**(138), 333–390 (1977)
27. Hackbush, W.: Convergence of multigrid iterations applied to difference equations. Math. Comput. **34**(150), 425–440 (1980)
28. Trottenberg, U., Oosterlee, C.W., Schuller, A.: Multigrid. Academic, San Diego (2000)
29. Chen, J., Ming, P.B.: An efficient multigrid method for molecular mechanics modeling in atomic solids. Commun. Comput. Phys. **10**(1), 70–89 (2011)
30. Born, M., Huang, K.: Dynamical Theory of Crystal Lattices. Oxford University Press, Oxford (1954)
31. Weinan, E., Engquist, B., Li, X., Ren, W., Vanden-Eijnden, E.: Heterogeneous multiscale methods: a review. Commun. Comput. Phys. **2**(3), 367–450 (2007)
32. Tadmor, E.B., Ortiz, M., Phillips, R.: Quasicontinuum analysis of defects in solids. Philos. Mag. A **73**, 1529–1563 (1996)
33. Hayes, R.L., Fago, M., Ortiz, M., Carter, E.A.: Prediction of dislocation nucleation during nanoindentation by the orbital-free density functional theory local quasi-continuum method. Multiscale Model. Simul. **4**(2), 359–389 (2006)
34. Gavini, V., Bhattacharya, K., Ortiz, M.: Quasi-continuum orbital-free density-functional theory: a route to multi-million atom non-periodic DFT calculation. J. Mech. Phys. Solid **55**(4), 697–718 (2007)
35. García-Cervera, C.J., Lu, J., E, W.: Asymptotics-based sub-linear scaling algorithms and application to the study of the electronic structure of materials. Commun. Math. Sci. **5**(4), 999–1026 (2007)
36. E, W., Lu, J.: The Kohn-Sham equations for deformed crystals, Mem. Am. Math. Soc. To appear (2012). (http://dx.doi.org/10.1090/S0065-9266-2012-00659-9)

Linear Time Independent Reaction Diffusion Equations: Computation

Christos Xenophontos
Department of Mathematics and Statistics, University of Cyprus, Nicosia, Cyprus

Mathematics Subject Classification

65N30; 65N50

Short Definition

Linear time-independent reaction-diffusion equations are a class of elliptic partial differential equations in which the highest derivative is multiplied by a small positive parameter which can approach zero. As a result, their solutions usually exhibit boundary layer behavior for small values of the parameter.

Introduction

We consider the following steady-state, reaction-diffusion boundary value problem: Find u such that

$$-\varepsilon^2\nabla^2 u + bu = f \text{ in } \Omega \subset \mathbf{R}^n, \quad u = 0 \text{ on } \partial\Omega, \quad (1)$$

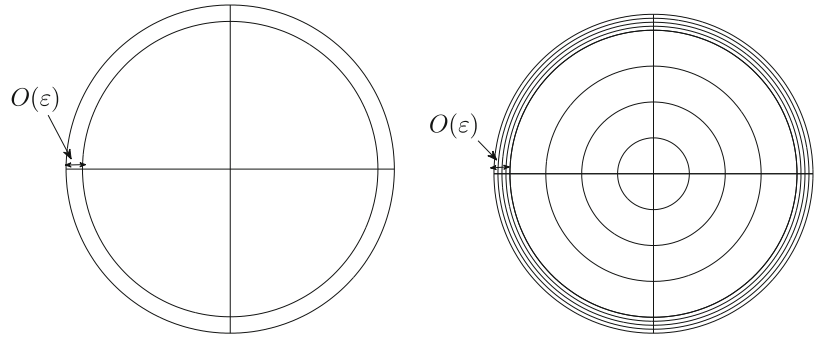
where n $(= 1, 2, 3)$ is the dimension, $\varepsilon \in (0, 1]$ is a given parameter, b, f are given functions of x $(= x_1, \ldots, x_n)$, and the domain Ω is assumed to be bounded with $\partial\Omega$ denoting its boundary. The homogeneous Dirichlet boundary condition is simply chosen for convenience; other boundary conditions may be treated as well.

The presence of ε in (1) causes the solution to, in general, have *boundary layers*, especially as $\varepsilon \to 0$. These are rapidly varying solution components which have support in a narrow neighborhood along $\partial\Omega$. This is in addition to any other "peculiarities" that might exist due to the possible lack of smoothness in the data and/or the domain. In order for the approximation to be reliable and robust, *all* features of the solution must be dealt with so that the accuracy is not affected (in a negative way) as $\varepsilon \to 0$.

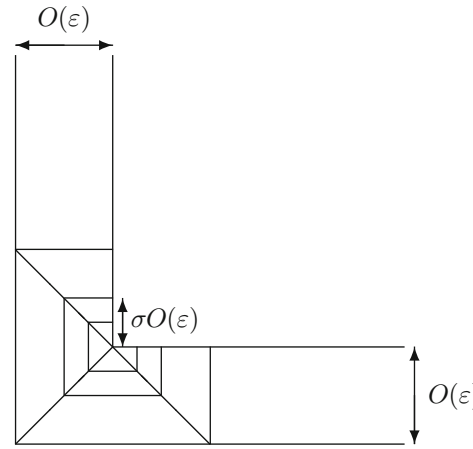
The approximation to the solution u of (1) may be obtained in a variety of ways: finite differences, spectral methods, and finite elements, to name a few. Although we will focus on the Finite Element Method (FEM), the guidelines given below apply to most other methods as well.

Mesh Design Principles

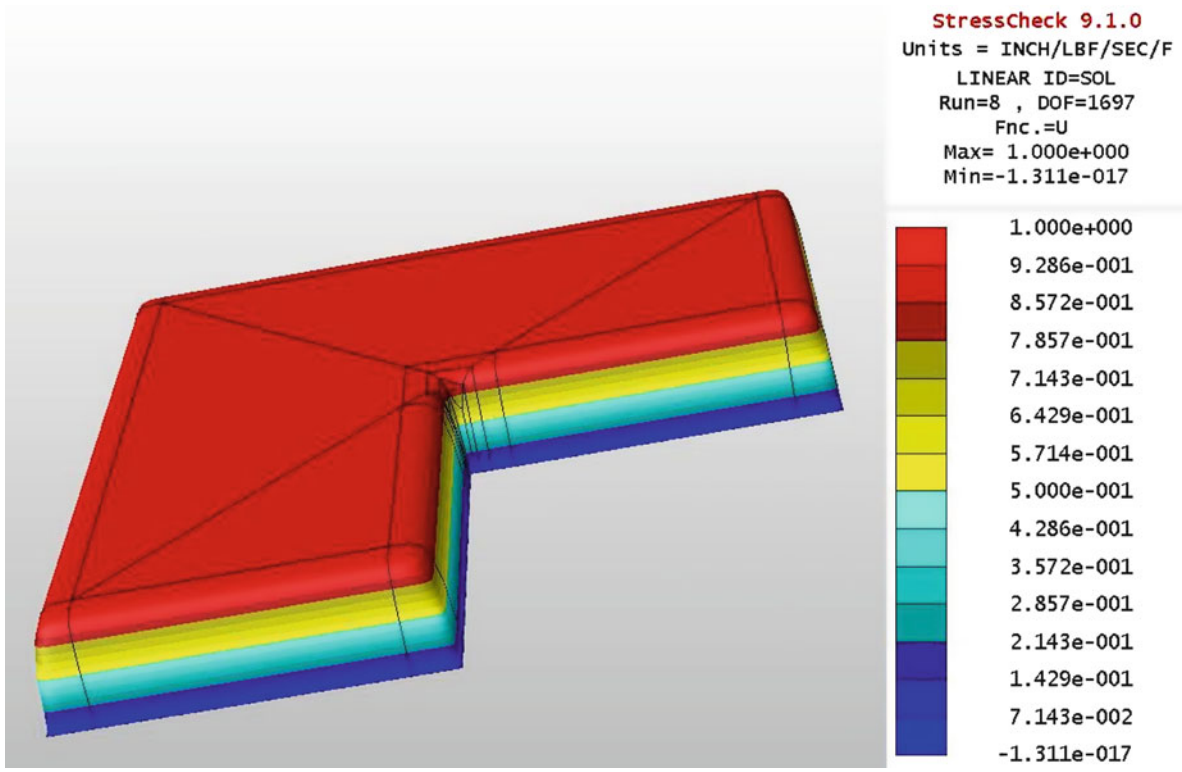
Whether one uses commercial software or writes their own subroutines, the *correct* mesh-degree combinations are as follows: If the data is smooth and the



Linear Time Independent Reaction Diffusion Equations: Computation, Fig. 1 Mesh design for a circle. *Left*: Initial h-FEM mesh or fixed p-FEM mesh. *Right*: Refined h-FEM mesh, in a piecewise uniform fashion (referred to as *Shishkin mesh* [6])



Linear Time Independent Reaction Diffusion Equations: Computation, Fig. 2 Mesh design near a reentrant corner; the parameter σ controls the geometric ratio and in this figure is chosen as 1/2; the "optimal" value is $\sigma \approx 0.15$



Linear Time Independent Reaction Diffusion Equations: Computation, Fig. 3 Approximate solution to (1) with $\varepsilon = 0.01, b = f = 1$

domain does not contain any corners or abrupt changes in the boundary conditions, the only feature of the solution that needs to be resolved is the boundary layer. For that, it suffices to construct the mesh in a way that it includes refinement along an $O(\varepsilon)$ neighborhood of the boundary. This is due to the fact that the boundary layer effect is essentially one dimensional, namely, in the direction normal to the boundary [2–7]. Figure 1 shows an example of such a minimal mesh when the domain is a circle.

If the domain contains corners, then corner singularities will also be present – this will also be the case if there is an abrupt change in the boundary conditions even if the boundary is smooth. The appropriate mesh to use in this case must also include sufficient refinement near each singularity in order for that feature to be adequately resolved (as well). This can be achieved by either the use of a nonuniform (e.g., *geometric* [1]) refinement near each corner or, alternatively, the use an adaptive method. For the former, we show in Fig. 2 an example of such a mesh near a reentrant corner.

In Fig. 3 we show the approximate solution to (1) with $\varepsilon = 0.01, b = f = 1$, when Ω is an L-shaped domain. The approximation was obtained with the p-FEM commercial software package StressCheck (ESRD, St Louis, MO, USA), using polynomials of degree $p = 8$. The mesh contains $O(\varepsilon)$ refinement along the boundary as well as geometric refinement near the reentrant corner as seen in Fig. 2. For more theoretical and practical considerations, as well as additional examples from solid mechanics, see [5].

References

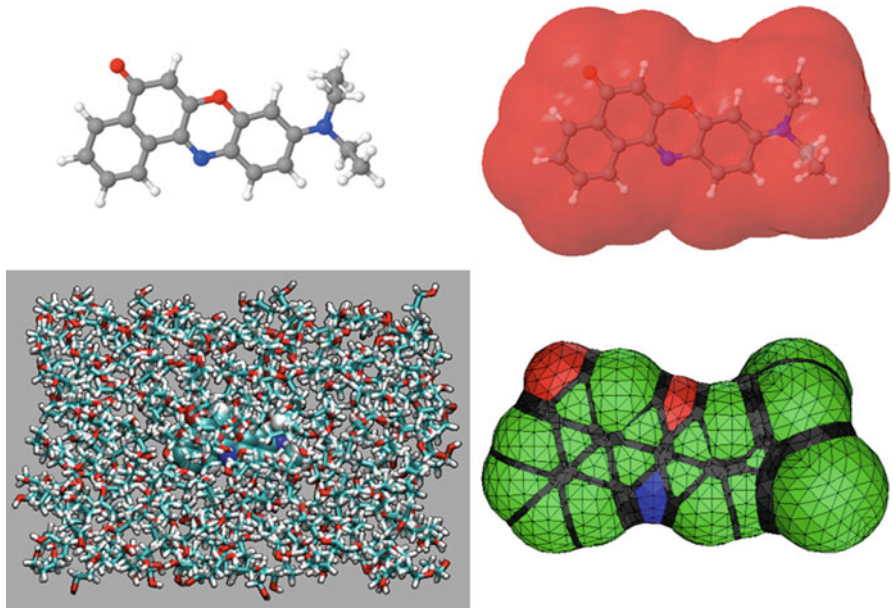
1. Babuška, I., Guo, B.: The $h-p$ version of the finite element method, Part 1: the basic approximation results. Comput. Mech. **1**, 21–41 (1986)
2. Melenk, J.M.: hp-Finite Element Methods for Singular Perturbations. Springer, Berlin/Heidelberg/New York (2002)
3. Melenk, J.M., Schwab, C.: Analytic regularity for a singularly perturbed problem. SIAM J. Math. Anal., SINUM **30**(2), 379–400 (1999)
4. Schwab, C., Suri, M.: The p and hp versions of the finite element method for problems with boundary layers. Math. Comp. **65**(216), 1403–1429 (1996)
5. Schwab, C., Suri, M., Xenophontos, C.: The hp finite element method for problems in mechanics with boundary layers. Comput. Methods Appl. Mech. Eng. **57**(3/4), 311–334 (1998)
6. Shishkin, G.: Grid approximation of singularly perturbed boundary value problems with a regular boundary layer. Sov. J. Numer. Anal. Math. Model. **4**, 397–417 (1989)
7. Xenophontos, C.: The hp finite element method for singularly perturbed problems, Ph.D. Dissertation, University of Maryland, Baltimore County (1996)

Liquid-Phase Simulation: Theory and Numerics of Hybrid Quantum-Mechanical/Classical Approaches

Benedetta Mennucci
Department of Chemistry, University of Pisa, Pisa, Italy

Description

A liquid represents an extremely complex system. Even if we limit the analysis to an equilibrium picture, the liquid can be seen as a large assembly of molecules undergoing incessant collisions and exchanging energy among colliding partners and among internal degrees of freedom. The particles are disordered at large scale, but often there is a local order that fades away. The same description can be used also for solutions where the collection of particles contains at least two types of molecules, those having a higher molar fraction are called the solvent, the others the solute. This purely classical description implicitly contains an essential component which is intrinsically nonclassical, namely, the molecular interactions determining the behavior of the liquid system. A correct description of these interactions should require the introduction of a quantum mechanical (QM) picture, but it is clear that a detailed QM description of a liquid is impossible due to the huge number of interacting molecules to be considered together with the huge number of different configurations of these molecules to be accounted for in order to get a statistically meaningful picture. There are two possible strategies commonly adopted to overcome this problem, either we go back to a fully classical picture in which a parameterized description of the intra- and intermolecular interactions is introduced, or we divide the entire system into two parts, one of larger interest (e.g., the solute) which is treated at QM level and the remainder which can be seen as a classical perturbation. These two strategies correspond to two alternative computational approaches, the full classical



Liquid-Phase Simulation: Theory and Numerics of Hybrid Quantum-Mechanical/Classical Approaches, Fig. 1 Example of QM/MM and QM/Continuum representations of typical organic chromophore (*Nile Red*) within an ethanol solution. In the last picture, the typical surface mesh used within the PCM approach is shown

molecular mechanics (MM) and the hybrid QM/classical approaches. In the former, all the molecules are treated at the same level introducing a classical force field to represent the intra- and intermolecular interactions [1] whereas the correct sampling can be obtained using either a dynamical or a statistical simulation: molecular dynamics (MD) or Monte Carlo (MC) methods are commonly used to this scope. By contrast, in the hybrid QM/classical approach, the solute is treated quantum-mechanically while the remainder (the solvent) is treated classically either using a MM description [7, 12] or a continuum approximation [13, 15] (see Fig. 1).

Within the continuum approximation, the microscopic nature of the solvent completely disappears and it is substituted by a macroscopic dielectric medium. This is clearly an extreme simplification but still can lead to accurate results of the effect of the environment on molecular properties and processes if a correct physical and numerical formulation is used. Moreover, the use of a dielectric medium also automatically solves the problem of a correct sampling. In fact, describing the solvent in terms of its macroscopic properties (such the dielectric permittivity) in most cases allows to use a single configuration, that is, the equilibrated solute within the dielectric, instead of requiring many solute-solvent configurations as in full MM or QM/MM formulations.

From this brief introduction, it comes out that the simulation of the liquid phase remains a challenge. Many alternative methodologies are available, and they rapidly change with the progress of the computing technology. This has the negative consequence that it is impossible to give an exhaustive overview of the subject but instead a preliminary choice on the range of methods which shall be covered has to be done. Due to the rapid increase of the computational power available at relatively low cost and of the easiness of use and accuracy of quantum-chemical softwares, it appears that hybrid QM/classical methods represent today one of the most promising strategies to simulate liquids with the level of details required to evaluate molecular properties and processes in condensed phase. It is therefore on this family of methods that we shall almost exclusively focus in the present contribution.

Hybrid QM/Classical Approaches

As said, the QM/classical strategy collects methods in which a target subsystem defined as the "solute" is described at QM level, and a secondary subsystem ("the solvent") is, on the contrary, modeled at a classical level using either a MM force field or a macroscopic

continuum medium with suitable properties. In both versions, a fundamental common aspect is present: The QM part can be modified in its electronic and nuclear characteristics by the presence of the classical part. This coupling between the two parts is made possible by introducing in the QM description of the isolated solute a new term which represents the effects exerted by the classical part. In a QM language, this is obtained by replacing the Hamiltonian operator representing the solute alone with a new or *effective* one including an additional solute-solvent interacting term, namely:

$$\hat{H}_{\text{eff}} |\Psi\rangle = \left(\hat{H}_0 + \hat{H}_{\text{env}}\right) |\Psi\rangle = E |\Psi\rangle \qquad (1)$$

where $\hat{H}_0$ and $|\Psi\rangle$ are the Hamiltonian and the wavefunction relative to the solute and $\hat{H}_{\text{env}}$ is the solvent induced term. As for isolated molecules, also the effective Schrödinger equation (1) cannot be treated without further approximations. What is important to stress, however, is that the addition of the new operator $\hat{H}_{\text{env}}$ does not change the formal and the numerical strategy to be used. As a result, the most commonly used approximations for isolated systems ▸Density Functional Theory, ▸Quantum Monte Carlo Methods in Chemistry, ▸Hartree–Fock Type Methods, ▸Coupled-Cluster Methods, are still valid for the liquid phase. However, the form of $\hat{H}_{\text{env}}$ which depends on the specific version of the QM/classical formulation used introduces some important specificities. Here below we briefly summarize the main ones for each of the two selected families of solvation methods.

QM/MM

If we adopt a microscopic description in terms of an MM force field, the effects that the classical part of the system exert on the QM part are of electrostatic, repulsive, and dispersive nature. The latter terms are of short-range character and in most combined QM/MM methods are described by empirical potentials independent of the QM electronic degrees of freedom, thus not affecting the solute wavefunction. On the contrary, the electrostatic contribution, usually depicted in terms of atomic charges placed on the atoms of the solvent molecules, will explicitly affect (or polarize) the solute wavefunction. Its effects will be introduced in $\hat{H}_{\text{env}}$ in terms of an additional one-electron term which represents the electrostatic energy between a set of point charges placed in the solvent and a solute charge distribution generating an electrostatic potential at the same points. This formulation of the QM/MM approach, generally indicated as "electrostatic embedding," differentiates from the more approximated version in which the QM–MM electrostatic interaction is treated on the same footing as the MM–MM electrostatics ("mechanical embedding").

To make the solvent effects more complete, in addition to point charges, we can introduce induced dipoles, describing each solvent atom (or group of atoms) in terms of an atomic charge and an atomic polarizability. As a result, not only the solute will be polarized by the solvent but also the solvent will respond to the solute so, to achieve a mutually polarized system. This formulation of the QM/MM approach is known as "polarized embedding."

Within this polarizable QM/MM formulation we get:

$$\hat{H}_{\text{env}} = \hat{H}_{\text{QM/MM}} + \hat{H}_{\text{MM}} \qquad (2)$$

$$\begin{aligned}\hat{H}_{\text{QM/MM}} &= \hat{H}^{\text{el}}_{\text{QM/MM}} + \hat{H}^{\text{pol}}_{\text{QM/MM}} \\ &= \sum_m q_m \hat{V}(r_m) - \frac{1}{2}\sum_a \mu_a^{\text{ind}} \hat{\mathbf{E}}_a^{\text{solute}}(r_a)\end{aligned} \qquad (3)$$

$$\begin{aligned}\hat{H}_{\text{MM}} = \hat{H}^{\text{el}}_{\text{MM}} + \hat{H}^{\text{pol}}_{\text{MM}} &= \sum_m \sum_{n>m} \frac{q_m q_n}{r_{mn}} \\ &- \frac{1}{2}\sum_a \mu_a^{\text{ind}} \sum_m \frac{q_m(\mathbf{r}_a - \mathbf{r}_m)}{|\mathbf{r}_a - \mathbf{r}_m|^3}\end{aligned} \qquad (4)$$

where $\hat{V}(r_m)$ and $\hat{\mathbf{E}}_a^{\text{solute}}(r_a)$ represent the electrostatic potential and the electric field operators due to the solute electrons and nuclei calculated at the MM sites. On the other hand, in (4) $\hat{H}^{\text{el}}_{\text{MM}}$ describes the electrostatic self-energy of the MM charges, while $\hat{H}^{\text{pol}}_{\text{MM}}$ represents the polarization interaction between such charges and the induced dipoles. We recall that the $\hat{H}^{\text{el}}_{\text{MM}}$ term enters in the effective Hamiltonian only as a constant energetic quantity, while the $\hat{H}^{\text{pol}}_{\text{MM}}$ contribution explicitly depends on the QM wavefunction.

Numerical Aspects of Polarizable MM Approaches

The dipoles induced on each MM polarizable site can be obtained assuming a linear approximation,

neglecting any contribution of magnetic character related to the total electric field, and using an isotropic polarizability for each selected point in the MM part of the system. The electric field which determines such dipoles contains a sum of contributions from the solute, from the solvent point charges, and from the induced dipole moments themselves. This mutual polarization between the dipoles can be solved through a matrix inversion approach, by introducing a matrix equation:

$$\mathbf{K}\mu_c^{\text{ind}} = \mathbf{E}_c \tag{5}$$

where the matrix $\mathbf{K}$ is of dimension $3\mathrm{N} \times 3\mathrm{N}$, N being the number of polarizable sites, and the vector $\mathbf{E}_c$ collects the c-th component of the electric field from the solute and the solvent permanent charge distribution. The form of matrix $\mathbf{K}$ will be determined uniquely by the position of the polarizable sites and the polarizability values, namely:

$$\begin{aligned}
\mathbf{K}_{i,i} &= \mathbf{K}_{i+N,i+N} = \mathbf{K}_{i+2N,i+2N} = 1/\alpha_i \\
\mathbf{K}_{i,i+N} &= \mathbf{K}_{i,i+2N} = \mathbf{K}_{i+N,i} = \mathbf{K}_{i+N,i+2N} \\
&= \mathbf{K}_{i+2N,i} = \mathbf{K}_{i+2N,i+N} = 0 \\
\mathbf{K}_{i+mN,j+nN} &= T_{i,j}^{\text{kl}}
\end{aligned}$$

with $n, m = 0, 1, 2$ and $k, l = x, y, z$
where the index i and $j \neq i$ run from 1 to N, and the dipole field tensor is given by:

$$\mathbf{T}_{i,j} = \frac{1}{r_{ij}^3}\mathbf{I} - \frac{3}{r_{ij}^5}\begin{bmatrix} r_x^2 & r_x r_y & r_x r_z \\ r_y r_x & r_y^2 & r_y r_z \\ r_z r_x & r_z r_y & r_z^2 \end{bmatrix} \tag{6}$$

The QM/MM formalism can accommodate almost any combination of QM and MM methods. The choice of the QM method follows the same criteria as in pure QM studies. Essentially, the QM code must be able to perform the self-consistent field (SCF) ▸ Hartree–Fock Type Methods treatment in the presence of the external point-charge (or dipole) field that represents the MM charge model in the case of electronic (or polarized) embedding. In practice, many current QM/MM applications use density-functional theory (DFT) ▸ Density Functional Theory as the QM method owing to its favorable computational-effort/accuracy ratio. Traditionally, semiempirical QM methods have been most popular, and they remain important for extensions of QM/MM approaches to molecular dynamics. The recent development of linear-scaling for correlation methods has significantly extended the size of systems that can be treated with such methods, up to several tens of atoms, and has made them a very accurate alternative to be coupled with an MM description of the environment. As far as the choice of MM method is concerned, all the many force fields available in the literature can, in principle, be coupled with a QM description.

QM/Continuum

The analysis of QM/classical methods is less straightforward if we adopt a continuum description. The basic formulation of continuum models requires the solution of a classical electrostatic problem (Poisson problem):

$$-\vec{\nabla}\cdot\left[\varepsilon(\vec{r})\vec{\nabla}V(\vec{r})\right] = 4\pi\rho_M(\vec{r}) \tag{7}$$

where $\rho_M(\vec{r})$ is the solute charge distribution and $\varepsilon(\vec{r})$ is the general position-dependent permittivity. If we assume that the charge distribution is contained in a molecular cavity C of proper shape and dimension built within a homogeneous and isotropic solvent, $\varepsilon(\vec{r})$ assumes the simple form:

$$\varepsilon(\vec{r}) = \begin{cases} 1 & \vec{r} \in C \\ \varepsilon & \vec{r} \notin C \end{cases} \tag{8}$$

where ε is the dielectric constant of the solvent.

Using the definition (8) with the appropriate boundary conditions, the electrostatic problem (7) can be solved in terms of a potential V which is the sum of the solute potential plus the contribution due to the reaction of the solvent (e.g., the polarization of the dielectric), namely $V(\vec{r}) = V_M(\vec{r}) + V_\sigma(\vec{r})$. Under the assumption that the charge distribution is entirely supported inside the cavity C, an integral representation of the reaction potential can be derived which introduces a fictitious (or *apparent*) charge distribution σ on the boundary between the solute and the solvent, that is, the surface of the cavity C, $\Gamma = \partial C$, namely:

$$V_\sigma(\vec{r}) = \int_\Gamma \frac{\sigma(\vec{s})}{|\vec{r} - \vec{s}|} d\vec{s} \tag{9}$$

The surface charge σ is solution of an integral equation on Γ, that is of an equation of the form [3–5]:

$$(A\sigma)(\vec{s}) = \int_{\Gamma} k_A(\vec{s},\vec{s}\,')\,\sigma(\vec{s}\,')\,d\vec{s}\,' = b_\rho(\vec{s}) \quad \forall \vec{s} \in \Gamma \tag{10}$$

where k_A is the Green kernel of some integral operator A and b_ρ depends linearly on the charge distribution ρ_M. This formulation has been adopted in different continuum solvation models, the most famous ones being the polarizable continuum model (PCM) [15] (in its different versions) and the conductor-like screening model (COSMO) [9]. Each different formulation corresponds to different choices for A, but in all cases, it is obtained in terms of a specific combination of the following kernels:

$$k_A(\vec{s},\vec{s}\,') = \begin{cases} \frac{1}{|\vec{s}-\vec{s}\,'|} \\ \frac{\partial}{\partial \hat{n}_s}\frac{1}{|\vec{s}-\vec{s}\,'|} \\ \frac{\partial}{\partial \hat{n}_{s'}}\frac{1}{|\vec{s}-\vec{s}\,'|} \end{cases} \tag{11}$$

where $\hat{n}_s$ represents the unit vector normal to the surface at point $\vec{s}$ and pointing toward the dielectric.

Also b_ρ changes according to the different formulation of the model. For instance, the original version of COSMO is obtained with:

$$b_\rho(\vec{s}) = -f(\epsilon)\int_{\mathbf{R}^3} \frac{\rho_M(\vec{r}\,')}{|\vec{s}-\vec{r}\,'|}\,d\vec{r}\,' \tag{12}$$

where $f(\epsilon) = (\epsilon - 1)/(\epsilon + 0.5)$.

Numerical Aspects of Polarizable Continuum Approaches

The reduction of the source of the solvent reaction potential to a charge distribution limited to a closed surface greatly simplifies the electrostatic problem with respect to other formulations in which the whole dielectric medium is considered as source of the reaction potential. In spite of this remarkable simplification, the integration of (10) over a surface of complex shape is computationally challenging. The solutions are generally based on a discretization of the integral into a finite number of elements. This discretization of Γ automatically leads to a discretization of $\sigma(\vec{s})$ in terms of point-like charges, namely if we assume that on each surface element $\sigma(\vec{s})$ does not significantly change, its effect can be simulated with that of a point charge of value $q(\vec{s}_i) = \sigma(\vec{s}_i)a_i$ where a_i is the area of the surface element i and $\vec{s}_i$ its representative point. This numerical method, which can be defined as P_0 collocation method, is not the only possible one (e.g., a Galerkin method could also be used); however, it is the most natural and easiest to implement for the specific case of apparent surface charge calculations [14].

The necessary preliminary step in the strategy is the generation of the surface elements (i.e., the surface mesh, see Fig. 1) as, once the mesh has been defined, the apparent charges q are obtained by solving a matrix equation, of the type

$$\mathbf{Qq} = -\mathbf{RV}_M \tag{13}$$

where $\mathbf{q}$ and $\mathbf{V}_M$ are the vectors containing the N values of the charge and the solute potential at the surface points, respectively. $\mathbf{Q}$ and $\mathbf{R}$ are the matrix analogs of the integral operators introduced in (10) to obtain the apparent charge distribution σ. In particular, the different kernels reported in the (11) can be written in terms of the following matrices:

$$\begin{aligned} S_{ij} &= \frac{1}{|\vec{s}_i - \vec{s}_j|} \\ D_{ij} &= \frac{(\vec{s}_i - \vec{s}_j)\cdot \hat{n}_j}{|\vec{s}_i - \vec{s}_j|^3} \\ D^*_{ij} &= \frac{(\vec{s}_j - \vec{s}_i)\cdot \hat{n}_i}{|\vec{s}_i - \vec{s}_j|^3} \end{aligned} \tag{14}$$

As concerns the diagonal elements of $\mathbf{S}$, $\mathbf{D}$ and $\mathbf{D}^*$ different numerical solutions have been proposed. In particular, those commonly used are $S_{ii} = k\sqrt{4\pi/a_i}$ and $D_{ii} = -(2\pi + \sum_{j\neq i} D_{ij}a_j)/a_i$ where the former derives from the exact formula of a flat circular element with k taking into account that the element is spherical, and the latter becomes exact when the size of all the elements tends to zero.

The approximation method described above belongs to the class of boundary element methods (BEM) [2]. BEM follows the same lines as finite element methods (FEM). In both cases, the approximation space is constructed from a mesh. In the context of continuum solvation models, FEM solves the (local) partial differential equation (7), complemented with convenient boundary conditions, a 3D mesh [6, 8], while BEM solves one of the (nonlocal) integral equations derived above, on a 2D mesh. In the former case, the resulting linear system is very large, but sparse. In the latter case, it is of much lower size, but full.

If we now reintroduce a QM description of the charge distribution ρ_M in terms of the wavefunction

which is solution of the (1), we can rewrite the solvent induced term $\hat{H}_{\text{env}}$ as:

$$\hat{H}_{\text{env}} = \hat{H}_{\text{QM/cont}} = \sum_m q(\vec{s}_i)\hat{V}(\vec{s}_i) \qquad (15)$$

where q are the solvent apparent charges and $\hat{V}$ is the electrostatic potential operator corresponding to the solute charge distribution. By comparing (15) with (4), it might seem that there is a perfect equivalence between the nonpolarizable part of the QM/MM method and the QM/continuum one. As a matter of fact this equivalence is only apparent as the apparent charges entering in (15) are not external parameters as it is for the MM charges but they are obtained solving a matrix equation which depends on the solute charge distribution. In (4), the induced dipoles μ_a^{ind} depend on the solute charge distribution exactly as the apparent charges.

The analogies and differences between QM/ Continuum and QM/MM approaches however are not only on the methodological aspects of their formulation and implementation. It is important to recall that the two approaches also present fundamental specificities from a physical point of view. By definition, continuum models introduce an averaged (bulk) description of the environment effects. This is necessarily reflected in the results that can be obtained with these methods. While continuum models can be successfully applied in all cases in which the environment acts as a mean-field perturbation, solvent-specific effects such as hydrogen bondings are not well reproduced. By contrast, QM/MM methods can properly describe many specific effects but, at the same time, they cannot be applied to simulate longer-to-bulk effects if they are not coupled to a sampling of the configurational space of the solute–solvent system. For this, a molecular dynamics (MD) or Monte Carlo (MC) simulation approach is needed with significant increase of the computational cost.

Conclusions

Many alternative strategies are available to simulate the liquid phase, each with its advantages and weaknesses. Here, in particular, the attention has been focused on the class of methods which combine a QM description of the subsystem of interest with a classical one for the remainder. This hybrid approach is extremely versatile, we can in fact tune the boundary between the two components of the system as well as extend the dimensions of the classical system and change its description using either an atomistic (MM) or a continuum approach. In addition, both QM/MM and QM/continuum methods can be applied to environments of increasing complexity [10, 11], from standard isotropic and homogeneous liquids, to gas–liquid or liquid–liquid interfaces and/or anisotropic liquid crystalline phases, just to quote few. The most important aspect of these methods, however, is that the QM approach, even if limited to just a part of the system, allows for a more accurate description of all those processes and phenomena which are mostly based on the electronic structure of the molecules constituting the liquid. In more details, QM/classical methods should be preferred over other fully classical approaches when the interest is not on the properties of the liquid itself but instead on the effects that the liquid exerts on a property or a process which can be localized on a specific part of the system. The realm of (bio)chemical reactivity in solution as well as the world of spectroscopies in condensed phase are examples where QM/classical methods really represent the most effective approach. Of course there are also drawbacks; in particular, the computational cost can increase enormously with respect to classical methods especially when the QM/Classical approach is coupled to molecular dynamics simulations. Moreover, the choice of the specific combination of the QM description and the classical one is not straightforward, but it has to be carefully chosen on the basis of the specific problem under investigation and the specific chemical system of interest. It is however clear that QM/classical methods represent one of the most powerful approaches to combine accuracy with complexity while still keeping a physically founded representation of the main interactions determining the behavior of the liquids.

References

1. Allen, M.P., Tildesley, D.: Computer Simulations of Liquids. Oxford University Press, London (1987)
2. Beskos, D.E. (ed.): Boundary Element Methods in Mechanics, vol 3. North–Holland, Amsterdam (1989)
3. Cances, E.: Integral equation approaches for continuum models. In: Mennucci, B., Cammi, R. (eds.) Continuum Solvation Models in Chemical Physics, From Theory to Applications, pp. 29–48. Wiley, Hoboken (2007)

4. Cances, E., Mennucci, B.: New applications of integral equations methods for solvation continuum models: ionic solutions and liquid crystals. J. Math. Chem. **23**, 309–326 (1998)
5. Cances, E., Le Bris, C., Mennucci, B., Tomasi, J.: Integral equation methods for molecular scale calculations in the liquid phase. Math. Models Methods Appl. Sci. **9**, 35–44 (1999)
6. Cortis, C., Friesner, R.: An automatic three-dimensional finite element mesh generation system for the poisson-boltzmann equation. J. Comput. Chem. **18**(13), 1570–1590 (1997)
7. Gao, J.L.: Hybrid quantum and molecular mechanical simulations: an alternative avenue to solvent effects in organic chemistry. Acc. Chem. Res. **29**(6), 298–305 (1996)
8. Holst, M., Baker, N., Wang, F.: Adaptive multilevel finite element solution of the Poisson–Boltzmann equation i. algorithms and examples. J. Comput. Chem. **21**, 1319–1342 (2000)
9. Klamt, A.: The COSMO and COSMORS solvation models. Wiley Interdiscip. Rev. Comput. Mol. Sci. **1**(5), 699–709 (2011)
10. Lin, H., Truhlar, D.G.: QM/MM: What have we learned, where are we, and where do we go from here? Theor. Chem. Acc. **117**, 185–199 (2007)
11. Mennucci, B.: Continuum solvation models: what else can we learn from them? J. Phys. Chem. Lett. **1**(10), 1666–1674 (2010)
12. Senn, H.M., Thiel, W.: QM/MM methods for biomolecular systems. Angew. Chem. Int. Ed. **48**(7), 1198–1229 (2009)
13. Tomasi, J., Persico, M.: Molecular interactions in solution – an overview of methods based on continuous distributions of the solvent. Chem. Rev. **94**(7), 2027–2094 (1994)
14. Tomasi, J., Mennucci, B., Laug, P.: The modeling and simulation of the liquid phase. In: Le Bris, C. (ed.) Handbook of Numerical Analysis: Special Volume. Computational Chemistry, pp. 271–323. Elsevier, Amsterdam (2003)
15. Tomasi, J., Mennucci, B., Cammi, R.: Quantum mechanical continuum solvation models. Chem. Rev. **105**(8), 2999–3093 (2005)

Lobatto Methods

Laurent O. Jay
Department of Mathematics, The University of Iowa, Iowa City, IA, USA

Introduction

Lobatto methods for the numerical integration of differential equations are named after Rehuel Lobatto. Rehuel Lobatto (1796–1866) was a Dutch mathematician working most of his life as an advisor for the government in the fields of life insurance and of weights and measures. In 1842, he was appointed professor of mathematics at the Royal Academy in Delft (known nowadays as Delft University of Technology). Lobatto methods are characterized by the use of approximations to the solution at the two end points t_n and t_{n+1} of each subinterval of integration $[t_n, t_{n+1}]$. Two well-known Lobatto methods based on the trapezoidal quadrature rule which are often used in practice are the *(implicit) trapezoidal rule* and the *Störmer-Verlet-leapfrog method*.

The (Implicit) Trapezoidal Rule

Consider a system of ordinary differential equations (ODEs):

$$\frac{d}{dt}y = f(t, y) \tag{1}$$

where $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$. Starting from y_0 at t_0 one step $(t_n, y_n) \mapsto (t_{n+1}, y_{n+1})$ of the (implicit) trapezoidal rule applied to (1) is given by the implicit relation:

$$y_{n+1} = y_n + \frac{h_n}{2}\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right)$$

where $h_n = t_{n+1} - t_n$ is the step size. The (implicit) trapezoidal rule is oftentimes called the *Crank-Nicholson method* when considered in the context of time-dependent partial differential equations (PDEs). This implicit method requires the solution of a system of d equations for $y_{n+1} \in \mathbb{R}^d$ that can be expressed as:

$$F(y_{n+1}) := y_{n+1} - y_n - \frac{h_n}{2}\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right) = 0$$

and which is nonlinear when $f(t, y)$ is nonlinear in y. Starting from an initial guess $y_{n+1}^{(0)} \approx y_{n+1}$, the solution y_{n+1} can be approximated iteratively by modified Newton iterations as follows:

$$y_{n+1}^{(k+1)} = y_{n+1}^{(k)} + p_{n+1}^{(k)}, \quad J_n p_{n+1}^{(k)} = -F(y_{n+1}^{(k)})$$

using, for example, an approximate Jacobian:

$$J_n = I_d - \frac{h_n}{2} D_y f(t_n, y_n) \approx D_y F(y_{n+1}^{(k)}).$$

Taking $J_n = I_d$ leads to fixed-point iterations:

$$y_{n+1}^{(k+1)} = y_n + \frac{h_n}{2}\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1}^{(k)})\right).$$

The Generalized Newton-Störmer-Verlet-Leapfrog Method

Consider now a partitioned system of ODEs:

$$\frac{d}{dt}q = v(t, p, q), \quad \frac{d}{dt}p = f(t, q, p) \tag{2}$$

where $v : \mathbb{R} \times \mathbb{R}^{d_q} \times \mathbb{R}^{d_p} \to \mathbb{R}^{d_q}$ and $f : \mathbb{R} \times \mathbb{R}^{d_q} \times \mathbb{R}^{d_p} \to \mathbb{R}^{d_p}$. Starting from (q_0, p_0) at t_0 one step $(t_n, q_n, p_n) \mapsto (t_{n+1}, q_{n+1}, p_{n+1})$ of the *generalized Newton-Störmer-Verlet-leapfrog method* applied to (2) reads:

$$\begin{aligned} p_{n+1/2} &= p_n + \frac{h_n}{2} f(t_n, q_n, p_{n+1/2}), \\ q_{n+1} &= q_n + \frac{h_n}{2}\left(v(t_n, q_n, p_{n+1/2}) \right. \\ &\qquad \left. + v(t_{n+1}, q_{n+1}, p_{n+1/2})\right), \\ p_{n+1} &= p_{n+1/2} + \frac{h_n}{2} f(t_{n+1}, q_{n+1}, p_{n+1/2}) \end{aligned} \tag{3}$$

where $h_n = t_{n+1} - t_n$ is the step size. The first equation is implicit for $p_{n+1/2}$, the second equation is implicit for q_{n+1}, and the last equation is explicit for p_{n+1}. When $v(t, q, p) = v(t, p)$ is independent of q, and $f(t, q, p) = f(t, q)$ is independent of p the method is fully explicit. If in addition $v(t, q, p) = v(p)$ is independent of t and q, the method can be simply expressed as:

$$\begin{aligned} p_{n+1/2} &= p_n + \frac{h_n}{2} f(t_n, q_n), \\ q_{n+1} &= q_n + h_n v(p_{n+1/2}), \\ p_{n+1} &= p_{n+1/2} + \frac{h_n}{2} f\ (t_{n+1}, q_{n+1}). \end{aligned}$$

This explicit method is often applied as follows:

$$\begin{aligned} p_{n+1/2} &= p_{n-1/2} + \frac{1}{2}(h_{n-1} + h_n) f\ (t_n, q_n), \\ q_{n+1} &= q_n + h_n v(p_{n+1/2}). \end{aligned}$$

Depending on the field of applications, this method is known under different names: the *Störmer method* in astronomy; the *Verlet method* in molecular dynamics; the *leapfrog method* in the context of time-dependent PDEs, in particular for wave equations. This method can be traced back to Newton's Principia (1687), see [10].

Lobatto Methods

In this entry, we consider families of Runge-Kutta (RK) methods based on Lobatto quadrature formulas whose simplest member is the trapezoidal quadrature rule. When applied to (1) Lobatto RK methods can be expressed as follows:

$$\begin{aligned} Y_{ni} &= y_n + h_n \sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_{nj}) \\ &\qquad \text{for } i = 1, \ldots, s, \end{aligned} \tag{4}$$

$$y_{n+1} = y_n + h_n \sum_{j=1}^{s} b_j f(t_n + c_j h, Y_{nj}) \tag{5}$$

where the stage value s satisfies $s \geq 2$ and the coefficients a_{ij}, b_j, c_j characterize the Lobatto RK method. The s intermediate values Y_{nj} for $j = 1, \ldots, s$ are called the *internal stages* and can be considered as approximations to the solution at $t_n + c_j h_n$, the main numerical RK approximation at $t_{n+1} = t_n + h_n$ is given by y_{n+1}. Lobatto RK methods are characterized by $c_1 = 0$ and $c_s = 1$. They can also be considered in combination with other families of RK methods, for example, with Gauss methods in the context of certain systems of differential-algebraic equations (DAEs), see the section "Lobatto Methods for DAEs" below. The symbol III is usually found in the literature associated to Lobatto methods, the symbols I and II being reserved for the two types of Radau methods. The (implicit) trapezoidal rule is the simplest member ($s = 2$) in the Lobatto IIIA family. The generalized Newton-Störmer-Verlet-leapfrog method seen above can be interpreted as a partitioned Runge-Kutta (PRK) resulting from the combination of the (implicit) trapezoidal rule and the Lobatto IIIB method for $s = 2$, see the section "Additive Lobatto Methods for Split and Partitioned ODEs" below.

Families of Lobatto Methods

For a fixed value of s, the various families of Lobatto methods described below all share the same coefficients b_j, c_j of the corresponding Lobatto quadrature formula.

Lobatto Quadrature Formulas

The problem of approximating a Riemann integral:

$$\int_{t_n}^{t_n+h_n} f(t)dt \tag{6}$$

with f assumed to be continuous is equivalent to the problem of solving the initial value problem at $t = t_n + h_n$:

$$\frac{d}{dt}y = f(t), \quad y(t_n) = 0$$

since $y(t_n + h_n) = \int_{t_n}^{t_n+h_n} f(t)dt$. The integral (6) can be approximated by using a standard quadrature formula:

$$\int_{t_n}^{t_n+h_n} f(t)dt \approx h_n \left(\sum_{i=1}^{s} b_i f(t_n + c_i h_n) \right)$$

with s *node* coefficients $c_1, \ldots, c_s$, and s *weight* coefficients $b_1, \ldots, b_s$. Lobatto quadrature formulas, also known as Gauss-Lobatto quadrature formulas in the literature, are given for $s \geq 2$ by a set of nodes and weights satisfying conditions described hereafter. The s nodes c_j are the roots of the polynomial of degree s:

$$\frac{d^{s-2}}{dt^{s-2}}(t^{s-1}(1-t)^{s-1}).$$

These nodes satisfy $c_1 = 0 < c_2 < \ldots < c_s = 1$. The weights b_j and nodes c_j satisfy the condition $B(2s-2)$ where:

$$B(p) \; : \; \sum_{j=1}^{s} b_j c_j^{k-1} = \frac{1}{k} \quad \text{for } k = 1, \ldots, p,$$

implying that the quadrature formula is of order $2s-2$. There exists an explicit formula for the weights

$$b_j = \frac{1}{s(s-1)P_{s-1}(2c_j-1)^2} > 0$$

$$\text{for } j = 1, \ldots, s \quad \left(b_1 = b_s = \frac{1}{s(s-1)} \right)$$

where

$$P_k(x) = \frac{1}{k!2^k}\frac{d^k}{dx^k}\left((x^2-1)^k\right)$$

is the kth Legendre polynomial. Lobatto quadrature formulas are symmetric, that is their nodes and weights satisfy:

$$b_{s+1-j} = b_j, \quad c_{s+1-j} = 1 - c_j \quad \text{for } j = 1, \ldots, s.$$

For $s = 3$, we obtain the famous Simpson's rule:

$$(b_1, b_2, b_3) = (1/6, 2/3, 1/6), (c_1, c_2, c_3) = (0, 1/2, 1).$$

Procedures to compute numerically accurately the nodes and weights of high order Lobatto quadrature formulas can be found in [7] and [23]. The subroutine `GQRUL` from the `IMSL/MATH-LIBRARY` can compute numerically these nodes and weights.

Lobatto Families

The families of Lobatto RK methods differ only in the values of their coefficients a_{ij}. Various equivalent definitions can be found in the literature. The coefficients a_{ij} of these families can be linearly implicitly defined with the help of so-called *simplifying assumptions*:

$$C(q) : \quad \sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{c_i^k}{k}$$

$$\text{for } i = 1, \ldots, s \text{ and } k = 1, \ldots, q,$$

$$D(r) : \quad \sum_{i=1}^{s} b_i c_i^{k-1} a_{ij} = \frac{b_j}{k}\left(1 - c_j^k\right)$$

$$\text{for } j = 1, \ldots, s \text{ and } k = 1, \ldots, r.$$

The importance of these simplifying assumptions comes from a fundamental result due to Butcher, see [5, 9], saying that a RK method satisfying the simplifying assumptions $B(p)$, $C(q)$, and $D(r)$ is of order at least $\min(p, 2q+2, q+r+1)$. The coefficients a_{ij}, b_j, c_j characterizing the Lobatto RK method (4) and (5) will be displayed below in the form of a table called a *Butcher-tableau*:

$$\begin{array}{r|ccccc} c_1 = 0 & a_{11} & a_{12} & \cdots & a_{1,s-1} & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2,s-1} & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{s-1} & a_{s-1,1} & a_{s-1,2} & \cdots & a_{s-1,s-1} & a_{s-1,s} \\ c_s = 1 & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}$$

In the four main families of Lobatto methods described below, namely Lobatto IIIA, Lobatto IIIB, Lobatto IIIC, and Lobatto IIIC*, only one method does not satisfy the relation $C(1)$, that is,

$$\sum_{j=1}^{s} a_{ij} = c_i \quad \text{for } i = 1, \ldots, s,$$

this is the Lobatto IIIB method for $s = 2$, see below. The Lobatto IIIA, IIIB, IIIC, and IIIC* methods can all be interpreted as perturbed collocation methods [19] and discontinuous collocation methods [11].

Lobatto IIIA

The coefficients a_{ij}^A of Lobatto IIIA methods can be defined by $C(s)$ (Table 1). They satisfy $D(s-2)$, $a_{sj}^A = b_j$ for $j = 1, \ldots, s$, and $a_{1j}^A = 0$ for $j = 1, \ldots, s$. Lobatto IIIA methods are symmetric and of nonstiff order $2s-2$. Their stability function $R(z)$ is given by the $(s-1, s-1)$-Padé approximation to e^z. They are A-stable, but not L-stable since $R(\infty) = (-1)^{s+1}$. They are not B-stable and thus not algebraically stable. They can be interpreted as collocation methods. Since the first internal stage Y_{n1} of Lobatto IIIA methods is explicit ($Y_{n1} = y_n$ and $f(t_n + c_1 h_n, Y_{n1}) = f(t_n, y_n)$) and the last internal stage satisfies $Y_{ns} = y_{n+1}$ (and thus $f(t_{n+1}, y_{n+1}) = f(t_n + c_s h_n, Y_{ns})$), these methods are comparable in terms of computational work to Gauss methods with $s-1$ internal stages since they also have the same nonstiff order $2s-2$. For $s = 2$, we obtain the (implicit) trapezoidal rule which is often expressed without its two internals stages Y_{n1}, Y_{n2} since they are respectively equal to y_n and y_{n+1}. The method for $s = 3$ is sometimes called the *Hermite-Simpson (or Clippinger-Dimsdale) method* and it has been used, for example, in trajectory optimization problems [4]. This method can be equivalently expressed in a compact form as:

$$\begin{aligned} Y_{n2} &= \frac{1}{2}(y_n + y_{n+1}) \\ &\quad + \frac{h_n}{8}(f(t_n, y_n) - f(t_{n+1}, y_{n+1})), \\ y_{n+1} &= y_n + \frac{h_n}{6}\left(f(t_n, y_n) + 4f(t_{n+1/2}, Y_{n2})\right. \\ &\quad \left. + f(t_{n+1}, y_{n+1})\right) \end{aligned}$$

where $t_{n+1/2} = t_n + h_n/2$. It can be even further reduced by rewriting

$$\begin{aligned} y_{n+1} &= y_n + \frac{h_n}{6}\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right) \\ &\quad + \frac{2h_n}{3} f\left(t_{n+1/2}, \frac{1}{2}(y_n + y_{n+1})\right. \\ &\quad \left. + \frac{h_n}{8}(f(t_n, y_n) - f(t_{n+1}, y_{n+1}))\right). \end{aligned}$$

Lobatto IIIB

The coefficients a_{ij}^B of Lobatto IIIB methods can be defined by $D(s)$ (Table 2). They satisfy $C(s-2)$, $a_{i1}^B = b_1$ for $i = 1, \ldots, s$ and $a_{is}^B = 0$ for $i = 1, \ldots, s$. Lobatto IIIB methods are symmetric and of nonstiff order $2s-2$. Their stability function $R(z)$ is given by the $(s-1, s-1)$-Padé approximation to e^z. They are A-stable, but not L-stable since $R(\infty) = (-1)^{s+1}$. They are not B-stable and thus not algebraically stable. The coefficients a_{ij}^B can also be obtained from the coefficients a_{ij}^A of Lobatto IIIA through the relations:

$$b_i a_{ij}^B + b_j a_{ji}^A - b_i b_j = 0 \quad \text{for } i, j = 1, \ldots, s,$$

or

$$a_{ij}^B = b_j - a_{s+1-i,s+1-j}^A \quad \text{for } i, j = 1, \ldots, s.$$

Lobatto IIIC

The coefficients a_{ij}^C of Lobatto IIIC methods can be defined by $a_{i1}^C = b_1$ for $i = 1, \ldots, s$ and $C(s-1)$ (Table 3). They satisfy $D(s-1)$ and $a_{sj}^C = b_j$ for $j = 1, \ldots, s$. Lobatto IIIC methods are of nonstiff order $2s-2$. They are not symmetric. Their stability function $R(z)$ is given by the $(s-2, s)$-Padé approximation to e^z. They are L-stable. They are algebraically stable and thus B-stable. They are excellent methods for stiff problems.

Lobatto Methods, Table 1 Coefficients of Lobatto IIIA for $s = 2, 3, 4, 5$

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline A_{s=2} & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline A_{s=3} & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}-\frac{\sqrt{5}}{10} & \frac{11+\sqrt{5}}{120} & \frac{25-\sqrt{5}}{120} & \frac{25-13\sqrt{5}}{120} & \frac{-1+\sqrt{5}}{120} \\ \frac{1}{2}+\frac{\sqrt{5}}{10} & \frac{11-\sqrt{5}}{120} & \frac{25+13\sqrt{5}}{120} & \frac{25+\sqrt{5}}{120} & \frac{-1-\sqrt{5}}{120} \\ 1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\ \hline A_{s=4} & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$$

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}-\frac{\sqrt{21}}{14} & \frac{119+3\sqrt{21}}{1960} & \frac{343-9\sqrt{21}}{2520} & \frac{392-96\sqrt{21}}{2205} & \frac{343-69\sqrt{21}}{2520} & \frac{-21+3\sqrt{21}}{1960} \\ \frac{1}{2} & \frac{13}{320} & \frac{392+105\sqrt{21}}{2880} & \frac{8}{45} & \frac{392-105\sqrt{21}}{2880} & \frac{3}{320} \\ \frac{1}{2}+\frac{\sqrt{21}}{14} & \frac{119-3\sqrt{21}}{1960} & \frac{343+69\sqrt{21}}{2520} & \frac{392+96\sqrt{21}}{2205} & \frac{343+9\sqrt{21}}{2520} & \frac{-21-3\sqrt{21}}{1960} \\ 1 & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \\ \hline A_{s=5} & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \end{array}$$

Lobatto IIIC*

Lobatto IIIC* are also known as Lobatto III methods [5], Butcher's Lobatto methods [9], and Lobatto III$\overline{C}$ methods [22] in the literature. (The name Lobatto IIIC* was suggested by Robert P.K. Chan in an e-mail correspondence with the author on June 13, 1995.) The coefficients $a_{ij}^{C^*}$ of Lobatto IIIC* methods can be defined by $a_{is}^{C^*} = 0$ for $i = 1, \ldots, s$ and $C(s-1)$ (Table 4). They satisfy $D(s-1)$ and $a_{1j}^{C^*} = 0$ for $j = 1, \ldots, s$. Lobatto IIIC* methods are of nonstiff order $2s-2$. They are not symmetric. Their stability function $R(z)$ is given by the $(s, s-2)$-Padé approximation to e^z. They are not A-stable. They are not B-stable and thus not algebraically stable. The Lobatto IIIC* method for $s = 2$ is sometimes called the *explicit trapezoidal rule*. The coefficients $a_{ij}^{C^*}$ can also be obtained from the coefficients a_{ij}^{C} of Lobatto IIIC through the relations:

$$b_i a_{ij}^{C^*} + b_j a_{ji}^{C} - b_i b_j = 0 \quad \text{for } i, j = 1, \ldots, s,$$

or

$$a_{ij}^{C^*} = b_j - a_{s+1-i,s+1-j}^{C} \quad \text{for } i, j = 1, \ldots, s.$$

Other Families of Lobatto Methods

Most Lobatto methods of interest found in the literature can be expressed as linear combinations of the four fundamental Lobatto IIIA, IIIB, IIIC, and IIIC* methods. In fact, one can consider a very general family of methods with three real parameters $(\alpha_A, \alpha_B, \alpha_C)$ by considering Lobatto coefficients of the form:

$$a_{ij}(\alpha_A, \alpha_B, \alpha_C) = \alpha_A a_{ij}^{A} + \alpha_B a_{ij}^{B} + \alpha_C a_{ij}^{C} + \alpha_{C^*} a_{ij}^{C^*} \tag{7}$$

where $\alpha_{C^*} = 1 - \alpha_A - \alpha_B - \alpha_C$. For any choice of $(\alpha_A, \alpha_B, \alpha_C)$ the corresponding Lobatto RK method is of nonstiff order $2s-2$ [13]. The Lobatto IIIS methods presented in [6] depend on a real parameter σ. They can be expressed as:

$$a_{ij}^{S}(\sigma) = (1-\sigma)\left(a_{ij}^{A} + a_{ij}^{B}\right) + \left(\sigma - \frac{1}{2}\right)\left(a_{ij}^{C} + a_{ij}^{C^*}\right)$$
$$\text{for } i, j = 1, \ldots, s,$$

corresponding to $\alpha_A = \alpha_B = 1 - \sigma$ and $\alpha_C = \alpha_{C^*} = \sigma - \frac{1}{2}$ in (7). These methods satisfy $C(s-2)$ and

Lobatto Methods, Table 2 Coefficients of Lobatto IIIB for $s = 2, 3, 4, 5$

$$\begin{array}{c|cc} 0 & \frac{1}{2} & 0 \\ 1 & \frac{1}{2} & 0 \\ \hline B_{s=2} & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{array}{c|ccc} 0 & \frac{1}{6} & -\frac{1}{6} & 0 \\ \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 \\ 1 & \frac{1}{6} & \frac{5}{6} & 0 \\ \hline B_{s=3} & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

$$\begin{array}{c|cccc} 0 & \frac{1}{12} & \frac{-1-\sqrt{5}}{24} & \frac{-1+\sqrt{5}}{24} & 0 \\ \frac{1}{2}-\frac{\sqrt{5}}{10} & \frac{1}{12} & \frac{25+\sqrt{5}}{120} & \frac{25-13\sqrt{5}}{120} & 0 \\ \frac{1}{2}+\frac{\sqrt{5}}{10} & \frac{1}{12} & \frac{25+13\sqrt{5}}{120} & \frac{25-\sqrt{5}}{120} & 0 \\ 1 & \frac{1}{12} & \frac{11-\sqrt{5}}{24} & \frac{11+\sqrt{5}}{24} & 0 \\ \hline B_{s=4} & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$$

$$\begin{array}{c|ccccc} 0 & \frac{1}{20} & \frac{-7-\sqrt{21}}{120} & \frac{1}{15} & \frac{-7+\sqrt{21}}{120} & 0 \\ \frac{1}{2}-\frac{\sqrt{21}}{14} & \frac{1}{20} & \frac{343+9\sqrt{21}}{2520} & \frac{56-15\sqrt{21}}{315} & \frac{343-69\sqrt{21}}{2520} & 0 \\ \frac{1}{2} & \frac{1}{20} & \frac{49+12\sqrt{21}}{360} & \frac{8}{45} & \frac{49-12\sqrt{21}}{360} & 0 \\ \frac{1}{2}+\frac{\sqrt{21}}{14} & \frac{1}{20} & \frac{343+69\sqrt{21}}{2520} & \frac{56+15\sqrt{21}}{315} & \frac{343-9\sqrt{21}}{2520} & 0 \\ 1 & \frac{1}{20} & \frac{119-3\sqrt{21}}{360} & \frac{13}{45} & \frac{119+3\sqrt{21}}{360} & 0 \\ \hline B_{s=5} & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \end{array}$$

$D(s-2)$. They are symmetric and symplectic. Their stability function $R(z)$ is given by the $(s-1, s-1)$-Padé approximation to e^z. They are A-stable, but not L-stable. They are algebraically stable and thus B-stable. The Lobatto IIIS coefficients for $\sigma = 1/2$ are given by:

$$a_{ij}^{S}(1/2) = \frac{1}{2}\left(a_{ij}^{A} + a_{ij}^{B}\right) \quad \text{for } i, j = 1, \ldots, s.$$

For $\sigma = 1$ we obtain the Lobatto IIID methods [6, 13]:

$$a_{ij}^{D} = a_{ij}^{S}(1) = \frac{1}{2}\left(a_{ij}^{C} + a_{ij}^{C^*}\right) \quad \text{for } i, j = 1, \ldots, s.$$

These methods are called Lobatto III_E in [19] and Lobatto IIIE in [22]. They satisfy $C(s-1)$ and $D(s-1)$, and they can be interpreted as perturbed collocation methods [19]. Another family of Lobatto RK methods is given by the Lobatto III_D family of [19] called here Lobatto IIINW where the coefficients for $s = 2, 3$ are given in Table 5. (Notice on p. 205 of [19] that $\gamma_1 = -4(2m-1)$.) These methods correspond to $\alpha_A = 2$, $\alpha_B = 2$, $\alpha_C = -1$, and $\alpha_{C^*} = -2$ in (7). Their stability function $R(z)$ is given by the $(s-2, s)$-Padé approximation to e^z. These methods are L-stable. They are algebraically stable and thus B-stable. They are of nonstiff order $2s-2$. They are not symmetric. They can be interpreted as perturbed collocation methods [19].

Additive Lobatto Methods for Split and Partitioned ODEs

Consider a split system of ODEs:

$$\frac{d}{dt}y = f_1(t, y) + f_2(t, y) \tag{8}$$

where $f_1, f_2 : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$. Starting from y_0 at t_0 one step $(t_n, y_n) \mapsto (t_{n+1}, y_{n+1})$ of an *additive Lobatto RK method* applied to (8) reads:

Lobatto Methods, Table 3 Coefficients of Lobatto IIIC for $s = 2, 3, 4, 5$

$$\begin{array}{c|cc} 0 & \frac{1}{2} & -\frac{1}{2} \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline C_{s=2} & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{array}{c|ccc} 0 & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{6} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline C_{s=3} & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

$$\begin{array}{c|cccc} 0 & \frac{1}{12} & -\frac{\sqrt{5}}{12} & \frac{\sqrt{5}}{12} & -\frac{1}{12} \\ \frac{1}{2}-\frac{\sqrt{5}}{10} & \frac{1}{12} & \frac{1}{4} & \frac{10-7\sqrt{5}}{60} & \frac{\sqrt{5}}{60} \\ \frac{1}{2}+\frac{\sqrt{5}}{10} & \frac{1}{12} & \frac{10+7\sqrt{5}}{60} & \frac{1}{4} & -\frac{\sqrt{5}}{60} \\ 1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\ \hline C_{s=4} & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$$

$$\begin{array}{c|ccccc} 0 & \frac{1}{20} & -\frac{7}{60} & \frac{2}{15} & -\frac{7}{60} & \frac{1}{20} \\ \frac{1}{2}-\frac{\sqrt{21}}{14} & \frac{1}{20} & \frac{29}{180} & \frac{47-15\sqrt{21}}{315} & \frac{203-30\sqrt{21}}{1260} & -\frac{3}{140} \\ \frac{1}{2} & \frac{1}{20} & \frac{329+105\sqrt{21}}{2880} & \frac{73}{360} & \frac{329-105\sqrt{21}}{2880} & \frac{3}{160} \\ \frac{1}{2}+\frac{\sqrt{21}}{14} & \frac{1}{20} & \frac{203+30\sqrt{21}}{1260} & \frac{47+15\sqrt{21}}{315} & \frac{29}{180} & -\frac{3}{140} \\ 1 & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \\ \hline C_{s=5} & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \end{array}$$

$$\begin{aligned} Y_{ni} &= y_n + h_n \sum_{j=1}^{s} (a_{1,ij} f_1(t_n + c_j h, Y_{nj}) \\ &\quad + a_{2,ij} f_2(t_n + c_j h, Y_{nj})) \\ &\quad \text{for } i = 1, \ldots, s, \\ y_{n+1} &= y_n + h_n \sum_{j=1}^{s} b_j (f_1(t_n + c_j h, Y_{nj}) \\ &\quad + f_2(t_n + c_j h, Y_{nj})) \end{aligned}$$

where $s \geq 2$ and the coefficients $a_{1,ij}, a_{2,ij}, b_j, c_j$ characterize the additive Lobatto RK method. Consider, for example, any coefficients $a_{1,ij}$ and $a_{2,ij}$ from the family (7), the additive method is of nonstiff order $2s - 2$ [13]. The partitioned system of ODEs (2) can be expressed in the form (8) by having $d = d_q + d_p$, $y = (q, p) \in \mathbb{R}^{d_q} \times \mathbb{R}^{d_p}$, and:

$$f_1(t, q, p) = \begin{pmatrix} v(t, q, p) \\ 0 \end{pmatrix},$$

$$f_2(t, q, p) = \begin{pmatrix} 0 \\ f(t, q, p) \end{pmatrix}.$$

Applying for $s = 2$ the Lobatto IIIA coefficients as $a_{1,ij}$ and the Lobatto IIIB coefficients as $a_{2,ij}$, we obtain again the generalized Newton-Störmer-Verlet-leapfrog method (3). Additive Lobatto methods have been considered in multibody dynamics in [13, 21]. Additive methods are more general than partitioned methods since partitioned system of ODEs can always be reformulated as a split system of ODEs, but the reverse is false in general.

Lobatto Methods for DAEs

An important use of Lobatto methods is for the solution of differential-algebraic equations (DAEs). DAEs consist generally of coupled systems of differential equations and nonlinear relations. They arise typically in mechanics and electrical/electronic circuits simulation.

Lobatto Methods, Table 4 Coefficients of Lobatto IIIC* for $s = 2, 3, 4, 5$

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline C^*_{s=2} & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\ 1 & 0 & 1 & 0 \\ \hline C^*_{s=3} & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}-\frac{\sqrt{5}}{10} & \frac{5+\sqrt{5}}{60} & \frac{1}{6} & \frac{15-7\sqrt{5}}{60} & 0 \\ \frac{1}{2}+\frac{\sqrt{5}}{10} & \frac{5-\sqrt{5}}{60} & \frac{15+7\sqrt{5}}{60} & \frac{1}{6} & 0 \\ 1 & \frac{1}{6} & \frac{5-\sqrt{5}}{12} & \frac{5+\sqrt{5}}{12} & 0 \\ \hline C^*_{s=4} & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$$

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}-\frac{\sqrt{21}}{14} & \frac{1}{14} & \frac{1}{9} & \frac{13-3\sqrt{21}}{63} & \frac{14-3\sqrt{21}}{126} & 0 \\ \frac{1}{2} & \frac{1}{32} & \frac{91+21\sqrt{21}}{576} & \frac{11}{72} & \frac{91-21\sqrt{21}}{576} & 0 \\ \frac{1}{2}+\frac{\sqrt{21}}{14} & \frac{1}{14} & \frac{14+3\sqrt{21}}{126} & \frac{13+3\sqrt{21}}{63} & \frac{1}{9} & 0 \\ 1 & 0 & \frac{7}{18} & \frac{2}{9} & \frac{7}{18} & 0 \\ \hline C^*_{s=5} & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \end{array}$$

Lobatto Methods, Table 5 Coefficients of Lobatto IIINW for $s = 2, 3$ [19]

$$\begin{array}{c|cc} 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & -\frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{array}{c|ccc} 0 & \frac{1}{6} & 0 & -\frac{1}{6} \\ \frac{1}{2} & \frac{1}{12} & \frac{5}{12} & 0 \\ 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

Consider, for example, a system of DAEs of the form:

$$\frac{d}{dt}y = f(t, y, \lambda), \quad 0 = k(t, y)$$

where $D_y k(t, y) D_\lambda f(t, y, \lambda)$ is nonsingular. Lobatto methods can be applied to this class of problems while preserving their classical order of convergence [14]. For example, the application of the two-stage Lobatto IIID method can be expressed as:

$$\begin{aligned} Y_{n1} &= y_n + \frac{h_n}{4}(f(t_n, Y_{n1}, \Lambda_{n1}) - f(t_{n+1}, Y_{n2}, \Lambda_{n2})), \\ Y_{n2} &= y_n + \frac{h_n}{4}(3f(t_n, Y_{n1}, \Lambda_{n1}) + f(t_{n+1}, Y_{n2}, \Lambda_{n2})), \\ y_{n+1} &= y_n + \frac{h_n}{2}(f(t_n, Y_{n1}, \Lambda_{n1}) + f(t_{n+1}, Y_{n2}, \Lambda_{n2})), \\ 0 &= \frac{1}{2}(k(t_n, Y_{n1}) + k(t_{n+1}, Y_{n2})), \\ 0 &= k(t_{n+1}, y_{n+1}). \end{aligned}$$

For such DAEs, a combination of Gauss and Lobatto coefficients is also considered in [18]. Consider now overdetermined system of DAEs (ODAEs) of the form:

$$\begin{aligned} &\frac{d}{dt}q = v(t, q, p), \ \frac{d}{dt}p = f(t, q, p, \lambda), 0 = g(t, q), \\ &\quad 0 = D_t g(t, q) + D_q g(t, q) v(t, q, p) \end{aligned} \tag{9}$$

where $D_q g(t, q) D_p v(t, q, p) D_\lambda f(t, q, p, \lambda)$ is nonsingular. Very general Lobatto methods can be applied to this type of ODAEs [13]. Hamiltonian and

Lagrangian systems with holonomic constraints can be expressed in the form (9). For such ODAEs, the application of Lobatto IIIA and IIIB methods can be shown to preserve their classical order of convergence, to be variational integrators, and to preserve a symplectic two-form [8, 11, 12, 17]. For example, the application of the two-stage Lobatto IIIA and IIIB method reads:

$$
\begin{aligned}
q_{n+1} &= q_n + \frac{h_n}{2}\left(v\left(t_n, q_n, p_{n+1/2}\right)\right. \\
&\qquad \left. + v\left(t_{n+1}, q_{n+1}, p_{n+1/2}\right)\right), \\
p_{n+1/2} &= p_n + \frac{h_n}{2} f\left(t_n, q_n, p_{n+1/2}, \Lambda_{n1}\right), \\
0 &= g(t_{n+1}, q_{n+1}), \\
p_{n+1} &= p_{n+1/2} + \frac{h_n}{2} f\left(t_{n+1}, q_{n+1}, p_{n+1/2}, \Lambda_{n2}\right) \\
0 &= D_t g(t_{n+1}, q_{n+1}) \\
&\quad + D_q g(t_{n+1}, q_{n+1}) v(t_{n+1}, q_{n+1}, p_{n+1}).
\end{aligned}
$$

Gauss methods with s stages can also be applied in combination with Lobatto methods with $s+1$ stages for this type of ODAEs when $f(t, q, p, \lambda)$ is decomposed in $f(t, q, p) + r(t, q, \lambda)$ and they also possess these aforementioned properties while generally requiring less computational effort [15]. For example, the application of the midpoint-trapezoidal method (the $(1, 1)$-Gauss-Lobatto SPARK method of Jay [15]) reads:

$$
\begin{aligned}
Q_{n1} &= q_n + \frac{h_n}{2} v(t_{n+1/2}, Q_{n1}, P_{n1}) = \frac{1}{2}(q_n + q_{n+1}), \\
P_{n1} &= p_n + \frac{h_n}{2} f(t_{n+1/2}, Q_{n1}, P_{n1}) \\
&\quad + \frac{h_n}{2} r(t_n, q_n, \Lambda_{n1}), \\
q_{n+1} &= q_n + h_n v(t_{n+1/2}, Q_{n1}, P_{n1}), \\
p_{n+1} &= p_n + h_n f(t_{n+1/2}, Q_{n1}, P_{n1}) \\
&\quad + h_n\left(\frac{1}{2} r(t_n, q_n, \Lambda_{n1}) + \frac{1}{2} r(t_{n+1}, q_{n+1}, \Lambda_{n2})\right), \\
0 &= g(t_{n+1}, q_{n+1}), \\
0 &= D_t g(t_{n+1}, q_{n+1}) \\
&\quad + D_q g(t_{n+1}, q_{n+1}) v(t_{n+1}, q_{n+1}, p_{n+1}).
\end{aligned}
$$

Lobatto Methods for Some Other Classes of Problems

Lobatto IIIA methods have been considered for boundary value problems (BVP) due to their good stability properties [1, 2]. The *MATLAB* code `bvp4c` for BVP is based on three-stage collocation at Lobatto points, hence it is equivalent to the three-stage Lobatto IIIA method [16]. Lobatto methods have also been applied to delay differential equations (DDEs) [3]. The combination of Lobatto IIIA and IIIB methods has also been considered for the discrete multisymplectic integration of certain Hamiltonian partial differential equations (PDEs) such as the nonlinear Schrödinger equation and certain nonlinear wave equations [20].

References

1. Ascher, U.M., Mattheij, R.M.M., Russell, R.D.: Numerical Solution of Boundary Value Problems for Ordinary Differential Equations. Classics in Applied Mathematics, vol. 13. SIAM, Philadelphia (1995)
2. Bashir-Ali, Z., Cash, J.R., Silva, H.H.M.: Lobatto deferred correction for stiff two-point boundary value problems. Comput. Math. Appl. **36**, 59–69 (1998)
3. Bellen, A., Guglielmi, N., Ruehli, A.E.: Methods for linear systems of circuit delay differential equations of neutral type. IEEE Trans. Circuits Syst. **46**, 212–216 (1999)
4. Betts, J.T.: Practical Methods for Optimal Control and Estimation Using Nonlinear Programming. Advances in Design and Control, 2nd edn. SIAM, Philadelphia (2008)
5. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations, 2nd edn. Wiley, Chichester (2008)
6. Chan, R.P.K.: On symmetric Runge-Kutta methods of high order. Computing **45**, 301–309 (1990)
7. Gautschi, W.: High-order Gauss-Lobatto formulae. Numer. Algorithms **25**, 213–222 (2000)
8. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems. Computational Mathematics, vol. 14, 2nd edn. Springer, Berlin (1996)
9. Hairer, E., Nørsett, S.P., Wanner, G.: Solving Ordinary Differential Equations I. Nonstiff Problems. Computational Mathematics, vol. 18, 2nd edn. Springer, Berlin (1993)
10. Hairer, E., Lubich, C., Wanner, G.: Geometric numerical integration illustrated by the Störmer/Verlet method. Acta Numer. vol. 12, 1–51 (2003)
11. Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration. Computational Mathematics, vol. 31, 2nd edn. Springer, Berlin (2006)
12. Jay, L.O.: Symplectic partitioned Runge-Kutta methods for constrained Hamiltonian systems. SIAM J. Numer. Anal. **33**, 368–387 (1996)
13. Jay, L.O.: Structure preservation for constrained dynamics with super partitioned additive Runge-Kutta methods. SIAM J. Sci. Comput. **20**, 416–446 (1998)

14. Jay, L.O.: Solution of index 2 implicit differential-algebraic equations by Lobatto Runge-Kutta methods. BIT **43**, 91–104 (2003)
15. Jay, L.O.: Specialized partitioned additive Runge-Kutta methods for systems of overdetermined DAEs with holonomic constraints. SIAM J. Numer. Anal. **45**, 1814–1842 (2007)
16. Kierzenka, J., Shampine, L.F.: A BVP solver based on residual control and the MATLAB PSE. ACM Trans. Math. Softw. **27**, 299–316 (2001)
17. Leimkuhler, B., Reich, S.: Simulating Hamiltonian Dynamics, Cambridge Monographs on Applied and Computational Mathematics, vol. 14. Cambridge University Press, Cambridge (2005)
18. Murua, A.: Partitioned Runge-Kutta methods for semi-explicit differential-algebraic systems of index 2. Technical Report EHU-KZAA-IKT-196, University of the Basque country (1996)
19. Nørsett, S.P., Wanner, G.: Perturbed collocation and Runge-Kutta methods. Numer. Math. **38**, 193–208 (1981)
20. Ryland, B.N., McLachlan, R.I.: On multisymplecticity of partitioned Runge-Kutta methods. SIAM J. Sci. Comput. **30**, 1318–1340 (2008)
21. Schaub, M., Simeon, B.: Blended Lobatto methods in multibody dynamics. Z Angew. Math. Mech. **83**, 720–728 (2003)
22. Sun, G.: A simple way of constructing symplectic Runge-Kutta methods. J. Comput. Math. **18**, 61–68 (2000)
23. von Matt, U.: Gauss quadrature. In: Gander, W., Hřebíček, J. (eds.) Solving Problems in Scientific Computing Using Maple and Matlab, vol. 14, 4th edn. Springer, Berlin (2004)

Logarithmic Norms

Gustaf Söderlind
Centre for Mathematical Sciences, Numerical Analysis, Lund University, Lund, Sweden

Introduction

The logarithmic norm is a real-valued functional on operators, quantifying the notions of *definiteness* for matrices; *monotonicity* for nonlinear maps; and *ellipticity* for differential operators. It is defined either in terms of an inner product in Hilbert space, or in terms of the operator norm on a Banach space.

The logarithmic norm has a wide range of applications in matrix theory, stability theory, and numerical analysis. It offers various quantitative bounds on (functions of) operators, operator spectra, resolvents, Rayliegh quotients, and the numerical range. It also offers error bounds and stability estimates in initial as well as boundary value problems and their discretizations. Some special fields in mathematics, such as semigroup theory, rely on notions that are strongly related to the logarithmic norm.

Let $|\cdot|$ denote an arbitrary vector norm on $\mathbb{C}^d$, as well as its subordinate operator norm on $\mathbb{C}^{d\times d}$. The classical definition of the *logarithmic norm* of $A \in \mathbb{C}^{d\times d}$ is

$$M[A] = \lim_{h\to 0+} \frac{|I + hA| - 1}{h}. \tag{1}$$

It is easily computed for the most common norms, see Table 1. In Hilbert space, where the norm is generated by an inner product $|x|^2 = \langle x, x\rangle$, one may alternatively define the *least upper bound* logarithmic norm $M[A]$ and the *greatest lower bound* logarithmic norm $m[A]$ such that for all x

$$m[A] \cdot |x|^2 \le \mathrm{Re}\,\langle x, Ax\rangle \le M[A] \cdot |x|^2. \tag{2}$$

Unlike (1), this also admits *unbounded operators*, while still agreeing with (1) if A is bounded, in which case it also holds that

$$m[A] = \lim_{h\to 0-} \frac{|I + hA| - 1}{h}. \tag{3}$$

The functionals $M[\cdot]$ and $m[\cdot]$ can further be extended to nonlinear maps, both in a Banach and a Hilbert space setting, so that the above definitions become special cases for linear operators.

The logarithmic norm has a large number of useful properties and satisfy several important inequalities. For $A, B \in \mathbb{C}^{d\times d}$, $\alpha \in \mathbb{R}$ and $z \in \mathbb{C}$, some of the most important are:

1. $-\mathrm{glb}[A] \le M[A] \le |A|$
2. $M[\alpha A] = \alpha M[A], \quad \alpha \ge 0$
3. $M[A + zI] = M[A] + \mathrm{Re}\, z$
4. $m[A] = -M[-A]$
5. $M[A] + m[B] \le M[A + B] \le M[A] + M[B]$
6. $|M[A] - M[B]| \le |A - B|$
7. $|m[A] - m[B]| \le |A - B|$
8. $\mathrm{e}^{tm[A]} \le |\mathrm{e}^{tA}| \le \mathrm{e}^{tM[A]}, \quad t \ge 0$
9. $M[A] < 0 \;\Rightarrow\; |A^{-1}| \le -1/M[A]$
10. $m[A] > 0 \;\Rightarrow\; |A^{-1}| \le 1/m[A]$.

Logarithmic Norms, Table 1 Computation of l^p vector, matrix, and logarithmic norms. Here $\rho[\cdot]$ and $\alpha[\cdot]$ denote the spectral radius and spectral abscissa of a matrix, respectively (From [3])

Vector norm	Matrix norm	Logarithmic norm
$\|x\|_1 = \sum_i \|x_i\|$	$\|A\|_1 = \max_j \sum_i \|a_{ij}\|$	$M_1[A] = \max_j \left(\operatorname{Re} a_{jj} + \sum_{i\neq j} \|a_{ij}\|\right)$
$\|x\|_2 = \sqrt{\sum_i \|x_i\|^2}$	$\|A\|_2 = \sqrt{\rho[A^{\mathrm{H}}A]}$	$M_2[A] = \alpha[(A + A^{\mathrm{H}})/2]$
$\|x\|_\infty = \max_i \|x_i\|$	$\|A\|_\infty = \max_i \sum_j \|a_{ij}\|$	$M_\infty[A] = \max_i \left(\operatorname{Re} a_{ii} + \sum_{j\neq i} \|a_{ij}\|\right)$

Differential Inequalities

The logarithmic norm was originally introduced for matrices, [3, 12], in order to establish bounds for solutions to a linear system

$$\dot{x} = Ax + r. \tag{4}$$

The norm of x satisfies the *differential inequality*

$$\mathrm{D}_t^+ |x| \le M[A] \cdot |x| + |r(t)|\,, \tag{5}$$

where $M[A]$ is the logarithmic norm of A and $\mathrm{D}_t^+|x|$ is the upper right *Dini derivative* of $|x|$ with respect to time. Consider first the homogeneous case $r \equiv 0$; this is akin to the *Grönwall lemma*. Then $x(t) = \mathrm{e}^{tA}x(0)$, and (5) provides the matrix exponential bound

$$|\mathrm{e}^{tA}| \le \mathrm{e}^{tM[A]}; \qquad t \ge 0. \tag{6}$$

Thus the condition $M[A] < 0$ implies that the matrix exponential is a *contraction (semi-)group*.

Consider next the case $x(0) = 0$, with $r \neq 0$. By integration of (5), the solution is then bounded on compact intervals by

$$|x(t)| \le \frac{\mathrm{e}^{tM[A]} - 1}{M[A]} \|r\|_\infty, \tag{7}$$

where $\|r\|_\infty = \sup_\tau |r(\tau)|$. If $M[A] < 0$, the bound also holds as $t \to \infty$, in which case

$$\|x\|_\infty \le -\frac{\|r\|_\infty}{M[A]}\,, \tag{8}$$

showing that x depends continuously on the data r.

Finally, consider $\dot{x} = Ax + r$ with $r \equiv$ const. If $M[A] < 0$, homogeneous solutions decay to a unique equilibrium $x = -A^{-1}r$. Taking $x(0) = -A^{-1}r$, (8) gives $|A^{-1}r| \le -|r|/M[A]$ for all r. Therefore, even the inverse of A can be bounded in terms of the logarithmic norm, as

$$M[A] < 0 \;\Rightarrow\; |A^{-1}| \le -\frac{1}{M[A]}\,. \tag{9}$$

This inequality is of particular importance also in boundary value problems, where it provides a bound for the inverse of an elliptic operator.

Spectral Bounds

For the spectrum of a general matrix A it holds that

$$\rho[A] \le |A|\,; \qquad \alpha[A] \le M[A], \tag{10}$$

where $\rho[A] = \max_i |\lambda_i|$ is the *spectral radius* of A and $\alpha[A] = \max_i \operatorname{Re}\lambda_i$ is the *spectral abscissa*. The operator norm is an upper bound for the *magnitude* of the eigenvalues, while the logarithmic norm is an upper bound for the *real part* of the eigenvalues. Equality is usually not attained, except in important special cases. For example, the Euclidean norms $|\cdot|_2$ and $M_2[\cdot]$ are sharp for the entire class of normal matrices.

All eigenvalues of A are thus contained in the strip $m[A] \le \operatorname{Re}\lambda \le M[A]$ (for any choice of norm). They are also contained in the annulus $\mathrm{glb}[A] \le |\lambda| \le |A|$. Further, from (2) it follows that $M[A]$ and $m[A]$ are the maximum and minimum of the Rayleigh quotient. This implies that $m[A] > 0$ generalizes and quantifies the notion of a *positive definite* matrix, while $M[A] < 0$ generalizes negative definiteness. Moreover, $M[A]$ and $m[A]$ are also the maximal and minimal real parts, respectively, of the numerical range of an operator [16].

Resolvents can also be bounded in half-planes. Thus, as a generalization of (9), one has

$$M[A] < \operatorname{Re} z \;\Rightarrow\; |(A - zI)^{-1}| < \frac{1}{\operatorname{Re} z - M[A]}.$$

A similar bound can be obtained in the half-plane $\operatorname{Re} z < m[A]$.

While the bounds above hold for all norms, some less obvious results can be obtained in Hilbert space. According to the well-known spectral theory of von Neumann, [17], if a polynomial has the property $|z| \le 1 \Rightarrow |P(z)| \le 1$, then this property can be extended to matrices and norms. Thus, if a matrix is a contraction with respect to an inner product norm, then so is $P(A)$, i.e., $|A|_H \le 1 \Rightarrow |P(A)|_H \le 1$, where the subscript H refers to the Hilbert space topology. This result also holds for *rational functions*, as well as over half-planes in $\mathbb{C}$. Thus, if R is a rational function such that $\operatorname{Re} z \le 0 \Rightarrow |R(z)| \le 1$, then $M_H[A] \le 0 \Rightarrow |R(A)|_H \le 1$.

This is of particular importance in the stability theory of Runge–Kutta methods for ordinary differential equations. When such a method is applied to the *linear test equation* $\dot{x} = \lambda x$ with step size h, the solution is advanced by a recursion of the form $x_{n+1} = R(h\lambda)x_n$, where the *stability function* $R(z)$ approximates e^z. The method is called *A-stable* if $\operatorname{Re} z \le 0 \Rightarrow |R(z)| \le 1$. It then follows that every A-stable Runge–Kutta method has the property that, when applied to a linear system $\dot{x} = Ax$,

$$M_H[A] \le 0 \;\Rightarrow\; |R(hA)|_H \le 1. \tag{11}$$

This implies that the method has stability properties similar to those of the differential equation, as both are contractive when $M_H[A] < 0$; by (6), we have

$$M_H[A] \le 0 \quad \Rightarrow \quad |\mathrm{e}^{hA}|_H \le 1. \tag{12}$$

Nonlinear Maps

The theory is easily extended to nonlinear maps, both in Banach and in Hilbert space. In Banach space, one defines the *least upper bound* (lub) and *greatest lower bound* (glb) *Lipschitz constants*, by

$$L[f] = \sup_{u \neq v} \frac{|f(u) - f(v)|}{|u - v|}\,;$$
$$l[f] = \inf_{u \neq v} \frac{|f(u) - f(v)|}{|u - v|}\,, \tag{13}$$

for $u, v \in D$, the domain of f. The lub Lipschitz constant is an *operator semi-norm* that generalizes the matrix norm: if $f = A$ is a linear map, then $L[A] = |A|$. One can then define two more functionals on D, the *lub logarithmic Lipschitz constant* and the *glb logarithmic Lipschitz constant*, by

$$M[f] = \lim_{h \to 0+} \frac{L[I + hf] - 1}{h}\,;$$
$$m[f] = \lim_{h \to 0-} \frac{L[I + hf] - 1}{h}\,. \tag{14}$$

Naturally, these definitions only apply to "bounded operators," which here correspond to Lipschitz maps. In Hilbert space, however, one can also include unbounded operators; in analogy with (2), one then defines $m_H[\cdot]$ and $M_H[\cdot]$ as the best constants such that the inequalities

$$m_H[f] \cdot |u - v|_H^2 \le \operatorname{Re} \langle u - v, f(u) - f(v) \rangle_H \le M_H[f] \cdot |u - v|_H^2 \tag{15}$$

hold for all $u, v \in D$. For Lipschitz maps, these definitions are compatible with (14), and the linear theory is fully extended to nonlinear problems. All previously listed general properties of the logarithmic norm are preserved, although attention must be paid to the domains of the operators involved. The terminology is also different. Thus, a map with $M[f] < 0$ (or $m[f] > 0$) is usually called *strongly monotone*. Such a map is one-to-one from D to $f(D)$ with a Lipschitz inverse:

$$M[f] < 0 \quad \Rightarrow \quad L[f^{-1}] \le -\frac{1}{M[f]}\,. \tag{16}$$

This extension of (9) quantifies the Browder and Minty theorem, also known as the *Uniform Monotonicity Theorem* [13].

The special bounds that could be obtained for matrices and linear operators in Hilbert space are more restricted for nonlinear maps, due to loss of commutativity. As a consequence, the result (11) does not hold in the nonlinear case without qualification. However, additional conditions can be imposed to construct Runge–Kutta methods that are contractive for problems $\dot{x} = f(x)$, with $M_H[f] \le 0$. Thus, *B-stable* Runge–Kutta methods (a subset of the A-stable methods) have this property for nonlinear systems [1].

Unbounded Operators in Hilbert Space

The use of logarithmic norms in infinite dimensional spaces is possible both in Banach and in Hilbert space. Only the latter is straightforward, but it offers adequate tools for many problems. A standard example is the parabolic reaction-diffusion equation

$$u_t = u_{xx} + g(u) \tag{17}$$

with boundary data $u(t,0) = u(t,1) = 0$. Consider functions $u, v \in H_0^1 \cap H^2 \subset L^2[0,1] = \mathcal{H}$, with the usual inner product and norm,

$$\langle u, v\rangle_{\mathcal{H}} = \int_0^1 u(x)v(x)\,\mathrm{d}x; \qquad \|u\|_{\mathcal{H}}^2 = \langle u, u\rangle_{\mathcal{H}}. \tag{18}$$

The problem (17) is then an abstract ODE $\dot{u} = f(u)$ on a Hilbert space. The logarithmic norm characterizes the stability of $u(t,\cdot)$ as $t \to \infty$, as well as the equilibrium solution, which satisfies the two-point boundary value problem

$$u'' + g(u) = 0; \qquad u(0) = u(1) = 0, \tag{19}$$

where $'$ denotes $\mathrm{d}/\mathrm{d}x$. The logarithmic norm $M_{\mathcal{H}}[\mathrm{d}^2/\mathrm{d}x^2]$ on $H_0^1 \cap H^2[0,1]$ is calculated using integration by parts,

$$\begin{aligned}\langle u, u''\rangle_{\mathcal{H}} &= -\langle u', u'\rangle_{\mathcal{H}} = -\int_0^1 |u'(x)|^2\mathrm{d}x \\ &\le -\pi^2 \int_0^1 |u(x)|^2\mathrm{d}x = -\pi^2\langle u, u\rangle_{\mathcal{H}}.\end{aligned}$$

The inequality at the center is a Sobolev inequality; it is sharp, as equality is attained for $u(x) = \sin \pi x$. Hence

$$M_{\mathcal{H}}[\mathrm{d}^2/\mathrm{d}x^2] = -\pi^2, \tag{20}$$

which quantifies that $-\mathrm{d}^2/\mathrm{d}x^2$ is *elliptic*.

As $M_{\mathcal{H}}[\cdot]$ is subadditive, $M_{\mathcal{H}}[f] = M_{\mathcal{H}}[\partial^2/\partial x^2 + g] \le M_{\mathcal{H}}[\partial^2/\partial x^2] + M_{\mathcal{H}}[g] = -\pi^2 + M_{\mathcal{H}}[g]$. Hence if the reaction term satisfies $M_{\mathcal{H}}[g] < \pi^2$ the solution $u(t,\cdot)$ of (17) is exponentially stable.

Moreover, if $M_{\mathcal{H}}[g] < \pi^2$, then $f = \mathrm{d}^2/\mathrm{d}x^2 + g$ is strongly monotone, with a Lipschitz continuous inverse on $L^2[0,1]$, implying that (19) has a unique solution, depending continuously on the data.

When the problem is discretized by the proper use of any finite difference or finite element method, the logarithmic norm of the discrete system is typically very close to that of the continuous system, provided that the inner products and norms are chosen in a compatible way. This means that one obtains similar bounds and estimates for the discrete system.

Literature

The two original, but independent, papers introducing the logarithmic norm are [3, p. 10] and [12, pp. 57–58], which also introduced the term "logarithmic norm." There are but a few surveys of the logarithmic norm and its applications. Two early surveys, including applications, are [5, 15]. The most modern one, taking a functional analytic approach, is [14], which also contains many references. Further extensions can also be found, to matrix pencils [9], and to nonlinear DAE stability [10].

Spectral bounds and resolvent behavior are dealt with at length in [16]. Bounds along the lines of [17], but for nonlinear systems, are of importance in the study of contractive methods for ODEs, see [1] for Runge–Kutta methods, and [4] for multistep methods. This also led to the study of "B-convergent" methods, in which convergence proofs were derived using only a monotonicity condition on f in Hilbert space, instead of the usual assumption of Lipschitz continuity [6, 11]. This is of particular importance for nonlinear PDE evolutions, where the contractivity and B-convergence of the implicit Euler method are used as standard proof techniques for existence and uniqueness, [2]. More recent developments for Runge–Kutta and multistep methods are found in [7, 8].

References

1. Butcher, J.C.: A stability property of implicit Runge–Kutta methods. BIT **15**, 358–361 (1975)
2. Crandall, M.G., Liggett, T.: Generation of semigroups of nonlinear transformation on general Banach spaces. Am. J. Math. **93**, 265–298 (1971)
3. Dahlquist, G.: Stability and error bounds in the numerical integration of ordinary differential equations. In: Transactions of the Royal Institute of Technology, Nr. 130, Stockholm (1959)
4. Dahlquist, G.: G-stability is equivalent to A-stability. BIT **18**, 384–401 (1978)

L

5. Desoer, C., Haneda, H.: The measure of a matrix as a tool to analyze computer algorithms for circuit analysis. IEEE Trans. Circuit Theory **19**, 480–486 (1972)
6. Frank, R., Schneid, J., Ueberhuber, C.W.: The concept of B-convergence. SIAM J. Numer. Anal. **18**, 753–780 (1981)
7. Hansen, E.: Convergence of multistep time discretizations of nonlinear dissipative evolution equations. SIAM J. Numer. Anal. **44**, 55–65 (2006)
8. Hansen, E.: Runge-Kutta time discretizations of nonlinear dissipative evolution equations. Math. Comp. **75**, 631–640 (2006)
9. Higueras, I., García–Celayeta, B.: Logarithmic norms for matrix pencils. SIAM J. Matrix Anal. **20**, 646–666 (1999)
10. Higueras, I., Söderlind, G.: Logarithmic norms and nonlinear DAE stability. BIT **42**, 823–841 (2002)
11. Kraaijevanger, J.F.B.M.: B-convergence of the implicit midpoint rule and the trapezoidal rule. BIT **25**, 652–666 (1985)
12. Lozinskii, S.M.: Error estimates for the numerical integration of ordinary differential equations, part I. Izv. Vyss. Uceb. Zaved Matematika **6**, 52–90 (1958) (In Russian)
13. Ortega, J.M., Rheinboldt, W.C.: Iterative Solution of Nonlinear Equations in Several Variables. Academic, New York (1970)
14. Söderlind, G.: The logarithmic norm. History and modern theory. BIT **46**, 631–652 (2006)
15. Ström, T.: On logarithmic norms. SIAM J. Numer. Anal. **2**, 741–753 (1975)
16. Trefethen, L.N., Embree, M.: Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators. Princeton University Press, Princeton (2005)
17. von Neumann, J.: Eine Spektraltheorie für allgemeine Operatoren eines unitären Raumes. Math. Nachr. **4**, 258–281 (1951)

Logical Characterizations of Complexity Classes

Martin Grohe
Department of Computer Science, RWTH Aachen University, Aachen, Germany

Mathematics Subject Classification

68Q19; 68Q15; 03C13

Short Definition

The complexity of a computational problem, originally defined in terms of the computational resources required to solve the problem, can be characterized in terms of the language resources required to describe the problem in a logical system. This yields logical characterizations of all standard complexity classes.

Description

It was realized from the beginnings of computability theory in the 1930s that there is a close connection between logic and computation. Indeed, various degrees of computability have natural characterizations in terms of logical definability. For example, the recursively enumerable sets of natural numbers are precisely the sets definable by an existential formula of first-order predicate logic in the language of arithmetic.

Descriptive Complexity Theory

Descriptive complexity may be viewed as a natural continuation of these results of computability theory in the realm of computational complexity. It provides characterizations of most standard complexity classes in terms of logical definability. Arguably the most important of these characterizations are given by the following two theorems:

Fagin's Theorem [7]. *A property of finite structures is decidable in nondeterministic polynomial time* NP *if and only if it is definable in existential second-order logic* ∃SO. *(Short:* ∃SO *captures* NP.*)*

Immerman-Vardi Theorem [12, 14]. *A property of ordered finite structures is decidable in polynomial time* P *if and only if it is definable in least fixed-point logic* LFP. *(Short:* LFP *captures* P *on ordered structures.)*

To explain these two theorems, we need to review the basic framework of computational complexity theory and some logic. Complexity classes are usually defined as classes of problems that can be solved with restricted resources such as time or space. To turn this into a precise mathematical definition, we need to fix a machine model and a coding scheme for representing computational problems as inputs. Typically, multitape Turing machines are used as machine model. Without much loss of generality, we can focus on decision problems (i.e., problems with a yes/no answer) and represent them by languages over the binary alphabet $\{0, 1\}$, i.e., as sets of strings of zeroes and ones. Obviously, complexity classes defined this way depend on both the machine model and the representation scheme,

but fortunately most classes are robust enough so that they end up the same for any "reasonable" machine model and representation.

Yet the instances of most computational problems are not naturally modeled as strings over a finite alphabet, but rather by richer mathematical structures. For example, instances of a network connectivity problem are naturally modeled as directed graphs and so are the instances of many combinatorial optimization problems. Boolean circuits can be modeled by labeled directed graphs. The standard relational database model represents databases by a collection of finite relations, i.e., a finite relational structure. Of course the instances of some problems, such as problems on the natural numbers (in binary representation) or pattern matching problems, are most naturally described by finite strings, but strings can also be viewed as specific finite structures. If we adopt finite structures as flexible models of the instances of computational problems, then decision problems become properties of finite structures or, equivalently, classes of finite structures closed under isomorphism. This is the point of view taken in descriptive complexity theory.

Logics express, or define, properties of structures. The logics considered in descriptive complexity theory are extensions of first-order predicate logic FO. Instead of going through formal definitions, we give three examples of logics and graph properties defined in these logics.

*Example 1 (*First-Order Logic*)* The diameter of a graph is the maximum distance between any two vertices of the graph. The following sentence of first-order logic in the language of graphs defines the property of a graph having diameter at most 2:

$$\forall x \forall y \Big(x = y \vee Exy \vee \exists z \big(Exz \wedge Ezy \big) \Big).$$

Here the variables x, y, z range over the vertices of a graph, and Exy expresses that the vertices interpreting x, y are adjacent.

It has turned out that first-order logic is too weak to express most properties that are interesting from a computational point of view. Second-order logic SO is much more powerful; actually it is too powerful to stay in the realm of efficient computation. Hence various fragments of SO are studied in the context of descriptive complexity theory. In SO, we not only have "individual variables" ranging over the vertices of a graph but also "set variables" ranging over sets of vertices and, more generally, "relation variables" ranging over relations between vertices. Existential second-order logic ∃SO is the fragment of SO consisting of all formulas that only use existential quantification over set and relation variables and where no existential quantifier binding a relation variable appears in the scope of a negation symbol.

*Example 2 (*Existential Second-Order Logic*)* A graph is 3-colorable if its vertices can be colored with three colors in such a way that no two adjacent vertices get the same color. The following sentence of existential second-order logic defines the property of a graph being 3-colorable:

$$\begin{aligned} \exists R \exists B \exists G \Big(& \forall x \big(Rx \vee Bx \vee Gx \big) \\ & \wedge \forall x \forall y \big(Exy \rightarrow \big(\neg(Rx \wedge Ry) \wedge \neg(Bx \wedge By) \\ & \wedge \neg(Gx \wedge Gy) \big) \big) \Big). \end{aligned}$$

Here the variables R, B, G are set variables representing the three colors, and x, y are individual variables. Rx expresses that the vertex interpreting x is contained in the set interpreting R.

Fixed-point logics are extensions of FO with a more algorithmic flavor than SO. They allow it to formalize inductive definitions, as illustrated by the following example.

*Example 3 (*Least Fixed-Point Logic*)* Suppose we want to define the transitive closure T of the edge relation of a graph $G = (V, E)$. It admits the following inductive definition: We let $T_1 := E$, and for all i we let T_{i+1} be the set of all pairs (u, v) of vertices such that there is a vertex w with $(v, w) \in T_i$ and $(w, u) \in T_i$. Then T is the union of all the T_i. Equivalently, we may define T as the least fixed point of the (monotone) operator

$$X \mapsto \Big\{ (v, w) \Big| (v, w) \in E \vee \exists z \big((v, z) \in X \wedge (z, w) \in X \big) \Big\}.$$

In least fixed-point logic LFP, we can form a formula

$$\mathrm{lfp}\,\big(Xxy \leftarrow Exy \vee \exists z (Xxz \wedge Xzy) \big)(v, w)$$

L

to define this least fixed point (and thus the transitive closure). If we call this formula $\psi(v, w)$, then the LFP-sentence $\forall v \forall w\big(v = w \vee \psi(v, w)\big)$ defines connectedness of (undirected) graphs.

To connect the properties of structures defined in our logics with complexity classes, we need to fix an encoding scheme for structures. It is common to use a generalization of the adjacency-matrix encoding of graphs to encode structures by binary strings. Unfortunately, a graph has different adjacency matrices, obtained by associating the vertices with the rows and columns of the matrix in different orders, and among these there is no distinguished canonical one that we could use as "the" encoding of the structure. This observation generalizes to arbitrary structures. Only if a structure B comes with a linear order of its elements, that is, it has a distinguished binary relation $\leq^B$ that is a linear order of its elements, then we can fix a canonical binary string $\langle B \rangle$ encoding B. We call such structures *ordered structures*, or we say that they have a *built-in order*. With each property $\mathcal{Q}$ of ordered structures, we associate the language $\mathcal{L}(\mathcal{Q}) := \{\langle B \rangle \mid B \text{ has property } \mathcal{Q}\}$. With a structure A without built-in order, we can only associate a language $\mathcal{L}(A)$ consisting of all encodings of A. Equivalently, we may view $\mathcal{L}(A)$ as the set of all strings $\langle B \rangle$ for all ordered expansions B of A. For a property $\mathcal{P}$ of structures, we let $\mathcal{L}(\mathcal{P})$ be the union of all $\mathcal{L}(A)$ for structures A that have property $\mathcal{P}$. Now we say that a logic L *captures* a complexity class K if for each property $\mathcal{P}$ of structures, there is an L-sentence that defines $\mathcal{P}$ if and only if $\mathcal{L}(\mathcal{P}) \in$ K. We say that L *captures* K *on ordered structures* if for each property $\mathcal{Q}$ of ordered structures, there is an L-sentence that defines $\mathcal{Q}$ if and only if $\mathcal{L}(\mathcal{Q}) \in$ K.

Fagin's Theorem and the Immerman-Vardi Theorem give logics capturing the complexity classes NP and P, respectively, the latter only on ordered structures. There are similar logical characterizations for most other complexity classes (for background and references, we refer the reader to the textbooks [6, 8, 13]). For the standard space complexity classes, we have the following characterizations: deterministic transitive closure logic DTC captures L ("logarithmic space") on ordered structures, transitive closure logic TC captures NL ("nondeterministic logarithmic space") on ordered structures, and partial fixed-point logic PFP captures PSPACE ("polynomial space") on ordered structures. While these characterizations use various extensions of first-order logic by fixed-point operators or similar "generalized quantifiers," we also have characterizations of various complexity classes by restrictions and extensions of second-order logic: second-order logic SO captures PH (the "polynomial hierarchy"). The "Krom fragment" of second-order logic captures NL on ordered structures, and the "Horn fragment" of second-order logic captures P on ordered structures. The extension of second-order logic with a (second-order) transitive closure operator captures PSPACE. There are also logical characterizations of complexity below L, but in addition to a built-in order, these require structures to have *built-in arithmetic*. For example, first-order logic FO captures dlogtime-uniform AC^0 on structures with built-in arithmetic.

Note that for the class P and smaller classes such as L and NL we only have logical characterizations on ordered structures. Indeed, it is a major open problem whether there are logical characterizations for these classes on arbitrary (not necessarily ordered) structures. Only partial results characterizing P on restricted classes of structures are known (the most powerful in [9]).

Function Algebras and Implicit Computational Complexity

An alternative way of characterizing complexity classes is inspired by the characterizations of the computable functions as recursive functions and by the λ-calculus. The idea is to describe the functions in a complexity class as an algebra of functions. We extend complexity classes K to classes of functions on binary strings and speak of K-functions. We usually think of K-functions as functions on the natural numbers (via a binary encoding). The classical result in this area is Cobham's characterization of the polynomial time computable functions using the following restricted version of primitive recursion: A $(k+1)$-ary function f on the natural numbers is defined from functions g, h_0, h_1, b by *bounded primitive recursion on notation* if for all $\bar{x}$ we have $f(\bar{x}, 0) = g(\bar{x})$ and $f(\bar{x}, 2y + i) = h_i(\bar{x}, y, f(\bar{x}, y))$ for $i = 0, y > 0$ and $i = 1, y \geq 0$, provided that $f(\bar{x}, y) \leq b(\bar{x}, y)$ for all $\bar{x}, y$. The addition "on notation" refers to the fact that this definition is most naturally understood if one thinks of natural numbers in binary notation.

Cobham's Theorem [4]. *The class of* P*-functions is the closure of the basic functions* $x \mapsto 0$ *("constant 0"),* $(x_1, \ldots, x_k) \mapsto x_i$ *for all* $i \leq k$ *("projections"),* $x \mapsto 2x$ *and* $x \mapsto 2x + 1$ *("successor functions"), and* $(x, y) \mapsto 2^{|x| \cdot |y|}$ *("smash function"), where* $|x|$ *denotes the length of the binary representation of* x*, under composition and bounded primitive recursion on notation.*

Similar characterizations are known for other complexity classes.

What is slightly unsatisfactory about Cobham's characterization of the P-functions is the explicit time bound b in the bounded primitive recursion scheme. Bellantoni and Cook [1] devised a refined primitive recursion scheme that distinguishes between different types of variables and how they may be used and characterize the P-functions without an explicit time bound. This is the starting point of the area of "implicit computational complexity" ([10] is a survey). While Bellantoni and Cook's recursion scheme is still fairly restrictive, in the sense that the type system excludes natural definitions of P-functions by primitive recursion, subsequently researchers have developed a variety of full (mostly functional) programming languages with very elaborate type systems guaranteeing that precisely the K-functions (for many of the standard complexity classes K) have programs in this language. The best known of these is Hofmann's functional language for the P-functions with a type system incorporating ideas from linear logic [11].

Proof Theory and Bounded Arithmetic

There is yet another line of logical characterizations of complexity classes. It is based on provability in formal system rather than just definability. Again, these characterizations have precursors in computability theory, in particular the characterization of the primitive recursive functions as precisely those functions that are Σ_1-definable in the fragment $\mathrm{i}\Sigma_1$ of Peano arithmetic.

The setup is fairly complicated, and we will only be able to scratch the surface; for a thorough treatment, we refer the reader to the survey [3] and the textbook [5]. Our basic logic is first-order logic in the language of arithmetic, consisting of the standard symbols $\leq$ (order), $+$ (addition), $\cdot$ (multiplication), 0, 1 (constants 0 and 1), and possibly additional function symbols. In the *standard model of arithmetic* N, all these symbols get their standard interpretations over the natural numbers. A *theory* is a set of first-order sentences that is closed under logical consequence. For example, $\mathsf{Th}(N)$ is the set of all sentences that are true in the standard model N. It follows from Gödel's First Incompleteness Theorem that $\mathsf{Th}(N)$ has no decidable axiom system. A decidable, yet still very powerful, theory that contained $\mathsf{Th}(N)$ is Peano arithmetic PA. It is axiomatized by a short list of basic axioms making sure that the basic symbols are interpreted right together with induction axioms of the form $\big(\phi(0) \wedge \forall x(\phi(x) \rightarrow \phi(x+1))\big) \rightarrow \forall x \phi(x)$ for all first-order formulas ϕ. Here, we are interested in fragments $\mathrm{i}\Phi$ of PA obtained by restricting the induction axioms to formulas $\phi \in \Phi$ for sets Φ of first-order formulas. Δ_0 denotes the set of all bounded first-order formulas, that is, formulas where all quantifications are of the form $\exists x \leq t$ or $\forall x \leq t$ for some term t that does not contain the variable x. Almost everything relevant for complexity theory takes place within Δ_0, but let us mention that Σ_1 is the set of all first-order formulas of the form $\exists x \phi$, where ϕ is a Δ_0-formula.

We say that a function f on the natural numbers is *definable in a theory* T if there is a formula $\phi(x, y)$ such that the theory T proves that for all x there is exactly one y such that $\phi(x, y)$ and for all natural numbers m, n the standard model N satisfies $\phi(m, n)$ if and only if $f(m) = n$. For example, it can be shown that the functions in the linear time hierarchy LTH are precisely the functions that are Δ_0-definable in the theory $\mathrm{i}\Delta_0$.

To characterize the classes P and NP and the other classes of the polynomial hierarchy, Buss introduced a hierarchy of very weak arithmetic theories S^i_2. They are obtained by even restricting the use of bounded quantifiers in Δ_0-formulas, defining a hierarchy of Σ^b_i-formulas within Δ_0 but at the same time using an extended language that also contains functions symbols like # (for the "smash" function $(x, y) \mapsto 2^{|x| \cdot |y|}$) and $|\ |$ (for the binary length).

Buss's Theorem [2]. *For all* $i \geq 1$*, the functions* Σ^b_i*-definable in* S^i_2 *are precisely the* Σ^P_{i-1}*-functions, where* $\Sigma^P_0 = \mathsf{P}$*,* $\Sigma^P_1 = \mathsf{NP}$*, and* Σ^P_i *are the* i*th level of the polynomial hierarchy.*

References

1. Bellantoni, S., Cook, S.: A new recursion-theoretic characterization of the polytime functions. Comput. Complex. **2**, 97–110 (1992)
2. Buss, S.: Bounded Arithmetic. Bibliopolis, Napoli (1986)
3. Buss, S.: First-order proof theory of arithmetic. In: Buss, S. (ed.) Handbook of Proof Theory, pp. 79–147. Elsevier, New York (1998)
4. Cobham, A.: The intrinsic computational difficulty of functions. In: Proceedings of the International Conference on Logic, Methodology, and Philosophy of Science, pp. 24–30. North-Holland, Amsterdam (1962)
5. Cook, S., Nguyen, P.: Logical Foundations of Proof Complexity. Perspectives in Logic. Cambridge University Press, Cambridge/New York (2010)
6. Ebbinghaus, H.-D., Flum, J.: Finite Model Theory. Springer, Berlin/New York (1995)
7. Fagin, R.: Generalized first–order spectra and polynomial–time recognizable sets. In: Karp, R. (ed.) Complexity of Computation, SIAM-AMS Proceedings, New York, vol. 7, pp. 43–73 (1974)
8. Grädel, E., Kolaitis, P., Libkin, L., Marx, M., Spencer, J., Vardi, M., Venema, Y., Weinstein, S.: Finite Model Theory and Its Applications. Springer, Berlin/New York (2007)
9. Grohe, M.: From polynomial time queries to graph structure theory. Commun. ACM **54**(6), 104–112 (2011)
10. Hofmann, M.: Programming languages capturing complexity classes. ACM SIGACT News **31**(1), 31–42 (2000)
11. Hofmann, M.: Linear types and non-size-increasing polynomial time computation. Inf. Comput. **183**, 57–85 (2003)
12. Immerman, N.: Relational queries computable in polynomial time (extended abstract). In: Proceedings of the 14th ACM Symposium on Theory of Computing, San Francisco, pp. 147–152 (1982)
13. Immerman, N.: Descriptive Complexity. Springer, New York (1999)
14. Vardi, M.: The complexity of relational query languages. In: Proceedings of the 14th ACM Symposium on Theory of Computing, San Francisco, pp. 137–146 (1982)

Lyapunov Exponents: Computation

Luca Dieci[1] and Erik S. Van Vleck[2]
[1]School of Mathematics, Georgia Institute of Technology, Atlanta, GA, USA
[2]Department of Mathematics, University of Kansas, Lawrence, KS, USA

History and Scope

In 1892, in his doctoral thesis *The general problem of the stability of motion* (reprinted in its original form in [33]), Lyapunov introduced several groundbreaking concepts to investigate stability in differential equations. These are collectively known as Lyapunov Stability Theory. Lyapunov was concerned with the asymptotic stability of solutions with respect to perturbations of initial data. Among other techniques (e.g., what are now known as first and second Lyapunov methods), he introduced a new tool to analyze the stability of solutions of linear time-varying systems of differential equations, the so-called characteristic numbers, now commonly and appropriately called *Lyapunov exponents*.

Simply put, these characteristic numbers play the role that the (real parts of the) eigenvalues play for time-invariant linear systems. Lyapunov considered the n-dimensional linear system

$$\dot{x} = A(t)x, t \geq 0\,, \tag{1}$$

where A is continuous and bounded: $\sup_t \|A(t)\| < \infty$. He showed that "if all characteristic numbers (see below for their definition) of (1) are negative, then the zero solution of (1) is asymptotically (in fact, exponentially) stable." He further proved an important characterization of stability relative to the perturbed linear system

$$\dot{x} = A(t)x + f(t,x)\,, \tag{2}$$

where $f(t,0) = 0$, so that $x = 0$ is a solution of (2), and further $f(t,x)$ is assumed to be "small" near $x = 0$ (this situation is what one expects from a linearized analysis about a bounded solution trajectory). Relative to (2), Lyapunov proved that "if the linear system (1) is *regular*, and all its characteristic numbers are negative, then the zero solution of (2) is asymptotically stable." About 30 years later, it was shown by Perron in [38] that the assumption of regularity cannot generally be removed.

Definition

We refer to the monograph [1] for a comprehensive definition of Lyapunov exponents, regularity, and so forth. Here, we simply recall some of the key concepts.

Consider (1) and let us stress that the matrix function $A(t)$ may be either given or obtained as the linearization about the solution of a nonlinear differential equation; e.g., $\dot{y} = f(y)$ and $A(t) = Df(y(t))$ (note that in this case, in general, A will depend on the initial condition used for the nonlinear problem). Now, let X

be a fundamental matrix solution of (1), and consider the quantities

$$\lambda_i = \limsup_{t\to\infty} \frac{1}{t} \ln ||X(t)e_i|| \ , \ i = 1, \ldots, n, \qquad (3)$$

where e_i denotes the ith standard unit vector, $i = 1, \ldots, n$. When $\sum_{i=1}^{n} \lambda_i$ is minimized with respect to all possible fundamental matrix solutions, then the λ_i are called the characteristic numbers, or Lyapunov exponents, of the system. It is customary to consider them ordered as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Similar definitions can be given for $t \to -\infty$ and/or with lim inf replacing the lim sup, but the description above is the prevailing one. An important consequence of *regularity* of a given system is that in (3) one has limits instead of lim sups.

More Recent Theory

Given that the condition of regularity is not easy to verify for a given system, it was unclear what practical use one was going to make of the Lyapunov exponents in order to study stability of a trajectory. Moreover, even assuming that the system is regular, it is effectively impossible to get a handle on the Lyapunov exponents except through their numerical approximation. It then becomes imperative to have some comfort that what one is trying to approximate is robust; in other words, it is the Lyapunov exponents themselves that will need to be stable with respect to perturbations of the function A in (1). Unfortunately, regularity is not sufficient for this purpose.

Major theoretical advances to resolve the two concerns above took place in the late 1960s, thanks to the work of Oseledec and Millionshchikov (e.g., see [36] and [34]). Oseledec was concerned with stability of trajectories on a (bounded) attractor, on which one has an invariant measure. In this case, Oseledec's *Multiplicative Ergodic Theorem* validates regularity for a broad class of linearized systems; the precise statement of this theorem is rather technical, but its practical impact is that (with respect to the invariant measure) almost all trajectories of the nonlinear system will give rise to a regular linearized problem. Millionshchikov introduced the concept of *integral separation*, which is the condition needed for stability of the Lyapunov exponents with respect to perturbations in the coefficient matrix, and further gave important results on the prevalence of this property within the class of linear systems.

Further Uses of Lyapunov Exponents

Lyapunov exponents found an incredible range of applicability in several contexts, and both theory and computational methods have been further extended to discrete dynamical systems, maps, time series, etc. In particular:

(i) The largest Lyapunov exponent of (2), λ_1, characterizes the rate of separation of trajectories (with infinitesimally close initial conditions). For this reason, a positive value of λ_1 (coupled with compactness of the phase space) is routinely taken as an indication that the system is *chaotic* (see [37]).

(ii) Lyapunov exponents are used to estimate *dimension* of attractors through the Kaplan-Yorke formula (Lyapunov dimension):

$$\mathrm{Dim}_L = k + (\lambda_1 + \lambda_2 + \cdots + \lambda_k)/|\lambda_{k+1}|$$

where k is the largest index i such that $\lambda_1 + \lambda_2 + \cdots + \lambda_i > 0$. See [31] for the original derivation of the formula and [9] for its application to the 2-d Navier-Stokes equation.

(iii) The sum of all the positive Lyapunov exponents is used to estimate the entropy of a dynamical system (see [3]).

(iv) Lyapunov exponents have also been used to characterize persistence and degree of smoothness of invariant manifolds (see [26] and see [12] for a numerical study).

(v) Lyapunov exponents have even been used in studies of piecewise-smooth differential equations, where a formal linearized problem as in (1) does not even exist (see [27, 35]).

(vi) Finally, there has been growing interest also in approximating bases for the *growth directions* associated to the Lyapunov exponents. In particular, there is interest in obtaining representations for the stable (and unstable) subspaces of (1) and in their use to ascertain stability of traveling waves. For example, see [23, 39].

Factorization Techniques

Many of the applications listed above are related to nonlinear problems, which in itself is witness

to the power of linearized analysis based on the Lyapunov exponents. Still, the computational task of approximating some or all of the Lyapunov exponents for dynamical systems defined by the flow of a differential equation is ultimately related to the linear problem (1), and we will thus focus on this linear problem.

Techniques for numerical approximation of Lyapunov exponents are based upon smooth matrix factorizations of fundamental matrix solutions X, to bring it into a form from which it is easier to extract the Lyapunov exponents. In practice, two techniques have been studied: based on the QR factorization of X and based on the SVD (singular value decomposition) of X. Although these techniques have been adapted to the case of incomplete decompositions (useful when only a few Lyapunov exponents are needed) or to problems with Hamiltonian structure, we only describe them in the general case when the entire set of Lyapunov exponents is sought, the problem at hand has no particular structure, and the system is regular. For extensions, see the references.

QR Methods

The idea of QR methods is to seek the factorization of a fundamental matrix solution as $X(t) = Q(t)R(t)$, for all t, where Q is an orthogonal matrix valued function and R is an upper triangular matrix valued function with positive diagonal entries. The validity of this factorization has been known since Perron [38] and Diliberto [25], and numerical techniques based upon the QR factorization date back at least to [4].

QR techniques come in two flavors, continuous and discrete, and methods for quantifying the error in approximation of Lyapunov exponents have been developed in both cases (see [15–17, 21, 40]).

Continuous QR

Upon differentiating the relation $X = QR$ and using (1), we have

$$AQR = Q\dot{R} + \dot{Q}R \quad \text{or} \quad \dot{Q} = AQ - QB\,, \tag{4}$$

where $\dot{R} = BR$; hence, B must be upper triangular. Now, let us formally set $S = Q^T\dot{Q}$ and note that since Q is orthogonal then S must be skew symmetric. Now, from $B = Q^TAQ - Q^T\dot{Q}$ it is easy to determine at once the strictly lower triangular part of S (and from this, all of it) and the entries of B. To sum up, we have two differential equations, for Q and for R. Given $X(0) = Q_0R_0$, we have

$$\dot{Q} = QS(Q,A)\,, \quad Q(0) = Q_0\,, \tag{5}$$

$$\dot{R} = B(t)R\,, \quad R(0) = R_0\,,$$
$$B := Q^TAQ - S(Q,A) \tag{6}$$

The diagonal entries of R are used to retrieve the exponents:

$$\lambda_i = \lim_{t\to\infty}\frac{1}{t}\int_0^t (Q^T(s)A(s)Q(s))_{ii}\,ds\,,\ i = 1,\ldots,n. \tag{7}$$

A unit upper triangular representation for the growth directions may be further determined by $\lim_{t\to\infty}\mathrm{diag}(R^{-1}(t))R(t)$ (see [13, 22, 23]).

Discrete QR

Here one seeks the QR factorization of the fundamental matrix X at discrete points $0 = t_0 < t_1 < \cdots < t_k < \cdots$, where $t_k = t_{k-1} + h_k$, $h_k \geq \hat{h} > 0$. Let $X_0 = Q_0R_0$, and suppose we seek the QR factorization of $X(t_{k+1})$. For $j = 0,\ldots,k$, progressively define $Z_{j+1}(t) = X(t,t_j)Q_j$, where $X(t,t_j)$ solves (1) for $t \geq t_j$, $X(t_j,t_j) = I$, and Z_{j+1} is the solution of

$$\begin{cases}\dot{Z}_{j+1} = A(t)Z_{j+1}\,, & t_j \leq t \leq t_{j+1}\\ Z_{j+1}(t_j) = Q_j\,.\end{cases} \tag{8}$$

Update the QR factorization as

$$Z_{j+1}(t_{j+1}) = Q_{j+1}R_{j+1}\,, \tag{9}$$

and finally observe that

$$X(t_{k+1}) = Q_{k+1}\,[R_{k+1}R_k\cdots R_1R_0] \tag{10}$$

is the QR factorization of $X(t_{k+1})$. The Lyapunov exponents are obtained from the relation

$$\lim_{k\to\infty}\frac{1}{t_k}\sum_{j=0}^{k}\log(R_j)_{ii}\,, i = 1,\ldots,n\,. \tag{11}$$

SVD Methods

Here one seeks to compute the SVD of X: $X(t) = U(t)\Sigma(t)V^T(t)$, for all t, where U and V are orthogonal and $\Sigma = \mathrm{diag}(\sigma_i,\ i = 1\ldots,n)$, with

$\sigma_1(t) \geq \sigma_2(t) \geq \cdots \geq \sigma_n(t)$. If the singular values are distinct, the following differential equations U, V, and Σ hold. Letting $G = U^T AU$, they are

$$\dot{U} = UH, \quad \dot{V}^T = -KV^T, \quad \dot{\Sigma} = D\Sigma, \qquad (12)$$

where $D = \text{diag}(G)$, $H^T = -H$, and $K^T = -K$, and for $i \neq j$,

$$H_{ij} = \frac{G_{ij}\sigma_j^2 + G_{ji}\sigma_i^2}{\sigma_j^2 - \sigma_i^2}, \quad K_{ij} = \frac{(G_{ij} + G_{ji})\sigma_i\sigma_j}{\sigma_j^2 - \sigma_i^2}. \qquad (13)$$

From the SVD of X, the Lyapunov exponents may be obtained as

$$\lim_{t\to\infty} \frac{1}{t} \ln \sigma_i(t). \qquad (14)$$

Finally, an orthogonal representation for the growth directions may be determined by $\lim_{t\to\infty} V(t)$ (see [10, 13, 22, 23]).

Numerical Implementation

Although algorithms based upon the above techniques appear deceivingly simple to implement, much care must be exercised in making sure that they perform as one would expect them to. (For example, in the continuous QR and SVD techniques, it is mandatory to maintain the factors Q, U, and V orthogonal.) `Fortran` software codes for approximating Lyapunov exponents of linear and nonlinear problems have been developed and tested extensively and provide a combined state of the knowledge insofar as numerical methods suited for this specific task. See [14, 20, 24].

Acknowledgements Erik Van Vleck acknowledges support from NSF grant DMS-1115408.

References

1. Adrianova, L.Ya.: Introduction to Linear Systems of Differential Equations (Trans. from the Russian by Peter Zhevandrov). Translations of Mathematical Monographs, vol. 146, pp. x+204. American Mathematical Society, Providence (1995)
2. Aston, P.J., Dellnitz, M.: The computation of Lyapunov exponents via spatial integration with application to blowout bifurcations. Comput. Methods Appl. Mech. Eng. **170**, 223–237 (1999)
3. Barreira, L., Pesin, Y.: Lyapunov Exponents and Smooth Ergodic Theory. University Lecture Series, vol. 23. American Mathematical Society, Providence (2001)
4. Benettin, G., Galgani, L., Giorgilli, A., Strelcyn, J.-M.: Lyapunov exponents for smooth dynamical systems and for Hamiltonian systems: a method for computing all of them. Part 1: Theory, and . . . Part 2: Numerical applications. Meccanica **15**, 9–20, 21–30 (1980)
5. Bridges, T., Reich, S.: Computing Lyapunov exponents on a Stiefel manifold. Physica D **156**, 219–238 (2001)
6. Bylov, B.F., Vinograd, R.E., Grobman, D.M., Nemyckii, V.V.: The Theory of Lyapunov Exponents and Its Applications to Problems of Stability. Nauka, Moscow (1966)
7. Calvo, M.P., Iserles, A., Zanna, A.: Numerical solution of isospectral flows. Math. Comput. **66**(220), 1461–1486 (1997)
8. Christiansen, F., Rugh, H.H.: Computing Lyapunov spectra with continuous Gram-Schmidt orthonormalization. Nonlinearity **10**, 1063–1072 (1997)
9. Constantin, P., Foias, C.: Global Lyapunov exponents, Kaplan-Yorke formulas and the dimension of the attractors for 2D Navier-Stokes equations. Commun. Pure Appl. Math. **38**, 1–27 (1985)
10. Dieci, L., Elia, C.: The singular value decomposition to approximate spectra of dynamical systems. Theoretical aspects. J. Differ. Equ. **230**(2), 502–531 (2006)
11. Dieci, L., Lopez, L.: Smooth SVD on symplectic group and Lyapunov exponents approximation. CALCOLO **43**(1), 1–15 (2006)
12. Dieci, L., Lorenz, J.: Lyapunov type numbers and torus breakdown: numerical aspects and a case study. Numer. Algorithms **14**, 79–102 (1997)
13. Dieci, L., Van Vleck, E.S.: Lyapunov spectral intervals: theory and computation. SIAM J. Numer. Anal. **40**(2), 516–542 (2002)
14. Dieci, L., Van Vleck, E.S.: `LESLIS` and `LESLIL`: codes for approximating Lyapunov exponents of linear systems. Technical report, Georgia Institute of Technology. http://www.math.gatech.edu/~dieci (2004)
15. Dieci, L., Van Vleck, E.S.: On the error in computing Lyapunov exponents by QR methods. Numer. Math. **101**(4), 619–642 (2005)
16. Dieci, L., Van Vleck, E.S.: Perturbation theory for approximation of Lyapunov exponents by QR methods. J. Dyn. Differ. Equ. **18**(3), 815–840 (2006)
17. Dieci, L., Van Vleck, E.S.: On the error in QR integration. SIAM J. Numer. Anal. **46**(3), 1166–1189 (2008)
18. Dieci, L., Russell, R.D., Van Vleck, E.S.: Unitary integrators and applications to continuous orthonormalization techniques. SIAM J. Numer. Anal. **31**(1), 261–281 (1994)
19. Dieci, L., Russell, R.D., Van Vleck, E.S.: On the computation of Lyapunov exponents for continuous dynamical systems. SIAM J. Numer. Anal. **34**, 402–423 (1997)
20. Dieci, L., Jolly, M., Van Vleck, E.S.: `LESNLS` and `LESNLL`: codes for approximating Lyapunov exponents of nonlinear systems. Technical report, Georgia Institute of Technology. http://www.math.gatech.edu/~dieci (2005)
21. Dieci, L., Jolly, M., Rosa, R., Van Vleck, E.: Error on approximation of Lyapunov exponents on inertial manifolds: the Kuramoto-Sivashinsky equation. J. Discret. Contin. Dyn. Syst. Ser. B **9**(3–4), 555–580 (2008)

22. Dieci, L., Elia, C., Van Vleck, E.S.: Exponential dichotomy on the real line: SVD and QR methods. J. Differ. Equ. **248**(2), 287–308 (2010)
23. Dieci, L., Elia, C., Van Vleck, E.S.: Detecting exponential dichotomy on the real line: SVD and QR algorithms. BIT **51**(3), 555–579 (2011)
24. Dieci, L., Jolly, M.S., Van Vleck, E.S.: Numerical techniques for approximating Lyapunov exponents and their implementation. ASME J. Comput. Nonlinear Dyn. **6**, 011003–1–7 (2011)
25. Diliberto, S.P.: On systems of ordinary differential equations. In: Lefschetz, S. (ed.) Contributions to the Theory of Nonlinear Oscillations. Annals of Mathematics Studies, vol. 20, pp. 1–38. Princeton University Press, Princeton (1950)
26. Fenichel, N.: Persistence and smoothness of invariant manifolds for flows. Indiana Univ. Math. J. **21**, 193–226 (1971)
27. Galvanetto, U.: Numerical computation of Lyapunov exponents in discontinuous maps implicitly defined. Comput. Phys. Commun. **131**, 1–9 (2000)
28. Geist, K., Parlitz, U., Lauterborn, W.: Comparison of different methods for computing Lyapunov exponents. Prog. Theor. Phys. **83**, 875–893 (1990)
29. Goldhirsch, I., Sulem, P.L., Orszag, S.A.: Stability and Lyapunov stability of dynamical systems: a differential approach and a numerical method. Physica D **27**, 311–337 (1987)
30. Greene, J.M., Kim, J.-S.: The calculation of Lyapunov spectra. Physica D **24**, 213–225 (1987)
31. Kaplan, J.L., Yorke, J.A.: Chaotic behavior of multidimensional difference equations. In: Peitgen, H.-O., Walter, H.-O. (eds.) Functional Differential Equations and Approximations of Fixed Points. Lecture Notes in Mathematics, vol. 730. Springer, Berlin (1979)
32. Leimkuhler, B.J., Van Vleck, E.S.: Orthosymplectic integration of linear Hamiltonian systems. Numer. Math. **77**(2), 269–282 (1997)
33. Lyapunov, A.: Problém Géneral de la Stabilité du Mouvement. Int. J. Control **53**, 531–773 (1992)
34. Millionshchikov, V.M.: Systems with integral division are everywhere dense in the set of all linear systems of differential equations. Differ. Uravn. **5**, 1167–1170 (1969)
35. Müller, P.: Calculation of Lyapunov exponents for dynamic systems with discontinuities. Chaos Solitons Fractals **5**, 1671–1681 (1995)
36. Oseledec, V.I.: A multiplicative ergodic theorem. Lyapunov characteristic numbers for dynamical systems. Trans. Mosc. Math. Soc. **19**, 197–231 (1968)
37. Ott, E.: Chaos in Dynamical Systems, 2nd edn. Cambridge University Press, Cambridge (2002)
38. Perron, O.: Die Ordnungszahlen Linearer Differentialgleichungssysteme. Math. Z. **31**, 748–766 (1930)
39. Sandstede, B.: Stability of travelling waves. In: Hasselblatt, B., Katok, A.B. (eds.) Handbook of Dynamical Systems, vol. 2, pp. 983–1055. North-Holland, Amsterdam (2002)
40. Van Vleck, E.S.: On the error in the product QR decomposition. SIAM J. Matrix Anal. Appl. **31**(4), 1775–1791 (2009/2010)
41. Wiesel, W.E.: Continuous-time algorithm for Lyapunov exponents: Part 1, and Part 2. Phys. Rev. E **47**, 3686–3697 (1993)