AAPT PHYSICS EDUCATION
AMERICAN JOURNAL of PHYSICS

Of pendulums, polymers, and robots: Computational mechanics with constraints

Franz J. Vesely

Citation: Am. J. Phys. **81**, 537 (2013); doi: 10.1119/1.4803533

View online: http://dx.doi.org/10.1119/1.4803533

View Table of Contents: http://ajp.aapt.org/resource/1/AJPIAS/v81/i7

Published by the American Association of Physics Teachers

SHARPEN YOUR COMPUTATIONAL SKILLS.
computing in SCIENCE & ENGINEERING
Scientific Computing with GPUs
Subscribe for $49 | year

COMPUTATIONAL PHYSICS

The Computational Physics Section publishes articles that help students and their instructors learn about the physics and the computational tools used in contemporary research. Most articles will be solicited, but interested authors should email a proposal to the editors of the Section, Jan Tobochnik (jant@kzoo.edu) or Harvey Gould (hgould@clarku.edu). Summarize the physics and the algorithm you wish to include in your submission and how the material would be accessible to advanced undergraduates or beginning graduate students.

Of pendulums, polymers, and robots: Computational mechanics with constraints

Franz J. Vesely
Computational Physics Group, Faculty of Physics, University of Vienna, Boltzmanngasse 5, A-1090 Vienna, Austria

(Received 26 March 2013; accepted 17 April 2013)

The motion of point masses under the influence of a potential can be computed by simple methods. However, if the trajectories are restricted by mechanical constraints such as strings, rails, crankshafts, and molecular bonds, special numerical techniques must be invoked. The need for efficient computational strategies is particularly pressing for molecular simulations, where large systems of compound molecules are tracked. The best strategy is the use of Cartesian coordinates in combination with constraint forces in the Lagrange formulation. This approach has led to the extremely successful SHAKE and RATTLE algorithms. The same ideas may be profitably applied in very different fields such as robotics, mechanics, and geometry, and the study of chaos in simple systems.

[http://dx.doi.org/10.1119/1.4803533]

I. INTRODUCTION

Theoretical mechanics was born in 1687. At that time, it was still the main intent of *philosophia naturalis* to detect divine harmony in all things natural. The motion of heavenly bodies, which had puzzled people for thousands of years, had become exciting again as Halley's comet swung by in 1682. Accordingly, Newton's explanation of this motion fascinated learned persons of his time.[1] In today's terminology, the circulation of planets around the Sun or the regular swing-by of comets is called "free motion"—the bodies are not free of forces, but they are free to go where the forces drive them. According to popular legend, it was the genius of Newton that allowed him to treat the celestial choreography and such mundane motions as the fall of an apple on a philosopher's head by the same mathematical formalism. Humans being what they are, they were quick to apply the same apparatus to the flight of cannonballs.

But how should we describe the swinging of a pendulum? And what about carts on rails, pulleys, piston rods, crankshafts, and and other parts of man-made machinery? With the onset of the first industrial revolution in the 18th century, it became important to understand the motion of mechanical parts. It is no coincidence that Lagrange first published his *Méchanique Analytique* in 1788.[2] He derived two versions of dynamical equations that could be applied to systems with mechanical constraints. The *Lagrange equations of the first kind* are based on the idea that the constraints may be replaced by suitable forces which are designed to enforce the constraints, and which are simply added to the given physical forces. In the *Lagrange equations of the second kind*, the generalized coordinates used to describe the system are chosen such that some of them correspond to the constraints of the motion. The respective coordinates are then constant, and the number of dynamical equations is reduced.

The planar pendulum demonstrates the idea. We can either describe the movement of the swinging mass by two arbitrary (usually Cartesian) coordinates, adding a force that keeps the distance ℓ to the pivot at a constant value or by the generalized coordinates (ℓ, ϕ). The first method requires three equations—two equations of motion and an expression for the restoring force. In the second approach only the coordinate ϕ is subject to a time evolution described by an equation of motion.

For pedagogical purposes, let us consider a simpler example. The uniform motion of a point around a circle of radius ℓ may be described in terms of the dynamical equation $\ddot{\mathbf{r}} = \mathbf{f}_c/m$, with a constraint force that is acting along the radius vector: $\mathbf{f}_c = -\lambda\mathbf{r}(t)$ such that it guarantees that $\sigma = |\mathbf{r}(t)|^2 - \ell^2 = 0$ at all times. This condition implies that $\dot{\sigma} = 2\,\mathbf{r}\cdot\dot{\mathbf{r}} = 0$, which means that the tangential velocity is normal to the radius vector, and $\ddot{\sigma} = 2\,|\dot{\mathbf{r}}|^2 + 2\,\mathbf{r}\cdot\ddot{\mathbf{r}} = 0$. If we substitute $\ddot{\mathbf{r}} = \mathbf{f}_c/m = -\lambda\mathbf{r}(t)/m$ and $r^2 = \ell^2$, we find $\lambda = mv^2/\ell^2$. Lagrange's first formulation tells us that we have to integrate the dynamical equation $\ddot{\mathbf{r}} = -(v^2/\ell^2)\mathbf{r}$ with a given (constant) tangential speed v.

Alternatively, Lagrange's second formulation describes this motion by $\ell =$ constant and $\ddot{\phi} = 0$, which leads to a fixed angular velocity $\dot{\phi} = \omega$. The solution is $\phi(t) = \phi(0) + \omega t$.

While savoring the formal elegance of Lagrange's second method, we have to consider if it is always the optimal basis for numerical work. This consideration is particularly important in simulations of interest in statistical mechanics, where the limitations of computers are felt most acutely, making the computing strategy an essential issue. In the early 1970s, molecular simulation proceeded from simple structureless models such as Lennard-Jones particles to more interesting objects such as linear diatomics.[3–5] Next on the agenda were polymers, the building blocks of most substances that play a role in biology and technology.[6] But here simulators encountered a serious difficulty—the interatomic bonds. If we envisage the bonds as stiff springs, the force constants that describe the small oscillations about the average bond lengths are large in comparison to the other forces acting between atoms. To correctly simulate these fast and small oscillations, we would have to use time steps that are less than the reciprocal vibratory frequencies—much too small for reasonable simulation times. We might argue on quantum-mechanical grounds that most bond vibrations have such high energy that they are frozen at normal temperatures. In other words, it is a promising *Ansatz* to treat the bonds as rigid, and thus model the polymer as a flexible chain of limbs and joints—a "Kramers chain."[7]

The optimal method for simulating such chains is related to Lagrange's first method. The apparent disadvantage of the larger number of equations of motion is more than offset by the simplicity and computational speed of the methods we will describe under the headings of "SHAKE" and "RATTLE."

But first we will discuss an algorithm that was designed for unconstrained motion and which has become the basis for constrained dynamics.

II. THE VERLET ALGORITHM

Except for a few simple cases, Newton's equation for the unconstrained motion of a particle under the action of a force can be solved only numerically. Of the many algorithms that produce a stepwise solution to the dynamical equation $\ddot{\mathbf{r}} = \mathbf{f}(\mathbf{r})/m$, only a very few are fast enough to be applicable in high performance simulations.

The atoms of noble gases may be represented by particles interacting via a Lennard-Jones potential. Following the pioneering work of Aneesur Rahman,[8] Louis Verlet performed the first extensive molecular dynamics investigation on such a model.[9] He used a simple and efficient algorithm that now bears his name, although a similar method had been used as early as 1905 by the mathematician Carl Størmer in his studies of the polar aurora.[10] Verlet's version reads

$$\mathbf{r}(t_{n+1}) = 2\,\mathbf{r}(t_n) - \mathbf{r}(t_{n-1}) + \frac{(\Delta t)^2}{m}\mathbf{f}(t_n), \tag{1}$$

where $\mathbf{f}$ is the force and Δt the time step. The local error of the total energy is fourth order in Δt. Due to its simplicity and computational speed, the method is popular in simulations of unconstrained motion and is the basis of the SHAKE algorithm.[11]

Note that the velocity $\mathbf{v} \equiv \dot{\mathbf{r}}$ does not appear explicitly in Eq. (1). The approximation for the velocity $\mathbf{v}(t_n) = [\mathbf{r}(t_{n+1}) - \mathbf{r}(t_{n-1})]/2\Delta t + O[(\Delta t)^2]$ is inaccurate and can be used only for crude checks of energy conservation. Also, the algorithm is not self-starting; in addition to the initial position $\mathbf{r}(t_0)$, we need $\mathbf{r}(t_{-1})$ to tackle the first time step. This need is not a problem in statistical mechanical simulations where the initial condition $[\mathbf{r}(0), \mathbf{v}(0)]$ is of no particular interest. In other contexts, however, it is desirable to have a method that starts with given positions and velocities. We could use the versatile and powerful Runge–Kutta algorithm but that would mean a distinctly lower calculational speed. It turns out that a slightly different but equivalent formulation of Verlet's algorithm, known as the velocity Verlet or Swope algorithm,[12] is self-starting and is given by

$$\mathbf{v}(t_{n+1/2}) = \mathbf{v}(t_n) + \frac{\Delta t}{2m}\mathbf{f}(t_n), \tag{2a}$$

$$\mathbf{r}(t_{n+1}) = \mathbf{r}(t_n) + \Delta t\,\mathbf{v}(t_{n+1/2}), \tag{2b}$$

$$\mathbf{v}(t_{n+1}) = \mathbf{v}(t_{n+1/2}) + \frac{\Delta t}{2m}\mathbf{f}(t_{n+1}). \tag{2c}$$

This algorithm is the basis for Andersen's RATTLE method.[13]

III. LAGRANGE MULTIPLIERS

Verlet's and other integration methods apply only to forces derived from a potential but not to Lagrange's constraint forces. To include the latter, a common strategy is to first ignore the constraints and proceed one time step as if the mass point were free to follow the given potential force. The appropriate Lagrange force is then applied to enforce the geometric conditions at the end of the time step. The procedure for one particle and one constraint $\sigma(\mathbf{r})$ is as follows:

(1) Proceed one time step using the Verlet algorithm. Given the initial position $\mathbf{r}(t_n)$ find the preliminary position $\mathbf{r}'$ at time $t_{n+1} = t_n + \Delta t$. This position does not satisfy the constraint equation, meaning that the value of $\sigma' \equiv \sigma(\mathbf{r}')$ differs from zero.

(2) Assume that the Lagrange force acts along the gradient of $\sigma(\mathbf{r}(t_n))$ and make the correction *Ansatz*

$$\mathbf{r}(t_{n+1}) = \mathbf{r}' - [\lambda(\Delta t)^2/m]\nabla\sigma(\mathbf{r}(t_n)). \tag{3}$$

We require that $\sigma(\mathbf{r}(t_{n+1})) = 0$ and obtain

$$0 = \sigma\left(\mathbf{r}' - \lambda\frac{(\Delta t)^2}{m}\nabla\sigma(\mathbf{r}(t_n))\right)$$
$$\approx \sigma(\mathbf{r}') - \lambda\frac{(\Delta t)^2}{m}\nabla\sigma(\mathbf{r}')\cdot\nabla\sigma(\mathbf{r}(t_n)), \tag{4}$$

which yields an approximate equation for the undetermined multiplier λ

$$\lambda = \frac{m}{(\Delta t)^2}\,\frac{\sigma'}{\nabla\sigma(\mathbf{r}')\cdot\nabla\sigma(\mathbf{r}(t_n))}. \tag{5}$$

(3) Treat Eq. (5) as a first estimate for λ and proceed by iteration:
- Substitute λ into Eq. (3) for $\mathbf{r}(t_{n+1})$ to obtain the improved position $\mathbf{r}'' = \mathbf{r}' - \lambda[(\Delta t)^2/m]\nabla\sigma(\mathbf{r}(t_n))$. The value of $\sigma'' \equiv \sigma(\mathbf{r}'')$ still differs from zero but should be smaller than σ'.
- Substitute σ'' and $\mathbf{r}''$ into Eq. (5) for λ and obtain another approximation for $\mathbf{r}(t_{n+1})$.
- Repeat until the constraint equation is satisfied to within a given tolerance.

A. Motion restricted to curves or surfaces

In two dimensions, the equation $\sigma(\mathbf{r}(t)) \equiv |\mathbf{r}(t)| - \ell^2 = 0$ states the requirement that a moving point stays on a circle.

Table I. Various plane curves and their Cartesian constraint functions.

Shape	Constraint function $\sigma(\mathbf{r})$	Gradient $\nabla\sigma(\mathbf{r})$
Parabola	$ax^2 - y$	$(2ax, -1)$
Ellipse	$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 - 1$	$2\left(\frac{x}{a^2}, \frac{y}{b^2}\right)$
Superellipse	$\left(\frac{\lvert x\rvert}{a}\right)^n + \left(\frac{\lvert y\rvert}{b}\right)^n - 1$	$n\left(\frac{\lvert x\rvert^{n-1}}{a^n}\,\mathrm{sgn}(x), \frac{\lvert y\rvert^{n-1}}{b^n}\,\mathrm{sgn}(y)\right)$
Discorectangle	$\frac{x^2}{r^2} - 1$	$\frac{2}{r^2}(x, 0)\ (\lvert y\rvert \le y_0)$
	$\frac{x^2 + (y - y_0)^2}{r^2} - 1$	$\frac{2}{r^2}(x, \mathrm{sgn}(y)(\lvert y\rvert - y_0))\ (y_0 < \lvert y\rvert \le y_0 + r)$
Bean curve	$x^4 + x^2y^2 + y^4 - x(x^2 + y^2)$	$(4x^3 - 3x^2 + 2xy^2 - y^2, 4y^3 + 2y(x^2 - x))$

More general constraint functions may be defined, stipulating the restriction to some curvilinear path. Roads and railroad tracks may be shaped like a clover leaf or a superellipse (see Table I). Since the age of chariot races, racing tracks have been constructed in the form of a stadium or discorectangle.[14] To describe the motion along a curved track, we usually apply the parametric representation of the curve, adding a relation between the parameter and the elapsed time. However, our discussion points to a road less traveled but equally passable. We have seen that to simulate the motion along a path defined by a Cartesian constraint function $\sigma(\mathbf{r})$ is an expression for the gradient $\nabla\sigma(\mathbf{r})$. By following the momentary tangent for a short time step and then iteratively enforcing the constraint, we can reproduce a table of path points at equidistant times t_n. If the constraint force—which does not appear explicitly in the algorithm—is of interest, it may be estimated *a posteriori* from the Newton-Gregory expression[15] $\ddot{\mathbf{r}}(t_{n-1}) \approx [\mathbf{r}(t_n) - 2\,\mathbf{r}(t_{n-1}) + \mathbf{r}(t_{n-2})]/(\Delta t)^2$.

As an example, consider uniform motion along the contour of an ellipse with $\sigma(\mathbf{r}) = (x/a)^2 + (y/b)^2 - 1$. Let $\mathbf{r}(t_n)$ and $\mathbf{r}(t_{n-1})$ be given. Then the free flight leads to $\mathbf{r}' = 2\,\mathbf{r}(t_n) - \mathbf{r}(t_{n-1})$, and the nonzero value of σ' may be computed. The local gradient at the trajectory point is $\nabla\sigma = (2x/a^2, 2y/b^2)$, which we can substitute into Eq. (5) to obtain a first estimate for λ. A few iterations will bring the point $\mathbf{r}(t_{n+1})$ back onto the ellipse.

Table I lists a few well-studied curves and their gradients. Problem 4 is devoted to the simulation of uniform motion around a superellipse or Lamé curve.

It is easy to see that the same mathematics can be applied to paths that are restricted to a surface embedded in three-dimensional space. Consider a ball rolling about in a parabolic well under the influence of the gravitational acceleration $\mathbf{g} = (0,\ 0,\ -9.81)$. The paraboloid's constraint function is $\sigma(\mathbf{r}) = a(x^2 + y^2) - z$, and its local gradient is $\nabla\sigma = (2ax,\ 2ay,\ -1)$. The elementary Verlet step is $\mathbf{r}(t_{n+1}) = 2\,\mathbf{r}(t_n) - \mathbf{r}(t_{n-1}) + \mathbf{g}(\Delta t)^2$, and the correction due to the Lagrange force $-\lambda\nabla\sigma$ can be determined by iteration. A similar example is the subject of problem 5.

B. Links and joints

As mentioned, an important model system in the physics of polymers is the "ideal" or Kramers chain. Rigid molecules may be built of closed loops of bonds with fixed lengths. In all such cases, the individual elements ("atoms") will have one or several bond constraints $\sigma_{ij} = 0$. Again, the strategy is to move each element i for one time step as if it were free to move according to the potential forces in the system. Then, we must determine *a posteriori* those values of the Lagrange forces that would have, when added to the physical forces, enforced the geometrical conditions to hold at the end of the time step

$$\mathbf{r}_i(t_{n+1}) = \mathbf{r}'_i - \frac{(\Delta t)^2}{m_i}\sum_j \lambda_{ij}\frac{\partial\sigma_{ij}}{\partial\mathbf{r}_i(t_n)}. \tag{6}$$

For bond constraints such as $\sigma_{ij} = |\mathbf{r}_{ij}|^2 - \ell^2$, where $\mathbf{r}_{ij} \equiv \mathbf{r}_j - \mathbf{r}_i$, the partial derivatives are of the simple form $\partial\sigma_{ij}/\partial\mathbf{r}_i = -2\,\mathbf{r}_{ij}$. If we introduce $\gamma_{ij} \equiv 2\lambda_{ij}(\Delta t)^2$, the updated positions may be written as

$$\mathbf{r}_i(t_{n+1}) = \mathbf{r}'_i + \frac{1}{m_i}\sum_j \gamma_{ij}\mathbf{r}_{ij}(t_n). \tag{7}$$

The Lagrange multipliers are determined by requiring that all constraints be intact at time t_{n+1}.

As an example, let us consider the smallest nontrivial Kramers chain molecule consisting of three atoms that are sequentially connected by massless rigid bonds. In each such trimer molecule, the two constraint equations involving the three atomic positions and the two bond lengths are $\sigma_{12}(\mathbf{r}_{12}) = |\mathbf{r}_{12}|^2 - \ell_{12}^2 = 0$ and $\sigma_{23}(\mathbf{r}_{23}) = |\mathbf{r}_{23}|^2 - \ell_{23}^2 = 0$. If we assume that we have already solved the unconstrained equations of motion to find $\mathbf{r}'_{1,2,3}$, we have

$$\mathbf{r}_1(t_{n+1}) = \mathbf{r}'_1 + \frac{1}{m_1}\gamma_{12}\mathbf{r}_{12}(t_n), \tag{8a}$$

$$\mathbf{r}_2(t_{n+1}) = \mathbf{r}'_2 + \frac{1}{m_2}[-\gamma_{12}\mathbf{r}_{12}(t_n) + \gamma_{23}\mathbf{r}_{23}(t_n)], \tag{8b}$$

$$\mathbf{r}_3(t_{n+1}) = \mathbf{r}'_3 - \frac{1}{m_3}\gamma_{23}\mathbf{r}_{23}(t_n). \tag{8c}$$

We substitute $\mathbf{r}_{1,2,3}$ into the bond equations and obtain the following system of equations for the unknown Lagrange multipliers:

$$\left|\mathbf{r}'_{12} - \gamma_{12}\mathbf{r}_{12}(t_n)\left(\frac{1}{m_2} + \frac{1}{m_1}\right) + \gamma_{23}\mathbf{r}_{23}(t_n)\frac{1}{m_2}\right|^2 - \ell_{12}^2 = 0, \tag{9a}$$

$$\left| \mathbf{r}'_{23} - \gamma_{23}\mathbf{r}_{23}(t_n)\left(\frac{1}{m_3}+\frac{1}{m_2}\right) + \gamma_{12}\mathbf{r}_{12}(t_n)\frac{1}{m_2}\right|^2 - \ell_{23}^2 = 0. \tag{9b}$$

Writing $\sigma'_{12} \equiv |\mathbf{r}'_{12}|^2 - \ell_{12}^2$ and $1/\mu_{12} \equiv 1/m_1 + 1/m_2$, these equations become

$$\sigma'_{12} - 2\frac{\gamma_{12}}{\mu_{12}}[\mathbf{r}_{12}(t_n)\cdot\mathbf{r}'_{12}] + 2\frac{\gamma_{23}}{m_2}[\mathbf{r}_{23}(t_n)\cdot\mathbf{r}'_{12}] + (\cdots)^2 = 0, \tag{10a}$$

$$\sigma'_{23} + 2\frac{\gamma_{12}}{m_2}[\mathbf{r}_{12}(t_n)\cdot\mathbf{r}'_{23}] - 2\frac{\gamma_{23}}{\mu_{23}}[\mathbf{r}_{23}(t_n)\cdot\mathbf{r}'_{23}] + (\cdots)^2 = 0, \tag{10b}$$

where $(\cdots)^2$ are terms that are quadratic in γ_{ij}.

In general, Eq. (10) is a system of quadratic equations; however, there are two good reasons to hope that the computing expense will be manageable. First, the matrix is sparse and often has useful symmetry. And second, the unknowns γ_{ij} are small so that the linear terms dominate in the equations. In the following sections, we discuss the most important solution methods.

IV. SHAKE

To describe the idea of Ryckaert *et al.*,[11] we again consider the trimer we discussed earlier.

(1) Given the positions $\mathbf{r}_i(t_{n-1})$ and $\mathbf{r}_i(t_n)$ at the last two time steps, apply Verlet's method to integrate the equations of motion for one time step without considering the constraint forces, that is, with $\gamma_{12} = \gamma_{23} = 0$; the resulting positions are denoted as $\mathbf{r}'_i$. Because these preliminary position vectors do not satisfy the constraint equations, the values of $\sigma_{12}(\mathbf{r}'_{12})$ and $\sigma_{23}(\mathbf{r}'_{23})$ have nonzero values which we denote as σ'_{12} and σ'_{23}.

(2) Make the correction *Ansatz*

$$\mathbf{r}''_1 = \mathbf{r}'_1 + \frac{1}{m_1}\gamma_{12}\mathbf{r}_{12}(t_n), \tag{11a}$$

$$\mathbf{r}''_2 = \mathbf{r}'_2 + \frac{1}{m_1}[-\gamma_{12}\mathbf{r}_{12}(t_n) + \gamma_{23}\mathbf{r}_{23}(t_n)], \tag{11b}$$

$$\mathbf{r}''_3 = \mathbf{r}'_3 - \frac{1}{m_3}\gamma_{23}\mathbf{r}_{23}(t_n), \tag{11c}$$

where γ_{ij} denotes the temporary—to be improved—values of the Lagrange parameters, and $\mathbf{r}''_i$ is the improved position that should eventually converge to $\mathbf{r}_i(t_{n+1})$. To find a first estimate of γ_{ij}, we require that the corrected positions satisfy the constraint equations as in Eq. (10). However, we now neglect the quadratic terms, giving

$$\sigma'_{12} - 2\frac{\gamma_{12}}{\mu_{12}}[\mathbf{r}_{12}(t_n)\cdot\mathbf{r}'_{12}] + 2\frac{\gamma_{23}}{m_2}[\mathbf{r}_{23}(t_n)\cdot\mathbf{r}'_{12}] = 0, \tag{12a}$$

$$\sigma'_{23} + 2\frac{\gamma_{12}}{m_2}[\mathbf{r}_{12}(t_n)\cdot\mathbf{r}'_{23}] - 2\frac{\gamma_{23}}{\mu_{23}}[\mathbf{r}_{23}(t_n)\cdot\mathbf{r}'_{23}] = 0. \tag{12b}$$

In principle, we could solve this system of linear equations exactly to obtain an improved estimate for γ_{ij}, to be substituted iteratively in the correction Eq. (11).

(3) At this point, another simplification is introduced. The exact solution of the linearized equation (12) at each iteration step involves a matrix inversion (see Sec. VI). Instead, we start from one end of the chain and consider only one constraint per atom, and one improvement step per constraint, as we go along. In other words, we first improve the bond $\mathbf{r}_{12}$ by displacing 1 and 2, and then repair the next constraint $\mathbf{r}_{23}$, thereby partly disrupting the first bond again. In terms of Eq. (12), this procedure means that only the diagonal terms with $\mathbf{r}_{12}(t_n)\cdot\mathbf{r}'_{12}$ and $\mathbf{r}_{23}(t_n)\cdot\mathbf{r}'_{23}$ are kept. In this way, the matrix inversion becomes simple and reads

$$\gamma_{12} = \frac{\mu_{12}}{2}\frac{\sigma'_{12}}{\mathbf{r}'_{12}\cdot\mathbf{r}_{12}(t_n)} \quad \text{and} \quad \gamma_{23} = \frac{\mu_{23}}{2}\frac{\sigma'_{23}}{\mathbf{r}'_{23}\cdot\mathbf{r}_{23}(t_n)}. \tag{13}$$

By going through the chain several times, the errors introduced by neglecting the quadratic terms and by considering only one constraint at a time will normally decrease sufficiently quickly. The remaining error is due to the assumption that the Lagrange force acting during the interval (t_n, t_{n+1}) is parallel to the gradient of σ at the beginning of this interval. Ryckaert *et al.*[11] showed that the local error induced by this simplification is of the same order, namely $(\Delta t)^4$, as the error introduced by the Verlet algorithm itself.

The generalization of the SHAKE technique to long chains is simple. Applications to very long chain molecules, particularly biomolecules, abound in the literature.[16–18] The widely used program packages GROMOS[19] and CHARMM[20] contain SHAKE subroutines as a matter of course.

The SHAKE algorithm was a breakthrough in molecular simulation. However, because it is based on the Verlet algorithm, it does not contain the velocities explicitly. This feature turned out to be a serious drawback when, around 1980, simulators started to implement thermostats in their model systems. The constancy of temperature can only be guaranteed by some sort of manipulation of the particle velocities.[21,22]

V. RATTLE

To overcome the limitations of SHAKE, Andersen[13] suggested using the velocity Verlet algorithm given in Eq. (2). He pointed out that to arrive at a usable procedure we only have to introduce two Lagrange multipliers for each constraint, namely λ_{ij}^R to fulfill the bond length requirement $\sigma_{ij}(t_{n+1}) = 0$, and a separate λ_{ij}^V to enforce $\mathbf{v}_{ij}(t_{n+1})\cdot\mathbf{r}_{ij}(t_{n+1}) = 0$ so that the relative velocity $\mathbf{v}_{ij}$ remains perpendicular to $\mathbf{r}_{ij}$. With the notation $\delta_{ij} \equiv \Delta t\,\lambda_{ij}$ Andersen's algorithm becomes

$$\mathbf{v}_i(t_{n+1/2}) = \mathbf{v}_i(t_n) + \frac{\Delta t}{2m_i}\mathbf{f}_i(t_n) + \frac{1}{m_i}\sum_j \delta_{ij}^R\,\mathbf{r}_{ij}(t_n), \tag{14a}$$

$$\mathbf{r}_i(t_{n+1}) = \mathbf{r}_i(t_n) + \Delta t\,\mathbf{v}_i(t_{n+1/2}), \tag{14b}$$

$$\mathbf{v}_i(t_{n+1}) = \mathbf{v}_i(t_{n+1/2}) + \frac{\Delta t}{2m_i}\mathbf{f}_i(t_{n+1}) + \frac{1}{m_i}\sum_j \delta_{ij}^V\,\mathbf{r}_{ij}(t_{n+1}), \tag{14c}$$

where the index j runs over those particle numbers that are linked to particle i by bond constraints.

At each time step, the first task is to find the exact $\mathbf{v}_i(t_{n+1/2})$ by iteration. As in SHAKE, we begin by ignoring the constraints, that is, by setting $\delta^R_{ij} = 0$, or $\mathbf{v}_i(t_{n+1/2}) \equiv \mathbf{v}_i(t_n) + (\Delta t/2m)\mathbf{f}_i(t_n)$ for all particles $i = 1 \ldots N$. Next, we start the iteration:

(1) Choose a constraint (i, j) and compute a preliminary new interparticle vector according to $\mathbf{r}'_{ij} = \mathbf{r}_{ij}(t_n) + \mathbf{v}_{ij}(t_{n+1/2})\Delta t$. We use $\sigma'_{ij} = |\mathbf{r}'_{ij}|^2 - \ell^2_{ij}$ and have for the first correction to the Lagrange multiplier

$$\delta^R_{ij} = \frac{\mu_{ij}}{2\Delta t}\,\frac{\sigma'_{ij}}{\mathbf{r}'_{ij}\cdot\mathbf{r}_{ij}(t_n)}. \tag{15}$$

If we substitute Eq. (15) into Eq. (14a), we find improved mid-time velocities for particles i and j. The next bond constraint is treated in the same way, and the entire chain is repeatedly iterated until the improvements become negligible and the vectors $\mathbf{v}_{n+1/2}$ and $\mathbf{r}_{n+1}$ can be accepted.

(2) Use $\mathbf{r}_i(t_{n+1})$ and evaluate the potential forces at t_{n+1}.

(3) Now we can determine $\mathbf{v}_i(t_{n+1})$ by iteration. We pick a constraint (i,j) and write for each of the two particles $\mathbf{v}' = \mathbf{v}_{n+1/2} + (\Delta t/2m)\mathbf{f}(t_{n+1})$, for the moment ignoring the bond forces. To improve on this estimate, we write $\mathbf{v}_i(t_{n+1}) = \mathbf{v}'_i - (1/m_i)\delta^V_{ij}\,\mathbf{r}_{ij}(t_{n+1})$ and $\mathbf{v}_j(t_{n+1}) = \mathbf{v}'_j + (1/m_j)\delta^V_{ij}\,\mathbf{r}_{ij}(t_{n+1})$. We recall the requirement $\mathbf{r}_{ij}(t_{n+1})\cdot\mathbf{v}_{ij}(t_{n+1}) = 0$ and find for the first estimate of the Lagrange multiplier that

$$\delta^V_{ij} = \frac{\mu_{ij}}{l^2_{ij}}[\mathbf{r}_{ij}(t_{n+1})\cdot\mathbf{v}'_{ij}]. \tag{16}$$

Again, all constraints are treated repeatedly until the velocities $\mathbf{v}_i(t_{n+1})$ are sufficiently accurate.

Andersen has shown that the local error introduced by the RATTLE scheme is fourth order in Δt, just as the error of the basic velocity Verlet formula.[13]

VI. SPECIAL ALGORITHMS FOR MOLECULAR SIMULATION

The methods we have described are applicable for both macroscopic mechanics and in molecular simulations. In the latter field, several more advanced methods have been developed to treat particular classes of molecular models, such as small, rigid entities consisting of a closed cage of bonds, and very long, flexible polymeric chains.

Both of the algorithms rely on a double iteration. Equation (12) is a linearly truncated version of the quadratic constraint equations, so the solution of the quadratic equations calls for one iterative procedure. The other repetitive operation consists in treating the set of bond constraints several times, always re-enforcing one constraint at a time.

Because the set of constraints provides us with a system of coupled equations, an alternative strategy is to apply a standard method for solving such a system. Such a strategy involves the inversion of the matrix defined by the set of equations. SHAKE and RATTLE circumvent the exact matrix inversion by the double recurrence we have described. We may understand this approach to be a variant of the iterative Gauss-Seidel-Newton method.[23]

Several algorithms have been proposed which invoke other, more efficient solution methods. Ciccotti and Ryckaert,[24] who were among the authors of the original SHAKE method, were also the first to point out that not only chains but also rigid molecular skeletons may be treated successfully by algorithms based on Lagrange's equations of the first kind. For example, a stiff molecule made up of three atoms or atomic groups may be described by three rigid bonds. They suggested inverting the matrix defined by the simultaneous linearized constraint equations exactly, and then iterating only to arrive at the solution to the full quadratic equations. Their matrix method[24] is applicable to small molecules such as CS_2, benzene, and CCl_4. For larger molecules, the matrix inversion becomes too computationally intensive due to the fact that matrix inversion is in general a N^3 operation, meaning that the computational load grows with the cube of the number of constraints. However, certain symmetry properties of the matrices in question allow for faster inversion algorithms, which is the starting point for several acceleration methods such as SETTLE,[25] SOR-SHAKE,[23] M-SHAKE,[26] and P-SHAKE.[27]

For long flexible chains, we can easily see that the set of linearized equations [see Eq. (12) for the trimer] is tridiagonal. The inversion of such matrices is particularly simple and fast and is an order N operation.[15,28] By using this fact, Bailey *et al.*[29] developed a very fast algorithm for the dynamics of long flexible polymers. They showed that for very long chains with up to 10,000 vertices, their MILC SHAKE scheme is 2–3 orders of magnitude faster than the algorithms we have discussed in the previous sections. For ten elements per chain, the improvement is still a factor of 3.

VII. APPLICATIONS

A. Double/triple pendulum

By proceeding beyond small angles, we may discuss the general solution of the simple pendulum in terms of elliptic integrals, and on the computational side the numerical inversion of higher functions to tabulate $\phi(t)$ instead of $t(\phi)$. Although the usual procedure is to solve the pendulum equation in terms of the one free generalized coordinate $\phi(t)$, it is worthwhile for pedagogical reasons to try the alternative route using Cartesian coordinates plus a Lagrange constraint force (see problem 8.2).

By going one step further and attaching a second mass to the first, one we obtain a system that is capable of chaotic behavior. The theory of double- and n-fold pendula is usually formulated in terms of Lagrange equations of the second kind or Hamiltonian mechanics, but numerical simulation is much simpler if written in terms of Cartesian coordinates and Lagrange-type constraint forces.[30,31]

Both the SHAKE and RATTLE algorithms may be employed here, but the latter lends itself more easily to arbitrary choices of initial angular velocities. Problems 1–3 lead to a basic simulation code for an n-fold pendulum. A JAVA applet written along those lines, entitled `Pendel2`, is available online.[32] The applet may be used to investigate the regular and chaotic behavior of the double or triple pendulum. The respective FORTRAN programs, `pnd2s.f` for SHAKE and `pnd2r.f` for RATTLE, are available for download.[32]

B. Robot arms

Generalized coordinates are commonly used in robotics to describe the motion of segmented arms. This choice seems natural because it keeps the number of equations of motion small. However, there are good arguments favoring the use of Cartesian coordinates with constraint forces.

Consider a simple manipulator consisting of several rigid segments ("limbs") connected by joints with servomotors and a final element that is to follow a desired trajectory, say, to carry some object from one place to another, or to paint some contour. The similarity to the Kramers chain polymer is obvious. By exploiting this similarity, we may attack a standard problem of robotics, known as the *inverse kinematic problem*, in a new way.

Let the angles between successive limbs change according to some temporal program $\phi_i(t)$. The end element of the arm, the "hand," will then go through a well-defined "world trajectory" $\mathbf{r}_e(t)$. Inversely, if a desired trajectory is given, we may ask for those $\phi_i(t)$ that produce it—this is the inverse kinematic problem of robot control. Because manipulators are usually redundant, having more joints than necessary, the problem is underdetermined and we may impose additional requirements such as obstacle avoidance and minimization of angular velocities and accelerations. In terms of generalized coordinates, solving the inverse problem with additional requirements may be a formidable task.

By using Cartesian coordinates, the problem is radically simplified. The constrained kinematics/stochastic optimization algorithm of Kastenmeier and Vesely[33,34] proceeds as follows: (a) move the robot hand stepwise through its required trajectory $\mathbf{r}_e(t_n)$; (b) invoke SHAKE to move the preceding joints according to the constraint of constant limb lengths; and (c) add small random variations to the joint positions and apply an optimizing strategy to satisfy the additional requirements. The motion of a two-dimensional redundant manipulator is demonstrated by the applet `Robie`.[32]

C. Random walk of a two-dimensional Kramers chain

Ryckaert and Andersen applied their procedures on top of the Verlet algorithm, thus simulating the deterministic motion of particles subject to geometrical constraints.[11,13] Alternatively, we may do a simple random walk with the individual parts of a chain and then enforce the bond constraints as we have discussed. This physical scenario is a Kramers chain in a thermal bath, receiving random kicks from its environment. The applet `Kramers` illustrates this kind of motion for the two-dimensional case;[32] the respective Fortran code, `kram5bd.f`, may be downloaded from the same site.

Because this example is a particularly simple implementation of SHAKE, the applet is equipped with many adjustable parameters, so that the user can develop a feel for the relative roles of the step size $|\Delta\mathbf{r}|$, the maximum number of SHAKE iterations, and the tolerance limit of the bond length deviations, that is, constraint violations.

Also interesting are the statistics of the angle $\phi_{i,i+1}$ between consecutive bonds. In particular, for the two-dimensional stochastic three-chain with two bonds, the probability density may be shown to be $p(\phi) \propto \sqrt{1-(\cos\phi)^2/4}$. At first sight, this result is surprising because we expect all bond angles to occur with the same probability (the atoms are supposed to have no interaction at all except for the bond constraint). If we were to replace the rigid constraints by more or less stiff springs, all the angles ϕ would be equally probable, independently of the spring constant. A careful analysis shows that this qualitative difference is due to the fact that in the rigid bond case there are degrees of freedom (along the bonds) that are frozen and do not contain thermal energy. Therefore, they do not count in the partition function. In contrast, each spring coordinate carries an average energy of $kT/2$ regardless of the strength of the spring. The full explanation is given in Fixman's 1974 paper.[35]

Table II. Some second-order surfaces and their Cartesian constraint functions. A pseudo-hyperboloid is generated by rotating the hyperbola $z = -a/\sqrt{x^2+y^2}$ around the z axis.

Shape	Constraint function $\sigma(\mathbf{r})$	Gradient $\nabla\sigma(\mathbf{r})$
Sphere	$(x^2+y^2+z^2)/r^2-1$	$(2x/r^2,\ 2y/r^2,\ 2z/r^2)$
Paraboloid	$a(x^2+y^2)-z$	$(2ax,\ 2ay,\ -1)$
Hyperboloid	$\frac{x^2+y^2}{a^2}-\frac{z^2}{c^2}-1$	$\left(\frac{2x}{a^2},\frac{2y}{a^2},-\frac{2z}{c^2}\right)$
Pseudo-hyperboloid	$\frac{-a}{\sqrt{x^2+y^2}}-z$	$\left(\frac{ax}{(x^2+y^2)^{3/2}},\frac{ay}{(x^2+y^2)^{3/2}},-1\right)$

D. Motion on curves or surfaces

By formulating the problem in terms of Lagrange equations of the first kind, we can treat plane curves such as those listed in Table I without much effort. By using the expressions for σ and $\nabla\sigma$ in Eq. (5) and iterating, we obtain a table of points on the trajectory. Potential forces such as gravity can be added easily, and the acceleration can be approximated using the Newton-Gregory approximation[15] to the second time derivative (see Sec. III A). The sample FORTRAN code, `curves.f`, provides the basic pattern.[32]

Table II shows the constraint functions and their gradients for a few second-order surfaces.[36] Again, the tabulated items suffice to follow the motion of an object gliding along these surfaces. The principle of the calculation may be appreciated from the sample FORTRAN code `surface.f`.

VIII. SUMMARY

Lagrange's first method may be converted into a veritable workhorse for the numerical treatment of mechanical motion with geometrical constraints. Human- and molecular-sized systems may be tackled with minimal analytical effort and at moderate computational expense. Present day molecular simulations are unthinkable without the SHAKE, RATTLE, and related algorithms. Tedious problems in robotics, such as multilink manipulators moving in the presence of obstacles, may be represented by short, modular pieces of code when they are formulated in Cartesian instead of generalized coordinates. The virtual double or triple pendulum provides a road into chaos. In addition, the motion along arbitrary tracks or on some given surface is most easily followed by introducing Lagrange forces that keep the moving object on the desired contour.

IX. SUGGESTED PROBLEMS

Problem 1—Circular motion according to Langrange

Use your favorite programming language to follow a particle around a circular path. Start at $\mathbf{r}_0 = (1, 0)$ and assume a tangential velocity $\mathbf{v}_0 = (0, -1)$ [or an appropriate $\mathbf{r}(-\Delta t)$] and track the uniform circular motion in discrete time steps $\Delta t = 0.01$. For the first time step, the procedure is as follows:

(1) Let the point move away along the local tangent $\mathbf{r}' = \mathbf{r}_0 + \mathbf{v}_0 \Delta t$ [or $\mathbf{r}' = 2\,\mathbf{r}_0 - \mathbf{r}(-\Delta t)$].

(2) Use the nonzero value of the constraint function $\sigma' = |\mathbf{r}'|^2 - \ell^2$ to restore the condition $\sigma(\Delta t) = 0$ by moving the point along the direction of $\nabla\sigma(\mathbf{r}_0) \propto \mathbf{r}_0$. Hence, $\mathbf{r}'' = \mathbf{r}' - (a/m)\mathbf{r}_0$ with $a = m\sigma'/2(\mathbf{r}' \cdot \mathbf{r}_0)$. Iterate until $\sigma \approx 0$.

Compare the values of $\phi(n\Delta t)$ produced in this manner to the simple expression $\phi(t) = \omega t$.

Problem 2—The pendulum: Circular motion with gravity

Do the same as in problem 1 but introduce the gravitational force $\mathbf{f} = [0, -mg]$, where $g = 9.81$ (and mass $m = 1$.) Use Verlet's algorithm for the "free" flight and add the constraint force according to the procedure we have given. There are two levels of difficulty. If we use Verlet's algorithm, we arrive at a simple program but we have to adjust the initial condition $\mathbf{r}_{-1}$ to approximate the desired initial velocity (for example, zero). Or we can use the velocity Verlet algorithm together with Andersen's constraint method (see Sec. V) to obtain a slightly more complex code which allows for the customary initial condition $(\mathbf{r}_0, \mathbf{v}_0)$.

Problem 3—The double pendulum

Add a second bob fixed to the first by a bond with fixed unit length. SHAKE is simpler, but RATTLE is more appropriate for the usual initial conditions.

Problem 4—Uniform motion around Sergels Torg

The landmark Sergels Square in Stockholm was designed by the Danish architect Piet Hein. Remarkably, it is shaped as a superellipse with $n = 5/2$ (see Table I) and aspect ratio $a/b = 6/5$. The driving lane around the square has a long half-axis of about 25 m.[37] Use the information listed in Table I and calculate the lateral acceleration $\mathbf{a}(t)$ experienced by the passengers of a car circumnavigating the square at a constant speed of 50 km/h. Make a table of $\mathbf{a}(t_n)$ and compare it to a similar table for a stadium-shaped drive with the same total length and width. Note where the maximum of acceleration occurs on both tracks.

The same calculation may be performed using an ellipse, a "bean curve" (see Table I), or other plane curves with known Cartesian representations.

Problem 5—Model gravitational well

Many science museums offer a hands-on experiment to visualize the action of the gravitational field around a large mass. It consists of a polished hyperboloidal surface with a little starting device to push off a steel ball in an arbitrary direction. The ball will roll around the central hole in an orbit that, when seen from above, resembles the familiar elliptical path of a planet.

Strictly speaking, the surface is a pseudo-hyperboloid in the sense that it is produced by rotating the hyperbola $z = -a/\rho$ (with $\rho \equiv \sqrt{x^2 + y^2}$) around the z-axis. The idea is that the resulting funnel mimicks the action of the $1/\rho$ gravitational potential.

The constraint function for this surface is $\sigma(\mathbf{r}) = -(a/\rho) - z$, with gradient $\nabla\sigma(\mathbf{r}) = (ax/\rho^3,\, ay/\rho^3,\, -1)$. Place the ball at some point $\mathbf{r}_0$ on the surface and choose an initial velocity $\mathbf{v}_0 \perp \nabla\sigma(\mathbf{r})$. Start the simulation by the simple Euler integration step[15] $\mathbf{r}' = \mathbf{r}_0 + \mathbf{v}_0\Delta t + \mathbf{g}(\Delta t)^2/2$, where $\mathbf{g} = (0,\, 0,\, -9.81)$ is the gravitational acceleration. Apply the correction due to the constraint force $-\lambda\nabla\sigma(\mathbf{r}_0)$ and compute the first trajectory point. From then on the Verlet expression $\mathbf{r}' = 2\,\mathbf{r}(t_n) - \mathbf{r}(t_{n-1}) + \mathbf{g}(\Delta t)^2$ is used for each free-flight step, followed by the constraint correction.

Note that the x-y trajectories are only roughly similar to Kepler ellipses because the three-dimensional equation of motion is not identical to that of motion in a gravitational field. However, there are circular orbits remaining in a constant z plane, as well as tilted ellipsoidal orbits which may—contrary to classical Kepler ellipses—precess around the center axis.

ACKNOWLEDGMENTS

This article is dedicated to the memory of Konrad Singer, exceptional human being and scientist.

[1] In Newton's time the term "learned person" usually applied to men. However, there were two important exceptions. The French mathematician and physicist Emilie du Châtelet translated with commentary the *Principia Mathematica*, which is still considered the standard French translation. Also, it was probably upon instigation by the English writer Lady Mary Wortley Montagu that Francesco Algarotti wrote his famous "Il Newtonianismo per le dame," in which he explained Newton's Optics to those who were barred from higher education.

[2] Joseph Louis de Lagrange, *Mécanique Analytique*, 4th ed. (Gauthier-Villars et fils, Paris, 1888–1889); first published in 1788.

[3] J. Barojas, D. Levesque, and B. Quentrec, "Simulation of diatomic homonuclear liquids," Phys. Rev. A **7**(3), 1092–1105 (1973).

[4] K. Singer, A. Taylor, and J. V. L. Singer, "Thermodynamic and structural properties of liquids modelled by '2-Lennard-Jones centres' pair potentials," Mol. Phys. **33**(6), 1757–1795 (1977).

[5] A. Rahman and F. H. Stillinger, "Molecular dynamics study of liquid water," J. Chem. Phys. **55**(7), 3336–3359 (1971).

[6] See, for example, M. Bishop, M. H. Kalos, and H. L. Frisch, "Molecular dynamics of polymeric systems," J. Chem. Phys. **70**, 1299–1304 (1979).

[7] H. A. Kramers, "The behavior of macromolecules in inhomogeneous flow," J. Chem. Phys. **14**, 415–423 (1946).

[8] A. Rahman, "Correlations in the motion of atoms in liquid argon," Phys. Rev. **136**(2A), A405–A411 (1964).

[9] L. Verlet, "Computer 'experiments' on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules," Phys. Rev. **159**(1), 98–103 (1967); "Computer 'experiments' on classical fluids. II. Equilibrium correlation functions," *ibid.* **165**(1), 201–214 (1968).

[10] C. Størmer, "Sur les trajectoires des corpuscles electrisées dans l'espace sous l'action du magnétisme terrestre avec application aux aurores boréales," Arch. Sci. Phys. Nat. Genève **24**, 350–357 (1907); C. Størmer, "Méthode d'intégration numérique des équations différentielles ordinaires," Comptes Rendus du Congrès International des Mathématiciens, Strasbourg, 1920. See also C. Størmer, *The Polar Aurora* (Clarendon Press, Oxford, 1955).

[11] J.-P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, "Numerical integration of the cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes," J. Comput. Phys. **23**(3), 327–341 (1977).

[12] W. C. Swope, H. C. Andersen, B. H. Berens, and K. R. Wilson, "A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters," J. Chem. Phys. **76**, 637–649 (1982).

[13] H. C. Andersen, "RATTLE: A 'velocity' version of the SHAKE algorithm for molecular dynamics calculations," J. Comput. Phys. **52**, 24–34 (1983).

[14] Race drivers and slot car aficionados are well aware of an unwelcome peculiarity of this latter shape: At the end of the straight leg there is a discontinuous onset of centrifugal force, which tends to overturn both large vehicles and their miniature counterparts.

[15] F. J. Vesely, *Computational Physics: An Introduction*, 2nd ed. (Kluwer/Plenum, NY, 2001).

[16] E. Barth, K. Kuczera, B. Leimkuhler, and R. Skeel, "Algorithms for constrained molecular dynamics," J. Comput. Chem. **16**(10), 1192–1209 (1995); see Sec. 3.3.

[17] R. J. Sadus, *Molecular Simulation of Fluids: Theory, Algorithms and Object-Orientation* (Elsevier, Amsterdam, 1999).

[18] D. Brown, "The force of constraint in predictor-corrector algorithms for SHAKE constraint dynamics," Mol. Sim. **18**(6), 339–406 (1997).

[19] M. Christen, P. H. Hünenberger *et al.*, "The GROMOS software for biomolecular simulation: GROMOS05," J. Comput. Chem. **26**(16), 1719–1751 (2005).

[20] B. R. Brooks, C. L. Brooks CL 3rd *et al.*, "CHARMM: The biomolecular simulation program," J. Comput. Chem. **30**(10), 1545–1614 (2009).

[21] H. C. Andersen, "Molecular dynamics simulations at constant pressure and/or temperature," J. Chem. Phys. **72**, 2384–2393 (1980).

[22] W. G. Hoover, D. J. Evans, R. B. Hickman, A. J. C. Ladd, W. T. Ashurst, and B. Moran, "Lennard-Jones triple-point bulk and shear viscosities. Green-Kubo theory, Hamiltonian mechanics, and nonequilibrium molecular dynamics," Phys. Rev. A **22**, 1690–1697 (1980).

[23] See Sec. 3.3 of Ref. 16.

[24] G. Ciccotti and J.-P. Ryckaert, "Molecular dynamics simulation of rigid molecules," Comput. Phys. Rep. **4**(6), 346–392 (1986). See also G. Ciccotti, M. Ferrario, and J.-P. Ryckaert, "Molecular dynamics of rigid systems in cartesian coordinates. A general formulation," Mol. Phys. **47**(6), 1253–1264 (1982).

[25] S. Miyamoto and P. A. Kollman, "SETTLE: An analytical version of the SHAKE and RATTLE algorithm for rigid water models," J. Comput. Chem. **13**(8), 952–962 (1992).

[26] V. Kräutler, W. F. van Gunsteren, and P. H. Hünenberger, "A fast SHAKE algorithm to solve distance constraint equations for small molecules in molecular dynamics simulations," J. Comput. Chem. **22**(5), 501–508 (2001).

[27] P. Gonnet, "P-SHAKE: A quadratically convergent SHAKE in $O(n^2)$," J. Comput. Phys. **220**(2), 740–750 (2007).

[28] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. (Cambridge U.P., NY, 2007).

[29] A. G. Bailey, C. P. Lowe, and A. P. Sutton, "Efficient constraint dynamics using MILC SHAKE," J. Comput. Phys. **227**(20), 8949–8959 (2008).

[30] A. Ohlhoff and P. H. Richter, "Forces in the double pendulum," J. Appl. Math. Mech. **80**(8), 517–534 (2000).

[31] Vl. N. Salnikov, "On the dynamics of the triple pendulum: Non-integrability, topological properties of the phase space," Lecture notes, Conference on Dynamical Interability (CIRM) (2006).

[32] See supplementary material at http://dx.doi.org/10.1119/1.4803533 for Application of the algorithms discussed here is further demonstrated by the material deposited at these sites. The three Java applets Pendel2, Kramers, and Robie are devoted to pendulum and chain simulations, as are the downloadable FORTRAN codes pnd2s.f, pnd2r.f, and kram5bd.f. The programs curves.f and surface.f concern motions that are restricted to plane curves or to surfaces in 3D space. And also on the author's webpage at <homepage.univie.ac.at/franz.vesely/ajp13>.

[33] Th. Kastenmeier and F. J. Vesely, "Numerical robot kinematics based on stochastic and molecular simulation methods," Robotica **14**, 329–337 (1996).

[34] R. H. Bakken and A. Hilton, "Real-time pose estimation using constrained dynamics," Lect. Notes Comput. Sci. **7378**, 37–46 (2012).

[35] M. Fixman, "Classical statistical mechanics of constraints: A Theorem and application to polymers," Proc. Natl. Acad. Sci. U.S.A. **71**(8), 3050–3053 (1974).

[36] D. Hilbert and S. Cohn-Vossen, "The second-order surfaces," in *Geometry and the Imagination* (Chelsea, New York, 1999), Sec. 3, pp. 12–19.

[37] The size of Sergels torg is difficult to find on the web, and the author gratefully acknowledges the help of Ann-Charlotte Jönsson of the Stockholm Visitors Board, who personally measured the square by hand. She reported a half-length of 22.4 m for the inner rim of the drive.