# Geometric Numerical Integration

## 4. Constrained mechanical systems

**Mario Fernández-Pendás**
Universidad de Oviedo/Universidá d'Uviéu

BCAM, May 20-24 2019


# Index course

1. Hamiltonian mechanics and numerical methods
2. Symplectic integration
3. Modified equations
4. **Constrained mechanical systems**
5. Adaptive geometric integrators

# Motivation

# Motivation

In the setting of applications, we will need to consider generalizations of the traditional $N$-body problem in which the basic modeling unit is not the point particle moving in Euclidean space but **an object moving in some constrained space**.

# Motivation

In the setting of applications, we will need to consider generalizations of the traditional $N$-body problem in which the basic modeling unit is not the point particle moving in Euclidean space but **an object moving in some constrained space**.

For instance, in molecular dynamics, the *bond stretch* between two atoms is typically modeled by a spring with rest length $L > 0$ as

$$U(q^1, q^2) = \frac{\alpha}{2} \left( \|q^1 - q^2\| - L \right)^2,$$

where $q^1$ and $q^2$ are the positions of the atoms and $\alpha$ is a positive parameter.

When $\alpha$ is large, the vibrational frequency is also large, while the variation in the length of the stretch from $L$ will typically be small. It is common practice to replace one or more of these bonds by **rigid length constraints**

$$\|q^1 - q^2\|^2 = L^2$$

# Motivation

## Why do we need constraints? Main advantages

- The maximal step size permitting an accurate integration is related to the frequency of the fastest motions in the system. Constraining *bond* lengths permits an **increase of the simulation time step**.

# Motivation

## Why do we need constraints? Main advantages

- The maximal step size permitting an accurate integration is related to the frequency of the fastest motions in the system. Constraining *bond* lengths permits an **increase of the simulation time step**.
- When the range of motional frequencies in a molecular system is very broad, the exchange of energy between the high- and low-frequency modes may be slow. The application of constraints remedies this problem by **narrowing the range of frequencies in the system**.

# Motivation

## Why do we need constraints? Main advantages

- The maximal step size permitting an accurate integration is related to the frequency of the fastest motions in the system. Constraining *bond* lengths permits an **increase of the simulation time step**.
- When the range of motional frequencies in a molecular system is very broad, the exchange of energy between the high- and low-frequency modes may be slow. The application of constraints remedies this problem by **narrowing the range of frequencies in the system**.
- If the frequency of a mode is larger than $k_B T/h$, the vibration must be treated quantum mechanically. Then, **treating bonds as constraints is a better approximation of their behavior than treating them as classical harmonic oscillators**.

# Motivation

We discuss now the *problem of simulating a mechanical system subject to one or several constraints*.

We will restrict ourselves to the treatment of constraints **which can be described by algebraic relations among the position variables of the system**, i.e., defined by equations of the form

$$g_i(\mathbf{q}) = 0, \quad i = 1, \ldots, m,$$

for smooth functions $g_i$.

A mechanical system subject to such constraints is typically called **holonomic**.
The derivation of the equations of motion for a holonomically constrained mechanical system is not more complicated than for an unconstrained system.

# Motivation

On the other hand, when it comes to numerical discretization, **the constraints introduce some challenges**:

- The propagation of errors in numerical algorithms for a constrained differential equation is more complicated than for ordinary differential equations.

- The constraints are often an intrinsic component of the modeling of the system, and the *configuration manifold* (the set of points for which the constraints are satisfied) is an essential part of the extension of the concept of symplecticness.

# Motivation

On the other hand, when it comes to numerical discretization, **the constraints introduce some challenges**:

- The propagation of errors in numerical algorithms for a constrained differential equation is more complicated than for ordinary differential equations.

- The constraints are often an intrinsic component of the modeling of the system, and the *configuration manifold* (the set of points for which the constraints are satisfied) is an essential part of the extension of the concept of symplecticness.

For these reasons, it seems important that **the constraints are accurately resolved at each step**.
This is in contrast to the case for unconstrained systems, where *a certain error growth is generally tolerated*.

# $N$-body systems with holonomic constraints

# $N$-body systems with holonomic constraints

Given $m$ algebraic constraints $g_i(\mathbf{q}) = 0$, $i = 1, \ldots, m$, on a multiparticle system, it can be shown that **the constraint forces due to each constraint act in the normal direction to the corresponding surface**.

The **equations of motion** have the following form:

$$\begin{aligned} \frac{d}{dt}\mathbf{q} &= M^{-1}\mathbf{p}, \\ \frac{d}{dt}\mathbf{p} &= -\nabla_{\mathbf{q}} U(\mathbf{q}) - \sum_{i=1}^{m} \nabla_{\mathbf{q}} g_i(\mathbf{q})\lambda_i, \\ g_i(\mathbf{q}) &= 0, \quad i = 1, \ldots, m, \end{aligned} \tag{1}$$

where $\lambda_i$, for $i = 1, \ldots, m$, are the corresponding **Lagrange multipliers**.

# $N$-body systems with holonomic constraints

We define the **configuration manifold** $\mathcal{M}$ as the space of all positions subject to the position constraints:

$$\mathcal{M} = \{\mathbf{q} \in \mathbb{R}^d | g_i(\mathbf{q}) = 0, \quad i = 1, \ldots, m\}.$$

# $N$-body systems with holonomic constraints

Let $\bar{\mathbf{q}}$ be a point of the configuration manifold $\mathcal{M}$, and consider the set of all smooth parameterized curves containing $\bar{\mathbf{q}}$ and lying in $\mathcal{M}$.
Each such parameterized curve $\mathbf{q}(t)$ (with $\mathbf{q}(t_0) = \bar{\mathbf{q}}$) has a certain momentum vector $\bar{\mathbf{p}}$ at $t = t_0$, $\bar{\mathbf{p}} = M\dot{\mathbf{q}}(t_0)$.
Obviously, because of $g_i(\mathbf{q}(t)) = 0$ for all $t$, we must have that

$$\frac{d}{dt} g_i(\mathbf{q}(t)) = \nabla_{\mathbf{q}} g_i(\mathbf{q}(t)) \cdot \dot{\mathbf{q}}(t) = \nabla_{\mathbf{q}} g_i(\mathbf{q}(t)) \cdot M^{-1}\mathbf{p}(t) = 0,$$

and, in particular,

$$\nabla_{\mathbf{q}} g_i(\bar{\mathbf{q}}) \cdot M^{-1}\bar{\mathbf{p}} = 0. \tag{2}$$

The set of all possible momentum vectors at the point $\bar{\mathbf{q}}$ is a linear vector space

$$T_{\bar{\mathbf{q}}}\mathcal{M} = \{\mathbf{p} \in \mathbb{R}^D | \nabla_{\mathbf{q}} g_i(\bar{\mathbf{q}}) \cdot M^{-1}\mathbf{p} = 0, \quad i = 1, \ldots, m\}$$

called the **tangent space** at $\bar{\mathbf{q}}$.

# $N$-body systems with holonomic constraints

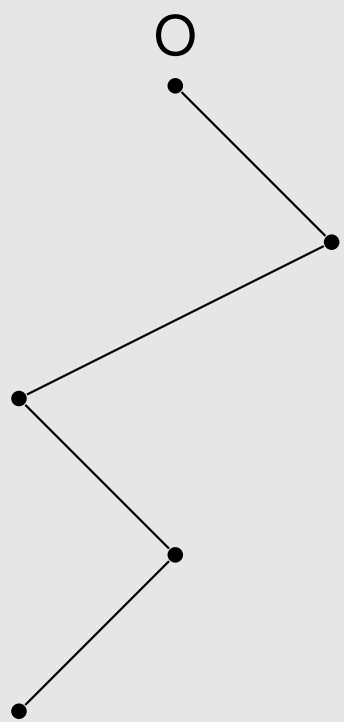
The **tangent bundle** of $\mathcal{M}$ is the space of all pairs $(\mathbf{q}, \mathbf{p})$ with $\mathbf{q}$ lying in $\mathcal{M}$ and $\mathbf{p}$ lying in $T_{\mathbf{q}}\mathcal{M}$. The tangent bundle is denoted $T\mathcal{M}$.

The system (1) can be written more compactly by introducing the vector function $g(\mathbf{q}) = (g_1(\mathbf{q}), \ldots, g_m(\mathbf{q}))^T$, denoting its Jacobian matrix by $G(\mathbf{q})^T = -\nabla_{\mathbf{q}} g(\mathbf{q})$ and letting $\lambda$ represent the $m$-vector of multipliers $\lambda = (\lambda_1, \ldots, \lambda_m)^T$:

$$\begin{aligned} \frac{d}{dt}\mathbf{q} &= M^{-1}\mathbf{p}, \\ \frac{d}{dt}\mathbf{p} &= F(\mathbf{q}) + G(\mathbf{q})^T\lambda, \\ g(\mathbf{q}) &= 0. \end{aligned} \tag{3}$$

# $N$-body systems with holonomic constraints

## Example[1]




Consider a simple constraint chain in the plane made up of $N$-point particles with masses $m_1, m_2, \ldots, m_N$, with each successive pair of particles joined by a length constraint with lengths $l_1, l_2, \ldots, l_{N-1}$. If the first point is linked to the origin by a similar length constraint (with length $l_0$), then we have a multiple pendulum as in the Figure.

[1] Leimkuhler and Reich, *Simulating Hamiltonian dynamics*, 2004.

# $N$-body systems with holonomic constraints

## Example (Continuation)

Let us number the Lagrange multipliers in accordance with the indexing of the constraints. Then the equations of motion are, for any of the internal nodes of the chain,

$$\frac{d}{dt}q^i = M^{-1}p^i,$$
$$\frac{d}{dt}p^i = F(q^i) + \lambda_{i-1}(q^i - q^{i-1}) + \lambda_i(q^i - q^{i+1}).$$

# $N$-body systems with holonomic constraints

## Example (Continuation)

On the other hand, the first node obeys

$$\frac{d}{dt}q^1 = M^{-1}p^1,$$
$$\frac{d}{dt}p^1 = F(q^1) + \lambda_0 q^1 + \lambda_1(q^1 - q^2),$$

and the last moves according to

$$\frac{d}{dt}q^N = M^{-1}p^N,$$
$$\frac{d}{dt}p^N = F(q^N) + \lambda_{N-1}(q^N - q^{N-1}).$$

# $N$-body systems with holonomic constraints

## Example (Continuation)

Here the constraints take the form

$$g_i(\mathbf{q}) = \frac{1}{2}\left(\|q^i - q^{i+1}\|^2 - l_i^2\right) = 0, \quad i = 1, \ldots, N-1,$$

while $g_0(\mathbf{q}) = \frac{1}{2}\left(\|q^1\|^2 - l_0^2\right) = 0$. The transpose of the constraint Jacobian matrix, $G(\mathbf{q})^T$ is

$$\begin{bmatrix} q^1 & q^1 - q^2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & q^2 - q^1 & q^2 - q^3 & \cdots & 0 & 0 & 0 \\ 0 & 0 & q^3 - q^2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & q^{N-1} - q^{N-2} & q^{N-1} - q^N \\ 0 & 0 & 0 & \cdots & 0 & 0 & q^N - q^{N-1} \end{bmatrix}.$$

We remark that, for simplicity, *we use the negative constraint Jacobian matrix*.

# Numerical methods

# Numerical methods

We now consider the **development of numerical integration methods suitable for integrating a constrained mechanical system**.

# Numerical methods

We now consider the **development of numerical integration methods suitable for integrating a constrained mechanical system**.

The obvious **geometric property that we would like to preserve is the constraint** $g(\mathbf{q}) = 0$**.** However, any such trajectory will also need to **satisfy the tangency condition** $\nabla_{\mathbf{q}} g_i(\bar{\mathbf{q}}) \cdot M^{-1}\bar{\mathbf{p}} = 0$**.**

# Numerical methods

We now consider the **development of numerical integration methods suitable for integrating a constrained mechanical system**.

The obvious **geometric property that we would like to preserve is the constraint** $g(\mathbf{q}) = 0$. However, any such trajectory will also need to **satisfy the tangency condition** $\nabla_{\mathbf{q}} g_i(\bar{\mathbf{q}}) \cdot M^{-1}\bar{\mathbf{p}} = 0$.

Thus, we are looking for methods for **solving the constrained equations of motion which preserve the tangent bundle** $T\mathcal{M}$.

# Numerical methods

Methods for constrained integration can be divided into two big classes:

- Methods based on **integration of some related (unconstrained) ordinary differential equation**.

- Methods based on **direct discretization of the constrained equations of motion**.

Here we consider methods of the second class.

# Numerical methods

- **SHAKE** discretization was proposed in 1976 by Ryckaert, Ciccotti, and Berendsen[2].
- A paper of Andersen[3] later introduced a related formulation called **RATTLE**.
- Leimkuhler and Skeel[4] analyzed and compared the two methods, discovering, apparently for the first time, that they were (i) **equivalent to each other** and (ii) **symplectic**.

*Our treatment is based on the exposition of Leimkuhler and Skeel*.

---

[2] Ryckaert, Ciccotti, and Berendsen, "Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes", 1977.

[3] Andersen, "Rattle: A "velocity" version of the shake algorithm for molecular dynamics calculations", 1983.

[4] Leimkuhler and Skeel, "Symplectic Numerical Integrators in Constrained Hamiltonian Systems", 1994.

# Numerical methods

Rewriting the constrained equations of motion as a single second-order equation in $\mathbf{q}$ and applying the standard leapfrog discretization, one quite naturally get the SHAKE discretization

$$\begin{aligned}
\mathbf{p}_{n+1/2} &= \mathbf{p}_{n-1/2} + hF(\mathbf{q}_n) + hG(\mathbf{q}_n)^T\lambda_n, \\
\mathbf{q}_{n+1} &= \mathbf{q}_n + hM^{-1}\mathbf{p}_{n+1/2}, \\
g(\mathbf{q}_{n+1}) &= 0.
\end{aligned}$$

If we further define $\mathbf{p}_n = (\mathbf{q}_{n+1} - \mathbf{q}_{n-1})/(2h)$, we obtain

$$\begin{aligned}
\mathbf{p}_{n+1/2} &= \mathbf{p}_n + \frac{h}{2}F(\mathbf{q}_n) + \frac{h}{2}G(\mathbf{q}_n)^T\lambda_n, \\
\mathbf{q}_{n+1} &= \mathbf{q}_n + hM^{-1}\mathbf{p}_{n+1/2}, \\
g(\mathbf{q}_{n+1}) &= 0, \\
\mathbf{p}_{n+1} &= \mathbf{p}_{n+1/2} + \frac{h}{2}F(\mathbf{q}_{n+1}) + \frac{h}{2}G(\mathbf{q}_{n+1})^T\lambda_{n+1}.
\end{aligned} \tag{4}$$

SHAKE can be viewed as a mapping of $\mathcal{M}$, but **it does not define a mapping of the tangent bundle** $T\mathcal{M}$.

# Numerical methods

The RATTLE method, is algebraically equivalent to SHAKE, is a **mapping of the tangent bundle** $T\mathcal{M}$.

RATTLE corrects the SHAKE solution so that it lies on $T\mathcal{M}$ through appropriate projection of the momenta $\mathbf{p}_{n+1}$ on to the tangency constraint (2). The result is the following scheme:

$$
\begin{aligned}
\mathbf{p}_{n+1/2} &= \mathbf{p}_n + \frac{h}{2}F(\mathbf{q}_n) + \frac{h}{2}G(\mathbf{q}_n)^T\lambda_n^{(r)}, \\
\mathbf{q}_{n+1} &= \mathbf{q}_n + hM^{-1}\mathbf{p}_{n+1/2}, \\
g(\mathbf{q}_{n+1}) &= 0, \\
\mathbf{p}_{n+1} &= \mathbf{p}_{n+1/2} + \frac{h}{2}F(\mathbf{q}_{n+1}) + \frac{h}{2}G(\mathbf{q}_{n+1})^T\lambda_{n+1}^{(v)}, \\
G(\mathbf{q}_{n+1})\mathbf{p}_{n+1} &= 0.
\end{aligned}
$$

# Numerical methods

The multipliers $\lambda_n^{(r)}$ are **chosen in order to enforce the position constraints**, while $\lambda_{n+1}^{(v)}$ **relates to the velocity constraints** and is determined at time $t_n$ by the linear system

$$\left(G(\mathbf{q}_n)G(\mathbf{q}_n)^T\right)\lambda_n^{(v)} = G(\mathbf{q}_n)\left(-\frac{2}{h}\mathbf{p}_{n-1/2} - F(\mathbf{q}_n)\right). \tag{5}$$

Combining the update of $\mathbf{p}_{n+1/2}$ with

$$\mathbf{p}_n = \mathbf{p}_{n-1/2} + \frac{h}{2}F(\mathbf{q}_n) + \frac{h}{2}G(\mathbf{q}_n)^T\lambda_n^{(v)},$$

results in

$$\mathbf{p}_{n+1/2} = \mathbf{p}_{n-1/2} + hF(\mathbf{q}_n) + \frac{h}{2}G(\mathbf{q}_n)^T\left(\lambda_n^{(r)} + \lambda_n^{(v)}\right), \tag{6}$$

where $\lambda_n^{(v)}$ is determined by (5) and is assumed to be computed at this stage. On the other hand, $\lambda_n^{(r)}$ is chosen so that

$$g(\mathbf{q}_{n+1}) = g(\mathbf{q}_n + hM^{-1}\mathbf{p}_{n+1/2}) = 0$$

at the next time level.

# Numerical methods

One can recognize that (6) is equivalent to the SHAKE update with $\lambda_n$ replaced by $\frac{1}{2}\left(\lambda_n^{(r)} + \lambda_n^{(v)}\right)$.

Thus, RATTLE and SHAKE are formally equivalent when viewed as iterations from $(\mathbf{q}_n, \mathbf{p}_{n-1/2})$ to $(\mathbf{q}_{n+1}, \mathbf{p}_{n+1/2})$.

# Numerical methods

One can recognize that (6) is equivalent to the SHAKE update with $\lambda_n$ replaced by $\frac{1}{2}\left(\lambda_n^{(r)} + \lambda_n^{(v)}\right)$.

Thus, RATTLE and SHAKE are formally equivalent when viewed as iterations from $(\mathbf{q}_n, \mathbf{p}_{n-1/2})$ to $(\mathbf{q}_{n+1}, \mathbf{p}_{n+1/2})$.

In other words, **a proper initialization of RATTLE would produce a sequence of approximations identical to that produced by SHAKE, except that the two solutions would differ in the momentum approximation $\mathbf{p}_n$ at the end of each time step**.

# Numerical methods

Since SHAKE and RATTLE are formally equivalent, **the propagation of errors is identical for the two methods**.

The modification of the end velocity update in the RATTLE step does not effect the stability or global convergence of the method in any way.

It is easy to derive the local error introduced in a single time step using SHAKE or RATTLE using the Taylor expansions of the solution and numerical solution in a step of size $h$, starting from some given point $(\mathbf{q}_n, \mathbf{p}_n) \in T\mathcal{M}$.
For both methods, this calculation shows that the local error is $\mathcal{O}(h^3)$.
Thus, **both methods are second-order accurate**.

# Numerical methods

At every step of the SHAKE (or RATTLE) discretization, **we need to solve a system of nonlinear equations of dimension equal to the number of constraints**.

In some cases, these constraints can be dealt with very easily. For instance, if the constraints are linear, the work involved is usually a step of Gaussian elimination or the use of some other linear solver, and unless the dimension of the system is exceptionally large, or the equations poorly conditioned, this computation will be easy to implement.

In other cases, the constraints may admit a decoupling that enables their simplified solution.

# Numerical methods

## Example[5]

Consider a system of $N$ particles, each of mass $m$, attached to the surface of the unit sphere and interacting in some homogeneous two-body potential.

[5] Leimkuhler and Reich, *Simulating Hamiltonian dynamics*, 2004.

[6] We change the notation of $\lambda$ here, the super index denotes the component, while the sub index is used for the discretization.

# Numerical methods

## Example[5]

Consider a system of $N$ particles, each of mass $m$, attached to the surface of the unit sphere and interacting in some homogeneous two-body potential.
The equations of motion for the $i$-th particle, $i = 1, \ldots, N$, are[6]

$$\begin{aligned} \frac{d}{dt} q^i &= M^{-1} p^i, \\ \frac{d}{dt} p^i &= F(q^i) - q^i \lambda^i, \\ \|q^i\|^2 &= 1. \end{aligned}$$

[5] Leimkuhler and Reich, *Simulating Hamiltonian dynamics*, 2004.

[6] We change the notation of $\lambda$ here, the super index denotes the component, while the sub index is used for the discretization.

## Example (Continuation)

Applying SHAKE discretization results in

$$\begin{aligned} p^i_{n+1/2} &= p^i_{n-1/2} + hF(q^i_n) + hG(q^i_n)^T \lambda^i_n, \\ q^i_{n+1} &= q^i_n + hM^{-1} p^i_{n+1/2}, \\ \|q^i_{n+1}\|^2 &= 1. \end{aligned}$$

# Numerical methods

## Example (Continuation)

Applying SHAKE discretization results in

$$\begin{aligned} p^i_{n+1/2} &= p^i_{n-1/2} + hF(q^i_n) + hG(q^i_n)^T \lambda^i_n, \\ q^i_{n+1} &= q^i_n + hM^{-1} p^i_{n+1/2}, \\ \|q^i_{n+1}\|^2 &= 1. \end{aligned}$$

Introducing $p^i_{n+1/2}$ in $q^i_{n+1}$ and then combining the result with the constraint $\|q^i_{n+1}\|^2 = 1$ results in $N$ **independent quadratic equations to be solved for the $N$ Lagrange multipliers**.

# Numerical methods

On the other hand, in most cases, such as for example the constraint chain, one has to solve a system of nonlinear equations for the multipliers of the form

$$g\left(\bar{\mathbf{q}}_{n+1} + h^2 \sum_{i=1}^{m} G(\mathbf{q}_n)^T \lambda_n\right) = 0, \tag{7}$$

where $\bar{\mathbf{q}}$ represents an unconstrained step using leapfrog.

The original SHAKE paper[7] also provided an iterative solver[8] for the nonlinear equations (7). In SHAKE iteration, we cycle through the constraints, adjusting one multiplier at each iteration.

[7] Ryckaert, Ciccotti, and Berendsen, "Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes", 1977.

[8] The term SHAKE is used typically to refer to the combined procedure of time discretization together with the iterative (*coordinate resetting*) algorithm for constraints.

# Numerical methods

The iteration is as follows:

1. We initialize

$$\mathbf{Q} := \bar{\mathbf{q}}_{n+1} = \mathbf{q}_n + hM^{-1}\mathbf{p}_{n-1/2} + h^2 F(\mathbf{q}),$$

which is equivalent to taking $\lambda_n$ to be zero in (7).

# Numerical methods

The iteration is as follows:

1. We initialize

$$\mathbf{Q} := \bar{\mathbf{q}}_{n+1} = \mathbf{q}_n + hM^{-1}\mathbf{p}_{n-1/2} + h^2 F(\mathbf{q}),$$

   which is equivalent to taking $\lambda_n$ to be zero in (7).

2. We cycle through the list of constraints and correct each constraint one after another by the following procedure. For $i = 1, \ldots, m$, compute an offset $\Delta\Lambda_i$ in order to satisfy the $i$-th linearized constraint equation

$$\Delta\Lambda_i := \frac{g_i(\mathbf{Q})}{G_i(\mathbf{Q})G_i(\mathbf{q}_n)},$$

   and update $\mathbf{Q}$ by

$$\mathbf{Q} := \mathbf{Q} + G_i(\mathbf{q})^T \Delta\Lambda_i.$$

# Numerical methods

**This cycle is repeated until all constraint residuals $g_i(\mathbf{Q})$ are smaller than some prescribed tolerance**.

At this point we set $\mathbf{q}_{n+1} = \mathbf{Q}$ and continue with the next time step.

As was shown in Barth et al., “Algorithms for constrained molecular dynamics”, 1995, **SHAKE iteration is really a variant of nonlinear Gauss-Seidel-Newton iteration**.
It can be shown that, given a good enough initial guess for the multiplier, or a small enough step size $h$, **this iterative method eventually converges**.

# Hamiltonian mechanics and the symplectic structure

# Hamiltonian mechanics and the symplectic structure

From the constrained equations of motion we can write the following augmented Hamiltonian

$$\tilde{H}(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\mathbf{q}) + g(\mathbf{q})^T\lambda.$$

# Hamiltonian mechanics and the symplectic structure

From the constrained equations of motion we can write the following augmented Hamiltonian

$$\tilde{H}(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\mathbf{q}) + g(\mathbf{q})^T\lambda = H(\mathbf{q},\mathbf{p}) + g(\mathbf{q})^T\lambda.$$

# Hamiltonian mechanics and the symplectic structure

From the constrained equations of motion we can write the following augmented Hamiltonian

$$\tilde{H}(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\mathbf{q}) + g(\mathbf{q})^T\lambda = H(\mathbf{q},\mathbf{p}) + g(\mathbf{q})^T\lambda.$$

Thus, the equations of motion can be written in the usual way for Hamiltonian systems

$$\begin{aligned} \frac{d}{dt}\mathbf{q} &= \frac{\partial H(\mathbf{q},\mathbf{p})}{\partial \mathbf{p}}, \\ \frac{d}{dt}\mathbf{p} &= -\frac{\partial H(\mathbf{q},\mathbf{p})}{\partial \mathbf{q}} + G(\mathbf{q})^T\lambda, \\ g(\mathbf{q}) &= 0. \end{aligned} \tag{8}$$

# Hamiltonian mechanics and the symplectic structure

The previous system is *a generalization of unconstrained Hamiltonian systems*.

According to a perspective due to Dirac[9], *the flow of a Hamiltonian system on a manifold can be embedded in the flow of an unconstrained Hamiltonian system*.

The solutions of the previous system can be viewed as evolving in the phase space of a standard Hamiltonian system with **extended Hamiltonian**

$$\hat{H}(\mathbf{q}, \mathbf{p}) = H(\mathbf{q}, \mathbf{p}) + g(\mathbf{q})^T \Lambda(\mathbf{q}^{-1}, \mathbf{p}),$$

for a certain $\Lambda(\mathbf{q}^{-1}, \mathbf{p})$. Using this setting, the **preservation of the wedge product can be proved**[10].

[9] For a discussion of Dirac's method, see Marsden and Ratiu, *Introduction to Mechanics and Symmetry*, 1999

[10] Section 7.4 in Leimkuhler and Reich, *Simulating Hamiltonian dynamics*, 2004.

# Hamiltonian mechanics and the symplectic structure

**Once the symplecticness of the constrained system is ensured, we want to see the symplecticness of SHAKE and RATTLE.**

# Hamiltonian mechanics and the symplectic structure

**Once the symplecticness of the constrained system is ensured, we want to see the symplecticness of SHAKE and RATTLE.**

We will follow the ideas of Leimkuhler and Skeel, “Symplectic Numerical Integrators in Constrained Hamiltonian Systems”, 1994.

# Hamiltonian mechanics and the symplectic structure

Let us consider again the Hamiltonian

$$\tilde{H}(\mathbf{q},\mathbf{p}) = H(\mathbf{q},\mathbf{p}) + g(\mathbf{q})^T\lambda.$$

Let us also assume that a second-order, symplectic, and symmetric integrator $\psi_h$ is known for the unconstrained system (for instance velocity Verlet). Then, the following method is a constraint-preserving, symplectic, symmetric, and second-order method for the constrained Hamiltonian system (with Hamiltonian $\tilde{H}$):

$$\begin{aligned}
\bar{\mathbf{p}}_n &= \mathbf{p}_n + \frac{h}{2}G(\mathbf{q}_n)^T\lambda_n^{(r)},\\
(\mathbf{q}_{n+1},\bar{\mathbf{p}}_{n+1}) &= \psi_h(\mathbf{q}_n,\bar{\mathbf{p}}_n),\\
g(\mathbf{q}_{n+1}) &= 0,\\
\mathbf{p}_{n+1} &= \bar{\mathbf{p}}_{n+1} + \frac{h}{2}G(\mathbf{q}_{n+1})^T\lambda_{n+1}^{(v)},\\
G(\mathbf{q}_{n+1})\mathbf{p}_{n+1} &= 0.
\end{aligned}$$

# Hamiltonian mechanics and the symplectic structure

Note that this method reduces to RATTLE taking velocity Verlet as $\psi_h$.

Let us now verify the geometric properties:

[11] Leimkuhler and Reich, *Simulating Hamiltonian dynamics*, 2004.

# Hamiltonian mechanics and the symplectic structure

Note that this method reduces to RATTLE taking velocity Verlet as $\psi_h$.

Let us now verify the geometric properties:

- The method is **consistent**.
- The method is **symplectic**. By the symplecticness of $\psi_h$, we know

$$d\mathbf{q}_n \wedge d\bar{\mathbf{p}}_n = d\mathbf{q}_{n+1} \wedge d\bar{\mathbf{p}}_{n+1}.$$

  Moreover, both momentum maps are also symplectic as a consequence of[11]: $d\mathbf{q} \wedge (G(\mathbf{q})^T \lambda) = 0$ for a differential 1-form $d\mathbf{q}$ satisfying $G(\mathbf{q})d\mathbf{q} = 0$.
- The method is **symmetric** as easily shown by replacing $h$ by $-h$ and $(\mathbf{q}_n, \mathbf{p}_n, \lambda_n^{(r)})$ by $(\mathbf{q}_{n+1}, \mathbf{p}_{n+1}, \lambda_{n+1}^{(v)})$.

[11] Leimkuhler and Reich, *Simulating Hamiltonian dynamics*, 2004.

# Hamiltonian mechanics and the symplectic structure

**The equivalent results have been proved for two-stage integrators in Fernández-Pendás, Akhmatskaya, and Sanz-Serna, "Adaptive multi-stage integrators for optimal energy conservation in molecular simulations", 2016.**

# Next session

- We will present **sampling techniques** which, combined symplectic numerical integrators, allow us to perform numerical dynamics simulations.

- We will discuss the idea of **adaptive integrators** showing their impact on the performance of realistic simulations.

- Andersen, H. C. “Rattle: A “velocity” version of the shake algorithm for molecular dynamics calculations”. In: *Journal of Computational Physics* 52.1 (1983), pp. 24–34.
- Barth, E. et al. “Algorithms for constrained molecular dynamics”. In: *Journal of Computational Chemistry* 16.10 (1995), pp. 1192–1209.
- Fernández-Pendás, M., E. Akhmatskaya, and J. M. Sanz-Serna. “Adaptive multi-stage integrators for optimal energy conservation in molecular simulations”. In: *Journal of Computational Physics* 327 (2016), pp. 434–449.
- Leimkuhler, B. J. and S. Reich. *Simulating Hamiltonian dynamics*. Cambridge monographs on applied and computational mathematics. Cambridge: Cambridge University, 2004.
- Leimkuhler, B. J. and R. D. Skeel. “Symplectic Numerical Integrators in Constrained Hamiltonian Systems”. In: *Journal of Computational Physics* 112.1 (1994), pp. 117–125.

- Marsden, J. E. and T. Ratiu. *Introduction to Mechanics and Symmetry*. Texts in Applied Mathematics vol 17. Springer-Verlag, New York, 1999.
- Ryckaert, J.-P., 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”. In: *Journal of Computational Physics* 23.3 (1977), pp. 327–341.