6 Appendix

6.1 Calculus of the jacobian matrix

6.1.1 Trapezoidal Rule Jacobian

6.1.2 Back Euler Jacobian

6.1.3 Midpoint Euler Jacobian

6.1.4 Energy Momentum Jacobian
List of Figures

2.1 Reference mechanical system sketch. ........................................ 14
3.1 Geometrical variables for plane surfaces. ............................. 17
3.2 Karush-Kuhn-Tucker conditions. .......................................... 18
3.3 Projection of $q_i$ onto surface $S$. ..................................... 20
3.4 Representation of covariant basis $n_i$ and contravariant basis $n^i$. 22
3.5 Projection of $q_i$ onto the shifted surface $S^*$. .................... 26
4.1 Toy problem 1. .................................................................... 28
4.2 Particle trajectories for different values of the penalty ($\Delta t = 0.05, t = 4$). ................................................................. 29
4.3 Particle two trajectory for different time-steps ($\epsilon = 10^3$). .... 30
4.4 Particle two trajectory integrated using the penalty method for different time-steps and algorithms. ............................................. 31
4.5 Total energy of the system for different values of the penalty ($\Delta t = 0.005, t = 4$). ................................................................. 32
4.6 Total energy of the system for different time-steps ($\epsilon = 10^3$). 33
4.7 System's total energy using the penalty method for different time-steps and algorithms. ......................................................... 34
4.8 Total linear momentum of the system for different values of the penalty ($\Delta t = 0.05, t = 4$). ......................................................... 35
4.9 Total linear momentum of the system for different time-steps ($\epsilon = 10^3$). ................................................................. 35
4.10 System's linear momentum energy using the penalty method for different time-steps and algorithms. .................................. 36
4.11 Trajectory of $m_2$ using the Lagrange multipliers method for different time-steps and algorithms. ................................................. 37
4.12 Benefits of introducing a new gap. ........................................... 38
4.13 Total energy of the system using the Lagrange multipliers method for different time-steps and algorithms. ...................... 39
4.14 Long time integration using the Lagrange multipliers method for $\Delta t = 0.01$. ................................................................. 39
4.15 Reaction force $\lambda$ calculated using the Lagrange multipliers method for different time-steps and algorithms. .................. 40
4.16 Total linear momentum of the system calculated using the Lagrange multipliers method for different time-steps and algorithms. 41
4.17 Trajectory of $m_2$ integrated with the Störmer-Verlet scheme and using the penalty method. .................................................. 42
4.18 Trajectory of $m_2$ integrated with the Störmer-Verlet scheme and using the Lagrange multipliers method. ...................... 43
4.19 Total energy of the system integrated with the Störmer-Verlet scheme and using the penalty method. .......................... 44
4.20 Total energy of the system integrated with the Störmer-Verlet scheme and using the penalty method. ...................... 44
4.21 Trajectory of $m_2$ using the dissipative penalty method for different values of $\epsilon$. ................................................................. 45
4.22 Trajectory of $m_2$ using the dissipative penalty method for different values of $\epsilon$. ................................................................. 46
4.23 Linear momentum evolution using the dissipative penalty method for different values of $\epsilon$. ................................................................. 47
4.24 Trajectory of $m_2$ using the modified Lagrange multipliers for different time-steps. .......................... 48
4.25 Detail of the surface violation using the modified Lagrange multipliers for different time-steps. .................. 48
4.26 Energy evolution using the modified Lagrange multipliers method for $\Delta t = 0.1$. ............................... 49
4.27 Linear momentum evolution using the modified Lagrange multipliers method for different time-steps. ......... 49
4.28 Toy problem 2. ............................................. 50
4.29 Sketch of the trajectory approximation for corners. ........... 51
4.30 Trajectory of collision into a corner integrated for different time-steps. ........................................... 52
4.31 Particle $m_2$ speed after a collision into a corner integrated for different time-steps. ............................... 52
4.32 Long-time integration of the toy problem 2 using averages. .... 53
4.33 Sketch of toy problem 3. ................................. 54
4.34 Total energy of the icosahedron system for different integration methods. ......................................... 54
4.35 Position of the icosahedron for different times and integrated with the modified Lagrange multipliers method. .... 55
5.1 Geometrical variables for impacts between particles. ........ 58

List of Tables

2.1 Time-integration algorithms for non-contact particle mechanics. 15
2.2 Jacobian matrix for non-contact time-integration algorithms. .... 15
2.3 Time integration conserving properties. ................................ 16
3.1 Residual contribution depending on the impact method. .......... 27
4.1 Convergence for different time-steps $\Delta t$ and for different values of the penalty $\epsilon$. ............................ 46
1 Introduction

1.1 Motivation

The rapid development of computing systems has made possible to recreate and predict physical processes on computers. Moreover, simulations allow to study processes that cannot be analyzed directly through experiments. The fact that costly experiments can be avoided accelerates the development of new products.

The motivation for developing structure-preserving algorithms came independently from different areas of research such as astronomy, molecular dynamics, mechanics, theoretical physics, and numerical analysis as well as from other areas of both applied and pure mathematics. It results that the preservation of geometric properties of the flow not only produces an improved qualitative behavior, but also permits a more accurate long time integration than with general-purpose methods.

Nowadays, impact algorithms do not provide global solutions. Their implementation involve complex problems, such as a wide casuistry range and a non-assured stability when non-linear differential equations are treated.

Then, the geometric numerical integration is an interesting way to tackle the issue owing to the fact that the invariants preservation of the system permits to check the goodness of the integration.

1.2 State of the art

In the last few decades, the theory of numerical methods for general ordinary differential equations has reached a certain maturity, and excellent general-purpose codes, mainly based on Runge-Kutta methods or linear multi-step methods, have become available. Nevertheless, non-linear differential equations can exhibit very complicated behavior over extended time intervals and even the fundamental questions of existence, uniqueness, and extendability of solutions are non solved issues.

In this line, some frictionless finite-element methods have been proposed for rigid bodies. The basis of those methods is the conservation of the total energy of the system. Armero developed in [2] a conserving penalty method focused on rigid bodies. In addition, some numerical schemes using the Lagrange multipliers methods are available in [5, 3].

Finally, there are available in [6] the Mortar methods focused on softening boundaries for solids. Nevertheless, these methods can not be extrapolated to our case.
2 Time integration algorithms for conservative particle systems

2.1 Mathematical background

2.1.1 Newton’s equations

Consider a system of \( N \) particles in a three dimensional space, each with position vector \( q_i(t) \) for \( i = 1, ..., N \). Note that each \( q_i(t) \in \mathbb{R}^3 \) is a 3-vector. We thus need \( 3N \) coordinates to specify the system, this is the configuration space. Newton’s 2nd law tells us that the equation of motion for the \( i \)th particle is

\[
\dot{p}_i = F^\text{ext}_i + F^c_i \tag{2.1.1}
\]

for \( i = 1, ..., N \). Where \( \dot{p}_i = m_i v_i \) is the linear momentum of the \( i \)th particle and \( v_i = \dot{q}_i \) is its velocity. We decompose the total force on the \( i \)th particle into an external force \( F^\text{ext}_i \) and a constraint force \( F^c_i \).

2.1.2 Holonomic constraints

By a constraint on a particle, we understand that the particle’s motion is limited in some rigid way.

Definition 1 (Holonomic constraints). For a system of particles with positions given by \( r_i(t) \) for \( i = 1, ..., N \), constraints that can be expressed in the form

\[
g(r_1, ..., r_N, t) = 0, \tag{2.1.2}
\]

are said to be holonomic. Note that they only involve the configuration coordinates.

We will only consider, now, systems for which the constraints are holonomic.

2.1.3 Degrees of freedom

Let us suppose that for the \( N \) particles there are \( m \) holonomic constraints given by Equation 2.1.2. The positions \( r_i(t) \) of all \( N \) particles are determined by \( 3N \) coordinates. However, due to the constraints, the positions \( r_i(t) \) are not all independent. In principle, we can use the \( m \) holonomic constraints to eliminate \( m \) of the \( 3N \) coordinates and we would be left with \( 3N - m \) independent coordinates.

Definition 2 (Degrees of freedom). The dimension of the configuration space is called the number of degrees of freedom.

Thus we can transform the old coordinates \( r_1, ..., r_N \) to new generalized coordinates \( q_1, ..., q_n \) where \( n = 3N - m \):

\[
r_1 = r_1(q_1, ..., q_n, t),
\]

\[
\vdots
\]

\[
r_N = r_N(q_1, ..., q_n, t).
\]
2.1.4 D’Alembert’s principle

We will restrict ourselves to systems for which the net work of the constraint forces is zero. We suppose
\[ \sum_{i=1}^{N} F_i^c \cdot d\mathbf{r}_i = 0, \]  
for every small change \( d\mathbf{r}_i \) of the configuration of the system (for fixed \( t \)). If we combine (2.1.3) with Newton’s 2nd law we obtain the D’Alembert’s principle.
\[ \sum_{i=1}^{N} (\dot{p}_i - F_i^{ext}) \cdot d\mathbf{r}_i = 0 \]  
(2.1.4)

2.1.5 Lagrange’s equations

Consider the transformation to generalized coordinates
\[ q_i^* = q_i^*(q_1, ..., q_n, t), \]
for \( i = 1, ..., N \). If we consider a small increment in the displacements \( dq_i^* \), then the corresponding increment in the work done by the external forces is
\[ \sum_{i=1}^{N} F_i^{ext} \cdot dq_i^* = \sum_{i,j=1}^{N,n} F_i^{ext} \cdot \frac{\partial q_i^*}{\partial q_j} dq_j = \sum_{j=1}^{n} Q_j \cdot dq_j, \]  
(2.1.5)

where \( Q_j \) may be understood as generalized forces
\[ Q_j = \sum_{i=1}^{N} F_i^{ext} \cdot \frac{\partial \mathbf{r}_i}{\partial q_j}. \]

Furthermore, the change in kinetic energy due to the increment in the displacements \( d\mathbf{r}_i \), is given by
\[ \sum_{i=1}^{N} \dot{p}_i \cdot d\mathbf{r}_i = \sum_{i=1}^{N} m_i \dot{\mathbf{v}}_i \cdot d\mathbf{r}_i = \sum_{i,j=1}^{N,n} m_i \dot{\mathbf{v}}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} dq_j \]  
(2.1.6)

Therefore, as documented in [1],
\[ \sum_{i=1}^{N} \dot{p}_i \cdot d\mathbf{r}_i = \sum_{j=1}^{n} \left( \sum_{i=1}^{N} m_i \dot{\mathbf{v}}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} \right) dq_j \]
\[ = \sum_{j=1}^{n} \left( \sum_{i=1}^{N} m_i \left( \frac{d}{dt} \left( m_i \dot{\mathbf{v}}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} \right) - m_i \dot{\mathbf{v}}_i \cdot \frac{\partial \dot{\mathbf{v}}_i}{\partial q_j} \right) \right) dq_j \]
\[ = \sum_{j=1}^{n} \left( \sum_{i=1}^{N} m_i \left( \frac{d}{dt} \left( m_i \dot{\mathbf{v}}_i \cdot \frac{\partial \mathbf{v}_i}{\partial q_j} \right) - m_i \dot{\mathbf{v}}_i \cdot \frac{\partial \dot{\mathbf{v}}_i}{\partial q_j} \right) \right) dq_j \]
\[ = \sum_{j=1}^{n} \left( \frac{d}{dt} \left( \frac{\partial}{\partial q_j} \left( \sum_{i=1}^{N} \frac{1}{2} m_i \dot{v}_i^2 \right) \right) - \frac{\partial}{\partial q_j} \left( \sum_{i=1}^{N} \frac{1}{2} m_i \dot{v}_i^2 \right) \right) dq_j. \]
Hence, if the kinetic energy is defined to be

\[ T := \sum_{i=1}^{N} \frac{1}{2} m_i |v_i|^2, \]

then, we see that D’Alembert’s principle is equivalent to

\[ \sum_{j=1}^{n} \left( \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} - Q_j \right) dq_j = 0. \]

Since the \( q_j \) for \( j = 1, ..., n \), where \( n = 3N - m \), are all independent, we have

\[ \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} - Q_j = 0, \]

for \( j = 1, ..., n \). If we now assume that the work done depends on the initial and final configurations only and not on the path between them, then it exists a potential function \( V(q_1, ..., q_n) \) such that

\[ Q_j = -\frac{\partial V}{\partial q_j}, \]

for \( j = 1, ..., n \) (such forces are said to be conservative). If we define the Lagrange function or Lagrangian to be

\[ L = T - V, \quad (2.1.7) \]

finally, we see that D’Alambert’s principle is equivalent to

\[ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = 0, \quad (2.1.8) \]

for \( j = 1, ..., n \). These are known as Lagrange’s equations. As already noted the \( n \)-dimensional subsurface of 3N-dimensional space, on which the solutions to Lagrange’s equations lie (where \( n = 3N - m \)), is called the configuration space. It is parameterized by the \( n \) generalized coordinates \( q_1, ..., q_n \).

2.1.6 Hamilton’s principle

**Definition 3 (Action).** If the Lagrangian \( L \) is the difference of the kinetic and potential energies for a system, \( L = T - V \), we define the action \( A \) from time \( t_1 \) to \( t_2 \) to be the functional

\[ A(q) := \int_{t_1}^{t_2} L(q, \dot{q}, t) \, dt, \quad (2.1.9) \]

where \( q = (q_1, ..., q_n) \).

**Theorem 1 (Hamilton’s principle of least action).** The correct path of motion of a mechanical system with holonomic constraints and conservative external forces, from time \( t_1 \) to \( t_2 \), is a stationary solution of the action. Indeed, the correct path of motion \( q = q(t) \), with \( q = (q_1, ..., q_n) \), necessarily and sufficiently satisfies Lagrange’s equations of motion.
2.1.7 Hamiltonian dynamics

Let us consider mechanical systems for which the applied forces have a generalized potential. For such systems, we can construct a Lagrangian \( L(q, \dot{q}, t) \), where \( q = (q_1, ..., q_n) \), which is the difference of the total kinetic and potential energies. These mechanical systems evolve according to the \( n \) Lagrange equations (Equation 2.1.8), which are second order ordinary differential equations. Therefore, the system is determined for all time once \( 2n \) initial conditions \((q(t_0), \dot{q}(t_0))\) are specified.

\textbf{Definition 4} (Generalized momenta). \textit{We define the generalized momenta for a Lagrangian mechanical system to be}

\[
p_j = \frac{\partial L}{\partial \dot{q}_j}
\]

for \( j = 1, ..., n \). Note that in general \( p_j = p_j(q, \dot{q}, t) \).

In terms of the generalized momenta, Lagrange’s equations become, for \( j = 1, ..., n \)

\[
\dot{p}_j = \frac{\partial L}{\partial q_j}
\]

Furthermore, it is possible to solve the relations defining the generalized momenta to find \( \dot{q}_j = \dot{q}_j(q, p, t) \).

\textbf{Definition 5} (Hamiltonian). \textit{We define the Hamiltonian function as the Legendre transform of the Lagrangian function, i.e. the Hamiltonian is defined by}

\[
H(q, p, t) := \dot{q} \cdot p - L(q, p, t), \tag{2.1.12}
\]

From Lagrange’s equations of motion and using the definitions for the generalized momenta (Equation 2.1.10) and Hamiltonian (Equation 2.1.12) we can deduce Hamilton’s equations of motion.

\textbf{Theorem 2} (Hamilton’s equations of motion). \textit{With the Hamiltonian defined as the Legendre transform of the Lagrangian, Lagrange’s equations of motion imply}

\[
\dot{q}_i = \frac{\partial H}{\partial p_i} \tag{2.1.13}
\]

\[
\dot{p}_i = \frac{\partial H}{\partial q_i} \tag{2.1.14}
\]

for \( i = 1, ..., n \). These are Hamilton’s canonical equations, consisting of \( 2n \) first order equations of motion.

By direct calculation, we can obtain

\[
\frac{\partial H}{\partial t} = -\frac{\partial L}{\partial t}. \tag{2.1.15}
\]

It is interesting to remark that the explicit dependence of \( H \) or \( L \) on \( t \) is present or absent together. Hence, if \( H \) does not explicitly depend on \( t \) then \( H \) is a
constant of the motion,
• conserved quantity,
• integral of the motion.

As documented in [1], the absence of explicit time dependence in the Hamiltonian $H$ could serve as a more general definition of a conservative system, though in general, $H$ may not be the total energy. However, for simple mechanical systems for which the kinetic energy $T = T(q)$ is a homogeneous quadratic function in $\dot{q}$, and the potential $V = V(q)$, then the Hamiltonian $H$ will be the total energy:

$$\sum_{i=1}^{n} \dot{q}_i \frac{\partial T}{\partial \dot{q}_i} = 2T \implies H = 2T - (T - V) \implies H = T + V$$

(2.1.16)

2.1.8 Conserving properties

For a conservative system of particles subject to the above hypothesis with no external loads applied the following properties hold:

1. Total energy, $H = K + U$.
2. Linear momentum, $P = \sum_{i} p_i$.
3. Angular momentum, $L = \sum_{i} q_i \times p_i$.
4. Symplecticity.

Setting $y^i = (p^i, q^i)$, the transformation $\phi_t : y_0 \rightarrow y(t)$ is symplectic if

$$\frac{\partial \phi_t}{\partial y_0} \begin{bmatrix} 0 & \text{Id} \\ -\text{Id} & 0 \end{bmatrix} \frac{\partial \phi_t}{\partial y_0} = \begin{bmatrix} 0 & \text{Id} \\ -\text{Id} & 0 \end{bmatrix}$$

2.2 Time integration algorithms

Our aim is to integrate Equation 2.3.3, which can be written as

$$\dot{y} = f(y),$$

where

$$y = \begin{pmatrix} p \\ q \end{pmatrix} \text{ and } f(y) = \begin{pmatrix} -\nabla_q U(q) \\ M^{-1}p \end{pmatrix}.$$
2.2.1 Numerical schemes and conserving properties

Let us first introduce our notation and define the total linear and angular momentum.

**Definition 6.** We define $\Delta(\bullet)$ as

$$\Delta(\bullet) = (\bullet)_{n+1} - (\bullet)_n$$

**Definition 7.** We define $(\bullet)_{n+\frac{1}{2}}$ as

$$(\bullet)_{n+\frac{1}{2}} = \frac{(\bullet)_{n+1} + (\bullet)_n}{2}$$

The total linear momentum is:

$$\mathbf{P} = \sum_{i=1}^{N} m^i \cdot \mathbf{v}^i, \quad (2.2.1)$$

and the total angular momentum is:

$$\mathbf{L} = \sum_{i=1}^{N} \mathbf{q}^i \times \mathbf{p}^i. \quad (2.2.2)$$

**Explicit methods**

Explicit methods calculate the state of the system at a later time from the state of the system at the current time without involving solving equations.

Explicit time-integrations algorithms are presented and briefly analyzed next.

**Definition 8** (Forward Euler). We define the Forward Euler method as an explicit numerical procedure for solving ordinary differential equations (ODEs) with a given initial value. Its numerical scheme is described by

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \cdot f(\mathbf{y}_n). \quad (2.2.3)$$

The Forward Euler method only conserves the linear momentum as it is proved next.

Using the total linear momentum described in (2.2.1) and setting $\mathbf{p}_{n+1}$ as:

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \Delta t \nabla_{\mathbf{q}_n} U(\mathbf{q}_n),$$

then, the variation of $\mathbf{P}$ is then:

$$\Delta \mathbf{P} = \sum_{i=1}^{N} m^i \cdot \mathbf{v}^i_{n+1} - \sum_{i=1}^{N} m^i \cdot \mathbf{v}^i_n$$

$$= \sum_{i=1}^{N} (\mathbf{p}^i_n - \Delta t \cdot \nabla_{\mathbf{q}_n} U(\mathbf{q}_n)) - \sum_{i=1}^{N} m^i \cdot \mathbf{v}^i_n$$

$$= \sum_{i=1}^{N} m^i \cdot \mathbf{v}^i_n + \sum_{i=1}^{N} \Delta t \cdot \nabla_{\mathbf{q}_n} U(\mathbf{q}_n) - \sum_{i=1}^{N} m^i \cdot \mathbf{v}^i_n$$

$$= \sum_{i=1}^{N} \Delta t \cdot \nabla_{\mathbf{q}_n} U(\mathbf{q}_n) = 0.$$  

Therefore, the linear momentum is a constant of the movement.
Definition 9 (Störmer-Verlet method). We define the Störmer-Verlet (leapfrog) method as an explicit numerical procedure for solving ordinary differential equations (ODEs) with a given initial value. Setting \( g(q) = -M^{-1} \nabla U(q) \) and \( q_{n+1} - 2q_n + q_{n-1} = \Delta t^2 g(q_n) \) we obtain the following numerical scheme

\[
\begin{align*}
 v_{n+\frac{1}{2}} &= v_n + \frac{\Delta t}{2} g(q_n), \\
 q_{n+1} &= q_n + \Delta t v_{n+\frac{1}{2}}, \\
 v_{n+1} &= v_{n+\frac{1}{2}} + \frac{\Delta t}{2} g(q_{n+1}).
\end{align*}
\] (2.2.4)

Implicit methods

Implicit methods calculate the state of the system at a later time from the state of the system at the current time usually involving the solution of a non-linear system of equations.

Implicit time-integration methods and their conserving properties are exposed next while the analysis of their behavior will be treated subsequently.

Definition 10 (Backward Euler). We define the Backward Euler method as an implicit numerical procedure for solving ordinary differential equations with a given initial value. Its numerical scheme is described by

\[
y_{n+1} = y_n + \Delta t \cdot f(y_{n+1}).
\] (2.2.5)

The only conserved property of the method is the linear momentum. It is proved in the same way as the Forward Euler.

Definition 11 (Trapezoidal rule). We define the Trapezoidal rule as an implicit numerical procedure for solving ordinary differential equations with a given initial value. Its numerical scheme is described by

\[
y_{n+1} = y_n + \frac{\Delta t}{2} \left( f(y_{n+1}) + f(y_n) \right).
\] (2.2.6)

The Trapezoidal rule algorithm only conserves the linear momentum. It is proved in the same way as above.

Definition 12 (Midpoint rule). We define the Midpoint rule as an implicit numerical procedure for solving ordinary differential equations with a given initial value. The numerical scheme is described by

\[
y_{n+1} = y_n + \Delta t \cdot f\left(\frac{y_{n+1} + y_n}{2}\right).
\] (2.2.7)

The algorithm ensures the conservation of the linear and angular momentum and the symplectycity. The conservation of the linear momentum is proved in the same way as the Forward Euler algorithm. As regards to the conservation of the total angular momentum of the system \( L \),

\[
L = \sum_{i=1}^{N} q_i \times p_i.
\]
its variation is,

\[ \Delta L = \sum_{i=1}^{N} \left( q_{n+1}^i \times p_{n+1}^i - q_n^i \times p_n^i \right) \]

\[ = \sum_{i=1}^{N} \Delta q^i \times p_{n+\frac{1}{2}}^i \]

\[ = \sum_{i=1}^{N} \Delta t M^{-1} p_{n+\frac{1}{2}}^i \times p_{n+\frac{1}{2}}^i = 0. \]

Therefore, the angular momentum is a constant of the movement.

**Definition 13** (Symplectic partitioned Euler). *We define the Symplectic Euler method as an implicit numerical procedure for solving ordinary differential equations with a given initial value. The numerical scheme is described by*

\[
\begin{align*}
    p_{n+1} & = p_n - \Delta t \nabla q U(q_n), \\
    q_{n+1} & = q_n + \Delta t M^{-1} p_{n+1}.
\end{align*}
\]

The symplectic Euler algorithm ensures the conservation of both momentums and the symplecticity. It is proved in the same way as in the Midpoint rule algorithm.

**Definition 14** (Energy-momentum integration). *We define the Energy-momentum method as an implicit numerical procedure for solving ordinary differential equations with a given initial value. The numerical scheme is described by*

\[
\begin{align*}
    p_{n+1}^i & = p_n^i - \Delta t \sum_{j=1}^{M_i} \frac{\Delta V(l^{ij})}{\Delta l^{ij}} \frac{q_{n+\frac{1}{2}}^j - q_{n+\frac{1}{2}}^j}{l_{n+\frac{1}{2}}^{ij}} \\
    q_{n+1}^i & = q_n^i + \Delta t M^{-1} p_{n+\frac{1}{2}}^i, \quad i = 1, \ldots, N.
\end{align*}
\]

*Note that \( \nabla q \) has been approximated by a numerical gradient \( \bar{\nabla} q \).*

The algorithm ensures the conservation of the linear and angular momentum and the energy. The conservation of the linear and angular momentum is proved in the same way as in the Midpoint Euler algorithm. As regards to the conservation of the system’s total energy let us set it as

\[ H = \frac{1}{2} \sum_{i=1}^{N} p \cdot M^{-1} p + \sum_{\forall M} V(l^{ij}) \]

then,

\[ \Delta K = \Delta p \cdot M^{-1} p_{n+\frac{1}{2}} \]

\[ = \frac{\Delta p}{\Delta t} \cdot \Delta q \]

\[ = -\frac{\Delta V}{\Delta t} l_{n+\frac{1}{2}} \cdot q = -\Delta V. \]

Thence, the total energy \( E \) is a constant of the movement.
2.3 Reference mechanical system

Let us suggest a reference mechanical system for which we will test the algorithms. This will be a system with $N$ particles with mass $m_i$ and $M$ elastic springs.

![Figure 2.1: Reference mechanical system sketch.](image)

The system has $3n$ degrees of freedom corresponding to the coordinates of each particle $q^i$, for $i = 1, ..., n$ with $q_0$ and $v_0$ as boundary conditions.

The elastic energy is a function of the length of each spring connecting masses $i$ and $j$. $V = V(l^{ij})$ and $l^{ij} = ||q^i - q^j||$.

If we define

$$
q = \begin{pmatrix} q^1 \\ \vdots \\ q^n \end{pmatrix}, \quad p = \begin{pmatrix} p^1 \\ \vdots \\ p^n \end{pmatrix} \quad \text{and} \quad M = \begin{bmatrix} m_1[I_3] \\ \vdots \\ m_n[I_3] \end{bmatrix},
$$

we then have,

$$
U^{ij}(q) = V(l^{ij}) = \frac{1}{2}k(l^{ij} - l_0^{ij})^2
$$

$$
H(q, p) = \sum_{i=1}^{N} \frac{1}{2m_i}p^i \cdot p^i + \sum_{\forall M} V(l^{ij}) = \frac{1}{2} \sum_{i=1}^{N} p \cdot M^{-1}p + \sum_{\forall M} V(l^{ij})
$$

Therefore, operating with Equation 2.3.2, Hamilton’s equations can be written as:

$$
\begin{cases}
\dot{p} = -\nabla_q U(q) \\
\dot{q} = M^{-1}p
\end{cases}
$$

(2.3.3)

where

$$
\nabla_q U = \sum_{j=1}^{M} \frac{\partial V(l^{ij})}{l^{ij}} \cdot (q^i - q^j)
$$

and $M_i$ is the number of springs connected to particle $i$.

In addition, since $M$ is constant, Equation 2.3.3 is equivalent to the following equilibrium equations

$$
g \equiv M\ddot{q} + \nabla_q U(q) = 0, \quad g \in \mathbb{R}^{3N}.
$$

(2.3.4)
The equilibrium equations for the different time-integration algorithms are summarized in Table 2.1.

<table>
<thead>
<tr>
<th>Equilibrium equations</th>
<th>Time-stepping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward Euler</td>
<td>$M\Delta v_t + \nabla U(q_n) = 0$</td>
</tr>
<tr>
<td>Backward Euler</td>
<td>$M\Delta v_t + \nabla U(q_{n+1}) = 0$</td>
</tr>
<tr>
<td>Symplectic Euler</td>
<td>$M\Delta v_t + \nabla U(q_n) = 0$</td>
</tr>
<tr>
<td>Trapezoidal rule</td>
<td>$M\Delta v_t + \frac{1}{2} (\nabla U(q_{n+1}) + \nabla U(q_n)) = 0$</td>
</tr>
<tr>
<td>Midpoint Euler</td>
<td>$M\Delta v_t + \nabla U(q_{n+\frac{1}{2}}) = 0$</td>
</tr>
<tr>
<td>Energy-momentum</td>
<td>$M\Delta v_t + \nabla U(q_n, q_{n+\frac{1}{2}}) = 0$</td>
</tr>
</tbody>
</table>

Table 2.1: Time-integration algorithms for non-contact particle mechanics.

In general they will be written as:

$$g^d + g^e = 0 \quad (2.3.5)$$

where $g^d$ and $g^e$ respectively represent the dynamic and elastic contribution of the residual vector.

In order to solve (2.3.5) we will use the standard Newton-Raphson procedure. Jacobian matrix for the different implicit algorithms are summarized in Table 2.2. Its calculus is justified in Annex 1.

<table>
<thead>
<tr>
<th>Jacobian matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backward Euler</td>
</tr>
<tr>
<td>Symplectic Euler</td>
</tr>
<tr>
<td>Trapezoidal rule</td>
</tr>
<tr>
<td>Midpoint Euler</td>
</tr>
<tr>
<td>Energy-momentum</td>
</tr>
</tbody>
</table>

Table 2.2: Jacobian matrix for non-contact time-integration algorithms.

Where,

$$q^* = \begin{bmatrix} q^1 - q^2 \\ q^2 - q^1 \end{bmatrix},$$

and,

$$I^* = \begin{bmatrix} I_3 & -I_3 \\ -I_3 & I_3 \end{bmatrix}.$$
<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>L</th>
<th>E</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward Euler</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Backward Euler</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Symplectic Euler</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Trapezoidal rule</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Midpoint Euler</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Energy-momentum</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
</tbody>
</table>

Table 2.3: Time integration conserving properties.
3 Contact time integration algorithms for conservative particle systems with fixed boundaries

3.1 Continuum equilibrium equations with contact conditions

3.1.1 Contact conditions

**Definition 15** (Signed distance). The signed distance (or gap) of a particle \( i \) to a surface \( S \) is:

\[
d_i = -n \cdot (q_i - q_{ci})
\]  

(3.1.1)

Note that admissible positions are then defined by \( d_i \leq 0 \), and that \( \nabla q d_i = -n \).

Position \( q_{ci} \) corresponds to the closest point to particle \( q_i \) of the contact surface \( S \) with external normal \( n \).

It is worth to note that in order to keep generality, we have defined the signed distance for any kind of surface but, for a plane surface it is not needed to calculate \( q_{ci} \); it can be replaced for any arbitrary point \( p \) located located onto the surface \( S \).

![Figure 3.1: Geometrical variables for plane surfaces.](image)

Equilibrium equations are the same as the no contact problem but now subjected to the following Karush-Kuhn-Tucker conditions [7].

**Definition 16** (Karush-Kuhn-Tucker conditions). If \( d^t \) is the vector of distances for all the particles to a surface \( S \) and \( \lambda^t \) is the vector of normal forces that \( S \) applies on them (\( \lambda > 0 \Rightarrow \) compression), we then define the Karush-Kuhn-Tucker conditions as:

\[
\lambda \geq 0
\]  

(3.1.2)

\[
d \leq 0
\]  

(3.1.3)

\[
\lambda \cdot d = 0
\]  

(3.1.4)

Equation (3.1.4) is the so-called complementary condition.

Therefore, the equilibrium equation (2.3.5) must be modified in order to incorporate the contact conditions (3.1.2) to (3.1.4).
In static problems, it is possible to add a contact potential $V^c$ to the elastic potential $U$ to obtain the modified equilibrium conditions. In this case, an additional residual contribution $g^c$ is added to (2.3.5). It results in the following extended equilibrium equations:

$$g^d + g^e + g^c = 0.$$ (3.1.5)

Depending on the method employed for imposing the constraints, the form of $V^c$ and $g^c$ varies.

### 3.1.2 Penalty formulation

#### Definitions of potential and contact residual part

Let us first define a couple of functions.

**Definition 17** (Mcauley bracket). The Mcauley bracket ($\langle \cdot \rangle$) is defined as:

$$\langle x \rangle = \begin{cases} 
  x & \text{when } x \leq 0 \\
  0 & \text{when } x < 0 
\end{cases}$$ (3.1.6)

**Definition 18** (Heaviside function). The Heaviside function $H$ is defined as:

$$H(x) = \begin{cases} 
  1 & \text{if } x \geq 0 \\
  0 & \text{if } x < 0 
\end{cases}$$ (3.1.7)

Note that $\nabla_\eta \langle x \rangle = H(x)$.

The Penalty formulation[7] consists in adding a quadratic potential $V^c_\epsilon$ proportional to a penalty parameter $\epsilon$ and to the signed distance constraint violation $d_i$ as:

$$V^c_\epsilon = \frac{1}{2} \epsilon (d_i)^2$$ (3.1.8)
Hence, the residual contribution due to the contact potential $V^c$ is given by:

$$g^c = \nabla_q V^c = -\epsilon \langle d_i \rangle n$$  \hspace{1cm} (3.1.9)

As we can see in (3.1.9), the fact that $g^c \propto \epsilon \langle d_i \rangle$ implies that increasing $\epsilon$, the accuracy of the constraints increases. Unavoidably, as increases the accuracy of the constraints the conditioning problems increases.

Thence, for a mechanical system with $K$ surfaces, the residual contribution of the $i$th particle results in the following expression:

$$g^c_{\epsilon i} = \sum_{j=1}^{K} -\epsilon_k \langle d_{i,k} \rangle n_k$$  \hspace{1cm} (3.1.10)

It is interesting to remark that no more than one surface constraint should be violated at the same time. Unfortunately, a not appropriated time discretization or a complex geometry entails these kinds of situations. The effects of this phenomena will be treated subsequently.

**Numerical implementation**

The equilibrium equation (3.1.5) adapted to the Penalty method only depends on displacements $q_i$. Then,

$$g^d + g^e + g^c = 0.$$  

can be iteratively solved for implicit methods.

### 3.1.3 Lagrange multipliers

**Definitions of potential and contact residual part**

The Lagrange multipliers $\lambda_i$ are the reactions forces due to the constraints, which give rise, as documented in [7], to the following form of the constraint potential:

$$V^c_\lambda = \langle \lambda \rangle d_i.$$  \hspace{1cm} (3.1.11)

Hence, the residual vector contribution due to the constraint is in this case given by,

$$g^c_\lambda = -\langle \lambda_i \rangle n$$  \hspace{1cm} (3.1.12)

Note that reaction forces $\lambda_i$ are here also considered as unknowns of the problem and, therefore, the global system of equations is expanded with the equations $\nabla_\lambda V^c_\lambda = 0$, that is,

$$H(\lambda_i) d_i = 0.$$  \hspace{1cm} (3.1.13)

Thence, for a mechanical system with $K$ surfaces, the residual contribution of the $i$th particle results in the following expression:

$$g^c_i = \sum_{j=1}^{K} -\langle \lambda_{i,k} \rangle n_k$$  \hspace{1cm} (3.1.14)
Numerical implementation

The equilibrium equation (3.1.5) adapted to the Lagrange multipliers method depends on displacements $q_i$, and reaction forces $\lambda_i$. Then,

$$g^d + g^e + g^c_{i\lambda} = 0.$$  \hfill (3.1.15)

has to be iteratively solved for the couples $(q_i, \lambda_i)$. The condition $d_i \leq 0$ can be iteratively imposed by updating accordingly $q_i$, and modifying the system according to the set of active constraints $\lambda_i \geq 0$.

Alternatively, and inspired in the Uzawa algorithm, the following two stage algorithm is proposed. In order to uncouple the unknowns, the values of $\lambda_i$ are fixed throughout the Newton-Raphson iterations of the displacement equations in (3.2.3). After convergence with some fixed values $\lambda_i^k$, the new values $\lambda_i^{k+1}$ are obtained according to the following update process:

1. Predict $\lambda$ increments: $\lambda_i^{k+1} = \lambda_i^k + \bar{g}_i \cdot n$
2. Update $\lambda$: $\lambda_i^{k+1} = \langle \lambda_i^{k+1} \rangle$

Where the residual vector $\bar{g}_i$ is the total residual associated to particle $i$ with the nodes projected onto surface $S (q_i^*)$, and with the velocities updated according to this displacement changes:

$$\bar{g}_i = g_i(q_i^*).$$  \hfill (3.1.16)

The vector measures the additional normal reactions that should be applied due to the constraint enforcement, and therefore, is a valid prediction of the additional reactions $\lambda_i$ to be applied.

![Figure 3.3: Projection of $q_i$ onto surface $S$.](image)

Constraint casuistry

As we discussed in Section 3.1.2 no more than one surface constraint should be violated at the same time step. Unfortunately, large time-steps favor this non-realistic situations. It is obvious that the problem may be solved using variable time stepping algorithms but complex geometries, as those involving
pronounced concavities or convexities, may lead to too short time increments, and short time-steps may entail conditioning problems.

We suggest for theses situations to allow more than one surface constraint violation and simplify the phenomena using as reactions forces as violations exists. It is important to note that no more than two violations should be allowed in a time-step for a proper simulation behavior and no more than three violations are algorithmically possible as it is discussed next.

Numerical simplifications will be based on the prediction of reaction forces $\lambda_i$ associated with the total residual $\mathbf{g}_i$ of particle $i$. Then, particles projection $\mathbf{q}^*_i$ should be modified depending on the number of surface violations:

1. **One surface violation.**
   No simplifications are needed for this case. The position of the projected particles may be calculated as:
   \[ \mathbf{q}^*_i = \mathbf{q}_i + d_{i,k} \mathbf{n}_k. \]

2. **Two surface violations.**
   Reaction forces $\lambda_i$ updates may be intuitively understood as a restriction on the movement into the surface constraint. Therefore, if two surfaces are violated, particle $i$ should be projected onto the intersection of both surfaces. Then, for an arbitrary point $\mathbf{q}_\beta \in S_1 \cap S_2$:
   \[ \mathbf{q}^*_i = \mathbf{q}_\beta + ((\mathbf{q}_i - \mathbf{q}_\beta) \cdot \mathbf{\tau}_\beta) \mathbf{\tau}_\beta \]

3. **Three or more surface violations.**
   In the case of three violations and proceeding as above, $\mathbf{q}_i$ should be projected onto the intersection of those three surfaces and it’s movement should be restricted in every direction. More than three violations are non-compatible with this formulation.

   It is obvious that the time-step has to be decreased for both cases.

**Lagrange multipliers prediction for multiple constraints**

Let us consider a particle with two boundaries $a$ and $b$. We will denote the respective Lagrange multipliers by $\lambda^a$ and $\lambda^b$. The estimation of both Lagrange multipliers and the computation of the contact residual $\mathbf{g}^c$ may be resorting on the concept of covariant and contravariant (or reciprocal) basis that will be explained next.

The normals $\mathbf{n}_a$ and $\mathbf{n}_b$ form (in general) a non-orthonormal covariant basis. Associated to the basis $\mathbf{n}^i$, we can construct a contravariant basis $\mathbf{n}^i$ such that

\[ \mathbf{n}^i \cdot \mathbf{n}_j = \delta^i_j \quad (3.1.17) \]

where $\delta^i_j$ is the Kronecker delta, i.e. $\delta^i_j = 1$ when $i = j$ and 0 otherwise. Actually, each vector of the basis $\mathbf{n}^i$ can be found by solving the system of equations in (3.1.17), for $j = 1, 2, 3$. Note that if $\mathbf{n}_i$ are unit vectors but not orthogonal, the contravariant vector $\mathbf{n}^i$ are not unique vectors.

Any vector $\mathbf{v}$ can be expressed in both bases as $\mathbf{v} = v^i \mathbf{n}_i = v_i \mathbf{n}^i$ and therefore, we can compute the contravariant and covariant components as $v^i = \mathbf{v} \cdot \mathbf{n}^i$ and $v_i = \mathbf{v} \cdot \mathbf{n}_i$. Figure 3.4 depicts an example of such bases.
In this context, the Lagrange multipliers \( \lambda^i \) can be interpreted as the contravariant components of the residual vector \( g^c \) in the covariant basis \( n_i \) as:

\[
g^c = H(d^a)(\lambda^i)n_a + H(d^b)(\lambda^i)n_b = \lambda^a n_a + \lambda^b n_b
\]

where we have assumed that both constraints are active, i.e. \( \dot{d}^i > 0 \) and \( \lambda^i > 0 \). The Lagrange multipliers \( \lambda^i \) must be then estimated from the contravariant components of \( g \) as,

\[
\lambda^{i,k+1} = \lambda^{i,k} + \mathbf{g} \cdot \mathbf{n}^i.
\]

It can be verified that when \( \mathbf{g} = 0 \), the projection of the equilibrium equations onto \( \mathbf{n}^i \) and \( \mathbf{n}^k \) yields:

\[
\begin{align*}
\mathbf{n}^a \cdot (g^d + g^e) + \lambda^a &= 0 \\
\mathbf{n}^b \cdot (g^d + g^e) + \lambda^b &= 0
\end{align*}
\]

that is, when the constraint \( i \) is inactive (\( \lambda^i = 0 \)), the resultant of dynamic and elastic forces along direction \( \mathbf{n}^i \) is zero.

### 3.2 Energy-momentum algorithm

Equilibrium equations in (3.1.5) can be derived from the following weak form (or virtual work principle) of the contact problem:

\[
\delta \mathbf{q}^t \left( g^d + g^e + g^c_{n+\frac{1}{2}} \right) = 0 \quad \forall \delta \mathbf{q},
\]

where \( \delta \mathbf{q} \) is the vector of admissible virtual displacements.

Thence, the satisfaction of the previous weak form (3.2.1) is equivalent to solve the following set of equations,

\[
\begin{align*}
g^d + g^e + g^c_{n+\frac{1}{2}} &= 0.
\end{align*}
\]

As documented in [8], energy-momentum algorithms are such that,

\[
\mathbf{v}_{n+\frac{1}{2}}^t \left( g^d + g^e + g^c_{n+\frac{1}{2}} \right) = \frac{\Delta E}{\Delta t}
\]
where $E$ is the total energy. Therefore, since the velocity $v$ is an admissible virtual displacement, the enforcement of the equilibrium equations in (3.2.2) implies the conservation of the total energy, that is $\Delta E = 0$. Furthermore, since the sole energy increments should be those of the kinetic energy, $\Delta K = v_{n+\frac{1}{2}}^t g^c_{n+\frac{1}{2}}$, and the elastic energy, $\Delta V = v_{n+\frac{1}{2}}^t g^e_{n+\frac{1}{2}}$, an energy preserving contact residual contribution should satisfy:

$$v_{n+\frac{1}{2}}^t g^e_{n+\frac{1}{2}} = 0. \tag{3.2.3}$$

Such a residual can be obtained by using an appropriate update procedure and setting:

$$g^c_{n+\frac{1}{2}} = -n_{n+\frac{1}{2}} \lambda. \tag{3.2.4}$$

Note that although in most of the examples the contact surface remains motionless and therefore, the normal vector $n$ is constant, we have kept the notation $n_{n+\frac{1}{2}}$ for generality.

Before going into the different algorithms let us define for their subsequent implementation the algorithmic distance rate.

**Definition 19 (Algorithmic distance rate).** The algorithmic distance rate $\dot{d}_n$ and $\dot{d}_{n+\frac{1}{2}}$ are defined as:

$$\dot{d}_n = -v_n^t n_n$$

$$\dot{d}_{n+\frac{1}{2}} = -v_{n+\frac{1}{2}}^t n_{n+\frac{1}{2}} = -\frac{1}{\Delta t} \Delta \mathbf{q}^t n_{n+\frac{1}{2}}, \tag{3.2.5}$$

where the subscript $i$ has been removed for clarity, and the velocity update $v_{n+\frac{1}{2}} = \frac{\Delta \mathbf{q}}{\Delta t}$ of the Energy-momentum algorithm has been used in (3.2.6).

Finally, the energy conservation condition in (3.2.3) can be now rewritten as:

$$-\lambda \dot{d}_{n+\frac{1}{2}}. \tag{3.2.7}$$

### 3.2.1 Dissipative penalty algorithm

**Definition of contact residual part**

The dissipative penalty algorithm, based in [2], consists in replacing the residual contribution in (3.2.4) by the following expression:

$$g^c_{n+\frac{1}{2}} = -H(d_{n+\frac{1}{2}}) \epsilon (\dot{d}_{n+\frac{1}{2}}) n_{n+\frac{1}{2}}. \tag{3.2.8}$$

Therefore, the energy variation $\frac{\Delta E}{\Delta t}$ results in:

$$\frac{\Delta E}{\Delta t} = -v_{n+\frac{1}{2}}^t g^c_{n+\frac{1}{2}}$$

$$= H(d_{n+\frac{1}{2}}) \epsilon (\dot{d}_{n+\frac{1}{2}}) v_{n+\frac{1}{2}}^t n_{n+\frac{1}{2}}$$

$$= -H(d_n) \epsilon (d_{n+\frac{1}{2}}) \dot{d}_{n+\frac{1}{2}}$$

$$= -H(d_n) \epsilon (\dot{d}_{n+\frac{1}{2}})^2 \leq 0,$$
wich demonstrates that the algorithm is dissipative and energetically stable.

For a mechanical system with \( K \) surfaces, the residual contribution of the \( i \)th particle results in the following expression:

\[
g_{n+\frac{1}{2}}^c = -\sum_{k=1}^{K} H(d_{n+\frac{1}{2},k})\epsilon_k(d_{n+\frac{1}{2},k})n_{n+\frac{1}{2},k}.
\]  

(3.2.9)

Numerical implementation

The equilibrium equation (3.2.2) adapted to the dissipative penalty method only depends on displacements \( q_i \). Then,

\[
g^d + g^e + g_{n+\frac{1}{2}}^c = 0.
\]

can be iteratively solved by the standard Newton-Raphson procedure.

3.2.2 Conservative penalty

Definitions of potential and contact residual part

Using the quadratic potential defined for the Penalty method in (3.1.8) and the modified gradient \( \nabla_q \) used in Definition 14 we then obtain the following residual contribution:

\[
g_{n+\frac{1}{2}}^c = \nabla_q U^c = -\frac{\Delta U^c}{\Delta d_i} n_{n+\frac{1}{2}}^c:
\]

(3.2.10)

Therefore, the energy variation \( \frac{\Delta E}{\Delta t} \) results in:

\[
\frac{\Delta E}{\Delta t} = -v_{n+\frac{1}{2}}^t g_{n+\frac{1}{2}}^c
\]

\[
= \frac{\Delta U^c}{\Delta d_i} v_{n+\frac{1}{2}}^t n_{n+\frac{1}{2}}^c = -\frac{\Delta U^c}{\Delta d_i} \frac{1}{\Delta t} \Delta q^i n_{n+\frac{1}{2}}^c
\]

\[
= \frac{\Delta U^c}{\Delta t} = -\frac{\Delta E}{\Delta t}
\]

3.2.3 Lagrange multipliers

Definition of contact residual part

The Lagrange multipliers algorithm for the Energy-momentum method consists of replacing the residual contribution in (3.2.4) by the following expression:

\[
g_{n+\frac{1}{2}}^c = -H(\dot{d}_n)\lambda.
\]

(3.2.11)

Therefore, for a mechanical system with \( K \) surfaces, the residual contribution of the \( i \)th particle is:

\[
g_{n+\frac{1}{2}}^c = -\sum_{k=1}^{K} H(\dot{d}_{n,k})\lambda_k.
\]

(3.2.12)

The condition in (3.2.7) can be satisfied by using the following update procedure:
1. If $d_n < 0$ or $\dot{d}_n < 0$
   (a) $\lambda = 0$
   (b) $\dot{d}_{n+\frac{1}{2}}$ unconstrained

2. If $d_n \geq 0$ and $\dot{d}_n = 0$
   (a) $\lambda \geq 0$
   (b) $\dot{d}_n u = 0$

3. If $d_n \geq 0$ and $\dot{d}_n > 0$
   (a) $\lambda \geq 0$
   (b) $\dot{d}_{n+\frac{1}{2}} = 0$

It is important to note that the subscript of the Lagrange multiplier has not been specified due to its algorithmic value. It can be understood as a mid-time value $\lambda_{n+\frac{1}{2}}$, but it is not the time-average value of any quantity.

We also note that this algorithm ensures the conservation of the energy, but it does not prevent situations where the constraints are violated, $\dot{d}_{n+\frac{1}{2}} \geq 0$.

**Numerical implementation**

The equilibrium equation (3.1.5) adapted to the Lagrange multipliers method depends on displacements $\mathbf{q}_i$ and reaction forces $\lambda_i$. Then,

$$ g^d + g^e + g^c \lambda = 0 $$

has to be iteratively solved for the couples $(\mathbf{q}_i, \lambda_i)$. The condition in (3.2.7) can be iteratively imposed by updating accordingly $\mathbf{q}_i$, and modifying the system according to the set of active constraints $\lambda_i \geq 0$.

Alternatively, as it is described in Section 3.1.3 a two-stage algorithm can be applied. In this case, for the update of $\lambda$ we suggest a similar algorithm to solve Equation 3.1.5, and in cases 2a and 3a above, where $\lambda$ may be non-zero, update it according to:

1. Predict $\lambda$ increments: $\lambda_i^{k+1} = \lambda_i^k + \bar{g}_i \cdot n_{n+\frac{1}{2}}$

2. Update $\lambda$: $\lambda_i^{k+1} = (\lambda_i^{k+1})$

Where the residual vector $\bar{g}_i$ is the total residual associated to particle $i$ with the nodes projected onto the shifted surface $d_{n+\frac{1}{2}} = 0$. It is done to compute the normal component of $\mathbf{q}_{n+1}$ according to $\dot{d}_{n+\frac{1}{2}} = 0$, which is equivalent to:

$$ q'_{n+1} n_{n+\frac{1}{2}} = q'_{n} n_{n} \quad (3.2.13) $$
Figure 3.5: Projection of $q_i$ onto the shifted surface $S^\ast$.

Constraint casuistry

Casuistry for the Lagrange multipliers energy-momentum algorithms has the same scheme as exposed in Section 3.1.3 but replacing the projection surface for the shifted surface $S^\ast$ that ensures the conservation of the properties.

Numerical simplifications are also based on the prediction of reaction forces $\lambda_i$ associated with the total residual $\bar{g}_i$ of particle $i$. Then, particle projection $q^\ast_i$ should be modified depending on the number of surface violations:

1. **One surface violation.**
   No simplifications are needed for this case. Position of the projection of the particles may be calculated as:
   \[
   q^\ast_i = q_i^{n+1} + \left( d_{i,k}^{n+1} - d_{i,k}^n \right) n_k^{n+1}.
   \]
   The movement is restricted onto the shifted surface.

2. **Two surface violations.**
   As described for the non-conservative method, the movement of $q_i$ is restricted onto the straight line $\beta \in S_1 \cap S_2$. Then,
   \[
   q^\ast_i = q_i^n + \left( (q_i^{n+1} - q_i^n) \cdot \tau_\beta \right) \tau_\beta
   \]

3. **Three or more surface violations.**
   In the case of three violations and proceeding as above, $q_i$ should be projected onto the intersection of those three shifted surfaces and its movement should be restricted in every direction. More than three violations are non-compatible with this formulation.

   As explained in Section 3.1.3, the time-step has to be decreased for both cases.

Lagrange multipliers prediction for multiple constraints

The numerical procedure to predict the Lagrange multipliers is identical as the one exposed in Section 3.1.3.
3.3 Summary

Table 3.3 summarizes the main algorithms used for impact mechanics. They are divided into two groups. Those focused on conserving the energetic properties and the rest.

<table>
<thead>
<tr>
<th>Algorithm Type</th>
<th>Method</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy-momentum</td>
<td>Dissipative penalty</td>
<td>(-H(d_{n+\frac{1}{2}})\epsilon\langle d_{n+\frac{1}{2}} \rangle n_{n+\frac{1}{2}})</td>
</tr>
<tr>
<td>algorithms</td>
<td>Conservative penalty</td>
<td>(-\frac{\Delta U^r}{\Delta t} n_{n+\frac{1}{2}})</td>
</tr>
<tr>
<td></td>
<td>Lagrange multipliers</td>
<td>(-H(d_n)\lambda)</td>
</tr>
<tr>
<td>Other</td>
<td>Penalty method</td>
<td>(-\epsilon\langle d_i \rangle n)</td>
</tr>
<tr>
<td>algorithms</td>
<td>Lagrange multipliers</td>
<td>(-\langle \lambda_i \rangle n)</td>
</tr>
</tbody>
</table>

Table 3.1: Residual contribution depending on the impact method.
4 Results

Non-linear differential equations can exhibit very complicated behavior over extended time intervals and even the fundamental questions of existence, uniqueness, and extendability of solutions are not solved issues. Then, the geometric numerical integration is an interesting way to tackle the issue owing to the fact that the invariants preservation of the system not only produces an improved qualitative behavior, but allows for a more accurate long-time integration. For these reasons we have designed several toy problems to analyze the goodness of its integration by the algorithms presented in Section 3.

4.1 Toy problem 1. Particles between two parallel constraints.

In order to test the algorithms we have designed the following problem, without immediate scientific interest, to illustrate in an intuitive way the behavior of the algorithms.

The problem consists of a two mass system united with an elastic spring and subjected to two surface constraints as it is presented in Figure 4.1.

![Figure 4.1: Toy problem 1.](image)

As exposed in Section 2.3, the total energy, the total linear and angular momentum and the symplecticity are preserved when there is no impact. However, after an impact only the argument of the linear and angular momentum is preserved. Therefore, we will focus on those properties to analyze the behavior of the different algorithms.

In addition, the problem is constructed in such a way that the vertical displacement of the particles is \( d_p = v_y \Delta t \). Furthermore, we will use it as another flag to check the goodness of our numerical experiments.
The main settings for the experiments are:

\[ m_1 = m_2 = 1 \]
\[ q_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \]
\[ q_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \]
\[ v = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \]

and two surface constraints \( S_1 \in (-2, y) \) and \( S_2 \in (2, y) \) of normals \( n_1 = (1, 0)^t \) and \( n_2 = (-1, 0)^t \) respectively.

### 4.2 Non-conservative methods

#### 4.2.1 Penalty method

The penalty method establishes the surface constraint as a spring layer. Therefore, the method supposes an elasticity for the surface. However, we are simulating a perfect elastic collision and the hypothesis of the method is only satisfied when the spring elasticity tends to infinity. The effects of this approximation will be analyzed subsequently.

**Particles trajectory**

The trajectories of \( m_1 \) and \( m_2 \), integrated with the Midpoint Euler method, for different values of the penalty \( \epsilon \) are presented in Figure 4.2. As we can see, the solution strongly depends on the elasticity constraint value.

![Figure 4.2: Particle trajectories for different values of the penalty (\( \Delta t = 0.05, t = 4 \).)](image)
As we discussed above, in order to approximate us to the hypothesis of a perfect elastic collision, the penalty value should be high. Indeed, for low values of $\epsilon$ the surface is largely penetrated. Moreover, the Figure 4.2 highlights the fact that as increases $\epsilon$ the violation of the surface decreases. Note that for $\epsilon = 10^4$ the constraint violation is almost inappreciable.

Unfortunately, the non-continuity introduced by the Karush-Kuhn-Tucker conditions is difficult to be reproduced numerically. In addition, as $\epsilon$ increases, the conditioning of the system’s matrix is deteriorated and the non-continuity becomes more accentuated. As we can see for $\epsilon = 10^4$, the constraint condition may be well satisfied but the trajectory of both particles seems to take the wrong path. The collision should produce a more accentuated movement of both particles to the left.

In order to illustrate the phenomena, we have represented the trajectory of $m_2$ for different time-steps and for a high value of $\epsilon$. Figure 4.3 shows that for such high values, only a good qualitative behavior of particles is obtained by a short time-step integration. This is due, as we discussed above, to the fact that such a non-linearity is difficult to reproduce numerically. Hence, the time-step should be adapted in the contact zone depending on the penalty parameter and the particles speed.

![Graph](image)

**Figure 4.3:** Particle two trajectory for different time-steps ($\epsilon = 10^4$).

Until now, we have analyzed the system’s behavior depending on the main variables of the penalty method using the mid-point Euler discretization. Although every integration method has its particularities, the obtained results using the penalty method are qualitatively similar. Moreover, as decreases the time-step, the results tend to the same solution. It could seem an obvious issue, but the different contact formulations integrated with the different discretizations will give opposite behaviors as we will see subsequently.

In Figure 4.4 there are plotted the solutions of the toy problem for different discretizations and for $\epsilon = 100$. As exposed previously, the solution for the different algorithms tend to one as decreases the time-step. Let us remark that
for a short time-step, the trapezoidal rule and the mid-point Euler algorithms present a more accurate behavior; the contact condition is clearly-defined for both algorithms. However, the other algorithms do not reproduce the sudden variation in the particles speed and, therefore, the particles remain excessively on the constraint.

Since there is any reference analytical solution the invariants preservation is our best flag to check the goodness of the solutions. We will subsequently analyze the total energy and the linear and angular momentum evolution depending on the chosen discretizations and time-step.

As regards to the frictionless hypothesis, the theoretical vertical distance covered by both particles should be \( d = v_y \Delta t = 4 \). The numerical results are those predicted and, therefore, the method properly reproduces such contact hypothesis.

Energy behavior

As exposed in Section 2.1.8, the total energy \( E \) of the system should be conserved after the collision. Hence, we will focus on it to check the goodness of
the penalty method integration.

Let us remark that the main algorithms have their own particularities regarding to the invariants conservation. Therefore, the analysis of the contact formulation will be always linked to the chosen discretization. In addition, the fact that a penalty value has to be taken extends the casuistry. In order to simplify this issue, and due to the mid-point Euler algorithm behavior representativity, we will illustrate the contact algorithm comportment through it.

Let us start analyzing the total energy evolution depending on the penalty value. As we discussed previously, the particular effects due to the chosen discretization decrease as the time-step decreases. In Figure 4.5 there is represented the total energy of the system using the mid-point Euler discretization with a short time-step.

For low values of $\epsilon$ the constraint surface is largely violated but, after the collision, the energy is well preserved. Contrary, for high values of $\epsilon$ the boundary is geometrically respected but the total energy of the system decreases as $\epsilon$ increases. This phenomenon is unavoidable and directly linked to the difficulties of reproducing numerically the discontinuity introduced by the constraint reaction. Note that as increases the penalty value, more accentuated becomes the discontinuity. It can only be improved decreasing the time-step as we will see subsequently.

Thence, it is necessary to choose an equilibrated value of the penalty that should tend to satisfy the geometrical constraints and the preservation of invariants for a particular time-step.

To illustrate the time-step influence on the total energy it is plotted, in Figure 4.6, its evolution depending on $\Delta t$. As the reader will note, the total energy recovery after the collision improves as the time-step decreases. Even that, to obtain accurate results, too short time-steps are needed and the algorithm tendency is to be dissipative. It is worth to remark that the algorithm is globally dissipative, then, the total energy may be increased from a step to the following

![Figure 4.5: Total energy of the system for different values of the penalty ($\Delta t = 0.005$, $t = 4$).](image)
one during the contact, but not after the collision.

As concerns the long-time integration, the dissipative character of the algorithm presents an additional difficulty: if the system’s energy is partially dissipated at every collision, particles will tend to remain still. Consequently, long-time integration cannot be tackled using the penalty method algorithm.

![Figure 4.6: Total energy of the system for different time-steps (\(\epsilon = 10^3\)).](image)

Until now we have exposed the qualitative behavior of the penalty algorithm through the mid-point Euler discretization but, as remarked previously, every algorithm has its particularities. To tackle the issue, there is plotted in Figure 4.7 the total energy evolution computed for \(\epsilon = 10^3\) and depending on the discretization method and on the time-step. The global tendency for all algorithms is to improve the energy recovery as decreases the time-step.

Let us first remind that the forward Euler algorithm has a natural tendency to dissipate the energy, the symplectic Euler tends to oscillate on the theoretical total energy, the trapezoidal rule tends to increase the energy of the system and the mid-point Euler has a more neutral effect on it.

As we can see in Figure 4.7 the best algorithm regarding the energy conservation is the trapezoidal rule. Even that, during the time between collisions it noticeably tends to increase the invariant. In addition, if we focus on the transition between \(\Delta t = 0.1\) to \(\Delta t = 0.05\) the algorithm already tends to draw the correct behavior for a long time-step while the others do not achieve to predict correctly the second collision (see Figure 4.4).

The mid-point Euler presents a similar but less accurate behavior than the trapezoidal rule. The main vantage of this discretization is that between impacts it conserves both momentums and the symplecticity (see Table 2.3). Consequently, its properties qualify it to be used for long-time integration purposes.

**Linear momentum behavior**

As exposed in Section 2.1.8, the total linear momentum \(P\) of the system should be conserved if no collision. In addition, the problem is constructed in such a way that after the right particle second collision, the system has to recovery the initial linear momentum. Let us remember that initially both particles move
to right. When the right particle collides into the constraint it bounces with the same speed as it came into. Then, both particles will move each one to the other one, and the spring will suffer a contraction and afterwards an elongation to finally end up in the same positions as when the second particle collided and with the same speed but sign changed. Therefore, the sum of each linear momentum norm of each particle should be exactly the same as initially.

In order to continue with the fixed scheme, we will go through the analysis of the invariant behavior using the Mid-point Euler discretization. Let us start presenting the evolution of the invariant depending on the penalty value. As we can see in Figure 4.8 none of the simulations with the different values for the penalty reach to recover the initial linear momentum. The phenomenon has to be explained in line with the energy variation. Firstly, the hypothesis of the method to put the collision on an equal footing with a spring layer breaks the symmetry of the trajectories and, therefore, the linear momentum should only tend to be recovered if the elasticity tend to infinity. Secondly, the fact to discretize the time implies, unavoidably, a loss of energy (see Figure 4.7) and, consequently, the speed of particles and the total linear momentum is reduced.

As regards the penalty value, the best results are obtained with a value of $\epsilon = 10^2$. This is due to the fact that the time discretization used is in balance with the value of the penalty. Notice that high values of $\epsilon$ do not reach an admissible
Figure 4.8: Total linear momentum of the system for different values of the penalty ($\Delta t = 0.05, t = 4$).

rate of recovery even for the energy. As remarked previously, high values of $\epsilon$ just reproduce properly the geometrical issues and the conservation of the invariants has to be tackled with very short time-steps that are computationally expensive.

Figure 4.9: Total linear momentum of the system for different time-steps ($\epsilon = 10^3$).

With the purpose of illustrating the sensitivity of the invariants to the time-step, there is plotted, in Figure 4.9, the evolution of the norm of the total linear momentum depending on the time-step. The figure evinces what we disclosed previously, this invariant is difficult to be reproduced numerically due to the hypothesis of the integration method. Notice that for $\epsilon = 10^3$ even the shortest time-step do not reach acceptable results.

Finally, we offer the Figure 4.10 where there is plotted the evolution of the
total linear momentum depending on the different algorithms and on the time-step. The aim of the figure is to show up the particularities of the different discretization. Notice that for long time-steps the momentum is not recovered properly due to the energy dissipation. For balanced ratios of $\frac{\epsilon}{\Delta t}$ (Figure 4.10 (c) and (d)) the rate of recovered linear momentum begins to be admissible but at the cost of expensive short steps. For these figures, and supposing that the contact condition is well reproduced, we can perceive an improved behavior for the trapezoidal rule and the Mid-point Euler discretization.

### 4.2.2 Lagrange multipliers

The Lagrange multipliers method is based on the calculus of the reaction forces that appears when particles collide into the constraints. However, we are simulating a perfect elastic collision and the reaction force should only appear a differential of time. That presents a difficult issue to solve, the time is discretized in $\Delta t$ but the reaction force $\lambda$ should appear only for $\Delta t \to 0$.

As exposed in Section 3.1.3, the reaction force $\lambda$ is calculated imposing the position of the particle to be onto the surface constraint after the violation. This supposes a big inaccuracy: the particle should never violate the constraint. In
order to improve the method we will introduce a new gap to redefine the free space as we will expose subsequently.

**Particles trajectory**

The trajectories of $m_2$, integrated with the Lagrange multipliers method for the different discretizations are presented in Figure 4.11. As we can see, the solution strongly depends on the discretization.

![Figure 4.11: Trajectory of $m_2$ using the Lagrange multipliers method for different time-steps and algorithms.](image)

The Figure 4.11 highlights the fact that the forward Euler and the symplectic Euler discretization do not properly reproduce the collision. The reader will note that the right particle remain onto the constraint surface while the spring is contracting. Therefore, the hypothesis of an elastic collision is not satisfied. The other discretization have a more realistic behavior even for large time-steps.

As regards the trajectory after the second collision, the symmetry of the toy problem brings us to expect that particles have the same incline trajectory as when they collided. Results do not satisfy the exposed symmetry. The loss of the energy of the system breaks the symmetry of the problem and the collisions induce oscillations. In addition, and contrary to what is expected,
the best results are obtained for a relatively large time-step ($\Delta t = 0.05$). The phenomenon will be discussed subsequently in line with the total energy of the system and with the total linear momentum.

Before going into the invariants analysis, let us treat the benefits of introducing a gap to modify the non-contact zone. The numerical treatment of the Karush-Kuhn-Tucker conditions may lead our algorithms to non-convergence due to the linearization of a non derivable point. To avoid the problem it is interesting to introduce a gap that defines a new contact zone. The particularity is that the signed distance used to calculate the reaction force $\lambda$ is referred to the initial constraint surface but the violation zone is referred to the new gap.

As we can see in Figure 4.12, the fact of introducing a new gap makes possible to introduce in the algorithms the reaction force one iteration before. Then, it allows us to properly satisfy the non-violation condition of our constraints and, therefore, our integration results to be more accurate. In addition, the new gap saves the complication of introducing a tolerance for the signed distance (this tolerance is used to save the non-convergence due to the non-derivable point).

![Figure 4.12: Benefits of introducing a new gap.](image)

Let us remark that the trajectories computed in Figure 4.11 use a gap $g^*$ according to the dimensions of the problem. In our case $g^* = -4 \cdot 10^{-3}$.

**Energy behavior**

As exposed in Section 2.1.8, the total energy $E$ of the system should be conserved after the collision. Hence, we will focus on it to check the goodness of the Lagrange multipliers integration method.

Due to the particularities of each discretization, we offer in Figure 4.13 the total energy of the system evolution depending on the different discretizations and on the time-stepping. As we can see, the Lagrange multipliers method does not reproduce properly, in general, the total energy invariance. The method is energetically unstable. Notice that for $\Delta t = 0.05$ the obtained results are acceptable, its energy ratio tends to one. However, for shortest time-steps the method experiences an important loss of energy after the collision. For the forward Euler and the symplectic Euler algorithms the great part of energy
loss happens at the first collision. Contrary, for the mid-point Euler and the trapezoidal rule algorithms it happens at the second collision. Moreover, if we focus on the transition between $\Delta t = 0.01$ and $\Delta t = 0.004$, we cannot appreciate any evidence of improvement on the energy conservation.

Figure 4.14: Long time integration using the Lagrange multipliers method for $\Delta t = 0.01$. 

Figure 4.13: Total energy of the system using the Lagrange multipliers method for different time-steps and algorithms.
Let us remark that even if in Figure 4.14 it seems, for $\Delta t = 0.1$ and $\Delta t = 0.05$, that the total energy tends to be conserved, it is only pure casualty that it behaves in such way. For random values of the initial speed, the system tends to loose all its energy for these time-steps.

Hence, the long-time integration cannot be tackled using the Lagrange multipliers method. As we can see in Figure 4.14 the methods is extremely dissipative at each collision, particularly, for the mid-point Euler and symplectic Euler discretization.

![Figure 4.15: Reaction force $\lambda$ calculated using the Lagrange multipliers method for different time-steps and algorithms.](image)

**Reaction forces**

The reaction forces $\lambda$ associated at each surface constraint are strongly dependent on the time-stepping and on the discretization. In addition, two of the discretization, the forward Euler and the symplectic Euler, offer an opposite behavior to the mid-point Euler and the trapezoidal rule. For the first couple, when the particle collides, it does not bounce, it remains onto the surface until the spring is not contracted -see Figure 4.11-. Contrary, the other two discretization reproduce properly the collision. Therefore, the reaction force $\lambda$
will take a different value for each case.

As we can see in Figure 4.15, there are two distinguishable behaviors. One has a peak of force when the particle collides. The other one has a lowest peak but the force is sustained over the time. The second behavior does not correspond to the reality, it is the result of an inappropriate method.

Notice that as the time-step decreases, the peak value of $\lambda$ increases for the two accurate discretization. Theoretically it should not occur, the value of the reaction force is a real number and it is delimited. Nevertheless, the fact of discretizing the time implies that the reaction force will be applied on the surface during a certain time-step and, therefore, the value of $\lambda$ will not match up with the analytical value of the Lagrange multipliers.

Linear momentum behavior

As exposed in Section 2.1.8, the total linear momentum $\mathbf{P}$ of the system should be conserved if no collision. In addition, the problem is constructed in such a way that after the right particle second collision, the system has to recover the initial linear momentum as explained in Section 4.3.1.

![Graphs showing linear momentum behavior for different time-steps and algorithms.](image)

Figure 4.16: Total linear momentum of the system calculated using the Lagrange multipliers method for different time-steps and algorithms.
The evolution of the total linear momentum has to be understood in line with the evolution of the total energy of the system. As we can see in Figure 4.16, the linear momentum has a similar recovery ratio to the total energy. This is due to the fact that when the particle collides, the energy falls and it has a direct impact on the linear momentum.

Let us remark that, as succeeded with the total energy of the system, the qualitative good results for $\Delta t = 0.1$ and $\Delta t = 0.05$ are a mere chance. It is just the result of our simply toy problem; for random initial speed values the method tends to dissipate all the energy and, consequently, to lose all its linear momentum.

### 4.2.3 Explicit algorithms. Störmer-Verlet (leap-frog method).

We have decided to separate the explicit algorithms of the implicit ones due to the fact that their computational costs are non-comparable in equality of step length. In addition, it is also interesting to confirm if the highest costs of the implicit algorithms are justified by an improvement of their qualitative performance.

We will tackle the Störmer-Verlet algorithm from two different points of view. Firstly, we will analyze it using the penalty method formulation for collisions and, finally, we will use it with the Lagrange multipliers method.

**Particles trajectory**

Let us start remarking that, as occurred with the implicit algorithms using the penalty method, the penalty value has a great influence on the obtained results. Indeed, this value establishes the constraint supposed elasticity and, moreover, it is a key point for the stability of the integration. We will illustrate it subsequently.

![Figure 4.17: Trajectory of $m_2$ integrated with the Störmer-Verlet scheme and using the penalty method.](image)

As we can see in Figure 4.17 and, in line with the implicit methods, the trajectory of the particle and the constraint violation strongly depend on the
penalty value. Nevertheless, the trajectory for high values of the penalty and short time steps is qualitatively improved. The reader will note that, as we have pointed previously, the trajectory after the second collision should be straight as it occurs for two different pairs of values \( \Delta t = 10^{-3} \) & \( \epsilon = 10^2 \) and \( \Delta t = 10^{-3} \) & \( \epsilon = 10^3 \). Obviously, the one with a highest value of the penalty has an improved performance due to the fact of its minor violation. Let us remark that these results have to be validated with an energetic analysis.

As regard the Lagrange multipliers, its results are disappointing. As illustrates the Figure 4.18, the particle does not bounce after the collision, it remains onto the surface constraint until the first particle reach to pull it out. Therefore, we will rule out this method.

![Figure 4.18: Trajectory of \( m_2 \) integrated with the Störmer-Verlet scheme and using the Lagrange multipliers method.](image)

**Energy performance**

In order to validate the obtained results it is required to analyze the total energy of the system. The leap-frog method produces interesting results regarding the influence of the penalty value on the stability of the system. As we have pointed out previously, every explicit method is very sensitive to sudden variations, as it is our case. Hence, the only way to tackle the integration of the trajectory is using very short time steps -as shorts as sudden are the variations. On the contrary, a non-balanced ratio \( \frac{\epsilon}{\Delta t} \) may lead to severe instabilities.

As we can see in Figure 4.20, the system tends to be unstable for every pair of values \( \epsilon \) and \( \Delta t \). Nevertheless, it is clear that the effect is more prominent for larger values of the time-step. Note that for \( \epsilon = 10^3 \) and \( \Delta t = 10^{-2} \) the total energy is triggered to the double. Unfortunately, this is an intrinsic effect of the explicit methods and they cannot be eliminated, just mitigated. In order to illustrate the phenomena, we offer in Figure 4.32 a comparison of the evolution of the total energy of the system for a long-time integration between two different ratios \( \frac{\epsilon}{\Delta t} \).
4.3 Conservative methods

4.3.1 Dissipative penalty method

The dissipative penalty method establishes the surface constraint as a spring layer as does the regular penalty method. In line with the ordinary method, it supposes an elasticity for the surface. However, and mutually to the ordinary method, the hypothesis of a perfect elastic collision is only satisfied when the spring elasticity tends to infinity.

In addition, the method is constructed in such a way that it is energetically stable. Indeed, \( \Delta \mathcal{E} = \frac{\Delta E}{\Delta t} \leq 0 \) as it is proved in Section 3.2.1. This construction ensures us that the method is dissipative and that its energy will not be triggered to infinity.

The main variable to adjust of the method is the penalty value, as succeeded with the ordinary method. However, even if this contact algorithm is designed to
be used with every method, we will only analyze it with the energy-momentum discretization.

**Particles trajectory**

Let us start analyzing the obtained particle trajectory for $m_2$ and for the different values of the penalty.

![Graph](image)

Figure 4.21: Trajectory of $m_2$ using the dissipative penalty method for different values of $\epsilon$.

As we can see in Figure 4.21 the method has difficulties to properly reproduce the collision. The constraint surface is largely violated as regards the time and the space. In addition, the method is very sensitive to the ratio $\frac{\Delta t}{\epsilon}$, i.e. for high values of $\epsilon$ it does not converge if the time-step is not really short. In order to illustrate the phenomenon we offer in Table 4.3.1 the convergence for different pairs of values. The reader will note that for low values of $\epsilon$ and short $\Delta t$ the algorithm does not converge neither. This is due to the linearization of the equations and that the iterations swing over the solution but do not achieve the assigned tolerance.

**Energy performance**

The algorithm is constructed to be energetically stable but, as discussed previously, the trajectory results have not been encouraging. We will tackle the energy evolution analyzing the influence of the penalty value on it.

As we can see in Figure 4.22, the system presents, no matter what penalty value is chosen, an important loss of energy after the collision. Even if the evolution of the energy satisfies the hypothesis of the method, the performance of the energy is not good.

The method is stable because it never increases the total energy. Nevertheless, the method does not delimit decreasings and, therefore, large amounts of energy can be lost when the collision. As it is in our case, when the particle bounce it looses speed and, consequently, kinetic energy.
Table 4.1: Convergence for different time-steps $\Delta t$ and for different values of the penalty $\epsilon$.

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>0.1</th>
<th>0.08</th>
<th>0.06</th>
<th>0.04</th>
<th>0.02</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>5</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>10</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>15</td>
<td>√</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>20</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>25</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>×</td>
</tr>
<tr>
<td>30</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>×</td>
</tr>
<tr>
<td>35</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>40</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>45</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>50</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

Figure 4.22: Trajectory of $m_2$ using the dissipative penalty method for different values of $\epsilon$.

The reader will note that as the penalty value increases the total energy is better conserved. This is an important differential fact to the regular penalty method. Let us remember that for admissible results, the regular method has to be used with a balanced $\frac{\Delta t}{\epsilon}$ ratio and, on the contrary to the dissipative method, increasing $\epsilon$ implied a bad energy performance.

As reflects the Table 4.3.1, there is no possibility to analyze the evolution of the invariant depending on the time-step due to the fact that chosen a penalty value there is a short range of $\Delta t$ that let the algorithm converge. Nevertheless, we have computed the solutions for low values of $\epsilon$ -they have a wider range- and the obtained results do not differ excessively from each one.

Finally, the low recovery rate of energy of the algorithm nullify it to be used for long-time integration purposes.
Linear momentum performance

Given the bad results of the particles’ trajectories and the energy evolution, the expected outcome is a low recovery rate after the collision. As we can see in Figure 4.22, the recovery rate is in line with the energy evolution in Figure 4.22.

Figure 4.23: Linear momentum evolution using the dissipative penalty method for different values of $\epsilon$.

4.3.2 Modified Lagrange multipliers

The modified Lagrange multipliers method is constructed in the same way as it is the regular one. The only difference between them is that the regular one satisfies the geometrical restrictions and the modified one satisfies the total energy invariance. Such difference is mathematically based on the reaction forces prediction. Then, the modified method projects the particle’s nodes onto the shifted surface $S^*$, while the regular method projects it on the real surface restriction $S$ -see Section 3.1.1-.

The problems with the calculus of the reactions forces and the introduction of a new gap are treated identically as in Section 4.2.2.

Particles trajectory

As we can see in Figure 4.24 the qualitative behavior of the particle is correct. Let us remember that due to the symmetry of the system and the hypothesis of a perfect elastic collision the trajectory of the particle after the collision should be straight. The integration method achieves to reproduce correctly this behavior even for large time-steps.

Let us remark that the method allows large violations of the surface constraint for long time-steps. On the contrary, for short time-steps this violation is substantially reduced as we can see in Figure 4.25. Then, as regards the qualitative behavior of the integration, the method used with a short time-stepping
Figure 4.24: Trajectory of $m_2$ using the modified Lagrange multipliers for different time-steps.

is really competitive in front of those exposed previously.

Figure 4.25: Detail of the surface violation using the modified Lagrange multipliers for different time-steps.

Energy performance

The algorithm is constructed in such a way that conserves the energy. Therefore, the invariance of the energy should be satisfied except for numerical errors. The reader will note that in Figure 4.26 there is only plotted the invariant evo-
olution for $\Delta t = 0.1$. This is due to the fact that other time-steps do not provide us more information. In addition, the variation of $E$ is so small that the axis of the figure have not been scaled properly -solutions have been computed with MATLAB and the error is under its precision.

![Graph](image1)

**Figure 4.26:** Energy evolution using the modified Lagrange multipliers method for $\Delta t = 0.1$.

**Linear momentum performance**

The linear momentum evolution is in line with the energy. Hence, its recovery rate should be satisfactory. As we can see in Figure 4.27, the recovery rate increases as decreases the time-step. For short time discretization the recovery rate tends to one.

![Graph](image2)

**Figure 4.27:** Linear momentum evolution using the modified Lagrange multipliers method for different time-steps.

The method reproduces correctly the behavior of the linear momentum.
4.4 Toy problem 2. Collision near corners.

In order to test the goodness of the comers approximation treatment we have designed the following problem, without immediate scientific interest, to illustrate in an intuitive way the behavior of the algorithms.

The problem consist in a two mass system united with an elastic spring and subjected to two orthogonal surface constraints as it is presented in Figure 4.28. The objective is to test if our approximations to collisions for long-time steps or complicated geometries are valid. Its formulation is defined in Section 3.1.3 and 3.2.3 for the Lagrange multipliers method and the modified one respectively. Nevertheless, we will only test the approximation for the modified Lagrange multipliers method with the energy-momentum discretization due to the satisfactory results of the previous test.

The initial settings of $m_1$ and $m_2$ are:

$$q_1 = \begin{pmatrix} 0.9 \\ 0.9 \\ 0 \end{pmatrix} \quad \text{and} \quad q_2 = \begin{pmatrix} 1.9 \\ 0.9 \\ 0 \end{pmatrix}$$

$$v_1 = v_2 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

and the constraint surfaces are defined by a point $\beta \in S_1 \& S_2$ and its respective normals:

$$\beta = \begin{pmatrix} 2.05 \\ 1.99 \\ 0 \end{pmatrix}$$

$$n_1 = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}, \quad n_2 = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$$
The expected behavior is that for long time-steps the second particle violates both surfaces. Therefore, two solutions are possible: diminish the time-step or use an approximation of the reaction forces. We will compare both to validate the approximation. It is interesting to remark that our approximation has not physical sense but it can be understood like a time average of the reaction forces applied on the particle. A sketch of the approximation is offered in Figure 4.29. As we can see in it, the approximation has no physical sense. It is constructed in such a way that satisfies our energetic requirements. Therefore, our objective is to check if the reaction forces average achieve to reproduce the particle’s bounce. In order to check it we will compare the evolution of the trajectory of the particles for different time steps.

As we have pointed previously, and since there is no analytical solution available for non-linear ODE, we can only evaluate our algorithms performance comparing the obtained results to some known flags of simple problems. In this case, and basing us on the great qualitative results obtained by the modified Lagrange multipliers algorithm, we will compare the trajectory of $m_2$ computed with a short time-step -setting it as the reference solution- to one computed with a longer one but introducing the exposed approximation.

As we can see in Figure 4.30 it exists a wide casuistry depending on the time-step. Nevertheless, one prominent phenomenon is the significant difference between the trajectories. The reader will remember that the modified Lagrange multipliers method permits the violation of the constraints at exchange of conserving the energy. This guarantees a softer response to the system in front of collisions. Nevertheless, collisions near corners involve abrupt speed changes and the method is not constructed to integrate such sudden variations. The average of the reactions forces is the only way to tackle the problem without decreasing the time-step.

It is interesting to remark that even the large variability of the position for a fixed time the behavior of the system is to tend in the same direction. Note that for $\Delta t = 0.1$ and $\Delta t = 0.05$ the tangent to the trajectory once the corner
Figure 4.30: Trajectory of collision into a corner integrated for different time-steps.

is left back is really similar. The Figure 4.31 illustrates this phenomena.

Figure 4.31: Particle $m_2$ speed after a collision into a corner integrated for different time-steps.

Unfortunately, long-time integration is really difficult to reproduce properly numerically. In order to illustrate it, we offer the Figure 4.32 where there are plotted the trajectories of $m_2$ for different time-steps and, where there is added another pair of restrictions coinciding with the axes. Even the density of lines and its chaotic structure it is easy to note that the symmetry of the system is broken for long integration times. Indeed, the vertical speed of both particles should be constant and equal to $\pm 1$. In addition, the constraints are placed in such a way that they ensure the invariance of $v_y$. According to
that, the trajectory of both particles should tend to describe a closed polygon. Nevertheless, in a certain time the average reaction is not properly estimated and the symmetry is broken. That phenomena leads to a chaotic problem that has no relation with the original one.

Hence, the obtained results are encouraging because even if they do not conserve the stability of the system for remarkable long integration times, they confirm the qualities of our suggested approximation. Note that the trajectory computed with our approximation tends to describe the same closed circuit that does the regular method with a ten times smaller time-step.

Finally, let us remark that the suggested approximation will be extremely useful when short time-steps are used and its decrease could cause conditioning problems. In addition, the approximation combined with an adaptable time-step will be a powerful tool to tackle collisions into complicated surfaces.

4.5 Toy problem 3. Icosahedron collision.

The algorithms have been previously tested with few particles and simply geometric structure problems and its performance has been extrapolated to all kind of problems. Therefore, it is necessary to check them in more complicated situations to ensure their reliability. For this reason we have designed a non realistic problem consistent in an icosahedron constructed of masses $m = 1$ for all its vertex and springs of elastic constant $k = 5$ for all the edges -see Figure 4.33-.

We only offer the position of the icosahedron integrated with the Lagrange multipliers method due to the fact that, as exposed previously, the performance of the other methods is limited. Nevertheless, we offer in Figure 4.34 a comparative graphic between the different algorithms and methods used until now.

As we can see in Figure 4.35, the solution obtained by the modified Lagrange multipliers method is qualitatively good because the icosahedron first collides then it is deformed and, finally, it bounces as it continues to being deformed.
As regards the quantitative performance, we can see in Figure 4.34 the evolution of the total energy of the system for different methods and algorithms. Note that MLM and ML correspond to the modified Lagrange multipliers method and the regular one respectively. In line with it, MPM and PM corresponds to the modified penalty method and the regular one. The obtained results support the previous ones. The only way to preserve the total energy of the system is to use the energy-momentum algorithm combined with the modified Lagrange multipliers method. As regard the others algorithms, there is no need to go into details but, just remark that the loss of energy rate is large and that even if if they are stable, the obtained results are not admissible.
Figure 4.35: Position of the icosahedron for different times and integrated with the modified Lagrange multipliers method.
5 Conclusions and future work

5.1 Conclusions

The main objective of this work has been to present some time integration algorithms for conservative particle systems with constraints. These algorithms are inspired by the ones applied in rigid body contacts using finite element methods. In addition, we have proposed a simplification for complex geometries.

The integration schemes used have been the Forward Euler, the Backward Euler, the Symplectic Euler, the Trapezoidal Rule, the Midpoint Euler and the Energy-momentum conserving method. All these schemes have been adapted to reproduce the Karush-Kuhn-Tucker conditions using different methods. Specifically, the regular penalty method, the Lagrange multipliers method and their respective adaptations to energy conserving schemes.

First, the penalty method establishes the surface constraint as a spring layer with a certain elasticity. However, we are simulating perfect elastic collisions and the hypothesis of the method is only satisfied when the spring elasticity tends to infinity. The method is easy to implement but, obviously, it does not respect the geometrical boundary. Indeed, if the surface constraint is treated as a spring layer, surface violations are allowed. In addition, the fact of introducing a spring elasticity leads us to introduce a new variable in our model, always subjected to empiricism and the type of problem that we are solving. Moreover, the numerical solution is strongly dependent on the used time-step so it leaves us in the difficult situation to evaluate the goodness of the method in a global point of view.

The results obtained with the regular penalty method have been unsatisfactory. The natural tendency of the method is to dissipate the energy of the system when particles collide regardless of the numerical scheme used. Furthermore, as increases the elasticity of the spring, the dissipative effect of the collision increases. Then, it is necessary to fix a minimum ratio between the time-step and the elasticity of the spring.

As regards the conserving penalty, the problems of the dissipation of the energy have been overcome. However, the use of high elasticities that guarantee the geometrical boundary lead the algorithm to conditioning problems. Nevertheless, the noticeable improvement done permits a more accurate long-time integration and a more realistic behavior of collisions using larger time-steps.

The other method treated is the Lagrange multipliers. The method is based on the calculus of the reaction forces that appear when particles collide into constraints. The reaction force is calculated imposing the position of the particle to be onto the surface constraint after the violation.

As regards the conserving penalty, the problems of the dissipation of the energy have been overcome. However, the use of high elasticities that guarantee the geometrical boundary lead the algorithm to conditioning problems. Nevertheless, the noticeable improvement done permits a more accurate long-time integration and a more realistic behavior of collisions using larger time-steps.

The other method treated is the Lagrange multipliers. The method is based on the calculus of the reaction forces that appear when particles collide into constraints. The reaction force is calculated imposing the position of the particle to be onto the surface constraint after the violation.

The results obtained with the regular Lagrange multipliers method are strongly dependent on the used numerical scheme. The Backward and Symplectic Euler present noticeable difficulties to reproduce correctly the collision of the particles and they are perceptibly dissipative. Conversely, the Trapezoidal Rule and the Midpoint Euler method present stability problems due to the continuum increase of energy.

As regards the modified Lagrange multipliers method, the conservation of the energy is achieved to the detriment of respecting the geometrical boundaries. Thanks to the invariance of the energy, a more accurate long-time integration is achieved, and other invariants such the total linear or angular momentum are
better preserved.

Finally, for complex boundaries the main algorithms present many difficulties. Let us assume that boundaries came from approximating them by a succession of planes. The presence of many of them combined with a high speed of the particles or a large time-step will entail the violation of more than one constraint. Obviously, this phenomenon is not physically possible but the numerical treatment of the problem leads to this kind of situations. The classical way to tackle this issue is to use adaptive time-step methods. However, as we discussed previously, diminishing the time-step may entail conditioning problems and the non-convergence of the algorithms.

To resolve these problems we have proposed for the Lagrange multipliers method an average reaction force, based on the decomposition of the reaction in a covariant and contravariant basis that corresponds to the violated surfaces. The obtained results have been encouraging, the qualitative behavior of the system has improved in equality of computational costs.

5.2 Future work

The algorithms developed in this work include impacts into moving surfaces but they have not been tested. We think that they should be checked in order to save the complex numerical difficulties that they will entail. In addition, and in line with the aforementioned, it will be interesting to develop reliable algorithms to solve impact between particles problems. This kind of problems are usually faced in molecular dynamics. We subsequently suggest a possible formulation for this problem.

The contact problem may be extrapolated to a problem of impacts between particles. Owing to the fact that particles are not treated here as volumetric elements, it is necessary to define a gap $G$ that will divide the space for every time step between a contact zone and a non contact zone. Thence, the surface $S$ is here described by the sphere of radius $G$ and center $q_i$, and its normal vector $n$ is calculated for every pair of particles $q_i$ and $q_j$ as:

$$n_{i,j} = \frac{q_j - q_i}{\|q_j - q_i\|}.$$  

Therefore, the impact distance $d_{i,j}^*$ between two particles is:

$$d_{i,j}^* = -\|q_j - q_i\| + G,$$

with admissible positions negative defined.

Once the impact distance is defined, the treatment of the problem is identical to the surface moving formulation. Nevertheless, the sudden variations of the surface $S$ will entail complex numerical problems such as the piercing of $S$. Moreover, the small size of the sphere radius will require very short time-steps that will entail conditioning problems.

Geometrical variables are represented in Figure 5.1.
Figure 5.1: Geometrical variables for impacts between particles.
6 Appendix

6.1 Calculus of the jacobian matrix

6.1.1 Trapezoidal Rule Jacobian

The equilibrium equations of our system are:

\[ g := M \frac{\Delta v}{\Delta t} + \frac{1}{2} (\nabla U(q_{n+1}) + \nabla U(q_n)) = 0 \]

\[ \Delta q_{n+\frac{1}{2}} = \frac{\Delta q}{\Delta t} \]

As result we have the following system of non-linear equations,

\[ \frac{2M}{\Delta t} (\frac{q_{n+1} - q_n}{\Delta t} - v_n) + \frac{1}{2} (\nabla U(q_{n+1}) + \nabla U(q_n)) = 0 \]

We define for every pair of particles \( q^1 \) and \( q^2 \) connected with an elastic spring of constant \( k \),

\[ q = \begin{pmatrix} q^1 \\ q^2 \end{pmatrix} \]

Then,

\[ U = \frac{1}{2} k (\sqrt{(q^1 - q^2)(q^1 - q^2)^T} - l_0)^2 = \frac{1}{2} k (l - l_0)^2 \]

\[ \frac{\partial U}{\partial t} = k(l - l_0) \]

\[ \nabla q_{n+1} l_{n+1} = \nabla q_{n+1} (l) = \frac{1}{l} \left( \frac{q^1 - q^2}{q^2 - q^1} \right) \]

So,

\[ \nabla q_{n+1} U = k (1 - \frac{l_0}{l_{n+1}}) \left( \frac{q^1 - q^2}{q^2 - q^1} \right) \]

Moreover,

\[ \nabla q_{n+1} (1 - \frac{l_0}{l_{n+1}}) = \frac{l_0}{l_{n+1}} \left( \frac{q^1 - q^2}{q^2 - q^1} \right) \]

\[ \nabla q_{n+1} \left( \frac{q^1 - q^2}{q^2 - q^1} \right) = \begin{bmatrix} I_3 & -I_3 \\ -I_3 & I_3 \end{bmatrix} \]

We just need to calculate the inertial part gradient,

\[ \nabla q_{n+1} \left( \frac{2M}{\Delta t} \right) = \frac{2M}{\Delta t} \left[ I_{6,6} \right] \]

Finally,

\[ J_g = \frac{2M}{\Delta t^2} \left[ I_6 \right] + \frac{k}{2} \frac{l_0}{l_{n+1}} \left( \frac{q^1 - q^2}{q^2 - q^1} \right) \otimes \left( \frac{q^1 - q^2}{q^2 - q^1} \right) + \frac{k}{2} (1 - \frac{l_0}{l_{n+1}}) \begin{bmatrix} I_3 & -I_3 \\ -I_3 & I_3 \end{bmatrix} \]

59
6.1.2 Back Euler Jacobian

Our system equilibrium equations are:

\[ g := M \frac{\Delta v}{\Delta t} + \nabla U(q_{n+1}) = 0 \]

\[ v_{n+1} = \frac{\Delta q}{\Delta t} \]

Operating in the same way as before we obtain the following jacobian matrix of the nonlinear system \( g \)

\[ J_g = \frac{M}{\Delta t^2} [I_6] + k \frac{l_0}{I_{3n+1}^3} (q^4 - q^2) \otimes (q^4 - q^2) + k(-l_0) \left[ \begin{array}{cc} I_3 & -I_3 \\ -I_3 & I_3 \end{array} \right] \]

6.1.3 Midpoint Euler Jacobian

\[ g := M \frac{\Delta v}{\Delta t} + \nabla U(q_{n+\frac{1}{2}}) = 0 \]

\[ v_{n+\frac{1}{2}} = \frac{\Delta q}{\Delta t} \]

\[ \frac{2M}{\Delta t} \left( 2 \frac{q_{n+\frac{1}{2}} - q_n}{\Delta t} - v_n \right) + \nabla U(q_{n+\frac{1}{2}}) = 0 \]

\[ \nabla q_{n+1} \otimes \nabla U(q_{n+\frac{1}{2}}) = \frac{1}{2} [I_6] \]

\[ \nabla q_{n+\frac{1}{2}} \otimes \nabla U(q_{n+\frac{1}{2}}) = k \frac{l_0}{I_{3n}^3} q \otimes q + k(1 - \frac{l_0}{T}) \left[ \begin{array}{cc} I_3 & -I_3 \\ -I_3 & I_3 \end{array} \right] \]

\[ \nabla q_{n+1} \otimes \nabla U(q_{n+\frac{1}{2}}) = \frac{k}{2} \left( \frac{l_0}{T} q \otimes q + \frac{l_0}{T} \right) \left[ \begin{array}{cc} I_3 & -I_3 \\ -I_3 & I_3 \end{array} \right] \]

\[ \nabla q_{n+1} \otimes \frac{2M}{\Delta t} \left( 2 \frac{q_{n+\frac{1}{2}} - q_n}{\Delta t} - v_n \right) = \frac{4M}{\Delta t^2} [I_6] \]

\[ J_g = \frac{2M}{\Delta t^2} + \frac{k}{2} \left( \frac{l_0}{T} q \otimes q + \frac{l_0}{T} \right) \left[ \begin{array}{cc} I_3 & -I_3 \\ -I_3 & I_3 \end{array} \right] \]
6.1.4 Energy Momentum Jacobian

\[ g := M \frac{\Delta v}{\Delta t} + \nabla U(q_{n+1}, q_n) = 0 \]

\[ v_{n+\frac{1}{2}} = \frac{\Delta q}{\Delta t} \]

\[ 2M \frac{\Delta}{\Delta t} \left( \frac{q_{n+\frac{1}{2}} - q_n}{\Delta t} - v_n \right) + \nabla U(q_{n+1}, q_n) = 0 \]

\[ q_{n+\frac{1}{2}} = \begin{pmatrix} q_{n+\frac{1}{2}}^1 - q_{n+\frac{1}{2}}^2 \\ q_{n+\frac{1}{2}}^2 - q_{n+\frac{1}{2}}^1 \end{pmatrix} \]

\[ \nabla U(q_{n+1}, q_{n+1}) = \Delta U \frac{1}{\Delta t} \frac{l_{n+\frac{1}{2}}}{q_{n+\frac{1}{2}}} \]

\[ \frac{\Delta U}{\Delta t} = \frac{1}{2} k (l_{n+1} + l_n - 2l_0) \]

\[ l_{n+\frac{1}{2}} = \sqrt{(q_{n+\frac{1}{2}}^1 - q_{n+\frac{1}{2}}^2)(q_{n+\frac{1}{2}}^2 - q_{n+\frac{1}{2}}^1)} \]

\[ \nabla_{q_{n+1}} \left[ \nabla (q_{n+1}, q_n) \right] = \nabla_{q_{n+1}} \frac{\Delta U}{\Delta t} \frac{l_{n+\frac{1}{2}}}{q_{n+\frac{1}{2}}} = \]

\[ = (\nabla_{q_{n+1}} \frac{\Delta U}{\Delta t}) \frac{l_{n+\frac{1}{2}}}{q_{n+\frac{1}{2}}} \]

\[ \nabla_{q_{n+1}} \frac{\Delta U}{\Delta t} \frac{l_{n+\frac{1}{2}}}{q_{n+\frac{1}{2}}} = \frac{k}{2l_{n+\frac{1}{2}}l_{n+1}} q_{n+\frac{1}{2}} \otimes q_{n+1} \]

\[ \nabla_{q_{n+1}} \left( \frac{q_{n+\frac{1}{2}}}{l_{n+\frac{1}{2}}} \right) = \frac{1}{2l_n + \frac{1}{2}} \left[ I_3, -I_3 \right] - \frac{1}{2l_{n+\frac{1}{2}}l_{n+1}} q_{n+\frac{1}{2}} \otimes q_{n+1} \]

\[ J_g = \frac{2M}{\Delta t^2} + \frac{k}{2l_n + \frac{1}{2}} \left( \frac{l_0}{l_{n+1}l_{n+\frac{1}{2}}} q_{n+\frac{1}{2}} \otimes q_{n+1} + (l_{n+12} - l_0) \left[ I_3, -I_3 \right] \right) \]
References


