Contact Dynamics in Robotics

Modeling and efficient resolution

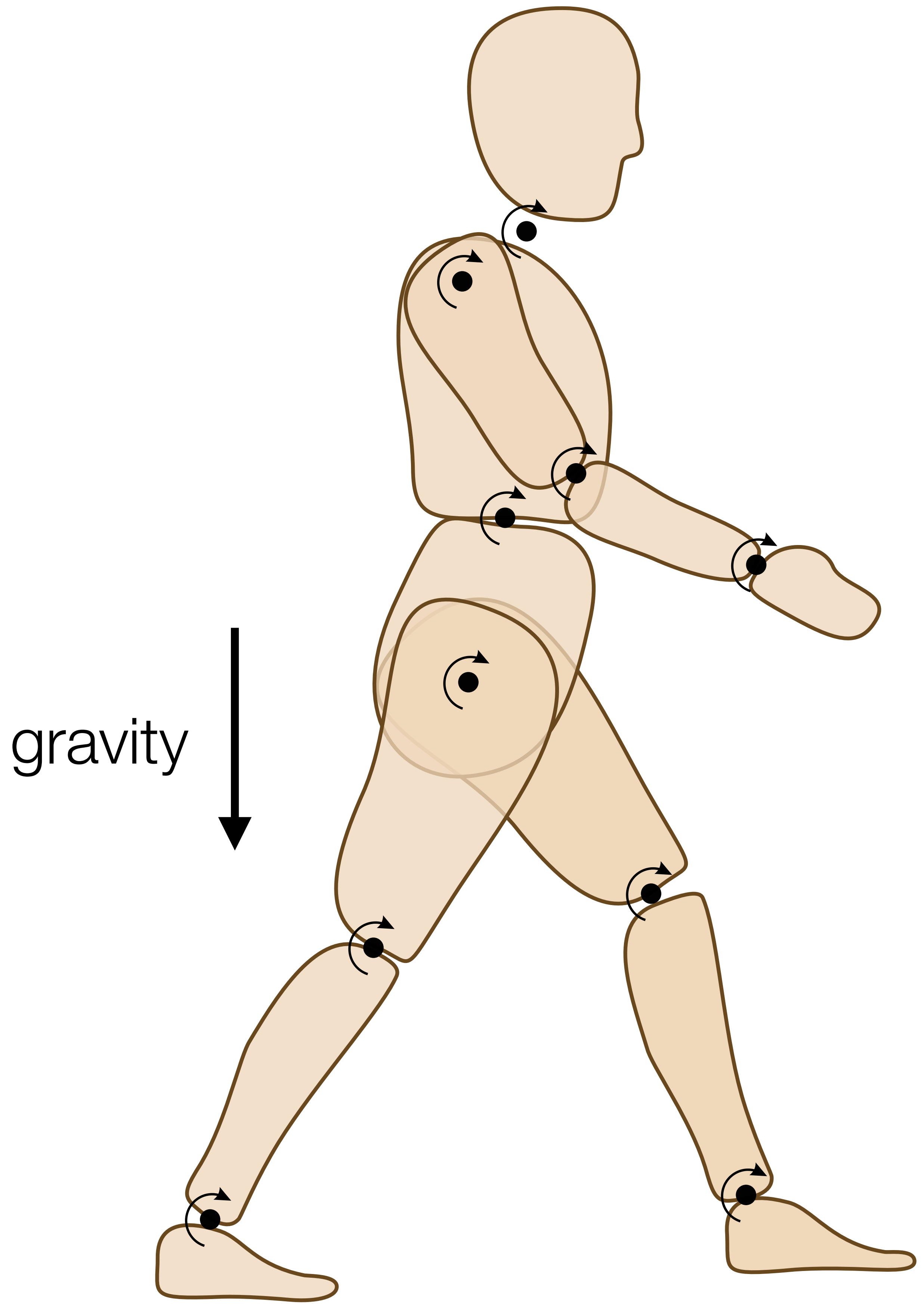
memmo
Memmo Summer School

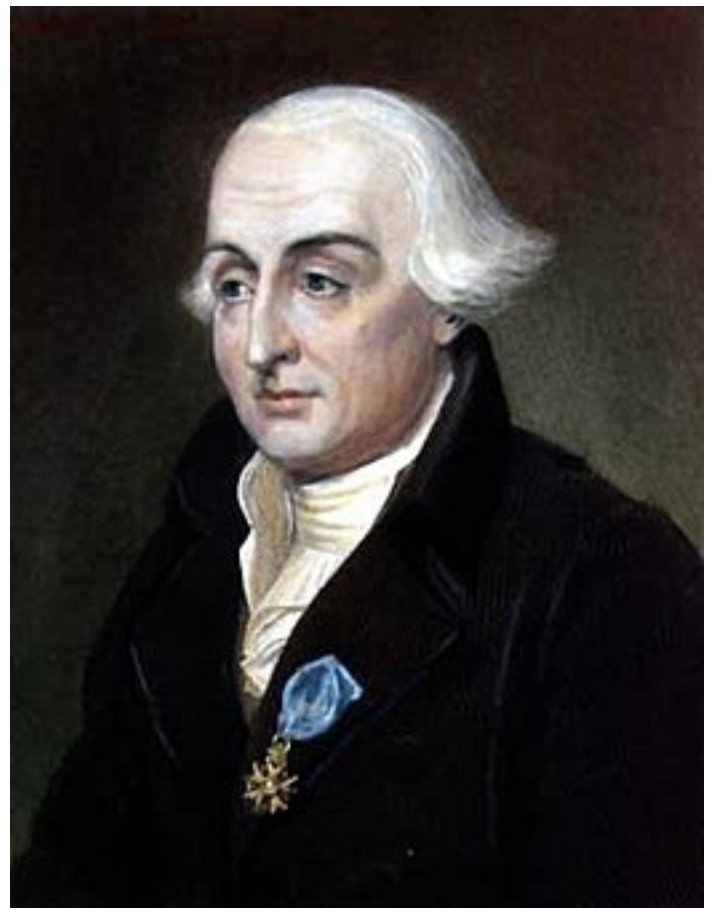
Justin Carpentier
Researcher, INRIA and ENS, Paris

PR[AI]RIE
PaRis Artificial Intelligence Research InstitutE

ENS
ÉCOLE NORMALE SUPÉRIEURE

Contact: the Physical Problem





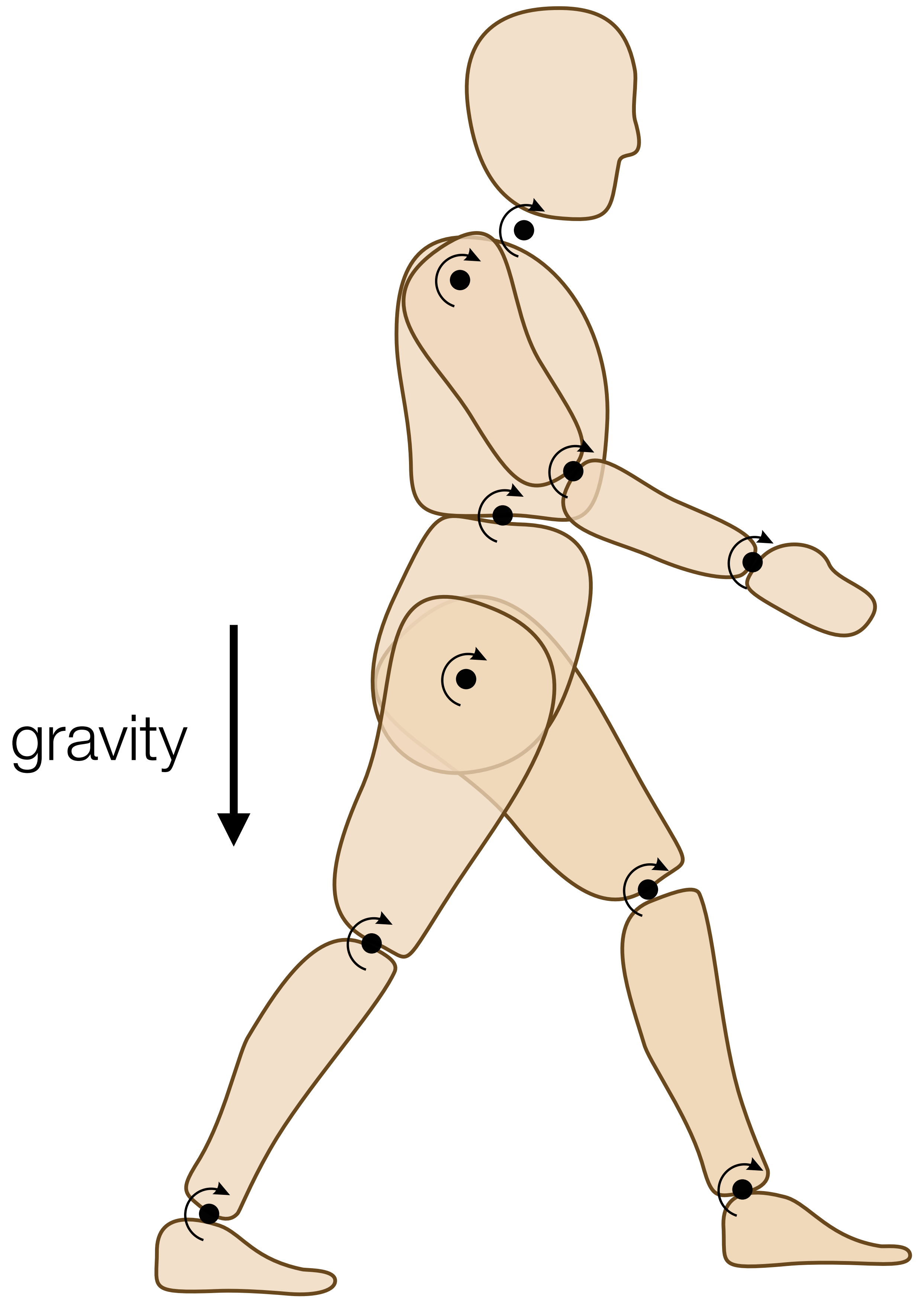
Joseph-Louis Lagrange

The poly-articulated system dynamics
is driven by the so-called **Lagrangian** dynamics:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau$$

Mass Matrix — Coriolis centrifugal — Gravity — Motor torque

Contact: the Physical Problem



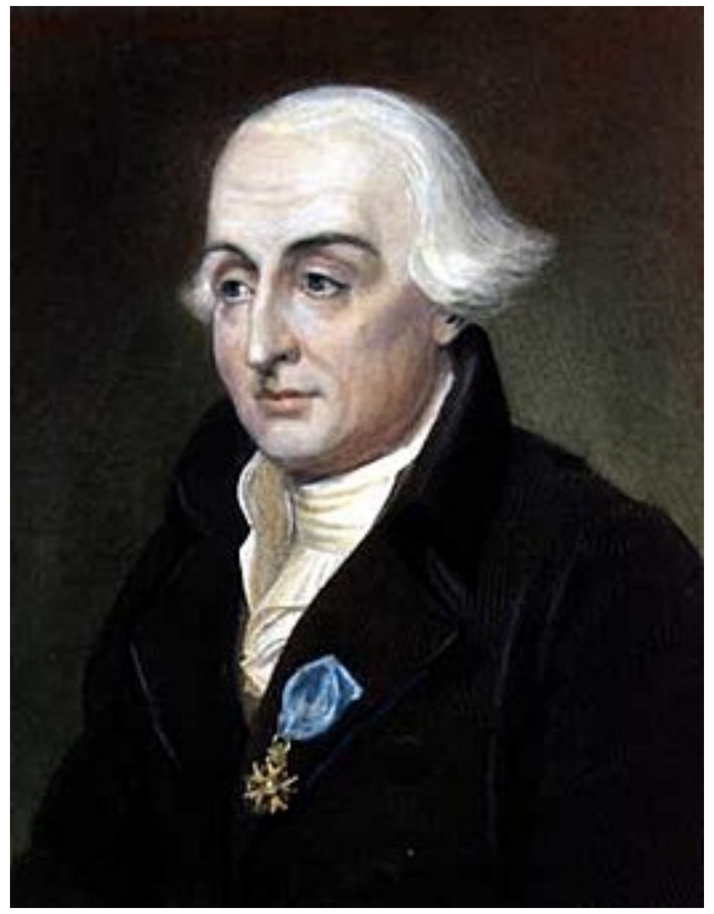
Joseph-Louis Lagrange

The poly-articulated system dynamics
is driven by the so-called **Lagrangian** dynamics:

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau$$

Mass Matrix	Coriolis centrifugal	Gravity	Motor torque

Contact: the Physical Problem



Joseph-Louis Lagrange

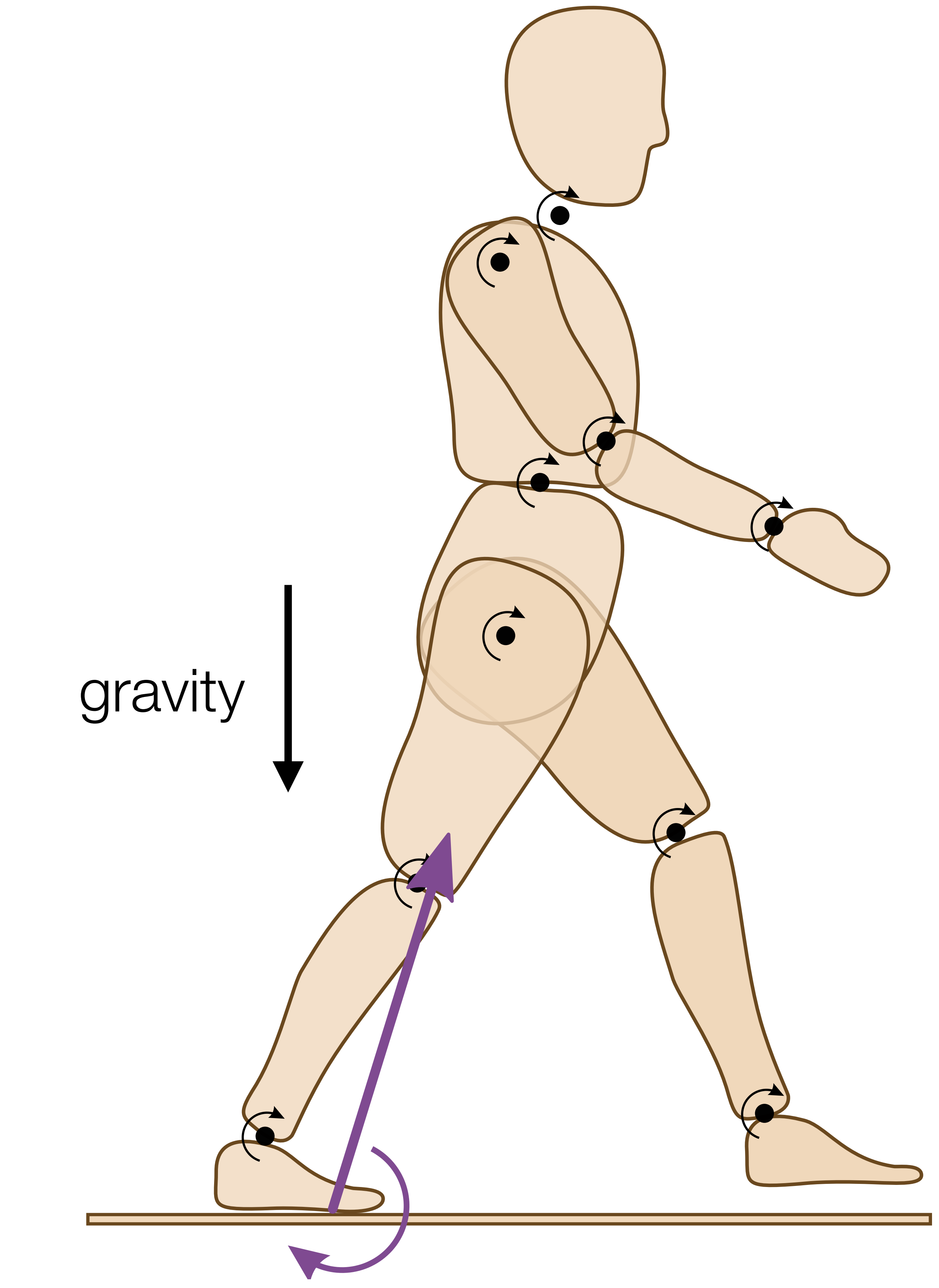
The poly-articulated system dynamics
is driven by the so-called **Lagrangian** dynamics:

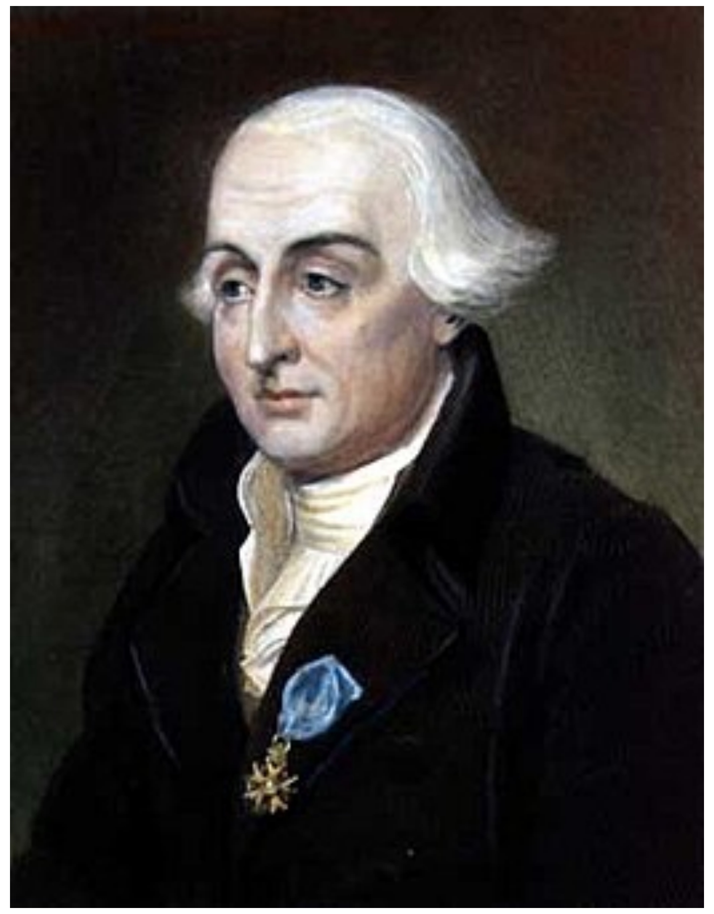
gravity ↓

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau$$

Mass Matrix — Coriolis centrifugal — Gravity — Motor torque

Contact: the Physical Problem





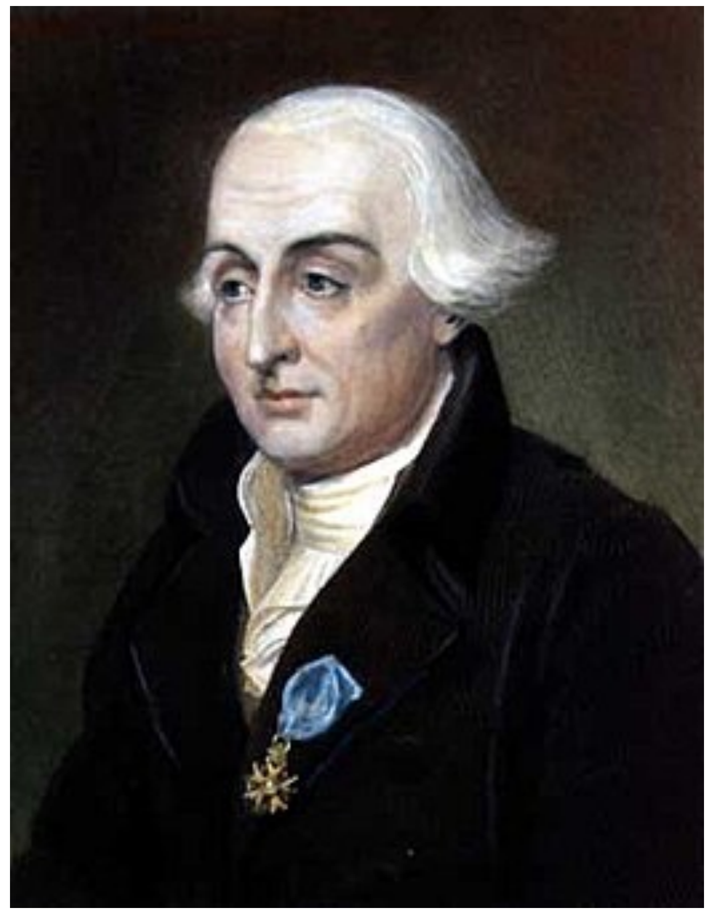
Joseph-Louis Lagrange

The poly-articulated system dynamics
is driven by the so-called **Lagrangian** dynamics:

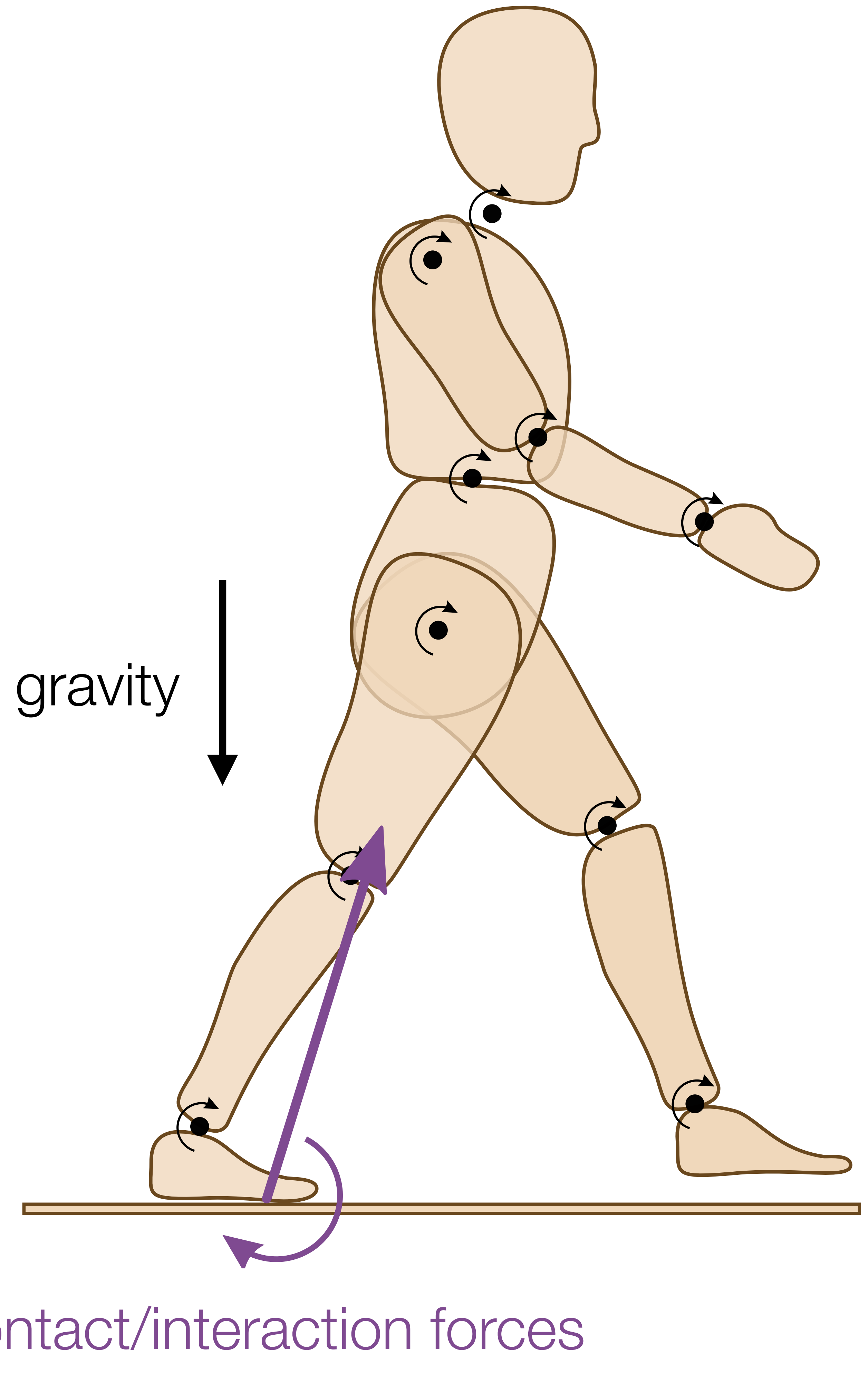
$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau$$

Mass Matrix — Coriolis centrifugal — Gravity — Motor torque

Contact: the Physical Problem



Joseph-Louis Lagrange



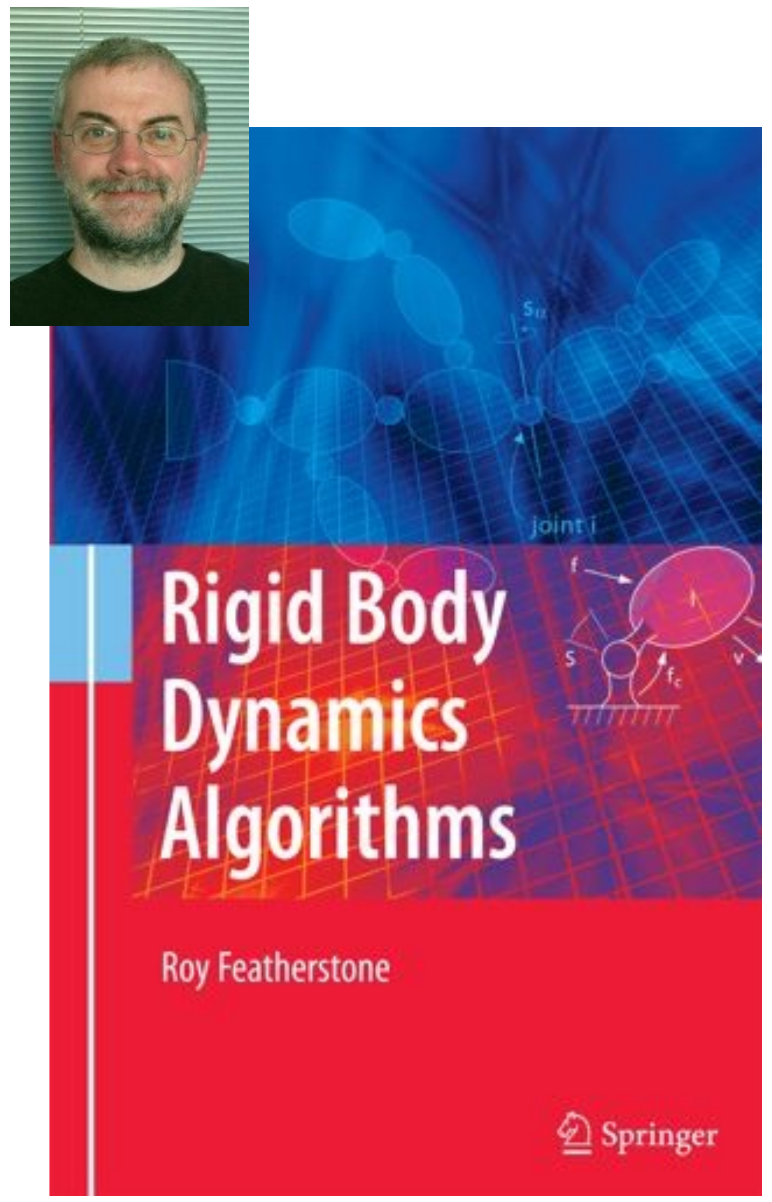
The poly-articulated system dynamics
is driven by the so-called **Lagrangian** dynamics:

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau + J_c^\top(q)\lambda_c$$

Mass Matrix — Coriolis centrifugal — Gravity — Motor torque — External forces

The Rigid Body Dynamics Algorithms

Goal: exploit at best the **sparsity** induced by the kinematic tree



Roy Featherstone

The Articulated Body Algorithm

$$\ddot{q} = \textbf{ForwardDynamics}\left(q, \dot{q}, \tau, \lambda_c\right)$$

Simulation

Control

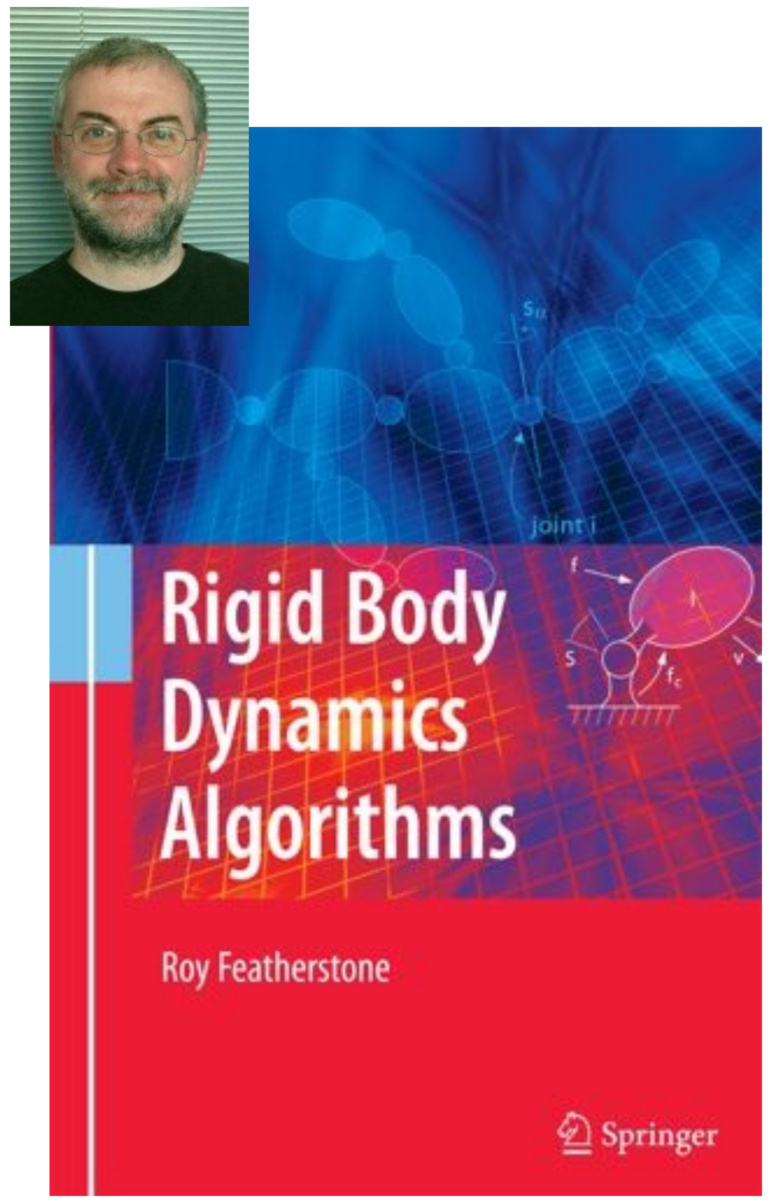
$$\tau = \textbf{InverseDynamics}\left(q, \dot{q}, \ddot{q}, \lambda_c\right)$$

The Recursive Newton-Euler Algorithm

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau + J_c^\top(q)\lambda_c$$

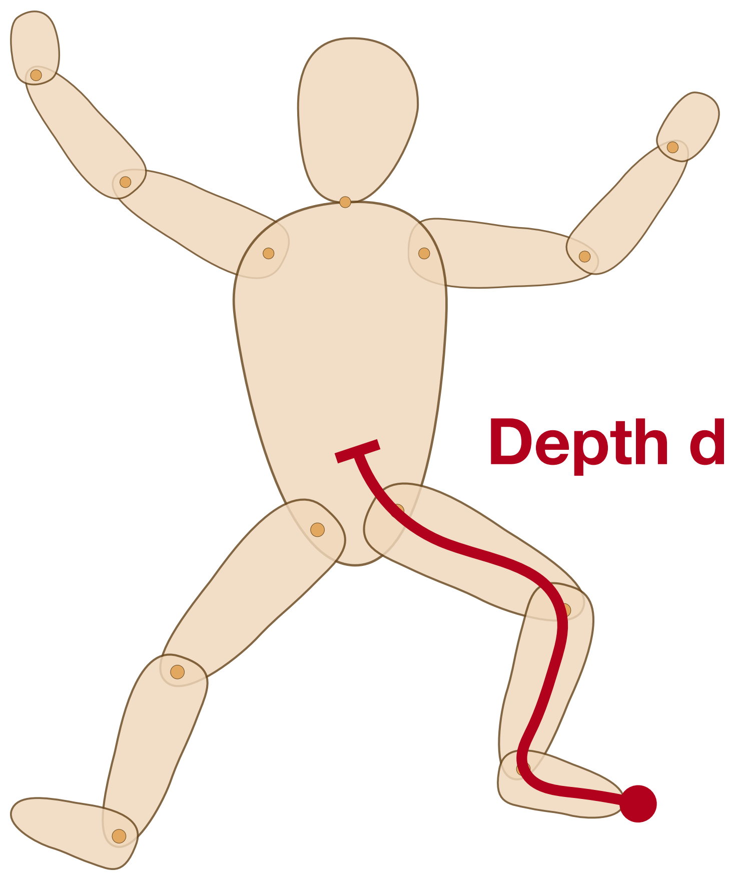
Mass Matrix	Coriolis centrifugal	Gravity	Motor torque	External forces

The Rigid Body Dynamics Algorithms



Roy Featherstone

Goal: exploit at best the **sparsity** induced by the kinematic tree



The Articulated Body Algorithm

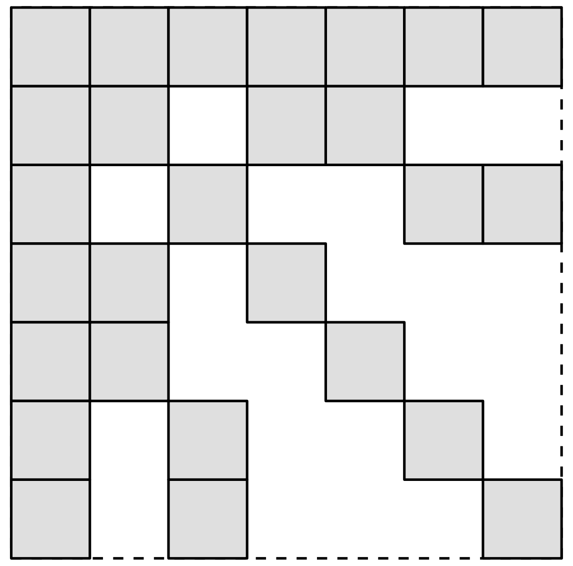
$$\ddot{q} = \textbf{ForwardDynamics}\left(q, \dot{q}, \tau, \lambda_c\right)$$

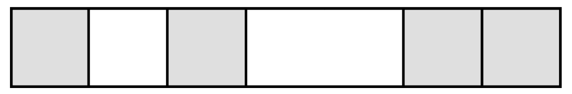
Simulation

Control

$$\tau = \textbf{InverseDynamics}\left(q, \dot{q}, \ddot{q}, \lambda_c\right)$$

The Recursive Newton-Euler Algorithm





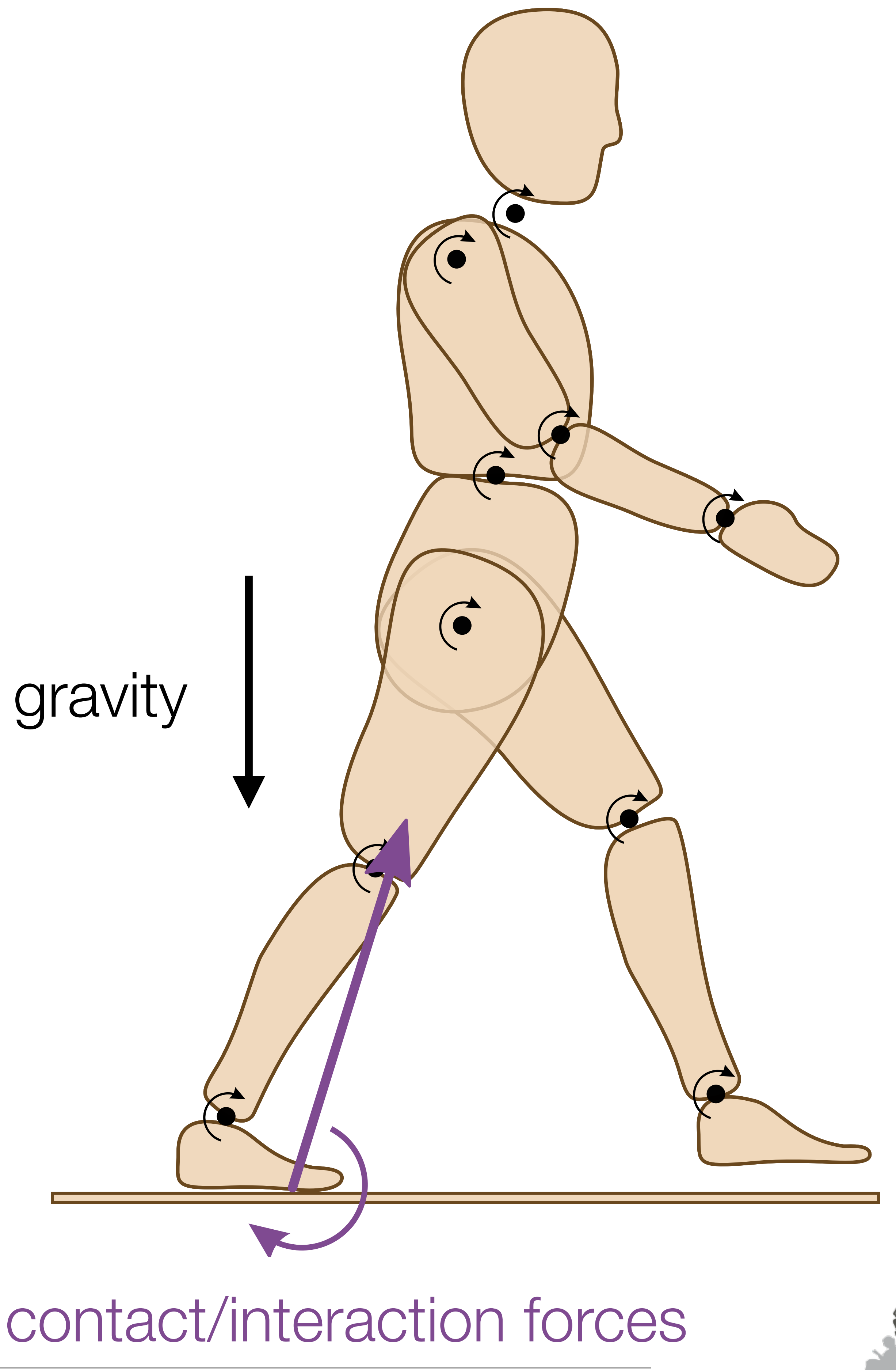
$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau + J_c^\top(q)\lambda_c$$

Mass Matrix | Coriolis centrifugal | Gravity | Motor torque | External forces

Gaol of this class

Understand the **various approaches** of the state of the art to compute λ_c in:

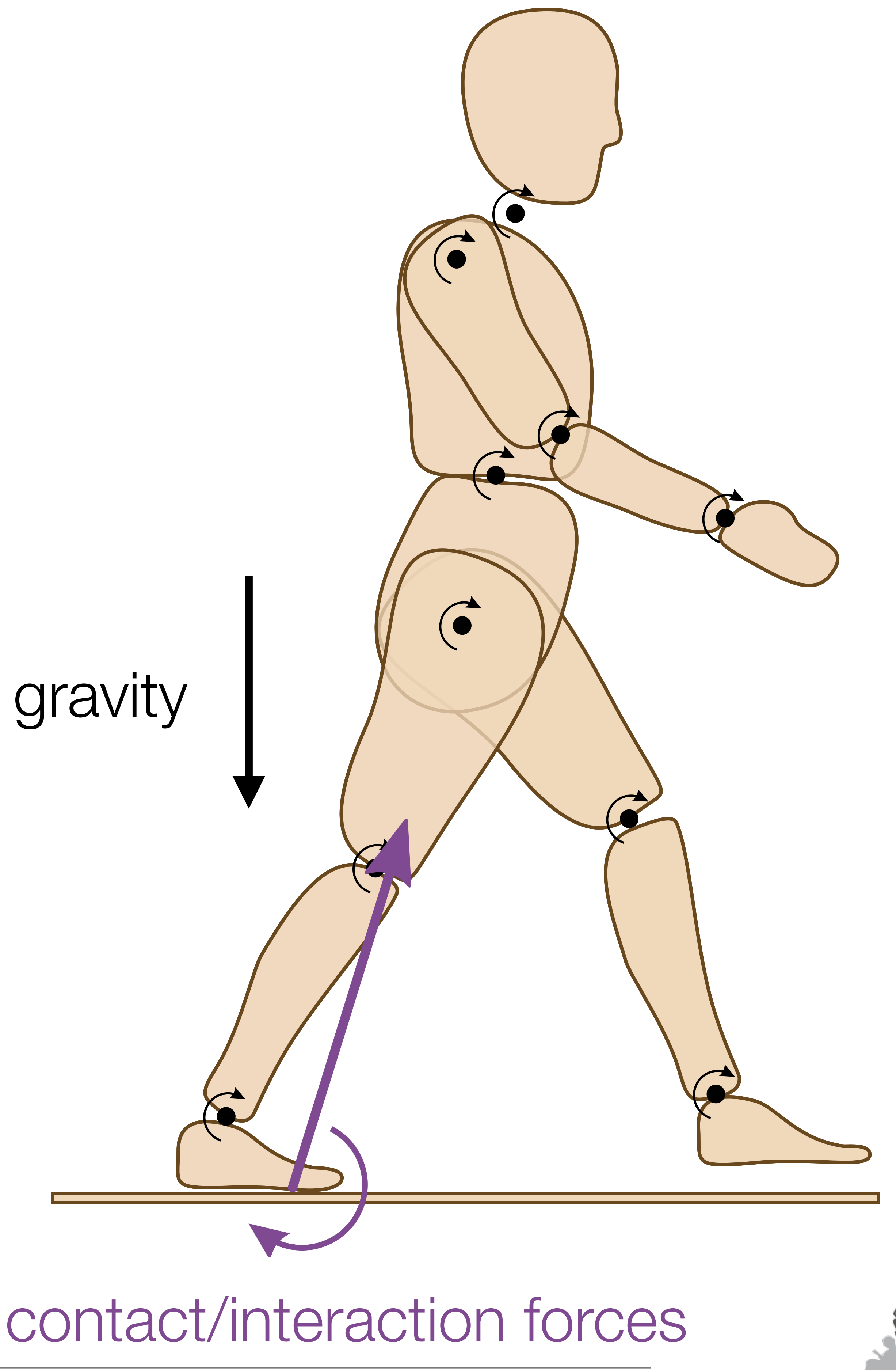
$$M(q)\ddot{q} \; + \; C(q,\dot{q}) \; + \; G(q) \; = \; \tau \; + \; J_c^\top(q)\,\lambda_c$$



Gaol of this class

Understand the **various approaches** of the state of the art to compute λ_c in:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau + J_c^\top(q)\,\lambda_c$$

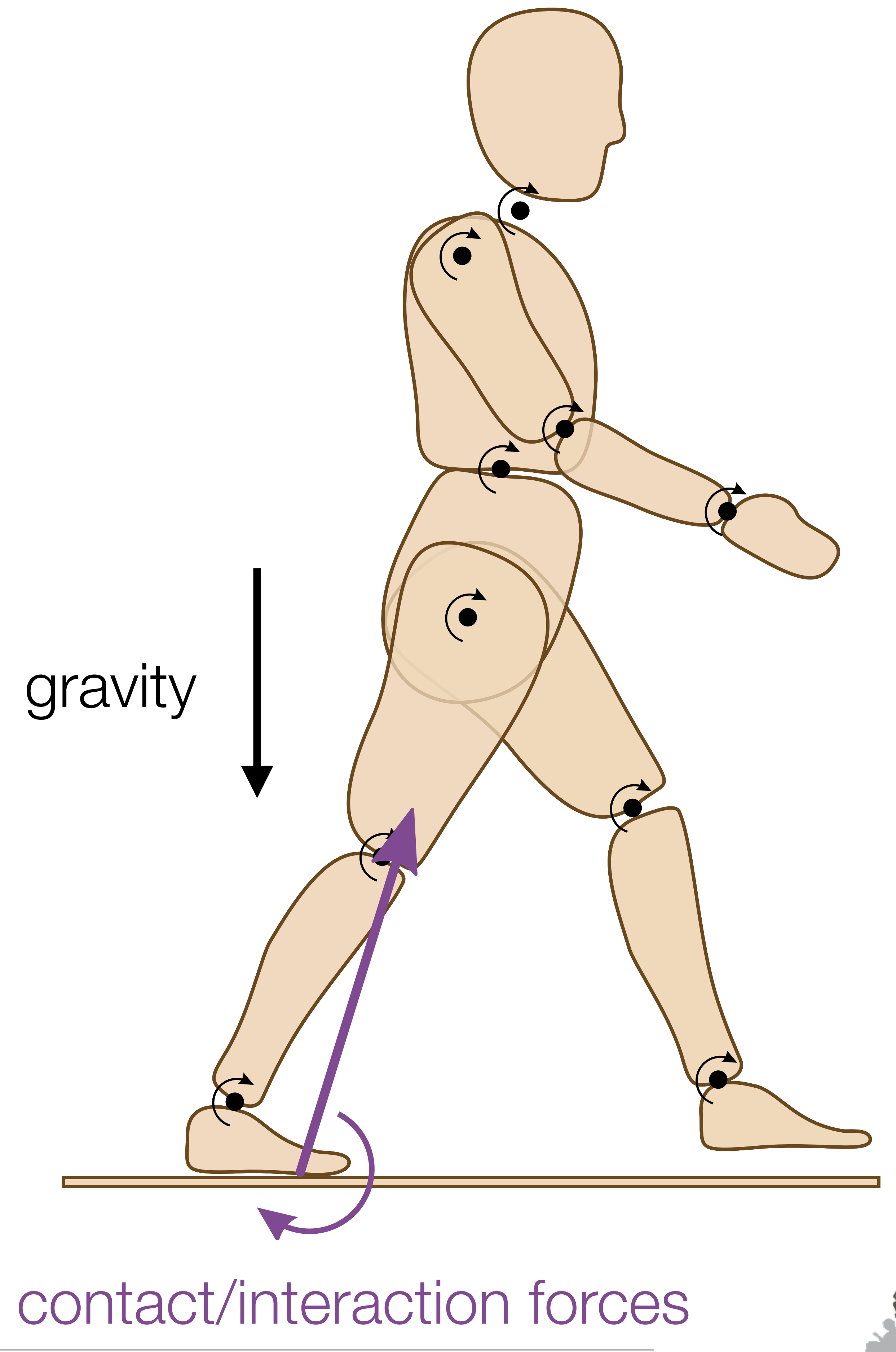


Gaol of this class

Understand the **various approaches** of the state of the art to compute λ_c in:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau + J_c^\top(q)\,\lambda_c$$

Soft contact — spring-damper model



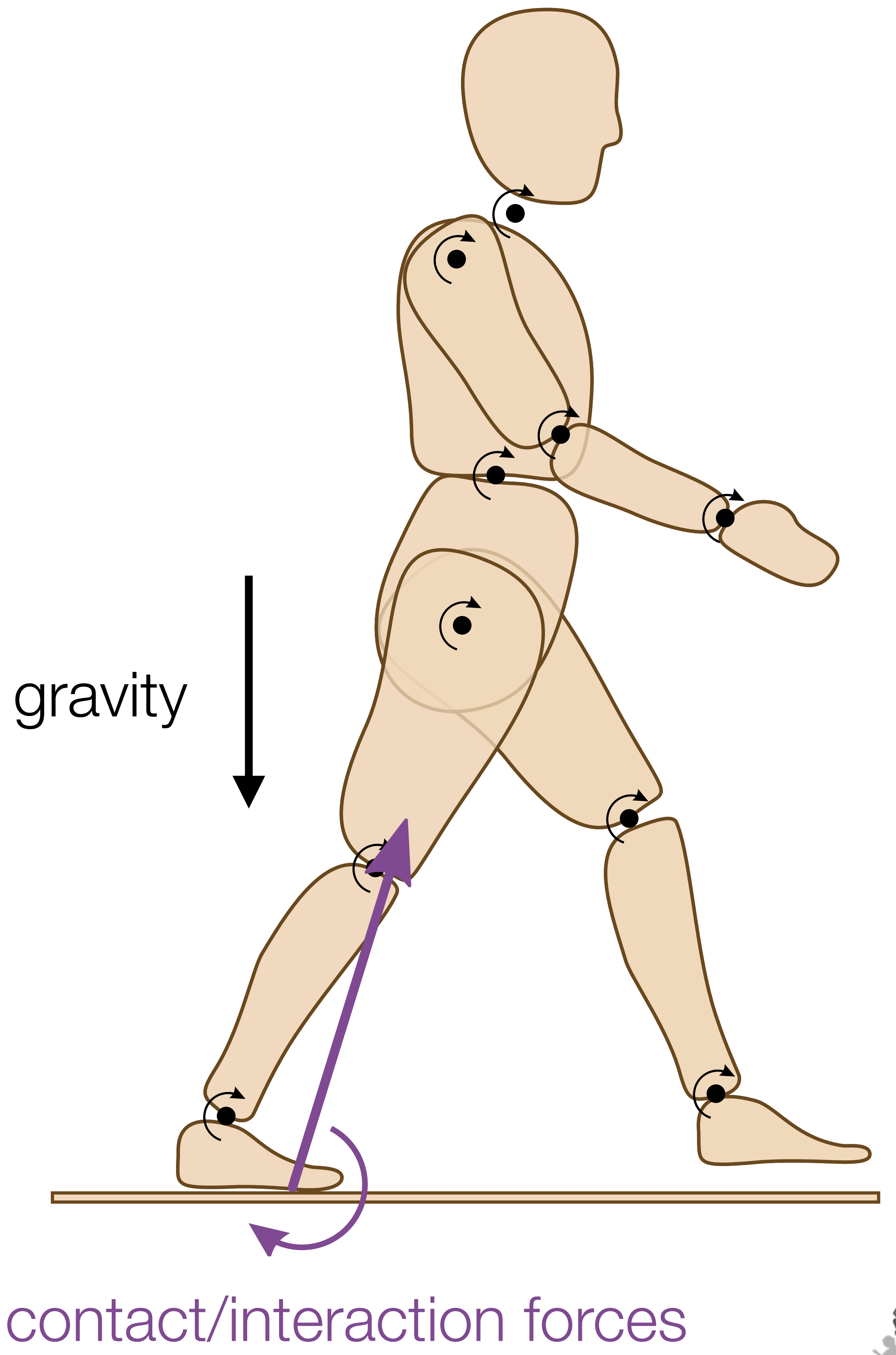
Gaol of this class

Understand the **various approaches** of the state of the art to compute λ_c in:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau + J_c^\top(q)\,\lambda_c$$

Soft contact
- spring-damper model

Rigid contact
- bilateral contact model
- unilateral contact model



Gaol of this class

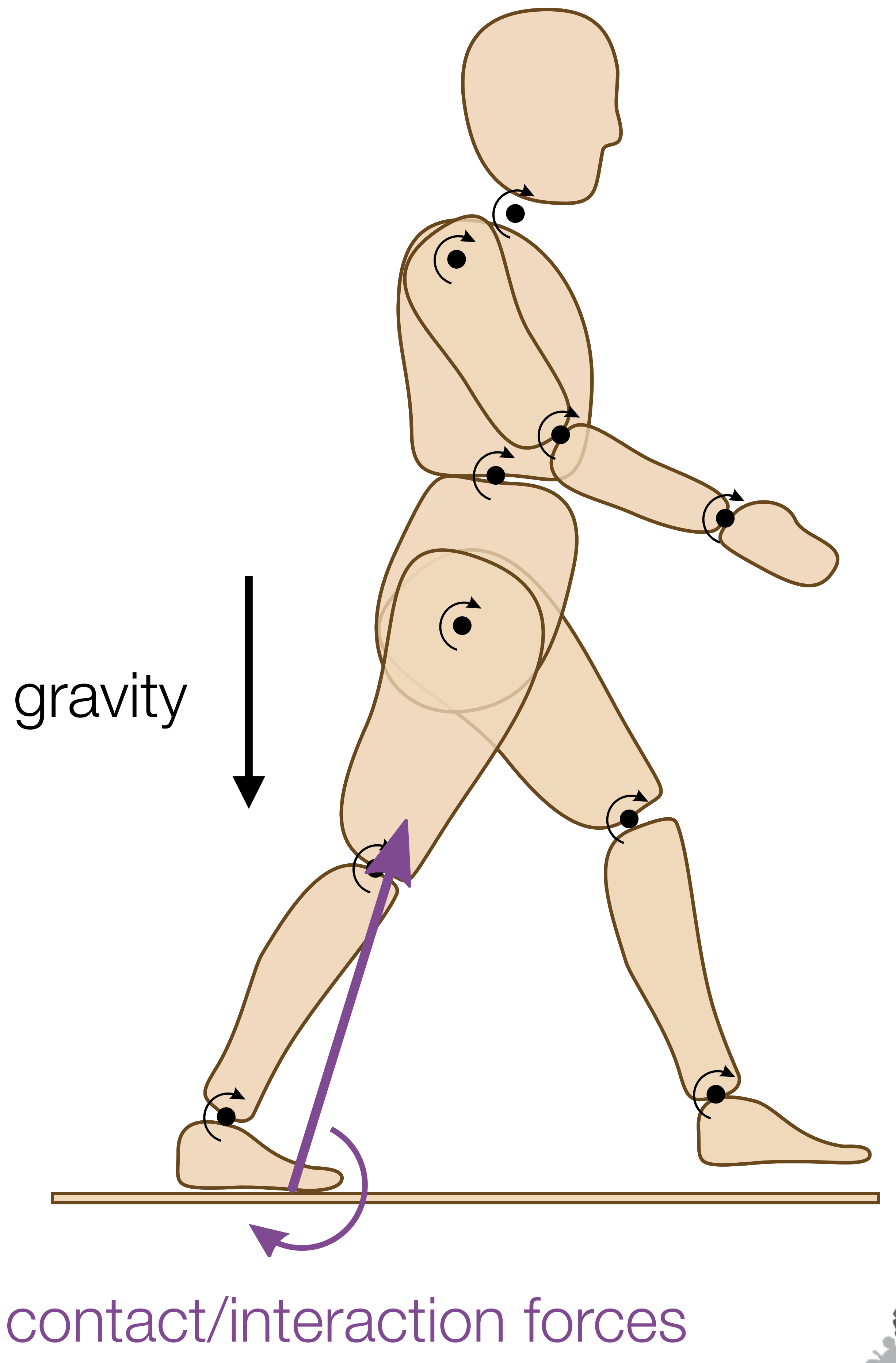
Understand the **various approaches** of the state of the art to compute λ_c in:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau + J_c^\top(q)\,\lambda_c$$

Soft contact
- spring-damper model

Rigid contact
- bilateral contact model
- unilateral contact model

Mixed contact
- the relaxed contact model



The Soft Contact Problem

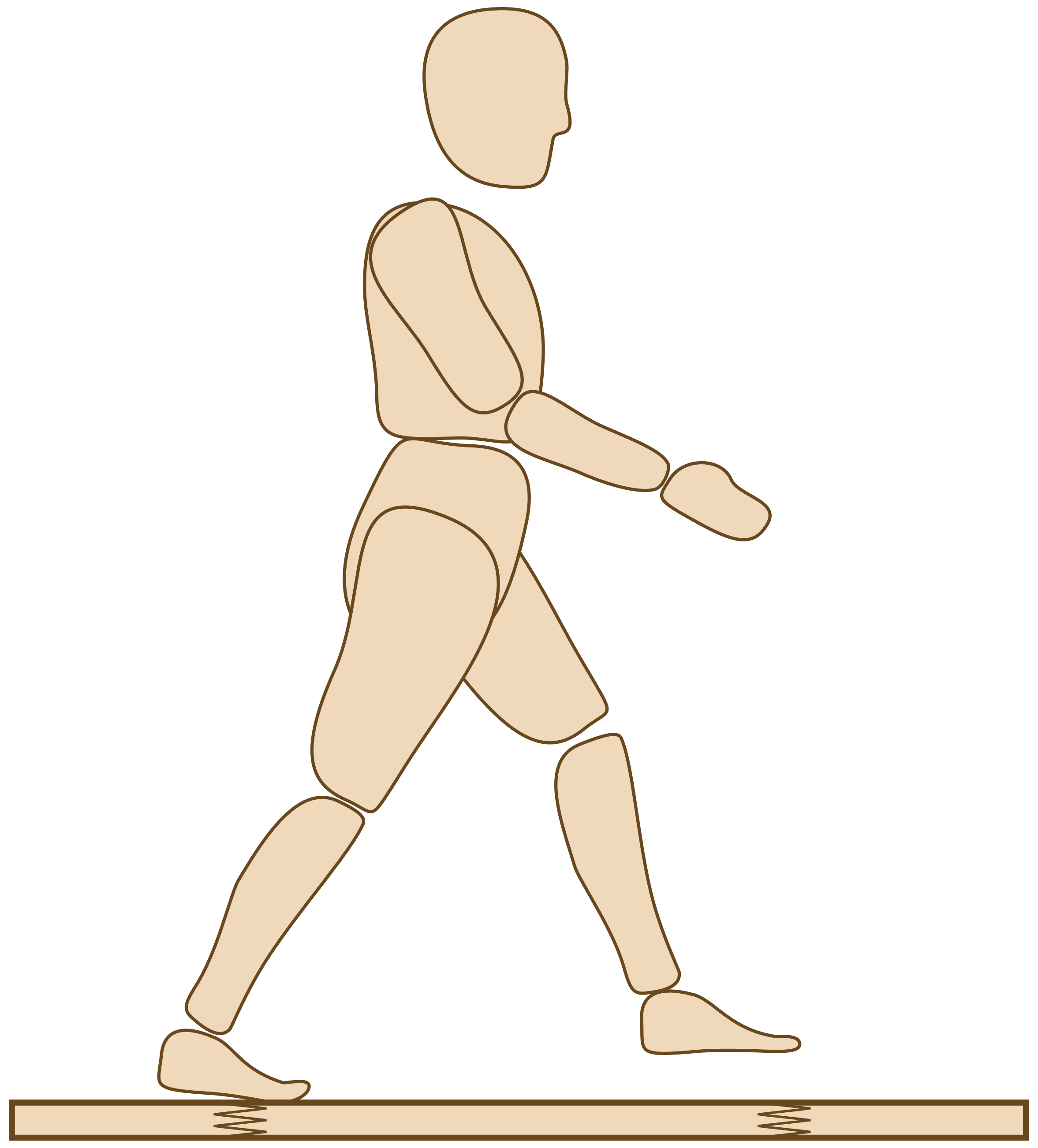

SALTO


SALTO

Soft contact: the spring-damper model

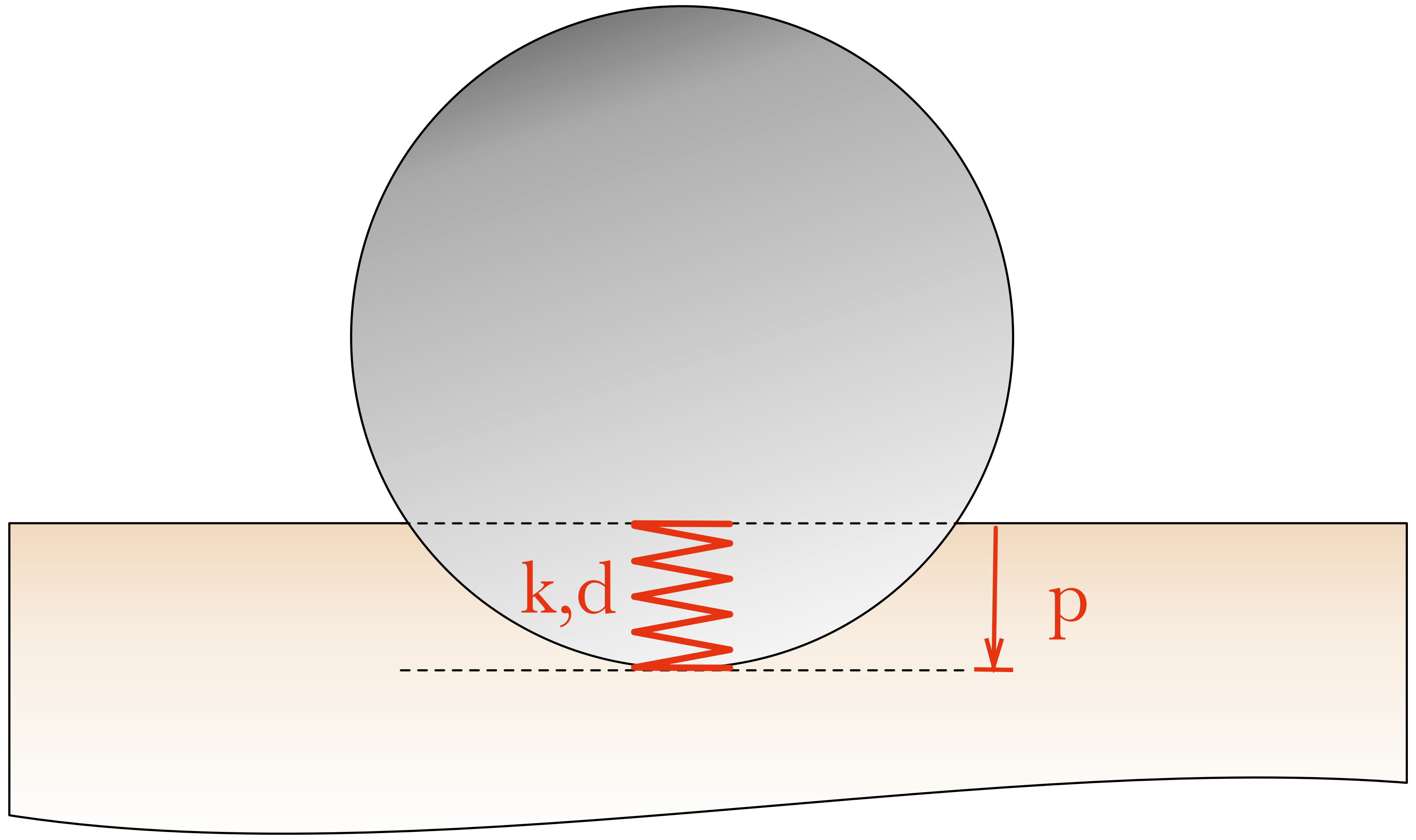
This is the **simplest** contact model, very **intuitive** and **straightforward** to implement

This contact model is defined by the spring k and the damper d quantities, reading:



$$\lambda_c^n = \max(-k\,.\,p - d\,.\,\dot{p}, 0)$$

the max function means:
the ground can ONLY push




z
x
y
Right
Left
General
1
40
80
120
160
200
240
280
320
360
400
440
480
520
565

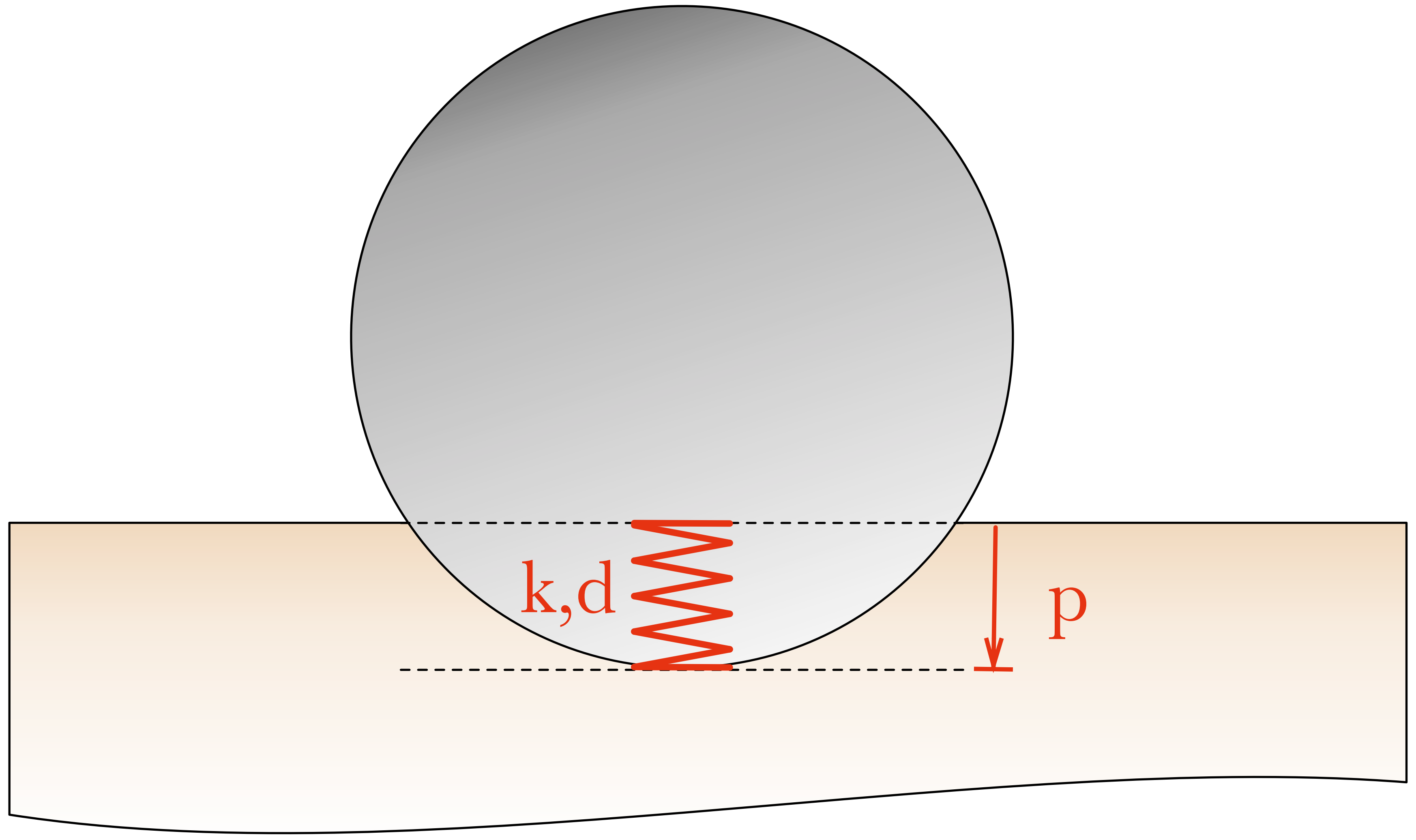

z
x
y
Right
Left
General
1
40
80
120
160
200
240
280
320
360
400
440
480
520
565

Soft contact: the spring-damper model

This is the **simplest** contact model, very **intuitive** and **straightforward** to implement

BUT

not **relevant** to model rigid interface ($k \to \infty$), requires **stable integrator** (stiff equation)



The Rigid Contact Problem

bilateral contacts

The Least-Action Principle

"Nature is thrifty in all its actions"

Pierre-Louis Maupertuis

This statement applies for many (almost all) physical problems, **from Mechanics to Relativity**

Pierre-Louis Maupertuis

The Least-Action Principle

"Nature is thrifty in all its actions"

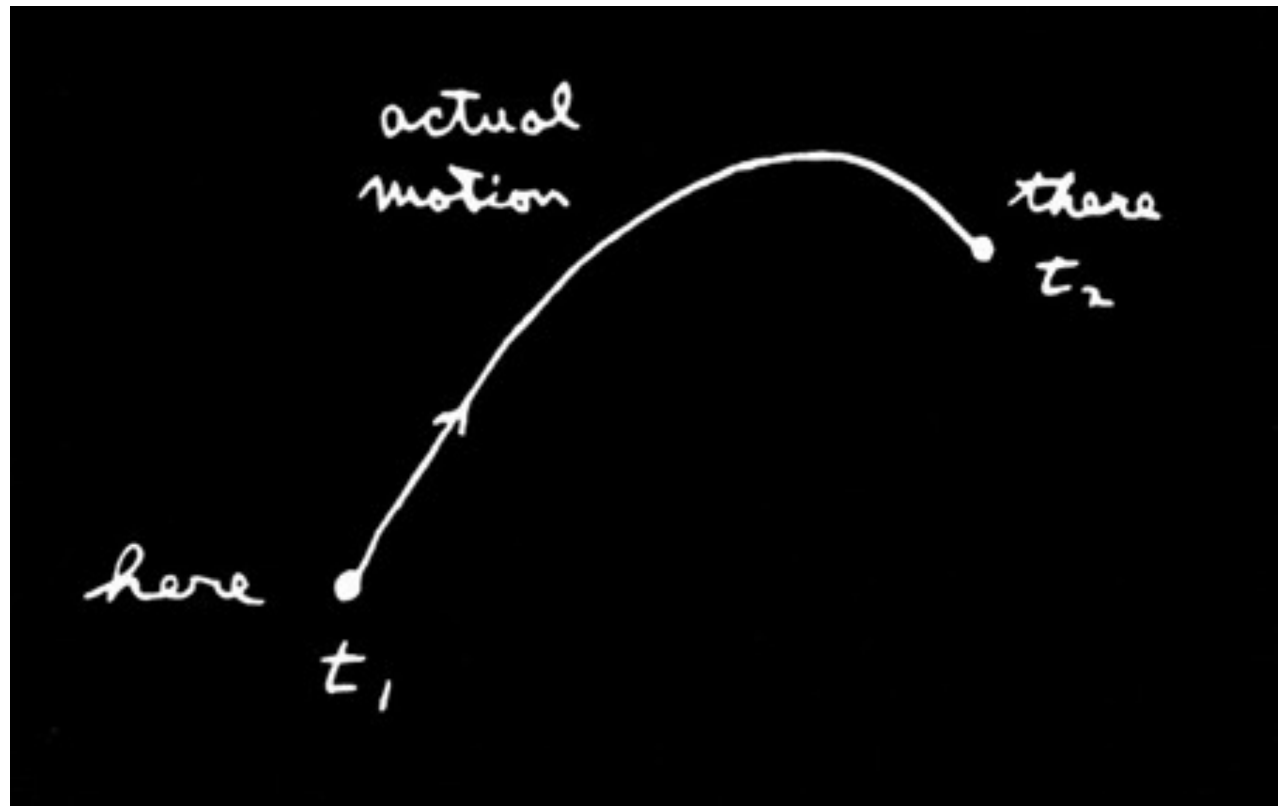
Pierre-Louis Maupertuis

Pierre-Louis Maupertuis

This statement applies for many (almost all) physical problems, **from Mechanics to Relativity**

In Mechanics, it corresponds to the minimization of the **action**, the integral of the **Kinetic - Potential energies** over time

$$S_1 = \int_{t1}^{t2} \frac{1}{2} m \left(\frac{dx}{dt}\right)^2 - mgx\, dt$$



The Least-Action Principle

Pierre-Louis Maupertuis

"Nature is thrifty in all its actions"

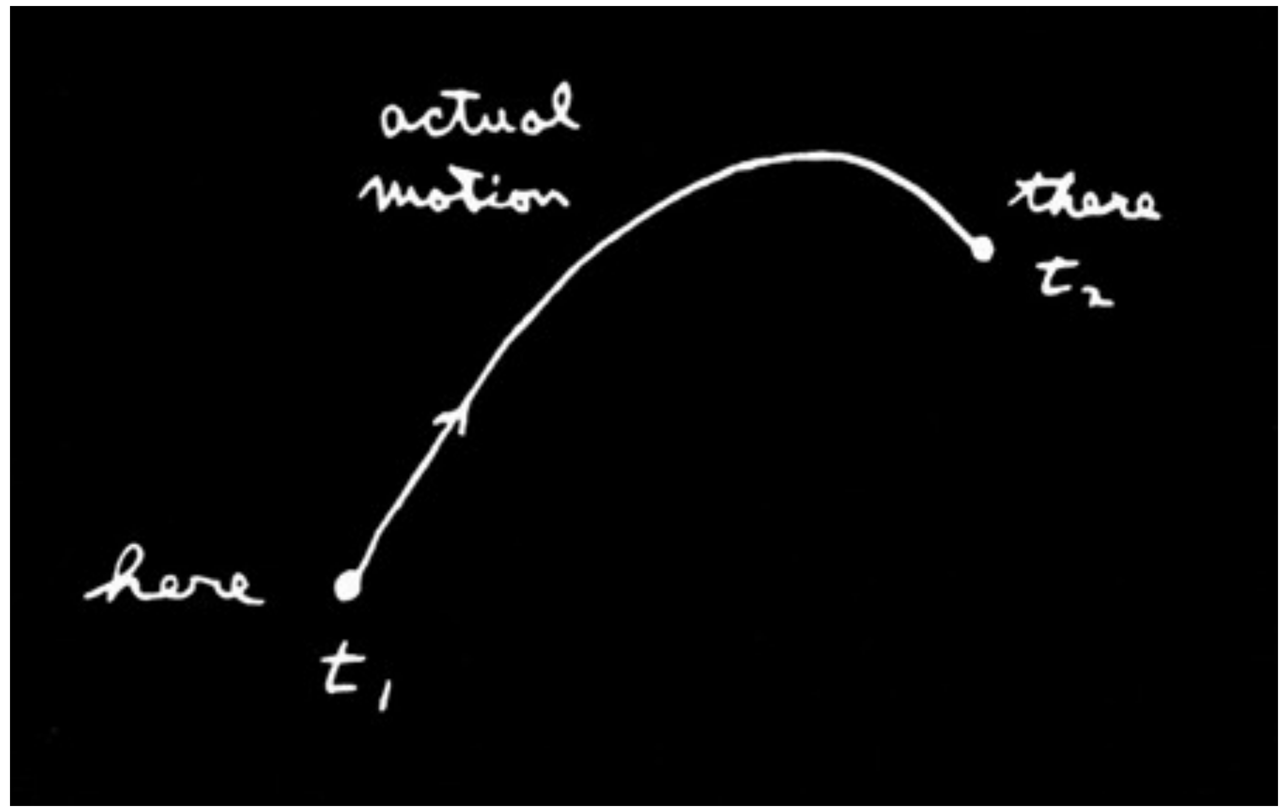
Pierre-Louis Maupertuis

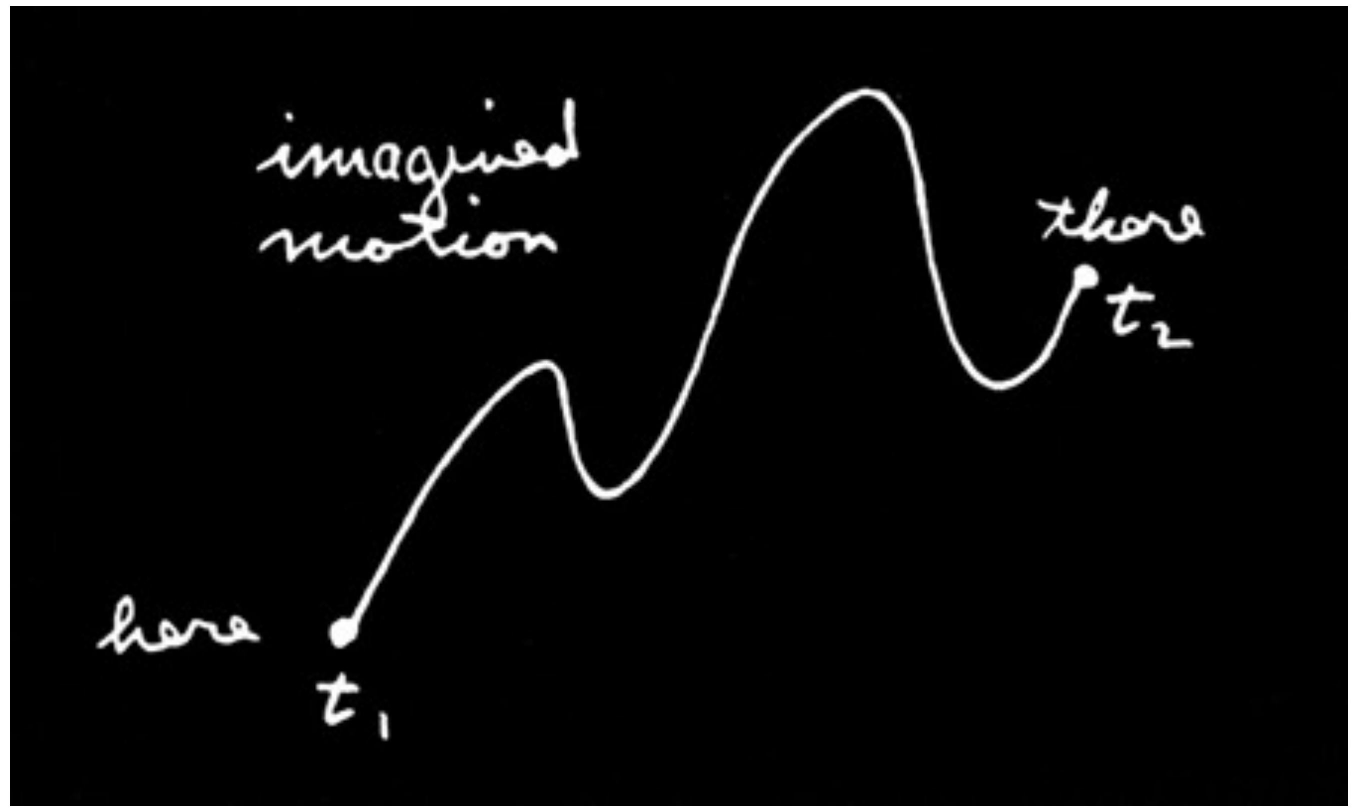
This statement applies for many (almost all) physical problems, **from Mechanics to Relativity**

In Mechanics, it corresponds to the minimization of the **action**, the integral of the **Kinetic - Potential energies** over time

$$S_1 = \int_{t1}^{t2} \frac{1}{2} m \left(\frac{dx}{dt} \right)^2 - mgx \, dt$$

$$S_2 = \int_{t1}^{t2} \frac{1}{2} m \left(\frac{dx}{dt} \right)^2 - mgx \, dt$$





The Least-Action Principle

"Nature is thrifty in all its actions"

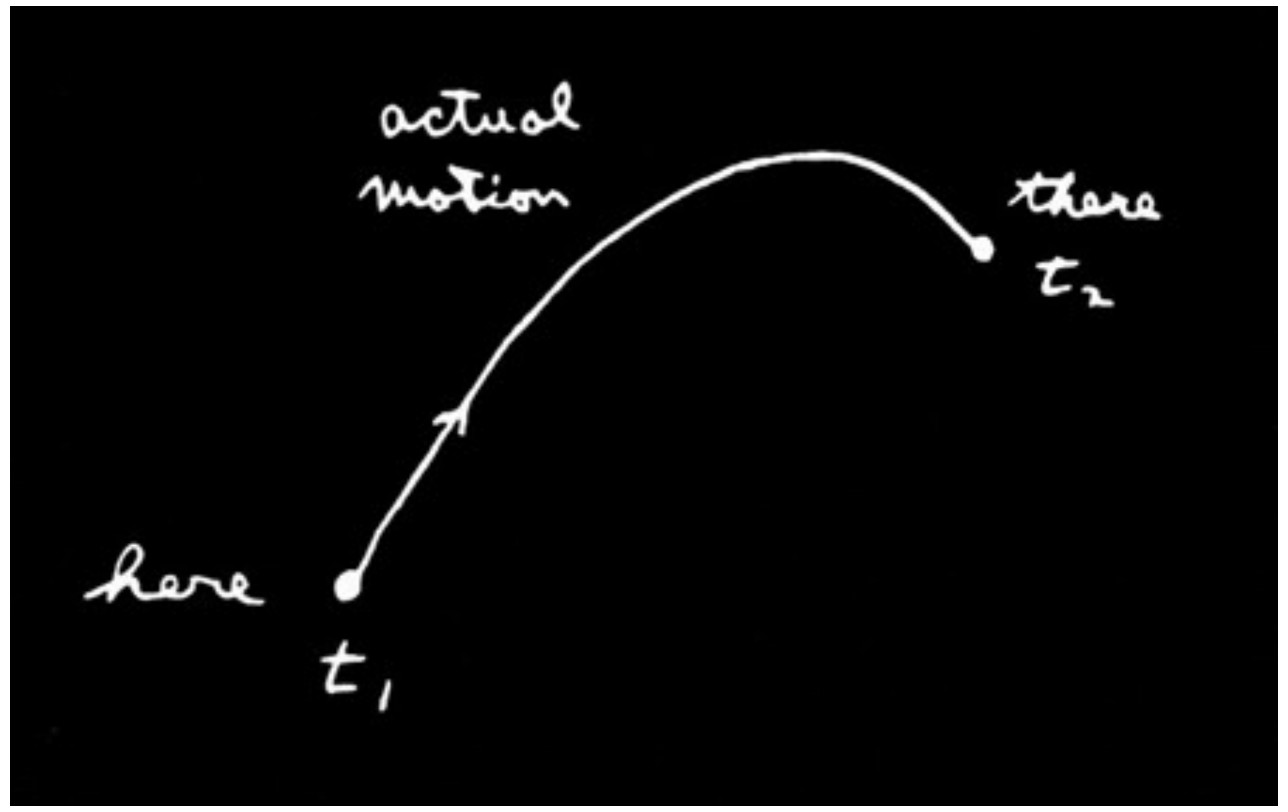
Pierre-Louis Maupertuis

Pierre-Louis Maupertuis

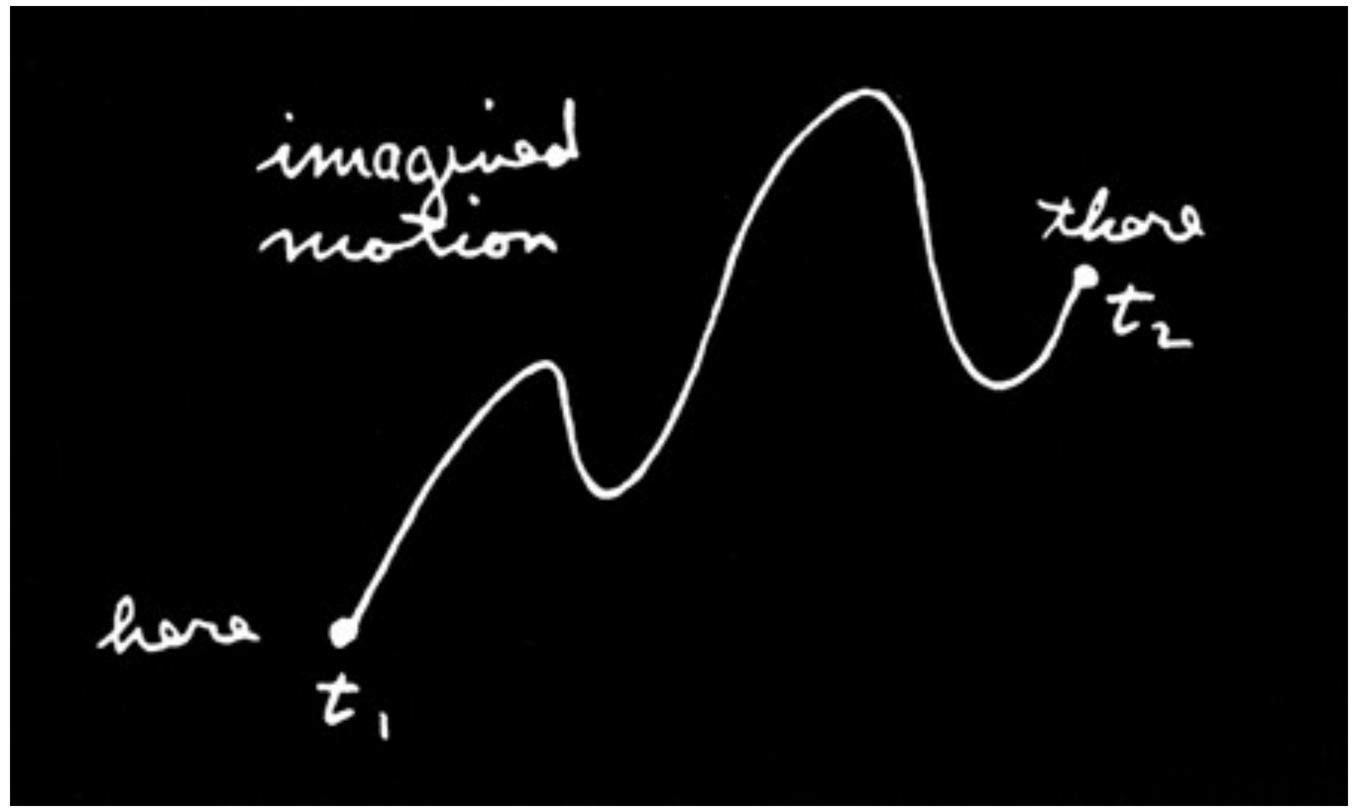
This statement applies for many (almost all) physical problems, **from Mechanics to Relativity**

In Mechanics, it corresponds to the minimization of the **action**, the integral of the **Kinetic - Potential energies** over time

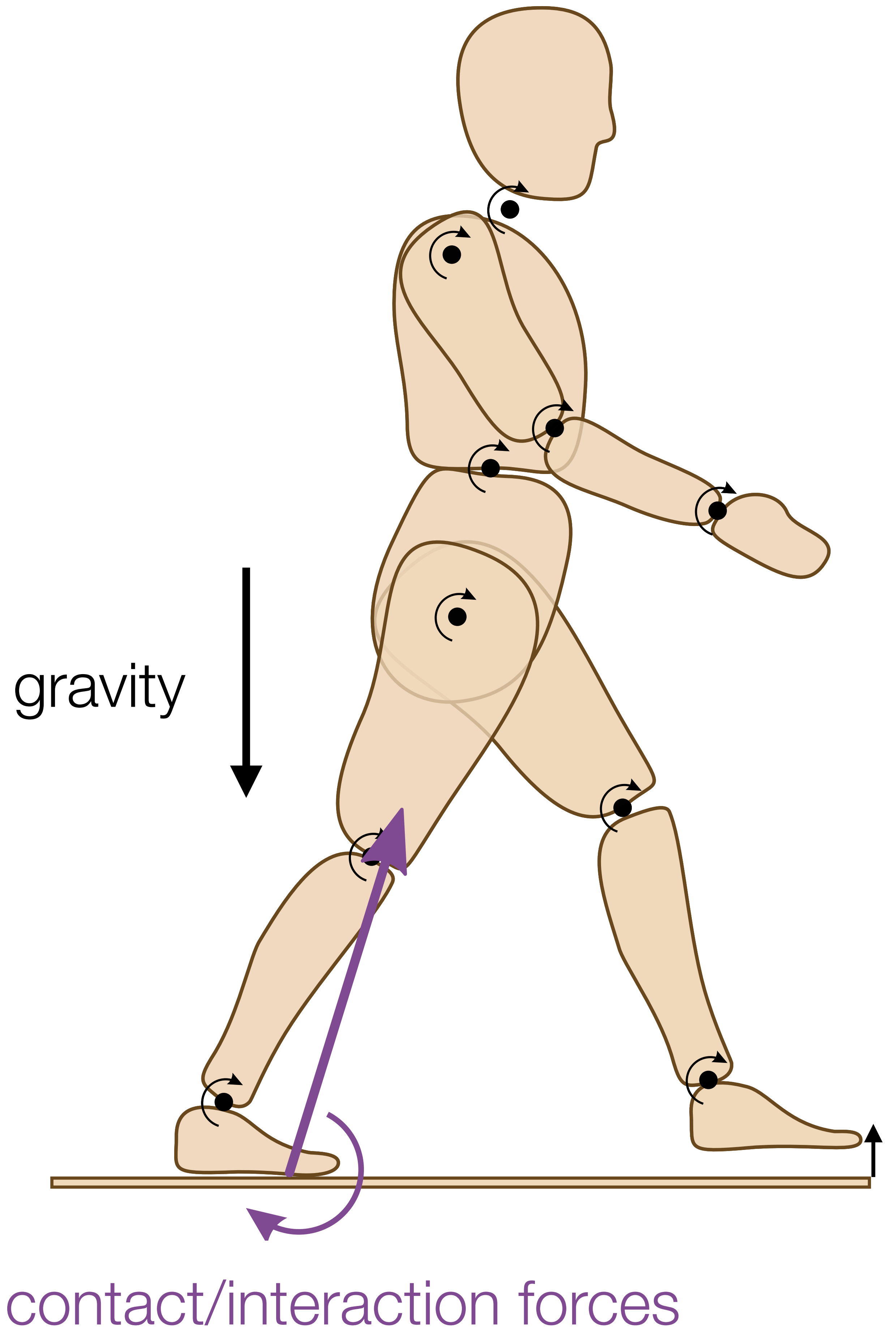
$$S_1 = \int_{t1}^{t2} \frac{1}{2}m\left(\frac{dx}{dt}\right)^2 - mgx\, dt \quad < \quad S_2 = \int_{t1}^{t2} \frac{1}{2}m\left(\frac{dx}{dt}\right)^2 - mgx\, dt$$



The solution is a stationary point, i.e $\delta S = 0$



The Least Action Principle as a classic QP



<u>Problem:</u> knowing q and $\dot{q}$, we aim at retrieving $\ddot{q}$ and λ_c

least distance w.r.t to the unconstrained acceleration

a metric induced by the kinetic energy

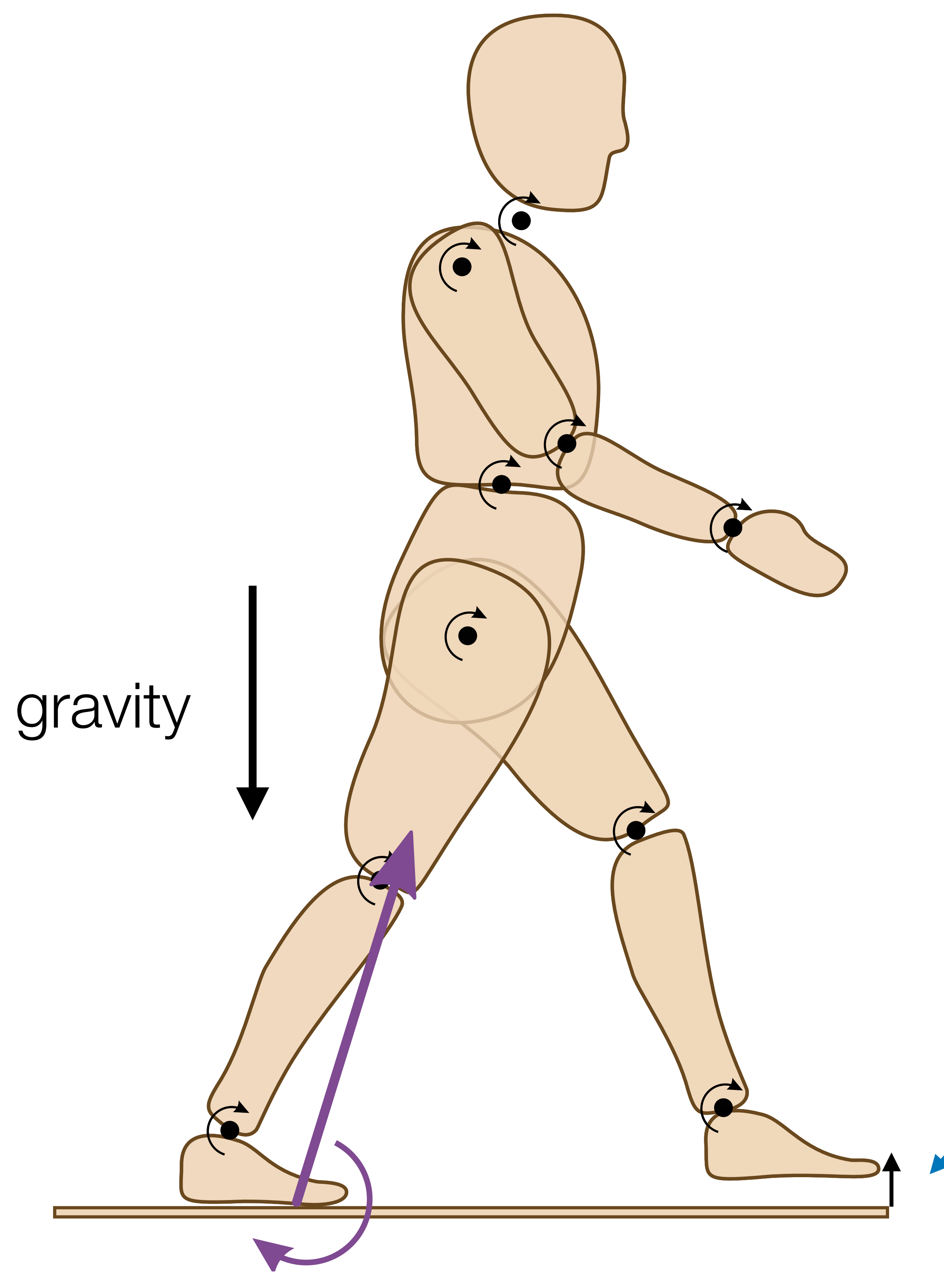
$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$

$$c(q) = 0$$

gap between floor and foot

where $\ddot{q}_f \stackrel{\text{def}}{=} M^{-1}(q)\big(\tau - C(q,\dot{q}) - G(q)\big)$ is the so-called **free acceleration** (without constraint)

The Least Action Principle as a classic QP



Problem: knowing q and $\dot{q}$, we aim at retrieving $\ddot{q}$ and λ_c

least distance w.r.t to the unconstrained acceleration

a metric induced by the kinetic energy

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$

$$c(q) = 0$$

gap between floor and foot

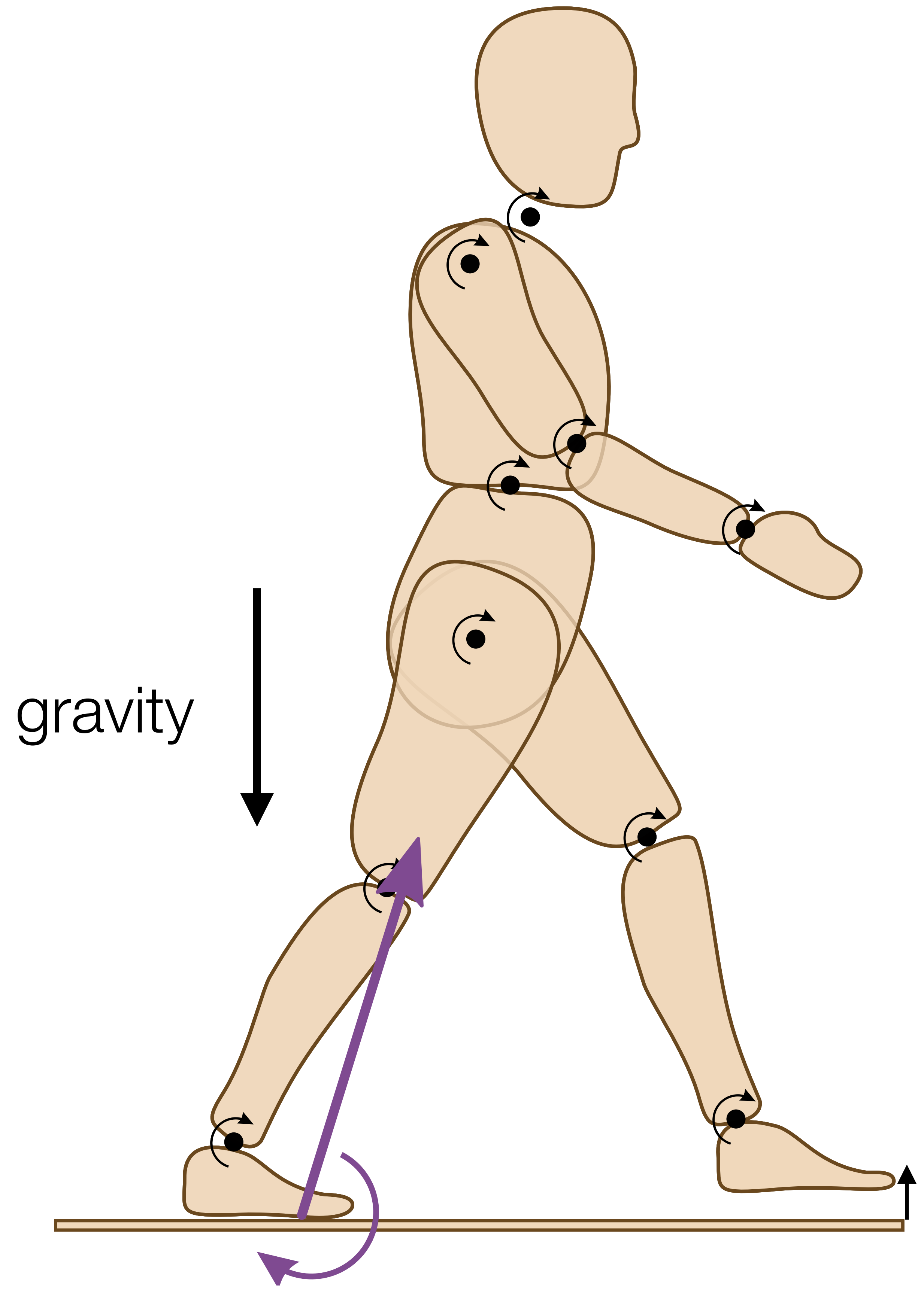
$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$

$$c(q) = 0$$

where $\ddot{q}_f \overset{\text{def}}{=} M^{-1}(q)\big(\tau - C(q, \dot{q}) - G(q)\big)$ is the so-called **free acceleration** (without constraint)

The Least Action Principle as a classic QP

<u>Problem:</u> knowing q and $\dot{q}$, we aim at retrieving $\ddot{q}$ and λ_c



least distance w.r.t to the unconstrained acceleration

a metric induced by the kinetic energy

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$

$$c(q) = 0$$

gap between floor and foot

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$

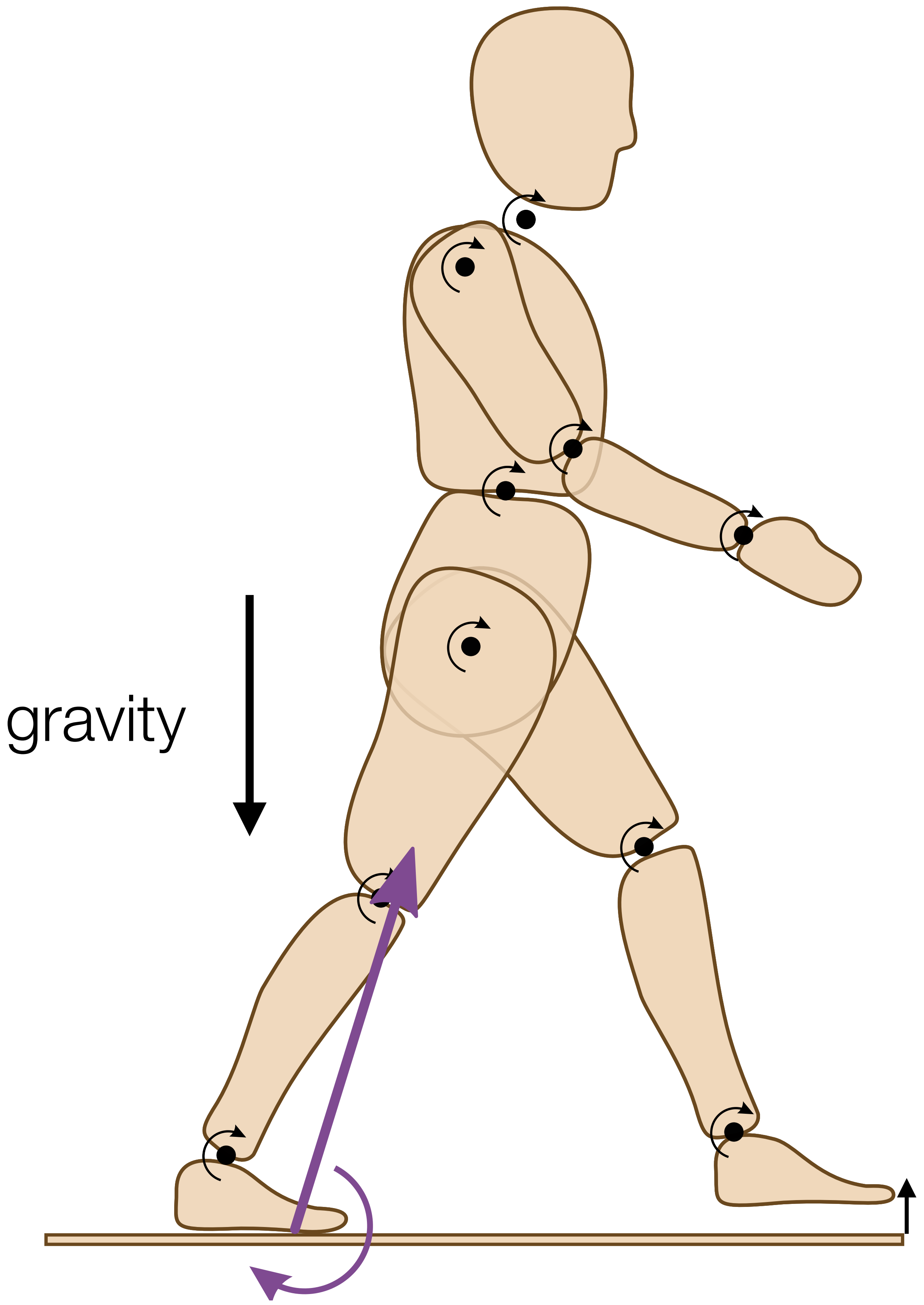
$$c(q) = 0$$

index reduction = time derivation

$$J_c(q)\,\dot{q} = 0$$

where $\ddot{q}_f \overset{\text{def}}{=} M^{-1}(q)\big(\tau - C(q,\dot{q}) - G(q)\big)$ is the so-called **free acceleration** (without constraint)

The Least Action Principle as a classic QP



<u>Problem:</u> knowing q and $\dot{q}$, we aim at retrieving $\ddot{q}$ and λ_c

least distance w.r.t to the unconstrained acceleration

a metric induced by the kinetic energy

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$

$$c(q) = 0$$

gap between floor and foot

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$

$$c(q) = 0$$

index reduction = time derivation

$$J_c(q)\,\dot{q} = 0$$

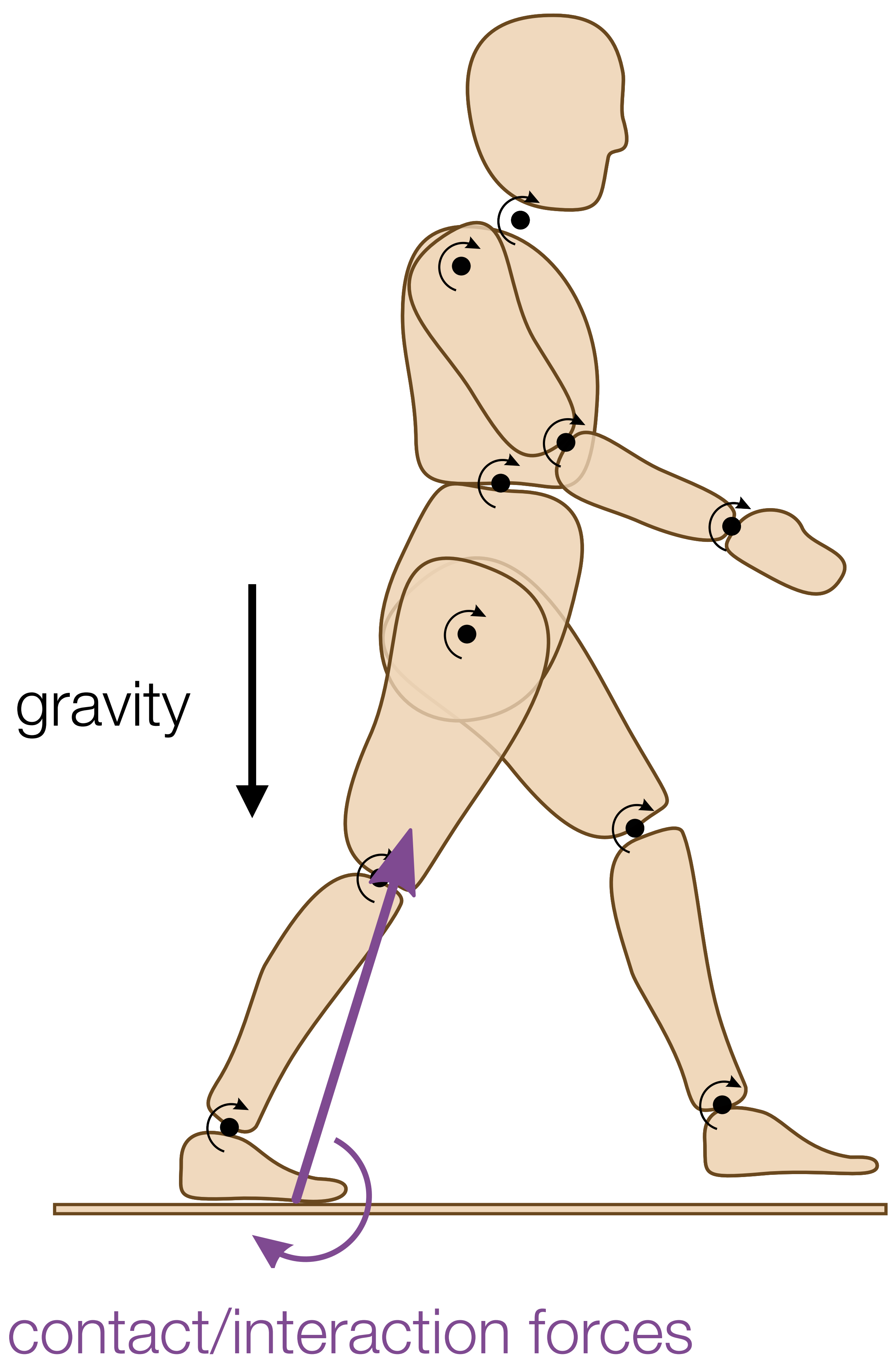
index reduction

$$J_c(q)\,\ddot{q} + \underbrace{\dot{J}_c(q,\dot{q})\dot{q}}_{\gamma_c(q,\dot{q})} = 0$$

the constraint differentiated twice w.r.t. time

where $\ddot{q}_f \stackrel{\text{def}}{=} M^{-1}(q)\big(\tau - C(q,\dot{q}) - G(q)\big)$ is the so-called **free acceleration** (without constraint)

The Least Action Principle as a classic QP

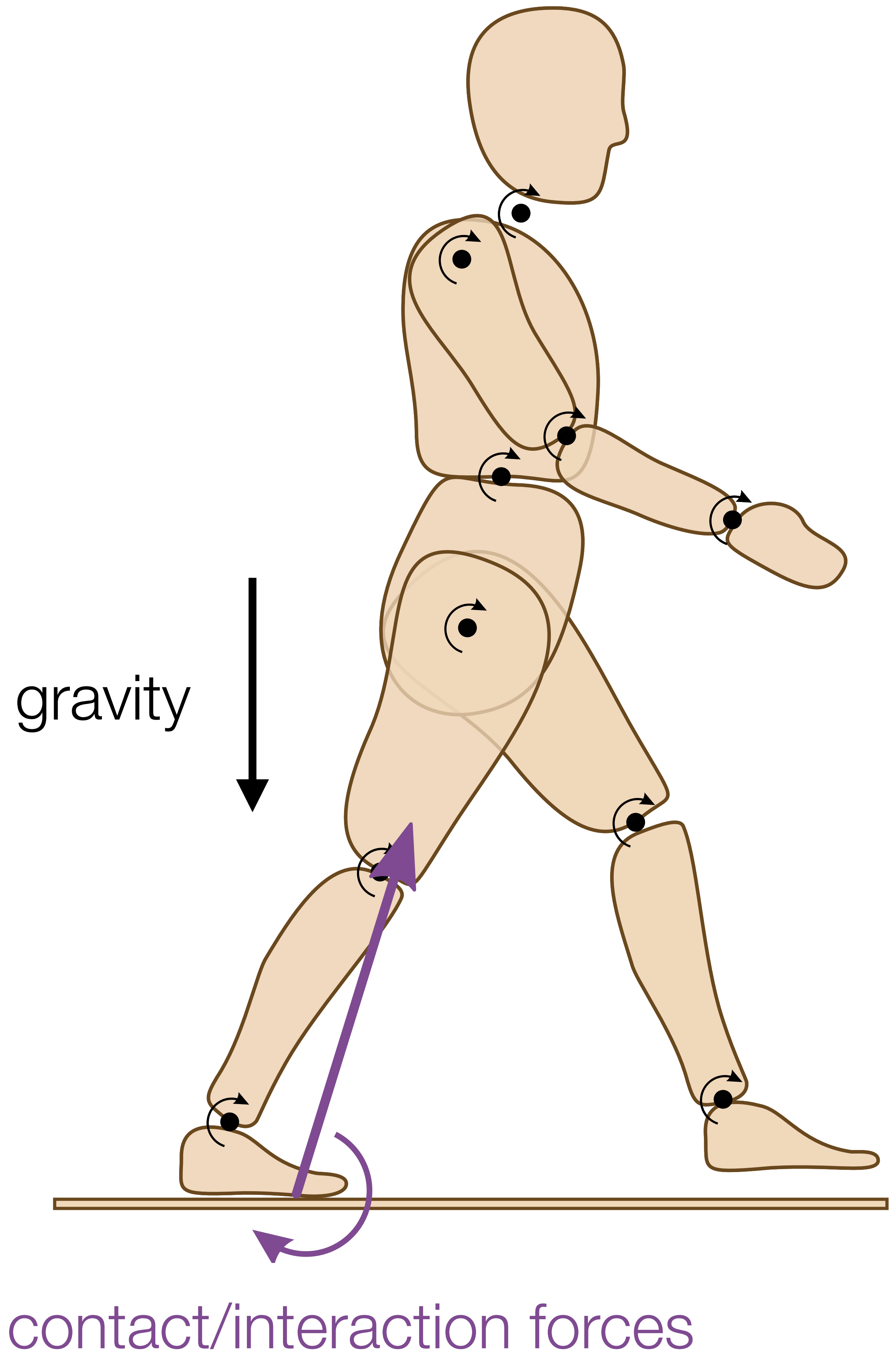


Problem: we have now formed an equality-constrained QP.

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$
$$J_c(q)\,\ddot{q} + \gamma_c(q,\dot{q}) = 0$$

How to solve it? Where do the contact forces lie?

The Least Action Principle as a classic QP



Problem: we have now formed an equality-constrained QP.

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$
$$J_c(q)\,\ddot{q} + \gamma_c(q,\dot{q}) = 0$$

How to solve it? Where do the contact forces lie?

The solution can be retrieved by **deriving the KKT conditions** of the QP problem via the so-called **Lagrangian:**

dual variable = contact forces

$$L(\ddot{q}, \lambda_c) = \underbrace{\frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}}_{\text{cost function}} - \lambda_c^\top \underbrace{\left(J_c(q)\ddot{q} + \gamma_c(q,\dot{q})\right)}_{\text{equality constraint}}$$

Solving the Lagrangian contact problem

dual variable = contact forces

$$L(\ddot{q}, \lambda_c) = \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)} - \lambda_c^\top \left(J_c(q)\ddot{q} + \gamma_c(q, \dot{q})\right)$$

cost function

equality constraint

Solving the Lagrangian contact problem

dual variable = contact forces

$$L(\ddot{q}, \lambda_c) = \underbrace{\frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}}_{\text{cost function}} - \lambda_c^\top \underbrace{\left(J_c(q)\ddot{q} + \gamma_c(q, \dot{q})\right)}_{\text{equality constraint}}$$

The **KKT conditions** of the QP problem are given by:

$$\nabla_{\ddot{q}} L = M(q)(\ddot{q} - \ddot{q}_f) - J_c(q)^\top \lambda_c = 0$$

Joint space force propagation

$$\nabla_{\lambda_c} L = J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = 0$$

Contact acceleration constraint

Solving the Lagrangian contact problem

dual variable = contact forces

$$L(\ddot{q}, \lambda_c) = \underbrace{\frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}}_{\text{cost function}} - \lambda_c^\top \underbrace{\left(J_c(q)\ddot{q} + \gamma_c(q, \dot{q})\right)}_{\text{equality constraint}}$$

The **KKT conditions** of the QP problem are given by:

$$\nabla_{\ddot{q}} L = M(q)(\ddot{q} - \ddot{q}_f) - J_c(q)^\top \lambda_c = 0 \quad \text{Joint space force propagation}$$

$$\nabla_{\lambda_c} L = J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = 0 \quad \text{Contact acceleration constraint}$$

rearranging a bit the terms, leads to:

$$M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f$$

$$J_c(q)\ddot{q} + 0 = -\gamma_c(q, \dot{q})$$

Solving the Lagrangian contact problem

dual variable = contact forces

$$L(\ddot{q}, \lambda_c) = \underbrace{\frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}}_{\text{cost function}} - \lambda_c^\top \underbrace{\left(J_c(q)\ddot{q} + \gamma_c(q, \dot{q})\right)}_{\text{equality constraint}}$$

The **KKT conditions** of the QP problem are given by:

$$\nabla_{\ddot{q}} L = M(q)(\ddot{q} - \ddot{q}_f) - J_c(q)^\top \lambda_c = 0 \quad \text{Joint space force propagation}$$

$$\nabla_{\lambda_c} L = J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = 0 \quad \text{Contact acceleration constraint}$$

rearranging a bit the terms, leads to:

$$M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f$$
$$J_c(q)\ddot{q} + 0 = -\gamma_c(q, \dot{q})$$

leading to the so-called **KKT dynamics**:

$$\underbrace{\begin{bmatrix} M(q) & J_c^\top(q) \\ J_c(q) & 0 \end{bmatrix}}_{K(q)} \begin{bmatrix} \ddot{q} \\ -\lambda_c \end{bmatrix} = \begin{bmatrix} M(q)\ddot{q}_f \\ -\gamma_c(q, \dot{q}) \end{bmatrix}$$

Solving the Lagrangian contact problem

dual variable = contact forces

$$L(\ddot{q}, \lambda_c) = \underbrace{\frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}}_{\text{cost function}} - \lambda_c^\top \underbrace{\left(J_c(q)\ddot{q} + \gamma_c(q, \dot{q})\right)}_{\text{equality constraint}}$$

The **KKT conditions** of the QP problem are given by:

$$\nabla_{\ddot{q}} L = M(q)(\ddot{q} - \ddot{q}_f) - J_c(q)^\top \lambda_c = 0$$ Joint space force propagation

$$\nabla_{\lambda_c} L = J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = 0$$ Contact acceleration constraint

rearranging a bit the terms, leads to:

$$M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f$$
$$J_c(q)\ddot{q} + 0 = -\gamma_c(q, \dot{q})$$

leading to the so-called **KKT dynamics**:

$$\underbrace{\begin{bmatrix} M(q) & J_c^\top(q) \\ J_c(q) & 0 \end{bmatrix}}_{K(q)} \begin{bmatrix} \ddot{q} \\ -\lambda_c \end{bmatrix} = \begin{bmatrix} M(q)\ddot{q}_f \\ -\gamma_c(q, \dot{q}) \end{bmatrix}$$

BUT, there might be one, redundant solutions or no solution at all:

wether (i) $J_c(q)$ is **full rank** (ii) $J_c(q)$ is **not full rank** or (ii) $\gamma_c(q, \dot{q})$ is **not in the range space** of $J_c(q)$

Explicit contact solution

We can analytically inverse the system to obtain the solution in **3 main steps**:

$$M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f$$

$$J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = 0$$

Explicit contact solution

1 - Express $\ddot{q}$ as function of $\ddot{q}_f$ and λ_c

$$\ddot{q} = \ddot{q}_f + M^{-1}(q)J_c(q)^\top \lambda_c$$

We can analytically inverse the system to obtain the solution in **3 main steps**:

$$M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f$$

$$J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = 0$$

Explicit contact solution

We can analytically inverse the system to obtain the solution in **3 main steps**:

$$M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f$$

$$J_c(q)\ddot{q} + \gamma_c(q,\dot{q}) = 0$$

1 - Express $\ddot{q}$ as function of $\ddot{q}_f$ and λ_c

$$\ddot{q} = \ddot{q}_f + M^{-1}(q)J_c(q)^\top \lambda_c$$

2 - Replace $\ddot{q}$ and get an expression depending only on λ_c

$$\underbrace{J_c(q)M^{-1}(q)J_c(q)^\top}_{G_c(q)} \lambda_c + \underbrace{J_c(q)\ddot{q}_f + \gamma_c(q,\dot{q})}_{a_{c,f}(q,\dot{q},\ddot{q}_f)} = 0$$

$G_c(q)$: Delassus' matrix / Inverse Operational Space Inertia Matrix

$a_{c,f}(q,\dot{q},\ddot{q}_f)$: Free contact acceleration

Explicit contact solution

We can analytically inverse the system to obtain the solution in **3 main steps**:

$$M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f$$

$$J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = 0$$

1 - Express $\ddot{q}$ as function of $\ddot{q}_f$ and λ_c

$$\ddot{q} = \ddot{q}_f + M^{-1}(q)J_c(q)^\top \lambda_c$$

2 - Replace $\ddot{q}$ and get an expression depending only on λ_c

$$\underbrace{J_c(q)M^{-1}(q)J_c(q)^\top}_{G_c(q)} \lambda_c + \underbrace{J_c(q)\ddot{q}_f + \gamma_c(q, \dot{q})}_{a_{c,f}(q,\dot{q},\ddot{q}_f)} = 0$$

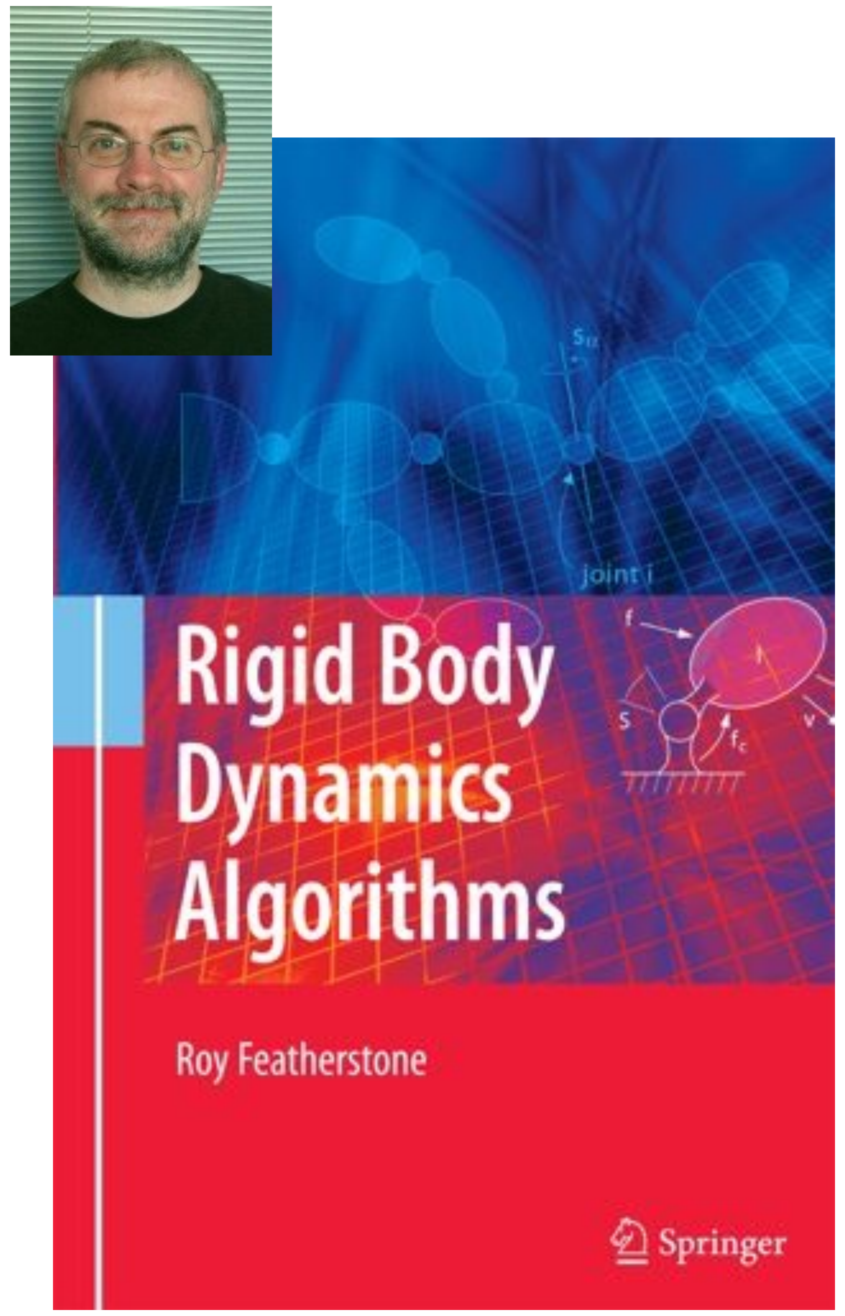
$G_c(q)$: Delassus' matrix
Inverse Operational Space Inertia Matrix

$a_{c,f}(q,\dot{q},\ddot{q}_f)$: Free contact acceleration

3 - Inverse $G(q)$ and find the optimal λ_c

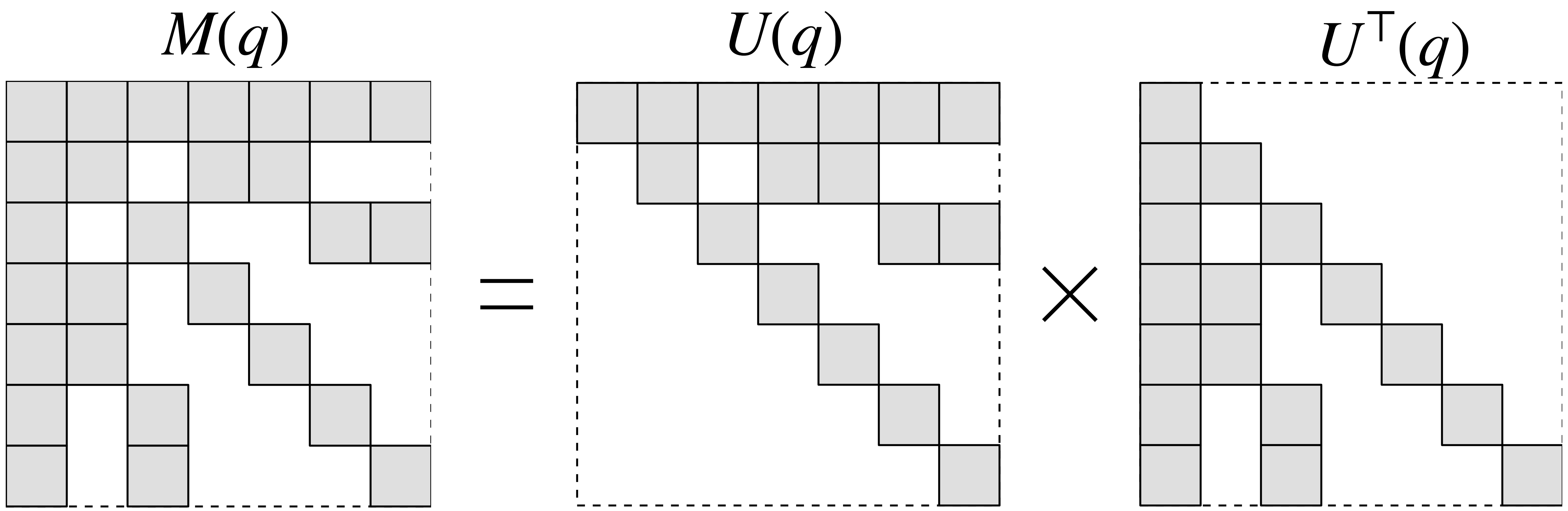
$$\lambda_c = -G_c^{-1}(q)\, a_{c,f}(q, \dot{q}, \ddot{q}_f)$$

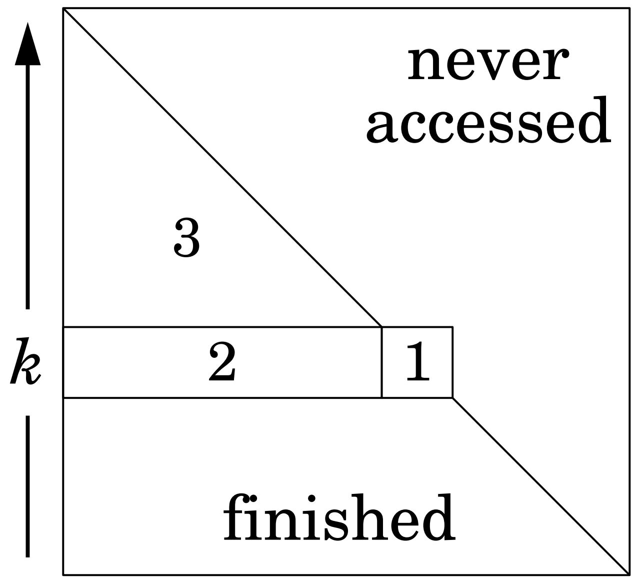
Mass Matrix: sparse Cholesky factorization



Goal: compute $G_c(q) \stackrel{\text{def}}{=} J_c(q)M^{-1}(q)J_c^\top(q)$ without computing $M^{-1}(q)$

Solution: exploiting **the sparsity** in the Cholesky factorization of $M(q)$



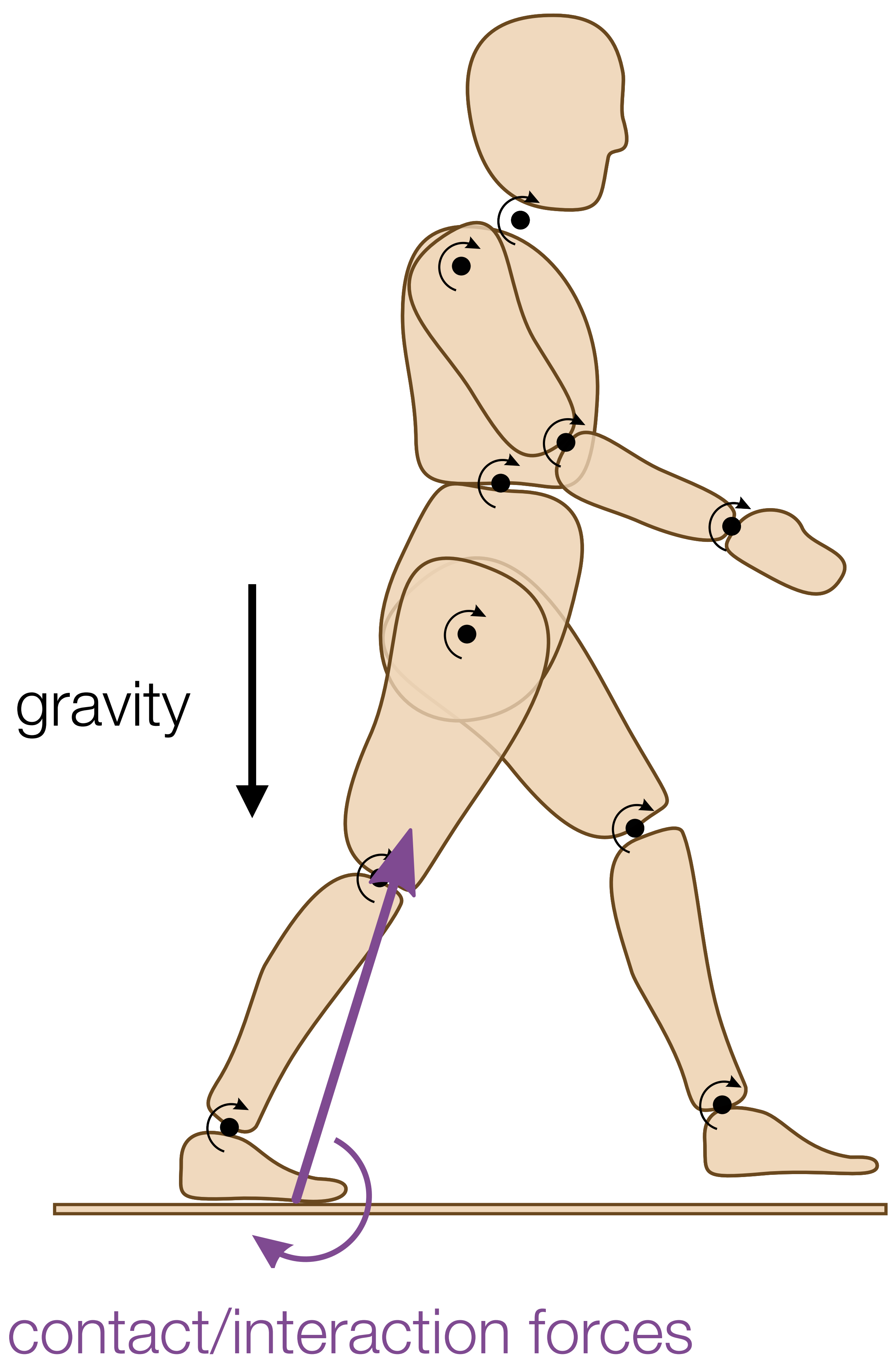


Cholesky factorization

1. $U_{k,k} = \sqrt{M_{k,k}}$
2. $U_{k,i} = M_{k,i}/U_{k,k}$
3. $U_{i,j} = M_{i,j} - U_{k,i}\,U_{k,j}$

The total complexity is $O(N^2)$ instead of $O(N^3)$ when using a **dense** Cholesky decomposition

The Maximum Dissipation Principle



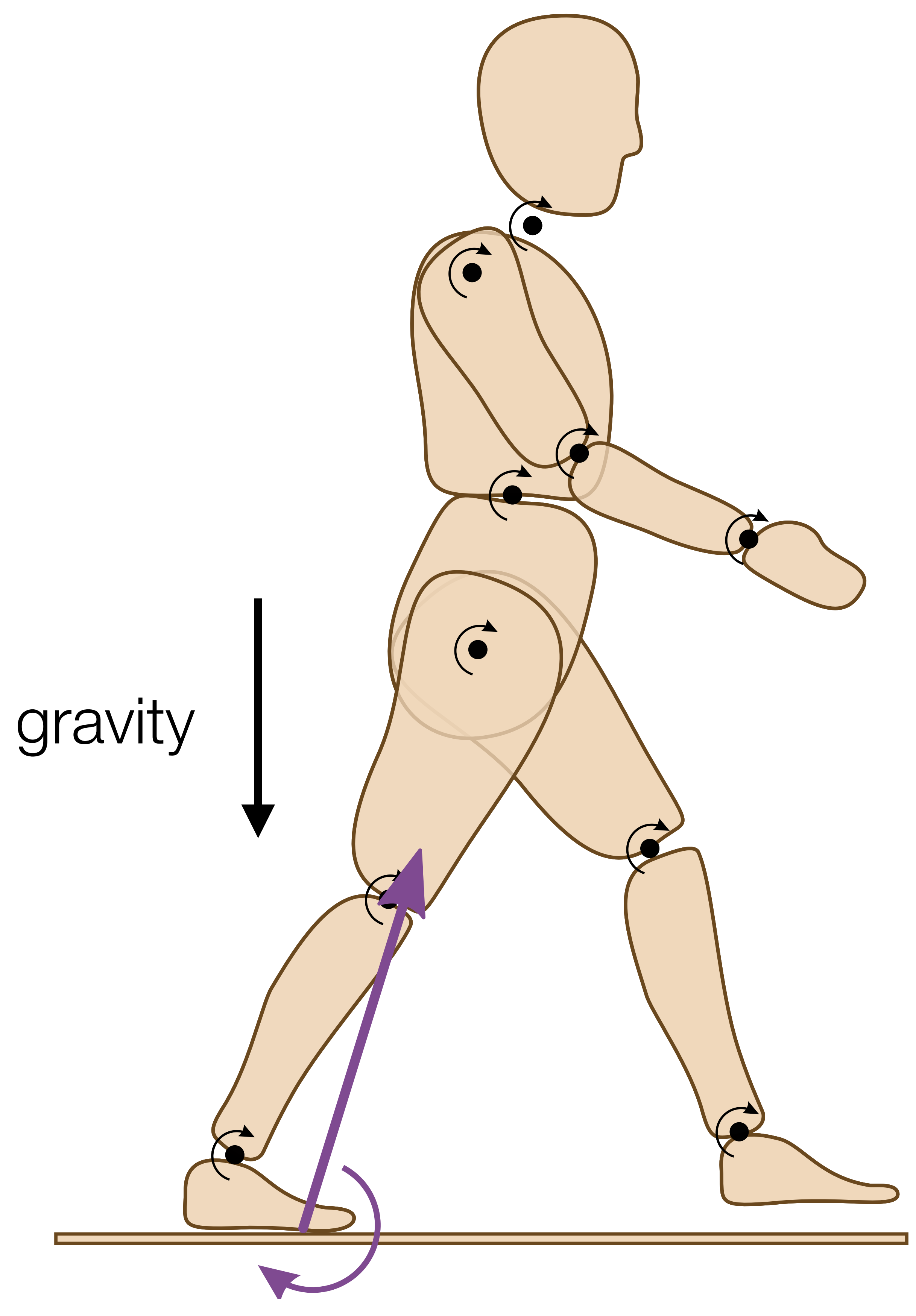
The contact forces λ_c fulfill the relation:

$$G_c(q)\lambda_c + a_{c,f}(q, \dot{q}, \ddot{q}_f) = 0$$

From an **energetic point of view**, this solution minimizes:

$$\min_{\lambda_c} \; \frac{1}{2}\lambda_c^\top G_c(q)\lambda_c + \lambda_c^\top a_{c,f}(q, \dot{q}, \ddot{q}_f)$$

The Maximum Dissipation Principle



The contact forces λ_c fulfill the relation:

$$G_c(q)\lambda_c + a_{c,f}(q, \dot{q}, \ddot{q}_f) = 0$$

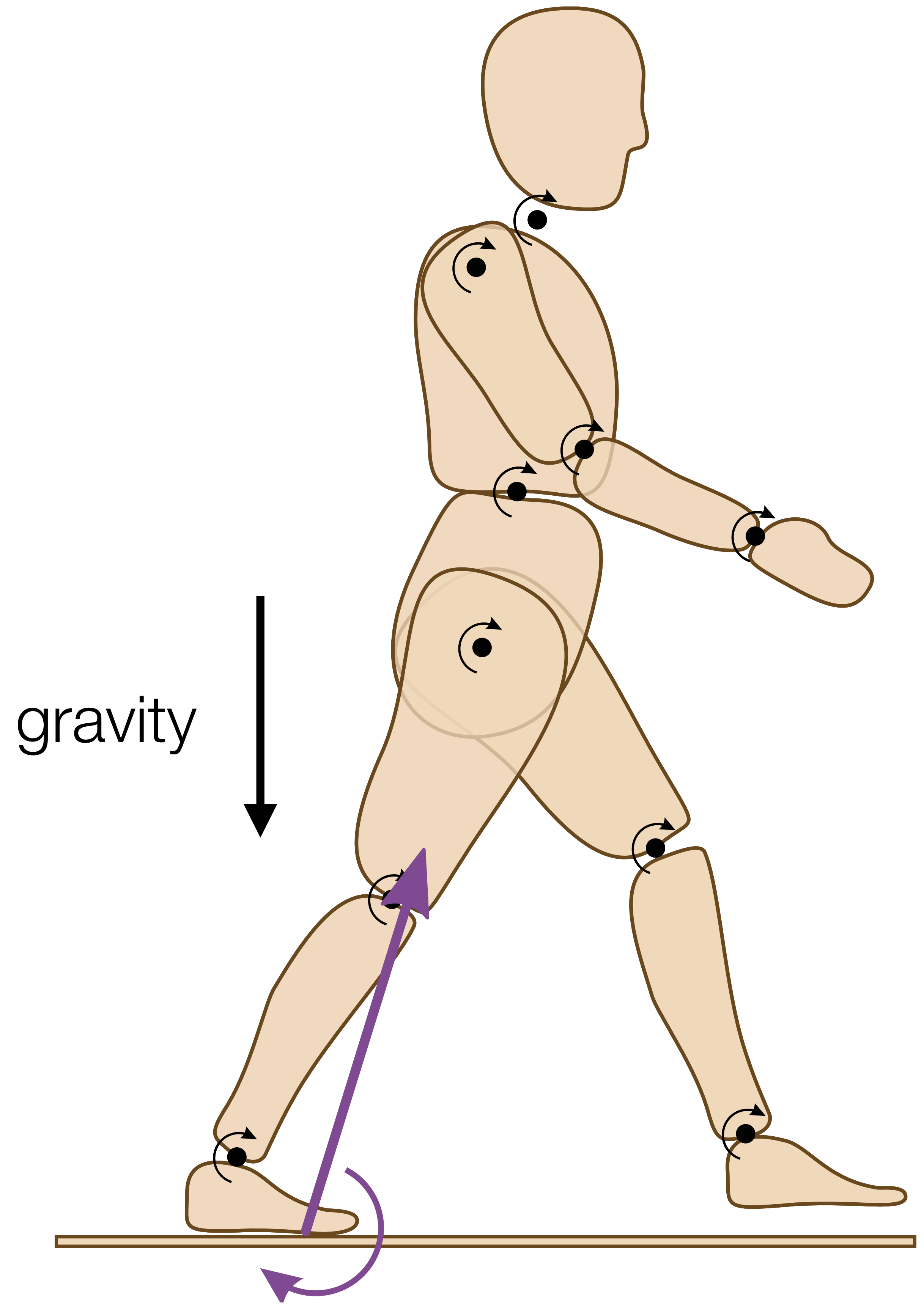
From an **energetic point of view**, this solution minimizes:

$$\min_{\lambda_c} \; \frac{1}{2}\lambda_c^\top G_c(q)\lambda_c + \lambda_c^\top a_{c,f}(q, \dot{q}, \ddot{q}_f)$$

or using a $\max$:

$$\max_{\lambda_c} \; -\frac{1}{2}\lambda_c^\top \underbrace{(G_c(q)\lambda_c + 2\lambda_c^\top a_{c,f}(q, \dot{q}, \ddot{q}_f))}_{a_c(q,\dot{q},\ddot{q})}$$

The Maximum Dissipation Principle



The contact forces λ_c fulfill the relation:

$$G_c(q)\lambda_c + a_{c,f}(q,\dot{q},\ddot{q}_f) = 0$$

From an **energetic point of view**, this solution minimizes:

$$\min_{\lambda_c} \ \frac{1}{2}\lambda_c^\top G_c(q)\lambda_c + \lambda_c^\top a_{c,f}(q,\dot{q},\ddot{q}_f)$$

or using a $\max$:

$$\max_{\lambda_c} \ -\frac{1}{2}\lambda_c^\top(G_c(q)\lambda_c + 2\underbrace{\lambda_c^\top a_{c,f}(q,\dot{q},\ddot{q}_f))}_{a_c(q,\dot{q},\ddot{q})}$$

dual problem: maximum dissipation

$$\min_{\ddot{q}} \ \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$

$$J_c(q)\,\ddot{q} + \dot{J}_c(q,\dot{q})\dot{q} = 0$$

primal problem: least action principle

The contact forces then tend to maximize the dissipation of the kinetic energy!

Analytical Derivatives of Rigid Contact Dynamics

Analytical Derivatives of Robot Dynamics

Numerical Optimal Control or **Reinforcement Learning** approaches require access to **Forward or Inverse Dynamics** functions and their **partial derivatives**

Inverse Dynamics

$$\tau = \mathbf{ID}\left(q, \dot{q}, \ddot{q}, \lambda_c\right)$$

$$\frac{\partial \mathbf{ID}}{\partial q}, \frac{\partial \mathbf{ID}}{\partial \dot{q}}, \frac{\partial \mathbf{ID}}{\partial \ddot{q}}, \frac{\partial \mathbf{ID}}{\partial \lambda_c}$$

Forward Dynamics

$$\ddot{q} = \mathbf{FD}(q, \dot{q}, \tau, \lambda_c)$$

$$\frac{\partial \mathbf{FD}}{\partial q}, \frac{\partial \mathbf{FD}}{\partial \dot{q}}, \frac{\partial \mathbf{FD}}{\partial \tau}, \frac{\partial \mathbf{FD}}{\partial \lambda_c}$$

Classic ways to evaluate Numerical Derivatives

Finite Differences

> Consider the input function as **a black-box**

$$y = f(x)$$

> Add a **small increment** on the input variable

$$\frac{dy}{dx} \approx \frac{f(x + dx) - f(x)}{dx}$$

Pros

> Works for any input function

> Easy implementation

Cons

> Not efficient

> Sensitive to numerical rounding errors

Classic ways to evaluate Numerical Derivatives

Finite Differences

> Consider the input function as **a black-box**

$$y = f(x)$$

> Add a **small increment** on the input variable

$$\frac{dy}{dx} \approx \frac{f(x + dx) - f(x)}{dx}$$

Pros

> Works for any input function

> Easy implementation

Cons

> Not efficient

> Sensitive to numerical rounding errors

Automatic Differentiation

> This time, we know the **elementary operations** in f

$$y = f(x) = a \,.\, \cos(x)$$

> Apply the **chain rule formula** and use derivatives of basic functions

$$\frac{dy}{dx} = \underbrace{\frac{d\,a}{dx}}_{=0} .\cos(x) + a \,.\, \frac{d\cos(x)}{dx} = -\,a \,.\, \sin(x)$$

Pros

> Efficient frameworks

> Very accurate

Cons

> Requires specific implementation

> Not able to exploit spatial algebra derivatives

Analytical Derivatives of Dynamics Algorithms

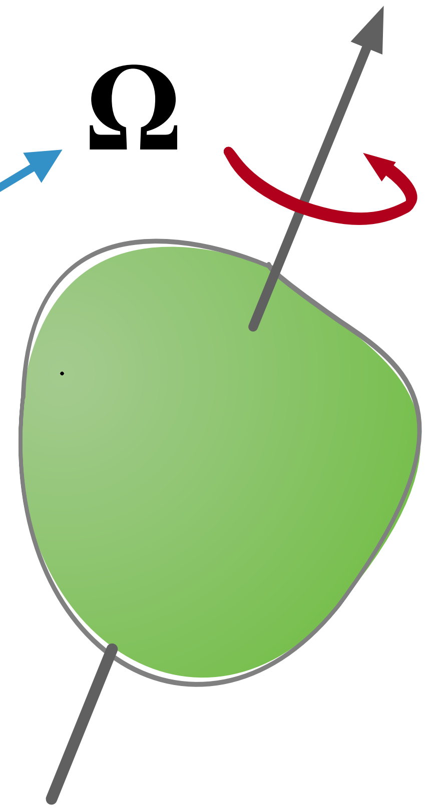
Why analytical derivatives?

We must exploit the **intrinsic geometry** of the **differential operators** involved in rigid motions

orientation matrix

$$\frac{d\mathbf{R}}{dt} = \mathbf{R}\,[\Omega]_\times$$

velocity vector



Analytical Derivatives of Dynamics Algorithms

The Recursive Newton-Euler algorithm
to compute $\boldsymbol{\tau} = \mathbf{ID}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$

Algorithm:

$\boldsymbol{v}_0 = \mathbf{0}$

$\boldsymbol{a}_0 = -\boldsymbol{a}_g$

for $i = 1$ **to** N_B **do**

$\quad [\boldsymbol{X}_J, \boldsymbol{S}_i, \boldsymbol{v}_J, \boldsymbol{c}_J] = \text{jcalc}(\text{jtype}(i), \boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)$

$\quad {}^i\boldsymbol{X}_{\lambda(i)} = \boldsymbol{X}_J \, \boldsymbol{X}_T(i)$

$\quad$ **if** $\lambda(i) \neq 0$ **then**

$\quad\quad {}^i\boldsymbol{X}_0 = {}^i\boldsymbol{X}_{\lambda(i)} \, {}^{\lambda(i)}\boldsymbol{X}_0$

$\quad$ **end**

$\quad \boldsymbol{v}_i = {}^i\boldsymbol{X}_{\lambda(i)} \, \boldsymbol{v}_{\lambda(i)} + \boldsymbol{v}_J$

$\quad \boldsymbol{a}_i = {}^i\boldsymbol{X}_{\lambda(i)} \, \boldsymbol{a}_{\lambda(i)} + \boldsymbol{S}_i \, \ddot{\boldsymbol{q}}_i + \boldsymbol{c}_J + \boldsymbol{v}_i \times \boldsymbol{v}_J$

$\quad \boldsymbol{f}_i = \boldsymbol{I}_i \, \boldsymbol{a}_i + \boldsymbol{v}_i \times^* \boldsymbol{I}_i \, \boldsymbol{v}_i - {}^i\boldsymbol{X}_0^* \, \boldsymbol{f}_i^x$

end

for $i = N_B$ **to** 1 **do**

$\quad \boldsymbol{\tau}_i = \boldsymbol{S}_i^T \boldsymbol{f}_i$

$\quad$ **if** $\lambda(i) \neq 0$ **then**

$\quad\quad \boldsymbol{f}_{\lambda(i)} = \boldsymbol{f}_{\lambda(i)} + {}^{\lambda(i)}\boldsymbol{X}_i^* \, \boldsymbol{f}_i$

$\quad$ **end**

end

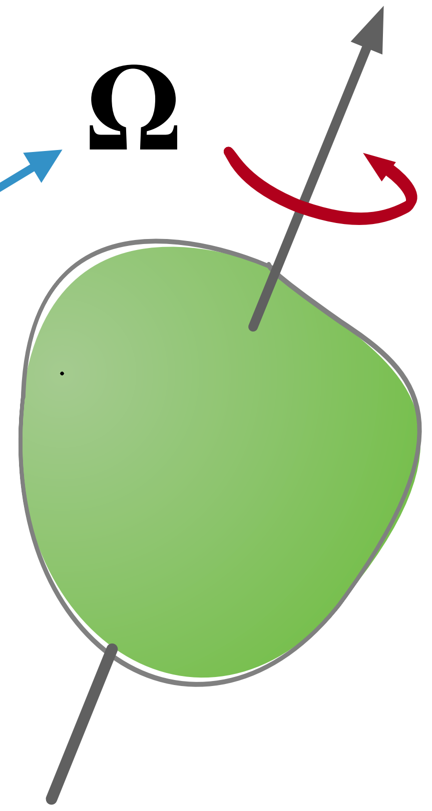
Why analytical derivatives?

We must exploit the **intrinsic geometry** of the **differential operators** involved in rigid motions

$$\frac{d\mathbf{R}}{dt} = \mathbf{R}\,[\Omega]_\times$$

orientation matrix

velocity vector



Summary of the methodology

Applying the **chain rule formula** on each line of the Recursive Newton-Euler algorithm AND **exploiting the sparsity** of spatial operations

Analytical Derivatives of Dynamics Algorithms

The Recursive Newton-Euler algorithm
to compute $\boldsymbol{\tau} = \mathbf{ID}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$

Algorithm:

$\boldsymbol{v}_0 = \mathbf{0}$
$\boldsymbol{a}_0 = -\boldsymbol{a}_g$
for $i = 1$ **to** N_B **do**
  $[\boldsymbol{X}_J, \boldsymbol{S}_i, \boldsymbol{v}_J, \boldsymbol{c}_J] = \text{jcalc}(\text{jtype}(i), \boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)$
  ${}^{i}\boldsymbol{X}_{\lambda(i)} = \boldsymbol{X}_J\, \boldsymbol{X}_T(i)$
  if $\lambda(i) \neq 0$ **then**
    ${}^{i}\boldsymbol{X}_0 = {}^{i}\boldsymbol{X}_{\lambda(i)}\, {}^{\lambda(i)}\boldsymbol{X}_0$
  end
  $\boldsymbol{v}_i = {}^{i}\boldsymbol{X}_{\lambda(i)}\, \boldsymbol{v}_{\lambda(i)} + \boldsymbol{v}_J$
  $\boldsymbol{a}_i = {}^{i}\boldsymbol{X}_{\lambda(i)}\, \boldsymbol{a}_{\lambda(i)} + \boldsymbol{S}_i\, \ddot{\boldsymbol{q}}_i + \boldsymbol{c}_J + \boldsymbol{v}_i \times \boldsymbol{v}_J$
  $\boldsymbol{f}_i = \boldsymbol{I}_i\, \boldsymbol{a}_i + \boldsymbol{v}_i \times^* \boldsymbol{I}_i\, \boldsymbol{v}_i - {}^{i}\boldsymbol{X}_0^*\, \boldsymbol{f}_i^x$
end
for $i = N_B$ **to** 1 **do**
  $\boldsymbol{\tau}_i = \boldsymbol{S}_i^{\mathrm{T}}\, \boldsymbol{f}_i$
  if $\lambda(i) \neq 0$ **then**
    $\boldsymbol{f}_{\lambda(i)} = \boldsymbol{f}_{\lambda(i)} + {}^{\lambda(i)}\boldsymbol{X}_i^*\, \boldsymbol{f}_i$
  end
end

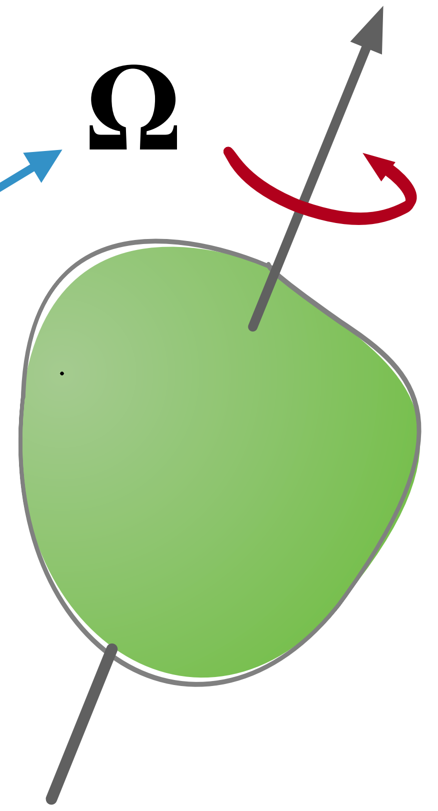
Why analytical derivatives?

We must exploit the **intrinsic geometry** of the **differential operators** involved in rigid motions

$$\frac{d\,\mathbf{R}}{dt} = \mathbf{R}\,[\Omega]_\times$$

orientation matrix

velocity vector



Summary of the methodology

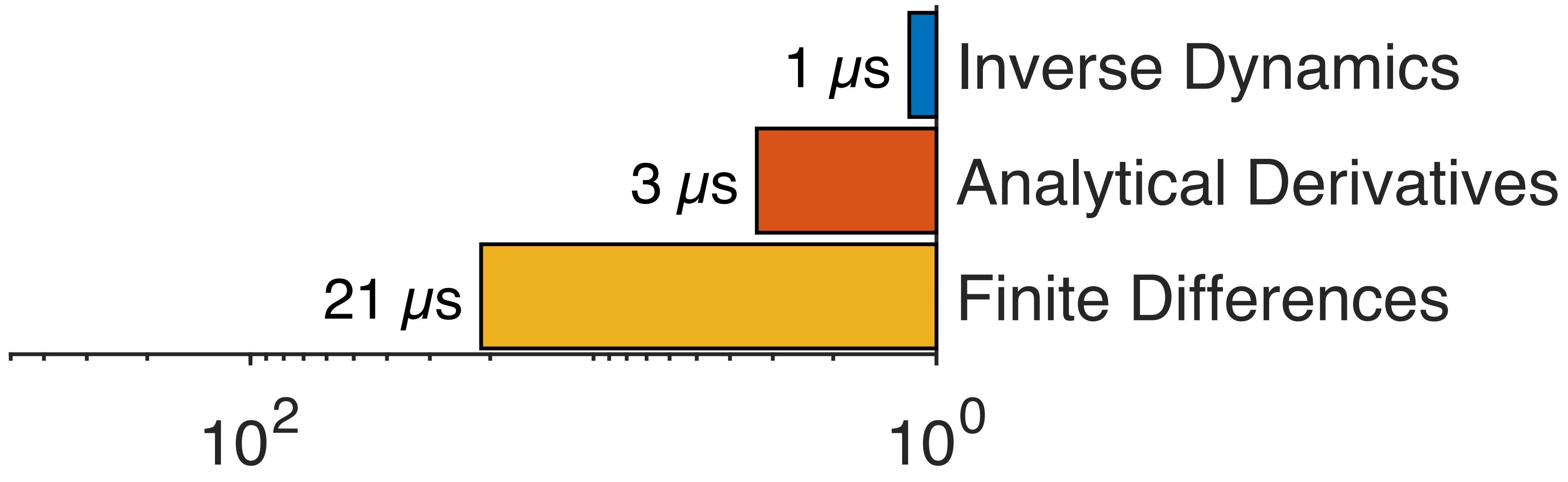
Applying the **chain rule formula** on each line of the Recursive Newton-Euler algorithm
AND **exploiting the sparsity** of spatial operations

Outcome

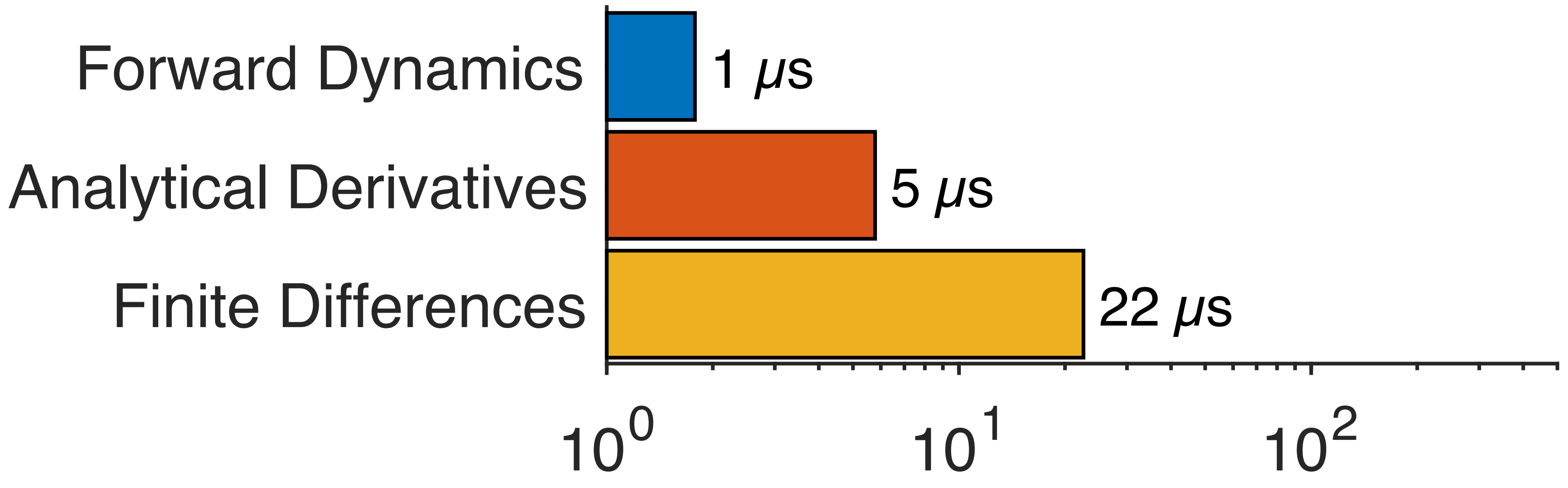
A **simple** but **efficient** algorithm, relying on spatial algebra
AND keeping a **minimal complexity of O(Nd)** WHILE **the state of the art is O(N²)**

Benchmarks of analytical derivatives

Inverse Dynamics



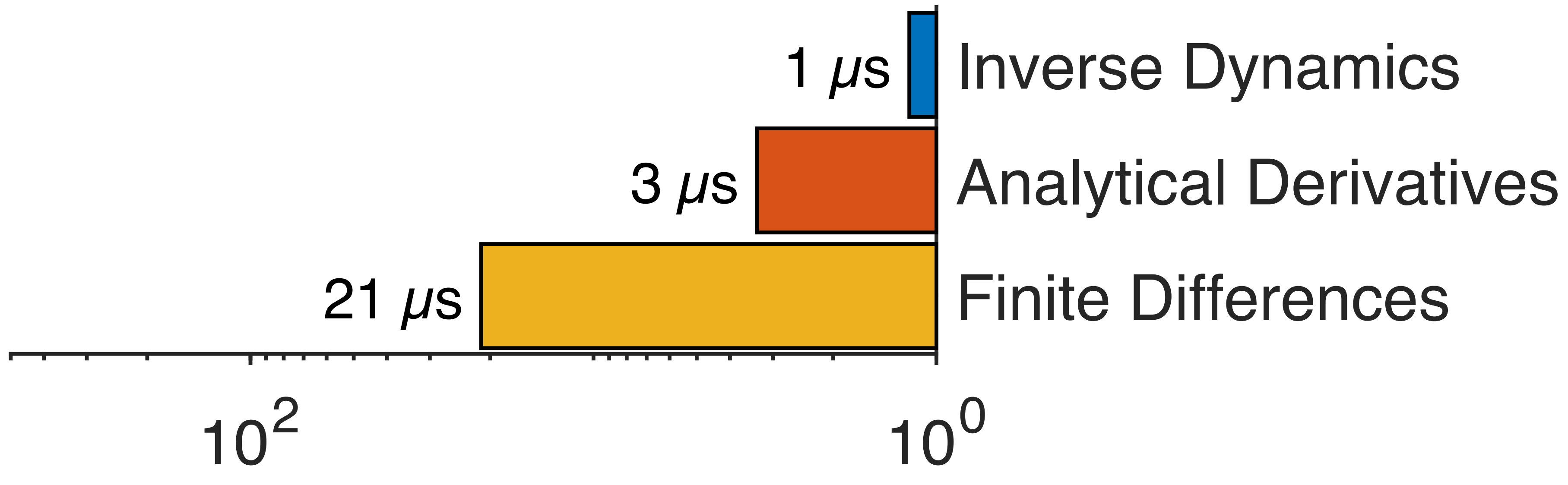
Forward Dynamics

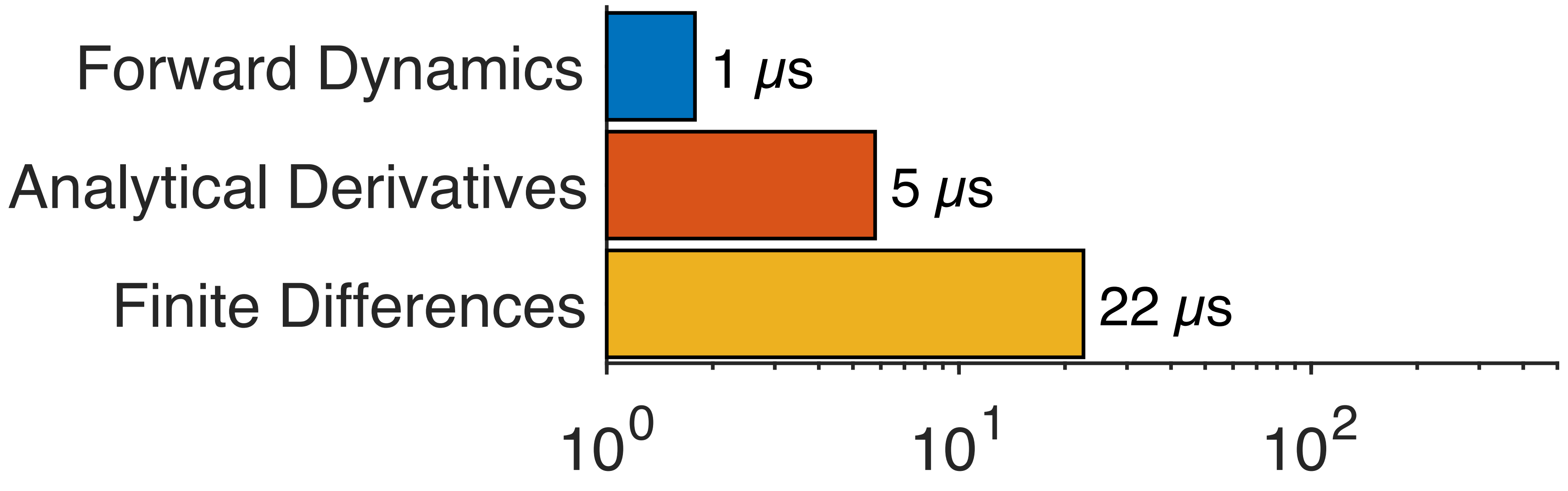


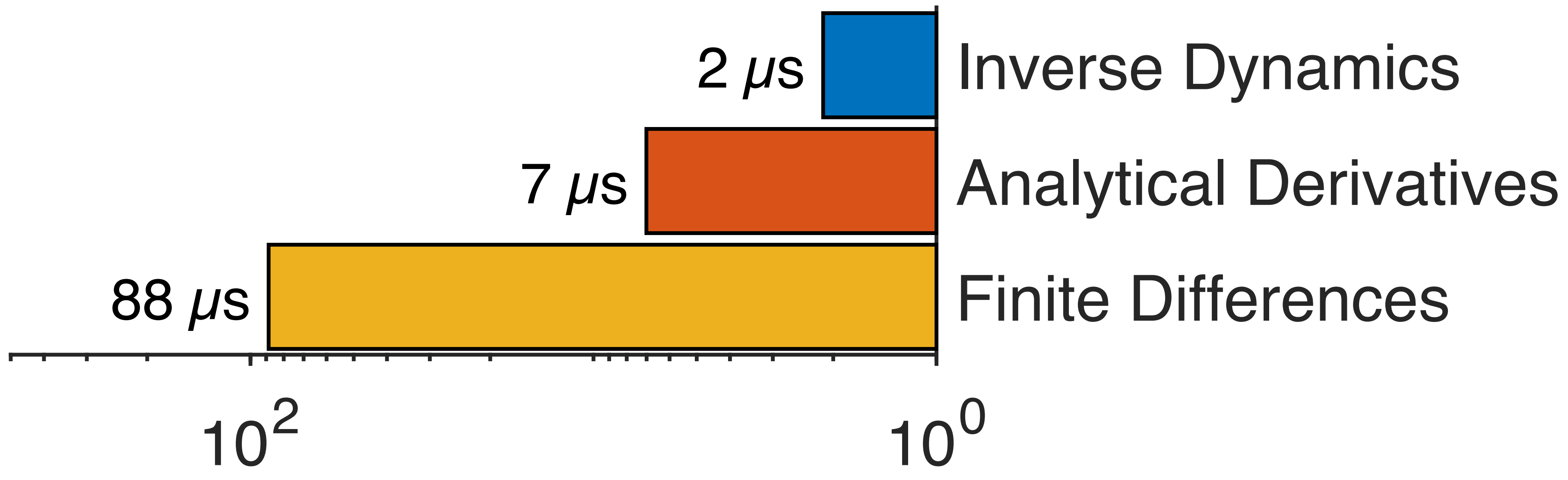
Benchmarks of analytical derivatives

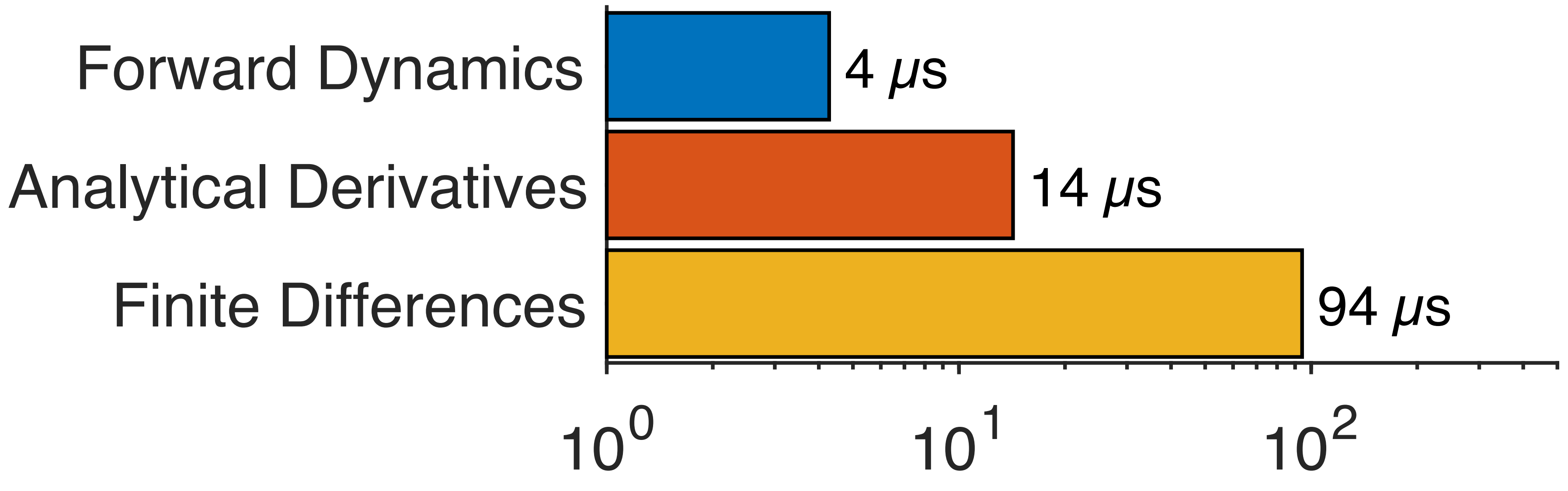
Inverse Dynamics

Forward Dynamics





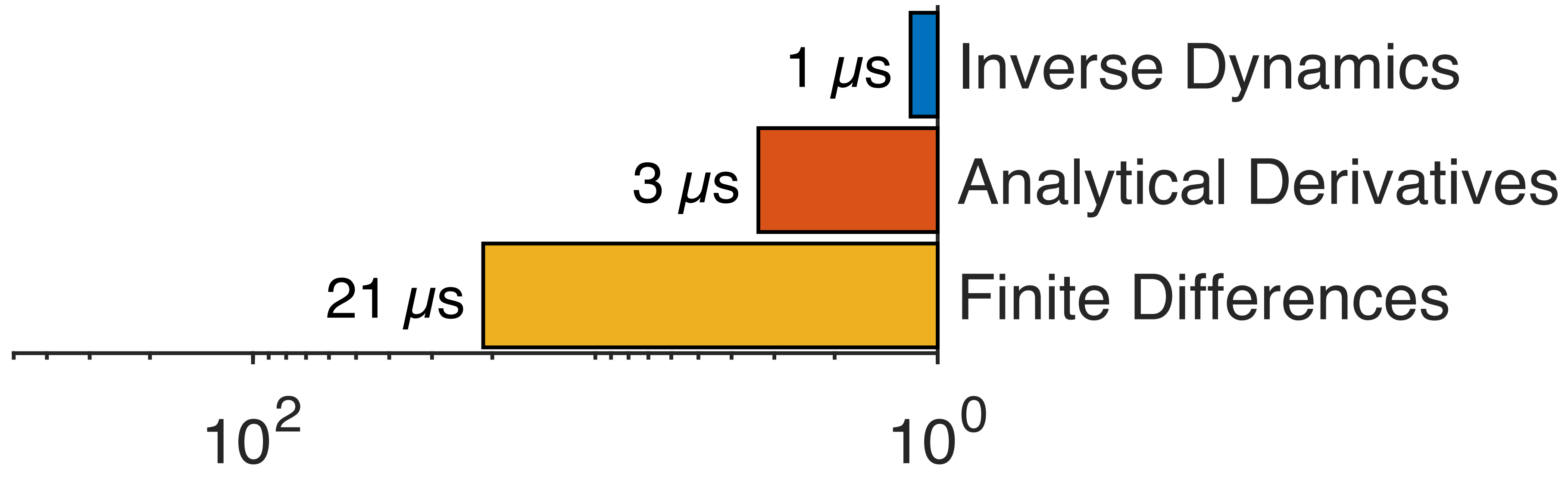


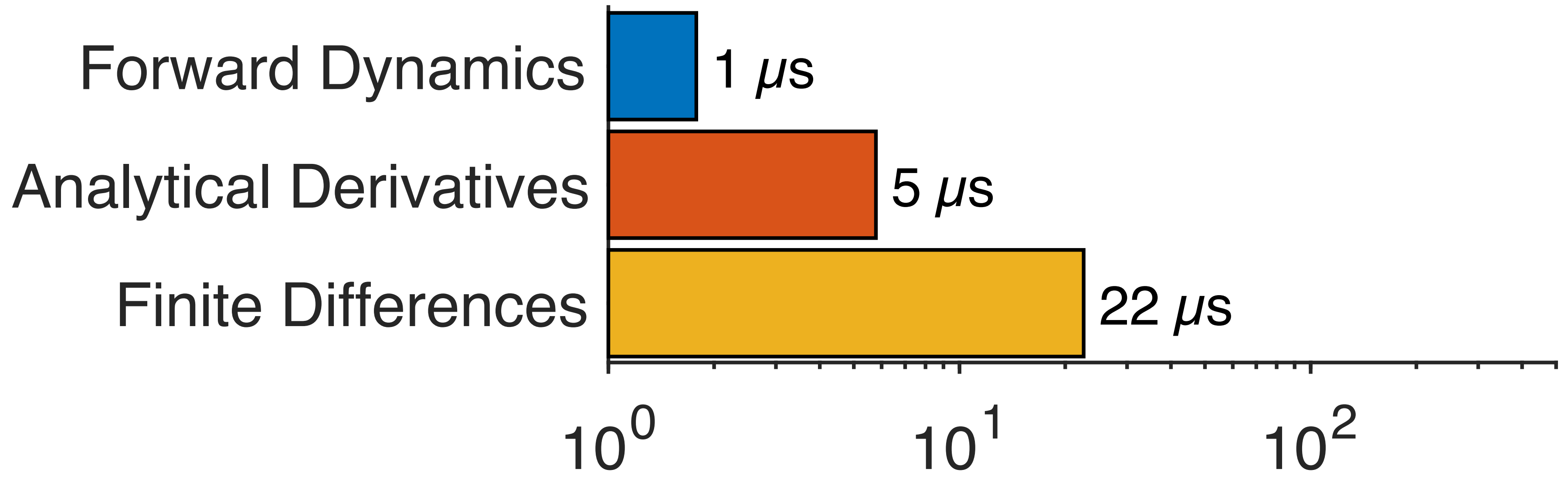


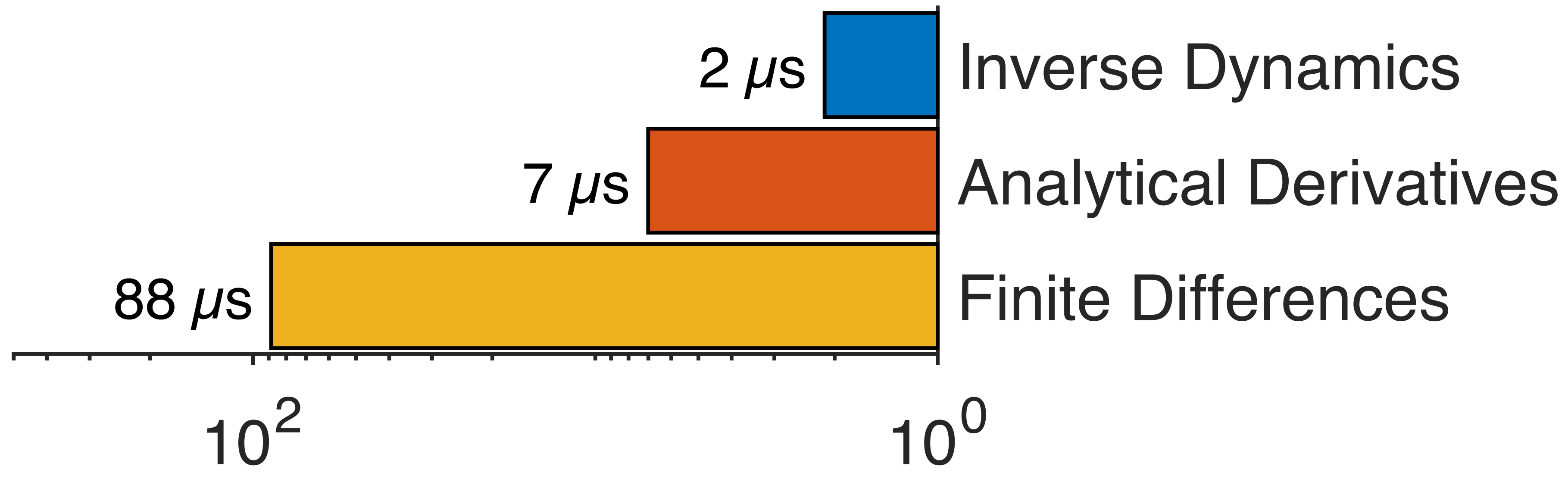
Benchmarks of analytical derivatives

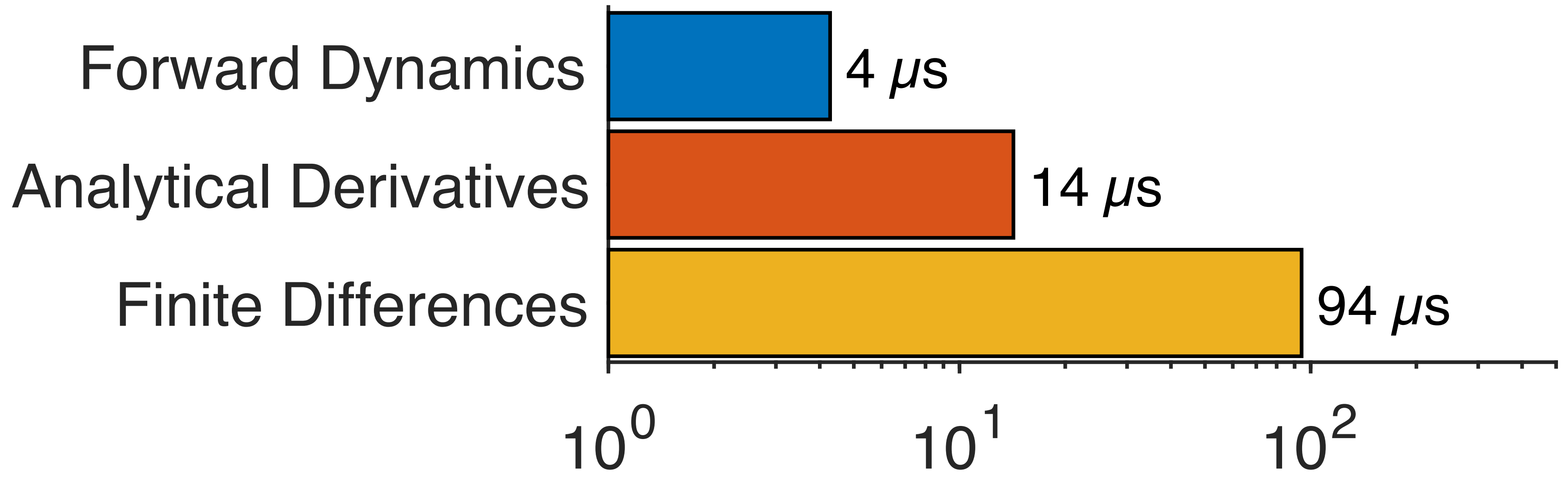
Inverse Dynamics

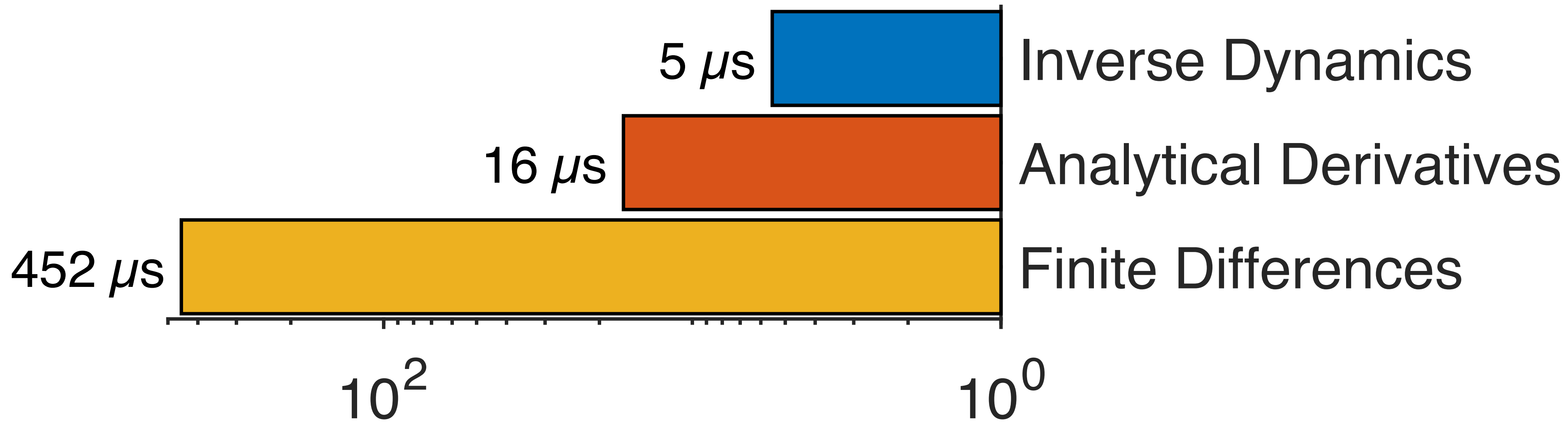
Forward Dynamics

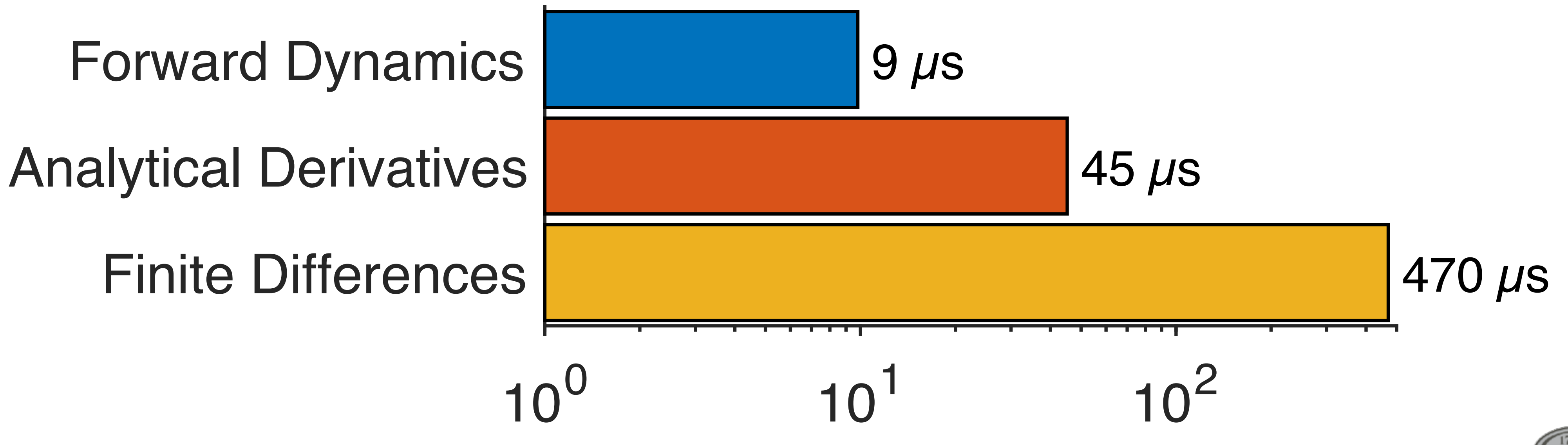












Analytical Derivatives of Contact Dynamics

Remind that the contact dynamics is provided by:

$$\underbrace{\begin{bmatrix} M(q) & J_c^\top(q) \\ J_c(q) & 0 \end{bmatrix}}_{K(q)} \begin{bmatrix} \ddot{q} \\ -\lambda_c \end{bmatrix} = \begin{bmatrix} M(q)\ddot{q}_f \\ -\gamma_c(q,\dot{q}) \end{bmatrix}$$

Without too much difficulty, one can show that the contact derivatives are given by:

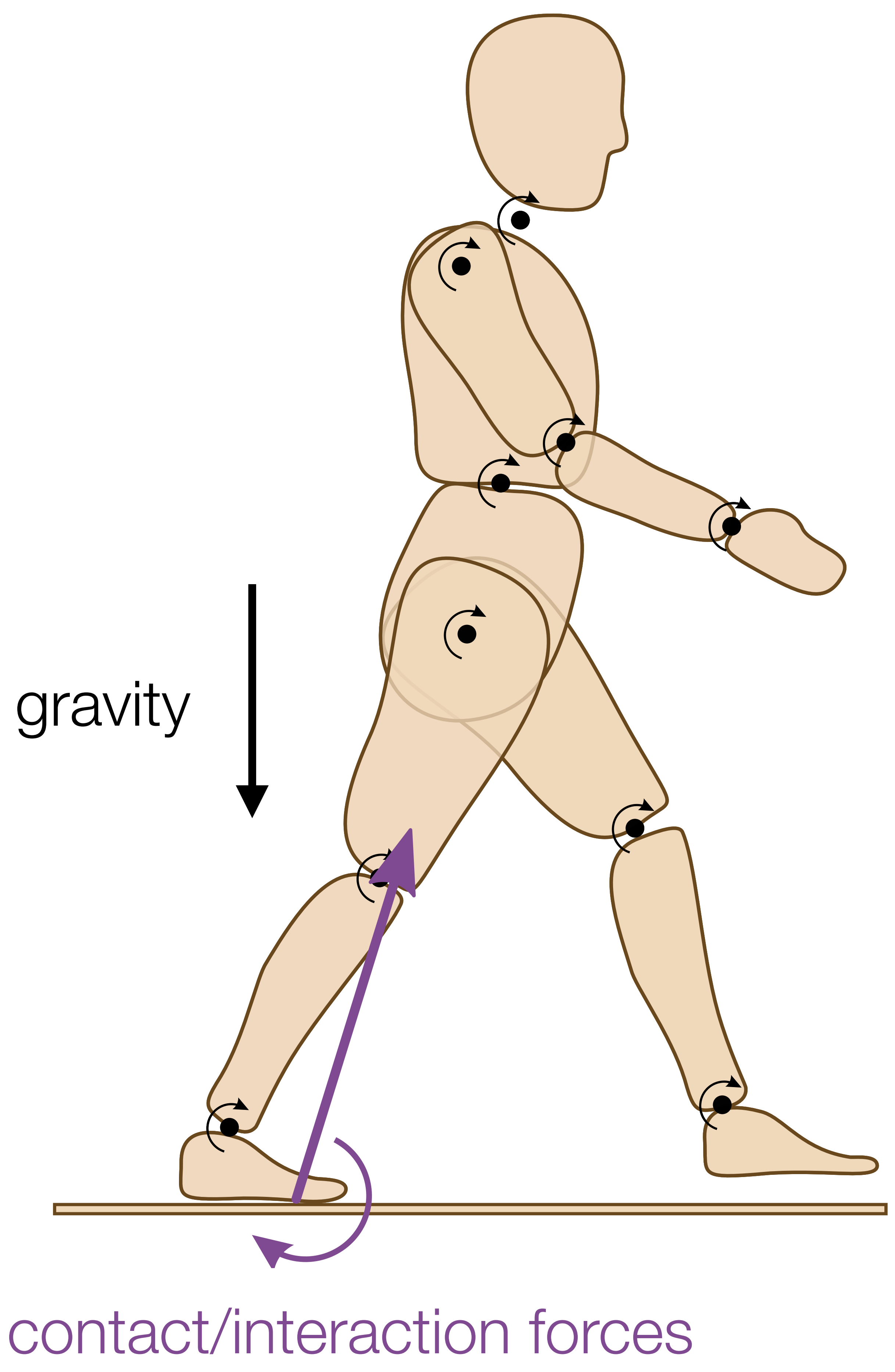
$$\begin{bmatrix} \frac{\partial \ddot{q}}{\partial x} \\ -\frac{\partial \lambda_c}{\partial x} \end{bmatrix} = -K^{-1}(q) \begin{bmatrix} \frac{\partial \text{ID}}{\partial x}(q,\dot{q},\ddot{q},\lambda_c) \\ \frac{\partial a_c}{\partial x}(q,\dot{q},\ddot{q}) \end{bmatrix}$$

Only depends on **known analytical derivatives**

The Rigid Contact Problem

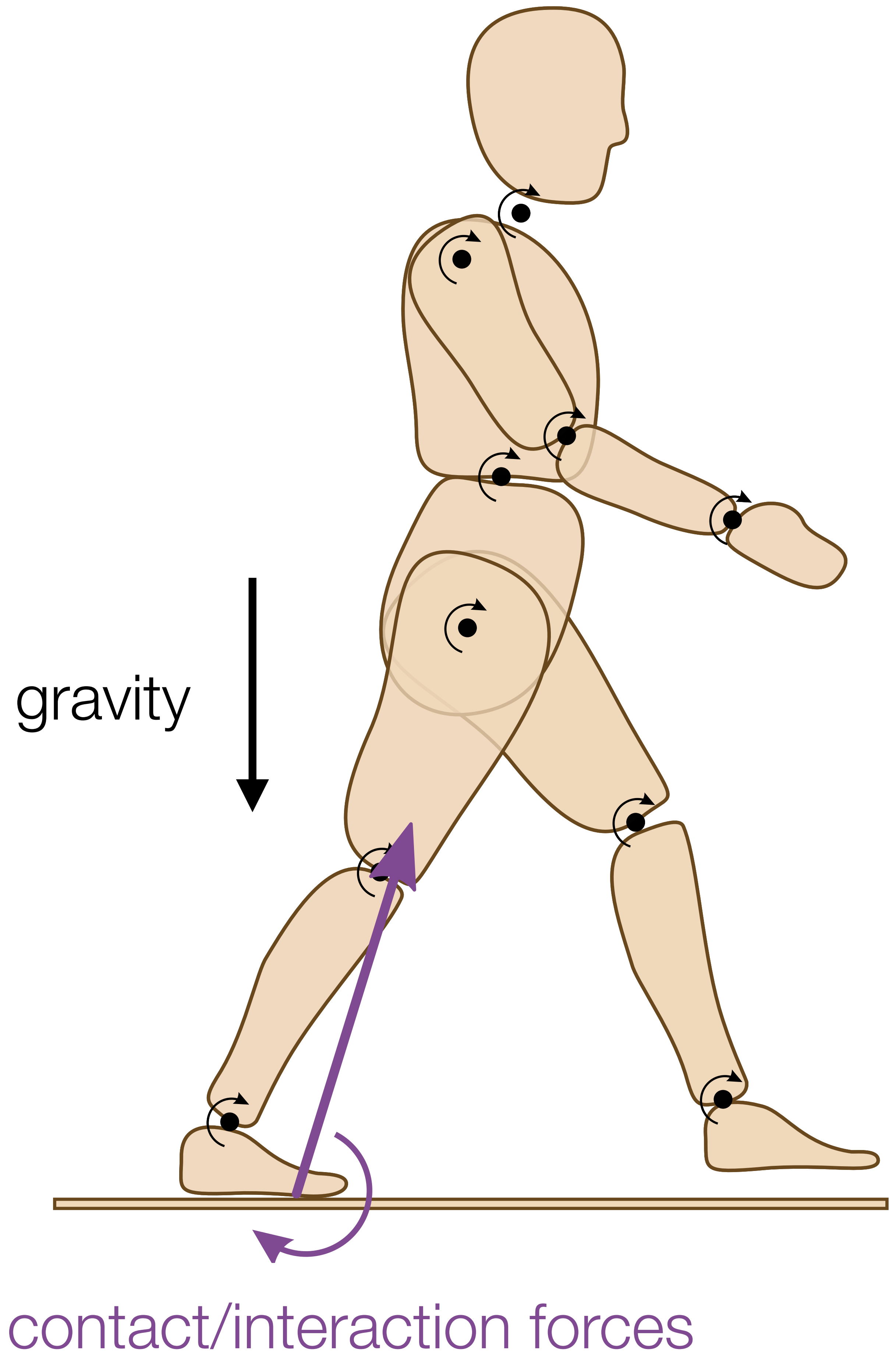
unilateral contacts

Unilateral Contact Model



When dealing with unilateral contact conditions, **three conditions** are required:

Unilateral Contact Model

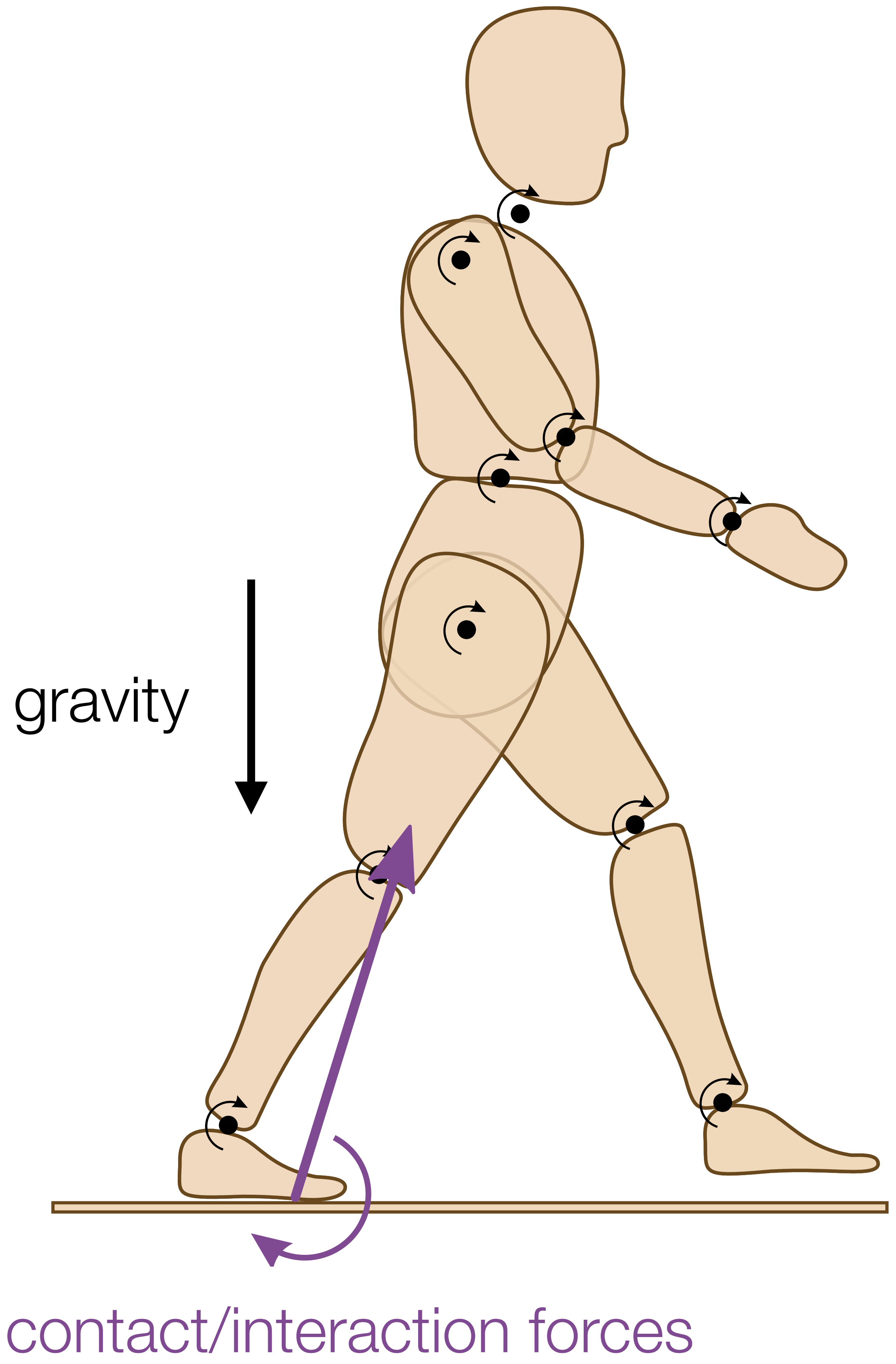


When dealing with unilateral contact conditions, **three conditions** are required:

- **Maximum dissipation:**
  the contact forces **should dissipate** at most the kinetic energy

$$\max_{\lambda_c} \; -\frac{1}{2}\lambda_c^\top (G_c(q)\lambda_c + 2\lambda_c^\top a_{c,f}(q,\dot{q},\ddot{q}_f))$$

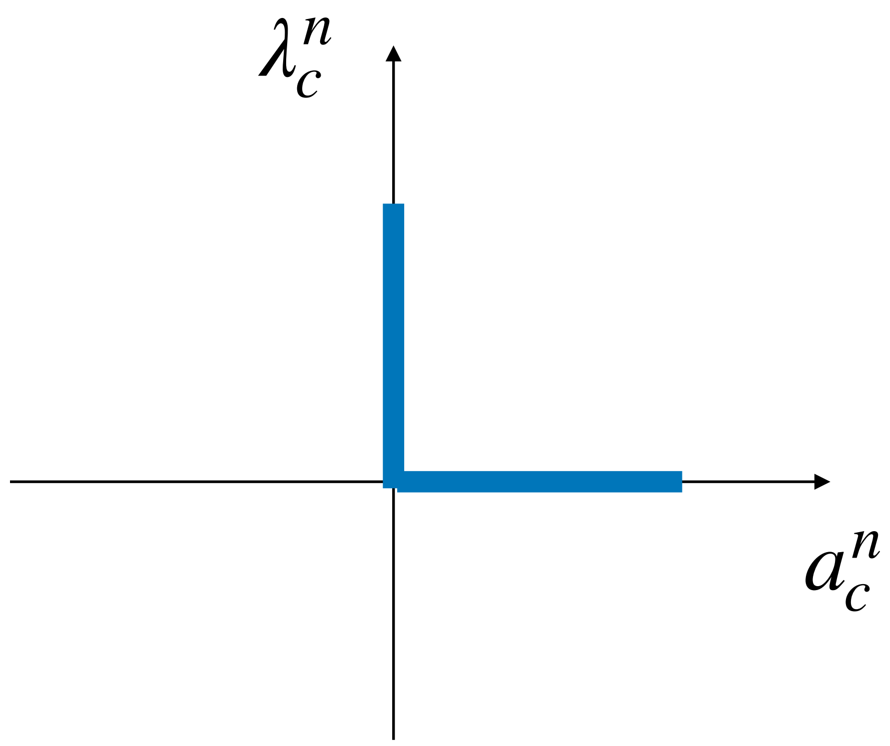
Unilateral Contact Model



When dealing with unilateral contact conditions,
three conditions are required:

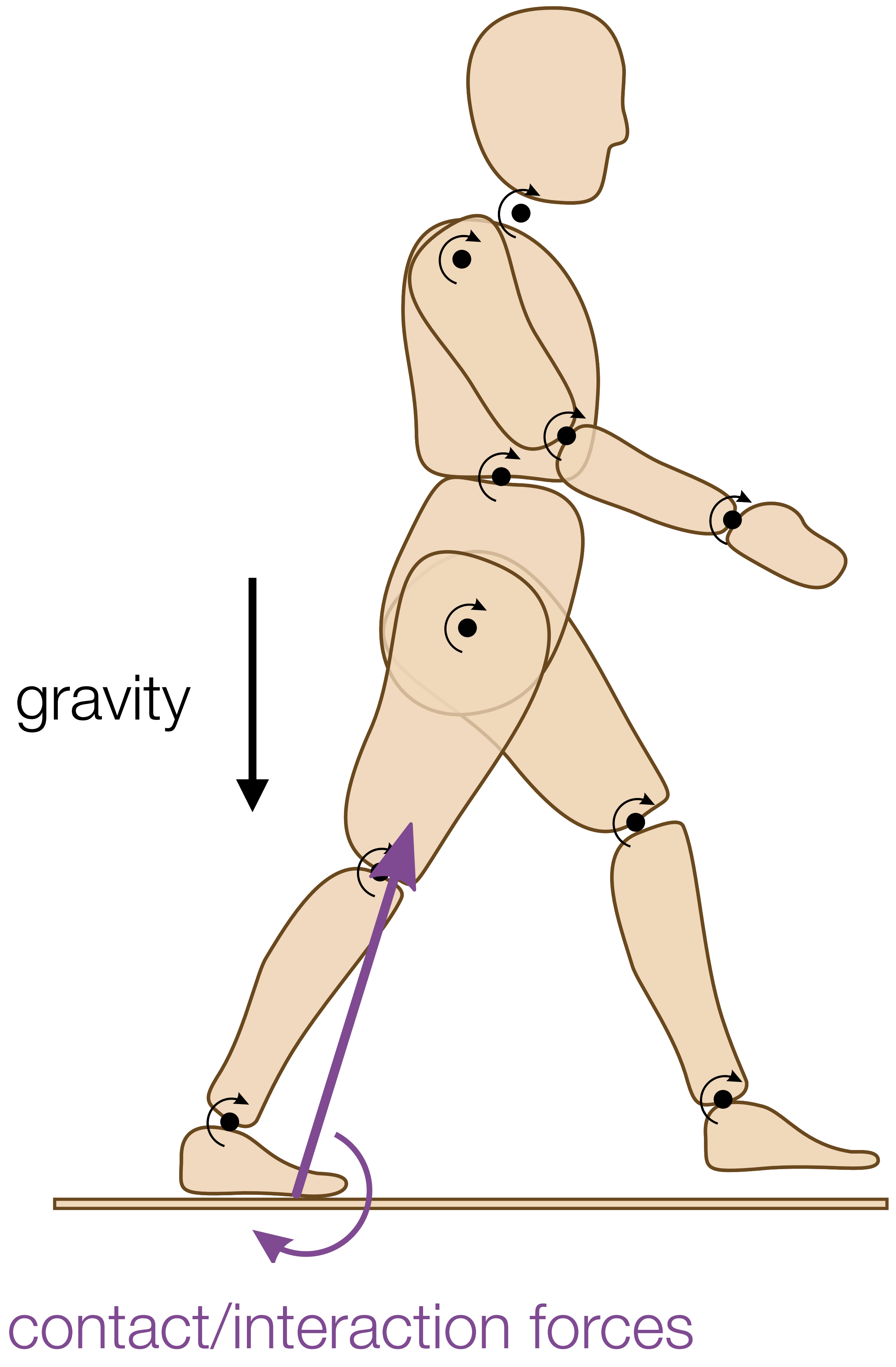
- **Maximum dissipation:**
  the contact forces **should dissipate** at most the kinetic energy
- **Complementary condition (Signorini's conditions):**
  the floor can **only push** (no pulling) + **no force** when the contact is about to open

$$\max_{\lambda_c} \; -\frac{1}{2}\lambda_c^\top (G_c(q)\lambda_c + 2\lambda_c^\top a_{c,f}(q,\dot{q},\ddot{q}_f))$$



$$0 \leq \lambda_{c,n} \perp a_{c,n} \geq 0$$

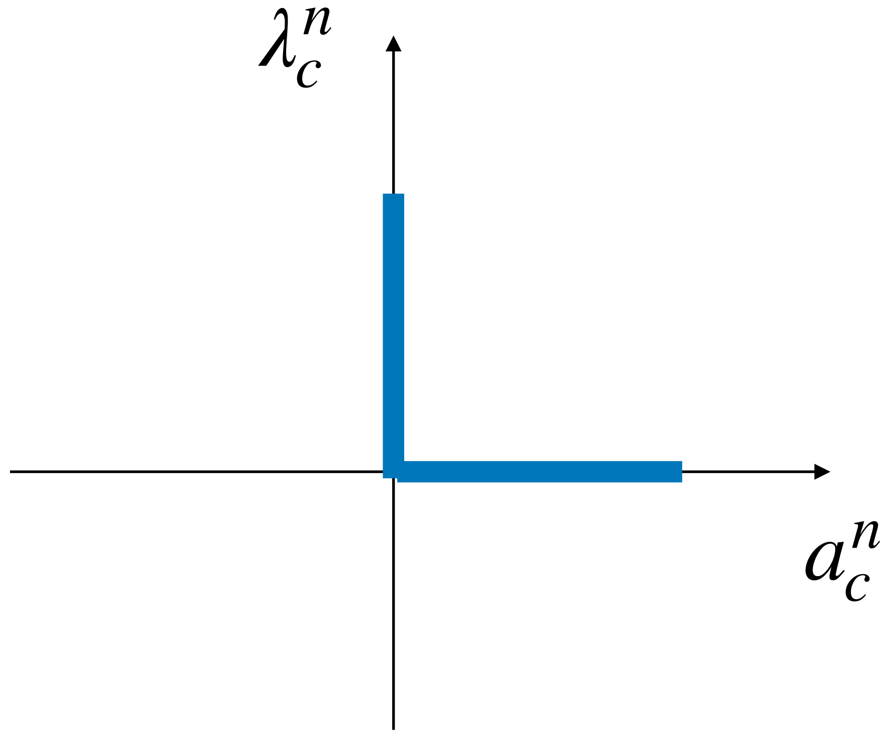
Unilateral Contact Model



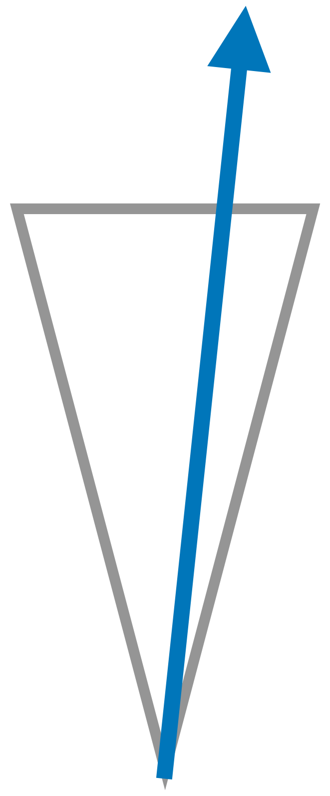
When dealing with unilateral contact conditions,
three conditions are required:

- **Maximum dissipation:**
  the contact forces **should dissipate** at most the kinetic energy
- **Complementary condition (Signorini's conditions):**
  the floor can **only push** (no pulling) + **no force** when the contact is about to open
- **Friction cone constraint (Coulomb law):**
  the lateral forces **are bounded** by the normal force

$$\max_{\lambda_c} \; -\frac{1}{2}\lambda_c^\top (G_c(q)\lambda_c + 2\lambda_c^\top a_{c,f}(q, \dot{q}, \ddot{q}_f))$$

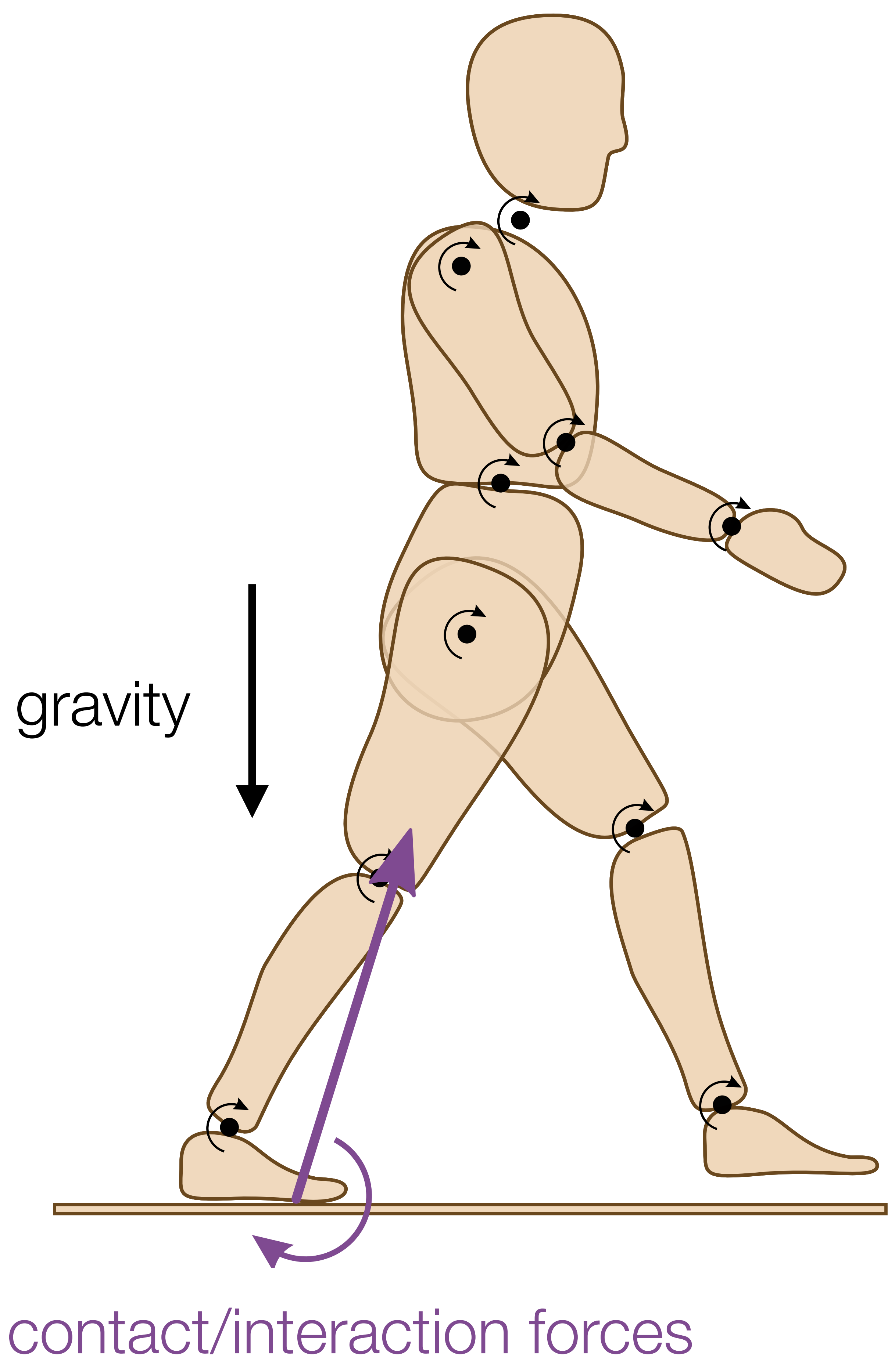


$$0 \leq \lambda_{c,n} \perp a_{c,n} \geq 0$$



$$\sqrt{\lambda_{c,x}^2 + \lambda_{c,y}^2} \leq \mu \lambda_{c,n}$$

Unilateral Contact Problem



The contact problem then corresponds to
a so-called **Nonlinear Complementary Problem**:

$$\min_{\lambda_c} \frac{1}{2}\lambda_c^\top G_c(q)\lambda_c + \lambda_c^\top a_{c,f}(q,\dot{q},\ddot{q}_f)$$ maximum dissipation

$$\sqrt{\lambda_{c,x}^2 + \lambda_{c,y}^2} \leq \mu \lambda_{c,n}$$ Coulomb friction

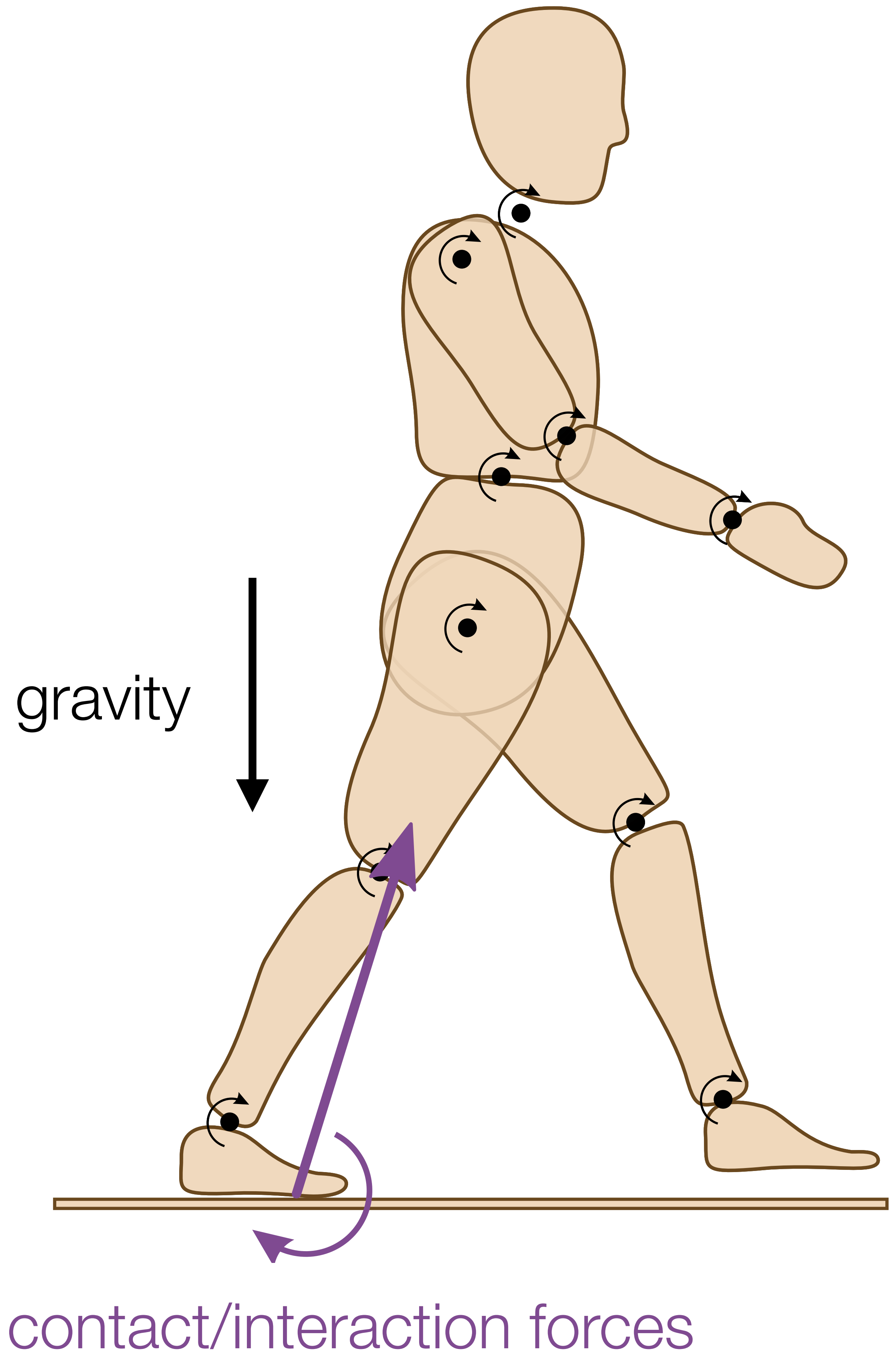
$$0 \leq \lambda_{c,n} \perp a_{c,n} \geq 0$$ contact complementarity

which is **nonconvex (hard to solve)**!

The Relaxed Contact Problem

a mix between rigid and soft

The Relaxed Contact Problem



The contact problem can be relaxed by
removing the complementarity condition AND regularization the forces:

$$\min_{\lambda_c} \frac{1}{2}\lambda_c^\top \left(G_c(q) + R\right)\lambda_c + \lambda_c^\top a_{c,f}(q, \dot{q}, \ddot{q}_f)$$

maximum dissipation + regularization

$$\sqrt{\lambda_{c,x}^2 + \lambda_{c,y}^2} \leq \mu \lambda_{c,n}$$

Coulomb friction

~~$0 \leq \lambda_{c,n} \perp a_{c,n} \geq 0$~~

No contact complementarity

which becomes **convex (easier to solve)**
but with some physical inconsistencies!