

Chapter 2

Mathematical Preliminaries

2.1 Biomechanical Action Principle and Variational Methods

There is a single mathematical, physical and biomechanical concept that underpins most of the formal derivations presented in this book. We call it *biomechanical action principle*. It is rooted in variational calculus and closely related to optimal control. It appears, in different ‘flavors’, in several chapters of this book, as: (i) *myofascioskeletal action principle*, (ii) *microstretch action principle*, and (iii) *lymphodynamics action principle*. In this section, we will try to give a ‘soft’, semiformal, biomechanical introduction to this important concept of modern science.

Any kind of *human movement*, considered either at the level of the whole myofascioskeletal system (or one of its parts, called the ‘kinetic chain’), or at the level of a single muscle, or at the level of a certain part of human fascia (e.g., lumbosacral fascia), or at the level of lymph movement/flow within a certain part of the body – can be represented by a set of n local *degrees-of-freedom* (DOF) with the associated set of n local trajectories $q^i(t)$, including local translations and rotations given as functions of time, for $i = 1, \dots, n$.

More specifically, any one of these local $q^i(t)$ -movements can be defined as a *transition* T from some *initial configuration* Σ_0 [with initial coordinates $q_0^i = q^i(t_0)$] – to the corresponding *final configuration* Σ_1 [with final coordinates $q_1^i = q^i(t_1)$]. So, each local movement, occurring at the level of the body, muscle, fascia or lymph, represents a transition $T : \Sigma_0 \rightarrow \Sigma_1$, from the initial configuration Σ_0 (coordinated by q_0^i), to the final configuration Σ_1 (coordinated by q_1^i).

This transition $T : \Sigma_0 \rightarrow \Sigma_1$ naturally occurs along a certain *path* (that is, a set of local trajectories), which can be either a direct path $q^i = q^i(t)$, or slightly deformed, indirect path $\bar{q}^i = \bar{q}^i(t)$. The difference between the direct and indirect paths:

$$\delta q^i(t) = \bar{q}^i(t) - q^i(t)$$

is called the *variation* of the path (that is, variation of all the DOF, both translational and rotational), where the symbol δ represents the small variation of the path (that is a commutative linear infinitesimal operator that is interchangeable with both derivatives and integrals; see, e.g. [5] and the references therein). The time derivative of $\delta q^i(t)$:

$$\delta \dot{q}^i(t) = \frac{d}{dt} [\delta q^i(t)] = \frac{d}{dt} [\bar{q}^i(t) - q^i(t)]$$

represents the small variation of the velocities, while the second time derivative:

$$\delta \ddot{q}^i(t) = \frac{d}{dt} [\delta \dot{q}^i(t)] = \frac{d^2}{dt^2} [\bar{q}^i(t) - q^i(t)]$$

represents the small variation of the accelerations.

Now, any (autonomous) dynamical quantity $F = F[q^i(t), \dot{q}^i(t)]$ defined on the direct path has the corresponding *deformation* defined on any indirect path:

$$\bar{F} = \bar{F}[\bar{q}^i(t), \bar{\dot{q}}^i(t)] = F[q^i(t) + \delta q^i(t), \dot{q}^i(t) + \delta \dot{q}^i(t)],$$

which allows the following Taylor expansion:

$$\bar{F} = F + \frac{\partial F}{\partial q^i} \delta q^i + \frac{\partial F}{\partial \dot{q}^i} \delta \dot{q}^i + \frac{\partial F}{\partial \ddot{q}^i} \delta \ddot{q}^i + R, \quad (R = \text{remainder terms}),$$

from which the quantity δF given by:

$$\delta F = \frac{\partial F}{\partial q^i} \delta q^i + \frac{\partial F}{\partial \dot{q}^i} \delta \dot{q}^i$$

is called the first variation of the dynamical quantity F , while the quantity $\delta^2 F$ given by:

$$\delta^2 F = \frac{\partial^2 F}{\partial q^i \partial q^j} \delta q^i \delta q^j + \frac{\partial^2 F}{\partial \dot{q}^i \partial \dot{q}^j} \delta \dot{q}^i \delta \dot{q}^j + \frac{\partial^2 F}{\partial \ddot{q}^i \partial \ddot{q}^j} \delta \ddot{q}^i \delta \ddot{q}^j$$

is called its second variation, etc.

For majority of biomechanical purposes, the dynamical quantity of interest would be the (autonomous) *Lagrangian* energy function $L = L[q^i(t), \dot{q}^i(t)]$, usually defined as kinetic minus potential energy, and only its first variation δL is usually used as¹:

¹The second variation of the Lagrangian:

$$\delta^2 L = \frac{\partial^2 L}{\partial q^i \partial q^j} \delta q^i \delta q^j + \frac{\partial^2 L}{\partial \dot{q}^i \partial \dot{q}^j} \delta \dot{q}^i \delta \dot{q}^j + \frac{\partial^2 L}{\partial \ddot{q}^i \partial \ddot{q}^j} \delta \ddot{q}^i \delta \ddot{q}^j$$

is used only in more sophisticated optimal control algorithms.

$$\delta L = \frac{\partial L}{\partial q^i} \delta q^i + \frac{\partial L}{\partial \dot{q}^i} \delta \dot{q}^i.$$

Given these basic ingredients of variational calculus, we can now formulate the so-called *Hamilton's action functional* as a temporal integral of the Lagrangian:

$$\mathcal{A}([q^i], t_0, t_1) = \int_{t_0}^{t_1} L[q^i(t), \dot{q}^i(t)] dt. \quad (2.1)$$

and the governing action principle called *Hamilton's principle* of the stationary action (see, e.g. [6] and the references therein):

$$\mathcal{A} \longrightarrow \min \quad \text{or} \quad \delta \mathcal{A} = 0, \quad (2.2)$$

which formally states that the transition $T : \Sigma_0 \rightarrow \Sigma_1$ from the initial configuration Σ_0 to the final configuration Σ_1 always occurs with the minimum action, that is, with the minimum energy expenditure. We can also interpret it as a general *optimal control* problem: the movement transition $T : \Sigma_0 \rightarrow \Sigma_1$ effectively minimizes the cost function \mathcal{A} given by (2.1).

Provided $\delta q^i \neq 0$ are arbitrary small variations with fixed endpoints: $\delta q^i(t_0) = \delta q^i(t_1) = 0$ and using standard techniques of the calculus of variations, from Hamilton's action principle (2.2) we have:

$$\delta \mathcal{A} = \delta \int_{t_0}^{t_1} L(q^i, \dot{q}^i) dt = \int_{t_0}^{t_1} \delta L(q^i, \dot{q}^i) dt = \int_{t_0}^{t_1} \frac{\delta \mathcal{A}}{\delta q^i} \delta q^i dt = 0,$$

where $\frac{\delta \mathcal{A}}{\delta q^i}$ is the functional derivative [6] defined as:

$$\frac{\delta \mathcal{A}}{\delta q^i} = \frac{\partial L}{\partial q^i} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}^i}.$$

In other words, the stationary action principle can be formulated as the vanishing functional derivative,

$$\frac{\delta \mathcal{A}}{\delta q^i} = 0.$$

In this way, we have:

$$\delta \mathcal{A} = \int_{t_0}^{t_1} \left(\frac{\partial L}{\partial q^i} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}^i} \right) \delta q^i dt = 0,$$

from which standard Lagrangian equations of motion follow as:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}^i} = \frac{\partial L}{\partial q^i}, \quad (\text{for } i = 1, \dots, n).$$

The above variational action derivations will be much clearer with a simple example. Consider a three-dimensional (3D) motion $q^i(t)$, $i = 1, 2, 3$, of a single *Newtonian particle* with mass m within the *potential field* $V(q^i)$. This could be, e.g., a bio-mechanical example of motion of the human body *center of mass* (CoM). Particle's action:

$$\mathcal{A}([q^i], t_0, t_1) = \int_{t_0}^{t_1} \left[\frac{1}{2} m (\dot{q}^i)^2 - V(q^i) \right] dt, \quad (2.3)$$

is a function of the initial and final times, t_0 and t_1 (written as $\mathcal{A}(t_0, t_1)$), and also a *functional of the path* $q^i(t)$ (written as $\mathcal{A}[q^i]$) from t_0 to t_1 . Consider a small variation of the path:

$$q^i(t) \rightarrow q^i(t) + \delta q^i(t),$$

which is reflected in the action (2.3) as:

$$\begin{aligned} \mathcal{A}[q^i + \delta q^i] &= \int_{t_0}^{t_1} \left(\frac{1}{2} m \left[\frac{d(q^i + \delta q^i)}{dt} \right]^2 - V(q^i + \delta q^i) \right) dt \\ &= \mathcal{A}[q^i] + \int_{t_0}^{t_1} [-m\ddot{q}^i - \partial_i V(q^i)] \delta q^i dt + m \int_{t_0}^{t_1} \frac{d}{dt} (\delta q^i \dot{q}^i) dt, \\ \text{where } \partial_i &\equiv \frac{\partial}{\partial q^i} \quad \text{and} \quad V(q^i + \delta q^i) = V(q^i) + \delta q^i \partial_i V(q^i). \end{aligned} \quad (2.4)$$

The integral $m \int_{t_0}^{t_1} \frac{d}{dt} (\delta q^i \dot{q}^i) dt$ is the so-called ‘surface’ term, which vanishes if the variations have fixed endpoints: $\delta q^i(t_0) = \delta q^i(t_1) = 0$.

Therefore the variation of the action (2.4) becomes:

$$\begin{aligned} \mathcal{A}[q^i + \delta q^i] &= \mathcal{A}[q^i] + \int_{t_0}^{t_1} \frac{\delta \mathcal{A}}{\delta q^i} \delta q^i dt, \quad \text{with:} \\ \frac{\delta \mathcal{A}}{\delta q^i} &= -(m\ddot{q}^i + \partial_i V(q^i)). \end{aligned}$$

In this way, the minimization of the particle action (2.3) is equivalent to:

$$\frac{\delta \mathcal{A}}{\delta q^i} \equiv -[m\ddot{q}^i + \partial_i V(q^i)] = 0 \implies m\ddot{q}^i = \partial_i V(q^i),$$

which are Lagrangian equations of motion for the Newtonian particle.

2.2 Basic Elastic Continuum

In this section we will give a brief review of classical 3D Euclidean linear theory of elastic soft bodies (soft tissues, or continua; see, e.g. [2]) – to be used extensively in the subsequent chapters. We will adopt standard Euclidean (or, Cartesian) tensor notation with *Cartesian coordinates* $\mathbf{x} = x_i = (x, y, z)$ and all indices ($i, j, k, l, m = 1, \dots, 3$) being subscripts. We will be using *Einstein summation convention* of summing over repeated indices, coma denoting partial derivatives (e.g., $u_{i,j} \equiv \partial_j u_i = \partial u_i / \partial x_j$, etc.) and, as before, the overdot representing the time derivative (e.g., $\dot{u}_i \equiv du_i / dt$).

The main infinitesimal tensor fields of standard linear elasticity are:

(i) *body force vector* $\mathbf{f} = f_i = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}$, representing a 3-axial Newtonian force²

per unit 3D-volume,

(ii) *displacement vector* $\mathbf{u} = u_i = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$,³

with its gradient tensor $\nabla \mathbf{u} = u_{i,j} = \begin{bmatrix} \partial_x u_x \\ \partial_y u_y \\ \partial_z u_z \end{bmatrix}$,

(iii) *strain tensor* $\mathbf{e} = e_{ik} = \begin{bmatrix} e_{xx} & e_{xy} & e_{xz} \\ e_{yx} & e_{yy} & e_{yz} \\ e_{zx} & e_{zy} & e_{zz} \end{bmatrix}$,⁴ derived from the displacement gra-

dient tensor $\nabla \mathbf{u} = u_{i,j}$, together with its associated *deformation tensor*:

$$\boldsymbol{\gamma} = \gamma_{ik} = \begin{bmatrix} \gamma_{xx} & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & \gamma_{yy} & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & \gamma_{zz} \end{bmatrix},$$

(iv) *Cauchy's stress tensor* $\boldsymbol{\sigma} = \sigma_{ik} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}$,⁵ that has physical dimen-

sion of force/area,

²Recall that *Newton's fundamental equation of force*: $\mathbf{f} = \dot{\mathbf{p}} = m\mathbf{a} = m\dot{\mathbf{v}} = m\ddot{\mathbf{x}}$, states that the application of the force vector \mathbf{f} to a particle \bullet of mass m , causes \bullet to move with the momentum $\mathbf{p} = m\mathbf{v}$, acceleration $\mathbf{a} = \ddot{\mathbf{x}}$ and velocity $\mathbf{v} = \dot{\mathbf{x}}$ in the direction \mathbf{x} .

³Note that in a more general, nonlinear Riemannian elasticity, the displacement vector is defined as the *deformation covector* (i.e., one-form): $\mathbf{u} = u_i dx^i$.

⁴In Riemannian elasticity, there are actually two strain tensors: the *Cauchy-Green strain tensor*, an infinitesimal tensor field generated during deformation, given by: $e_{ik}^{\text{CG}} = g_{ik} dx^i dx^k$, and the relative, or *Green-Lagrange strain tensor*, measuring the metric-change between the undeformed and deformed states, given by: $e_{ik}^{\text{GL}} = \frac{1}{2}(g_{ik} - \delta_{ik}) dx^i dx^k$.

⁵In case of large (or, finite) deformations, the Cauchy stress tensor generalizes to the (first and second) *Piola-Kirchhoff stress tensors*.

- (v) *elasticity tensor* $\mathbf{E} = E_{ijkl}$, a fourth-order tensor that has physical dimension of $\text{stress} = \frac{\text{force}}{\text{area}}$, and
 (vi) *mass-density scalar* ρ .

The main principle of linear elasticity can be formulated as: the stress $\boldsymbol{\sigma}$ applied to an elastic material produces the proportional strain (or, deformation) \mathbf{e} , where the proportionality factor is given by the elasticity tensor \mathbf{C} . Both the stress and strain tensors are required to be *symmetric* (having 6 independent components out of total 9). This is *Cauchy's second law of motion*: $\sigma_{ik} = \sigma_{ki}$, which implies the following symmetries of the elasticity tensor: $E_{ijkl} = E_{klij} = E_{jikl} = E_{ijlk}$ (which has 21 independent components out of total 81; see [3, 7] and the references therein).

The Euclidean elasticity is based on the following three relations (given both in standard vector and in Cartesian tensor notation):

Strain-displacement relations: $\mathbf{e} = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T]$, or in tensor components:

$$e_{ik} = \frac{1}{2} (u_{i,k} + u_{k,i});$$

Newtonian continuum equations of motion: $\nabla \cdot \boldsymbol{\sigma} + \mathbf{f} = \rho \ddot{\mathbf{u}}$, also called the *momentum law*, in components given by: $\sigma_{ik,k} + f_i = \rho \ddot{u}_i$; if the material continuum is in static equilibrium, this reduces to *Cauchy's first law of motion*:

$$\sigma_{ik,k} + f_i = 0;$$

Constitutive stress-strain relations: $\boldsymbol{\sigma} = \mathbf{C} \mathbf{e}$, in components: $\sigma_{ik} = E_{iklm} e_{lm}$, which is the generalized *Hooke's law* for a homogeneous anisotropic body.

The above elasticity relations simplify (significantly) in the special case of *isotropic media*,⁶ where the stiffness tensor can be written in terms of the *bulk incompressibility modulus* B and the *shear rigidity modulus* (or, Lamé's second parameter) μ , as:

$$E_{ijkl} = B \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk} - \frac{2}{3} \delta_{ij} \delta_{kl}).$$

Now, the constitutive stress-strain relations can be written in terms of *Lamé's first parameter* λ , as:

$$\sigma_{ij} = \lambda \delta_{ij} e_{kk} + 2\mu e_{ij},$$

where $\text{Tr}(\mathbf{e}) = e_{kk}$ is the trace of the strain tensor, while the strain can be written in terms of *Poisson's ratio* ν and *Young's elasticity modulus* $E = \frac{\text{tensile stress}}{\text{tensile strain}}$, as:

$$e_{ij} = \frac{1}{2\mu} \sigma_{ij} - \frac{\nu}{E} \delta_{ij} \sigma_{kk} = \frac{1}{E} [(1 + \nu) \sigma_{ij} - \nu \delta_{ij} \sigma_{kk}].$$

⁶Physical properties of isotropic media are independent of directions in the 3D Euclidean space.

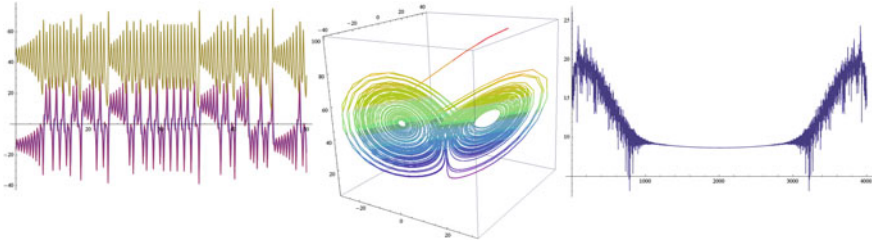


Fig. 2.1 Simulation of the Lorenz attractor: time series (*left*), phase space (*middle*) and FFT spectrum (*right*)

2.3 Basic Dynamical Simulations

In this section we present the basic symbolic derivations and numerical simulations to be used in subsequent chapters.

2.3.1 Basic Attractor Dynamics Simulation

Here we present a *Mathematica*[®] simulator for dynamics of the four well-known three-parameter attractor systems, which exhibit chaotic behavior for certain values of their parameters. The implemented systems are (see, e.g. [4])⁷:

Lorenz attractor (see Fig. 2.1), a temporal dynamical system representing a truncated version of the vector *Navier-Stokes equation*, defined by the following ODEs:

$$\dot{x} = a(y - x), \quad \dot{y} = x(b - z) - y, \quad \dot{z} = xy - cz,$$

where (a, b, c) are the constant parameters with specific chaotic ranges:
 $a \in [16, 20]$, $b \in [45, 56]$, $c \in [1, 6]$;

Rossler attractor (see Fig. 2.2), defined by the following ODEs:

$$\dot{x} = -y - z, \quad \dot{y} = x + ay, \quad \dot{z} = b + z(x - c),$$

where $a \in [0.15, 0.3]$, $b \in [0.15, 0.4]$, $c \in [1, 10]$;

⁷We actually present a generic simulator with three more nameless attractor systems, to demonstrate how easy it is to extend this simulator for other applications.

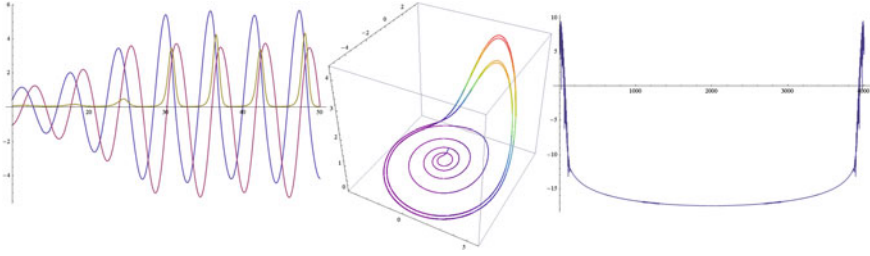


Fig. 2.2 Simulation of the Rossler attractor: time series (*left*), phase space (*middle*) and FFT spectrum (*right*)

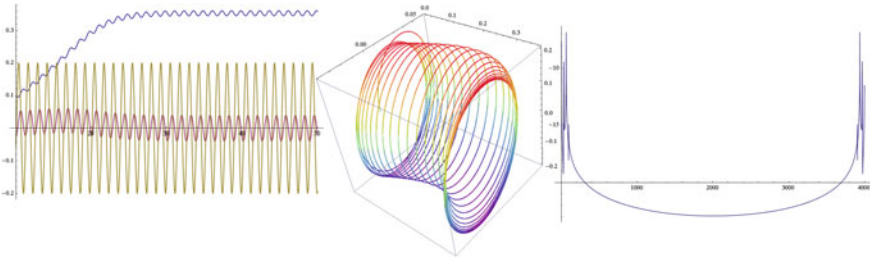


Fig. 2.3 Simulation of the Duffing oscillator with harmonic forcing: time series (*left*), phase space (*middle*) and FFT spectrum (*right*)

Duffing oscillator with harmonic forcing (see Fig. 2.3), defined by the following ODEs:

$$\dot{x} = y, \quad \dot{y} + a\dot{x} - bx + cx^3 = z, \quad \dot{z} = \cos(5t),$$

where $a \in [0, 3]$, $b \in [0, 1]$, $c \in [1, 3]$;

Van der Pol oscillator with harmonic forcing (see Fig. 2.4), defined by the following ODEs:

$$\dot{x} = y, \quad \dot{y} - a(1 - 4bx^2)y + c^2x = z, \quad \dot{z} = \cos(5t),$$

where: $a \in [1, 9]$, $b \in [2, 9]$, $c \in [1, 9]$;

All systems are simulated in the time range: $t \in [0, 50]$. The simulator gives outputs in three different forms: (i) time series: $(x(t), y(t), z(t))$, (ii) X – Y – Z phase space, and (iii) FFT spectrum. It is based on *Mathematica*'s commands *Manipulate* (designed for simulation experiments) and *NDSolve* (a powerful numerical solver/integrator of differential equations, both ODEs and PDEs). Here is the full code for the simulator:

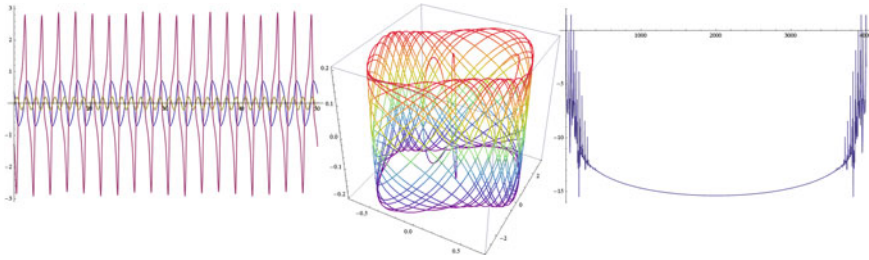


Fig. 2.4 Simulation of the Van der Pol oscillator with harmonic forcing: time series (*left*), phase space (*middle*) and FFT spectrum (*right*)

```

Manipulate[Module[{Lorenz,Rossler,ForcedDuffing,ForcedVanDerPol,
Another1,Another2,Another3,sol,x,y,z,t,pl1,pl2,pl3,data,fdata},

Lorenz={ {x'[t]==a (y[t]-x[t]),y'[t]==x[t] (b-z[t])-y[t],z'[t]==x[t] y[t]-c z[t],
x[0]==y[0]==20,z[0]==100} };

Rossler={ {x'[t]==-y[t]-z[t],y'[t]==x[t]+α 1 y[t],z'[t]==β 1+z[t] (x[t]-γ 1),
x[0]==0.2,y[0]==0.3,z[0]==0.5} };

ForcedDuffing={ {x'[t]==y[t],y'[t]+α 2 x'[t]-β 2 x[t]+γ 2 x[t]3==z[t],
z'[t]==Cos[5t],x[0]==y[0]==z[0]==0} };

ForcedVanDerPol={ {x'[t]==y[t],y'[t]-α 3 (1-4 β 3 x[t]2) y[t]+γ 32 x[t]==z[t],
z'[t]==Cos[5t],x[0]==y[0]==z[0]==0} };

Another1={ {x'[t]==a 1 x[t]+y[t] z[t],y'[t]==b 1 y[t]-x[t] z[t],
z'[t]==c 1 z[t]+x[t] y[t],x[0]==-1,y[0]==1,z[0]==2} };

Another2={ {x'[t]==-y[t]-z[t],y'[t]==x[t]-0.5 a 2 y[t],
z'[t]==0.1+Abs[b 2]+x[t] z[t]+10 c 2 z[t],x[0]==y[0]==-1.2,z[0]==1.2} };

Another3={ {x'[t]==a 3 y[t] z[t]2, y'[t]==Abs[b 3] x[t] z[t],
z'[t]==c 3 x[t] y[t]2,x[0]==1,y[0]==2,z[0]==1.8} };

sol=QuietNDSolve[Lorenz,{x[t],y[t],z[t]},{t,0,50},MaxSteps→ ∞]; (*Eqs.*)

data=Transpose[Table[Evaluate[x[t]*y[t]*z[t]/.sol],{t,10,50,0.01}]];

fdata=Log[Abs[Fourier[data]]2]; (* FFT spectrum defined here *)

pl1=Plot[Evaluate[{x[t],y[t],z[t]}/.sol],{t,10,50},PlotRange→ All,

```

```
PlotStyle→AbsoluteThickness[1.5],ImageSize→{500,350}];
```

```
p12=ParametricPlot3D[Evaluate[{x[t],y[t],z[t]}/.sol],{t,0,50},
BoxRatios→{1,1,1},PlotRange→All,ColorFunction→"Rainbow",
PlotStyle→AbsoluteThickness[1.5],ImageSize→{500,400}];
```

```
p13=ListLinePlot[fdata,PlotRange→All,PlotStyle→
AbsoluteThickness[1.1],ImageSize→{500,350}];
```

```
Which[Analysis=="Time series",Show[p11],Analysis=="Phase space",
Show[p12],Analysis=="FFTspectrum",Show[p13]],
Delimiter,Style["Lorenz",10],
{{a,16,a},16,20,0.1,ImageSize→Small,Appearance→"Labeled"},
{{b,45.92,b},45,56,1,ImageSize→Small,Appearance→"Labeled"},
{{c,4,c},1,0.6,0.1,ImageSize→Small,Appearance→"Labeled"},
Delimiter,Style["Rossler",10],
{{α1,0.2,a},0.15,0.3,0.01,ImageSize→Small,Appearance→"Labeled"},
{{β1,0.2,b},0.15,0.4,0.01,ImageSize→Small,Appearance→"Labeled"},
{{γ1,3.0,c},1,10,0.1,ImageSize→Small,Appearance→"Labeled"},
Delimiter,Style["ForcedDuffing",10],
{{α2,1.5,a},0,3,0.01,ImageSize→Small,Appearance→"Labeled"},
{{β2,0.25,b},0,1,0.01,ImageSize→Small,Appearance→"Labeled"},
{{γ2,2,c},1,3,0.01,ImageSize→Small,Appearance→"Labeled"},
Delimiter,Style["ForcedVanDerPol",10],
{{α3,3,a},1,9,0.1,ImageSize→Small,Appearance→"Labeled"},
{{β3,2,b},2,9,0.1,ImageSize→Small,Appearance→"Labeled"},
{{γ3,3,c},1,9,0.1,ImageSize→Small,Appearance→"Labeled"},
Delimiter,Style["Another1",10],
{{a1,-0.4,a},-1,-0.25,0.001,ImageSize→Small,Appearance→"Labeled"},
{{b1,0.3,b},-1,1,0.001,ImageSize→Small,Appearance→"Labeled"},
{{c1,-0.1,c},-1,0,0.001,ImageSize→Small,Appearance→"Labeled"},
Delimiter,Style["Another2",10],
{{a2,-0.4,a},-1,-0.25,0.001,ImageSize→Small,Appearance→"Labeled"},
{{b2,0.3,b},-1,1,0.001,ImageSize→Small,Appearance→"Labeled"},
{{c2,-0.8,c},-1,0,0.001,ImageSize→Small,Appearance→"Labeled"},
Delimiter,Style["Another3",10],
{{a3,-0.4,a},-1,-0.25,0.001,ImageSize→Small,Appearance→"Labeled"},
{{b3,-0.7,b},-1,1,0.001,ImageSize→Small,Appearance→"Labeled"},
{{c3,-0.8,c},-1,0,0.001,ImageSize→Small,Appearance→"Labeled"},
Delimiter,{Analysis,"Time series"},
{"Time series","Phase space","FFT spectrum"}},
TrackedSymbols→Manipulate]
```

2.3.2 Action Principle and Basic Variational Derivations

In this subsection we give some introductory examples of using the action principle and variational methods in *Mathematica* to derive Newtonian equations of motion for some common dynamical systems used in biomechanics – from their kinetic and potential energies.

Harmonic Oscillator

We start with a simple and ubiquitous harmonic oscillator. Given the input data: mass m , displacement $x(t)$, velocity $x'(t)$ and spring constant k , the following code gives oscillator's equation of motion with acceleration $x''(t)$ and initial conditions ($x(0) = 0$, $x'(0) = 1$):

```
<<VariationalMethods`
Kinetic energy defined:
  KE =  $\frac{1}{2} m x'[t]^2$ ;
Potential energy defined:
  PE =  $\frac{1}{2} k x[t]^2$ ;
Lagrangian calculated:
  Lagrn = KE - PE;
Euler-Lagrange equation computed:
  DifEq = EulerEquations[Lagrn, x[t], t];
Output printout:
  Print["Lagrangian of the system = ", Lagrn];
  Print["Equation of motion: ", DifEq];
Lagrangian of the system =  $\frac{1}{2} m x'(t)^2 - \frac{1}{2} k x(t)^2$ 
```

Equation of motion: $m x''(t) - k x(t) = 0$

Initial conditions:

```
IC = {x[0] == 0, x'[0] == 1};
```

Analytical solution:

```
sol = DSolve[ODE, IC, x[t], t]
```

```
sol =  $\left\{ \left\{ x(t) \rightarrow \frac{\sqrt{m}}{\sqrt{k}} \sin \left( \frac{\sqrt{k} t}{\sqrt{m}} \right) \right\} \right\}$ 
```

Pendulum

Next, we compute a simple pendulum. Given the input data: mass m , length L , angle $q(t)$ and angular velocity $q'(t)$ the following code gives pendulum's equation of motion with acceleration $q''(t)$ and its analytical solution for the case of passing through the equilibrium $q(0) = 0$ with angular velocity $q'(0) = 1$:

```
<<VariationalMethods`
Kinetic energy defined:
  KE = 1/2 m q'[t]^2;
Potential energy defined:
  PE = m g L Cos[q[t]];
Lagrangian calculated:
  Lagrn = KE - PE;
Euler-Lagrange equation computed:
  DifEq = EulerEquations[Lagrn, q[t], t];
Output printout:
  Print["Lagrangian of the system = ", Lagrn];
  Print["Equation of motion: ", DifEq];

Lagrangian of the system =  $gLm\text{Cos}[q[t]] + \frac{1}{2}mL^2q'(t)$ 

Equation of motion:  $-Lm(g\sin(q(t)) + Lq''(t)) = 0$ 

Initial conditions:
IC = {q[0] == 0, q'[0] == 1};

Analytical solution:
sol = DSolve[ODE, IC, q[t], t]

sol =  $\left\{\left\{q(t) \rightarrow 2\text{JacobiAmplitude}\left(\frac{t}{2}, \frac{4g}{L}\right)\right\}\right\}$ 

Series expansion (first 10 terms) of JacobiAmplitudes:
Series[2 JacobiAmplitude[t/2, (4 g)/L], t, 0, 10]

sol =  $t - \frac{gt^3}{6L} + \frac{t^5(g^2+gL)}{120L^2} + \frac{t^7(-g^3-11g^2L-gL^2)}{5040L^3} + \frac{t^9(g^4+102g^3L+57g^2L^2+gL^3)}{362880L^4}$ 
```

1D Particle

Now, we compute a simple particle on a line. Given the input data: mass m , displacement $x(t)$, velocity $x'(t)$ and gravity g , the following code gives particle's equation of motion with acceleration $x''(t)$:

```
<<VariationalMethods'
Kinetic energy defined:
  KE = 1/2 m x'[t]^2;
Potential energy defined:
  PE = m g Cos[x[t]];
Lagrangian calculated:
  Lagrn = KE - PE;
Euler-Lagrange equation computed:
  DifEq = EulerEquations[Lagrn, x[t], t];
Output printout:
  Print["Lagrangian of the system = ", Lagrn];
  Print["Equation of motion: ", DifEq];

Lagrangian of the system = 1/2 m x'(t)^2 - m g cos(x(t))

Equation of motion: m(g sin(x(t)) - x''(t)) = 0
```

3D Particle

Now, we compute a simple particle in 3D. Given the input data: masses m_i , displacements $x_i(t)$, velocities $x'_i(t)$ and gravity g , the following code gives particle's 3D equations of motion with accelerations $x''_i(t)$:

```
<<VariationalMethods'
Kinetic energy defined:
  KE = 1/2 Sum_{i=1}^3 m_i x'_i[t]^2;
Potential energy defined:
  PE = g Sum_{i=1}^3 m_i Cos[x_i[t]];
Lagrangian calculated:
  Lagrn = KE - PE;
Euler-Lagrange equations computed:
  DifEq = EulerEquations[Lagrn, x[t], t];
Output printout:
  Print["Lagrangian of the system = ", Lagrn];
  Print["Equations of motion = ", DifEq];

Lagrangian of the system =
  1/2 (m_1((x_1)'(t))^2 + m_2((x_2)'(t))^2
  + m_3((x_3)'(t))^2) - g(m_1 cos(x_1(t))
  + m_2 cos(x_2(t)) + m_3 cos(x_3(t)))

Equations of motion =
  {m_1(g sin(x_1(t)) - (x_1)''(t)) = 0,
   m_2(g sin(x_2(t)) - (x_2)''(t)) = 0,
   m_3(g sin(x_3(t)) - (x_3)''(t)) = 0}
```

Harmonic Chain

Given the input data: masses m_i , displacements $x_i(t)$, velocities $x'_i(t)$ and spring constants k_i , the following code gives equations of motion of a 7D harmonic chain:

```
<<VariationalMethods'
Kinetic energy defined:
  KE = 1/2 * Sum_{i=1}^n m_i x'_i[t]^2;
Potential energy defined:
  PE = 1/2 * Sum_{i=1}^n k_i (x_i[t] - x_{i-1}[t])^2;
Lagrangian calculated:
  Lagrn = KE - PE;
Euler-Lagrange equation computed:
  DifEq = EulerEquations[Lagrn, x[t], t];
Output printout:
  Print["Lagrangian of the system = ", Lagrn];
  Print["Equation of motion: ", DifEq];
```

Lagrangian of the system =

$$\frac{1}{2}(m_1((x_1)'(t))^2 + m_2((x_2)'(t))^2 + m_3((x_3)'(t))^2 + m_4((x_4)'(t))^2 + m_5((x_5)'(t))^2 + m_6((x_6)'(t))^2 + m_7((x_7)'(t))^2) + \frac{1}{2}(-k_1(x_1(t) - x_0(t))^2 - k_2(x_2(t) - x_1(t))^2 - k_3(x_3(t) - x_2(t))^2 - k_4(x_4(t) - x_3(t))^2 - k_5(x_5(t) - x_4(t))^2 - k_6(x_6(t) - x_5(t))^2 - k_7(x_7(t) - x_6(t))^2)$$

Equations of motion =

$$\{-m_1(x_1)''(t) + k_1(x_0(t) - x_1(t)) + k_2(x_2(t) - x_1(t)) = 0, \\ -m_2(x_2)''(t) + k_2(x_1(t) - x_2(t)) + k_3(x_3(t) - x_2(t)) = 0, \\ -m_3(x_3)''(t) + k_3(x_2(t) - x_3(t)) + k_4(x_4(t) - x_3(t)) = 0, \\ -m_4(x_4)''(t) + k_4(x_3(t) - x_4(t)) + k_5(x_5(t) - x_4(t)) = 0, \\ -m_5(x_5)''(t) + k_5(x_4(t) - x_5(t)) + k_6(x_6(t) - x_5(t)) = 0, \\ -m_6(x_6)''(t) + k_6(x_5(t) - x_6(t)) + k_7(x_7(t) - x_6(t)) = 0, \\ k_7(x_6(t) - x_7(t)) - m_7(x_7)''(t) = 0\}$$

Basic Kinetic chain

Given the input data: masses m_i , displacements $x_i(t)$, velocities $x'_i(t)$ and spring constants k_i , the following code gives equations of motion of a 9D kinetic chain with the quartic (4th order) potential:

```
<<VariationalMethods'; n = 9;
Kinetic energy defined:
  KE = 1/2 * Sum_{i=1}^n m_i^2 (x_i'[t] - x'_{i-1}[t])^2;
Potential energy defined:
  PE = Sum_{i=1}^n k_i (x_i[t] - x_{i-1}[t])^4;
Lagrangian calculated:
  Lagrn = KE - PE;
```

Euler-Lagrange equations computed:

DifEq = EulerEquations[Lagr, x[t], t];

Output printout:

Print["Lagrangian of the system = ", Lagr];

Print["Equations of motion = ", DifEq];

Lagrangian of the system =

$$\begin{aligned} & \frac{1}{2}(m_1^2((x_1)'(t) - (x_0)'(t))^2 + m_2^2((x_2)'(t) - (x_1)'(t))^2 + m_3^2((x_3)'(t) - (x_2)'(t))^2 + \\ & m_4^2((x_4)'(t) - (x_3)'(t))^2 + m_5^2((x_5)'(t) - (x_4)'(t))^2 + m_6^2((x_6)'(t) - (x_5)'(t))^2 + \\ & m_7^2((x_7)'(t) - (x_6)'(t))^2 + m_8^2((x_8)'(t) - (x_7)'(t))^2 + m_9^2((x_9)'(t) - (x_8)'(t))^2) - \\ & k_1(x_1(t) - x_0(t))^4 - k_2(x_2(t) - x_1(t))^4 - k_3(x_3(t) - x_2(t))^4 - k_4(x_4(t) - x_3(t))^4 - \\ & k_5(x_5(t) - x_4(t))^4 - k_6(x_6(t) - x_5(t))^4 - k_7(x_7(t) - x_6(t))^4 - k_8(x_8(t) - x_7(t))^4 - \\ & k_9(x_9(t) - x_8(t))^4 \end{aligned}$$

Equations of motion =

$$\begin{aligned} & \{ \frac{1}{2}(2m_1^2((x_0)''(t) - (x_1)''(t)) + 2m_2^2((x_2)''(t) - (x_1)''(t)) + 8k_1(x_0(t) - x_1(t))^3 + 8k_2(x_2(t) - x_1(t))^3) = 0, \\ & \frac{1}{2}(2m_2^2((x_1)''(t) - (x_2)''(t)) + 2m_3^2((x_3)''(t) - (x_2)''(t)) + 8k_2(x_1(t) - x_2(t))^3 + 8k_3(x_3(t) - x_2(t))^3) = 0, \\ & \frac{1}{2}(2m_3^2((x_2)''(t) - (x_3)''(t)) + 2m_4^2((x_4)''(t) - (x_3)''(t)) + 8k_3(x_2(t) - x_3(t))^3 + 8k_4(x_4(t) - x_3(t))^3) = 0, \\ & \frac{1}{2}(2m_4^2((x_3)''(t) - (x_4)''(t)) + 2m_5^2((x_5)''(t) - (x_4)''(t)) + 8k_4(x_3(t) - x_4(t))^3 + 8k_5(x_5(t) - x_4(t))^3) = 0, \\ & \frac{1}{2}(2m_5^2((x_4)''(t) - (x_5)''(t)) + 2m_6^2((x_6)''(t) - (x_5)''(t)) + 8k_5(x_4(t) - x_5(t))^3 + 8k_6(x_6(t) - x_5(t))^3) = 0, \\ & \frac{1}{2}(2m_6^2((x_5)''(t) - (x_6)''(t)) + 2m_7^2((x_7)''(t) - (x_6)''(t)) + 8k_6(x_5(t) - x_6(t))^3 + 8k_7(x_7(t) - x_6(t))^3) = 0, \\ & \frac{1}{2}(2m_7^2((x_6)''(t) - (x_7)''(t)) + 2m_8^2((x_8)''(t) - (x_7)''(t)) + 8k_7(x_6(t) - x_7(t))^3 + 8k_8(x_8(t) - x_7(t))^3) = 0, \\ & \frac{1}{2}(2m_8^2((x_7)''(t) - (x_8)''(t)) + 2m_9^2((x_9)''(t) - (x_8)''(t)) + 8k_8(x_7(t) - x_8(t))^3 + 8k_9(x_9(t) - x_8(t))^3) = 0, \\ & m_9^2((x_8)''(t) - (x_9)''(t)) + 4k_9(x_8(t) - x_9(t))^3 = 0 \} \end{aligned}$$

Linear String (Wave) Equation

Given the transversal string displacement $u[x, t]$, the following code gives the standard 1D wave equation:

<<VariationalMethods‘

Kinetic energy defined:

$$KE = \frac{1}{2} D[u[x, t], t]^2;$$

Potential energy defined:

$$PE = \frac{1}{2} D[u[x, t], x]^2;$$

Lagrangian calculated:

$$Lagr = KE - PE;$$

Euler-Lagrange equation computed:

$$DifEq = EulerEquations[Lagr, u[x, t], x, t];$$

Output printout:

Print["Lagrangian of the system = ", Lagr];

Print["Equation of motion: ", DifEq];

$$\text{Lagrangian of the system} = \frac{1}{2} u^{(0,1)}(x, t)^2 - \frac{1}{2} u^{(1,0)}(x, t)^2 \iff \frac{1}{2} u_t(x, t)^2 - \frac{1}{2} u_x(x, t)^2$$

$$\text{Equation of motion:} \\ u^{(2,0)}(x, t) - u^{(0,2)}(x, t) = 0 \iff u_{xx}(x, t) = u_{tt}(x, t)$$

Nonlinear String (Sine-Gordon) Equation

Given the transversal displacement $u[x, t]$ of a nonlinear string, the following code gives the 1D Sine-Gordon equation:

```
<<VariationalMethods`
Kinetic energy defined:
  KE = 1/2 D[u[x, t], t]^2;
Potential energy defined:
  PE = 1/2 D[u[x, t], x]^2 + Cos[u[x, t]];
Lagrangian calculated:
  Lagrn = KE - PE;
Euler-Lagrange equation computed:
  DifEq = EulerEquations[Lagrn, u[x, t], x, t];
Output printout:
  Print["Lagrangian of the system = ", Lagrn];
  Print["Equation of motion: ", DifEq];
```

$$\text{Lagrangian of the system} = \frac{1}{2} u^{(0,1)}(x, t)^2 - \frac{1}{2} u^{(1,0)}(x, t)^2 - \cos[u(x, t)] \iff \frac{1}{2} u_t(x, t)^2 - \frac{1}{2} u_x(x, t)^2 - \cos[u(x, t)]$$

$$\text{Equation of motion:} \\ u^{(2,0)}(x, t) - u^{(0,2)}(x, t) + \sin[u(x, t)] = 0 \iff u_{tt}(x, t) = u_{xx}(x, t) - \sin[u(x, t)]$$

3D Laplace Equation

The Laplace PDE is a textbook model for various stationary fields: electrostatic, magnetostatic, thermostatic, etc. Here we give a variational derivation for the 3D Laplace equation from a given vector Lagrangian in Cartesian coordinates:

```
<<VariationalMethods`
Lagrangian defined:
  Lagrn = 1/2 Grad[u[x, y, z], x, y, z].Grad[u[x, y, z], x, y, z];
Euler-Lagrange equation computed:
  DifEq = EulerEquations[Lagrn, u[x, y, z], x, y, z];
Output printout:
```


Print["Lagrangian of the system =", Lagrn];

Print["Equation of motion: ", DifEq];

Lagrangian of the system =

$$\frac{1}{4} [u^{(0,0,1)}(x, y, z)^2 + u^{(0,1,0)}(x, y, z)^2 + u^{(1,0,0)}(x, y, z)^2]^2$$

$$\iff \frac{1}{4} [u_x(x, y, z)^2 + u_y(x, y, z)^2 + u_z(x, y, z)^2]^2$$

Equation of motion:

$$-u^{(0,0,2)}(x, y, z) - u^{(0,2,0)}(x, y, z) - u^{(2,0,0)}(x, y, z) = 0$$

$$\iff u_{xx}(x, y, z) + u_{yy}(x, y, z) + u_{zz}(x, y, z) = 0$$

2.3.3 Basic Vector Calculus Implementation

In this subsection we give some introductory examples of using vector calculus in *Mathematica*[®], to be utilized in the subsequent chapters.

Gradient of a 3D scalar field

Cartesian coordinates:

Grad[u[x, y, z], x, y, z] gives:

$$\{u^{(1,0,0)}(x, y, z), u^{(0,1,0)}(x, y, z), u^{(0,0,1)}(x, y, z)\}$$

$$\iff [u_x(x, y, z), u_y(x, y, z), u_z(x, y, z)]$$

Cylindrical coordinates:

Grad[u[r, θ, z], r, θ, z, "Cylindrical"] gives:

$$\left\{ u^{(1,0,0)}(r, \theta, z), \frac{u^{(0,1,0)}(r, \theta, z)}{r}, u^{(0,0,1)}(r, \theta, z) \right\}$$

$$\iff [u_x(r, \theta, z), \frac{1}{r}u_\theta(r, \theta, z), u_z(r, \theta, z)]$$

Spherical coordinates:

Grad[u[r, θ, φ], r, θ, φ, "Spherical"] gives:

$$\left\{ u^{(1,0,0)}(r, \theta, \phi), \frac{u^{(0,1,0)}(r, \theta, \phi)}{r}, \frac{\csc(\theta)u^{(0,0,1)}(r, \theta, \phi)}{r} \right\}$$

$$\iff \left[u_x(r, \theta, \phi), \frac{1}{r}u_\theta(r, \theta, \phi), \frac{1}{r \sin(\theta)}u_\phi(r, \theta, \phi) \right]$$

Divergence of a 3D vector field

Cartesian coordinates:

Div[u[x, y, z], v[x, y, z], w[x, y, z], x, y, z] gives:

$$u^{(1,0,0)}(x, y, z) + v^{(0,1,0)}(x, y, z) + w^{(0,0,1)}(x, y, z)$$

$$\iff [u_x(x, y, z), v_y(x, y, z), w_z(x, y, z)]$$

Cylindrical coordinates:

$\text{Div}[u[r, \theta, z], v[r, \theta, z], w[r, \theta, z], r, \theta, z, \text{"Cylindrical"}]$ gives:

$$u^{(1,0,0)}(r, \theta, z) + \frac{u(r, \theta, z) + v^{(0,1,0)}(r, \theta, z)}{r} + w^{(0,0,1)}(r, \theta, z) \\ \iff \left[u_r(r, \theta, z) + \frac{u(r, \theta, z) + v_\theta(r, \theta, z)}{r} + w_z(r, \theta, z) \right]$$

Spherical coordinates:

$\text{Div}[u[r, \theta, \phi], v[r, \theta, \phi], w[r, \theta, \phi], r, \theta, \phi, \text{"Spherical"}]$ gives:

$$u^{(1,0,0)}(r, \theta, \phi) + \frac{u(r, \theta, \phi) + v^{(0,1,0)}(r, \theta, \phi)}{r} + \frac{\csc(\theta) (\sin(\theta)u(r, \theta, \phi) + \cos(\theta)v(r, \theta, \phi) + w^{(0,0,1)}(r, \theta, \phi))}{r} \\ \iff \left[u_r(r, \theta, \phi) + \frac{u(r, \theta, \phi) + v_\theta(r, \theta, \phi)}{r} + \frac{\sin(\theta)u(r, \theta, \phi) + \cos(\theta)v(r, \theta, \phi) + w_\phi(r, \theta, \phi)}{r \sin(\theta)} \right]$$

Curl of a 3D Vector Field

Cartesian coordinates:

$\text{Curl}[u[x, y, z], v[x, y, z], w[x, y, z], x, y, z]$ gives:

$$w^{(0,1,0)}(x, y, z) - v^{(0,0,1)}(x, y, z), \\ u^{(0,0,1)}(x, y, z) - w^{(1,0,0)}(x, y, z), \\ v^{(1,0,0)}(x, y, z) - u^{(0,1,0)}(x, y, z) \\ \iff \\ w^{(0,1,0)}(x, y, z) - v^{(0,0,1)}(x, y, z), \\ u^{(0,0,1)}(x, y, z) - w^{(1,0,0)}(x, y, z), \\ v^{(1,0,0)}(x, y, z) - u^{(0,1,0)}(x, y, z)$$

Cylindrical coordinates:

$\text{Curl}[u[r, \theta, z], v[r, \theta, z], w[r, \theta, z], r, \theta, z, \text{"Cylindrical"}]$ gives:

$$\frac{w^{(0,1,0)}(r, \theta, z)}{r} - v^{(0,0,1)}(r, \theta, z), \\ u^{(0,0,1)}(r, \theta, z) - \frac{w^{(1,0,0)}(r, \theta, z)}{r}, \\ v^{(1,0,0)}(r, \theta, z) - \frac{u^{(0,1,0)}(r, \theta, z) - v(r, \theta, z)}{r} \\ \iff \\ \frac{1}{r} w_\theta(r, \theta, z) - v_z(r, \theta, z), \\ u_z(r, \theta, z) - w_r(r, \theta, z), \\ v_r(r, \theta, z) - \frac{1}{r} [u_\theta(r, \theta, z) - v(r, \theta, z)]$$

Spherical coordinates:

$\text{Curl}[u[r, \theta, \phi], v[r, \theta, \phi], w[r, \theta, \phi], r, \theta, \phi, \text{"Spherical"}]$ gives:

$$\frac{w^{(0,1,0)}(r, \theta, \phi)}{r} - \frac{\csc(\theta) (v^{(0,0,1)}(r, \theta, \phi) - \cos(\theta)w(r, \theta, \phi))}{r}, \\ \frac{\csc(\theta) (u^{(0,0,1)}(r, \theta, \phi) - \sin(\theta)w(r, \theta, \phi))}{r} - w^{(1,0,0)}(r, \theta, \phi), \\ v^{(1,0,0)}(r, \theta, \phi) - \frac{u^{(0,1,0)}(r, \theta, \phi) - v(r, \theta, \phi)}{r} \\ \iff \\ \frac{1}{r} w_\theta(r, \theta, \phi) - \frac{1}{r \sin(\theta)} [v_\phi(r, \theta, \phi) - w(r, \theta, \phi) \cos(\theta)], \\ \frac{1}{r \sin(\theta)} [u_\phi(r, \theta, \phi) - w(r, \theta, \phi) \sin(\theta)] - w_r(r, \theta, \phi),$$

$$v_r(r, \theta, \phi) - \frac{1}{r} [u_\theta(r, \theta, \phi) - v(r, \theta, \phi)]$$

Laplacian of a 3D scalar field

Cartesian coordinates:

Laplacian[u[x, y, z], x, y, z] gives:

$$u^{(0,0,2)}(x, y, z) + u^{(0,2,0)}(x, y, z) + u^{(2,0,0)}(x, y, z) \\ \iff u_{xx}(x, y, z) + u_{yy}(x, y, z) + u_{zz}(x, y, z)$$

Cylindrical coordinates:

Laplacian[u[r, θ, z], r, θ, z, "Cylindrical"] gives:

$$u^{(0,0,2)}(r, \theta, z) + \frac{\frac{u^{(0,2,0)}(r, \theta, z)}{r} + u^{(1,0,0)}(r, \theta, z)}{r} + u^{(2,0,0)}(r, \theta, z) \\ \iff u_{rr}(r, \theta, z) + \frac{1}{r} \left[\frac{1}{r} u_{\theta\theta}(r, \theta, z) + u_r(r, \theta, z) \right] + u_{zz}(r, \theta, z)$$

Spherical coordinates:

Laplacian[u[r, θ, φ], r, θ, φ, "Spherical"] gives:

$$\frac{\frac{u^{(0,2,0)}(r, \theta, \phi)}{r} + u^{(1,0,0)}(r, \theta, \phi)}{r} + u^{(2,0,0)}(r, \theta, \phi) + \\ \frac{\csc(\theta) \left(\sin(\theta) u^{(1,0,0)}(r, \theta, \phi) + \frac{\cos(\theta) u^{(0,1,0)}(r, \theta, \phi)}{r} + \frac{\csc(\theta) u^{(0,0,2)}(r, \theta, \phi)}{r} \right)}{r} \\ \iff u_{rr}(r, \theta, \phi) + \frac{1}{r} \left[\frac{1}{r} u_{\theta\theta}(r, \theta, \phi) + u_r(r, \theta, \phi) \right] + \\ \frac{1}{r \sin(\theta)} \left[u_r(r, \theta, \phi) \sin(\theta) + \frac{1}{r} u_\theta(r, \theta, \phi) \cos(\theta) + \frac{1}{r \sin(\theta)} u_{\phi\phi}(r, \theta, \phi) \right]$$

Gradient of a 3D Vector Field (in Cylindrical Coordinates)

Grad[u[r, θ, z], v[r, θ, z], w[r, θ, z], r, θ, z, "Cylindrical"] gives:

$$\begin{pmatrix} u^{(1,0,0)}(r, \theta, z) & \frac{u^{(0,1,0)}(r, \theta, z) - v(r, \theta, z)}{r} & u^{(0,0,1)}(r, \theta, z) \\ v^{(1,0,0)}(r, \theta, z) & \frac{u(r, \theta, z) + v^{(0,1,0)}(r, \theta, z)}{r} & v^{(0,0,1)}(r, \theta, z) \\ w^{(1,0,0)}(r, \theta, z) & \frac{w^{(0,1,0)}(r, \theta, z)}{r} & w^{(0,0,1)}(r, \theta, z) \end{pmatrix} \\ \iff \begin{bmatrix} u_r(r, \theta, z) & \frac{u_\theta(r, \theta, z) - v(r, \theta, z)}{r} & u_z(r, \theta, z) \\ v_r(r, \theta, z) & \frac{u(r, \theta, z) + v_\theta(r, \theta, z)}{r} & v_z(r, \theta, z) \\ w_r(r, \theta, z) & \frac{w_\theta(r, \theta, z)}{r} & w_z(r, \theta, z) \end{bmatrix}$$

2.3.4 Basic Elastic Continuum Implementation

Using the free-available *Mathematica* packages for tensor calculus and continuum mechanics (see [1]) and adopting Einstein's summation convention over repeated indices, we can derive the following tensor equations of an elastic continuum:

Kinematic relation between the displacement vector u_i and strain tensor e_{ij} :

$$e_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i});$$

Equilibrium condition including the stress tensor σ_{ij} , the acceleration vector \ddot{u}_i (multiplied by the mass-density ρ) and the force vector f_i are:

$$\sigma_{ji,j} = \rho \ddot{u}_i - f_i;$$

Hooke's stress-strain law: with the elasticity tensor E_{ijklm} :

$$\sigma_{ij} = E_{ijklm} e_{lm},$$

which in terms of Lamé's elastic moduli λ and μ simplifies to:

$$\sigma_{ik} = \lambda e_{mm} \delta_{ik} + 2\mu e_{ik}.$$

To start with, the so-called *compatibility conditions* between the strain tensor e_{ij} and the deformation tensor γ_{ij} read:

$$\begin{aligned} \frac{\partial^2 e_y}{\partial z \partial z} + \frac{\partial^2 e_z}{\partial y \partial y} &= \frac{\partial^2 \gamma_{yz}}{\partial y \partial z} \iff e_{y,z,z} + e_{z,y,y} = \gamma_{yz,y,z}, \\ \frac{\partial^2 e_x}{\partial z \partial z} + \frac{\partial^2 e_z}{\partial x \partial x} &= \frac{\partial^2 \gamma_{xz}}{\partial x \partial z} \iff e_{x,z,z} + e_{z,x,x} = \gamma_{xz,x,z}, \\ \frac{\partial^2 e_x}{\partial y \partial y} + \frac{\partial^2 e_y}{\partial x \partial x} &= \frac{\partial^2 \gamma_{xy}}{\partial x \partial y} \iff e_{x,y,y} + e_{y,x,x} = \gamma_{xy,x,y}, \\ 2 \frac{\partial^2 e_z}{\partial x \partial y} + \frac{\partial^2 \gamma_{xy}}{\partial z \partial z} &= \frac{\partial^2 \gamma_{xz}}{\partial y \partial z} + \frac{\partial^2 \gamma_{yz}}{\partial x \partial z} \iff 2e_{z,x,y} + \gamma_{xy,z,z} = \gamma_{xz,y,z} + \gamma_{yz,x,z}, \\ 2 \frac{\partial^2 e_x}{\partial y \partial z} + \frac{\partial^2 \gamma_{yz}}{\partial x \partial x} &= \frac{\partial^2 \gamma_{xy}}{\partial x \partial z} + \frac{\partial^2 \gamma_{xz}}{\partial x \partial y} \iff 2e_{x,y,z} + \gamma_{yz,x,x} = \gamma_{xy,x,z} + \gamma_{xz,x,y}, \\ 2 \frac{\partial^2 e_y}{\partial x \partial z} + \frac{\partial^2 \gamma_{xz}}{\partial y \partial y} &= \frac{\partial^2 \gamma_{xy}}{\partial y \partial z} + \frac{\partial^2 \gamma_{yz}}{\partial x \partial y} \iff 2e_{y,x,z} + \gamma_{xz,y,y} = \gamma_{xy,y,z} + \gamma_{yz,x,y}. \end{aligned}$$

From the action principle perspective, the essential quantity is the *strain energy density*, defined as the half-stress-strain product:

$$\begin{aligned} a &= \frac{1}{2} e_{ij} \sigma_{ij} = \frac{1}{2} (e_{11} \sigma_{11} + 2e_{12} \sigma_{12} + 2e_{13} \sigma_{13} + e_{22} \sigma_{22} + 2e_{23} \sigma_{23} + e_{33} \sigma_{33}) \\ &= \frac{1}{2} (\gamma_{xy} \sigma_{xy} + \gamma_{zx} \sigma_{zx} + \gamma_{yz} \sigma_{yz} + e_x \sigma_x + e_y \sigma_y + e_z \sigma_z). \end{aligned}$$

The *elasticity tensor* $E = E_{ijklm}$ has the following 21 independent components:

$$(E_{1111}, E_{1112}, E_{1113}, E_{1122}, E_{1123}, E_{1133}, E_{1212}, E_{1213}, E_{1222}, E_{1223}, E_{1233}, E_{1313}, E_{1322}, E_{1323}, E_{1333}, E_{2222}, E_{2223}, E_{2233}, E_{2323}, E_{2333}, E_{3333}).$$

These elastic components can be evaluated using Young's modulus $E = \frac{\mu(3\lambda+2\mu)}{\lambda+\mu}$, and Poisson's ratio $\nu = \frac{\lambda}{2(\lambda+\mu)}$, expressed here in terms of Lamé's elastic moduli λ and μ . In this way, we can define:

$$\begin{aligned}
E_{1111} &= \frac{E(\nu-1)}{2\nu^2+\nu-1}, & E_{1112} &= 0, & E_{1113} &= 0, \\
E_{1122} &= -\frac{E\nu}{2\nu^2+\nu-1}, & E_{1123} &= 0, & E_{1133} &= -\frac{E\nu}{2\nu^2+\nu-1}, \\
E_{2222} &= \frac{E(\nu-1)}{2\nu^2+\nu-1}, & E_{2223} &= 0, & E_{1222} &= 0, \\
E_{2233} &= -\frac{E\nu}{2\nu^2+\nu-1}, & E_{1322} &= 0, & E_{1122} &= -\frac{E\nu}{2\nu^2+\nu-1}, \\
E_{3333} &= \frac{E(\nu-1)}{2\nu^2+\nu-1}, & E_{1333} &= 0, & E_{2333} &= 0, \\
E_{1133} &= -\frac{E\nu}{2\nu^2+\nu-1}, & E_{1233} &= 0, & E_{2233} &= -\frac{E\nu}{2\nu^2+\nu-1}, \\
E_{2223} &= 0, & E_{2323} &= \frac{E}{2\nu+2}, & E_{1223} &= 0, & E_{2333} &= 0, & E_{1323} &= 0, & E_{1123} &= 0, \\
E_{1333} &= 0, & E_{1313} &= \frac{E}{2\nu+2}, & E_{1323} &= 0, & E_{1113} &= 0, & E_{1213} &= 0, & E_{1322} &= 0.
\end{aligned}$$

Using the converse relations, defining Lamé's elastic moduli λ and μ in terms of Young's modulus and Poisson's ratio, as:

$$\lambda = -\frac{E\nu}{(\nu-1)(\nu+1)}, \quad \mu = \frac{E}{2(\nu+1)},$$

we can evaluate the Hooke's law: $\sigma_{ij} = E_{ijkl}e_{lm}$ as:

$$\sigma_x = -\frac{E(e_x + \nu e_y)}{\nu^2 - 1}, \quad \sigma_y = -\frac{E(\nu e_x + e_y)}{\nu^2 - 1}, \quad \sigma_{xy} = \frac{E e_{xy}}{\nu + 1}.$$

For more elastic continuum derivations, see *Mathematica* packages listed in [1] and the associated internet links.

References

1. Cabrera, R.: Clifford Algebra, Wolfram Library Archive (2015). <http://library.wolfram.com/infocenter/MathSource/5101/>
2. Fung, F.C.: A First Course in Continuum Mechanics, 3rd edn. Prentice-Hall, Englewood Cliffs (1994)
3. Landau, L.D., Lifshitz, E.M.: Theory of Elasticity, 3rd edn. Butterworth Heinemann, Oxford (1986)
4. Weisstein, E.W.: Lorenz Attractor, Wolfram MathWorld (2015)
5. Wikipedia: Calculus of Variations (2015)
6. Wikipedia: Hamilton's Principle (2015)
7. Wikipedia: Linear Elasticity (2015)

The Evolved Athlete: A Guide for Elite Sport
Enhancement

Ivancevic, T.; Lukman, L.; Gojkovic, Z.; Greenberg, R.;

Greenberg, H.; Jovanovic, B.; Lukman, A.

2017, XXVI, 185 p. 94 illus., 88 illus. in color., Hardcover

ISBN: 978-3-319-57927-6