III Geometric Numerical Analysis and Deep Learning

Ishan Nath, Lent 2024

Based on Lectures by Dr. Davide Murari

March 20, 2025

Contents

1 One-step Methods

The problem we are studying is the initial-value ODE

$$\dot{x}(t) = F(x(t)), \qquad x(0) = x_0.$$

We can make F autonomous, by adding t as a dimension and stating $\dot{t} = 1$. $F : \mathbb{R}^d \to \mathbb{R}^d$ is a 'smooth' vector field.

Our goal is, given $T > 0$ and times $0 = t_0 < t_1 < t_2 < \cdots < T$, find values x_i where $x_i \approx x(t_i)$. We assume a uniform grid: $h = t_{i+1} - t_i$.

A *one-step method* is a map $\varphi^h_F : \mathbb{R}^d \to \mathbb{R}^d$. Then $x_{i+1} = \varphi^h_F(x_i)$.

The *flow* is $\phi_F : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$, given by $(t, x_0) \mapsto x(t)$, the solution to the IVP at time t. We also write this as $\phi^t_F(x_0)$. So $\phi^h_F(x(t)) = x(t + h)$.

Definition 1.1 (Order of a one-step method)**.** A one-step method $\phi^h_F : \mathbb{R}^d \to \mathbb{R}^d$ applied to a 'smooth' vector field $F : \mathbb{R}^d \times \mathbb{R}^d$ is of order $p \in \mathbb{N}$ if

$$\varphi^h_F = \phi^h_F + \mathcal{O}(h^{p+1}).$$

Example 1.1. (Explicit Euler Method)

We take

$$\varphi^h_F(x) = x + hF(x).$$

By Taylor expansion,

$$\phi^h_F(x) = x + hF(x) + \mathcal{O}(h^2).$$

Hence φ_F has order $p = 1$.

1.1 Runge-Kutta Methods

These methods are characterized by a *tableaux*: a triple (A, b, c) where $A \in \mathbb{R}^{s \times s}$, $b, c \in \mathbb{R}^s$.

Definition 1.2 (Runge-Kutta Method)**.** Consider a non-autonomous ODE $x'(t) = F(t, x(t))$. A *Runge-Kutta method* of s-stages based on tableaux (A, b, c) is a one-step method defined as

$$\phi^h_F(x_n) = x_n + h \sum_{i=1}^{n} b_i F(t_n + c_i h, K_i),$$

where

$$K_i = x_n + h\sum_{j=1}^{s} A_{ij}F(t_n + c_jh, K_j).$$

The K_i are defined implicitly; to solve we need to solve $d \times s$ non-linear equations, which may be costly. Making A lower triangular may fix this by making the problem explicit.

The Euler method is a $(0, 1, 0)$ Runge-Kutta method.

1.2 Colocation Methods

For these class of methods, we will assume that between two time steps t_n and t_{n+1}, the solution $x(t)$ behaves like $\tilde{x}(t)$, where $\tilde{x}$ is a real polynomial of degree s. To find suitable $\tilde{x}$, we must supply some conditions.

1. $\tilde{x}(t_n) = x_n$.
2. At times $t_{n,i}$, which are between t_n and t_{n+1}, $\dot{\tilde{x}}(t_{n,i}) = F(\tilde{x}(t_{n,i}))$.

The last enforced condition is similar to the loss term in a PINN.

We typically take $t_{n,i} = t_n + c_ih$, for some $0 \leq c_1 < c_2 < \cdots < c_s \leq 1$.

It turns out that these methods are Runge-Kutta methods.

Proof: Since $\dot{\tilde{x}}$ is degree $s-1$, it is characterized by s points. In particular, its values at times $t_{n,i}$ are enough. This means that

$$\dot{\tilde{x}}(t) = \sum_{i=1}^{s} F(\tilde{x}(t_n + c_ih))\ell_i\left(\frac{t - t_n}{h}\right),$$

where ℓ_i are the elementary Lagrange polynomials,

$$\ell_i(t) = \prod_{\substack{j=1 \\ j\neq i}}^{s} \frac{t - c_j}{c_i - c_j}.$$

We can show $\ell_i(c_j) = \delta_{ij}$. To find the values, we integrate the above expression from t_n to $t_n + c_ih$:

$$\int_{t_n}^{t_n+c_Ih} \dot{\tilde{x}}(t)\,\mathrm{d}t = \sum_{j=1}^{s} F(\tilde{x}(t_n + c_jh)) \int_{t_n}^{t_n+c_ih} \ell_j\left(\frac{t - t_n}{h}\right)\mathrm{d}t,$$

or

$$\tilde{x}(t_n + c_i h) = \tilde{x}(t_n) + h \sum_{j=1}^{s} F(\tilde{x}(t_n + c_j h)) \int_0^{c_i} \ell_j(s)\,\mathrm{d}s.$$

This reminds us of the Runge-Kutta methods, where

$$K_i = x_n + h \sum_{j=1}^{s} a_{ij} F(K_j),$$

where a_{ij} is our integral, and

$$x_{n+1} = \tilde{x}(t_n + h) = x_n + h \sum_{j=1}^{s} b_i F(K_i),$$

where

$$b_i = \int_0^1 \ell_j(s)\,\mathrm{d}s$$

1.3 Gauss-Legendre Colocation Methods

These are colocation methods where $c_1, \dots, c_s \in [0,1]$ are the zeroes of the degree s Legendre polynomial P_s,

$$P_0(x) = 1, \qquad P_1(x) = x, \qquad P_2(x) = \frac{1}{2}(3x^2 - 1),$$

and in general

$$P_n(x) = \frac{1}{2^n n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n} (x^2 - 1)^n.$$

Importantly, they are orthogonal in $L^2(-1,1)$. Remember quadrature rules:

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{i=1}^{s} w_i f(x_i) + \mathrm{err}(f).$$

If we pick c_i to be the zeroes of the Legendre polynomial,

$$\int_{t_n}^{t_n+h} f(t)\,\mathrm{d}t = h \sum_{i=1}^{s} w_i f(t_n + c_i h) + \mathrm{err}(f),$$

where

$$|\mathrm{err}(f)| \le c h^{2s+1} \max_{t \in [t_0, t_0+h]} |f^{(2s)}(t)|.$$

Proposition 1.1 (Grobern-Alekseev Formula)**.** *Let us consider two initial value problems:*

$$\begin{cases} \dot{x}(t) = F(x(t)), \\ x(0) = x_0, \end{cases} \qquad \begin{cases} \dot{y}(t) = F(y(t)) + G(y(t)), \\ y(0) = x_0, \end{cases}$$

with $F \in C^1(\mathbb{R}^d, \mathbb{R}^d)$, and supposing these admit a unique solution, then for all $t \geq 0$,

$$y(t) - x(t) = \int_0^t \left.\frac{\partial \phi_F^{t-z}(z_0)}{\partial z_0}\right|_{z_0 = y(z)} G(y(z))\,\mathrm{d}z.$$

Theorem 1.1. *The Gauss-Legendre collocation methods based on s collocation points are of order $2s$.*

Proof: Notice that $x, \tilde{x}$ satisfy

$$\begin{cases} \dot{\tilde{x}}(t) = F(\tilde{x}(t)) + (\dot{\tilde{x}}(t) - F(\tilde{x}(t))), \\ \tilde{x}(t_n) = x_n, \end{cases} \qquad \begin{cases} \dot{x}(t) = F(x(t)), \\ x(t_n) = x_n, \end{cases}$$

then we can use the GA formula to see

$$\begin{aligned} \tilde{x}(t_{n+1}) - x(t_{n+1}) &= \int_{t_n}^{t_{n+1}} \left.\frac{\partial \phi_F^{t_{n+1}-z}(z_0)}{\partial z_0}\right|_{z_0 = \tilde{x}(z)} (\dot{\tilde{x}}(z) - F(\tilde{x}(z)))\,\mathrm{d}z \\ &= h \sum_{i=1}^{s} w_i \left.\frac{\partial \phi_F^{t_{n+1}-t_{n,i}}(z_0)}{\partial z_0}\right|_{z_0 = \tilde{x}(t_{n,i})} (\dot{\tilde{x}}(t_{n,i}) - F(\tilde{x}(t_{n,i}))) + \text{err}, \end{aligned}$$

but the defect terms are 0, so we are left with the error terms, which for the zeroes of the Legendre polynomials are $\mathcal{O}(h^{2s+1})$. This gives an order $2s$ method.

Example 1.2.

Consider the implicit midpoint equation:

$$x_{n+1} = x_n + hF\left(\frac{x_n + x_{n+1}}{2}\right).$$

This does not look like a Runge-Kutta method, but if we check $A = [1/2]$, $c = [1/2]$ and $b = 1$, we get

$$K_1 = x_n + \frac{h}{2}F(K_1), \qquad x_{n+1} = x_n + hF(K_1).$$

Now notice that

$$\frac{x_n + x_{n+1}}{2} = \frac{x_n}{2} + \frac{x_n + hF(K_1)}{2} = x_n + \frac{h}{2}F(K_1) = K_1.$$

1.4 A-Stability or Linear Stability

Consider the linear DE $\dot{x} = \lambda x$, $x(0) = 1$ for $\lambda \in \mathbb{C}$. We have analytic solution $x(t) = e^{\lambda t}$.

Consider discretizing this problem, for $\Re(\lambda) < 0$. Then

$$x_{n+1} = x_n + h\lambda x_n = (1 + h\lambda)x_n.$$

So $x_n = (1 + h\lambda)^n x_0$. If $\Re\lambda < 0$, we expect $x_n \to 0$, but this is only the case if $|1 + h\lambda| < 1$, which bounds our step size h.

Lemma 1.1 (Stability Function)**.** *Let us consider the linear test equation $\dot{x} = \lambda x$. If we apply an s-stage Runge-Kutta method with tableaux (A, b, c), then the update map is $\phi^h(x_n) = R(h\lambda)x_n$, where*

$$R(z) = 1 + zb^T(I_s - zA)^{-1}1_s.$$

$R(z)$ is called the stability function *of the Runge-Kutta method, and it is a rational function.*

Proof: Recall that

$$K_i = x_n h\lambda \sum_{j=1}^{s} a_{ij}K_j,$$

and

$$x_{n+1} = x_n + h\lambda \sum_{i=1}^{s} K_i.$$

Now $a_{ij}K_j = (AK)_i$, where $K = (K_1, \dots, K_s)$. This lets us write

$$K = x_n 1_s + h\lambda AK \implies K = (I - h\lambda A)^{-1}x_n 1_s$$

Applying this to x_{n+1}, we see

$$\begin{aligned} x_{n+1} &= x_n + h\lambda b^T((I - h\lambda A)^{-1}x_n 1_s) \\ &= [1 + h\lambda b^T(I - h\lambda A)^{-1}1_s]x_n = R(h\lambda)x_n. \end{aligned}$$

Now we want to show this is a rational function:

$$\begin{aligned} R(z) &= 1 + zb^T(I_s - zA)^{-1}1_s = \det(I_s + z(I_s - zA)^{-1}1_S b^T) \\ &= \det((I_s - zA)^{-1}(I_s - zA) + z(I_s - zA)^{-1}1_s b^T) \\ &= \frac{\det(I_s - zA + z1_s b^T)}{\det(I_s - zA)}. \end{aligned}$$

Hence R is rational. If A is explicit, so it is lower-triangular, then $\det(I_s - zA) = 1$, so R is a polynomial of degree s.

Example 1.3.

Consider the implicit method

$$x_{n+1} = x_n + hF(x_{n+1}) = x_n + \lambda x_{n+1}.$$

Solving this,

$$x_{n+1} = \frac{1}{1 - h\lambda} x_n = R(h\lambda)x_n,$$

so $R(z) = 1/(1 - z)$.

Definition 1.3 (A-stable Runge-Kutta Method)**.** A Runge-Kutta method ϕ^h of tableaux (A, b, c) and stability function $R(z)1 + zb^T(I_s - zA)^{-1}1_s$ is *A-stable* or *linearly stable* if

$$\mathbb{C}^- = \{z \in \mathbb{C} \mid \Re(z) < 0\} \subseteq S = \{z \in \mathbb{C} \mid |R(z)| < 1\}.$$

Example 1.4.

Recall that for explicit Euler, $R(z) = 1 + z$, and for implicit Euler $R(z) = 1/(1 - z)$.

For explicit Euler,

$$|R(z)| < 1 \iff |z + 1| < 1,$$

which corresponds to a circle around -1 of radius 1. This does not cover the entirety of $\mathbb{C}^-$, so this method is not stable.

On the other hand, for implicit Euler,

$$|R(z)| < 1 \iff |z - 1| > 1.$$

This is the exterior of the circle around 1 of radius 1, which contains all of

$\mathbb{C}^-$, hence this method is stable.

We can generalize this observation about the explicit Euler method to all explicit methods. If ϕ^h is an explicit Runge-Kutta method, then $R(z)$ is a polynomial, hence $|R(z)| < 1$ is bounded So these are never A-stable.

In fact we can say a bit more about R for explicit schemes. Recall $x(h) = e^{\lambda h} x_0$. So if $x_1 = R(\lambda h) x_0 + \mathcal{O}(h^{p+1})$, $R(\lambda h)$ must coincide with $e^{\lambda h}$ up to the order p terms.

2 Energy-Preserving Numerical Methods

Consider the linear dynamical system

$$\dot{q} = p, \qquad \dot{p} = -q.$$

This can model a spring system, and is the first-order linearization of $\ddot{q} = -q$. Letting $x = (q, p)$, we can rewrite this as

$$\dot{x} = J\nabla H(x),$$

where

$$J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \qquad H(x) = \frac{\|x\|^2}{2} = \frac{p^2 + q^2}{2}.$$

The Lie derivative of H is

$$\frac{\mathrm{d}}{\mathrm{d}t} H(x(t)) = \partial_q H \cdot \dot{q} + \partial_p H \cdot \dot{p} = q \cdot p - p \cdot q = 0.$$

So H is constant along the curves of x. Using explicit Euler,

$$\begin{aligned} x_{n+1} &= x_n + hJ\nabla H(x_n) = [I + hJ]x_n, \\ H(x_{n+1}) &= \frac{1}{2} x_n^T (I + hJ)^T (I + hJ) x_n \\ &= \frac{1}{2} x_n^T (I + hJ + hJ^T + h^2 J^T J) x_n. \end{aligned}$$

We want H to be conserved. Note that J satisfies the following properties:

$$J^T = -J, \qquad J^T J = -J^2 = I.$$

Hence this gives

$$H(x_{n+1}) = \frac{1}{2} x_n^T (1 + h^2) I x_n = (1 + h^2) \frac{\|x_n\|_2^2}{2} = (1 + h^2) H(x_n).$$

So the energy is not conserved, but has an error term of size h^2. If we take non-zero step size, this means the energy of our approximation will increase gradually, and we will 'spiral out'.

Proposition 2.1. *Consider $\dot{x} = f(x)$ such that there exists $H \in C^\infty(\mathbb{R}^d, \mathbb{R})$ with $H(x(t)) = H(x_0)$. Then a one-step method $\phi_F^h : \mathbb{R}^d \to \mathbb{R}^d$ of order p will satisfy*

$$H(\phi_F^h(x_0)) = H(x_0) + \mathcal{O}(h^{p+1}).$$

Example 2.1.

We can check that for the implicit midpoint method

$$x_{n+1} = x_n + hF\left(\frac{x_n + x_{n+1}}{2}\right) = x_n + hJ(x_n + x_{n+1}),$$

the energy is exactly conserved.

2.1 Neural Networks

Mathematically, *neural networks* are just a parametric map $\mathcal{N}_\theta : \mathbb{R}^c \to \mathbb{R}^d$, defined by composing L functions, called *layers*:

$$\mathcal{N}_\theta = F_{\theta_L} \circ \cdot F_{\theta_1},$$

where $F_{\theta_i} : \mathbb{R}^{c_i} \to \mathbb{R}^{c_{i+1}}$.

Usually we have alternating linear maps with non-linear functions entrywise σ.

σ is the *activation function*, e.g. RELU, or tanh or sigmoid.

Choosing $L_i(\mathbf{x}) = A_i\mathbf{x} + \mathbf{b}_i$, the layer is $F_{\theta_i}(\mathbf{x}) = \sigma(A_i\mathbf{x} + \mathbf{b}_i)$, a *fully-connected neural network*.

If $\mathbf{L}_i(\mathbf{x}) = k_i * \mathbf{x} + \mathbf{b}_i$, we have a *convolutional neural network*.

The weights θ of the neural network $\mathcal{N}_\theta$ are usually found by solving an optimisation problem; the process is called *network training*.

The *loss function* is the function to be minimized, as a result of the data or properties we want the network to satisfy. A typical loss function is the *mean-squared error*

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \|\mathcal{N}_\theta(\mathbf{x}_i) - \mathbf{y}_i\|_2^2.$$

A fundamental result is the universal approximation theorem: a one-layer neural network can represent any function on a compact set up to error ε, as long as σ is not a polynomial.

A particularly interesting architecture is given by *residual networks*, where

$$F_{\theta_i}(\mathbf{x}) = \mathbf{x} + \mathcal{F}_{\theta_i}(\mathbf{x}).$$

Here e.g. $\mathcal{F}$ could be a fully-connected layer, then a non-linearity. These were introduced as they are easier to train when the network has a high number of layers.

Consider a classification problem, where $\mathbf{x}_i$ is say an image, and y_i is a classification. We say $\mathcal{N}_\theta(\mathbf{x}) \in [0,1]$, and classify $y_i = 1$ if $\mathcal{N}_\theta(\mathbf{x}_i) > 0.5$.

Typically one uses binary cross entropy. To minimize $\mathcal{L}(\theta)$, we typically use gradient descent:

$$\theta_{k+1} = \theta_k - \tau \nabla \mathcal{L}(\theta_k).$$

If the gradient entries are too large or small, we struggle to find a meaningful set of weights.

For a ResNet, the layer

$$F_{\theta_i}(\mathbf{x}) = \mathbf{x} + B_i^T \sigma(A_i \mathbf{x} + \mathbf{b}_i) = \mathbf{x} + \mathcal{F}_{\theta_i}(\mathbf{x})$$

is an explicit Euler step of size 1 for the initial value problem

$$\mathbf{y}(0) = \mathbf{x}, \qquad \dot{\mathbf{y}}(t) = B_i^T \sigma(A_i \mathbf{y}(t) + \mathbf{b}_i) = \mathcal{F}_{\theta_i}(\mathbf{y}(t)).$$

We can define *ResNet-like neural networks* by choosing parametric functions $\mathcal{S}_\sigma$ and a numerical method $\varphi_{\mathcal{F}}^h$, like explicit Euler, and set

$$\mathcal{N}_\theta(\mathbf{x}) = \varphi_{\mathcal{F}_{\theta_L}}^{h_L} \circ \cdot \circ \varphi_{\mathcal{F}_{\theta_1}}^{h_1}(\mathbf{x}).$$

We can combine these with lifting and projection layers, as for usual neural networks.

We can also use neural networks to discover differential equations. If we for example say

$$\dot{\mathbf{x}} = \mathcal{N}_\theta(\mathbf{x}),$$

and solve with numerical methods we can discover the equations governed by the system. If we let

$$\dot{\mathbf{x}} = J \nabla H_\theta(\mathbf{x}),$$

this gives a network with a Hamiltonian. We can also use this to solve differential equations: say we want a network $\mathcal{N}_\theta : [0, \Delta t] \times \mathbb{R}^d \to \mathbb{R}^d$ that generates time evolution. We can then train with a suitable loss function.

3 ODEs with a First Integral

Let us consider the ODE

$$\dot{x} = F(x), \tag{†}$$

for $F : \mathbb{R}^d \to \mathbb{R}^d$. A scalar valued function $I : \mathbb{R}^d \to \mathbb{R}$ is a *first integral* if it is constant along the solutions of (†).

In the case I is continuously differentiable, we can equivalently say

$$\frac{\mathrm{d}}{\mathrm{d}t} I(\phi_F^t(x_0)) = 0 \iff \nabla I(\phi_F^t(x_0)) \cdot F(\phi_F^t(x_0)),$$

or removing the dependence on x_0,

$$\nabla I(x) \cdot F(x) = 0,$$

for all $x \in \mathbb{R}^d$.

A vector field $F : \mathbb{R}^d \to \mathbb{R}^d$ admits a first integral $I : \mathbb{R}^d \to \mathbb{R}$ which is C^1, if and only if it can be written as

$$F(x) = S(x)\nabla I(x),$$

where S is skew symmetric, i.e. $S(x)^T = -S(x) = A(x) - A(x)^T$.

Proof: If F is of this form, then

$$\nabla I(x)^T S(x) \nabla I(x) = 0.$$

On the other hand, if I is a first integral, let

$$S(x) = \frac{F(x)\nabla I(x)^T - \nabla I(x) F(x)^T}{\|\nabla I(x)\|^2}.$$

Then S is clearly skew symmetric, and

$$\begin{aligned} S(x)\nabla I(x) &= \frac{F(x)\nabla I(x)^T \nabla I(x) - \nabla I(x) F(x)^T \nabla I(x)}{\|\nabla I(x)\|^2} \\ &= F(x). \end{aligned}$$

3.1 Runge-Kutta for Linear First Integrals

Definition 3.1. A one-step method $\varphi_F^h : \mathbb{R}^d \to \mathbb{R}^d$ applied to $\dot{x} = F(x)$, with $\nabla I(x)^T F(x) = 0$, is said to preserve the first integral $I : \mathbb{R}^d \to \mathbb{R}$ if for all $x_0 \in \mathbb{R}^d$,

$$I(\varphi_F^h(x_0)) = I(x_0).$$

In particular, if $I(x_0) = c$, then $\phi_F^t(x_0)$ is tangent to the submanifold

$$\{z \in \mathbb{R}^d \mid I(z) = c\} = I_c,$$

the *level set* of c.

Theorem 3.1. *Let $\varphi^h : \mathbb{R}^d \to \mathbb{R}^d$ be an arbitrary explicit or implicit Runge-Kutta scheme. Let $\dot{x} = F(x)$ be such that $v^T F(x) = 0$ for some $v \in \mathbb{R}^d$. Then*

$$I(\varphi_F^h(x)) = I(x)$$

for all $x \in \mathbb{R}^d$, with $I(x) = v^T x$.

Proof: Let φ^h be a Runge-Kutta method with tableaux (A, b, c), so

$$\begin{aligned} x_{n+1} = \varphi_F^h(x_n) &= x_n + h \sum_{i=1}^s b_i F(K_i), \\ K_i &= x_n + h \sum_{j=1}^s a_{ij} F(K_j). \end{aligned}$$

Then we find

$$\begin{aligned} I(x_{n+1}) = v^T x_{n+1} &= v^T x_n + h \sum_{i=1}^s b_i v^T F(K_i) \\ &= v^T x_n = I(x_n). \end{aligned}$$

Example 3.1. (SIR Model)

Consider the SIR model of population dynamics: the population $P = S+I+R$, where S is susceptible, I is infected and R is recovered. So we can consider the phase space $X = (S, I, R) \in \mathbb{R}^3$.

We assume no one dies, so $\dot{P} = 0$. Hence $1^T X = 0$.

3.2 Quadratic Invariants

We consider first integrals of the form

$$Q(x) = x^T C x,$$

for some $C^T = C$.

Theorem 3.2. *Consider the Runge-Kutta method φ^h of tableaux (A, b, c), and a vector field $F : \mathbb{R}^d \to \mathbb{R}^d$ such that $x^TCF(x) = 0$, i.e. $\nabla Q(x) \cdot F(x) = 0$ with $Q(x) = x^TCx$.*

Define the matrices

$$B = \operatorname{diag}(b), \qquad M = BA + A^TB - bb^T.$$

If we have $M = 0$, it holds that

$$Q(\varphi_F^h(x)) = Q(x)$$

for all $x \in \mathbb{R}^d$.

Looking at M, the general term is

$$m_{ij} = b_ia_{ij} + a_{ji}b_j - b_ib_j.$$

Example 3.2.

The diagonal term is $2b_ia_{ii} - b_i^2$. If the method is explicit, $a_{ii} = 0$, so $m_{ii} = -b_i^2$. Hence for $M = 0$, we need $b = 0$, i.e. our method is the identity. Hence no non-trivial explicit method can preserve the quadratic first integral.

However, for the midpoint method, $A = 1/2$, $c = 1/2$ and $b = 1$, and we can verify

$$M = \frac{1}{2} + \frac{1}{2} - 1 = 0.$$

Proof: Recall our definitions for x_{n+1} and K_i. We will use the following rewriting:

$$K_i = x_n + h\sum_{j=1}^{s} a_{ij}F(K_j) \implies x_n = K_i - h\sum_{j=1}^{s} a_{ij}F(K_j).$$

Then note that

$$\begin{aligned} Q(x_{n+1}) = x_{n+1}^TCx_{n+1} &= \left(x_n + h\sum_{i=1}^{s} b_iF(K_i)\right)^T C \left(x_n + h\sum_{j=1}^{s} b_jF(K_j)\right) \\ &= x_n^TCx_n + 2h\sum_{i=1}^{s} b_ix_n^TCF(K_i) + h^2\sum_{i,j=1}^{s} b_ib_jF(K_i)^TCF(K_j) \end{aligned}$$

Replacing x_n^T with our formula involving K_i, we find this is

$$\begin{aligned}
&= Q(x_n) + 2h\sum_{i=1}^s b_i \underbrace{K_i^T CF(K_i)}_{0} - 2h^2 \sum_{i,j=1}^s b_i a_{ij} F(K_i)^T CF(K_j) \\
&\qquad + h^2 \sum_{i,j=1}^s b_i b_j F(K_i)^T CF(K_j) \\
&= Q(x_n) + h^2 \sum_{i,j=1}^s (b_i b_j - b_i a_{ij} - b_j a_{ji}) F(K_i)^T CF(K_j) \\
&= Q(x_n) - h^2 \sum_{i,j=1}^s m_{ij} F(K_i)^T CF(K_j),
\end{aligned}$$

which equals to $Q(x_n)$ if $M = 0$.

Proposition 3.1. *All Gauss-Legendre collocation methods preserve quadratic first integrals.*

Proof: Recall the collocation methods: we have a polynomial u such that $x_{n+1} = u(t_n + h)$, where

$$\dot{u}(t_n + c_i h) = F(u(t_n + c_i h)),$$

for $0 \le c_1 < c_2 < \cdots < c_s \le 1$.

Define a function $q(t) = Q(u(t))$, where $Q(x) = x^T Cx$. To be energy preserving, we need $q(t_n) = q(t_{n+1})$, i.e.

$$\int_{t_n}^{t_{n+1}} \dot{q}(t)\,\mathrm{d}t = \int_{t_n}^{t_{n+1}} 2\dot{u}(t)^T Cu(t)\,\mathrm{d}t = 0.$$

But we can evaluate this using a quadrature rule: it is a polynomial of degree at most $2s - 1$, hence the quadrature rule is exact, so

$$L = h\int_0^1 \dot{q}(t_n + sh)\,\mathrm{d}s = h\sum_{i=1}^s \dot{q}(t_n + c_i h) \cdot p_i.$$

But note that

$$\dot{q}(t_n + c_i h) = 2\dot{u}(t_n + c_i h)^T Cu(t_n + c_i h) = 2F(u(t_n + c_i h))^T Cu(t_n + c_i h) = 0.$$

Example 3.3. (Rigid-Body)

Consider $x \in \mathbb{R}^3$, with

$$\dot{x} = \begin{pmatrix} 0 & x_3/I_3 & -x_2/I_2 \\ x_3/I_3 & 0 & x_1/I_1 \\ x_2/I_2 & -x_1/I_1 & 0 \end{pmatrix} x, \qquad I = \operatorname{diag}(I_1, I_2, I_3).$$

Then we can find two first integrals

$$H_1(x) = \frac{\|x\|_2^2}{2}, \qquad H_2(x) = \frac{1}{2} x^T I^{-1} x.$$

3.3 Higher-order First Integrals

We aim to show the following.

Proposition 3.2. *For $n \geq 3$, there is no Runge-Kutta method that can preserve all polynomial first integrals of degree n.*

Lemma 3.1. *Let $B : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ be a matrix-valued function with*

$$\operatorname{tr}(B(Y)) = 0$$

for all $y \in \mathbb{R}^{d\times d}$. Then $g(Y) = \det Y$ is a first integral of the differential equation

$$\dot{Y} = B(Y)Y.$$

Proof: Note that

$$Y(t+h) = Y(t) + hB(Y(t))Y(t) + \mathcal{O}(h^2),$$

and

$$\det(Y + hBY) = \det(I + hB)\det Y = \det Y(1 + h\operatorname{tr} B + \mathcal{O}(h^2)),$$

so we find

$$g'(Y) = \lim_{h\to 0} \frac{\det Y(1 + h\operatorname{tr} B) - \det Y}{h} = \det Y \operatorname{tr} B = 0.$$

In general we find the result that $g'(Y) = g(Y)\operatorname{tr} B$.

Lemma 3.2. *Let $R(z)$ be a differentiable function defined in a neighbourhood of $z = 0$, with $R(0) = R'(0) = 1$. Then for $d \geq 3$, $\det R(B) = 1$ for all $B \in \mathbb{R}^{d\times d}$ with $\operatorname{tr} B = 0$ if and only if $R(z) = e^z$.*

Proof: If R is exponential, then $R(tB) = \exp(tB) = Y(t)$, where

$$\dot{Y}(t) = BY(t), \qquad Y(0) = I.$$

Then from the previous theorem, $\det Y(t) = \det Y(0) = 1$.

For the forward direction, suppose for all B with $\operatorname{tr} B = 0$, $\det R(B) = 1$. Let

$$B = \operatorname{diag}(\mu, \nu, -(\mu+\nu), 0, \dots, 0),$$

for μ, ν small enough. Then

$$R(B) = \operatorname{diag}(R(\mu), R(\nu), R(-(\mu+\nu)), R(0), \dots, R(0)).$$

As $R(0) = 0$, $\det R(B) = 1 = R(\mu)R(\nu)R(-(\mu+\nu))$. Setting $\mu = -\nu$, we find $R(\mu)R(-\mu) = 1$, and hence $R(\mu)R(\nu) = R(\mu+\nu)$, so now

$$\frac{R(\mu+h) - R(\mu)}{h} = R(\mu)\frac{R(h) - 1}{h} \to R(\mu)R'(0) = R(\mu).$$

So $R'(\mu) = R(\mu)$, i.e. R is exponential.

In general, if we apply Runge-Kutta method (A, b, c) to $\dot{x} = Bx$, then we can show that

$$\varphi^h(x_n) = x_{n+1} = R(hB)x_n.$$

Theorem 3.3. *For $d \geq 3$, no Runge-Kutta method can preserve all polynomial first integrals of degree d.*

Proof: Let us consider a generic matrix $B \in \mathbb{R}^{d\times d}$, with $\operatorname{tr} B = 0$, and define the matrix ODE $\dot{Y} = BY$.

We know that $g(Y) = \det Y$ is invariant, and that $Y_{n+1} = R(hB)Y_n$. Hence

$$\det(Y_{n+1}) = \det(R(hB))\det(Y_n).$$

To preserve $\det Y$, we must have $\det(R(hB)) = 1$ for all B. But the only R that satisfies this is $R(Z) = \exp Z$.

Since we know that R is a rational function, this can never be the case.

What happens for specific higher-degree polynomials, or non-polynomial first integrals?

4 Projection Methods

Example 4.1. (Simple Pendulum)

Consider the ODE

$$\dot{x} = p, \qquad \dot{p} = -\sin x.$$

In this case, the system is generated by Hamiltonian

$$H(x,p) = \frac{p^2}{2} - \cos x.$$

This is not a polynomial. Can we design a procedure that preserves this?

The idea is as follows. Consider the set $\{(q,p) \in \mathbb{R}^2 \mid H(q,p) = H(q_0,p_0)\}$, the level set obtained by starting at (q_0,p_0). To stay in the level set, we can start by doing our one-step method, and then 'projecting' back onto the level set.

Consider $\dot{x} = f(x)$, which is known to preserve $I : \mathbb{R}^d \to \mathbb{R}$, with $I(\phi_F^t(x_00)) = I(x_0)$ for all $x_0 \in \mathbb{R}^d$, and $t \geq 0$. This means that

$$\phi_F^t(x_0) \in \mathcal{M}_{x_0} = \{x \in \mathbb{R}^d \mid I(x) = I(x_0)\} \subseteq \mathbb{R}^d,$$

with dimension $d-1$. A *projection method* $\varphi_F^h : \mathbb{R}^d \to \mathbb{R}^d$ is defined as

$$\varphi_F^h(x_0) = \Pi_{\mathcal{M}_{x_0}} \circ \tilde{\varphi}_F^h(x_0),$$

where $\tilde{\varphi}_F^h(x_0)$ is our base method. To realise the projection $\Pi_{\mathcal{M}_{x_0}}$, we usually solve

$$\Pi_{M_{x_0}}(x) = \underset{y}{\operatorname{argmin}} \|y - x\|_2^2.$$

In practice, we define

$$\tilde{x}_1 = \tilde{\varphi}_F^h(x_0), \qquad x_1(\lambda) = \tilde{x}_1 + \lambda \nabla I(\tilde{x}_1),$$

and then solve for $\lambda \in \mathbb{R}$ such that $I(x_1(\lambda)) = I(x_0)$.

Example 4.2.

Consider $I(x) = \|x\|_2^2/2$, for $x \in \mathbb{R}^d$, and $\dot{x} = (S - S^T)x$.

Let $\tilde{\varphi}^h : \mathbb{R}^d \to \mathbb{R}^d$ be an arbitrary one-step method, and

$$\tilde{x}_1 = \tilde{\varphi}^h(x_0), \qquad x_1(\lambda) = \tilde{x}_1 + \lambda \nabla I(\tilde{x}_1) = (1+\lambda)\tilde{x}_1.$$

Then,

$$I(x_0) = \frac{\|x_0\|^2}{2} = I(x_1(\lambda)) = \frac{\|x(\lambda)\|^2}{2} = \frac{(1+\lambda)^2}{2}\|\tilde{x}_1\|^2,$$

so we can choose

$$\lambda = -1 \pm \frac{\|x_0\|}{\|\tilde{\varphi}^h(x_0)\|} \implies \lambda = -1 + \frac{\|x_0\|}{\|\tilde{\varphi}^h(x_0)\|},$$

since when $h = 0$, we should have the identity map, and $\lambda = 0$. Hence, simplifying we get

$$x_1(\lambda) = \frac{\tilde{\varphi}^h(x_0)}{\|\tilde{\varphi}^h(x_0)\|}\|x_0\|.$$

This can be generalized to $I(x) = (x^T C x)/2$.

We saw that if $\varphi_F^h : \mathbb{R}^d \to \mathbb{R}^d$ is of order p, then

$$I(\varphi_F^h(x_0)) = I(x_0) + \mathcal{O}(h^{p+1}).$$

Let's consider $F : \mathbb{R}^d \to \mathbb{R}^d$ that preserves $n < d$ functionally independent first integrals $I = (I_1, \dots, I_n) : \mathbb{R}^d \to \mathbb{R}^n$.

Functionally independent means that the level set of c has codimension n, for every c in the range of I, or $\operatorname{rank}(\partial_x I(x)) = n$.

We consider

$$\tilde{x}_1 = \tilde{\varphi}_F^h(x_0), \qquad x_1(\vec{\lambda}) = \tilde{x}_1 + (\partial_x I(\tilde{x}_1))^T \vec{\lambda} = \tilde{x}_1 + \sum_{i=1}^n \lambda_i \nabla_i I_i(\tilde{x}_1).$$

We want to find $\vec{\lambda} \in \mathbb{R}^n$ such that

$$I(x_1(\vec{\lambda})) = I(x_0).$$

Lemma 4.1. *Let $F : \mathbb{R}^d \to \mathbb{R}^d$ be as above. Consider the projection method defined by*

$$\tilde{x}_1 = \tilde{\varphi}_F^h(x_0), \qquad x_1(\vec{\lambda}) = \tilde{x}_1 + \partial_x I(\tilde{x}_1)^T \vec{\lambda}.$$

Then there exists $\bar{h} > 0$ such that $\vec{\lambda} = \vec{\lambda}(h)$ is a well defined function $\vec{\lambda} : [0, \bar{h}] \to \mathbb{R}^n$ with $\vec{\lambda}(0) = \vec{0}$, and $g(x_1(\vec{\lambda}(h))) = 0$ for all $h \in [0, \bar{h}]$.

Proof: We define a function

$$G(h,\vec{\lambda}) = g(x_1(\vec{\lambda})) = I(\tilde{\varphi}^h(x) + \partial_x I(\tilde{\varphi}^h(x))^T \vec{\lambda}) - I(x).$$

We note that $G(0,\vec{0}) = 0$. Moreover,

$$\partial_{\vec{\lambda}} G(h,\vec{\lambda})\bigg|_{(h,\vec{\lambda})=(0,0)} = \partial_x I(x)^T \partial_x I(x),$$

which is non-singular. So the implicit function theorem ensures that there is $\bar{h}$ such that $\vec{\lambda} = \vec{\lambda}(h)$ satisfies

$$G(h,\vec{\lambda}(h)) = 0, \qquad \vec{\lambda}(0) = \vec{0}.$$

Do these methods preserve the accuracy of our base method? The answer is yes.

Lemma 4.2 (Order of Projection Methods)**.** *Let F, I be as above, and assume that $\tilde{\varphi}_F^h$ is of order p. Let h be small enough so that the previous lemma applies. Then the projection method $x_1(\vec{\lambda}(h))$ is still of order p.*

Proof: We Taylor expand:

$$\begin{aligned} G(h,\vec{\lambda}) &= I(\tilde{\varphi}^h(x_0) + \partial_x I(\tilde{\varphi}^h(x_0))^T \vec{\lambda}) - I(x_0), \\ G(h,\vec{\lambda}) &= G(h,\vec{0}) + (\partial_{\vec{\lambda}} G(h,\vec{0}))\vec{\lambda} + \mathcal{O}(\|\vec{\lambda}\|^2), \\ G(h,\vec{0}) &= I(\tilde{\varphi}^h(x_0)) - I(x_0) = \mathcal{O}(h^{p+1}). \end{aligned}$$

So,

$$\begin{aligned} \tilde{\varphi}_F^h(x_0) &= \phi_F^h(x_0) + \mathcal{O}(h^{p+1}), \\ \implies I(\tilde{\varphi}_F^h(x_0)) &= I(x_0) + \mathcal{O}(h^{p+1}). \end{aligned}$$

Therefore,

$$\partial_{\vec{\lambda}} G(h,\vec{\lambda})\bigg|_{\vec{\lambda}=0} = \partial_{\vec{\lambda}} G(0,\vec{\lambda})\bigg|_{\vec{\lambda}=\vec{0}} + \mathcal{O}(h).$$

This gives

$$0 = G(h,\vec{\lambda}(h)) = \mathcal{O}(h^{p+1}) + \vec{\lambda}(h)\mathcal{O}(h) + \partial_x I(x) \partial_x I(x)^T \vec{\lambda}(h) + \mathcal{O}(\|\vec{\lambda}\|^2).$$

This shows that $\vec{\lambda}(h) \in \mathcal{O}(h^{p+1})$. Hence,

$$x_1(\vec{\lambda}(h)) = \tilde{\varphi}^h(x_0) + \partial_x I(\tilde{\varphi}^h(x_0))^T \vec{\lambda}(h) = \phi_F^h(x_0) + \mathcal{O}(h^{p+1}).$$

Some downsides of projection methods: suppose we want to preserve $Q(x) = \|x\|^2/2$. Then our projection method

$$\varphi^h(x) = \frac{\tilde{\varphi}^h(x)}{\|\tilde{\varphi}^h(x)\|_2}\|x\|_2.$$

This works great; a general Runge-Kutta method does not preserve this in general. However what a general RK method does preserve is any linear invariant. Hence if we want a solution that preserves Q as well as a linear energy, it is hard to do so with a projection-based method.

5 Discrete Gradient Methods

Suppose we want to solve the ODE

$$\dot{x} = F(x) = S(X)\nabla I(X),$$

where $I : \mathbb{R}^d \to \mathbb{R}$ is smooth, and S is as before, such that $S(x)\nabla I(x) = F(x)$. We can also consider

$$F(x) = J\nabla H(x),$$

where J is symplectic.

Definition 5.1. Let $I : \mathbb{R}^d \to \mathbb{R}$ be a smooth function. A *discrete gradient* of I is a function $\bar{\nabla} I : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ such that:

- $\lim_{y\to x} \bar{\nabla} I(x, y) = \nabla I(x)$.
- $\bar{\nabla} I(x, y)^T (y - x) = I(y) - I(x)$.

We have several different choices of discrete gradients.

Example 5.1. (Examples of Discrete Gradient)

1. The average vector field:

$$\bar{\nabla} I(x, y) = \int_0^1 \nabla I((1 - s)x + sy)\, \mathrm{d}y.$$

2. The Gonzalez discrete gradient:

$$\bar{\nabla} I(x, y) = \nabla I\left(\frac{x + y}{2}\right) + \frac{I(y) - I(x) - (y - x)^T \nabla\left(\frac{x+y}{2}\right)}{\|y - x\|^2}(y - x).$$

3. Itoh-Abe: coordinate-wise, it is given as

$$\bar{\nabla} I(x, y)_i = \frac{I(y_1, \ldots, y_i, x_{i+1}, \ldots, x_d) - I(y_1, \ldots, y_{i-1}, x_i, \ldots, x_d)}{y_i - x_i}.$$

We define a *discrete gradient method* based on the discrete gradient $\bar{\nabla} I$ by utilizing update

$$x_{n+1} = x_n + h\bar{S}(x_n, x_{n+1})\bar{\nabla} I(x_n, x_{n+1}),$$

where $\bar{S} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is such that:

- $\lim_{y\to x} \bar{S}(x, y) = S(x)$,

- $\bar{S}(x,y)^T = -\bar{S}(y,x)$.

Proposition 5.1. *The discrete gradient method conserves I.*

Proof: We have

$$\begin{aligned} I(x_{n+1}) - I(x_n) &= \bar{\nabla} I(x_n, x_{n+1})^T (x_{n+1} - x_n) \\ &= \frac{1}{h} \bar{\nabla} I(x_n, x_{n+1})^T \bar{S}(x_n, x_{n+1}) \bar{\nabla} I(x_n, x_{n+1}) = 0, \end{aligned}$$

by skew-symmetry of S.

We show that the average vector field is a disctete gradient.

Proposition 5.2. *Let $I : \mathbb{R}^d \to \mathbb{R}$ be smooth. Then the average vector field discrete gradient defined as*

$$\bar{\nabla} I(x,y) = \int_0^1 \nabla I((1-s)x + sy)\, \mathrm{d}s$$

is a discrete gradient.

Proof: First, it is consistent as

$$\bar{\nabla} I(x,x) = \nabla I(x) \int_0^1 \mathrm{d}s = \nabla I(x).$$

Moreover,

$$\begin{aligned} I(y) - I(x) &= \int_0^1 \frac{\mathrm{d}}{\mathrm{d}s} I((1-s)x + sy)\, \mathrm{d}s \\ &= \int_0^1 \nabla I((1-s)x + sy)^T (y-x)\, \mathrm{d}s \\ &= \bar{I}(x,y)^T (y-x). \end{aligned}$$

We showed that there is no RK method that preserves any polynomial first integral of degree $d \geq 3$. We will now show that for a specific polynomial, this is possible.

Let us consider $\dot{x} = S\nabla I(x)$, where $S \in \mathbb{R}^{d\times d}$, $S^T = -S$ and $I \in P^m(\mathbb{R})$, a polynomial.

Proposition 5.3. *Let $b_1, \dots, b_s, c_1, \dots, c_s$ define a quadrature rule of polynomial* *order $m-1$. Consider $\dot{x} = S\nabla I(x)$. Then the s-stage RK method*

$$x_{n+1} = x_n + h\sum_{i=1}^{s} b_i F(x_n + c_i(x_{n+1} - x_n))$$

preserves I.

Proof: We have that $F = S\nabla I(x) \in P^{m-1}(\mathbb{R})$, so

$$\begin{aligned}\int_0^1 F((1-s)x_n + sx_{n+1})\,\mathrm{d}s &= \sum_{i=1}^{s} b_i F(x_n + c_i(x_{n+1} - x_n)) \\ &= S\int_0^1 \nabla I((1-s)x_n + sx_{n+1})\,\mathrm{d}s \\ &= S\bar{\nabla} I(x_n, x_{n+1}),\end{aligned}$$

hence

$$x_{n+1} = x_n + hS\bar{\nabla} I(x_n, x_{n+1}) \implies I(x_{n+1}) = I(x_n).$$

5.1 Dispersive Systems

Sometimes there is not a fixed energy; a system may disperse or decrease an energy. Examples include

$$\dot{x} = -\nabla V(x),$$

for $V : \mathbb{R}^d \to \mathbb{R}$ convex and C^1, or

$$\dot{x} = P\nabla V(x),$$

for $P^T = P$ and $P \leq 0$.

Can we adapt the methods we have before? We can take

$$x_{n+1} = x_n + hP\bar{\nabla} V(x_n, x_{n+1}).$$

Then,

$$\begin{aligned}V(x_{n+1}) - V(x_n) &= \bar{\nabla} V(x_n, x_{n+1})(x_{n+1} - x)n) \\ &= \frac{1}{h}\bar{\nabla} V(x_n, x_{n+1})^T P\bar{\nabla} V(x_n, x_{n+1}) \\ &\leq 0,\end{aligned}$$

by non-positivity of P.

6 ML Detour

Recall ResNets: the layers

$$F_{\theta_i}(\mathbf{x}) = \mathbf{x} + B_i^T \sigma(A_i \mathbf{x} + \mathbf{b}_i) = \mathbf{x} + \mathcal{F}_{\theta_i}(\mathbf{x})$$

are explicit Euler steps of size 1 for the initial value problem

$$\begin{aligned} \dot{\mathbf{y}}(t) &= B_i^T \sigma(A_i \mathbf{y}(t) + \mathbf{b}_i) = \mathcal{F}_{\theta_i}(\mathbf{y}(t)), \\ \mathbf{y}(0) &= \mathbf{x}. \end{aligned}$$

We can define *ResNet-like neural networks* by choosing a family of parametric functions $\mathcal{S}_\Theta$ and a numerical method $\varphi_{\mathcal{F}}^h$, like explicit Euler, and set

$$\mathcal{N}_\theta(\mathbf{x}) = \varphi_{\mathcal{F}_{\theta_L}}^{h_L} \circ \cdots \circ \varphi_{\mathcal{F}_{\theta_1}}^{h_1}(\mathbf{x}).$$

We could also combine these blocks with lifting and projection layers.

Neural networks ca find accurate solutions, but tend not to be interpretable or reproduce desired properties. We can try tackling some of these issues by applying the theory of dynamical systems and geometric integration.

To build networks satisfying the property, we can either restrict the parametrisation $\mathcal{N}_\theta$ or modify the loss function. For example,

$$\mathcal{N}_\theta(\mathbf{x}) = \frac{\tilde{\mathcal{N}}_\theta(\mathbf{x})}{\|\tilde{\mathcal{N}}_\theta(\mathbf{x})\|_2} \|\mathbf{x}\|_2.$$

Or we can add a regulariser:

$$\tilde{\mathcal{L}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|\mathcal{N}_\theta(\mathbf{x}_i) - \mathbf{y}_i\|_2^2 + \frac{1}{N} \sum_{i=1}^{N} (\|\mathbf{x}_i\|_2 - \|\mathcal{N}_\theta(\mathbf{x}_i)\|_2)^2.$$

Note all restrictions are as effective: $\mathcal{N}_R(\mathbf{x}) = R^T x$ where R is orthogonal is norm-preserving, but probably not expressive enough.

For our case, we:

- Choose a property $\mathcal{P}$ that the network has to satisfy, e.g. norm conservation.
- Choose a family of parametric vector fields $\mathcal{S}_\Theta$ whose solutions satisfy $\mathcal{P}$, for example
$$\dot{\mathbf{x}}(t) = \mathcal{F}_\theta(\mathbf{x}(t)) = \mathcal{S}_\theta(\mathbf{x}(t))\mathbf{x}(t).$$
- Chose a numerical method $\Psi_{\mathcal{F}_\theta}^h$ that preserves $\mathcal{P}$ at a discrete level, for example a projection method.

- The resulting network

  $$\mathcal{N}_\theta = \Psi^{h_L}_{\mathcal{F}_{\theta_L}} \circ \cdots \circ \Psi^{h_1}_{\mathcal{F}_{\theta_1}}$$

  will preserve $\mathcal{P}$.

Example 6.1.

For the SIR model with linear first integral $I(x, y, z) = x + y + z = 1^T x$, we can define our network as a ResNet with

$$\mathbf{x} \mapsto \mathbf{x} + hS_{\theta_i}(\mathbf{x})\mathbf{1}.$$

For the PDE

$$\partial_t u = \partial_x u + \partial_y u,$$

this conserves the ℓ^2 norm of the solution, so we can use a ResNet with layers based on the projection method.

Example 6.2.

Consider approximating the unknown Hamiltonian

$$H(q, p) = 2mgl(1 - \cos q) + \frac{l^2}{2m} p^2,$$

based on trajectory data.

In this case one can define a network $\mathcal{N}_\theta : \mathbb{R}^2 \to \mathbb{R}$ which should resemble H. To find the approximation, we can optimise the loss function

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{n=1}^{N} \left\| \varphi^h_{X_{\mathcal{N}_\theta}}(\mathbf{x}^n_0) - \mathbf{x}^n_1 \right\|^2,$$

where $\mathbf{x}^n_1 \approx \phi^h_{X_H}(\mathbf{x}^n_0)$.

7 Hamiltonian Systems

When we refer to Hamiltonian systems, we are looking at

$$\dot{x} = J\nabla H(x),$$

where $H : \mathbb{R}^{2d} \to \mathbb{R}$ is the *Hamiltonian energy matrix*, and

$$J = \begin{pmatrix} 0 & \mathrm{id} \\ -\mathrm{id} & 0 \end{pmatrix} \in \mathbb{R}^{2d \times 2d},$$

the *symplectic matrix*. Often times, we have $x = (q, p) \in \mathbb{R}^d \times \mathbb{R}^d$, and

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

a kinetic term plus a potential.

Examples include four particles in $\mathbb{R}^3$, living in $d = 12$, where there is some potential terms between pairs of particles and kinetic terms. Then we get

$$H(q, p) = \frac{1}{2} \sum_{i=1}^{N} \frac{\|p_i\|^2}{m_i} + \sum_{i,j=1}^{N} \phi(\|q_i - q_j\|).$$

In general, we care a lot about Hamiltonians of the form

$$H(q, p) = K(p) + U(q).$$

J induces a *symplectic form* Ω given by

$$\Omega(u, v) = u^T J v.$$

Definition 7.1. A map $F : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is *symplectic* if it preserves Ω.

Example 7.1.

If $F(x) = Ax$, then for this to be symplectic we must have

$$(Au)^T JAv = u^T A^T JAv = u^T Jv \implies A^T JA = J.$$

Such an A is a *symplectic matrix*.

Proposition 7.1. *Let $F : \mathbb{R}^{2d} \times \mathbb{R}^{2d}$ be C^d. Then it is symplectic if and only if:*

- *$\Omega(F'(x)u, F'(x)v) = \Omega(u, v)$ for all $u, v \in \mathbb{R}^{2d}$*

- *$F(x)'^T JF'(x) = J$ for all $x \in \mathbb{R}^{2d}$.*

Proposition 7.2. *Let H be a twice continuously differentiable function on $U \subseteq \mathbb{R}^{2d}$ open. Then for each fixed $t \in \mathbb{R}$, the time t flow map $\phi^t_{X_H}$ is symplectic.*

Here X_H is the vector field with energy H.

Proof: The map $\phi^t_{X_H}(x_0)$ solves the initial value problem

$$\dot{x}(t) = X_H(x(t)) = J\nabla H(x(t)), \qquad x(0) = x_0.$$

We need that for all $t \in \mathbb{R}$,

$$\left(\frac{\partial \phi^t_{X_H}(x_0)}{\partial x_0}\right)^T J \left(\frac{\partial \phi^t_{X_H}(x_0)}{\partial x_0}\right) = J.$$

From now on, write

$$S_{x_0}(t) = \frac{\partial \phi^t_{X_H}(x_0)}{\partial x_0}.$$

Our proof will do the following:

- Show it is true at $t = 0$.
- Show that $$\frac{\mathrm{d}}{\mathrm{d}t}\left(S_{x_0}^T J S_{x_0}\right) = 0.$$

Differentiating with respect to x_0 the equations of $\phi^t_{X_H}$, we have that

$$\frac{\mathrm{d}}{\mathrm{d}t} S_{x_0}(t) = J\nabla^2 H(\phi^t_{X_H}(x_0)) S_{x_0}(t), \qquad S_{x_0}(0) = \mathrm{id}\,.$$

The $t = 0$ is true as at $t = 0$, $S_{x_0} = I$ which is clearly symplectic. The second point is true as

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(S_{x_0}^T J S_{x_0}) &= \dot{S}_{x_0}^T J S_{x_0} + S_{x_0}^T J \dot{S}_{x_0} = (J\nabla^2 H S_{x_0})^T J S_{x_0} + S_{x_0}^T J (J\nabla^2 H S_{x_0}) \\ &= S_{x_0}^T \nabla^2 H J^T J S_{x_0} + S_{x_0}^T J J \nabla^2 H S_{x_0} = 0,\end{aligned}$$

from properties of the symplectic matrix, since

$$-J = J^T, \qquad J^2 = -JJ^T.$$

Lemma 7.1 (Volume Preservation)**.** *Let $F : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ be a symplectic diffeomor-*

phism. Then F also preserves the canonical volume form *of* $\mathbb{R}^{2d}$*:*

$$\mathrm{vol}(F(\Omega)) = \int_{F(\Omega)} \mathrm{d}x_1 \dots \mathrm{d}x_{2d} = \mathrm{vol}(\Omega) = \int_{\Omega} \mathrm{d}x_1 \dots \mathrm{d}x_{2d},$$

for any $\Omega \subseteq \mathbb{R}^{2d}$ open.

Proof: We can apply a change of basis:

$$\int_{F(\Omega)} \mathrm{d}x_1 \dots \mathrm{d}x_{2d} = \int_{\Omega} |\det F'(x)| \,\mathrm{d}x_1 \dots \mathrm{d}x_{2d},$$

so we just need to show $|\det F'(x)| = 1$. This is true as $F'JF'^T = J$, so taking the determinant shows this. We can show the determinant is exactly 1 by looking at the Pfaffian.

Lemma 7.2. *Let $F : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ be a C^1 vector field with flow at time t, ϕ_F^t. Then ϕ_F^t is symplectic for all t if and only if F is a Hamiltonian system of the form*

$$F(x) = J\nabla H(x)$$

with $H \in C^2(\mathbb{R}^{2d}, \mathbb{R})$.

Proof: Suppose that ϕ_F^t is symplectic and let $S_F(t)$ be its *sensitivity matrix*:

$$S_F(t) = \frac{\partial \phi_F^t(x_0)}{\partial x_0}.$$

Then we know the evolution of S. If ϕ_F^t is symplectic, then for all $t \in \mathbb{R}$,

$$\begin{aligned} 0 &= \frac{\mathrm{d}}{\mathrm{d}t}\left(S_F^T(t) J S_F(t)\right) = \dot{S}_F^T J S_F + S_F^T J \dot{S}_F \\ &= S_F^T (F')^T J S_F + S_F^T J F' S_F \\ &= S_F^T [(F')^T J + JF'] S_F. \end{aligned}$$

This means that $(F')^T J = -JF' = J^T F' = (F')^T J^T$. The Jacobian of JF is JF'. So JF' is symmetric. Hence there exists $H \in C^2$ such that

$$J^T JF = -J\nabla H(x),$$

or

$$F = J\nabla H(x).$$

Check notes.

Definition 7.2 (Symplectic One-Step Method)**.** A one-step method $\varphi^h : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is *symplectic* if the map φ^h is symplectic whenever applied to a Hamiltonian system.

7.1 Symplectic Splitting Fields

Consider the equation $\dot{x} = F(x)$, for $F : \mathbb{R}^n \to \mathbb{R}^n$ where $F(x) = F_1(x) + F_2(x)$, where we know $\phi^t_{F_1}$ and $\phi^t_{F_2}$.

Is is true that $\phi^t_F = \phi^t_{F_1} \circ \phi^t_{F_2}$? Suppose that

$$F(x) = Ax + Bx.$$

Then

$$\phi^t_{F_1} = e^{At}x, \qquad \phi^t_{F_2} = e^{Bt}x, \qquad \phi^t_F = e^{(A+B)t}x.$$

This condition holds if and only if $[A, B] = 0$.

Example 7.2.

Consider a Hamiltonian system with $x = (q, p) \in \mathbb{R}^2$, and

$$H(q, p) = \frac{1}{2}(q^2 + p^2).$$

This has dynamics $\dot{q} = p$, $\dot{p} = -q$. We can split it into

$$\dot{x} = \begin{pmatrix} p \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ q \end{pmatrix},$$

where the first function is F_1, and the second is F_2. Then if $\dot{x} = F_1(x)$, we must have $p(t) = p(0)$, and $q(t) = q(0) + tp(0)$. Similarly, $\dot{x} = F_2(x)$ has solution $(q_0, p_0 - tq_0)$.

The combination of these two maps does not lead to the overall solution $\ddot{q} = -q$, which is what we want. This is because we do not have $[F_1, F_2] = 0$, in the Lie derivative sense.

However for our purposes, we can approximate

$$\phi^h_F \approx \phi^h_{F_1} \circ \phi^h_{F_2}.$$

In this way, we an let

$$\varphi^h_F = \phi^h_{F_1} \circ \phi^h_{F_2},$$

or swapped. This is known as the *Lie-Trotter splitting method*. There is also *Strang splitting*, where

$$\varphi^h_F = \phi^{h/2}_{F_2} \circ \phi^h_{F_1} \circ \phi^{h/2}_{F_2}.$$

We can show that Strang splitting is a second order method.

Proposition 7.3. *Lie-Trotter has order 1.*

Proof: We have $\phi^h_F(x) = x + hF(x) + \mathcal{O}(h^2)$. Now,

$$\begin{aligned}\varphi^h_F(x) &= \phi^h_{F_1}(\phi^h_{F_2}(x)) = \phi^h_{F_1}(x + hF_2(x) + \mathcal{O}(h^2)) \\ &= x + hF_1(x) + hF_2(x)(x + hF_1(x)) + \mathcal{O}(h^2) \\ &= x + h(F_1(x) + F_2(x)) + \mathcal{O}(h^2) = \phi^h_F(x) + \mathcal{O}(h^2).\end{aligned}$$

Lemma 7.3. *Let $F, G : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ be symplectic. Then $H_1 = F \circ G$, $H_2 = G \circ F$ are symplectic.*

Proof: This is an application of the chain rule. Note that

$$H_1'(x) = F'(G(x))G'(x),$$

so

$$H_1'(x)^T J H_1'(x) = G'(x)^T F'(G(x))^T J F'(G(x))G'(x) = G'(x)^T J G'(x) = J.$$

7.2 Separable Hamiltonian Systems

Consider Hamiltonians of the form

$$H(p, q) = K(p) + U(q).$$

Then we find that

$$J\nabla H(x) = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \begin{pmatrix} \nabla U(q) \\ \nabla K(p) \end{pmatrix}$$

Often times we have $K(p) = \frac{1}{2} p^T M p$, and $U(q) = \sum V(\|q_i - q_j\|)$.

We can write

$$X_H(q, p) = \begin{pmatrix} \nabla K(p) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ -\nabla U(q) \end{pmatrix} = X_K(q, p) + X_U(q, p).$$

From this we can use splitting methods, as we know that

$$\begin{aligned}\phi^t_{X_K}(q_0, p_0) &= (q_0 + t\nabla K(p_0), p_0), \\ \phi^t_{X_U}(q_0, p_0) &= (q_0, p_0 - t\nabla U(q_0)).\end{aligned}$$

Then, we know

$$\varphi^h_{X_H}(q_0, p_0) = \phi^h_{X_K} \circ \phi^h_{X_U}(q_0, p_0)$$

is a symplectic method of order 1. This is known as *symplectic Euler*. Similarly,

$$\varphi^h_{X_H} = \phi^{h/2}_{X_H} \circ \phi^h_{X_U} \circ \phi^{h/2}_{X_H}$$

is symplectic and of order 2. This is the *leapfrog* or *Störder-Verlet* method.

7.3 Symplectic Runge-Kutta Schemes

Proposition 7.4. *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a vector field, and $\varphi^h : \mathbb{R}^n \to \mathbb{R}^n$ be a RK method. Then the following diagram commutes:*

$$\begin{tikzcd}
\begin{cases} \dot{x} = F(x) \\ x(0) = x_0 \end{cases} \arrow[r, "\partial x_n"] \arrow[d, "\varphi^h"] & \begin{cases} \dot{x} = F(x),\, \dot{S} = F'(x)S \\ x(0) = x_0,\, S(0) = I_n \end{cases} \arrow[d] \\
x_1 = \varphi^h(x_0) \arrow[r] & \begin{cases} x_1 = \varphi^h_F(x_0), \\ S_1 = \varphi^h_{F_1}(S_0). \end{cases}
\end{tikzcd}$$

We can increase the order of methods arbitrarily. Suppose ϕ^h is a method of order p. Consider

$$\psi^h = \phi^{\gamma_s h} \circ \cdots \circ \phi^{\gamma_1 h}.$$

Theorem 7.1. *If $\gamma_1 + \cdots + \gamma_s = 1$, and $\gamma_1^{p+1} + \cdots + \gamma_s^{p+1} = 0$, then ψ^h has order $p + 1$.*

This is the *Yoshida trick*.

We return to the proof of the proposition above.

Proof: Let $\varphi^h_F : \mathbb{R}^n \to \mathbb{R}^n$, with tableaux (A, b, c) for $A \in \mathbb{R}^{s\times s}$, $b, c \in \mathbb{R}^s$. Note

$$x_1 = x_0 + h\sum_{i=1}^{s} b_i F(K_i),$$

$$K_i = x_0 + h\sum_{j=1}^{s} a_{ij} F(K_j).$$

Differentiating both of these with respect to x_0, we find

$$\begin{aligned}\frac{\partial x_1}{\partial x_0} &= I_n + h\sum_{i=1}^{s} b_i \frac{\partial F(K_i)}{\partial x_0} \\ &= I_n + h\sum_{i=1}^{s} b_i F'(K_i)\frac{\partial K_i}{\partial x_0}, \\ \frac{\partial K_i}{\partial x_0} &= I_n + h\sum_{j=1}^{s} a_{ij} F'(K_j)\frac{\partial K_j}{\partial x_0}.\end{aligned}$$

This is from going down and right in the diagram. Consider instead going right then down. Then we get $x_1 = \varphi^h(x_0)$, and

$$\begin{aligned}S_1 &= S_0 + h\sum_{i=1}^{s} b_i F'(K_i)K_i, \\ K_i &= S_0 + h\sum_{j=1}^{s} a_{ij} F'(K_j)K_j.\end{aligned}$$

This is enough to show the two paths commute.

We know that, in a Hamiltonian system,

$$\dot{x} = J\nabla H(x), \qquad dotS = J\nabla^2 H(x)S.$$

If $x(t) = \phi^t_{X_H}(x_0)$, then $S^T JS = J$ for all t.

Theorem 7.2 (Sympl-RK Schemes). *Let φ^h be the Runge-Kutta method defined by a tableaux (A, b, c) with $M = BA + A^T b - bb^T = 0$, where $B = \operatorname{diag}(b)$.*

Then φ^h is a symplectic method.

Proof: $\varphi^h : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is symplectic if

$$\left(\frac{\partial \varphi^h(0)}{\partial x}\right)^T J \left(\frac{\partial \varphi^h(x)}{\partial x}\right) = J.$$

But, from the commutation of the diagram,

$$S_1 = \varphi^h_y(S_0) \implies S_1^T JS_1 = J,$$

since φ^h preserves quadratic first integrals.

We have seen previously that we can preserve the energy. Can we preserve the energy and the symplectic form?

7.4 Energy Preservation and Long-Term Simulations

Theorem 7.3. *Let $\dot{x} = J\nabla H(x)$ be a Hamiltonian system with Hamiltonian H, and assume that it has no other conserved quantity. Let φ^h be a symplectic and energy-preserving method for the system. Then φ^h reproduces the exact solution up to a time reparametrisation.*

8 Backward Error Analysis

In regular error analysis, we let $x_1 = \varphi^h(x_0)$, $x(h) = \phi_F^h(x_0)$, and if $\|x_1 - x(h)\| \in \mathcal{O}(h^{p+1})$, then φ^h is of order p.

Another way of doing error analysis is as follows. Let us find the vector field $F_h : \mathbb{R}^d \to \mathbb{R}^d$ such that $\dot{y} = F_h(y)$, with $y(h) = \varphi^h(y(0))$, and

$$y(nh) = \varphi^h \circ \cdots \circ \varphi^h(y(0)).$$

To do this, we use a series expansion:

$$F_h(x) = F(x) + hF_2(x) + h^2F_3(x) + \cdots$$

Then since $\dot{y} = F_h(y)$, expanding we see

$$\begin{aligned} y(t+h) = y(t) &+ h(F(y) + hF_2(y) + \cdots) \\ &+ \frac{h^2}{2}(F'(y) + hF_2'(y) + \cdots) + \cdots \end{aligned}$$

Suppose that

$$y(t+h) = \varphi^h(y(t)) = \phi_F^h(y(t)) + \mathcal{O}(h^{p+1}).$$

If φ^h is a method of order p, then

$$F_h(y) = F(y) + h^pF_{p+1}(y) + h^{p+1}F_{p+2}(y) + \mathcal{O}(h^{p+2}).$$

Theorem 8.1. *Let φ^h be a symplectic method of order p applied to the Hamiltonian system $\dot{x} = J\nabla H(x)$, with $H : \mathbb{R}^{2d} \to \mathbb{R}$. Then the modified equation $\dot{y} = F_h(y)$ is also Hamiltonian.*

So if H is a smooth function, then there exist smooth functions $H_{p+1}, H_{p+2}, \ldots : \mathbb{R}^{2d} \to \mathbb{R}$ such that

$$F_h(x) = J(\nabla H(x) + h^p\nabla H_{p+1}(x) + h^{p+1}\nabla H_{p+2}(x) + \mathcal{O}(h^{p+2})).$$

Proof: Assume that $F_i(y) = J\nabla H_i(y)$ for $i = p+1, \ldots, R$. We now show that $F_{R+1}(y) = J\nabla H_{R+1}(y)$. We define the *truncated modified vector field* as

$$F_{h,R}(x) = F(x) + h^pF_{p+1}(x) + h^{p+1}F_{p+2}(x) + \cdots + h^{R-1}F_R(x).$$

By the assumption, $F_{h,R}$ is Hamiltonian. We call ϕ_R^t the flow at time t of

$F_{h,R}$. We can say that

$$\phi^h_{F_h}(x) = \phi^h_R(x) + h^{R+1}F_{R+1}(x) + \mathcal{O}(h^{R+2}).$$

Then note that

$$\phi'_{F_h}(x) = (\phi^h_R)'(x) + h^{R+1}F'_{R+1}(x) + \mathcal{O}(h^{R+2}),$$

and since ϕ_{F_h} is symplectic,

$$\begin{aligned} J = (\phi'_{F_h}(x))^T J(\phi'_{F_h}(x)) &= (\phi^{h'}_R(x))^T J(\phi^{h'}_R(x))^T + h^{R+1}(\phi^{h'}_R(x))^T JF'_{R+1}(x) \\ &\quad + h^{R+1}F'_{R+1}(x)^T J(\phi^{h'}_R(x)) + \mathcal{O}(h^{R+2}). \qquad (*) \end{aligned}$$

Note that

$$\phi^h_R(x) = x + hF_{h,R}(x) + \mathcal{O}(h^2) \implies \phi^{h'}_R = I + \mathcal{O}(h),$$

so $(*)$ becomes

$$J + h^{R+1}(JF'_{R+1}(x) + F'_{R+1}(x)^T J) + \mathcal{O}(h^{R+2}).$$

Looking at the order $R+1$ terms,

$$JF'_{R+1}(x) = -F'_{R+1}(x)^T J = F'_{R+1}(x)J^T,$$

since $J^T = -J$. Hence $JF_{R+1}(x) = -\nabla H_{R+1}(x)$ for some energy function H_{R+1}, and

$$F_{R+1}(x) = -J^T\nabla H_{R+1}(x) = J\nabla H_{R+1}(x).$$

So F_{R+1} is Hamiltonian, as desired.

We let

$$\tilde{H}_N(x) = H(x) + h^p H_{p+1}(x) + \cdots + h^{N-1}H_N(x)$$

be the truncated modified Hamiltonian for a symplectic integration φ^h of order p applied to $\dot{x} = J\nabla H(x)$.

Theorem 8.2. *Consider a Hamiltonian system with analytic $H : D \to \mathbb{R}$, for $D \subseteq \mathbb{R}^{2d}$, and apply a symplectic method φ^h with stepsize $h > 0$. If the numerical solution stays in a compact subset $K \subseteq D$, then there exists $h_0 > 0$ and an $N = N(h)$ such that:*

- *$\tilde{H}_n(y_n) = \tilde{H}_n(x_0) + \mathcal{O}(e^{-h_0/2h})$, where $y_N = \varphi^h \circ \cdots \circ \varphi^h(x_0)$.*
- *$H(y_n) = H(x_0) + \mathcal{O}(h^p)$, where $nh \leq e^{h_0/2h}$.*

9 Non-expansive and Deep Learning Theory

Let $N_\theta : \mathbb{R}^d \to \mathbb{R}^c$, for $\theta \in \Theta \subseteq \mathbb{R}^p$. Suppose our dataset is $\mathcal{T} = \{(x_i, y_i)\}$, and we would like to have $N_\theta(x_i) = y_i$. The simplest loss is

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|N_\theta(x_i) - y_i\|^2.$$

Suppose that

$$N_\theta = F_{\theta_L} \circ \cdots \circ F_{\theta_1},$$

where L is the number of layers. Let $\theta = (\theta_1, \ldots, \theta_L)$. We call θ_{ij} the i'th component of θ_j. Then we let

$$\theta_{ij}^{K+1} = \theta_{ij}^{K} = \frac{\tau_K}{N} \sum_{n=1}^{N} \partial_{\theta_{ij}} L_n(\theta^K),$$

where

$$L_n(\theta) = \|N(x_n) - y_n\|^2.$$

We set $x^{j+1} = F_{\theta_j(x^j)}$, and $x^1 = x$. So, $N_\theta(x^1) = x^{L+1}$. Note that

$$\partial_{\theta_{ij}} L_n = \langle \partial_{x_n^{j+1}} L_n, \partial_{\theta_{ij}} x_n^{j+1} \rangle = \left\langle \left(\prod_{l=j+1}^{L} \partial_{x_n^l} x_n^{l+1} \right) \partial_{x_n^{l+1}} L_n, \partial_{\theta_{ij}} x_n^{j+1} \right\rangle.$$

But by CS, we can bound this norm: it is at most

$$\left\| \prod_{l=j+1}^{L} \partial_{x_n^l} x_n^{l+1} \right\|_2 \le \prod_{l=j+1}^{L} \|\partial_{x_n^l} x_n^{l+1}\|_2 \to 0.$$

This is the *vanishing gradient problem.*

We know that

$$\partial_{x_n^l} x_n^{l+1} = \partial_{x_n^l} F_{\theta_l}(x_n^l).$$

Suppose F is some symplectic map, say $\varphi_{X_{\theta_l}}^{h_l}$. Then we know

$$\|J\|_2 = \|F'(x)^T J F'(x)\|_2 \le \|F'(x)\|_2^2 \cdot \|J\|_2,$$

hence $\|F'(x)\|_2 \ge 1$. So we do not have a vanishing gradient. Suppose the l'th layer has Hamiltonian

$$H_{\theta_l}(x) = \langle 1, \gamma(A_x + b) \rangle.$$

Then note

$$\nabla H_{\theta_l}(x) = A^T \sigma(Ax + b),$$

where $\sigma = \gamma'$. This is like a single hidden-layer network, but which is forced to have zero Jacobian. Then we can let

$$F_{\theta_l}(x) = \varphi^{h_l}_{X_{H_{\theta_l}}}(x).$$

Note that if $F_{\theta_l}(x) = x + h_l J \nabla H_l(x)$, this is not necessarily symplectic, as we have integrated with a non-symplectic integrator.

So we would like to integrate with a symplectic integrator. This is possible if $H(q,p) = K(p) + U(q)$, in which case we can use symplectic Euler:

$$\varphi^h_{X_H} = \phi^h_{X_K} \circ \phi^h_{X_u}.$$

This is possible if we constrain the weights A to be block-diagonal, i.e.

$$A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}.$$

Then we get

$$H_{\theta_l}(x) = \langle 1, \gamma(A_1 q + b_1)\rangle + \langle 1, \gamma(A_2 q + b_2)\rangle.$$

In this case,

$$X_{H_{\theta_l}}(x) = \begin{pmatrix} A_2^T \sigma(A_2 p + b_2) \\ -A_1^T \sigma(A_1 q + b_1) \end{pmatrix}.$$

10 Non-expansive Dynamical Systems

Consider a system $\dot{x} = F(x)$, such that for some norm $\|\cdot\|$ and all $t \geq 0$,

$$\|\phi_F^t(x) - \phi_F^t(y)\| \leq \|x - y\|.$$

Such a system is called *non-expansive*. This norm does not need to be the Euclidean norm. For $0 < h \ll 1$, we have

$$\begin{aligned}\phi_F^{t+h}(x) &= \phi_F^t(x) + hF(\phi_F^t(x)) + \mathcal{O}(h^2),\\ \phi_F^{t+h}(y) &= \phi_F^t(y) + hF(\phi_F^t(y)) + \mathcal{O}(h^2).\end{aligned}$$

We consider the norm $\|\cdot\|$ defined by the inner product. Then, the difference in the norms are

$$2h\langle F(\phi_F^t(y)) - F(\phi_F^t(x)), \phi_F^t(x) - \phi_F^t(y)\rangle + \mathcal{O}(h^2).$$

For this to always be non-positive for $h > 0$, we need that this object is negative. Set $g(t) = \|\phi_F^t(y) - \phi_F^t(x)\|^2$, and

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(e^{-2\nu t}g(t)) &= e^{-2\nu t}\dot{g}(t) - 2\nu e^{-2\nu t}g(t)\\ &= e^{-2\nu t}(\dot{g}(t) - 2\nu g(t)).\end{aligned}$$

Suppose that there is $\nu \in \mathbb{R}$ such that

$$\langle F(y) - F(x), y - x\rangle \leq \nu\|y - x\|^2.$$

Then we find

$$\frac{\mathrm{d}}{\mathrm{d}t}g(t)^2 \leq 2\nu g(t)^2 \implies \frac{\mathrm{d}}{\mathrm{d}t}(e^{-2\nu t}g(t)) \leq 0.$$

Therefore, $e^{-2\nu t}g(t) \leq g(0)$, so

$$\|\phi_F^t(y) - \phi_F^t(x)\| \leq e^{\nu t}\|y - x\|.$$

Definition 10.1. The vector field $F : \mathbb{R}^d \to \mathbb{R}^d$ is *one-sided Lipschitz continuous* if it satisfies

$$\langle F(u) - F(v), u - v\rangle \leq \nu\|u - v\|^2$$

for all $u, v \in \mathbb{R}^d$ for a scalar $\nu \in \mathbb{R}$. F is *non-expansive* if $\nu \leq 0$, and *contractive* if $\nu < 0$.

Lemma 10.1. *An L-Lipschitz continuous vector field $F : \mathbb{R}^d \to \mathbb{R}^d$ is also one-sided Lipschitz continuous.*

Proof: We have

$$\langle F(u)-F(v),u-v\rangle \le \|F(u)-F(v)\|\cdot\|u-v\| \le \mathrm{Lip}(F)\|u-v\|^2.$$

If $F:\mathbb{R}^d\to\mathbb{R}^d$ is C^1, then for all x,y by MVT we know there is $z=sx+(1-s)y$ such that $F(y)-F(x)=F'(z)(y-x)$. Hence

$$\langle F(y)-F(x),y-x\rangle = \langle F'(z)(y-x),y-x\rangle.$$

Suppose that we know that

$$\sup_{\substack{x\in\mathbb{R}^d\\ v\in\mathbb{R}^d\setminus\{0\}}} \frac{\langle F'(x)v,v\rangle}{\|v\|^2} \le \nu.$$

If this is the regular inner product, this is equivalent to

$$\sup_{x\in\mathbb{R}^d} \lambda_{max}\left(\frac{F'(x)+F'(x)^T}{2}\right) \le \nu.$$

For $d=1$, this corresponds to $F'(x)\le\nu$. For example, $F(x)=-x^3$ is one-sided Lipschitz continuous, but not Lipschitz continuous.

Suppose our norm is generated by an inner product.

Definition 10.2. A numerical method $\varphi^h:\mathbb{R}^d\to\mathbb{R}^d$ is *B-stable* if, when applied to any vector field $F:\mathbb{R}^d\to\mathbb{R}^d$ which satisfies

$$\|\phi_F^t(y)-\phi_F^t(x)\| \le \|y-x\|$$

for all $t\ge 0$ and x,y, then for all $h\ge 0$,

$$\|\varphi_F^h(y)-\varphi_F^h(x)\| \le \|y-x\|.$$

Remark. Recall our equation $\dot{x}=\lambda x$, for $\lambda=\alpha+i\beta$ with $\Re(\lambda)=\alpha$. This is equivalent to

$$\dot{u}=\begin{pmatrix}\alpha & -\beta\\ \beta & \alpha\end{pmatrix}u=F(u),$$

with

$$F'(u)=\begin{pmatrix}\alpha & -\beta\\ \beta & \alpha\end{pmatrix}, \qquad \frac{F'(u)+F'(u)^T}{2}=\alpha I.$$

This is contractive. Hence we see that since no explicit RK method is A-stable, the same is true for B-stability.

Proposition 10.1. *A RK method φ^h of tableaux (A,b,c) is B-stable if, when we write $B=\mathrm{diag}(b)$ and $M=BA+A^TB-bb^T$, they are both symmetric and positive semi-definite.*

Proof: Let

$$x_1 = \varphi^h(x_0) + h \sum_{i=1}^{s} b_i F(K_i),$$
$$K_i = x_0 + h \sum_{j=1}^{s} a_{ij} F(K_j).$$

Then we can write similarly for y_1 and L_i. We write $\delta x_R = y_R - x_R$, $\delta F_i = F(K_i) - F(L_i)$ and $\delta K_i = K_i - L_1$. Then

$$\|\delta x_1\|^2 = \langle \delta x_1, \delta x_1 \rangle = \|\delta x_0\|^2 + h^2 \sum_{i,j=1}^{s} b_i b_j \langle \delta F_i, \delta F_j \rangle + 2h \sum_{i=1}^{s} b_i \langle \delta x_0, \delta F_i \rangle.$$

We can write, by linearity,

$$\delta x_0 = \delta K_i - h \sum_{j=1}^{s} a_{ij} \delta F_j.$$

This can be plugged in to get

$$\begin{aligned} \|\delta x_1\|^2 = \|\delta x_0\|^2 + h^2 \sum_{i,j=1}^{s} b_i b_j \langle \delta F_i, \delta F_j \rangle + 2h \sum_{i=1}^{s} b_i \langle \delta K_i, \delta F_i \rangle \\ - 2h^2 \sum_{i,j=1}^{s} b_i a_{ij} \langle \delta F_i, \delta F_j \rangle, \\ \implies \|\delta x_1\|^2 \le \|\delta x_0\|^2 - h^2 \sum_{i,j=1}^{s} m_{ij} \langle \delta F_i, \delta F_j \rangle, \end{aligned}$$

using the fact that $\langle \delta K_i, \delta F_i \rangle \le 0$.

Note that $M \ge 0$, and rewriting this we get

$$\|\delta x_1\|^2 - \|\delta x_0\|^2 = -h^2 \delta F^T M \delta F \le 0.$$

So this method is B-stable.

10.1 Conditionally Non-linearly Stable Methods

Consider $V(x) = \|x\|_2^2/2$. Then $\dot{x} = -\nabla V(x) = -x$ gives $x(t) = e^{-t}x(0)$. Hence

$$\|x(t) - y(t)\|_2 = e^{-t}\|x(0) - y(0)\| < \|x(0) - y(0)\|_2.$$

So if we let

$$x_{n+1} = \varphi^h(x_n) = x_n - h\nabla V(x_n) = (1-h)x_n,$$

then

$$\|y_{n+1} - x_{n+1}\|_2 = |1-h|\|y_n - x_n\|.$$

If $h \in [0,2]$, then φ^h is non-expansive, and 1-Lipschitz as well. See circle contractivity for more.

Definition 10.3. A convex and continuously differentiable function $V : \mathbb{R}^d \to \mathbb{R}$ is *L-smooth* if its gradient is L-Lipschitz, i.e.

$$\|\nabla V(y) - \nabla V(x)\| \leq L\|y - x\|.$$

Theorem 10.1 (Baillon-Haddad). *V is L-smooth if and only if*

$$\langle \nabla V(x) - \nabla V(y), x - y\rangle \geq \frac{1}{L}\|\nabla V(x) - \nabla V(y)\|_2^2.$$

We now consider $V : \mathbb{R}^d \to \mathbb{R}$ L-smooth, and the dynamics $\dot{x} = -\nabla V(x) = F(x)$. Then

$$\langle F(x) - F(y), x - y\rangle \leq -\frac{1}{L}\|F(x) - F(y)\|_2^2.$$

Consider again $x_{n+1} = x_n - h\nabla V(x_n)$, $y_{n+1} = y_n - h\nabla V(y_n)$. Then

$$\begin{aligned}\|y_{n+1} - x_{n+1}\|_2^2 &= \|y_n - x_n\|_2^2 + h^2\|\nabla V(y_n) - \nabla V(x_n)\|_2^2 - 2h\langle \nabla V(y_n) - \nabla V(x_n), y_n - x_n\rangle \\ &\leq \|y_n - x_n\|_2^2 + h\left(h - \frac{2}{L}\right)\|\nabla V(y_n) - \nabla V(x_n)\|.\end{aligned}$$

If $h \leq 2/L$, then $\|x_{n+1} - y_{n+1}\| \leq \|x_n - y_n\|$, and hence exponential Euler is non-expansive when applied to $\dot{x} = -\nabla V(x)$ for V L-smooth.

If x^* is the fixed point of the flow, i.e. it is the minimum of $V(x)$, then

$$\|x(t) - x^*\|_2 \leq \|x(0) - x^*\|_2.$$

Returning to our example $V(x) = \|x\|^2/2$, then this is L-smooth with $L = 1$, hence $h \leq 2/L = 2$ is non-expansive.

10.2 1-Lipschitz and Non-expansive Neural Networks

Consider the setting of an inverse problem, where one wants to build denoising algorithms with convergent behaviour.

Suppose we want to reconstruct X. Then we want to find

$$\min_Y \|X - Y\|^2 + R(Y),$$

where R is a regularization term. R can be learnt using a neural network. We can learn Y using gradient descent. If $V = V_1 + V_2$, then

$$\dot{X} = -\nabla V_1(X) - \nabla V_2(X),$$

and so we can step

$$X_{n+1} = \varphi^2 \circ \varphi^1 \circ (X_n).$$

Instead of learning V_2 or R, we can learn φ^2 directly. If φ^2 has non-expansive guarantees, then this will be convergent.

Another setting is robust classification. Suppose we have a bunch of images, and we want to learn a classifier associating each image to a label. We want the network to be not too sensitive to perturbations.

We have $\Omega \subseteq \mathbb{R}^d$, and $\ell : \Omega \to \{1, \ldots, C\}$, where C is the number of classes of the set. If $\Omega_i = \ell^{-1}(i)$, then

$$\Omega = \bigcup_{i=1}^{C} \Omega_i, \qquad \Omega_i \cap \Omega_j = \emptyset.$$

Define a network $N_\theta : \mathbb{R}^d \to \mathbb{R}^C$, where $\hat{K}(x)$ is the prediction of N_θ for $x \in \Omega$. So

$$\hat{K}(x) = \text{argmax} N_\theta(x)^T e_k.$$

Typically we do softmax, i.e. $N_\theta(x) \to y \in \mathbb{R}^C$ with

$$y_k = \frac{\exp(N_\theta(x)_k)}{\sum \exp(N_\theta(x)_i)}.$$

Suppose we have a Lipschitz constant of N_θ, so

$$\|N_\theta(y) - N_\theta(x)\|_2 \le l\|y - x\|_2$$

for all $x, y \in \mathbb{R}^d$. This is not too good of a robustness measure. Instead, we desire a good *margin of classification*:

$$m(x) = N_\theta(x)_{\hat{K}(x)} - \max_{K \neq \hat{K}} N_\theta(x)_K.$$

Proposition 10.2. *Let us consider an l-Lipschitz neural network $N_\theta : \mathbb{R}^d \to \mathbb{R}^C$. Then for every $y \in \mathbb{R}^d$ such that*

$$\|x - y\|_2 \leq \frac{m(x)}{\sqrt{2}l},$$

the prediction $\hat{K}(y)$ will coincide with $\hat{K}(x)$.

10.3 1-Lipschitz Neural Networks based on Gradient Flows

Consider $N_\theta = F_{\theta_L} \circ \cdots \circ F_{\theta_1}$, where $F_{\theta_i}(x) = B_i \sigma(A_i x + b_i)$, with $\|B_1\|2 = \|A_i\|_2$ and $\mathrm{Lip}(\sigma) \leq 1$.

If instead we have a ResNet $F_{\theta_i}(x) = x + B_i \sigma(A_i x + b_i)$, then

$$\|F_{\theta_i}(x) - F_{\theta_i}(y)\| \leq (1 + \|B_i\|_2 \|A_2\|_2 \mathrm{Lip}(\sigma)) \|x - y\|.$$

We call $F_{\theta_i} = \varphi^h_{X_i}$, where $X_i(x) = -\nabla V_i(x)$. If $V_i(x) = \mathbf{1}^T \gamma(A_i x + b_i)$ where $\gamma' = \sigma$, then $\nabla V_i(x) = A_i^T \sigma(A_i x + b_i)$.

Two commons forms of σ are

$$\begin{aligned} &\sigma(x) = \mathrm{ReLu}(x) = \max(0, x) = x^+, \\ &\sigma(x) - \mathrm{LeakyReLu}(x) = \max(ax, x) \qquad a \in (0, 1). \end{aligned}$$

We can then set $\gamma = \int \sigma$.

Index