1 Preliminaries

For population $N(t)$, and max supported pop N^*. Let $n = \frac{N}{N^*}$.

Logistic model:

$$\frac{dn}{dt} = rn(1-n)\Bigg|\,\text{General Solution} : n(t) = \frac{ce^{rt}}{1+ce^{rt}}.$$

Logistic model (with predation): $\frac{dn}{dt} = rn(1-\frac{n}{s}) - \frac{n^2}{1-n^2}$

Lotka-Volterra predator-prey model: n = prey population, p = predator population:

$$\frac{dn}{dt} = rn(1-p), \quad \frac{dp}{dt} = p(1-n)$$

Converting non-autonomous systems to autonomous:

For non-autonomous system of d equations:

$\frac{dy_1}{dt} = f_1(t, y_1, ...y_d)$,
$\frac{dy_2}{dt} = f_2(t, y_1, ...y_d)$,
$\vdots$
$\frac{dy_d}{dt} = f_d(t, y_1, ...y_d)$.

We can make it into an autonomous one by introducing the variable $y_{d+1} \equiv t$ in place of t, (thus creating an autonomous system of $d+1$ variables) and introduce new indep var s such that $ds/dt = 1$:

$\frac{dy_1}{ds} = f_1(t, y_1, ...y_{d+1})$,
$\frac{dy_2}{ds} = f_2(t, y_1, ...y_{d+1})$,
$\vdots$
$\frac{dy_d}{ds} = f_d(t, y_1, ...y_{d+1})$,
$\frac{dy_{d+1}}{ds} = 1$.

Picard's Thm: let the function $f(\cdot,\cdot)$ be a continuous function of its arguments in a region of the plane containing the rectangle $D = \{(t,y) : t_0 \le t \le T, |y - y_0| \le K\}$, $T, K > 0$. Suppose $\exists L$ (Lipschitz constant) such that $|f(t,u) - f(t,v)| \le L|u-v|$ whenever $(t,u),(t,v) \in D$. Since D closed & bounded, $\exists M_f > 0$ such that $M_f = \max\{|f(t,u)| : (t,u) \in D\}$. Assume $M_f(T - t_0) \le K$. Then there exists a unique continuously differentiable function $t \mapsto y(t)$, defined on $[t_0, T]$, such that $\frac{dy}{dt} = f(t,y), y(t_0) = y_0$.

Note: $\frac{\partial f}{\partial y}$ being continuous $\Rightarrow$ existence of a Lipschitz constant, We can take $L = \sup_D \left[\frac{\partial f}{\partial y}\right]$.

Thm 1.3.2 (Local Existence / Uniqueness of Solutions for ODE Systems): Suppose $f : \mathbb{R} \to \mathbb{R}^d \times \mathbb{R}^d$ is continuous and has continuous partial derivatives w.r.t. all components of the dependant var y in a neighbourhood of the point (t_0, y_0). Then there is an interval $I = (t_0 - \delta, t_0 + \delta)$ containing a unique function y (continuously differentiable on I) that satisfies $\frac{dy}{dt} = f(t,y), y(t_0) = y_0$.

2 Euler's Method and Taylor Series Methods

Euler's Method: for finding approximate solutions to $\frac{dy}{dt} = f(t,y)$ in time interval $t \in [a,b]$.

Let $t_m = a + mh$, $h \in \mathbb{R}$ small. Start with $y_0 = y(t_0) = y(a)$ and use iteration function $\boxed{y_{n+1} = y_n + hf(t_n, y_n)}$. $y_i \approx y(t_i)$.

Error: $\boxed{e_n = |y_n - y(t_n)|}$.

Lemma 2.6.1: Suppose a given sequence of non-negative numbers (v_n) satisfies $v_{n+1} \le Av_n + B$, $A > 1, B > 0$. Then for $n = 0, 1, 2, ...,$ $\boxed{v_n \le A^n v_0 + \frac{A^n - 1}{A - 1}B}$.

Bound on error: $\boxed{|e_n| \le e^{(b-a)L}\frac{M}{2L}h = Dh}$ where h is the timestep, M bounds $|y''(t)|$, and L bounds $\frac{\partial y}{\partial t}$.

Thm 2.6.1 Consider the IVP $\frac{dy}{dt} = f(t,y), y(a) = y_0$. Suppose $\exists$ unique, twice-differentiable function f, continuous everywhere with continuous, bounded partial derivative $\left|\frac{\partial f}{\partial y}\right| < L$ with $L > 0$.Then for $n = 0, 1, ..., N$, and some $D > 0$, the solution y_n given by Euler's method at t_n satisfies $\boxed{e_n = |y_n - y(t_n)| \le Dh}$ where $h = (b-a)/N, t_n = a + hn$.

Big O (Landau) notation: If method is $O(h^p)$ then the error decays at least as quickly as h^p (for small h).

$\boxed{(\exists\, h_0, C > 0) : (\forall\, 0 < h < h_0), |z| \le Ch^p \Rightarrow z = O(h^p)}$

Say z is of order p.

The Flow Map: Consider IVP $\frac{dy}{dt} = f(t,y), y(a) = y_0$ with unique solution in $t \in [a,b]$. Starting at arbitrary $t_0 \in [a,b], y_0$ we may see where $y(t)$ ends up, denoted $y(t; t_0, y_0)$.

Fix t_0 and look at the **flow map** $\boxed{\Phi_{t_0,h}(y_0) = y(t_0 + h; t_0, y_0)}$.

(*actually a family of maps parameterised by* h).

Numerical methods approximate flow maps: Euler's method approximates flow map with $\hat{\Phi}_{t,h}(y) = y + hf(t,y)$.

One-step methods: approximate the solution through the iteration of an approximated flow map.

Constructing Taylor series methods:

Start with Taylor series:

$y(t_0 + h) = y(t_0) + y'(t_0)h + \frac{1}{2}y''(t_0)h^2 + \frac{1}{6}y'''(t_0)h^3 + ...$

Also $y'(t) = f(t,y) \Rightarrow$

$y'' = \frac{d}{dt}f(t,y) = \frac{\partial}{\partial t}f(t,y)\frac{dt}{dt} + \frac{\partial}{\partial y}f(t,y)\frac{dy}{dt}$

$= \frac{\partial}{\partial t}f(t,y) + \frac{\partial}{\partial y}f(t,y)y' = f_t + f_y f$. (By chain rule)

$\Phi_{t,h}(y) = y + hf(t,y) + \frac{1}{2}h^2(f_t(t,y) + f_y(t,y)f(t,y)) + \frac{1}{6}y'''h^3 + ...$

Which we can truncate to get the 2nd order Taylor series method

$\boxed{\hat{\Phi}_{t,h}(y) = y + hf(t,y) + \frac{1}{2}h^2(f_t(t,y) + f_y(t,y)f(t,y))}$.

3 Convergence of One-Step Methods

Def 3.1.2 (Convergence): A method is said to be **convergent** iff for any T,

$$\lim_{\substack{h\to 0\\ h=T/N}} \max_{n=0,1,...,N} \|e_n\| = 0.$$

Def 3.2.1 (Local Error): The **local error** of a one-step method is the difference between the flow map Φ_h and it's descrete approximation Ψ_h

$$le(y,h) = \Psi_h(y) - \Phi_h(y).$$

It measures how much error is introduced in a single timestep of size h.

Def 3.2.2 (Consistency): Suppose the local error for our method satisfies

$$\|le(y,h)\| \le Ch^{p+1}$$

where C is a constant that depends on $y(t)$ and it's derivatives, and $p \ge 1$. Then the method is **consistent** at order p.

Def 3.2.3 (Stability): Suppose that a method satisfies an $h-$independant Lipschitz condition on D (spatial domain)

$$\|\Psi_h(u) - \Psi_h(v)\| \le (1 + h\hat{L})\|u - v\| \quad \forall\, u, v \in D.$$

Then the method is **stable**. *Note* $\hat{L}$ *need not be the same Lipschitz constant as for the vector field.*

Thm 3.2.1 (Convergence of One-Step Methods): Given a differential equation and a one-step method Ψ_h which is **consistent** and **stable**. Then the method is **convergent**.

Interpolating Polynomials: Given s distinct *abscissa points* $c_0, ...c_s$ and *data points* $g_0, ..., g_s$, there exists a unique interpolating polynomial $P(x) \in \mathbb{P}_{s-1}$ passing through all points (c_i, g_i).

Lagrange Polynomials: For a set of abscissae $c_0, ...c_s$, the Lagrange polynomials $\ell_i,\ i = 1, ..., s$ are defined by

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

The Lagrange polynomial ℓ_i is the interpolating polynomial through the data $g_j = \begin{cases} 1 \text{ if } j = i \\ 0 \text{ if } j \neq i \end{cases}$. $\{\ell_i\}$ form a basis for $\mathbb{P}_{s-1}$, and any polynomial $Q(x)$ has the simple form

$$Q(x) = \sum_{i=1}^{s} Q(c_i)\ell_i(x) = \sum_{i=1}^{s} g_i \ell_i(x).$$

Numerical Quadrature: Given a smooth function $g(x) : \mathbb{R} \to \mathbb{R}$, and s **quadrature points** $0 \leq c_1 < ... < c_s \leq 1$ we can estimate $\int_0^1 g(x)\, dx$ by integrating the corresponding interpolating polynomial $P(x) \in \mathbb{P}_{s-1}$. Define the weights

$$b_i = \int_0^1 \ell_i(x)\, dx.$$

Then our approximate integral is

$$\int_0^1 g(x)\, dx \approx \int P(x)\, dx = \sum_{i=1}^{s} g(c_i) \int_0^1 \ell_i(x)\, dx = \sum_{i=1}^{s} b_i g(c_i)$$

Therefore for interval $[t_0, t_0 + h]$ we have

$$\int_{t_0}^{t_0+h} g(x)\, dx \approx \int_{t_0}^{t_0+h} P(x)\, dx = \sum_{i=1}^{s} b_i g(t_0 + hc_i).$$

A quadrature rule has **order** p if it integrates any polynomial $\in \mathbb{P}_{p-1}$ exactly. We always have $p \geq s$, and for optimal choice of c_i we have $p = 2s$.

One-Step Collocation: Given an ODE we wish to construct the **collocation polynomial** $u(t) \in \mathbb{R}^d$ that satisfies

$$u(t_0) = y_0 u'(t_0 + c_i h) = f(u(t_0 + c_i h)).$$

In particular, it agrees with our solut at t_0, and it's derivative matches that of the solution at each $c_1, ..., c_s$. We can use such a polynomial to approximate a numerical solution to our ODE by decomposing it's derivative u' into Lagrange polynomial components, and then integrating over $[t_0, t_0 + h]$ to get $u(t_0 + h) = u(t_1)$.

Let $F_i = u'(t_0 + hc_i)$ be the value of the derivative of the polynomial at node c_i. Then

$$F_i = f(y_0 + h\sum_{j=1}^{s} a_{ij} F_j),\ \text{(A)}$$

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i F_i.\ \text{(B)}$$

where

$$a_{ij} = \int_0^{c_i} \ell_j(x)\, dx,$$

$$b_i = \int_0^1 \ell_i(x)\, dx.$$

First solve the sd-dimensional system of non-linear equations given by (A), and plug into (B).

Rem 3.6.1 (Continuous Approximations): Collocation provides a continuous approximation $u(t)$ of the solution $y(t)$ on each interval $[t_n, t_{n+1}]$.

Rem 3.6.2 (Optimal Node Placement): For optimal order of accuracy, use **Gauss-Legendre** collocation methods. This means placing nodes at roots of shifted Legendre polynomials. For $s = 1, 2, 3$, the optimal nodes are

$$c_1 = \frac{1}{2}, \quad p = 2,$$

$$c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6},\ c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}, \quad p = 4,$$

$$c_1 = \frac{1}{2} - \frac{\sqrt{15}}{10},\ c_2 = \frac{1}{2},\ c_3 = \frac{1}{2} + \frac{\sqrt{15}}{10}, \quad p = 6.$$

Runge-Kutta Methods (Autonymous Case): Generalisation of collocation methods, since they don't have coefficients that rely on integrals of Lagrange polynomials, they can have any coefficients. A **Runge-Kutta** method is any method of form

$$Y_i = y_n + h\sum_{j=1}^{s} a_{ij} f(Y_j), \quad i = 1, ..., s,$$

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i f(Y_i).$$

where s is the **number of stages**, b_i are the **weights**, and a_{ij} are the **internal coefficients**. Such a method generates the discrete flow-map

$$\Psi_h(y) = y + h\sum_{i=1}^{s} b_i f(Y_i(y, h)).$$

Butcher Tables: Store coefficients of Runge-Kutta methods in form

$$\begin{array}{c|c} \text{c} & \text{A} \\ \hline & b^T \end{array}$$

where $c = (c_i),\ b = (b_j),\ A = (a_{ij})$. If A is lower triangular (with 0s on diag) then the method is **explicit**, otherwise **implicit**.

Butcher Table Examples:

Euler Method (Explicit):

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}.$$

Trapezoidal (Implicit):

$$\begin{array}{c|cc} 0 & & \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}.$$

Order Conditions for RK Methods:

Consider an RK method with $b = (b_j),\ A = (a_{ij})$. For the order to be p, the following conditions must be satisfied (as well as any conditions for it to be $< p$):

$$p = 1 \Rightarrow \sum_{i=1}^{s} b_i = 1,$$

$$p = 2 \Rightarrow \sum_{i=1}^{s} b_i c_i = \frac{1}{2}, \quad \text{That is } b^T c = \frac{1}{2},$$

$$p = 3 \Rightarrow \sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3} \text{ And } \sum_{i=1}^{s}\sum_{j=1}^{s} b_i a_{ij} c_j = \frac{1}{6}.$$

There is no explicit RK method of order greater than 4. For best order, use Gauss-Legendre methods, which have order $p = 2s$.