Genetic algorithms: A possible migration model

Master degree in Modelling for Science and Engineering

December 3, 2021

1 Introduction

At La Banya, gulls (both Audouin's gulls and their main competitor yellow-legged gulls) build their nest in clumped groups (what is called a sub-colony). Censuses are performed yearly depending on the size of the sub-colony.

The obtained data is shown in the following table:

year	pop.	year	pop.	year	pop.	year	pop.
1981	36	**1990**	4300	**1999**	10189	**2008**	13031
1982	200	**1991**	3950	**2000**	10537	**2009**	9762
1983	546	**1992**	6174	**2001**	11666	**2010**	11271
1984	1200	**1993**	9373	**2002**	10122	**2011**	8688
1985	1200	**1994**	10143	**2003**	10355	**2012**	7571
1986	2200	**1995**	10327	**2004**	9168	**2013**	6983
1987	1850	**1996**	11328	**2005**	13988	**2014**	4778
1988	2861	**1997**	11725	**2006**	15329	**2015**	2067
1989	4266	**1998**	11691	**2007**	14177	**2016**	1586
						2017	793

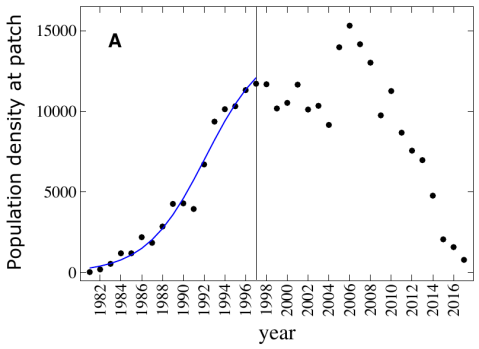

Table 1: The Andouin's population data at La Banya from 1981 to 2017. Predators (foxes) appeared in 1997. In 2005 there is an abrupt change (at the moment without explanation) provoked by external reasons. The period 1981–1997 known as *First Epoch* is characterised by a logistic growth due to the absence of predators and the fact that the population did not exhaust the food carrying capacity. The period 2006–2017 known as *Second Epoch* is characterised by a regular behaviour with migration.

Several biotic and abiotic drivers can influence population fluctuations at the study patch. However, previous studies show that local biotic drivers explain better these fluctuations than global oceanographic indexes. Among the these biotic drivers, interference competition with the dominant yellow-legged gull and predation and disturbance by invasive carnivores (mainly foxes) are the main factors affecting all crucial demographic parameters, namely adult survival, fertility and dispersal (both immigration and dispersal at spatial mesoscale). The main difference between these two drivers is that yellow-legged gulls are competitors with a long shared evolutionary history and long-term stability occurs when the two species occur in a specific patch. On the contrary, gulls have not developed evolutionary defences to cope with terrestrial predators like carnivores, and this is why they select for breeding patches isolated and protected against the invasions of the predators.

Population density of yellow-legged gulls and the number of carnivores present at La Banya have been estimated over the years, and gull carcasses and tracks in the sand have provided estimates of yearly predation rates that varied with the individual predator and its foraging preferences. Other biotic factor is food availability, and a proxy to assess its temporal variability is through the statistics of landings of trawlers in the harbors close to the study site, which are highly correlated with the amounts of fish discarded. Food per capita decreased as population density approached the carrying capacity during the mid 90's and also because trawler catches per unit effort have decreased in recent decades due to overharvesting of fish stocks. Adult survival, which is the vital rate with largest elasticity for the population dynamics of the gulls, changes with bycatch mortality at longline fisheries and by carnivore predation.

Previous studies have shown that bycatch is relatively constant over the years, whereas carnivore density may vary with breeding season, although values were always low. Predation rate increased with the density of carnivores, but some noise for this association occurred due to individual carnivore preferences for gull predation. However, these predation rates did not significantly affect adult survival, whereas they increased dispersal probabilities to other patches (either occupied or empty).

In summary, there is no record of a decrease of food availability in absolute and per capita values (i.e. accounting for density-dependence), nor a decrease of local survival by carnivore predation or an increase of competition with the dominant yellow-legged gulls. Thus, these variables cannot explain the decline of population density of Audouin's gulls to patch collapse at La Banya since 2006, which should respond to an increase of dispersal to other patches.

2 Mathematical model with dispersal by social copying

We introduce a mean field model using an ordinary differential equation modeling key ecological processes expected to explain the field data. Our hypothesis is that the presence of predators triggers a social response of the birds that start dispersing in an inverse, density-dependent manner. That is, the less individuals at the patch, the faster the dispersal rate. The mathematical model describes the population dynamics of birds (variable x) in the patch of study. The model can be considered as a single-patch system considering immigration and dispersal of individuals. Other processes considered are intra-specific competition for resources and density-independent death rates. As we thoroughly explain below, the model incorporates a function incorporating a social copying dispersal process assumed to occur due to the presence of predators. The model will be adapted to the dynamics and processes hypothesized for the different epochs: a first epoch before predators arrival (1981-1997) and a second epoch comprised between 1998 and 2017, containing the full collapse of the population during years 2006-2017 since predators were removed in 2017. The model reads:

$$\frac{dx}{dt} = \gamma x\left(1-\frac{x}{K}\right) - \varepsilon x - \lambda\,\Psi(x,\mu,\sigma,\delta), \tag{1}$$

with initial population $x(0)$. This equation considers the following ecological processes:

$$\frac{dx}{dt} = \underbrace{\alpha x}_{\substack{\text{Immigration,}\\ \text{growth and}\\ \text{death}}} - \underbrace{\gamma\frac{x^2}{K}}_{\substack{\text{Nonlinear}\\ \text{competition}\\ \text{term}}} - \underbrace{\lambda\Psi(x,\mu,\sigma,\delta)}_{\text{Dispersal by social copying}} \tag{2}$$

with $\alpha = \gamma - \epsilon$ (units: birds/year). Equation (1) considers an initial exponential increase of the population proportional to parameter γ (including both the reproduction of birds and the arrival of new individuals from other patches of the metapopulation, which is made proportional to the population present at the patch). This population increase is constrained by a logistic function with carrying capacity K (units: birds), introducing intra-specific competition for resources. Also, we consider density-independent death rate, proportional to parameter ϵ. The competition term will be also written as βx^2, with $\beta = \gamma/K$ (units: years^{-1}). The nonlinear dispersal function given by

$$\Psi(x,\mu,\sigma,\delta) := \begin{cases} \frac{1-\mathcal{E}_{\text{dir}}(x,\mu,\sigma,\delta)}{1-\mathcal{E}_{\text{dir}}(0,\mu,\sigma,\delta)} & \text{when } 0\le x\le\delta,\\ \frac{1-\mathcal{E}(x,\sigma,\delta)}{1-\mathcal{E}_{\text{dir}}(0,\mu,\sigma,\delta)} & \text{when } x\ge\delta, \end{cases} \tag{3}$$

$$\mathcal{E}_{\text{dir}}(x,\mu,\sigma,\delta) := \left(\mu\frac{\Theta+\sigma\delta}{2\Theta+\sigma\delta}\left(1-\frac{x}{\delta}\right)+\frac{x}{\delta}\right)\mathcal{E}(x,\sigma,\delta), \tag{4}$$

where

$$\mathcal{E}(x,\sigma,\delta) := \frac{\sigma(x-\delta)}{\Theta+\sigma|x-\delta|}, \tag{5}$$

is an *Elliot sigmoid* Θ–scaled, σ–strengthened, and δ–displaced. All the model parameters are non-negative and we have fixed $\Theta := 1000$ (this parameter controls how stretched is the sigmoid function and it is related with the order of magnitude of the carrying capacity K). Figure 1 shows some examples of the shape of the function Ψ for different values of the parameters. The function Ψ is designed so that the dispersal response of the population of birds generically increases when the population numbers at the patch diminish. Finally, parameter λ is the dispersal rate that parameterize the impact of function Ψ (units: birds/year) in Eq. (2).

The following proposition and lemma summarize the mathematical properties of the functions $\Psi(x,\mu,\sigma,\delta)$, $\mathcal{E}(x,\sigma,\delta)$ and $\mathcal{E}_{\text{dir}}(x,\mu,\sigma,\delta)$.

Lemma 1 (On the functions $\mathcal{E}(x,\sigma,\delta)$ and $\mathcal{E}_{\text{dir}}(x,\mu,\sigma,\delta)$)**.** *For all $\mu,\sigma,\delta\ge 0$ and $x\ge 0$ we have*

(1) $\mathcal{E}(0,\sigma,\delta) = -\frac{\sigma\delta}{\Theta+\sigma\delta}$, *and* $\mathcal{E}_{\text{dir}}(0,\mu,\sigma,\delta) = -\mu\frac{\sigma\delta}{2\Theta+\sigma\delta}$,

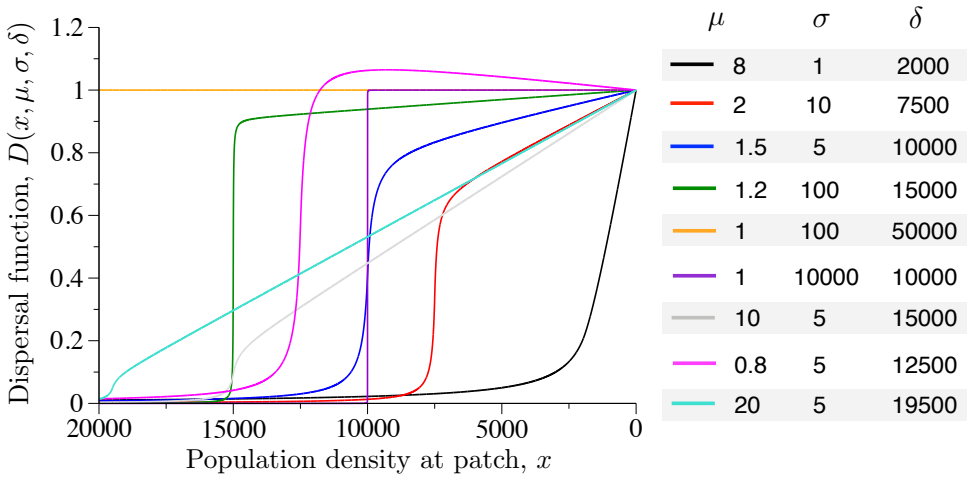

Figure 1: Shapes of the function $\Psi(x,\mu,\sigma,\delta)$ used to model social copying behaviour during dispersal. We display several shapes tuning three parameters μ,σ and δ, ranging from constant dispersal (orange line below), to exponential-like (black curve) or to sigmoid-like (e.g. red, blue or violet curves). The parameter μ determines if the curves intersect 0 population density from below ($\mu>1$) or from above ($0\le\mu<1$) $\Psi(x,\mu,\sigma,\delta)=1$. The parameter σ determines how steep is the sigmoid and δ denotes the population size at which the curve starts bending.

(2) $\mathcal{E}_{\mathsf{dir}}(\delta,\mu,\sigma,\delta)=\mathcal{E}(\delta,\sigma,\delta)=0$,

(3) $\mathcal{E}_{\mathsf{dir}}(x,\mu,0,\delta)=\mathcal{E}(x,0,\delta)\equiv 0$ *for every* $x\ge 0$,

(4) $-1<\mathcal{E}(x,\sigma,\delta)<1$,

(5) $\frac{\mathrm{d}}{\mathrm{d}x}\mathcal{E}(x,\sigma,\delta)=\frac{\Theta\sigma}{(\Theta+\sigma|x-\delta|)^2}>0$, *and*

(6) $\lim_{x\to+\infty}\mathcal{E}(x,\sigma,\delta)=1$ *provided that* $\sigma>0$.

When $\sigma>0$, $\mathcal{E}$ *and* $\mathcal{E}_{\mathsf{dir}}$ *are continuous as functions of* x. *Moreover, for* $\mu\ge 0$ *and* $0\le x\le\delta$,

$$\frac{\mathrm{d}}{\mathrm{d}x}\mathcal{E}_{\mathsf{dir}}(x,\mu,\sigma,\delta)=\frac{\sigma}{\delta(2\Theta+\sigma\delta)(\Theta+\sigma z)}\left(-\Gamma z+\Big(\mu\delta(\Theta+\sigma\delta)+\Gamma(\delta-z)\Big)\frac{\Theta}{\Theta+\sigma z}\right),$$

where $\Gamma:=(2-\mu)\Theta+(1-\mu)\sigma\delta$ *and* $z=\delta-x$.

Proposition 2 (On the function $\Psi(x,\mu,\sigma,\delta)$)**.** *For every* $\mu,\delta\ge 0$ *and* $x\ge 0$ *we have* $\Psi(x,\mu,0,\delta)\equiv 1$. *Moreover, for* $\sigma>0$ *we have*

(a) The function $\Psi(x,\mu,\sigma,\delta)$, *as a function of* x, *is continuous, differentiable, and strictly positive.*

(b) $\Psi(0,\mu,\sigma,\delta)=1$ *and* $\lim_{x\to+\infty}\Psi(x,\mu,\sigma,\delta)=0$. *Moreover, for every* $\sigma\ge 1$ *and* $0<x\le\delta$ *we have*

$$\Psi(x,\mu,\sigma,\delta)<1+\frac{1-\mu}{(1+\mu)}\frac{x}{\delta}.$$

(c) If $\mu\ge 1$, *then* $\Psi(x,\mu,\sigma,\delta)$ *is strictly decreasing as a function of* x. *Moreover,* $\frac{\mathrm{d}}{\mathrm{d}x}\Psi(x,\mu,\sigma,\delta)\big|_{x=0}$ *is 0 when* $\mu=1$ *and negative when* $\mu>1$.

(d) For $0\le\mu<1$ *and* $\delta>0$, $\Psi(x,\mu,\sigma,\delta)$ *is a unimodal function with a maximum at* $x^*\in(0,\delta)$ *(that is,* Ψ *is strictly increasing in* $[0,x^*]$ *and strictly decreasing in* $[x^*,+\infty)$*). In particular,* $\frac{\mathrm{d}}{\mathrm{d}x}\Psi(x,\mu,\sigma,\delta)>0$ *for every* $x\in[0,x^*)$. *On the other hand, for every* $x\in[0,\delta]$, $\Psi(x,\mu,\sigma,\delta)\le\Psi(x^*,\mu,\sigma,\delta)<2$.

By using the logistic growth Model (2) (with $\lambda=0$) for the First Epoch data (no migration) one can estimate (as intrinsic parameters of the model):

Parameter	units	Range or value	Meaning or description
K	birds	16651.2696	Carrying Capacity.
γ	birds^2/year	0.406001835194	Intrinsic growth rate.
ε	birds/year	0.057052426616	Death rate.
$\alpha=\gamma-\varepsilon$	birds/year	0.3489494085776018	Neat population growth rate.
$\beta=\frac{\gamma}{K}$	birds^2/year	0.000024382635446	Intrinsic growth rate over the carrying capacity.

3 The exercise

Fit the parameters of Model

$$\frac{dx}{dt} = \varphi x - \beta x^2 - \lambda \Psi(x, \mu, \sigma, \delta) \tag{2}$$

to the Second Epoch data to check the hypothesis that Andouin's migration occurs with social copying with a Genetic Algorithm.

The solution of this model is denoted by $x(t) = x_{\varphi,\lambda,\mu,\sigma,\delta}(t)$, and its parameters are:

Parameter	Range or value	Meaning or description
β	0.000024382635446	Intrinsic growth rate over the carrying capacity. Estimated with the Fisrt Epoch Data.
$\varphi = \alpha - \rho$	$\leq \alpha = 0.3489494085776018$	Neat population growth rate. It includes a linear migration term of the form ρx, where ρ is the linear dispersal rate.
$x_{\varphi,\lambda,\mu,\sigma,\delta}(0)$	$[0, K]$	ODE's Initial condition.
λ	$\mathbb{R}^+$	Non-linear Dispersal Rate.
μ	$\mathbb{R}^+$	Determines $\frac{\mathrm{d}}{\mathrm{d}x}\Psi(x,\mu,\sigma,\delta)\big\|_{x=0}$. It is $\begin{cases} 0 & \text{when } \mu = 1, \\ \text{negative} & \text{when } \mu > 1, \text{ and} \\ \text{positive} & \text{when } \mu < 1. \end{cases}$
σ	$\mathbb{R}^+$	Determines the "slopes" of the sigmoids. $\sigma \approx 600$ approximates a Heaviside function.
δ	$\mathbb{R}^+$	Point of change of concavity of $\Psi(x, \mu, \sigma, \delta)$.

Observe that the solution $x(t)$ depends on the initial condition $x(0) \in [0, K]$, that must be considered a free parameter as well.

4 Proposed solution strategy

The exercise is to be solved with a minimising genetic algorithm with an appropriate fitness function.

Please, be aware that the solution of the ODE has a rather strong sensitive dependence with respect to parameters and initial condition); meaning that the genetic algorithm will have difficulties in finding the solution.

4.1 Individuals

Clearly, an individual in the population is specified by six chromosomes corresponding to the six free parameters.

As it has been explained, the proof of Holland's Convergence Theorem works in the setting of genes or chromosomes consisting in unsigned integers expressed in binary. Consequently the above "real numbers phenotype" is better encoded in the form of a *discretized* genotype consisting in unsigned integers. In the following table we explain, for each parameter, the theoretical range (given in the above table), an effective (reasonable, common sense) search range and a reasonable sensitivity (or better said precision), thus fixing the range and discretization formula for the genotype.

Parameter	Theoretical Range	Phenotype		Genotype	
		Effective Search Range	precision or discretization step	Integer Search Range	Factor (formula) from genotype to phenotype
$x(0)$	$[0, K]$	$[0, 16600]$	10^{-2}	$[0, 2^{21} - 1]$	$\frac{16600}{2^{21}-1} \approx 0.0079155006005766\cdots$
φ	$(-\infty, \alpha]$	$[-100, 0.35]$	10^{-8}	$[0, 2^{34} - 1]$	$\mathtt{g} \cdot \frac{100.35}{2^{34}-1} - 100 \approx$ $\mathtt{g} \cdot 5.841138773 \cdot 10^{-9} - 100$
λ	$\mathbb{R}^+$	$[0, 3000]$	10^{-4}	$[0, 2^{25} - 1]$	$\frac{3000}{2^{25}-1} \approx 8.940696982762\cdots 10^{-5}$
μ	$\mathbb{R}^+$	$[0, 20]$	10^{-6}	$[0, 2^{25} - 1]$	$\frac{20}{2^{25}-1} \approx 5.960464655174\cdots 10^{-7}$
σ	$\mathbb{R}^+$	$[0, 1000]$	10^{-2}	$[0, 2^{17} - 1]$	$\frac{1000}{2^{17}-1} \approx 0.007629452739355006\cdots$
δ	$\mathbb{R}^+$	$[0, 25000]$	1	$[0, 2^{15} - 1]$	$\frac{25000}{2^{15}-1} \approx 0.7629627368999298\cdots$

Observations:

- All upper limit and precision values for the phenotype have been set to "common sense reasonable values".
- All upper limit values of the genotype have been chosen to be the smallest possible powers of two that satisfy the following condition:

  $$\text{genotype upper limit of the form } 2^n > \text{phenotype upper limit/precision.}$$

  For example, for the initial conditions the above formula gives

  $$2^{21} = 2,097,152 > 1,660,000 = 16,600/10^{-2}.$$

- Observe that the number $2^n - 1$ when written in binary in 64 a bits representation, has a string of $64 - n$ consecutive zeroes at the left, and a string of n consecutive ones at the right. Moreover, the expression in binary of all integers in the range $[0, 2^n - 1]$ has a string of at least $64 - n$ consecutive zeroes at the left. This is very useful, when programming crossovers and mutations, to avoid complicate feasibility tests.
- All powers of two in the above table have exponent less than or equal to 40, and there are some with exponents larger than 32. So, the base data type for the genes to store these genotype elements must be `unsigned long int`.

4.2 Fitness function

The Genetic Algorithm must identify an individual that could possibly have generated the observed data for the Second Epoch. This is done by finding the "fittest" individual from the point of view of generating the observed data. In other words, the fitness function must measure how similar is the observed data to solution of the ODE which has a given individual as parameters

More precisely, an individual Ind contains all the necessary parameters to compute the solution of Model (2) for $t = 1, 2, \ldots, 11$.

Two possible norms that measure the agreement between the pandemic data generated by Ind and the figured pandemic public data are:

$$\max\left\{\Big(x(t) - z(t+2006)\Big)^2 : t = 0, 1, 2, \ldots, 11\right\}, \tag{6}$$

and

$$\sum_{t=0}^{11} W_t\Big(x(t) - z(t+2006)\Big)^2, \tag{7}$$

where $z(y)$ denotes the population size of Andouine seagulls at year y and $W_0, W_1, \ldots, W_{11} \geq 0$ are weights. Clearly, a value zero in the above fitness function indicates that Ind's phenotype *is* the one that drives the pandemic through Model 2.

4.3 Integrating an ODE: Computing the values of $x(t)$ for $t = 1, 2, \ldots, 100$

We will use the Runge-Kutta-Fehlberg method of order 7-8 with adaptive space (see the appendix to this document).

In the file `RKF78.c` (also needed `RKF78.h` for definitions and prototypes) there is an implementation for ODE's and another one for systems (see the implementation notes in `RKF78.c` for the meaning of parameters and how to use the procedure).

However as an example on how to use `RKF78` we provide here a full programmed implementation of the computation of the values $x(t)$ for $t = 0, 1, 2, \ldots, 11$.

```
#define ElliotSigmoidSCALE 1000
#define TwoElliotSigmoidSCALE 2000

double ElliotSigmoid(double x, double sigma, double delta) {
   x = sigma*(x-delta);
   return x/(ElliotSigmoidSCALE + fabs(x));
}
```

```
double Psi(double x, double mu, double sigma, double delta){
    if(fabs(sigma) < ZeRoParsThreshold) return 1.0;
    double ES = ElliotSigmoid(x, sigma, delta);
    sigma *= delta; x /= delta;
    if(x < delta) {
       ES = ES * (x + (mu*(1.0-x)*(sigma + ElliotSigmoidSCALE)) / (sigma + TwoElliotSigmoidSCALE));
    }
    return ((1 - ES)*(sigma + TwoElliotSigmoidSCALE)) / (sigma*(1+mu) + TwoElliotSigmoidSCALE);
}

typedef struct {
   double phi;
   double beta;
   double lambda;
   double mu;
   double sigma;
   double delta;
} ODE_Parameters;

void MigrationODE(double t, double x, double *der, void *Params){
    ODE_Parameters *par = (ODE_Parameters *) Params; // Pointer cast to save typing and thinking
    *der = par->phi * x - par->beta*x*x - par->lambda*Psi(x, par->mu, par->sigma, par->delta);
}

#define HMAX  1.0
#define HMIN  1.e-6
#define RKTOL 1.e-8
int Generate_EDO_Prediction( double *xt, double x0,
                             unsigned short number_of_years,
                             ODE_Parameters *pars ){
    register unsigned ty;
    xt[0] = x0;  // Storing IC x(0)
    for(ty=1; ty < number_of_years; ty++) xt[ty] = 0.0;

    double t = 0.0, err, h = 1.e-3;
    for(ty=1; ty < number_of_years; ty++) { int status;
        while(t+h < ty) {
          status = RKF78(&t, &x0, &h, &err, HMIN, HMAX, RKTOL, pars, MigrationODE);
          if(status) return status;
        }  // Adaptative stepsize h. To assure stopping at t = ty
        h = ty - t;
        status = RKF78(&t, &x0, &h, &err, HMIN, HMAX, RKTOL, pars, MigrationODE);
        if(status) return status;
        xt[ty] = x0;
    }
    return 0;
}
```

Appendix A
Runge-Kutta Methods

The Runge-Kutta methods are an important family of iterative methods for the approximation of solutions of ODE's, that were develoved around 1900 by the german mathematicians C. Runge (1856–1927) and M.W. Kutta (1867–1944). We start with the considereation of the explicit methods. Let us consider an initail value problem (IVP)

$$\frac{d\mathbf{x}}{dt} = f(t, \mathbf{x}(t)), \tag{A.1}$$

$\mathbf{x}(t) = (x_1(t), x_2(t), \dots x_n(t))^T$, $f \in [a,b] \times \mathbb{R}^n \to \mathbb{R}^n$, with an initial condition

$$\mathbf{x}(0) = \mathbf{x}_0\,. \tag{A.2}$$

We are interested in a numerical approximation of the continuously differentiable solution $\mathbf{x}(t)$ of the IVP (A.1)–(A.2) over the time interval $t \in [a,b]$. To this aim we subdivide the interval $[a,b]$ into M equal subintervals and select *the mesh points* t_j [11, 8]

$$t_j = a + jh\,, \qquad j = 0, 1, \dots, M, \quad h = \frac{b-a}{M}\,. \tag{A.3}$$

The value h is called *a step size*.

The family of explicit Runge–Kutta (RK) methods of the m'th stage is given by [11, 9]

$$\mathbf{x}(t_{n+1}) := \mathbf{x}_{n+1} = \mathbf{x}_n + h\sum_{i=1}^{m} c_i k_i\,, \tag{A.4}$$

where

$$
\begin{aligned}
k_1 &= f(t_n, \mathbf{x}_n), \\
k_2 &= f(t_n + \alpha_2 h, \mathbf{x}_n + h\beta_{21} k_1(t_n, \mathbf{x}_n)), \\
k_3 &= f(t_n + \alpha_3 h, \mathbf{x}_n + h(\beta_{31} k_1(t_n, \mathbf{x}_n) + \beta_{32} k_2(t_n, \mathbf{x}_n))), \\
&\vdots \\
k_m &= f(t_n + \alpha_m h, \mathbf{x}_n + h \sum_{j=1}^{m-1} \beta_{mj} k_j).
\end{aligned}
$$

To specify a particular method, we need to provide the integer m (the number of stages), and the coefficients α_i (for $i = 2, 3, ..., m$), β_{ij} (for $1 \le j < i \le m$), and c_i (for $i = 1, 2, ..., m$). These data are usually arranged in a co-called *Butcher tableau* (after John C. Butcher) [11, 9]:

Table A.1 The Butcher tableau.

$$
\begin{array}{c|cccc}
0 & & & & \\
\alpha_2 & \beta_{21} & & & \\
\alpha_3 & \beta_{31} & \beta_{32} & & \\
\vdots & \vdots & \vdots & \ddots & \\
\vdots & \vdots & \vdots & & \\
\alpha_m & \beta_{m1} & \beta_{m2} & \cdots\cdots & \beta_{mm-1} \\
\hline
 & c_1 & c_2 & \cdots\cdots & c_{m-1} \quad c_m
\end{array}
$$

Examples

1. Let $m = 1$. Then

$$
\begin{aligned}
k_1 &= f(t_n, \mathbf{x}_n), \\
\mathbf{x}_{n+1} &= \mathbf{x}_n + h c_1 f(t_n, \mathbf{x}_n).
\end{aligned}
$$

   On the other hand, the Taylor expansion yields

$$
\mathbf{x}_{n+1} = \mathbf{x}_n + h\dot{\mathbf{x}}\big|_{t_n} + \cdots = \mathbf{x}_n + h f(t_n, \mathbf{x}_n) + \mathcal{O}(h^2) \Rightarrow c_1 = 1.
$$

   Thus, the first-stage RK-method is equivalent to the explicit Euler's method. Note that the Euler's method is of the first order of accuracy. Thus we can speak about the RK method of the first order.
2. Now consider the case $m = 2$. In this case Eq. (A.4) is equivalent to the system

$$
\begin{aligned}
k_1 &= f(t_n, \mathbf{x}_n)\,, \\
k_2 &= f(t_n + \alpha_2 h, \mathbf{x}_n + h\beta_{21} k_1)\,, \\
\mathbf{x}_{n+1} &= \mathbf{x}_n + h(c_1 k_1 + c_2 k_2)\,.
\end{aligned}
\tag{A.5}
$$

Now let us write down the Taylor series expansion of $\mathbf{x}$ in the neighborhood of t_n up to the h^2 term, i.e.,

$$
\mathbf{x}_{n+1} = \mathbf{x}_n + h \left.\frac{d\mathbf{x}}{dt}\right|_{t_n} + \frac{h^2}{2} \left.\frac{d^2\mathbf{x}}{dt^2}\right|_{t_n} + \mathscr{O}(h^3)\,.
$$

However, we know that $\dot{\mathbf{x}} = f(t, \mathbf{x})$, so that

$$
\frac{d^2\mathbf{x}}{dt^2} := \frac{df(t,\mathbf{x})}{dt} = \frac{\partial f(t,\mathbf{x})}{\partial t} + f(t,\mathbf{x}) \frac{\partial f(t,\mathbf{x})}{\partial \mathbf{x}}\,.
$$

Hence the Taylor series expansion can be rewritten as

$$
\mathbf{x}_{n+1} - \mathbf{x}_n = h f(t_n, \mathbf{x}_n) + \frac{h^2}{2} \left(\frac{\partial f}{\partial t} + f \frac{\partial f}{\partial \mathbf{x}} \right) \Bigg|_{(t_n, \mathbf{x}_n)} + \mathscr{O}(h^3)\,. \tag{A.6}
$$

On the other hand, the term k_2 in the proposed RK method can also expanded to $\mathscr{O}(h^3)$ as

$$
k_2 = f(t_n + \alpha_2 h, \mathbf{x}_n + h\beta_{21} k_1) = h f(t_n, \mathbf{x}_n) + h\,\alpha_2 \left.\frac{\partial f}{\partial t}\right|_{(t_n, \mathbf{x}_n)} + h\beta_{21} f \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{(t_n, \mathbf{x}_n)} + \mathscr{O}(h^3)\,.
$$

Now, substituting this relation for k_2 into the last equation of (A.5), we achieve the following expression:

$$
\mathbf{x}_{n+1} - \mathbf{x}_n = h\,(c_1 + c_2)\, f(t_n, \mathbf{x}_n) + h^2 c_2\,\alpha_2 \left.\frac{\partial f}{\partial t}\right|_{(t_n, \mathbf{x}_n)} + h^2 c_2 \beta_{21} f \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{(t_n, \mathbf{x}_n)} + \mathscr{O}(h^3)\,.
$$

Making comparision the last equation and Eq. (A.6) we can write down the system of algebraic equations for unknown coefficients

$$
\begin{aligned}
c_1 + c_2 &= 1\,, \\
c_2\,\alpha_2 &= \frac{1}{2}\,, \\
c_2\,\beta_{21} &= \frac{1}{2}\,.
\end{aligned}
$$

The system involves four unknowns in three equations. That is, one additional condition must be supplied to solve the system. We discuss two useful choices, namely

a) Let $\alpha_2 = 1$. Then $c_2 = 1/2$, $c_1 = 1/2$, $\beta_{21} = 1$. The corresponding Butcher tableau reads:

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

Thus, in this case the two-stages RK method takes the form

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\Big(f(t_n, \mathbf{x}_n) + f(t_n + h, \mathbf{x}_n + h f(t_n, \mathbf{x}_n)) \Big),$$

and is equivalent to the Heun's method, so we refer the last method to as RK-method of the second order.

b) Now let $\alpha_2 = 1/2$. In this case $c_2 = 1$, $c_1 = 0$, $\beta_{21} = 1/2$. The corresponding Butcher tableau reads:

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

In this case the second-order RK method (A.4) can be written as

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\, f\big(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{h}{2} f(t_n, \mathbf{x}_n)\big)$$

and is called the *RK2 method*.

RK4 Methods

One member of the family of Runge–Kutta methods (A.4) is often referred to as *RK4 method* or *classical RK method* and represents one of the solutions corresponding to the case $m = 4$. In this case, by matching coefficients with those of the Taylor series one obtains the following system of equations [8]

$$
\begin{aligned}
c_1 + c_2 + c_3 + c_4 &= 1\,, \\
\beta_{21} &= \alpha_2\,, \\
\beta_{31} + \beta_{32} &= \alpha_3\,, \\
c_2\alpha_2 + c_3\alpha_3 + c_4\alpha_4 &= \frac{1}{2}\,, \\
c_2\alpha_2^2 + c_3\alpha_3^2 + c_4\alpha_4^2 &= \frac{1}{3}\,, \\
c_2\alpha_2^3 + c_3\alpha_3^3 + c_4\alpha_4^3 &= \frac{1}{4}\,, \\
c_3\alpha_2\beta_{32} + c_4(\alpha_2\beta_{42} + \alpha_3\beta_{43}) &= \frac{1}{6}\,, \\
c_3\alpha_2\alpha_3\beta_{32} + c_4\alpha_4(\alpha_2\beta_{42} + \alpha_3\beta_{43}) &= \frac{1}{8}\,, \\
c_3\alpha_2^2\beta_{32} + c_4(\alpha_2^2\beta_{42} + \alpha_3^2\beta_{43}) &= \frac{1}{12}\,, \\
c_4\alpha_2\beta_{32}\beta_{43} &= \frac{1}{24}\,.
\end{aligned}
$$

The system involves thirteen unknowns in eleven equations. That is, two additional condition must be supplied to solve the system. The most useful choices is [9]

$$
\alpha_2 = \frac{1}{2}\,, \qquad \beta_{31} = 0\,.
$$

The corresponding Butcher tableau is presented in Table A.2. The tableau A.2 yields

Table A.2 The Butcher tableau corresponding to the RK4 method.

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

the equivalent corresponding equations defining the classical RK4 method:

$$
\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{6}\big(k_1 + 2k_2 + 2k_3 + k_4\big), \tag{A.7}
$$

where

$$
\begin{aligned}
k_1 &= f(t_n, \mathbf{x}_n),\\
k_2 &= f(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{h}{2}k_1),\\
k_3 &= f(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{h}{2}k_2),\\
k_4 &= f(t_n + h, \mathbf{x}_n + hk_3).
\end{aligned}
$$

This method is reasonably simple and robust and is a good general candidate for numerical solution of ODE's when combined with an intelligent adaptive step-size routine or an embedded methods (,e.g., so-called Runge-Kutta-Fehlberg methods (RKF45)).

Remark:

Notice that except for the classical method (A.7), one can also construct other RK4 methods. We mention only so-called $3/8$-*Runge-Kutta method*. The Brutcher tableau, corresponding to this method is presented in Table A.3.

Table A.3 The Butcher tableau corresponding to the $3/8$- Runge-Kutta method.

0				
1/3	1/3			
2/3	-1/3	1		
1	1	-1	1	
	1/8	3/8	3/8	1/8

Geometrical interpretation of the RK4 method

Let us consider a curve $\mathbf{x}(t)$, obtained by (A.7) over a single time step from t_n to t_{n+1}. The next value of approximation $\mathbf{x}_{n+1}$ is obtained ty integrating the slope function, i.e.,

$$
\mathbf{x}_{n+1} - \mathbf{x}_n = \int_{t_n}^{t_{n+1}} f(t, \mathbf{x})\, dt\,. \tag{A.8}
$$

Now, if the Simpson's rule is applied, the approximation to the integral of the last equation reads [10]

$$
\int_{t_n}^{t_{n+1}} f(t, \mathbf{x})\, dt \approx \frac{h}{6}\left(f(t_n, \mathbf{x}(t_n)) + 4\, f(t_n + \frac{h}{2}, \mathbf{x}(t_n + \frac{h}{2})) + f(t_{n+1}, \mathbf{x}(t_{n+1}))\right). \tag{A.9}
$$

On the other hand, the values k_1, k_2, k_3 and k_4 are approximations for slopes of the curve $\mathbf{x}$, i.e., k_1 is the slope of the left end of the interval, k_2 and k_3 describe two estimations of the slope in the middle of the time interval, whereas k_4 corresponds to the slope at the right. Hence, we can choose $f(t_n, \mathbf{x}(t_n)) = k_1$ and $f(t_{n+1}, \mathbf{x}(t_{n+1})) = k_4$, whereas for the value in the middle we choose the average of k_2 and k_3, i.e.,

$$f(t_n + \frac{h}{2}, \mathbf{x}(t_n + \frac{h}{2})) = \frac{k_2 + k_3}{2}.$$

Then Eq. (A.8) becomes

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{6}\left(k_1 + \frac{4(k_2 + k_3)}{2} + k_4\right),$$

which is equivalent to the RK4 schema (A.7).

Stage versus Order

The local truncation error ε for the method (A.7) can be estimated from the error term for the Simpson's rule (A.9) and equals [10, 8]

$$\varepsilon_{n+1} = -h^5 \frac{\mathbf{x}^{(4)}}{2880}.$$

Now we can estimate the final global error E, if we suppose that only the error above is presented. After M steps the accumulated error for the RK4 method reads

$$E(\mathbf{x}(b), h) = -\sum_{k=1}^{M} h^5 \frac{\mathbf{x}^{(4)}}{2880} \approx \frac{b-a}{2880}\mathbf{x}^{(4)} h = \mathscr{O}(h^4).$$

That is, the RK4 method (A.7) is of the fourth order. Now, let us compare two appximations, obtained using the time steps h and $h/2$. For the step size h we have

$$E(\mathbf{x}(b), h) \approx K h^4,$$

with $K = \text{const}$. Hence, for the step $h/2$ we get

$$E(\mathbf{x}(b), \frac{h}{2}) = K\frac{h^4}{16} \approx \frac{1}{16}E(\mathbf{x}(b), h).$$

That is, if the step size in (A.7) is reduced by the factor of two, the global error of the method will be reduced by the factor of $1/16$.

Remark:

In general there are two ways to improve the accuracy:

1. One can reduce the time step h, i.e., the amount of steps increases;
2. The method of the higher convergency order can be used.

However, increasing of the convergency order p is reasonable only up to some limit, given by so-called *Butcher barrier* [11], which says, that the amount of stages m grows faster, as the order p. In other words, *for $m \geq 5$ there are no explicit RK methods with the convergency order $p = m$ (the corresponding system is unsolvable)*. Hence, in order to reach convergency order five one needs six stages. Notice that further increasing of the stage $m = 7$ leads to the convergency order $p = 5$ as well.

A.0.1 Adaptive stepsize control and embedded methods

As mentioned above, one way to guarantee accuracy in the solution of (A.1)–(A.1) is to solve the problem twice using step sizes h and $h/2$. To illustrate this approach, let us consider the RK method of the order p and denote an exact solution at the point $t_{n+1} = t_n + h$ by $\widetilde{\mathbf{x}}_{n+1}$, whereas $\mathbf{x}_1$ and $\mathbf{x}_2$ represent the approximate solutions, corresponding to the step sizes h and $h/2$. Now let us perform one step with the step size h and after that two steps each of size $h/2$. In this case the true solution and two numerical approximations are related by

$$\widetilde{\mathbf{x}}_{n+1} = \mathbf{x}_1 + Ch^{p+1} + \mathscr{O}(h^{p+2}),$$
$$\widetilde{\mathbf{x}}_{n+1} = \mathbf{x}_2 + 2C\left(\frac{h}{2}\right)^{p+1} + \mathscr{O}(h^{p+2}).$$

That is,

$$|\mathbf{x}_1 - \mathbf{x}_2| = Ch^{p+1}\left(1 - \frac{1}{2^p}\right) \Leftrightarrow C = \frac{|\mathbf{x}_1 - \mathbf{x}_2|}{(1-2^{-p})h^{p+1}}.$$

Substituing the relation for C in the second estimate for the true solution we get

$$\widetilde{\mathbf{x}}_{n+1} = \mathbf{x}_2 + \varepsilon + \mathscr{O}(h^{p+2}),$$

where

$$\varepsilon = \frac{|\mathbf{x}_1 - \mathbf{x}_2|}{2^p - 1}$$

can be considered as a convenient *indicator* of the truncation error. That is, we have improved our estimate to the order $p+1$. For example, for $p = 4$ we get

$$\widetilde{\mathbf{x}}_{n+1} = \mathbf{x}_2 + \frac{|\mathbf{x}_1 - \mathbf{x}_2|}{15} + \mathscr{O}(h^6).$$

This estimate is accurate to fifth order, one order higter than with the original step h. However, this method is not efficient. First of all, it requires a significant amount

of computation (we should solve the equation three times at each time step). The second point is, that we have no possibility to control the truncation error of the method (higher order means not always higher accuracy).
However we can use an estimate ε for the *step size control*, namely we can compare ε with some *desired accuracy* ε_0 (see Fig A.1).

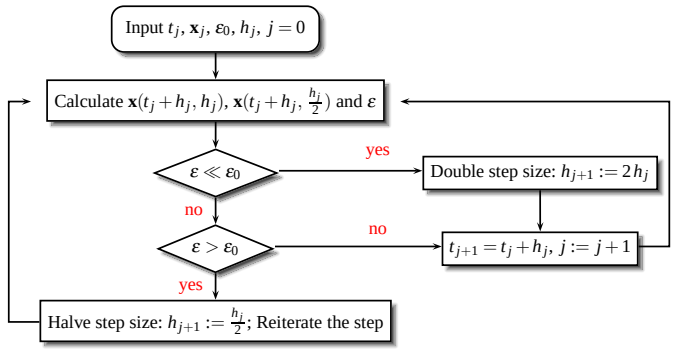

Fig. A.1 Flow diagramm of the step size control by use of the step doubling method.

Alternatively, using the estimate ε, we can try to formulate the following problem of the *adaptive step size control*, namely: Using the given values $\mathbf{x}_j$ and t_j, find the largest possible step size h_{new}, so that the truncation error after the step with this step size remains below some given desired accuracy ε_0, i.e,

$$C h_{new}^{p+1} \le \varepsilon_0 \Leftrightarrow \left(\frac{h_{new}}{h}\right)^{p+1} \frac{|\mathbf{x}_1 - \mathbf{x}_2|}{1-2^{-p}} \le \varepsilon_0 .$$

That is,

$$h_{new} = h\left(\frac{\varepsilon_0}{\varepsilon}\right)^{1/p+1} .$$

Then if the two answers are in close agreement, the approximation is accepted. If $\varepsilon > \varepsilon_0$ the step size has to be decreased, whereas the relation $\varepsilon < \varepsilon_0$ means, that the step size has to be increased in the next step.

Notice that because our estimate of error is not exact, we should put some "safety" factor $\beta \simeq 1$ [11, 9]. Usually, $\beta = 0.8, 0.9$. The flow diagramm, corresponding to the the adaptive step size control is shown on Fig. A.2

Notice one additional technical point. The choise of the desired error ε_0 depends on the IVP we are interested in. In some applications it is convinient to set ε_0 propotional to h [9]. In this case the exponent $1/p+1$ in the estimate of the new time step is no longer correct (if h is reduced from a too-large value, the new predicted value h_{new} will fail to meet the desired accuracy, so instead of $1/p+1$ we should scale with $1/p$ (see [9] for details)). That is, the optimal new step size can be written as

$$h_{new} = \begin{cases} \beta h\left(\frac{\varepsilon_0}{\varepsilon}\right)^{1/p+1}, & \varepsilon \ge \varepsilon_0, \\ \beta h\left(\frac{\varepsilon_0}{\varepsilon}\right)^{1/p}, & \varepsilon < \varepsilon_0, \end{cases} \tag{A.10}$$

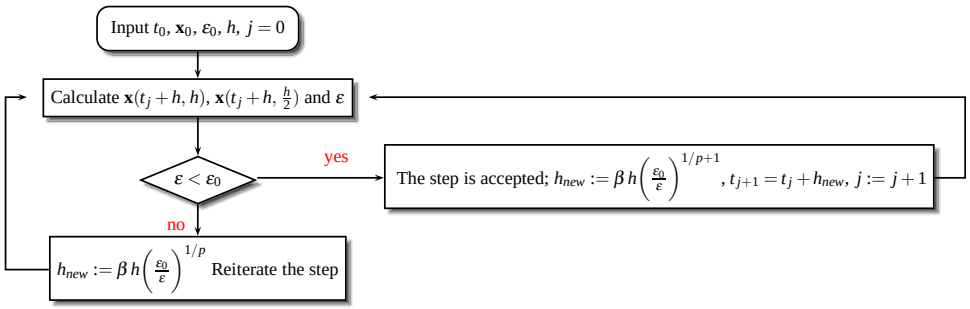

Fig. A.2 Flow diagramm of the adaptive step size control by use of the step doubling method.

where β is a "safety" factor.

Runge-Kutta-Fehlberg method

The alternative stepsize adjustment algorithm is based on the *embedded Runge-Kutta formulas*, originally invented by Fehlberg and is called *the Runge-Kutta-Fehlberg methods (RKF45)* [11, 10]. At each step, two different approximations for the solution are made and compared. Usually an fourth-order method with five stages is used together with an fifth-order method with six stages, that uses all of the points of the first one. The general form of a fifth-order Runge-Kutta with six stages is

$$\begin{aligned} k_1 &= f(t, \mathbf{x}), \\ k_2 &= f(t+\alpha_2 h, \mathbf{x}+h\beta_{21}k_1), \\ &\vdots \\ k_6 &= f(t+\alpha_6 h, \mathbf{x}+h\sum_{j=1}^{5}\beta_{6j}k_j). \end{aligned}$$

The embedded fourth-order formula is

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\sum_{i=1}^{6} c_i k_i + \mathcal{O}(h^5).$$

And a better value for the solution is determined using a Runge-Kutta method of fifth-order:

$$\mathbf{x}^*_{n+1} = \mathbf{x}_n + h\sum_{i=1}^{6} c^*_i k_i + \mathcal{O}(h^6)$$

The two particlular choises of unknown parametrs of the method are given in Tables A.4–A.5.

The error estimate is

$$\varepsilon = |\mathbf{x}_{n+1} - \mathbf{x}^*_{n+1}| = \sum_{i=1}^{6}(c_i - c^*_i)k_i.$$

Table A.4 Fehlberg parameters of the Runge-Kutta-Fehlberg 4(5) method.

1/4	1/4					
3/8	3/32	9/32				
12/13	1932/2197	-7200/2197	7296/2197			
1	439/216	-8	3680/513	-845/4104		
1/2	-8/27	2	-3544/2565	1859/4104	-11/40	
	25/216	0	1408/2565	2197/4104	-1/5	
	16/135	0	6656/12825	28561/56430	-9/50	2/55

Table A.5 Cash-Karp parameters of the Runge-Kutta-Fehlberg 4(5) method.

1/5	1/5					
3/10	3/40	9/40				
3/5	3/10	-9/10	6/5			
1	-11/54	5/2	-70/27	35/27		
7/8	1631/55296	175/512	575/13828	44275/110592	253/4096	
	37/378	0	250/621	125/594	512/1771	
	2825/27648	0	18575/48384	13525/55296	277/14336	1/4

As was mentioned above, if we take the current step h and produce an error ε, the corresponding "optimal" step h_{opt} is estimated as

$$h_{opt} = \beta h \left(\frac{\varepsilon_{tol}}{\varepsilon} \right)^{0.2},$$

where ε_{tol} is a desired accuracy and β is a "safety" factor, $\beta \simeq 1$. Then if the two answers are in close agreement, the approximation is accepted. If $\varepsilon > \varepsilon_{tol}$ the step size has to be decreased, whereas the relation $\varepsilon < \varepsilon_{tol}$ means, that the step size are to be increased in the next step.
Using Eq. (A.10), the optimal step can be often written as

$$h_{opt} = \begin{cases} \beta h \left(\frac{\varepsilon_{tol}}{\varepsilon} \right)^{0.2}, & \varepsilon \geq \varepsilon_{tol}, \\ \beta h \left(\frac{\varepsilon_{tol}}{\varepsilon} \right)^{0.25}, & \varepsilon < \varepsilon_{tol}, \end{cases}$$