Ordinary Differential Equations

as an alternative to agent-based modelling

eX Modelo school

OpenMOLE

June 26, 2019

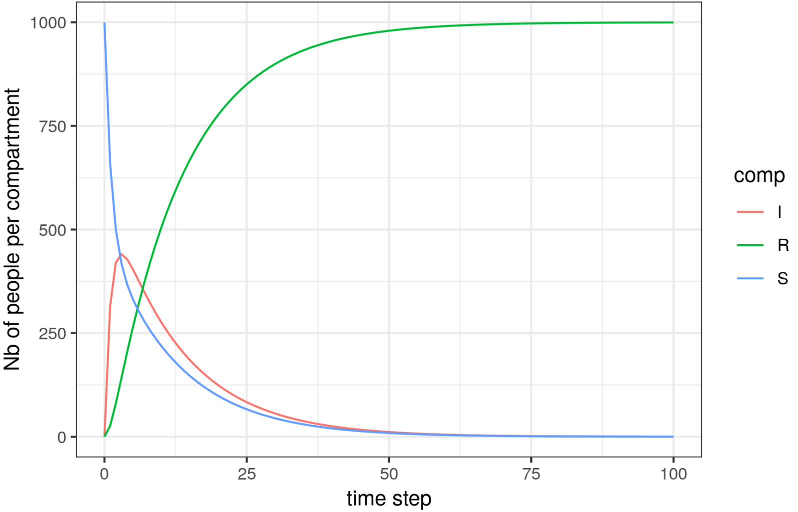
ODE systems

$\rightarrow$ widely used to model transmission phenomena

- population split into compartments
- system of ordinary differential equations

$$\begin{cases} \dfrac{\mathrm{d}S}{\mathrm{d}t} = -\beta S + \lambda I \\[2ex] \dfrac{\mathrm{d}I}{\mathrm{d}t} = \beta S - (\lambda + \gamma) I \\[2ex] \dfrac{\mathrm{d}R}{\mathrm{d}t} = \gamma I \end{cases}$$

SIR model dynamics



ODE	ABM
Equation-based	Individual-based
Generic mechanisms	Precise mechanisms
Population scale	Individual scale
Needs less resources	Computationally expensive

A Zombie situation

How could we model the Zombie invasion?

- Which mechanisms?
- Which parameters?

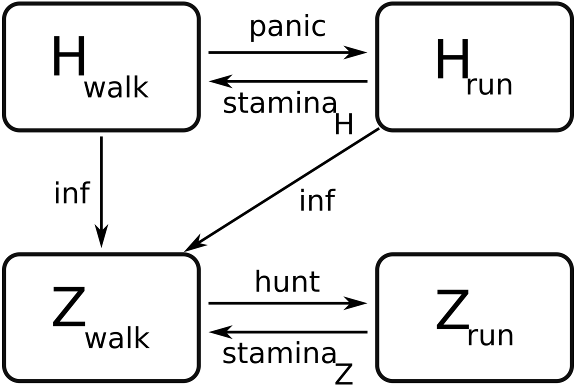
How could we model the Zombie invasion?

- Which mechanisms?
- Which parameters?

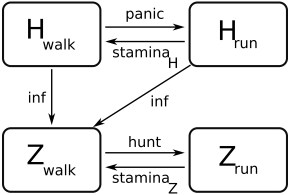
How can we assess our model's ability to reproduce the real data?

- Which metrics?
- Which fitness function?

A very simple ODE model



A very simple ODE model



$$
\begin{cases}
\dfrac{\mathrm{d}H_{walk}}{\mathrm{d}t} = -(panic + inf) * H_{walk} + exhaustH * H_{run} \\[2ex]
\dfrac{\mathrm{d}H_{run}}{\mathrm{d}t} = panic * H_{walk} - (exhaustH + inf) * H_{run} \\[2ex]
\dfrac{\mathrm{d}Z_{walk}}{\mathrm{d}t} = inf * (H_{walk} + H_{run}) - hunt * Z_{walk} + exhaustZ * Z_{run} \\[2ex]
\dfrac{\mathrm{d}Z_{run}}{\mathrm{d}t} = hunt * Z_{walk} - exhaustZ * Z_{run}
\end{cases}
$$

A very simple ODE model

OpenMOLE

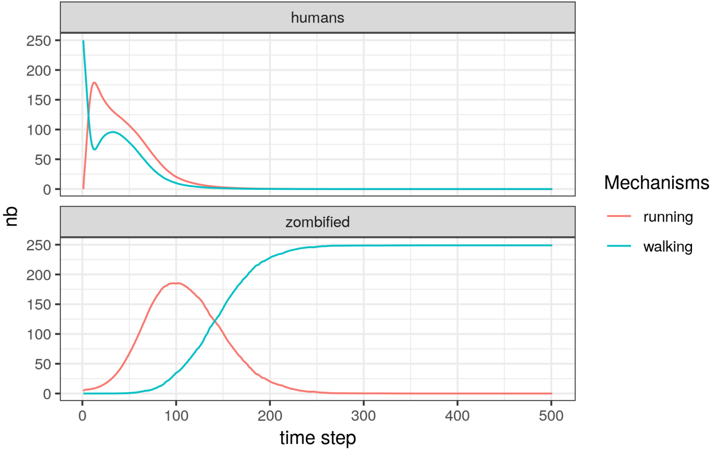
$$\begin{cases} N & = & H_{walk} + H_{run} + Z_{walk} + Z_{run} \\ panic & = & panic_0 * (Z_{walk} + Z_{run})/N \\ hunt & = & hunt0 * (H_{walk} + H_{run})/N \end{cases}$$

Exploration

First step: Calibrate

We have some real time series of zombie invasion
$\rightarrow$ find the parameter values to best fit them

We have some real time series of zombie invasion
$\rightarrow$ find the parameter values to best fit them



Process

Process

- Embed the model in OpenMOLE

First step: Calibrate

Process

- Embed the model in OpenMOLE
- Define a fitness function

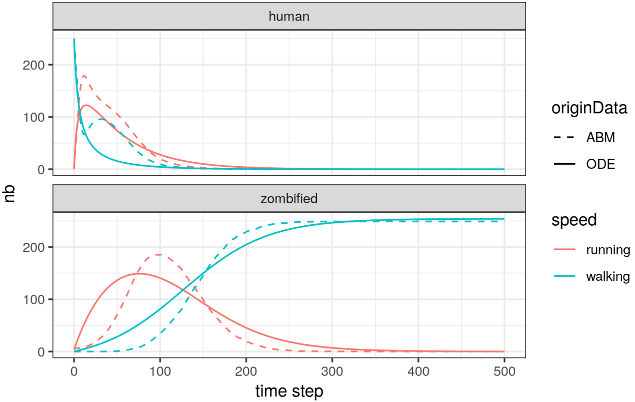
Process

- Embed the model in OpenMOLE
- Define a fitness function
- Write a calibration task

OpenMOLE

Parameter set

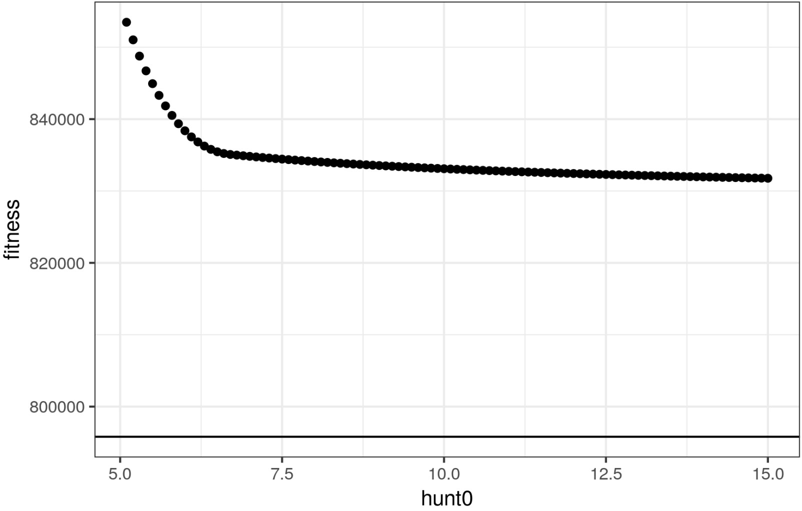
$panic_0$	7.25
$stamina_H$	0.99
$hunt_0$	10.15
$stamina_Z$	1.28
inf	0.02
fitness	780 394



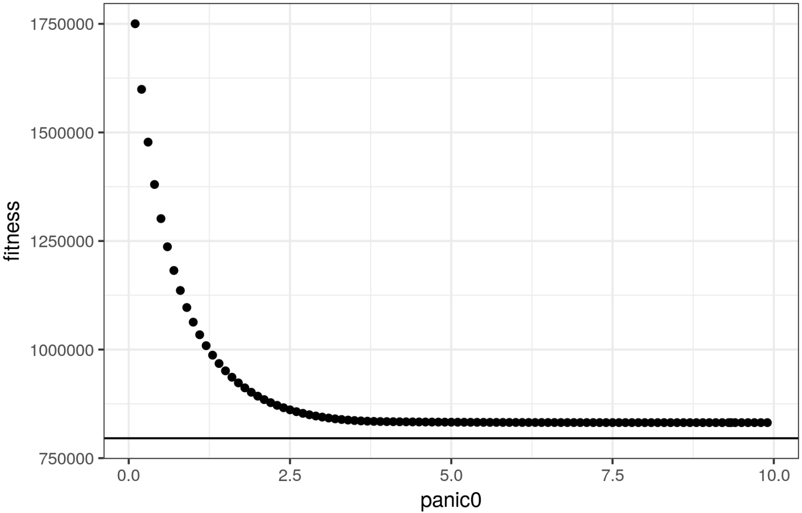
Second step: Profiles

Profiles for each of the 5 parameters:
$rightarrow\ panic_0, stamina_H, hunt_0, stamina_Z, inf_0$

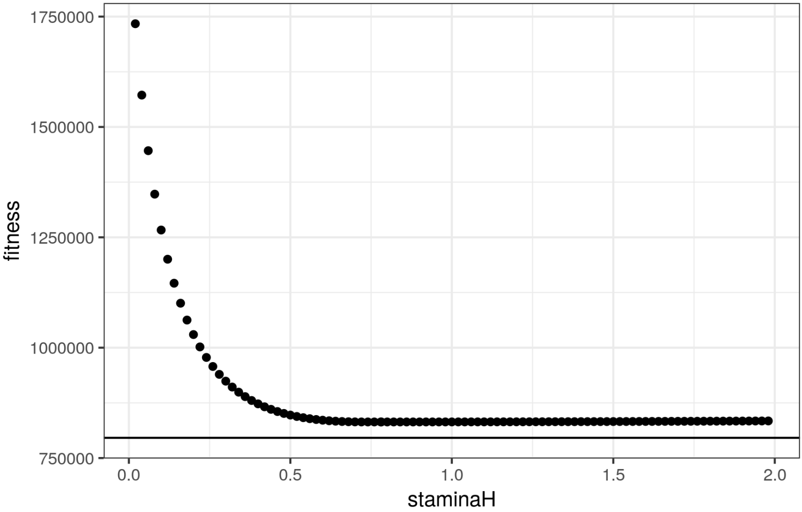
Profiles



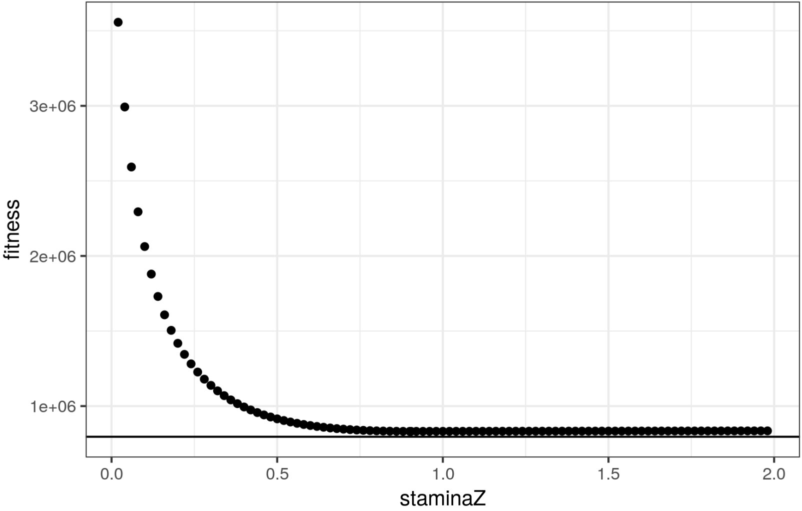
Profiles

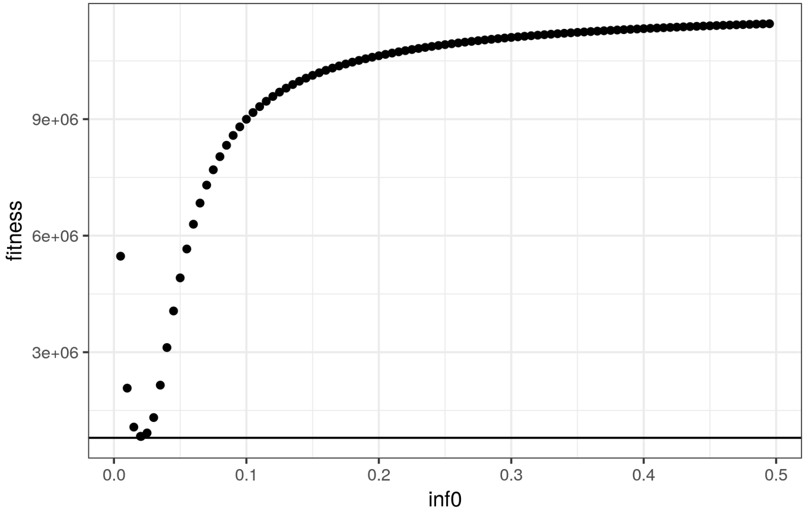


1750000
1500000
1250000
1000000
750000
fitness
0.0
2.5
5.0
7.5
10.0
panic0


1750000
1500000
1250000
1000000
750000
fitness
0.0
0.5
1.0
1.5
2.0
staminaH

Profiles




fitness
9e+06
6e+06
3e+06
0.0
0.1
0.2
0.3
0.4
0.5
inf0

Adding complexity

New mechanisms

What mechanisms could we add to better represent the complexity of our Zombie situation?

What mechanisms could we add to better represent the complexity of our Zombie situation?

The parcimony issue

What mechanisms could we add to better represent the complexity of our Zombie situation?

The parcimony issue

- Do the new mechanisms really improve the fitness?

What mechanisms could we add to better represent the complexity of our Zombie situation?

The parcimony issue

- Do the new mechanisms really improve the fitness?
- Do we need them all?

What mechanisms could we add to better represent the complexity of our Zombie situation?

The parcimony issue

- Do the new mechanisms really improve the fitness?
- Do we need them all?
- What are the best combinations?

Process

- Embed the model in OpenMOLE **DONE**

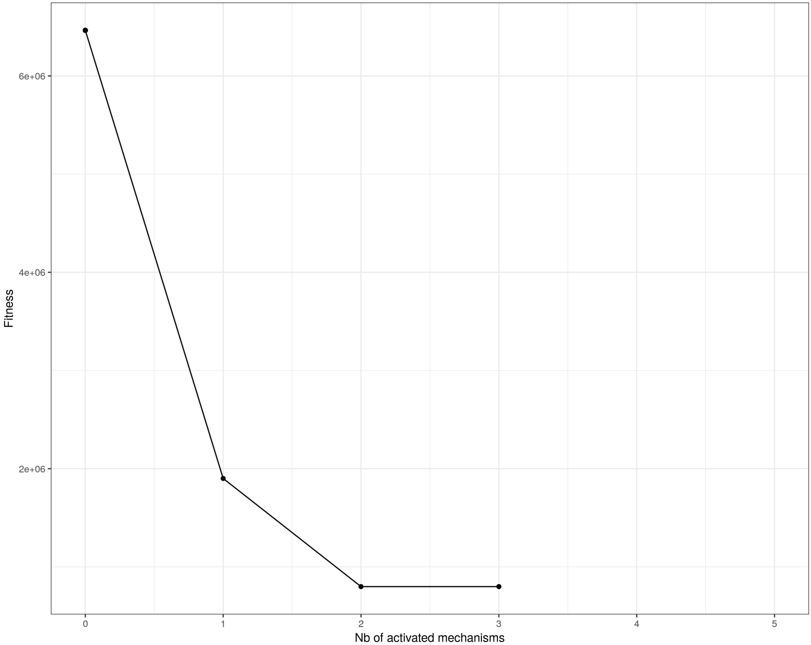
Process

- Embed the model in OpenMOLE **DONE**
- Define a **second** fitness function

Process

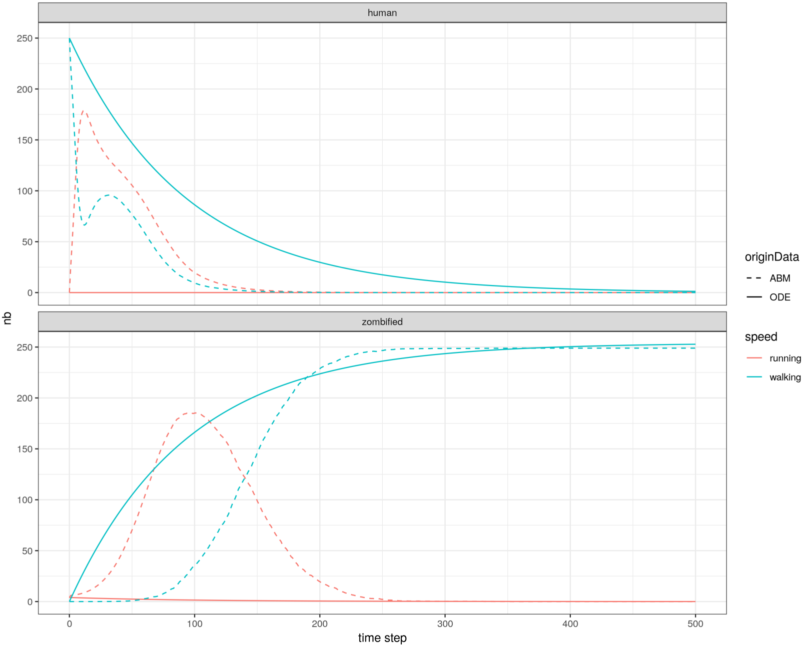
- Embed the model in OpenMOLE **DONE**
- Define a **second** fitness function
- **Modify** the calibration task

Pareto front

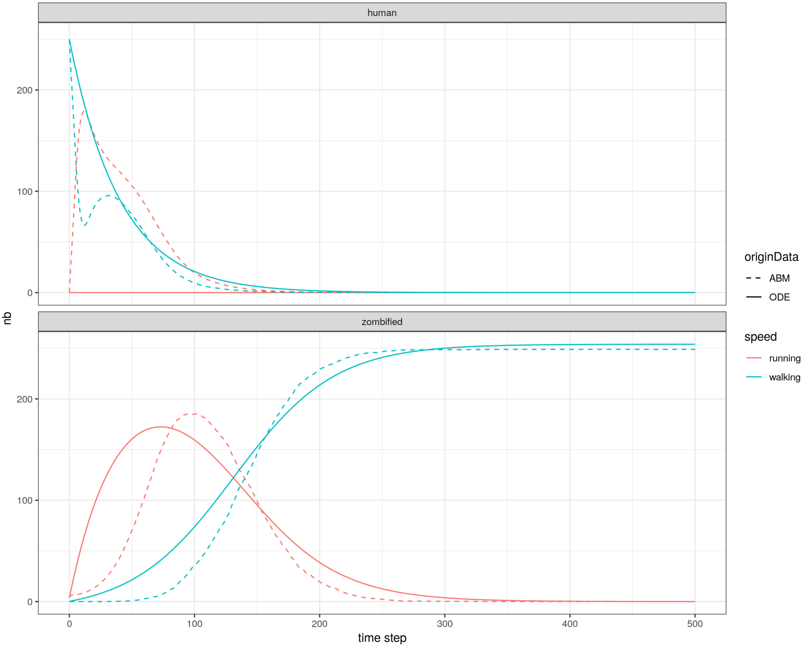


Dynamics for 0 mechanism activated

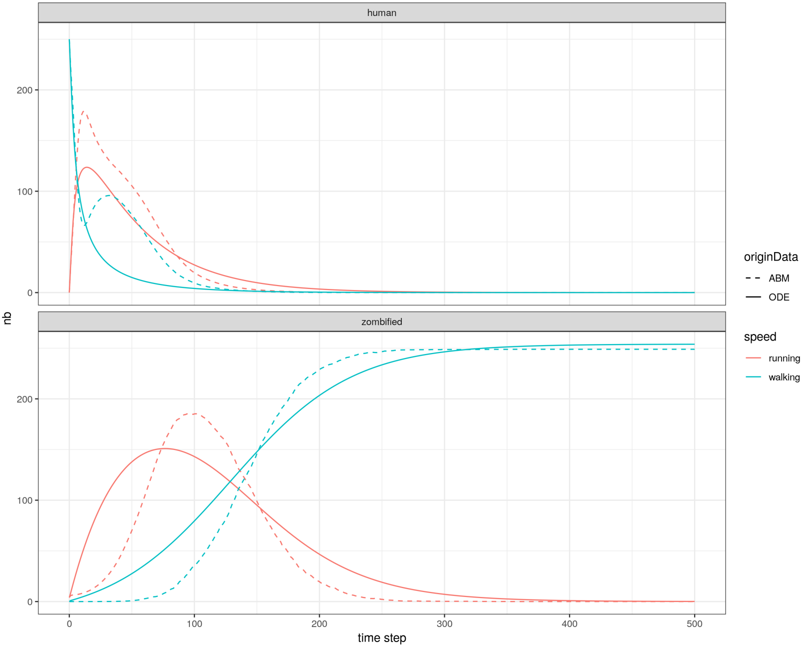
OpenMOLE



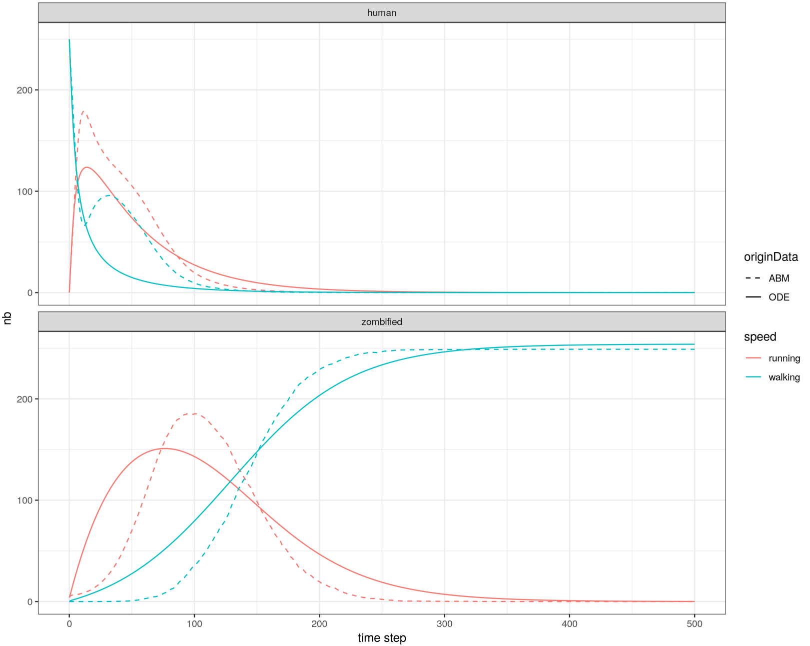
Dynamics for 1 mechanism activated



Dynamics for 2 mechanisms activated



Dynamics for 3 mechanisms activated



Some mathematics

Recall the ODE system:

$$\begin{cases} \dfrac{\mathrm{d}H_{walk}}{\mathrm{d}t} & = & -(panic0 * \dfrac{Z_{walk} + Z_{run}}{N} + inf) * H_{walk} + exhaustH * H_{run} \\ \dfrac{\mathrm{d}H_{run}}{\mathrm{d}t} & = & panic0 * \dfrac{Z_{walk} + Z_{run}}{N} * H_{walk} - (exhaustH + inf) * H_{run} \\ \dfrac{\mathrm{d}Z_{walk}}{\mathrm{d}t} & = & inf * (H_{walk} + H_{run}) - hunt0 * \dfrac{H_{walk} + H_{run}}{N} * Z_{walk} + exhaustZ * Z_{run} \\ \dfrac{\mathrm{d}Z_{run}}{\mathrm{d}t} & = & hunt0 * \dfrac{H_{walk} + H_{run}}{N} * Z_{walk} - exhaustZ * Z_{run} \end{cases}$$

Where $N := H_{walk} + H_{run} + Z_{walk} + Z_{run}$

First-order **nonlinear** (autonomous) ordinary differential equation... a priori no explicit solution, hence numerical solutions.

Total population size

Summing the 4 equations, we have:

$$\frac{\mathrm{d}N}{\mathrm{d}t} = \frac{\mathrm{d}H_{walk}}{\mathrm{d}t} + \frac{\mathrm{d}H_{run}}{\mathrm{d}t} + \frac{\mathrm{d}Z_{walk}}{\mathrm{d}t} + \frac{\mathrm{d}Z_{run}}{\mathrm{d}t} = 0$$

So $N(t)$ is constant:

$$N(t) = N(0), \text{ for all } t.$$

We find back that the population size (human + zombies) is constant: natural !

Human / zombies population size

Let's note $H := H_{walk} + H_{run}$
Summing the first equations, we have:

$$\frac{\mathrm{d}H}{\mathrm{d}t} = \frac{\mathrm{d}H_{walk}}{\mathrm{d}t} + \frac{\mathrm{d}H_{run}}{\mathrm{d}t} = -inf * (H_{walk}(t) + H_{run}(t)) = -inf * H(t)$$

First-order linear ordinary differential equation with constant coefficient: explicit solution !

$$H(t) = H(0) * e^{-inf * t}$$

Likewise,

$$Z(t) := Z_{walk} + Z_{run} = N - H(t)$$

Equilibrium points

Definition: The point $x \in \mathbb{R}^4$ is an *equilibrium point* for the differential equation $X' = F(X)$ if $F(X) = 0$.

For example, points of the form $(0, 0, N, 0)$ are equilibrium points. They correspond to a population composed of walking zombies.

New mechanism

OpenMOLE

$$
\begin{cases}
\dfrac{\mathrm{d}H_{walk}}{\mathrm{d}t} & = & -(panic + inf + out) * H_{walk} + exhaustH * H_{run} \\
\dfrac{\mathrm{d}H_{run}}{\mathrm{d}t} & = & panic * H_{walk} - (exhaustH + inf + out) * H_{run} \\
\dfrac{\mathrm{d}Z_{walk}}{\mathrm{d}t} & = & inf * (H_{walk} + H_{run}) - (hunt + die) * Z_{walk} + exhaustZ * Z_{run} \\
\dfrac{\mathrm{d}Z_{run}}{\mathrm{d}t} & = & hunt * Z_{walk} - (exhaustZ + die) * Z_{run}
\end{cases}
$$

New mechanism

$$
\begin{cases}
N & = & H_{walk} + H_{run} + Z_{walk} + Z_{run} \\
panic & = & panic0 * (Z_{walk} + Z_{run})/N \\
hunt & = & hunt0 * (H_{walk} + H_{run})/N \\
inf & = & inf0 * (1 - fightback) \\
out & = & out0 * (H_{walk} + H_{run})/N \\
die & = & die0 * (H_{walk} + H_{run})/N
\end{cases}
$$

However, the population size is no more constant, due to $out0$ and $die0$.

Then, $H := H_{walk} + H_{run}$ satisfies :

$$\frac{\mathrm{d}H}{\mathrm{d}t} = -(inf + out0.\frac{H}{N}).H$$

for which a solution is not as simple as for the previous model.