Abstract

Our analysis explores each aspect of the dragon's characteristics and behavior using a suite of quantitative and qualitative methods.

First we approximated various body dimensions by extrapolating from known data by scaling with respect to fixed proportions, which provided reasonable numbers to use in more sophisticated models. Next we gauged ecological impact by considering both a predator-prey scenario and effect on food webs. The *Lotka-Volterra* equations and *Michaelis-Menton* harvesting models were considered. The stability of the proposed food webs were analyzed using *Pimm-Lawton* eigenanalysis. Using these models, we concluded that letting a dragon loose will run the risk of destabilizing a typical ecosystem without the presence of another competing super-predator.

Metabolic scaling relations were used to estimate daily energy expenditure (DEE), as well as daily energy intake (DEI). To make our analysis more specific to the dragon, we used the *Flight* Software by Pennycuick, to compute chemical and mechanical power theoretically needed for a dragon to sustain level flight. We also focused on the dragon's unique hunting habits, e.g. using fire to roast its prey before consumption, analyzing the net gain in energy of consuming various prey species using USDA data.

An estimate of rate of energy expenditure during flight of $38,000$ kCal for a small dragon (~ 6 years old, $1,500$kg, 44.8m wingspan, 500m wing area) was obtained for standard atmospheric conditions.

In the second part of the paper, we employed a macroscopic random walk simulation, which was partially driven by a klinokinetic factor that biased the dragon towards warmer climates, whose purpose was twofold:

- Gauge climate-based differences in energy expenditure
- Obtain a visualization of the home range of a dragon in the form of a probability density distribution

In order to carry out the random simulations, we created a discretized map of Westeros and embedded a heat density gradient in it. Then, we modelled flight of free-roaming dragons by *correlated random walks* with *differential klinokinesis* to account for their affinity for warmer climates. From this model of movement, we computed the energy expenditure of a dragon as it flew over the arctic, temperate, and arid regions in the North, middle, and South, respectively. Continent-wide roaming was also considered.

It was found that dragon roaming in the cold climates in the north or warm climates in the south can have a difference in energy expenditure of up to **4.2%** compared with a dragon flying in the temperate central basin of the map. Moreover, we performed a *kernel density estimation* with a Gaussian kernel on samples of 100 random walks each to obtain regions of likelihood for a dragon's position during a day-long flight given a starting location; this gave us clear qualitative estimates and visualizations of a dragon's home-range, and hence the area required for a dragon to live.

Key Words: **Random Walks**, **Klinokinetic Factor**, **Kernel Density Estimation**, **Stability Analysis**, **Lotka-Volterra**, **Allometry**

Contents

0.1 Introduction

DRAGONS occupy a central role in the storyline of the mega-franchise that is *Game of Thrones*. They are at once coveted prizes and feared weapons of war—capable of inflicting destruction on a massive scale. Gaining a dragon's alliance would mean gaining a tremendous advantage over the rest of the warring factions and turning the tides of the plot-line.

According to lore, the dragons were once spread far and wide in the realm of Westeros, ruling alongside the Targaryens, who sat on the throne for three centuries before being driven into exile. In the course of the Rebellion that embroiled Westeros in war—pitting army against army and dragon against dragon—they were rendered all but extinct save for three petrified eggs, vestiges of the War of the Five Kings. [6]

It is natural to wonder if a dragon could walk the Earth (Westeros) while obeying the laws of reality – magic put aside. Should the dragons of Westeros be reified as living, breathing creatures subject to the constraints of physical and thermodynamic laws, how would they figure into a realist's conception of the world?

Using a medley of approaches developed in our human world, including but not limited to **random walks** on a map of Westeros equipped with a heat gradient, **stability analysis**, **dynamical systems theory**, and **kernel density estimation**, we attempt to shed light on the dragon's behavior both at a macroscopic and organismal level. For the former, we mapped out migration flight paths over various climates and estimated home ranges. For the latter, we modelled growth trajectory, metabolic rate, diet, habits, and energy expenditure. Last, we briefly consider real world applications of our analysis.

The lack of concrete data in the fiction universe posed a unique challenge to constructing viable models. In response, we dug deep into the fiction universe to uncover as many facts and data points as possible and opted for qualitative analysis in cases where data could not be reasonably inferred or extrapolated—such as when assessing ecological impact.

0.2 General Assumptions

0.2.1 Dragon-Based Assumptions

- Dragons generally share similarities with reptiles and birds. We have specific assumptions related to this below.
- Dragons prefer to stay in warmer climates. When the Valyrians discovered dragons, they were "nesting near the warmth of the Fourteen Fires", a series of volcanoes. [6] Moreover, reptiles and birds both move toward warmer areas during the winter.

- We assume dragons undergo indeterminate growth, i.e. grow continuously from birth to death. This is seen from a variety of sources ([5], [7]), as well as from their readily observed progression in size from Seasons 1 through 7.
- For simplicity we assume that the natural lifespan of a dragon is 200 years. This is reasonable, given known figures such as Balerion's age of death (220 years) and Vhagar's age of death (180 years).
- We assume the dragon is endothermic, meaning that it is capable of generating heat internally while reptiles cannot [15]

0.2.2 Environment

- We only consider the land of Westeros. Almost all of the events in the stories occur in Westeros, and as such we have much more information on it than any other parts of the world that the stories take place in.

0.3 Anatomy, Size, and Growth

The dragons in Game of Thrones adhere to the classical image of a dragon, with dentate ridges of horns that run from the back of the skull to the jawline, serpentine bodies with long necks and tails, and a layer of protective scales that cover its entire body. Although the newborn hatchling weighs a mere 10 kilograms, the weight of a small cat, the dragon grows very quickly kudos to a protein-filled diet and a ravenous appetite, subsisting on a strict diet of meat, be it sheep, fish, horse, or man. [6]

There are incomplete data pertaining to the physical dimensions of the dragon at various ages, and. In order get a clearer picture of progress of the dragon's biological development over time, we assumed that $\ell^3 \propto m$ and that $\ell^2 \propto A$, where ℓ is the length of a representative body part, such as wingspan, m is mass, and A is wing area. Starting with the known statistics in bold, we scaled accordingly to fill in the rest of the table below.

Various Dimensions					
Age (years)	0	1	6	124	220
Weight (kg)	**10**	**35**	**1500**	24,700	137,000
Wingspan (m)	8.43	12.80	**44.80**	**114**	**202**
Head to Tail (m)	6.02	9.14	**32**	81.4	144
Body Length (m)	1.20	1.83	**6.4**	16.28	28.86
Tail (m)	3.60	5.49	**19.2**	48.84	86.58
Skull (m)	0.3	0.458	**1.6**	4.07	7.215
Wing area (m^2)	17.58	40.88	**500**	3,240	10,200

The method of scaling was not perfectly sound, as the numbers generated for a newborn are patently overestimates. The true wingspan of a newborn should be closer to 2 meters,

Symbol	Description
D	Dragon Population $\mathrm{kg\,s^{-1}}$
P	Prey Population
β	Biomass Conversion Efficiency
h	Harvest Rate
m	Death Rate of Predator
r	Growth Rate
d	Half Saturation Constant
w	Maximum Capture Rate
α	Inverse of Carrying Capacity

Table 1: Symbols of Population Models

judging from scenes from the TV series. Overall the estimates are reasonably close, given the paucity of data points.

0.4 Ecological Impact

0.4.1 Prey Model - Lotka-Volterra

- We assume that dragons undergo natural cycles of birth and death. [6]).

The classical Lotka-Volterra predator-prey equations for describing the dynamics of the interaction between consumer and prey populations are given by:

$$\frac{dP}{dt} = rP - hPD \tag{1}$$

$$\frac{dD}{dt} = \beta hPD - mD \tag{2}$$

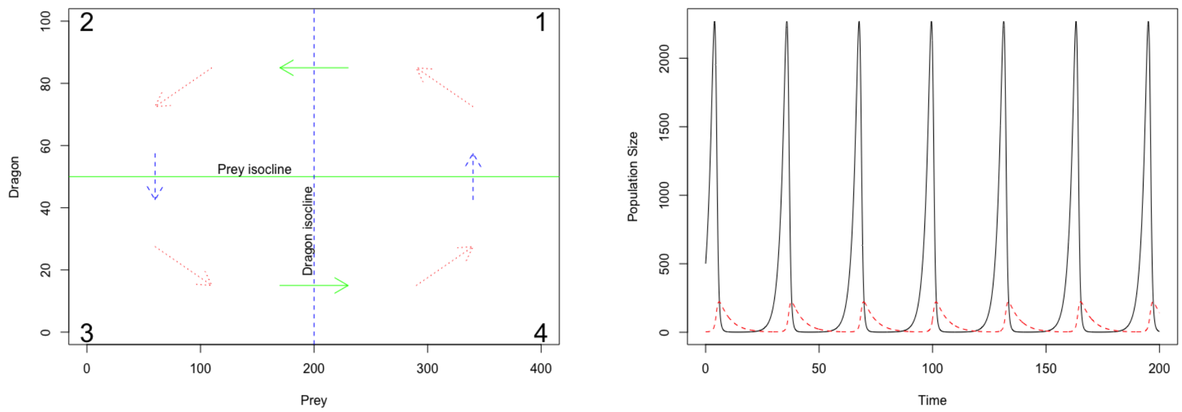
We are interested in finding the **zero net growth isoclines** (ZNGI), which are sets of all points for which the predator/prey growth rates are zero, we set equations (1) and (2) above to zero, because their intersection corresponds to an equilibrium point where both predator and prey populations stop changing. For this case, $(D, P) = (r/h, m/\beta P)$ is the intersection point. The development of predator and prey populations is depicted in Figure 1.

Michaelis-Menton Predator Harvesting Model

We model the population growth of the prey population under the logistic model, and model harvesting by the Michaelis–Menten predator harvesting model.

$$\frac{dP}{dt} = rP(1 - \alpha P) - w\frac{P}{d+P}D \tag{3}$$

$$\frac{dD}{dt} = \beta w\frac{P}{d+P}D - mD \tag{4}$$



(a) Isocline plot for predator and prey populations

(b) Time series plot for dragon (dashed red line) and prey (black line) populations. Observe the cyclical nature of the graph, which is explained by the isocline plot on the left.

Figure 1: Isocline and Time Series Plots

Solving for the isoclines (sets of points where the partials vanish), we get

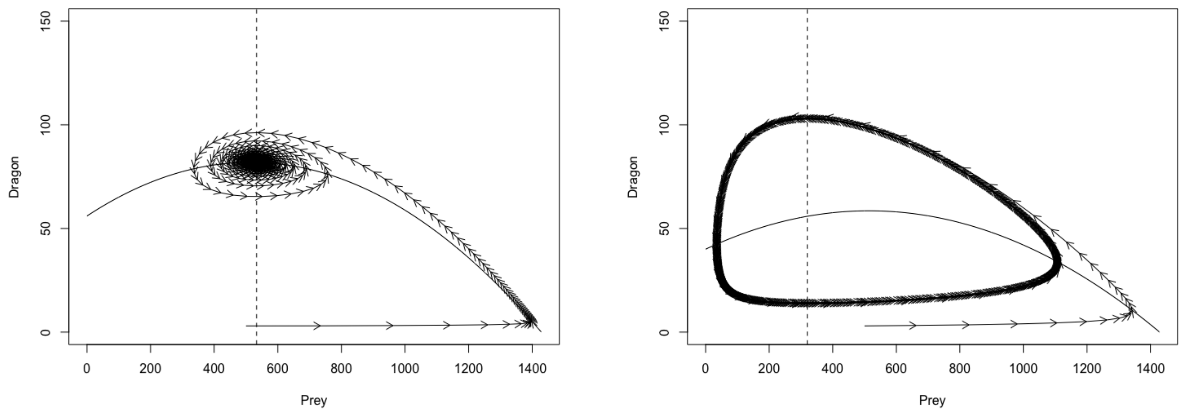
$$P = \frac{md}{\beta w - m} \qquad D = \frac{d+P}{w} r(1-\alpha P) \tag{5}$$

In order to proceed with stability and eigen-analysis, we compute the Jacobian matrix of second partials as follows.

$$\begin{pmatrix} \frac{\partial^2 P}{\partial P^2} & \frac{\partial^2 P}{\partial D^2} \\ \frac{\partial^2 D}{\partial P^2} & \frac{\partial^2 D}{\partial D^2} \end{pmatrix} = \begin{pmatrix} b - 2\alpha b P - \left(\frac{wD}{P+D} - \frac{wPD}{(P+D)^2}\right) & -w\frac{P}{P+D} \\ \frac{\beta w D}{D+P} - \frac{\beta w P D}{(P+D)^2} & \beta w \frac{P}{P+D} - m \end{pmatrix} \tag{6}$$

The dynamical behavior of a large class of predator-prey models (including this one) is dictated by Kolmogorov's theorem, which says there exists either a stable equilibrium or a stable limit cycle. Moreover, the Poincaré–Bendixson Theorem states that when an unstable equilibrium gives rise to a stable equilibrium cycle, it must be contained within the cycle [12].

Assuming that the three dragons are released in a region with one species of prey with initial quantity 500. Then by varying r, β, m, w, d, we obtain various phase portraits shown in **Appendix B**.



(a) There is a stable equilibrium at approximately $(530, 70)$, where both dragon and prey populations stop changing. The parameters are $r = 0.5$, $\beta = 0.07$, $m = 0.2$, $w = 5$, $d = 400$, $\alpha = 0.0007$.

(b) The equilibirum point at which the isoclines intersect is unstable, resulting in a stable limit cycle. The parameters are $r = 0.5$, $\beta = 0.09$, $m = 0.2$, $w = 5$, $d = 400$, $\alpha = 0.0007$

Figure 2: Phase portraits of predator-prey dynamical systems.

0.5 Food Webs and Stability of Ecosystems

0.5.1 Theory

The interaction between various species in a food web is exceedingly complex. Therefore to model impact, we use random simulation to sidestep the intractability of the system differential equations. Return time, formally defined as the time required to reduce a perturbation by 63%, will be used as a measure of the stability of an ecosystem. For our purposes, we will let

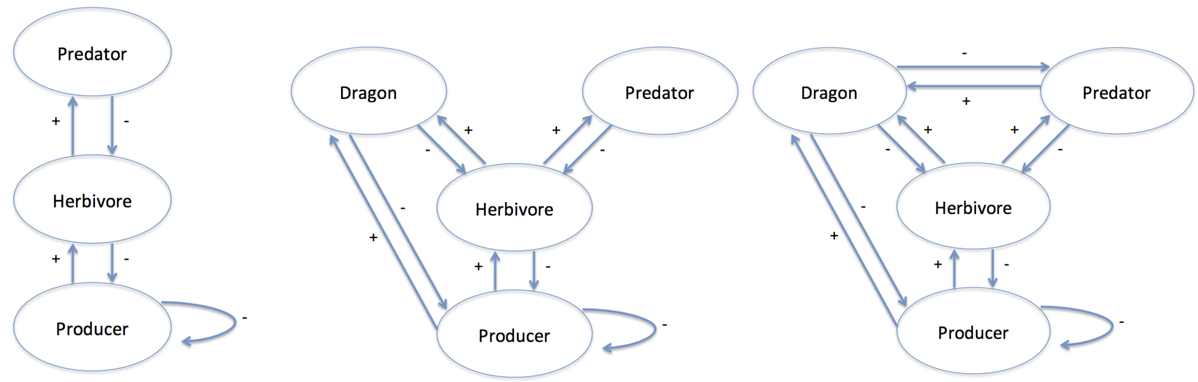
$$R = -\frac{1}{\lambda} \tag{7}$$

where R is the return time and λ is the largest real part of the eigenvalues of the Jacobian matrix of the food web being analyzed [17].

0.5.2 Case Studies

Assumptions

- We assume that in any given ecosystem, the dragon is at the top of the food chain, as they have no known predators, and indeed, few weaknesses at all. Legend says a dragon can only be killed by a spear in the eye or by the fire of another dragons. [6]



(a) (b) (c)

Figure 3: Proposed Ecosystem Food Webs

- We assume that introducing the the dragon to an ecosystem will have an extremely harmful impact on herbivores, as well as a slightly harmful effect to producers like *ghost-grass*, because of **collateral damage** from lighting its prey on fire.

Method

To model the ecosystem in question, we begin with a simple model with with three trophic levels: predator, herbivore, and producer (**Figure 3a**). We introduce the dragon in **Figure 1b**, and an alternate configuration in **Figure 1c**.

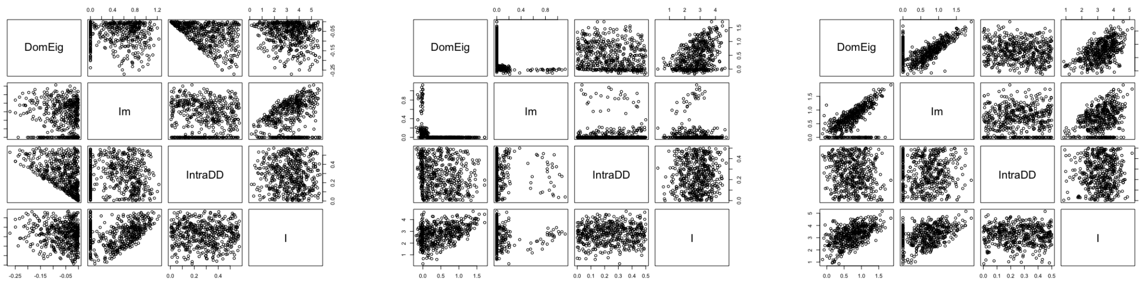
To conduct stability analysis on the proposed food webs, we refer to the numerical method outlined in [17], which consists of specifying food web interaction matrix, which represents a template for randomly generating Jacobian matrices that encode the rate of change of the growth rate of each species with respect to other species.

We generate 500 Jacobians for each web, extract relevant statistics such as $\max \mathrm{Re}\{\lambda\}$ (real part of dominant eigenvalue), Im (imaginary part of DomEig), and I (Average Interaction Strength), and negative denity dependence, and plot the data in **Figure 3**, to show qualitative relationships between the variables using the PimmLawton functions in **R**.

We postulate that the food web interaction matrices [17] corresponding to Food Webs a), b), and c) (**Figure 3**) have the forms, respectively:

$$
\begin{array}{c c}
 & \begin{array}{ccc} a & b & c \end{array} \\
\begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} -1 & -10 & 0 \\ 0.1 & 0 & -10 \\ 0 & 0.1 & 0 \end{bmatrix}
\end{array}
\quad
\begin{array}{c c}
 & \begin{array}{cccc} a & b & c & d \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} -1 & -10 & 0 & -1 \\ 0.1 & 0 & -10 & -10 \\ 0 & 0.1 & 0 & 0 \\ 0.1 & 0.1 & 0 & 0 \end{bmatrix}
\end{array}
\quad
\begin{array}{c c}
 & \begin{array}{cccc} a & b & c & d \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} -1 & -10 & 0 & -1 \\ 0.1 & 0 & -10 & -10 \\ 0 & 0.1 & 0 & -10 \\ 0.1 & 0.1 & 0.1 & 0 \end{bmatrix}
\end{array}
$$

By definition, entry (i, j) describes either the extent of damage that can be done by j



(a) There appears to be no relationship between intraspecific density dependence (IntraDD) and interaction strength (I). Also, since the dominant eigenvalues are negative, perturbations dissipate quickly.

(b) Since the dominant eigenvalues are positive, the perturbations would increase in intensity. There is no discernable relation between IntraDD and interaction strength.

(c) It is interesting to note that for food web 3, there is no correlation between IntraDD and the real/imaginary parts of the dominant eigenvalue. Also, there appears to be a strong positive correlation between the real and imaginary parts of the dominant eigenvalue, which is not observed in the previous plots.

Figure 4: Pimm-Lawton Plots

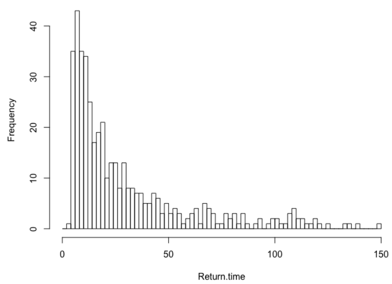
on i (a negative number), or the positive effect i can have on j because j feeds on i. In the stability analysis itself, Jacobian matrices are randomly generated using the food web interaction matrix. A random number is generated in the range M_{ij}. In the matrices above, (a, b, c, d) correspond to $(\text{producer}, \text{herbivore}, \text{predator}, \text{dragon})$, respectively.

0.5.3 Results

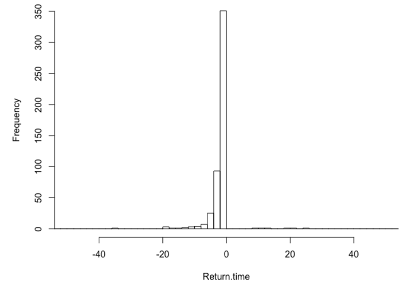
When the dragon has is postulated to have a detrimental effect on all trophic levels in the food web **Figure 4c**, the correlation between intraspecific negative density dependence of the producer and other statistics is completely erased

We also glimpse the impact that the dragon has on a relatively stable ecosystem (**Figure 4a**), characterized by small, positive return rates. Adding the dragon into the mix, specifically in the fashion shown in **Figure 3b**, we find that the return rates are mostly negative (**Figure 5b**).

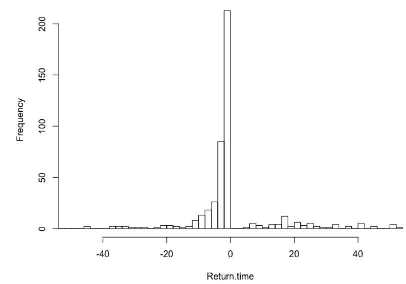
That the the return times associated with food webs b) and c) are negative indicates that the perturbation "was closer to zero in the past", and thus is increasing in magnitude – a sure sign of instability.



(a) A right-skewed distribution of return times. Most return rates are relatively small and positive, which indicates stability. positive.



(b) Food web



(c) Food Web

Figure 5: Histograms of return rates for simulations involving food webs a), b), and c). Return rate $R = -1/\lambda_{max}$ is used as a measure of stability of a food web. Negative R value signifies instability, as does large return rates. Food webs b) and c) are less stable than a), which suggests the introduction of a super-predator like the dragon can destabilize the system.

0.6 Energy Part I

0.6.1 Energy Expenditure via Allometry and Kleibler's Law

- We assume allometric equations used for carnivorous mammals can accurately model dragons.

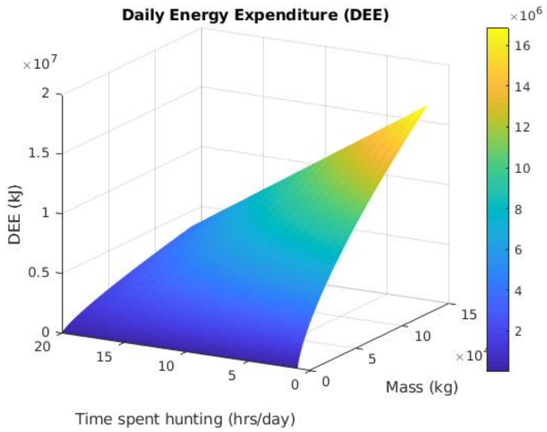
An easily noticeable characteristic of the dragon is its tremendous size. Adult dragons were known to swallow mammoths whole and could engulf a town in its shadow by flying over it. As can be seen in **Figure 5**, mature dragons can have a wingspan of over 3000m and a weight of over $24,000$kg. Size-related variation, however, can often times be aptly described by the so-called allometric equations, which take the form

$$Y = Y_0 M^b \tag{8}$$

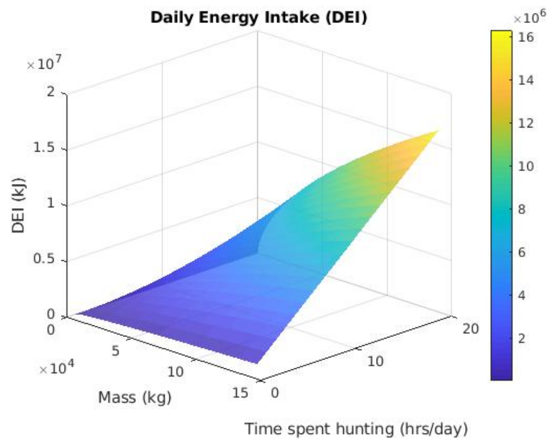
The results derived by Carbone et al allows us to obtain an estimate of the energy expenditure and net energy gain of the dragon in one fell swoop:

$$\begin{aligned} \text{NEG} &= 0.66(I)T_H - E_r T_r - E_h T_h && (9) \\ &= 0.66 I T_h - 5.5 M^{0.75} T_r - (10.7 M^{0.684} v + 6.03 M^{0.697}) T_h && (10) \end{aligned}$$

where NEG is net energy gain, E_h and E_r are the energy expenditure rates while hunting and resting (kj/hr), T_h and T_r are the times spent hunting and resting (kj/day), and M is body mass in kg, and I is intake rate per hour hunting (kj/hr), v is average speed while



(a) Plot of Daily energy expenditure versus T_H and m. There is a direct, albeit nonlinear, relationship between the variables.



(b) Daily energy intake as a function of T_H and m.

hunting m/s, and 0.66 is the assimilation coefficient. I was estimated to be $1010M^{0.6}$, so for a 1500kg dragon, the hunting efficiency is estimated to be $81,000$kj/h [4].

The drawback of this model is that it does not directly take into account the dragon's unique mechanism of hunting (roasting its prey alive) or its flight capability, both of which may cause the dragon's energy requirements and metabolic characteristics to deviate from expected. To provide a more accurate estimate in context, we address the factors of fire and flight specifically in the next energy model.

0.7 Energy Part II

0.7.1 Quantifying the Energy of Flight

Assumptions

- We assume that the dragons have a streamlined body for ease of calculation. This body type is characterized by a "circular cross section" with the "the widest cross section roughly a quarter to a third of the body length behind the front end, and the rear end tapering to a point" (Pennycuick, p.55). [13]
- The conditions in Westeros, including gravitational field strength, temperature, and pressure are comparable to that of Earth. This appears reasonable as humans do live in Westeros and appear to have the expected physical response to the environment.

Symbol	Description	Units
P_{ind}	Induced Power	$\text{kg}\,\text{s}^{-1}$
P_{par}	Parasite Power	$\text{kg}\,\text{s}^{-1}$
P_{pro}	Profile Power	$\text{kg}\,\text{s}^{-1}$
P_{mech}	Mechanical Power	
B	Wing Span	
g	Acceleration due to Gravity	
m	Dragon Mass	kg
V_t	True Airspeed	
ρ	Air Density	
S_b	Body Frontal Area	
C_{Db}	Body Drag Coefficient	
ˆ	Superscripts, as in x^2	

Table 2: Constants for Flight Power

The Power Curve

We use *Flight Program* furnished by Pennycuick, to calculate the minimum chemical power – or the rate at which fuel energy is required in aerobic flight – that a dragon with a weight of 1500 kg and wing span and wing area of 44.8m and 500m^2, respectively, needs to maintain level flight under standard atmospheric conditions (air pressure of $1.225kg/m^3$ and at sea level. We use this and related curves below in our calculations of flight energy expenditure.

Since the dragon is suspended in air, it must deliver enough power from its muscles to accelerate air downwards, so as to support the weight of its body. The rate at which work is done with the muscles is called **induced power**. For forward flight, the induced power is given by

$$P_{\text{ind}} = \frac{2k(mg)^2}{(V_t \pi B^2 \rho)}$$

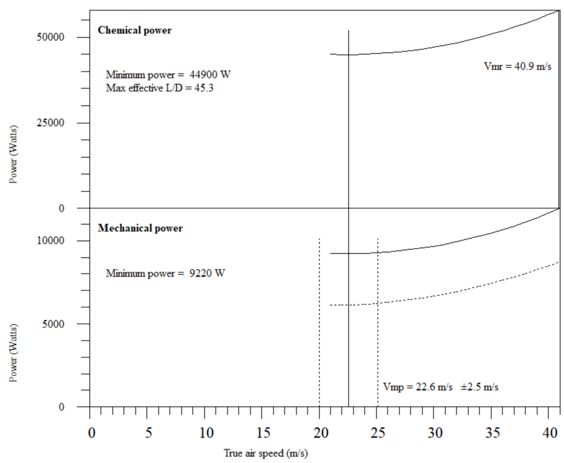
where k is the induced power factor, a constant that is commonly approximated as $1.1 - 1.2$.

In addition to induced power, the dragon must propel its body forward against the resistance of of the air, at least in horizontal flight. The power needed to overcome the drag force is called **parasite power**. Parasite power is given by the formula

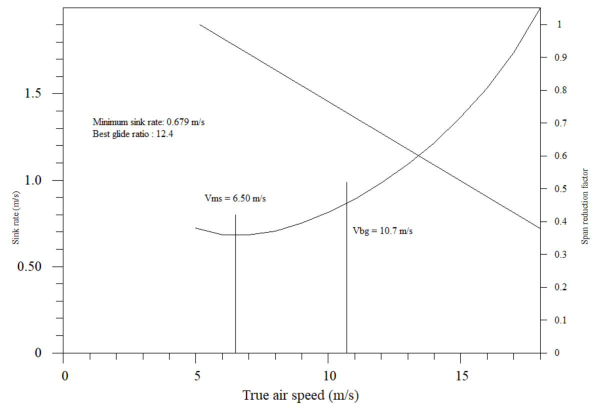
$$P_{\text{par}} = \frac{\rho V_t^3 S_b C_{Db}}{2}$$

where S_b is the body frontal area and C_{Db} the body drag coefficient. To estimate S_b, the formula $S_b = 0.00813m^{0.666}$ is used, where m is in kilograms and S_b is in square meters. On the other hand, an empirically accepted value for the body drag coefficient for flying species is 0.1 [13].

We have the key relation $P_{\text{mech}} = P_{\text{ind}} + P_{\text{par}}$. Chemical power is found from mechanical power by factoring in the inefficiencies of converting fuel into usable energy (Figure 6).



(a) Power curve for dragon weighing 1500kg, with a wingspan of 44.8 m, and wing area of 500m^2, and air density of 1.225 kg/m^3. Observe that minimum mechanical power and chemical power needed to support flight are 9220W is 44900W., respectively



(b) Glide polar curve for dragon weighing 1500kg, with a wingspan of 44.8 m, and wing area of 500m^2, and air density of 1.225 kg/m^3.

Figure 6: Power and Glide polar curves.

0.7.2 Quantifying the Energy of Hunting

- Dragons are obligate carnivores that eat land animals cooked by their fire breath. [6]
- We assume that the dragon's fire-breathing energy efficiency is similar to that of a standard electric oven, i.e. 12%. [21]

Speculating on the mechanism used by the dragons in fire production, we first assume that dragons have a similar digestive tract to that of reptiles. After ingestion, food moves along the gastrointestinal tract for mechanical and chemical breakdown so that absorption of nutrients can take place in the intestines. Bacterial microbiota reside in the gut to aid in the digestion of remaining food particles that the stomach and small intestine were unable to fully breakdown. In this process, gut microbiota release gas (including methane and hydrogen) from fermentation. [11] In reptiles, this excess gas is expelled via bloating or burping. Dragons, on the other hand, can store the excess gas for later use. With a spark caused by swallowed rocks, methane combusts in the presence of oxygen via the following exothermic reaction: [10]

$$CH_4(g) + 2O_2(g) \rightarrow CO_2(g) + 2H_2O(g) + heat$$

We assume that the heat generated from this reaction is channeled into cooking the desired creature with a **12%** efficiency, which is the cooking efficiency of a standard electric oven. That means that **88%** of the heat generated is lost to the surrounding environment.

According to De et al., the minimum amount of heat needed to cook 1 kg of raw goat meat is 626 ± 4 kJ. We assume that 626 kJ is required to fully cook 1 kg of any creature that the dragon wishes to consume.

0.7.3 Results

Based on caloric data from the USDA website, we calculated the total energy a dragon can obtain from consuming sheep, horse, chicken, cattle, and pig, assuming that any given creature is 60% meat by mass, whereas the other 40% of a given creature's mass accounts for water, bones, etc. The following table compiles our estimates of energy losses and gains from roasting and consuming these four creatures.

Animal	Avg. weight (kg)	Energy expended (kJ)	Energy gained (kJ)	Net Gain (kJ)
Sheep	77	2.41×10^5	5.68×10^5	3.27×10^5
Horse	550	1.72×10^6	2.42×10^6	6.99×10^6
Chicken	2	6.26×10^3	1.20×10^4	5.74×10^3
Cattle	753	2.36×10^6	4.10×10^6	1.75×10^6
Pig	80	2.50×10^5	4.86×10^5	2.36×10^5

Table 3: From the USDA, we obtained caloric values by kilogram for lamb, horse meat, chicken, beef, and pork. We assumed that 60% of a creature's mass is meat. With the average weight of sheep, horse, chicken, cattle, and pig and their corresponding caloric values, we estimated the energy gained in kJ from consuming a given creature in the table above. Furthermore, since dragons require their meat to be roasted before consumption, we have also included estimates of the energy expended in kJ, assuming that it takes 626 kJ to roast 1 kg of raw meat.[5] The final column records the net energy gain from roasting and eating each creature.

0.8 Modelling Macroscopic Behavior: The Correlated Random Walk on Westeros

0.8.1 Correlated Random Walk with Differential Klinokinesis

We use a correlated random walk to model the movement of a dragon when it is free to roam. An uncorrelated random walk, in which the next direction of movement is independent of any previous directions, would not be sufficiently accurate since it would give paths that are too erratic. A dragon should tend to continue moving forward in the direction that it was already moving in at a given step, as other bilaterally symmetric animals with cephalocaudal growing patterns tend to. [3] Thus, we use a correlated random walk model based on Bovet and Benhamou [3] in which the direction of movement at each step is the previous direction plus some small normally-distributed shift.

$$\begin{aligned}
\alpha_i &\sim N(0, \sigma) \\
\theta_{i+1} &= \theta_i + \alpha_i \\
X_{i+1} &= X_i + P_i * \cos(\theta_{i+1}) \\
Y_{i+1} &= Y_i + P_i * \sin(\theta_{i+1})
\end{aligned}$$

$$\alpha_i = \text{change in angle at step } i, \quad P = \text{step length at step } i$$

$$\theta_i = \text{direction of movement at step } i, \quad X_i = \text{x-coordinate at step } i$$

$$Y_i = \text{y-coordinate at step } i, \quad \sigma = \text{standard deviation of change in direction}$$

Under our assumptions, dragons prefer areas of warmer climate. Thus, we introduce a temperature gradient $T(x, y)$ on our map of Westeros (see directly below). To account for the preference of dragons to warmer temperatures, we add differential klinokinesis as described by Benhamou and Bovet [2] with respect to the temperature gradient of the map to our model. Adding onto the correlated random walk, we make the step length vary in each step, by setting

$$P_i = \frac{P_b}{(1 - k \cos \tau_i)^2}$$

$$\tau_i = \text{angle between direction of last step } (\theta_i) \text{ and gradient}$$

$$k = \text{klinokinetic factor}, \quad P_b = \text{base step length}$$

The klinokinetic factor is a number in $(-1, 1)$ that quantifies how attracted or repelled a dragon is to the direction of the density gradient, with -1 being most repelled and 1 being most attracted. We use a positive factor that we empirically choose—we could find no relevant quantitative data for this quality of the beasts (see sensitivity analysis for justification). We use numerical differentiation techniques to compute the direction of the temperature gradient in each coordinate. In particular, we take second order central differences, with forward or backward differences instead along the boundary of the map.

0.8.2 Simulation Model

In order to model movement of the dragon about Westeros, we here create a representation of Westeros that we can perform computations on. We take a map of Westeros from the site "A Wiki of Ice and Fire", [20] and represent it in a 1449×563 matrix, with a natural correspondence between pixels and elements of a matrix (awoiaf). We fix the size of Westeros to be 3420 miles $\times 1410$ miles [14], so that each vertical unit has a length of about 2.36 miles and each horizontal a length of about 2.52 miles. For each point on the map, we assign a temperature based roughly on a paper in which researchers publishing under the name of Samwell Tarly model the climate of Westeros. [18] We consider Westeros to be in the winter season throughout these computations, so we take the temperatures for their winter model. Implementation of a random walk can be found in the appendix. For the rest of this text, unless stated otherwise, each simulation of a flight has the following properties, noting that we prefer to overestimate flight distance and speed because sources have varying speculations on properties of dragons:

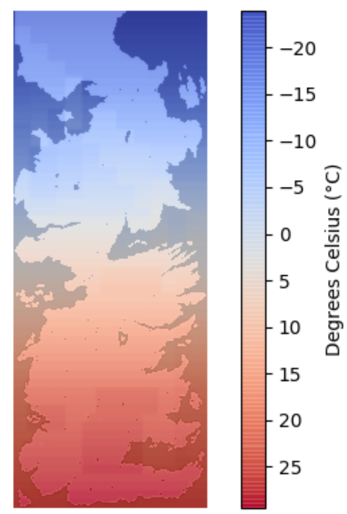


Figure 7: Temperature Map of Westeros during Winter. Temperatures approximately taken from (Tarly [18])

1. We use our above correlated random walk model with differential klinokineses at $k = .2$
2. Temperatures are fixed in each flight.
3. Each flight is $20,000$ steps of a random walk. Each step represent 3 seconds of time, so that each flight is about 16.67 hours long. This approximately models the dragon flying for a day nonstop.
4. We fix a base step size P_b of .1 pixels, so the dragon travels approximately .252 miles per three seconds, and thus about 302.4 miles per hour at base speed. There is much variance in dragon-speed estimations, but this is plausible under some of the higher-speed calculations. [8]
5. The dragon is the same one that we use above, namely a 6 year old, 1500 kilogram beast with power requirements for flight as given in the power curves above.
6. The dragon stays near Westeros and hence does not fly off the map. If it tries to fly past a boundary in a step, it instead stays still that step.

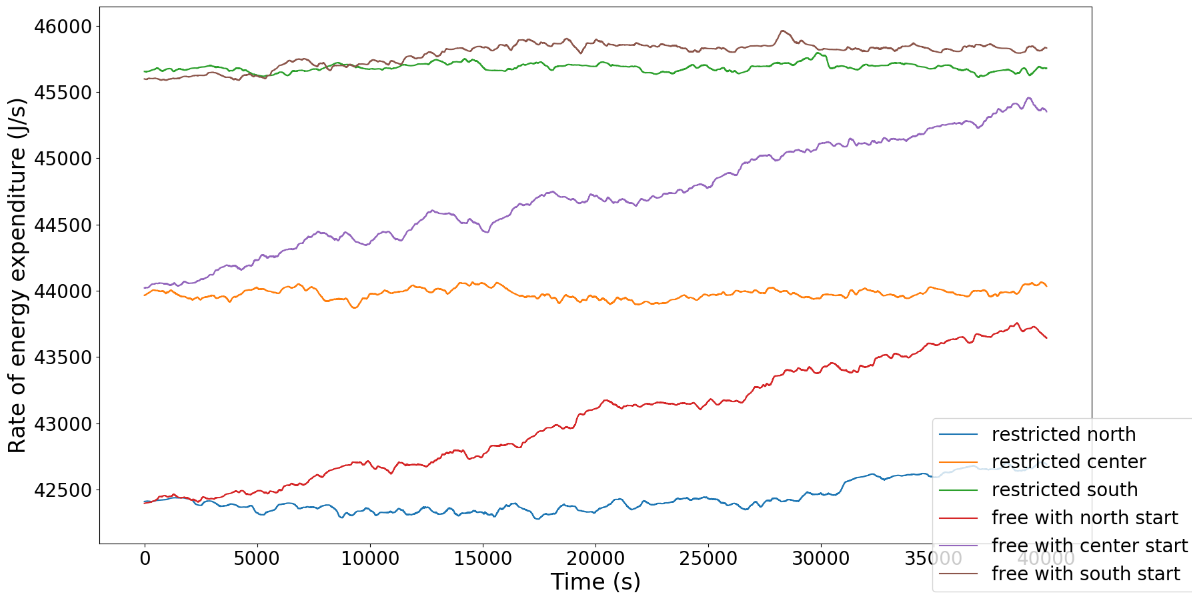


Figure 8: Rate of energy expenditure while flying in different regions of the map. The restricted flights are correlated random walks with no temperature preference ($k = 0$). Note that in the free-roaming flights, power increases over time since the dragon tends towards warmer parts of Westeros.

0.8.3 Energy Expended during Flight

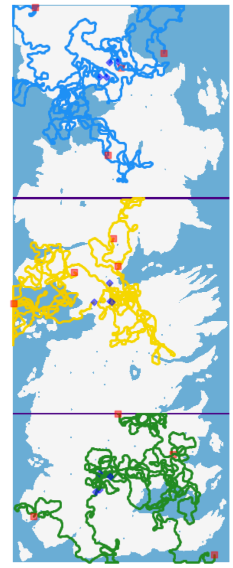
To determine the energy expended during flight, we simplify to an integration of power over time $\int_0^{t_f} P(T(t))\, dt$, where $P(T)$ is power at a given temperature, and $T(t)$ is temperature at time T of a random walk. We have power's relationship to temperature through Pennycuicks *Flight Program* where we input our dragon's properties as stated above. As a dragon moves across Westeros during a flight, the temperature of its environment changes—most importantly also changing the air density. To calculate the energy, we numerically integrate power by the trapezoid method as measured at each step in a flight's random walk.

We make separate energy calculations for flights in different regions to explore possible effects of climate on the resources required to maintain one. We simulate flights constrained to the north, center, and south of Westeros, and also flights in free-roam but starting at either the north, center, or south.

The further south a dragon is flying in Westeros, the more energy it expends. Note that BMR for endotherms increases in climates that are cold as they expend energy to maintain their body heat, but we ignore this effect since in birds, flying expends magnitudes more energy than BMR. [7] As a reference for the size of these numbers, using the 4.396×10^5 kJ expenditure in the flights of the center region, a dragon would need to eat about 1.34 sheep, .63 horses, 76.58 chickens, .25 cattle, or 1.87 pigs to sustain 2.77 hours of flying. Under these numbers, we consider it completely unfeasible to sustain off of chickens, and note that a dragon would have to eat a fair amount of pigs to have sufficient energy for flight.

Number	Region	Energy (kJ)	Change from center
1	Restricted to north	4.241×10^5	-3.53%
2	Restricted to center	4.396×10^5	0
3	Restricted to south	4.566×10^5	3.87%
4	Free-roam starting from north	4.278×10^5	-2.68%
5	Free-roam starting from center	4.439×10^5	.98%
6	Free-roam starting from south	4.581×10^5	4.21%

(a) Energy expenditure in kilojoules for 2.77 hours of flying.

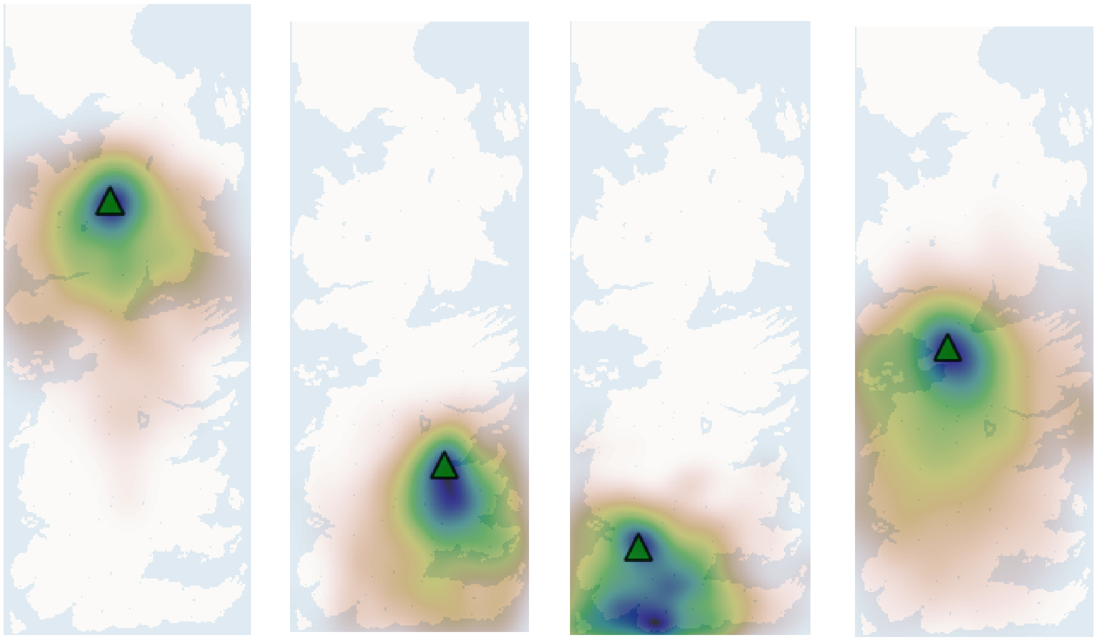


(b) Restricted flight paths in each region. Red squares mark the end of flights.

0.8.4 Home Range Size

In order to determine the approximate areas that dragons spend time around, we use the concept of home-range of a roaming animal—the region in which a given animal lives and moves around. To obtain nice approximations and visualizations for home-range of a dragon, we carry out 100 random walks starting from each of the important locations of Winterfell, King's Landing, the Twins, and Highgarden in Westeros. These locations are major settings in the plot of the stories, so they are like locations for dragons to either stay or be raised. With the paths of these random walks, we use Gaussian kernel density estimation to give a probability density of spots on the map where dragons are most likely to have visited during a flight. Below we plot the intensity of the calculated densities, where the more opaque colors denoted areas where a dragon is more likely to have visited.

Kernel density estimation is a widely used and generally applicable method for home-range analysis. [16] We use a Gaussian kernel for simplicity, so that we can use the implementation in the scipy.stats python package. In our application of the KDE, we include only every 1 in 1500 data points of the random walks into the computation of the density. It is indeed quite a computationally expensive set of calculations, and we noticed some instability when points very close together were included in the sample set, possibly due to linear dependence in covariance matrix computations. Moreover, KDE works effectively on uncorrelated data, and has been empirically shown to be a good model for certain GPS datasets that may have lacking precision ([16], [19])—omitting data points in our implementation somewhat reflects these constraints.



(a) Winterfell (b) King's Landing (c) Highgarden (d) The Twins

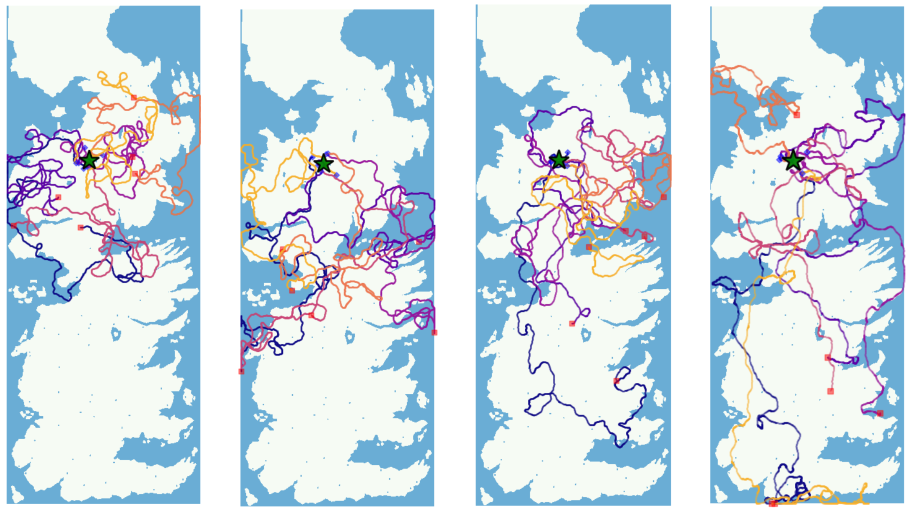
Figure 9: Approximate home range visualization by Gaussian Kernel Density Estimation. 10 random walks were started from Winterfell and 10 from King's Landing to simulate approximate range of a dragon's flight in a free-roam starting from a home location.

We compare the KDEs with a map of Westeros [9] to see regions in which a dragon starting in a certain origin can be expected to be spotted if it were allowed to roam freely. For instance, if a dragon were to start from Winterfell , it would be likely to fly south by the city of White Harbor, southwest by the town of Barrowton and around the forests of Wolfswood right by Winterfell.

0.9 Sensitivity Analysis

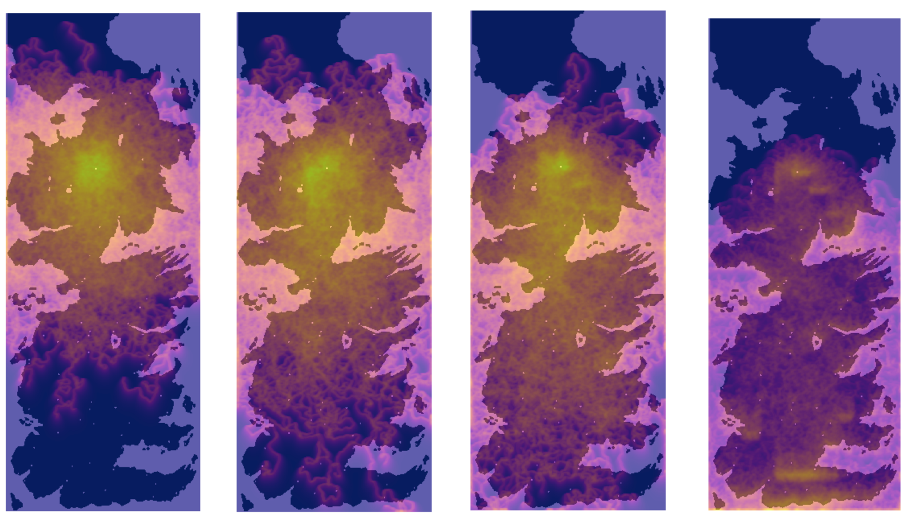
0.9.1 Sensitivity with respect to the Klinokinetic factor

We vary the klinokinetic factor k in our correlated random walk model. As seen in the figures below, increasing k has a significant effect on the location of the dragon. In our above simulations, we used $k = .2$, which seems to be indeed a choice that gives reasonable flight paths. At $k = .1$, the flight paths show little to no preference for warmer regions while at $k = .5$ the flight paths show unreasonable preference for the hottest regions in the south.



(a) $k = .1$ (b) $k = .2$ (c) $k = .3$ (d) $k = .5$

Figure 10: Random walks with different k factors. There are 6 simulated flights per picture, with the green star marking the starting region (Winterfell), blue diamonds near the star marking exact starting locations, and the red squares marking ends of flights. Note that these are random walks, so the variance is high between simulations.



(a) $k = .1$ (b) $k = .2$ (c) $k = .3$ (d) $k = .5$

Figure 11: Heatmaps of duration of flying over regions of the map. The more bright and yellow, the more time that that a dragon has spent in the area. 3500 random walks were simulated for each value of k.

0.10 Real-world applications

Our analysis of dragons drew on principles and practices from areas of theoretical ecology and scientific research, from allometry to predator-prey interactions to the physics of flight.

By allowing the dragons to infiltrate an already existent ecosystem, we have given the dragons the role of an invasive apex predator. Our study illuminates the detrimental impact that invasive species have on native plants and animals, reducing biodiversity and permanently altering the environment. Furthermore, the massive size of the dragons presents an intriguing case study on the energy expenditure of large endotherms.

0.11 Conclusion

All in all, even with the obvious restrictions that reasoning about fictional beasts in a fictional world brings, we were able to identify and alter applicable models developed in our own world. Moreover, due to the contributions of many members of the GOT fanbase, we were able to utilize data collected from the internet that was directly related to the problems that we were dealing with in this paper.

In our sensitivity analysis, varying the klinokinetic factor k shows that our random walk model of flight is behaving as intended, with plausible flight maps and heatmaps that are expected given the equations that govern the random walks. Thus, we can be confident in the output of the model: the energy output in different climates and the approximate home-ranges of dragons based in various populated areas of Westeros. Moreover, the plots of the random walks and heatmaps have value in and of themselves beyond just validating our previously computed data; they provide clear visuals that simulate the movement of a fictional creature that we cannot directly observe in any other way.

However, our models do suffer from some nontrivial weakness. We only consider the climate and conditions of Westeros during the winter. Although the winter climate plays a big role in the plot, it would be worthwhile to also consider the summer climate of Westeros as conditions may be significantly different for dragons. Similarly, our computations are all based on a single dragon. Varying weight, length, wingspan, and other quantities across dragons could have produced significantly different results. Moreover, we lack real data for predator-prey/food web modeling. Hence, we could only produce qualitative results in those sections.

With more time, and if we were to continue this work in the future, we could use the Lotka-Volterra competition models to investigate the effect of a super-predator such as the dragon on less dominant predators. We could also examine the extent of intraspecific competition between the dragons, as well as trends in dragon population growth over long periods of time. Addressing some of the above weaknesses such as by modelling various dragons throughout their growth periods would also be of priority.

Bibliography

[1] Afolayan, R P B Deland, M Rutley, David Bottema, Cynthia L Ewers, A Ponzoni, Raul Pitchford, Wayne. (2019). Prediction of carcass meat, fat and bone yield across diverse cattle genotypes using live-animal measurements.

[2] Benhamou, Simon, and Pierre Bovet. "How Animals Use Their Environment: a New Look at Kinesis." Animal Behaviour, vol. 38, no. 3, 1989, pp. 375–383., doi:10.1016/s0003-3472(89)80030-2.

[3] Bovet, Pierre, and Simon Benhamou. "Spatial Analysis of Animals' Movements Using a Correlated Random Walk Model." Journal of Theoretical Biology, vol. 131, no. 4, 1988, pp. 419–433., doi:10.1016/s0022-5193(88)80038-9.

[4] Carbone, Chris, et al. "The Costs of Carnivory." PLoS Biology, vol. 5, no. 2, 2007, doi:10.1371/journal.pbio.0050022.

[5] De, Dilip Kumar, et al. "Cooking with Minimum Energy and Protection of Environments and Health." IERI Procedia, vol. 9, 2014, pp. 148–155., doi:10.1016/j.ieri.2014.09.055.

[6] "Dragons." Game of Thrones Wiki, gameofthrones.fandom.com/wiki/Dragons.

[7] Goldstein, David L. "Estimates of Daily Energy Expenditure in Birds: The Time-Energy Budget as an Integrator of Laboratory and Field Studies." American Zoologist, vol. 28, no. 3, 1988, pp. 829–844., doi:10.1093/icb/28.3.829.

[8] "How Fast Do Dragons Fly? Speed of Dragonflight?" A Forum of Ice and Fire, 11 Mar. 2016, asoiaf.westeros.org/index.php?

[9] "Interactive Game of Thrones Map." Interactive Game of Thrones Map with Spoilers Control, quartermaester.info/.

[10] Kaita, Constance. "How Do Dragons Breathe Fire?" Why-Sci, 2013, why-sci.com/dragons/.

[11] Kohl, Kevin D., et al. "Gut Microbial Ecology of Lizards: Insights into Diversity in the Wild, Effects of Captivity, Variation across Gut Regions and Transmission." Molecular Ecology, vol. 26, no. 4, 2016, pp. 1175–1189., doi:10.1111/mec.13921.

[12] Pastor, John. Mathematical Ecology of Populations and Ecosystems. Wiley-Blackwell Pub., 2008.

[13] Pennycuick, Colin J. Modelling the Flying Bird. Elsevier, 2008.

[14] Price, Rob. "Here's the True Size of the Seven Kingdoms of 'Game of Thrones'." The Daily Dot, 3 Aug. 2017, www.dailydot.com/parsec/game-of-thrones-westeros-mapped/.

[15] Pulliam-Moore, Charles. "What Would It Take to Kill Daenerys' Dragons?" io9, io9.Gizmodo.com, 13 July 2017, io9.gizmodo.com/what-would-it-take-to-kill-daenerys-dragons-1796798663.

[16] Seaman, D. Erran, and Roger A. Powell. "An Evaluation of the Accuracy of Kernel Density Estimators for Home Range Analysis." Ecology, vol. 77, no. 7, 1996, pp. 2075–2085., doi:10.2307/2265701.

[17] Stevens, M. Henry H. A Primer of Ecology with R. Springer, 2010.

[18] Tarly, Samwell. "The Climate of the World of Game of Thrones." Philosophical Transactions of the Royal Society of King's Landing, vol. 1, no. 1.

[19] Walter, D., et al. "What Is the Proper Method to Delineate Home Range of an Animal Using Today's Advanced GPS Telemetry Systems: The Initial Step." Modern Telemetry, 2011, doi:10.5772/24660.

[20] "Westeros with Borders." A Wiki of Ice and Fire.

[21] Wilson, Alex. "Efficient Cooking." Green Building Advisor, The Taunton Press, 8 Aug. 2018, www.greenbuildingadvisor.com/article/efficient-cooking.

.1 Python Code for Random Walk Simulation

```
# performs one random walk
def gen_randwalk(heat, num_pts = 20000, P = .1, x_restr=(0,map_x),
                 origin=(450,240), k=.2):

    x0, y0 = origin[0], origin[1]
    # initial positions random
    x_lst = [np.random.uniform(x0 - 35, x0 + 35)]
    y_lst = [np.random.uniform(y0 - 35, y0 + 35)]
    theta_lst = [np.random.uniform() * 2 * np.pi] # rand init theta
    Pb = P # base step length

    row, col = int(round(x_lst[0])), int(round(y_lst[0]))
    P_lst = [Pb]
    heat_lst=[heat[row,col]]
    coord_lst = [(row,col)]
    grad_angle_lst = []
    i = 0
    while(len(x_lst) < num_pts):
        alpha = np.random.normal(scale = .1)
        theta = theta_lst[-1] + alpha
        row, col = int(round(x_lst[i])), int(round(y_lst[i]))
        grad_angle = np.arccos( grad[0][row][col]/
                     np.linalg.norm([grad[0][row][col], grad[1][row][col]]) )
        if grad[1][row][col] < 0:
            grad_angle = -grad_angle

        if i != 0:
            tau = grad_angle - theta_lst[i-1]
        else:
            tau = np.pi/2 # does not scale P
        P = Pb /(1 - k*np.cos(tau))**2
        x_step = P*np.cos(theta)
        y_step = P*np.sin(theta)
        new_x = x_lst[-1] + x_step
        new_y = y_lst[-1] + y_step

        # do not allow dragon to fly off map
        new_x = min(max(new_x, x_restr[0]), x_restr[1] - 1)
        new_y = min(max(new_y, 0), map_y-1)

        # keep needed data
        x_lst.append(new_x)
```

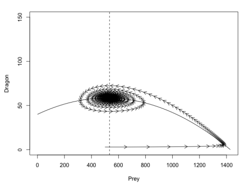
```
        y_lst.append(new_y)
        theta_lst.append(theta)
        P_lst.append(P)
        grad_angle_lst.append(grad_angle)
        heat_lst.append(heat[row,col])
        coord_lst.append((row,col))
        i += 1

    ret = {"x_lst" : x_lst, "y_lst" : y_lst,
            "theta_lst" : theta_lst, "P_lst": P_lst, "grad_angle_lst":grad_angle_lst,
             "heat_lst": heat_lst, "coord_lst": coord_lst}
    return ret

# Estimate home-range starting at Winterfell and King's Landing
# Uses gaussian_kde to do so
def est_home_range(num_walks=100):
    flights = [{"ret":[]} for _ in range(2)]

    for _ in range(num_walks):
        flights[0]["ret"].append(gen_randwalk(NUM_PTS, heat,
                 P = .1, origin=(450, 240), k=.2)) # winterfell
        flights[1]["ret"].append(gen_randwalk(NUM_PTS, heat,
                 P = .1, origin=(1050, 365), k=.2)) # king's landing
    print("finished walks")

    for i in range(2):
        x_lsts = np.array([ret["x_lst"] for ret in flights[i]["ret"]])
        y_lsts = np.array([ret["y_lst"] for ret in flights[i]["ret"]])
        f_x_lsts = x_lsts.flatten()
        f_y_lsts = y_lsts.flatten()
        X, Y = np.mgrid[0:map_x:1, 0:map_y:1]
        pos = np.vstack([X.ravel(), Y.ravel()])
        vals = np.vstack([f_x_lsts, f_y_lsts])

        kernel = stats.gaussian_kde(vals[:,0::1500])
        Z = np.reshape(kernel(pos).T, X.shape)
        cmaps = colors.ListedColormap([plt.cm.Blues(.5),(plt.cm.Greens(1))])
        implot = plt.imshow(in_map3, cmap=cmaps)
        plt.imshow(Z, cmap=plt.cm.gist_earth_r, alpha=.8)
        if i == 0:
            plt.plot(240, 450, marker="^", markersize=12, markerfacecolor="green", alpha
                markeredgewidth=1.3, markeredgecolor="k") # winterfell
        else:
            plt.plot(365, 1050, marker="^", markersize=12, markerfacecolor="green", alph
                markeredgewidth=1.3, markeredgecolor="k") # king's landing
```

```
        plt.axis("off")
        plt.show()
    return

# Plots random walks on Westeros
def draw_rand_walk(num_pts=20000, num_walks = 6):
    cmap = plt.cm.plasma
    cmap2 = colors.ListedColormap([plt.cm.Blues(.5),(plt.cm.Greens(1))])
    norm = colors.BoundaryNorm([0, .5, 1], cmap2.N)
    implot = plt.imshow(in_map3, cmap=cmap2, norm=norm)

    for i in range(num_walks):
        ret = gen_randwalk(heat, num_pts=num_pts)
        x_lst = ret["x_lst"]
        y_lst = ret["y_lst"]
        # change coords
        xs = y_lst
        ys = np.array(x_lst)

        color = cmap(i * 1/num_walks)
        plt.scatter(xs, ys, s=1, c=[color], alpha=.12)
        plt.scatter(xs[0], ys[0], c='blue', # plot startpoint blue
                s=20, marker="D", alpha=.5)
        plt.scatter(xs[-1], ys[-1], c='red', # plot endpoint red
                s=30, marker="s", alpha=.5)
    plt.plot(240, 450, marker="*", markersize=24, markerfacecolor="green",
             markeredgewidth=2, markeredgecolor="k", alpha=.95)
    plt.axis("off")
    plt.show()
    return

def compute_energies(rw_dict):
    # rw_dict is the return value of gen_randwalk
    a, b, _, _, _ = linregress(np.arange(-24,27,2), watt_t)
    heat_lst = rw_dict["heat_lst"]
    P_density = [a*T + b for T in heat_lst] # power density
    ts = np.arange(0,10000,.5)
    energies = cumtrapz(P_density, ts)

    ret = {"energies":energies, "P_density": P_density}
    return ret

# plots frequency heatmap on top of map of Westeros
def interlay_freqs(in_map, freqs):
    # in_map
```

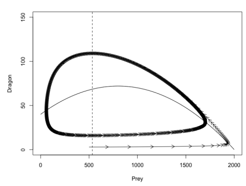
```
    plt.imshow(in_map, cmap="Greens")
    freqs = np.power(freqs, 1/4) # concave mapping
    ip = plt.imshow(freqs, cmap="plasma", alpha=.65)
    plt.axis("off")
    plt.show()
    return

def get_freq_map(in_map, itrs = 350, k=.2):
    start_time = time.time()
    freqs = np.zeros(in_map.shape)
    for i in range(itrs):
        ret = gen_dk_lst3(NUM_PTS, heat)
        if i % 100 == 0:
            print("walk", i, "of", itrs, "complete")
        coord_lst = ret["coord_lst"]
        unique, counts = np.unique(coord_lst, return_counts = True, axis = 0)
        for i in range(len(unique)):
            freqs[unique[i][0], unique[i][1]] += counts[i]

    freqs = freqs/ (itrs * num_pts)
    # smoothing
    for i in range(0,freqs.shape[0] - 0):
        for j in range(0,freqs.shape[1] - 0):
            freqs[i,j] = np.mean(freqs[max(i-5, 0):min(i+6, freqs.shape[0]),
                max(j-4,0):min(j+5,freqs.shape[1])])

    print("--- %s minutes ---" % ((time.time() - start_time)/60))
    # plots the frequencies
    interlay_freqs(in_map, freqs)
```

.2 Sensitivity of Dynamical Systems

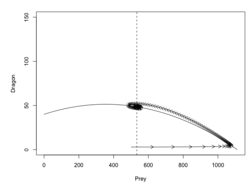
[H]



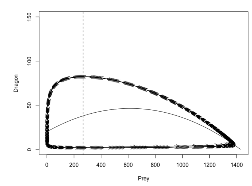
(a) r=0.5; b=0.07; m=0.2; w=5; d=400; alpha=0.0007



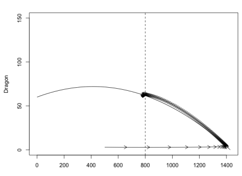
(b) r=0.5; b=0.07; m=0.2; w=5; d=400; alpha=0.0005



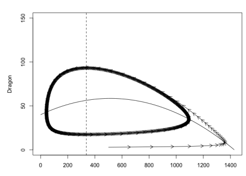
(c) r=0.5; b=0.07; m=0.2; w=5; d=400; alpha=0.0009



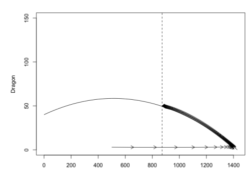
(d) r=0.5; b=0.07; m=0.2; w=5; d=200; alpha=0.0007



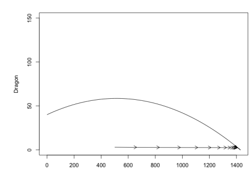
(e) r=0.5; b=0.07; m=0.2; w=5; d=600; alpha=0.0007



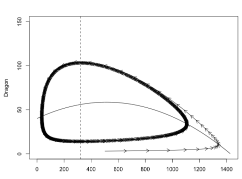
(f) r=0.5; b=0.07; m=0.16; w=5; d=400; alpha=0.0007



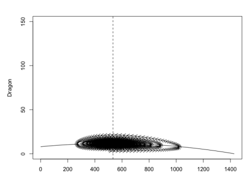
(g) r=0.5; b=0.07; m=0.24; w=5; d=400; alpha=0.0007



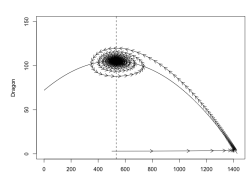
(h) r=0.5; b=0.05; m=0.2; w=5; d=400; alpha=0.0007



(i) r=0.5; b=0.09; m=0.2; w=5; d=400; alpha=0.0007



(j) r=0.1; b=0.07; m=0.2; w=5; d=400; alpha=0.0007



(k) r=0.9; b=0.07; m=0.2; w=5; d=400; alpha=0.0007

Figure 12