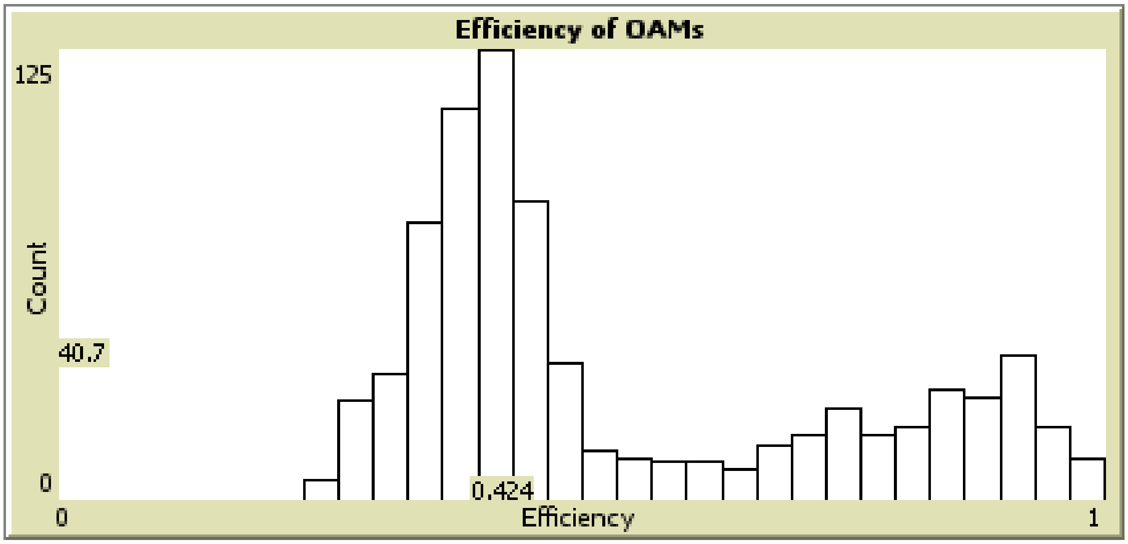
HIGH-LEVEL DESIGN
and
USER DOCUMENTATION
For MppLab
(MAXIMUM POWER PRINCIPLE LABORATORY)



Author: Garvin H Boyle
Date: 27 March 2017
orrery-software.webs.com

1 References 1
2 Background 1
  2.1 About the References 1
  2.2 Overview 1
3 Purposes 4
4 Discussion 5
    4.1.1 Legend of Acronyms and Variable Names: 6
    4.1.2 The Role of Atwood’s Machine 8
    4.1.3 Operational Concepts 9
    4.1.4 User Experience 9
    4.1.5 Things For Users to Try 10
  4.2 Technical Perspective 11
    4.2.1 Agent Based Models 11
    4.2.2 Verification, Validation, Simulation and Demonstration; 12
    4.2.3 Development Platform 13
    4.2.4 Maximum Power Emergence 13
    4.2.5 Initialization – The “Setup” Button 14
    4.2.6 Operations – The “One Tick” and “Go” Buttons 14
      4.2.6.1 Sub-Routine Do-Pre-Tick 15
      4.2.6.2 Sub-Routine Do-Move 15
      4.2.6.3 Sub-Routine Do-Feed 15
      4.2.6.4 Sub-Routine Do-Reproduce 16
      4.2.6.5 Sub-Routine Do-Die 18
      4.2.6.6 Sub-Routine Do-Post-Tick 18
    4.2.7 The Mathematics of MppLab 18
      4.2.7.1 General Equations for OAM operation 19
      4.2.7.2 The Mathematics of Energy Transfer 20
      4.2.7.3 Aggregates 22

5 Annex A – Descriptions of User Interface Panels 23
  5.1 Panel 01 – The Model 23
    5.1.1 Panel 01a – Main Controls and Monitors 23
    5.1.2 Panel 01b – Sources, Sinks, Flows 25
  5.2 Panel 02 – Statistics Re OAMs, Trophic Levels 26
  5.3 Panel 03 – Advanced System Parameters 27
  5.4 Panel 04 – Debug and Data Collection Tools 29
    5.4.1 Panel 04a – Debug tools 29
    5.4.2 Panel 04b – CSV Data Collection Tools 30
  5.5 Panel 05 – Energy Histograms 30
  5.6 Panel 01 – Genome Histograms 31
  5.7 Panel 07 – Line Graphs – Life Functions 33
  5.8 Panel 08 – Line Graphs – Energy Flows 34
6 Annex B – Related Models 35
  6.1 PSoup 35
  6.2 ModEco 35
  6.3 ModEco and the PMM 36
  6.4 EiLab 36
  6.5 CmLab 36
  6.6 OamLab 37
  6.7 MppLab 37
  6.8 TpLab 37
7 Annex C – Sample Debug Output 38
  7.1 To Setup 38
  7.2 To do-pre-tick 39
  7.3 To do-move 39
  7.4 To do-feed 39
  7.5 To do-reproduce 40
  7.6 To do-die 40
  7.7 To do-post-tick 40

High-Level Design
And
USER DOCUMENTATION
For MppLab
(MAXIMUM POWER PRINCIPLE LABORATORY)

1 References

A. G.H. Boyle (2017) MppLab_V1.09.nlogo
B. G.H. Boyle (2015) 150101 NTF Atwood's Machine R4.docx
C. G.H. Boyle (2015) 150105 NTF AM Shape Study R1.docx
D. G.H. Boyle (2015) 150113 NTF Atwood's Machine Revisited R4.docx
E. G.H. Boyle (2015) 150418 NTF Three Shapes of AM Revisited R2.docx
F. G.H. Boyle (2015) 170330 NTF High-Level Design - OAMLab R5.docx
G. G.H. Boyle's Website: Orrery-software.webs.com
H. A.J. Lotka (1922) 'Contribution to the Energetics of Evolution', Proceedings of the National Academy of Sciences of the United States of America, Vol. 8, No. 6 (Jun. 15, 1922), pp. 147-151.
I. A.J. Lotka (1922) 'Natural Selection as a Physical Principle', Proceedings of the National Academy of Sciences of the United States of America, Vol. 8, No. 6 (Jun. 15, 1922), pp. 151-154.
J. H.T. Odum and R.C. Pinkerton (1955) 'Time's speed regulator: The optimum efficiency for maximum output in physical and biological systems ', Am. Sci., 43 pp. 331–343.
K. H.T. Odum (1971) 'Environment, Power and Society', John Wiley, New York, 336 pp.
L. W. Silvert (1982) 'The theory of power and efficiency in ecology', Ecol. Modelling, 15: 159-164.
M. C.A.S Hall, J.A. Standord and R.F. Hauer (1992) 'The distribution and abundance of organisms as a consequence of energy balances along multiple environmental gradients', OIKOS 65: 377-390.
N. C.A.S Hall, Ed. (1995) 'Maximum Power: The Ideas and Applications of H.T.Odum', University Press of Colorado, 393 pp.
O. C.A.S Hall ' The continuing importance of maximum power', Ecological Modelling 178 (2004) 107-113.
P. T.T. Cai, T.W. Olsen and D.E. Campbell 'Maximum (em)power: a foundational principle linking man and nature', Ecological Modelling 178 (2004) 115-119.
Q. T.T. Cai, C.L. Montague and J.S. Davis 'The maximum power principle: An empirical investigation', Ecological Modelling 190 (2006) 317-335.
R. R. Constanza (2007) - He has written a brief but thought-provoking article on the MPP at http://www.eoearth.org/view/article/154526/ .
S. E. Morbius (2104) - He has pulled together a variety of other references to the MPP and shared them at this site. http://www.reddit.com/r/dredmorbius/comments/2hz2lk/darwinlotka_energy_law/

2 Background

2.1 About the References

This design document describes the Ref A agent-based computer model written in NetLogo back in early 2015 which, unfortunately, has gone undocumented until now.

The diary entries at Refs B through E are highly recommended read-in material for those who have a serious interest in this model. The document at Ref F is about the sister model to this one, **OamLab**, and it is recommended you study that model in parallel, or prior to studying this one. While exploring the nature of Atwood's Machine (AM, or the AM) and its relationship to the "Maximum Power Principle" (MPP), these were some of my early explorations and the foundation of much of what is contained herein. They can be downloaded from the Ref G internet site. Refs H through J are a few seminal papers from which the concept of the "Maximum Power Principle" has sprung. And Ref L is a paper in which Silvert took some issue with Odum's ideas. The other references show some ongoing interest in this proposed principle.

An AM is a closed mechanical system. At Ref D, in a thought experiment, I imagined the existence of a linkable chain or web of Halves of Open Atwood Machines (OAMs). An OAM is a totally imaginary concept which is not practical to build, but which captures the essence of an AM that can function as a non-isolated energy system. A Half OAM (or HOAM) contains a hold/release mechanism, a pulley, and a mass. Each such half-a-machine has the ability to link on the left, or on the right, to another such half-a-machine. Together the two linked HOAMs make a complete OAM.

I came to think that, if the MPP can do what H.T. Odum believed it could do, then any such evolving system will configure itself such that the OAMs will function at an intermediate efficiency but at maximum power. I decided to build a series of three models that would explore that thought. **OamLab** was the first; **MppLab** is the second; and **TpLab** is the third. These models have all been constructed as part of a personal search for an understanding of the dynamics of sustainable economic systems. For a description of all of my related models, see Annex B.

2.2 Overview

This overview is pulled from the 'Info' tab of the Ref A model.

> ***MppLab*** *- This is the second of a series of three planned models in a study of the maximum power principle (MPP) - the other two models being* ***OamLab*** *and* ***TpLab****. In* ***MppLab*** *energy arrives in a steady stream from the Sun and is captured by plants that act as the primary producers in a trophic web. The artificially distinct trophic chains studied in* ***OamLab*** *now merge to form a complex evolving web of organisms from herbivores to apex carnivores. By this means we have deeper insight into the basic implications of Lotka's argument (Lotka, 1922a) and the arguments of Odum and Pinkerton (1955). While Lotka's focus was on system-wide processes consisting of many serial energy transformations,*

Odum's focus was on individual energy transformations. While Lotka argued that system-wide processes maximize the degradation of energy, Odum argued that the individual energy transformations within them each minimized the degradation of energy. Can both opinions be true? How do we reconcile such clearly antithetic views? Is it possible to model the evolution of a complete trophic web, from primary producers to apex predators, in which (a) organisms diverge into separate species; (b) individual energy transformations evolve to become ever more powerful in their preservation of energy; and (c) system-wide trophic web evolves to become ever more powerful in its degradation of that flow of energy? With ***MppLab*** *I seek insight into some of these questions.*

MAXIMUM POWER PRINCIPLE (MPP) *- It is clear that, when isolated and left alone, all systems die, run down, erode, decay and/or dissipate their matter and energy. This process is closely associated with the phenomenon of entropy production, and the 2nd law of thermodynamics (sometimes referred to as the maximum entropy principle, or MEP) easily describes how this happens. Any such energetically isolated system automatically alters and reconfigures itself, moving through a series of ever more probable states along a trajectory of ever increasing entropy, until a most probable configuration or steady state of maximal entropy is achieved. This steady state of maximal entropy is usually characterized by an excess of sameness, a lack of structures, shapes or spatial variations, and by characteristic distributions of energies among the parts. Once having achieved such a configuration, the isolated system then remains in a that steady state of maximal entropy (a most probable state) forever after. However, when we look around ourselves, virtually everything we see is characterized by remarkable improbable variety, a plethora of structures and shapes, and turbulent distributions of energy. Clearly, when a system is not isolated and left alone, there is another dynamic able to overpower the 2nd law and undo its nasty work. Such is the nature of the proposed 4th law of thermodynamics. A.J. Lotka, H.T. Odum and R.C. Pinkerton called this the Maximum Power Principle (MPP). H.T. Odum (1924-2002) went on, over four decades, to argue that the MPP is the best candidate for the fourth law of thermodynamics, having explanatory value for such things as ecosystems, economies, and other self-organizing systems. However, in spite of the fact that there is plenty of anecdotal evidence in support of the concept, it remains little understood, and little studied it seems. The MPP says (my words) that any self-organizing system that is open with respect to a flow of energy will configure itself to store/consume/use energy at the maximum rate possible. This model is a "laboratory" in which I can study the nature of the MPP.*

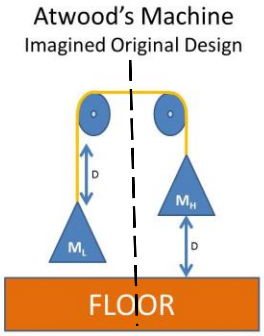
Atwood's Machine *- The approach in* ***OamLab*** *and* ***MppLab*** *is to study the Lotka/Odum MPP using a marvelous gizmo called Atwood's Machine (AM). This machine was invented in 1784 by the English mathematician George Atwood for the study of Newton's laws of motion. It has since become a common device in the design of a variety of lifts which use counterweights. Odum and Pinkerton used the AM as an example in their 1955 study of the MPP. In a variety of diary notes that I have written in an attempt to understand the MPP, I have developed a rather arcane notation of OAMs, HOAMs and OAM chains. It is highly recommended that interested people read those notes prior to studying this model. They are summarized below.*

SUSTAINABLE ECONOMICS *-* ***MppLab*** *is being written as part of a personal study of the dynamics of a sustainable economy. It has become clear to me that a modern economy is a sterling example of a system that has evolved to extract matter and energy from its* *environment, and to degrade and disperse those resources at a maximal rate. That observation has led to the study of the MPP. During the course of that study I have become aware that there are two principles that have been consistently proposed by scientists as candidates for the fourth law of thermodynamics. One is the MPP, as described above. The other is often referred to as the Maximum Entropy Production Principle (or MEPP). Note that this goes beyond the 2nd law - the MEP. The MEPP says (my words) that any self-organizing system that is open with respect to a flow of energy will configure itself to continuously* ***produce*** *entropy at the maximum rate possible. Now, I have intentionally reworded the two principles to draw attention to the similarity. I believe I have not mangled either idea too badly. I am personally convinced of two things:*

- *The MPP and the MEPP are two sides of a single little-understood phenomenon, and in both cases the proponents argue that these laws play a significant role in the organization of both ecosystems and economic systems. In these days of daily news of ecological and economic perils, it seems we might want to put some effort into understanding such a phenomenon. And so, my study of economics has taken an unexpected turn towards the MPP and the MEPP for a while.*
- *The very same phenomenon that is actively shaping our ecological and economic destinies is also active in many agent-based models (ABMs). I am NOT saying that I believe they are simulated in ABMs. I believe they are the organizing phenomena that cause unexpected emergent behaviour in ABMs. These principles are exhibited, or can be exhibited, in ABMs. What better place to study them, then, than in ABMs?*

Atwood's Machine Understood *- I will here provide a description of an AM that will help you understand the* ***OamLab*** *and* ***MppLab*** *applications. This differs only in mechanical design from other descriptions you will find, but not in function. An AM consists of two masses, one heavy and one light, coupled by a rope that is hung over a pair of non-leveraging simple pulleys. Most descriptions show a single pulley. I prefer to view it as a pair of pulleys, because I am going to split the AM into two halves, and each half has a pulley, a mass, an energy sink, a rope to couple masses together when needed, and a hold/release latch that can be used to pin the mass in place after it has been raised. In the AM, the heavy mass is primed with gravitational potential energy as it is raised a distance D off of the floor and latched in place. It is then coupled, using the ropes, to the lighter mass that rests on the floor, as a counterweight. When the heavy mass is released, it glides slowly to the floor as the lighter counterweight mass slowly rises. When the lighter mass has risen a distance of D, it is latched in place. At that point, the two masses may again be uncoupled.*

Figure 01 – Atwood's Machine



I apologize for the use of a variety of acronyms in the following explanation, and in the interface of the model, and in the code, but without them I find I get really tangled up in my words. So, here is a list of the acronyms that are important. For more detail, read the notes mentioned in the reference section:

*- **AM - Atwood's Machine in its original design** - a mechanical system that is closed with respect to energy, in the sense that most analyses of the original system start with the heavy mass already raised, and do not consider how the energy is first put into it. I use the AM as a metaphor for time-regulated transformation of energy. According to Odum, all energy-driven changes require time to unfold, and I am using the mechanics of a mechanical system to stand in as a time-based speed regulator for the biomechanical and biochemical changes that might happen as energy is passed from predator to prey in an ecosystem. I am using the well-understood mechanics of the AM to model the biochemistry of digestion.*
*- **OAM - Open Atwood's Machine** - an imaginary version of the AM that is open with respect to energy, allowing a flow of energy into the system on the right, and out again on the left, as it runs its course.*
*- **HOAM - Half of an OAM** - two of which remain when an OAM has its masses uncoupled and it is split in half. Each half consists of, as stated above, a pulley, a rope, a mass, an energy sink (the floor), and a hold-release latch that can be used to pin the mass up off of the floor.*
*- **RH-HOAM** - An HOAM that has been linked to the right side of another HOAM to form an OAM. The RH-HOAM must have a mass larger than the mass in its counterpart.*
*- **LH-HOAM** - An HOAM that has been linked to the left side of another HOAM to form an OAM. The LH-HOAM must have a mass lighter than the mass in its counterpart.*
*- **Chain of HOAMs** - a series of HOAMs formed into a distinct chain. They are linkable in pairs to form OAMs such that each side-by-side pair of HOAMS can form a well-formed OAM. Energy can flow into the head of the chain, and, as OAMs are formed, and as the masses are coupled and uncoupled, the energy flows through the chain to the tail. Each HOAM may be uncoupled, coupled as an RH-HOAM, or coupled as an LH-HOAM. You can think of such a chain as a simulation of energy flow through a trophic chain (along one chain in a web), of energy flow through the organic molecules that make up the cytoplasm in a cell, or, possibly, of capital flows through a supply chain of economic agents in an economy. [This is the crazy concept I am chasing in my study of sustainable economics.]*
*- **Web of HOAMs** - a collection of free-moving HOAMs that are freely linkable to other HOAMs with which they come in contact. They link in pairs to form OAMs such that each side-by-side pair of HOAMS forms a well-formed OAM. Energy flows from HOAM to HOAM as the masses are coupled, forming OAMs, and uncoupled again. Ultimately, the energy flows through the web. Each HOAM may exist in uncoupled form, coupled as an RH-HOAM, or coupled as an LH-HOAM. You can think of such a web as an analogy of energy flowing through an ecosystem, of energy flowing through the organic molecules that make up the cytoplasm in a cell, or, possibly, of capital flowing through independent economic agents in an economy. [Again, this is the crazy concept I am chasing in my study of sustainable economics.]*

*In **OamLab** I present preformed and persistent chains of OAMs competing for survival. In **MppLab** there are free-swimming HOAMs that form OAMs at will and so make virtual chains of OAMs of fleeting existence as the HOAMs compete for survival and devour one another in that competition.*

3 Purposes

Construction of this model is part of a larger project to understand sustainable economics. Models built during the course of this study are **PSoup**, **ModEco** (C++ platform), **ModEco and the PMM** (NetLogo platform), **EiLab** (C++ platform), **CmLab**, **OamLab**, **TpLab** and this model.

A brief description of each of these models can be found at Annex B.

Small groups of scientists have proposed many principles to be recognized as the fourth law of thermodynamics to address the self-organization of systems that are open with respect to energy flows, among which I have two favorites. The first is the Maximum Entropy Production Principle (MEPP), and the other is the Maximum Power Principle (MPP). Neither of these principles is widely studied or accepted.

I have become convinced that these two concepts are two sides of the same phenomenon – a phenomenon that plays a significant role in the dynamics of economic systems. Of the two, the MPP seems to be the more complex, and also to have anecdotal support that easily explains its function in terms of everyday experiences.

But, I have also become convinced that the very same phenomena that are actively shaping our ecological and economic destinies are also implicitly active in many agent-based models (ABMs). I am NOT saying that I believe these phenomena are simulated in ABMs. I believe they are among the organizing phenomena that cause unexpected emergent behaviour in ABMs. These principles are exhibited, or can be exhibited, in ABMs. What better place to study them, then, than in ABMs? This is why I have developed the range of models described in Annex B.

So, of the two concepts MEPP and MPP, it seems that the MPP may be most accessible to direct exploration via ABMs, at the moment, due to the insight provided via the AM. I am also working on the MEPP, but I think I have a better chance of resolving the implications of the MPP first – at least, to my satisfaction.

Therefore, the purposes for building this **MppLab** model are:

- To explicitly demonstrate the MPP in an agent-based model.
- To lay the ground work for connecting the MPP to the MEPP.

I usually only do high-level designs before I dive into coding, and then make specific design notes as needed to cover difficult technical details. This document is a summary of those various design notes, but is, in no way, meant to be a complete design document.

Therefore, the purposes of this document are:

- To summarize some of the more technical decisions made while constructing the model, and record them after the fact, and
- To provide some user documentation that supplements the contents of the Ref A model.

4 Discussion

The discussion proceeds in two parts. In the first part I address the concepts that inspired and shaped the development of the model. In the second part I address some technical aspects of the implementation. Then, there are three annexes. In Annex A I describe the user interface in some detail. In Annex B I briefly describe some of the related models developed in my study of the dynamics of sustainable economic systems. In Annex C I present some of the output from the "debug" feature of the model.

4.1.1 Legend of Acronyms and Variable Names:

When describing this model which is the result of a highly imaginary "thought experiment" one has each foot, metaphorically speaking, planted firmly each in three different paradigms. On one side, we have the paradigm of Newtonian mechanics with its masses, ropes and pulleys and its precise analytical equations and mathematical symbols. On the other side, we have the paradigm of ecosystems with predators and prey participating in trophic chains as they capture and consume energy, and as they capture and consume one another in the process. Relatively speaking, ecosystems are trackless jungles, when compared to mechanical systems. There are conceptual bridges across the gap between these two paradigms, and those bridges get a lot of traffic. In short, it gets very wordy describing those bridges each time I wish to cross one, so I use acronyms to reduce the wordiness. Then, a third paradigm had to be introduced – the paradigm of computer models with its own subroutines, variables, and arrays. The following definitions were developed within the referenced diary notes and within the **OamLab** application, and are a merger of the three different paradigms from which they spring: some are computer code variables, with long names; some are mathematical symbols, with short names; some are acronyms replacing long descriptive names, and all are doing double duty as bridges between the mechanical systems of Newton and the ecological systems of Odum. Throughout this document, those variables that are instances of computer code variables or model parameters are put into bold print, as they can be found somewhere in the Ref A model.

- **tick** – a discrete time step in the model, or the current time step measured from tick 0
- AM, or the AM – Atwood's Machine
- M_L – the lighter of the two masses in an AM or OAM; also presented as $\textbf{getf}_{pred}$
- M_H – the heavier of the two masses in an AM or OAM; also presented as $\textbf{getf}_{prey}$
- M_t – total of two masses within an AM or OAM; $M_t = M_H + M_L$
- M_d – difference of two masses within an AM or OAM; $M_d = M_H - M_L$
- D – Drop distance; = **g-drop-distance**; a parameter of the model
- g – acceleration due to gravity; = **g-acceleration**; a parameter of the model
- W_t – Total energy in the AM at start, all stored in M_H; $W_t = g \times D \times M_H$ = **max-potential**
- W_u – Still usable energy transferred from M_H to M_L; $W_u = g \times D \times M_L$
- W_e – Energy converted to kinetic energy then exhausted as waste heat; $W_e = W_t - W_u$
- η or E_u – H. T. Odum's measure of efficiency; $\eta = (W_u / W_t) = (M_L / M_H)$
- T_b – Base time to drop for an uncoupled mass
- T_d – Time to drop for an AM or OAM; $T_d > T_b$
- OAM – Open Atwood's Machine, of which energy flows into it, through it, and out again
- HOAM – Half OAM, which, when joined to another HOAM, makes an OAM
- RH-OAM – Right Hand HOAM, having the heavier mass of the two (M_H)
- LH-OAM – Left Handed HOAM, having the lighter mass of the two (M_L)
- MEP – Maximum Entropy Principle (2nd law of thermodynamics)
- MPP – A. J. Lotka's and H. T. Odum's Maximum Power Principle (proposed 4th law)
- MEPP – Maximum Entropy Production Principle (proposed 4th law)

An AM starts with stored total gravitational potential energy (W_t) equal to [the acceleration due to gravity (g)] times [the heavier mass (M_H)] times [the distance from the mass to the floor (D)]. We write that as $W_t = g \times D \times M_H$. After M_H is released, and as the AM runs to completion, this energy is transformed in two ways. The coupled mass assembly (M_t) accelerates, and picks up kinetic energy. At the same time, the lighter mass (M_L) is raised off the floor by a distance D, gaining gravitational potential energy (W_U) according to the equation $W_U = g \times D \times M_L$. When M_H strikes the floor, the kinetic energy of both masses is dissipated into a heat sink. The energy is either transferred from the RH-HOAM to the LH-HOAM as stored high-grade gravitational potential energy (W_u), or exhausted as low-grade waste heat (W_e). No entropy is produced as W_U is transferred and stored (due to the frictionless nature of the pulleys and ropes). Entropy is produced as the waste energy (W_e) is exhausted. Odum defines the efficiency (η or E_U) of the AM in transferring and storing high-grade energy as $\eta = W_U / W_T$, which simplifies to $\eta = M_L / M_H$. The behaviour of the AM varies across the spectrum of settings of M_H and M_L:

- When $M_H >> M_L$ then η is close to zero, M_H falls quickly, and the time-to-drop is short, close to T_b. This baseline value of time-to-drop (T_b) is given by the formula $T_b = \sqrt[2]{2D/g}$. Most of the initial endowment of gravitational potential energy is transformed to waste heat, and very little useful energy is stored in the LH-HOAM.
- When M_H is just slightly larger than M_L, then M_H descends very slowly, the time-to-drop is very long, η is very close to 1, and most of the endowment of energy is transformed into useful energy stored in the LH-HOAM.
- Curiously, the fastest transfer of the endowment of energy to still useful energy in the LH-HOAM (i.e. the maximum power) occurs when $M_H=2 \times M_L$. That is, maximum useful power happens when and $\eta = M_L / M_H = 0.5$.

The MPP says (my words again, Odum's vision) that any evolving self-organizing system based on such HOAMs will configure itself in such a way that the temporary OAMs, as they form, tend to operate at maximum power and at half efficiency.

Another vision of the MPP (Lotka's) puts it this way. In every self-organizing system there are processes that gather and process energy. Through a kind of Darwinian process of natural selection, those processes that garner and transfer the largest amount of useful energy (W_U) persist and increase their ability to garner energy, and prevent the less effective processes from persisting, and so the system evolves (reconfigures itself) such that it is garnering and processing energy at maximum power.

Research questions for MppLab – Here are my research questions:

- Under what conditions does such a self-organizing system of competing HOAMs converge to a system that has a common or average efficiency of $\eta <= ½$, as predicted by Odum in consideration of the operation of the AM?
- What system dynamics can be understood from the study of such evolving self-organizing systems?

I think that, if this MPP concept contains any real useful insight, then HOAMs exist as molecules, organisms, or economic agents that interact temporarily and transfer energy or capital. As the energy (or capital) flows through such systems from component to component (from HOAM to HOAM), some is degraded and exhausted, while the rest is passed on down the trophic web.

4.1.2 The Role of Atwood's Machine

Please note that there was one aspect of the original version of **OamLab** and of this model in particular that caused some confusion when I tried to explain them to others. In the original version, I used the "mass" from Atwood's Machine as a primary phenotypic attribute (gene) of my agents, but I did not in any way intend it to represent the real mass of the agents. It represented the genetic ability of organisms to consume and digest one another. To reduce the confusion I have modified the terminology within the code and in the user interface, but the logic of the code has not changed. The confusion arose from my "borrowing" of the mechanics of Atwood's Machine (the AM) to model the chemistry of digestion. In particular, in place of "mass" I have now recoded that attribute as a gene that I am calling the "genetic energy transfer factor", or **getf** for short. The logic for doing so is as follows:

- Odum showed that in the dynamics of the AM we find a great example of the trade-off between power and efficiency that is ubiquitous and a fundamental characteristic of ALL energy transfers. He also argued that all energy transfers take time, and that the MPP is the common time regulator (see Odum and Pinkerton, 1955).
- I do not have empirical data describing chemical digestion of one species by another. In fact, to simulate a real-world trophic web, I would need such data for all species trying to digest all other species. And, for each such potential transfer of energy, I would need both efficiency and speed of all such energy transfers.
- On the other hand, we have an exquisitely precise analytic description of the energy transfer that happens in the AM, providing both efficiency and power (rate of energy transfer), and implicit therein is the time required to complete the transfer (called the drop time).
- So, I reasoned, I could emulate the efficiency and rate of energy transfer of digestion using the precise analytic description of the AM. Since the mass of the two components of the AM are the critical factors, I have inserted a proxy for mass into all organisms, and I call that the "genetic energy transfer factor" (or **getf**). Whenever two organisms meet, I use their respective **getf** values to determine:
    - (a) who is predator and who is prey;
    - (b) how long it takes for predator to digest prey; and
    - (c) how much of the consumed energy of the prey is turned to waste heat, and how much is retained as "still-useful-energy" in the predator.
- This does not give me a simulation of any particular act of digestion, but it does give me a template for emulation of all possible acts of digestion (all possible efficiencies and powers) and thus allows me to investigate the nature of real-world trophic webs.

So, to be clear, the concept of mass that we all know and understand as playing a significant role in biology is not explicitly used anywhere in this model. But, the ability of one organism to digest another organism is determined by the interaction of the relative values of the **getf** factors, and I use the mathematics of energy transfers that derives from the AM and the masses therein to compute the effects of **getf** values when they interact. Mass in the AM has an analogous dynamic role as **getf** has in an organism, playing similar roles in determining the dynamics of energetic interactions.

How does the **getf** work? Well, when two HOAMs form an OAM, the side with the smaller mass receives energy in any energy transfer. Similarly, when two HOAMs meet and form an OAM in **MppLab**, then the HOAM (the organism) with the lesser **getf** value receives energy (eats) and the HOAM with the greater **getf** delivers energy (gets eaten). You can think of **getf** as an indicator of the "quality of the energy carried". Energy transfers would then always go from higher quality to lower quality. That idea sort of works here!

4.1.3 Operational Concepts

- **Agents –** Two types of agents – autotrophs and heterotrophs – are placed in the arena. Each organism is represented by a daisy (autotrophs) or an arrow (heterotrophs). The heterotrophs move forward somewhat randomly. Each agent of either kind is able to function as an HOAM when one is consumed by another. During consumption, an OAM is temporarily formed from the two HOAMs, and energy is transferred, and some is expelled as waste heat.
- **Autotrophs –** Each autotroph contains a deposit of energy available for a heterotroph to eat. During each tick of the model some autotrophs are eaten and disappear. But, the population of autotrophs is replenished at the end of each tick of the model, up to a maximum sustained level. So, the population of autotrophs provides a source of energy that the heterotrophs can draw upon.
- **Heterotrophs** – Each heterotroph is able to eat either autotrophs or heterotrophs, but only if the genetic energy transfer factor (**getf**) of both organisms are compatible. When two organisms meet (are in adjacent cells) they compare **getf**, and the organism with the lower value takes on the role of predator, while the other becomes the prey. But the **getf** of the prey must then be between two additional genetic values associated with the predator called the lower bound (lb) and upper bound (ub). If the **getf** of the prey falls within this range, the predator attempts a capture and begins to devour the prey. At this point, the two HOAMs freeze in place on the screen, and remain frozen until the prey is completely devoured, or released by the predator. Only one predator feeds on such captured prey.
- **getf –** The **getf** of all autotrophs is always set at 128. The **getf** of heterotrophs is initially set at 100, but is determined by a gene that can mutate and evolve. On initialization, heterotrophs can only eat autotrophs and are not able to eat other heterotrophs. But over time the **getf** values spread out from 100, and as soon as some **getf** values are less than others, the possibility of heterotroph cannibalism arises. The heterotroph with the smallest **getf** value becomes the apex predator, and all others with **getf** below the lb value of that apex predator become potential prey of it, or of some other organism. A trophic web is created.

4.1.4 User Experience

You have heard, I suppose, of one-player games. One of my tutored students calls programs like this a zero-player game. You set the parameters, you start it, and you watch it.

Agent colours – agents come in four colours:

- **Green –** autotrophs (plants) appear as black-eyed daisies with green leaves and petals, when they are not being eaten.
- **Red –** heterotrophs appear as red arrows, when they are not feeding or being fed upon.
- **Yellow –** heterotrophs appear as yellow arrows when they are feeding upon an autotroph or another heterotroph.
- **Purple –** autotrophs and heterotrophs both appear in purple when being fed upon by a heterotroph. This is true for both scenarios described below. There is always a nearby yellow heterotroph pointing at the purple agent. That is the agent that is feeding upon it.

Scenarios – there are two scenarios available: Herbivores and Omnivores.

- In the **Herbivores** scenario heterotrophs are purely herbivorous, and are not able to eat each other. If any heterotroph suffers a mutation that puts the **getf** of autotrophs (plants) out of range (i.e. no longer in the interval [lb, ub], then that heterotroph cannot feed, and it will die. This is analogous to a two-species ecosystem demonstrating trophic chains with only two links, and is not particularly interesting.
- In the **Omnivores** scenario heterotrophs may eat autotrophs that fall within the [lb, ub] interval, but also may eat other heterotrophs that fall in that range. A free-wheeling ecosystem emerges with up to six levels of heterotrophs feeding upon one another, that can be interpreted as a multi-species trophic web. I can measure trophic level of organisms in two ways: as [one plus the maximum trophic level of the prey eaten so far] or as [relative distance from the base **getf** of autotrophs]. In either case, trophic webs of more than five trophic levels emerge.

Quick Start – To perform a run of the model:

- Select a scenario;
- Select a seed number for the Pseudo-Random Number Generator (PRNG);
- Decide whether you want the lower bound (lb) and upper bound (ub) restrictions to be applied (default setting is recommended - both on);
- Click on the "Go" button.

There are many optional sliders in the user interface to the right, and many monitors and graphs, all of which are explained in detail in Annex A.

4.1.5 Things For Users to Try

- **SEEDS** - Note the behaviour of the system for the same scenario that differ only by different random seeds.
- **EXPERIMENTATION** - Explore some of the possible variations on the scenarios. Alter the scenarios and some setup parameters, and note how the different averages and quantities react. E.g. keep your eye on drop time and η.
- **A VIEW INSIDE** - [NOTE: Do not let this run for a long time. It generates very large debug

log files in a short time.] Turn on the debug feature. This is a four step process: (Step 1) Open the command centre by clicking on the teeny tiny upwards-pointing arrow at the bottom left of the screen. (Step 2) Then when the command centre appears at the bottom of the screen, click on the 'upsize' and 'clear' buttons at the top right of the command centre. (Step 3) Then use the debug button and the associated 'chooser' (scroll the view screen to the right to find these in Panel 04a) to turn on debug reporting for one or all of the steps. You are now in debug mode. (Step 4) Click on 'One Tick'. Then scroll through the debug info that has been written into the command centre log. It has also been written to a log file on the hard drive of your computer.

- **DATA EXPORT** – Instructions for this can be found in Panel 04b. Right click on any of the graphs, and choose export. Or, open the command centre as described above, and in the observer command line input box at the very bottom of the screen, enter the command EXPORT-WORLD "filename.CSV", where filename is a name of your choosing. I suggest, that e.g. if the model is at tick 35, the filename might be "world35.CSV". The quotes are needed. Then, use MS Excel, or a similar spreadsheet to load that file. Every piece of data in the model will be found there somewhere in the output.
- **ALL OF THE WAY INSIDE** - There are key routines in the code that provide insight into how the model operates. Go to the code tab and search for each of these routines:
    - **f-set-getf-derived-autotroph-characters**
    - **f-set-getf-derived-heterotroph-characters**
    - **f-stabilize-autotroph-population**
    - **f-heterotroph-expends-EPM**
    - **f-hunt-for-prey**
    - **f-eat-or-be-eaten**
    - **f-effect-per-tick-xrg-xfer-in-oam**
    - **f-reproduce-heterotroph**

4.2 Technical Perspective

4.2.1 Agent Based Models

MppLab is an agent-based model (ABM). There are two kinds of ecological agents that interact with each other – plants and animals – making a variety of exchanges of energy. The model is designed from the bottom up, as opposed to the top-down approach of most models. Each agent has attributes such as stored energy, age, and genome that control its behaviour. As agents move about and meet in the arena, they interact by rules governed by their genes and other attributes. Out of this emerges an ecosystem that functions at maximum power with average efficiency of 0.50.

The key attributes of the autotrophs (plants) are:

- `getf            ;; the getf (mass) in this HOAM`
- `max-potential   ;; maximum energy allowed; = getf*g*D`
- `cur-potential   ;; current holding of energy`

The heterotrophs are the active agents. They act as predators, and are often prey. In their role as predators, they keep track of various bits of data associated with the feeding event in progress. These are identified as OAM-related. This allows me to avoid using a third breed of agent for OAMs in the NetLogo code.

The key attributes of the heterotrophs (animals) are:

- `age                ;; age of this heterotroph`
- `getf               ;; the getf (mass) of this heterotroph`
- `lb-genetic-factor ;; gene to set lower bound on getf of prey.`
- `ub-genetic-factor ;; gene to set upper bound on getf of prey.`
- `RET                ;; Reproductive Energy Threshold`
- `EPM                ;; Energy Per Move`
- `max-potential      ;; maximum energy allowed; = getf*g*D`
- `cur-potential      ;; current holding of energy`
- `trophic-level-floated ;; trophic level of this heterotroph`

OAM-related temporary attributes of heterotrophs

- `Eu-in-oam          ;; Eu is H.T.Odum's efficiency (ML/MH)`
- `max-drop-time      ;; Drop time for this OAM (Td).`
- `cur-drop-time      ;; Time since drop started (T0).`
- `rem-drop-time      ;; Remaining drop time (Tr=Td-T0).`
- `drop-time-ratio    ;; Fraction of drop time remaining (Tr/Td).`

4.2.2 Verification, Validation, Simulation and Demonstration;

When a new computer model is constructed there are two ways to test it:

- **Verification** – Is the code free of error – does the software process data as intended?
- **Validation** – Is the model a valid simulation of reality?

A computer model can represent reality in two distinctly different ways:

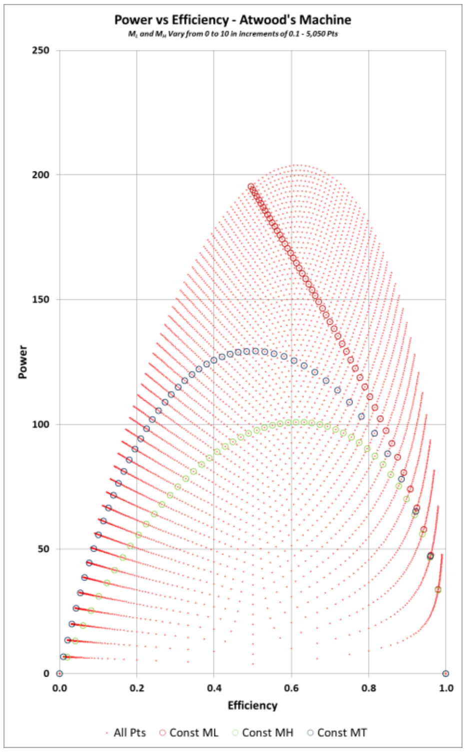
- **Simulation** – The model is attempting to reproduce, with as much accuracy as possible, a data stream that can be obtained empirically from measurements in the real world. When simulation is sufficiently accurate, then the model may be used to predict future events from past events. Weather models, for example, are of this type. Usually they are built on macro-level mathematical models involving otherwise intractable differential equations.
- **Demonstration** – The model is demonstrating dynamic behaviour that is implicit in its design, and can be studied as a dynamic system in its own right. While the understandings gained from such studies may be applicable to real-world systems, there is no intent to use the model to duplicate empirical data streams obtained from real-world systems. The “Game of Life”, or the study of “Finite State Machines” would fall into this category. Whereas few would argue that such models simulate, or even emulate, reality, they are clearly applicable, and the things learned in their study has been hugely informative. ABMs excel at this sort of modelling.

MppLab is NOT a simulation of anything. It is a demonstration of the dynamics that emerge out of the rules of random predation with digestion modeled using the mathematics of Atwood’s Machine. That is all that it is. The process of “validation” of a model involves comparison of model output with real-world data. MppLab is not meant to be validated by comparing numeric output from this model to real-world data. The goal is NOT to replicate some stream of real-world data accurately. The goal is to generate data that exemplifies the interactions of such rule-based systems, so that the dynamic relationships between variables can be better understood.

On the other hand, the process of "verification" of computer code is the process of finding and eliminating all of the bugs and programming errors. A computer program is considered verified if it has no logical bugs, and runs as intended. This application contains many hundreds of debug calls that are designed to find and eliminate all such bugs. As far as I know, the ecological engine of this application is 100% verified to be running as I intended. All design decisions are documented in detail in this document, and in the code itself, and all evidence of bugs of all sorts in the ecological engine has been addressed.

So, I prefer to say that MppLab is not a simulation of any ecosystem that needs to be verified. Rather, it is a demonstration of the dynamics of an abstracted ecosystem.

Figure 02 – Fingerprint Graph – The AM



4.2.3 Development Platform

I used NetLogo 5.0.5 for fast development. I got a working model up and running in seven days in early 2015. I have then returned to it many times to improve it as I learned how it worked.

4.2.4 Maximum Power Emergence

The goal was to see a true trophic web emerge out of a simple ecosystem. There are two optional scenarios. The "Herbivores" scenario is a practice scenario used to debug the concepts and set baseline expectations. The "Omnivores" scenario is meant to enable the emergence of a trophic web, under the effects of the MPP as exhibited in Atwood's Machine.

When working with the AM as a model there are at least three different "power vs efficiency" curves that can emerge. (See Figure 02, which is explained in detail in Ref E.) Two of these curves peak at intermediate levels of efficiency, as described by Odum, but the third one peaks at efficiency of $\eta = 0$. Each of these three outcomes depends on the lab procedure used to vary the weights in the AM when building the curve:

- If the sum of the masses M_t is held constant and M_H and M_L are varied in conformance with that constraint, we get the blue curve for which power peaks at $\eta = 0.5$ as described by Odum and Pinkerton (1955).
- If we hold M_H constant, and vary M_L from 0 to M_H, then we get the green curve for which

power peaks at $\eta = 0.618$. Curiously, this is $\eta = (1 - \tau)$ where $\tau = 1.618$ is the golden mean.

- But, if we hold M_L constant, and vary M_H from M_L to ∞, then we get the red curve, for which power peaks at $\eta = 0$.

Clearly other constraints on the way the weights are varied can produce other more exotic curves. My intuition after researching these curves was that those for which power peaks at some intermediate efficiency are characteristic of persistent phenomena, and those for which power peaks at either $\eta = 0$ or $\eta = 1$ would not result in persistent systems.

In **OamLab** the efficiency of a mature steady-state system is $\eta = 0.618$. But in this model, i.e. in **MppLab**, the efficiency of a mature steady-state system is $\eta = 0.5$. But, it was not always thus. In my first implementation, the system quickly evolved to a condition of $\eta = 0$ and the ecosystem collapsed. I argue that it evolves along the red curve towards a condition of maximum power and zero efficiency, at which point it becomes unsustainable. My intuition is consistent with the results of these two models. To correct that problem, I had to introduce two new genes into the genome of the heterotrophs (**lb-genetic-factor** and **ub-genetic-factor**) and two new switches in Panel 01 (**gb-h-ub-gene-active** and **gb-h-lb-gene-active**). These switches ensure that the new genes are used when prey are being selected. Any run that has these switches turned off is liable to collapse.

4.2.5 Initialization – The “Setup” Button

As for all NetLogo models, there are two setup routines. When you first load the file in which the model is stored, the “startup” routine is automatically called. This routine does three things. It makes sure that the debug mode is turned off. Then it resets all sliders to their default values. Then it runs the setup routine discussed immediately below. The sub-routine that resets all sliders does not reset the choosers. The choice of scenario is persistent through a save. All other choices are reset to default values on loading the model.

The “setup” routine is called by the “startup” routine, but also can be invoked at any time by the user by pressing the “Setup” button. It creates a population of autotrophs and heterotrophs having the attributes determined by the various sliders. Each of those sliders are described in detail in description of the user interface in Annex A. The autotrophs (daisies) are scattered randomly about the arena. The heterotrophs (animals) are arranged in a circle in the middle of the arena.

4.2.6 Operations – The “One Tick” and “Go” Buttons

The model steps forward in time in discrete time units called ticks. The “One Tick” button tells NetLogo to process a single tick of time. The “Go” button tells NetLogo to process ticks repeatedly one after another until the “Go” button is pressed again, at which time processing will stop.

There are many versions of the “Go” button placed in the various panels, for convenience of use. It is possible to have more than one running at once, but the results are faulty, and the program may seize. It is not recommended to have more than one “Go” button active at once.

Both of these buttons run a routine called “go”. This routine checks for a stop code, and then runs a number of sub-routines in order, as follows:

- do-pre-tick
- do-move
- do-feed
- do-reproduce
- do-die
- do-post-tick

Every one of these sub-routines is called, in order, during the execution of one tick.

4.2.6.1 Sub-Routine Do-Pre-Tick

The do-pre-tick sub-routine performs a number of administrative tasks that need to be taken care of before each tick begins. These include advancing the tick timer (telling NetLogo that a new tick has started); clearing the statistical registers that count the number of events, or sum the amount of energy spent, during a tick; topping up the number of autotrophs that have been killed; aligning any slider-derived parameters that may have changed; and, finally, causing all heterotrophs to age by one tick.

4.2.6.2 Sub-Routine Do-Move

Autotrophs (plants) do not move.

There are three optional actions for heterotrophs:

- Those that are free-roaming step forward and expend an amount of energy equal to EPM calculated using the parameter **g-h-EPM-factor**. Heterotrophs (animals) move using the “dodger” pattern from the **PSoup** application (See description in Annex B). That is, there is an 80% probability that they will step straight forward, but a 10% probability each that they will turn either to the right or left by 45 degrees before stepping forward.
- Those that are currently predators are feeding, and do not move. Nevertheless they expend energy as if moving, in support of ongoing life functions;
- Those that are captured as prey do not move, and do not expend any energy in trying to move. They are effectively dead.

Energy that is expended in movement is recorded in a register to be displayed in a graph at the end of the tick. Any agent that has insufficient energy to support ongoing life functions is marked as starved, and will die during the next execution of the “do-die” subroutine.

4.2.6.3 Sub-Routine Do-Feed

Autotrophs (plants) do not feed.

Heterotrophs (animals) will be in one of three conditions when this sub-routine starts to run:

Condition 1 – Hunting for prey – they may or may not be already satiated. If satiated, they do nothing. If hungry, they “hunt”. A list of potential prey is composed. In the “Herbivores” scenario the prey list contains only plants. In the “Omnivores” scenario the prey list contains both plants and animals. There are eight neighbor cells around the cell occupied by the heterotroph. If prey is available, the hunter bites it and begins to feed. The prey must fulfill these criteria:

- It is in one of the eight neighbouring cells;
- It is not already participating in an OAM (eating or being eaten);
- It’s **getf** value is greater than the **getf** value of the hunter;
- If **gb-h-lb-gene-active** is true, then it’s **getf** value must exceed the hunter’s **lb-genetic-factor** attribute;
- If **gb-h-ub-gene-active** is true, then it’s **getf** value must be less than the hunter’s **ub-genetic-factor** attribute;
- It is the most energetic of all agents fulfilling the above list.

An OAM is formed in which the prey takes the role of an RH-HOAM and the hunter/predator takes the role of LH-HOAM. The hunter is coloured yellow, and the prey is coloured purple.

Each organism is assigned a trophic level. The trophic level of a plant is 0 (zero). The trophic level of a heterotroph starts at the level of the mother, and is then adjusted with each prey that is eaten. The formula is [trophic-level] = 1 + [[sum of trophic levels of all prey eaten so far] / [number of prey eaten so far]]. In words, it has a trophic level one more than the average trophic level of its prey.

The final step in the hunt is execution of the sub-routine “**f-effect-per-tick-xrg-xfer-in-oam**”, which is described below in section 4.2.7.2.

If there are no prey that meet the conditions described above, then nothing happens.

If it is not hunting, then it is either eating, or being eaten, in which case the sub-routine “**f-eat-or-be-eaten**” is run.

Condition 2 – Eating prey – they are already connected to a prey organism in an operational OAM. Energy has already been transferred into them, and it must continue.

If the prey has any energy remaining to be eaten, execute the sub-routine “**f-effect-per-tick-xrg-xfer-in-oam**”, which is described below in section 4.2.7.2.

Condition 3 – Being eaten as prey – they are already connected to a predator organism in an operational OAM. Energy has already been transferred out of them, and it must continue. No action is required by the prey, as the predator takes the action in eating them.

4.2.6.4 Sub-Routine Do-Reproduce

In this model, autotrophs (plants) do not reproduce. When eaten they are simply replaced at the beginning of the next tick, if there is enough sunshine to construct them, and enough space in the arena to place them.

Heterotrophs reproduce via fission when they are healthy enough. Health is judged by the sufficiency of the energy they have collected (and not spent) as they gad about and eat prey. Each heterotroph has an attribute **RET** (an acronym for Reproductive Energy Threshold – a concept borrowed from my **PSoup** model). If **cur-potential** > **RET** then they can undergo fission. **RET** is calculated in the sub-routine "**f-set-getf-derived-heterotroph-characters**" at time of birth. The maximum amount of energy a heterotroph may hold is calculated as **max-potential** = [**getf** × **g-acceleration** × **g-drop-distance**]. Note: Compare that calculation to $E_t = g \times D \times M_H$. Then **RET** is calculated as [**g-h-RET-factor** × **max-potential**]. The default value for **g-h-RET-factor** is 0.95, meaning that a heterotroph must be at 95% of energy capacity to reproduce via fission.

A heterotroph that is being eaten cannot reproduce. A heterotroph that is feeding can reproduce, but releases the dead prey at that point.

Two daughters are born, and the mother is removed from the model (flagged for immediate death in the do-die sub-routine. Each daughter received three genes from the mother: **getf**, **lb-genetic-factor**, and **ub-genetic-factor**. These are the genes that control decisions about who is prey, and who is predator. As the genes are being transferred, there is a possibility for mutation, mediated by the sub-routine "**f-mutate-new-heterotroph**". Each daughter receives half of the energy previously held by the mother, but only if there is room to hold it. When the **getf** value is mutated, it may decrease in size, causing a decrease in the total energy that can be held. In very rare cases, the daughter may not be able to hold half of the energy of the mother, but only if the mutation removed more than half of the daughter's ability to hold energy.

f-mutate-new-heterotroph – this is the sub-routine in which the mutation of genes occurs. Three genes can mutate:

- The gene **getf** (an acronym for genetic energy transfer factor) can rise or fall by a factor between 0 and **g-mutation-factor**. If the default value of the slider **g-mutation-factor** is maintained, during every reproductive event the **getf** of each daughter will rise or fall by 10% or less. This slider controls the rate of change of the **getf** attributes of the heterotrophs. On mutation of **getf**, the attributes **RET** and **EPM** are immediately set to scaled values using the sub-routine **f-set-getf-derived-heterotroph-characters**.
- The gene **lb-genetic-factor** (an acronym for lower-bound-genetic-factor) can rise or fall by a factor between 0 and [2.5 times **g-mutation-factor**]. Note that this gene can change at a rate 2.5 times faster than the **getf** gene. If unconstrained, it would suffer a random walk, and the distribution of values would approach a Gaussian (normal or bell) distribution. However, the **g-h-lb-lower-bound** slider sets a lower bound on the **lb-genetic-factor** gene. I.e. there is a system-wide lower bound on the value of this gene, and this gene sets a lower bound on the interval in which prey may be found. If a mutated gene is less than the value set on this slider, the overage is reflected above the slider value. The result is a maximum-entropy-determined distribution (see histograms in panel 06). Evolutionary pressure causes the values of this gene to decline and cluster close to the limit.

- The gene **ub-genetic-factor** (an acronym for upper-bound-genetic-factor) can rise or fall by a factor between 0 and [2.5 times **g-mutation-factor**]. This is the reverse of the previous gene. Note that this gene can also change at a rate 2.5 times faster than the **getf** gene. Again, if unconstrained, the distribution of values would approach a Gaussian (normal or bell) distribution. However, the **g-h-ub-upper-bound** slider sets an upper bound on the **ub-genetic-factor** gene. I.e. there is a system-wide upper bound on the value of this gene, and this gene sets an upper bound on the interval in which prey may be found. If a mutated gene is more than the value set on this slider, the overage is reflected below the slider value. The result is a maximum-entropy-determined distribution (see histograms in panel 06). Evolutionary pressure causes the values of this gene to rise and cluster close to the limit.

For more information on the reasons for, and the functioning of, the **lb-genetic-factor** and **ub-genetic-factor** genes, see sections 4.2.4 and 5.1.

4.2.6.5 Sub-Routine Do-Die

Autotrophs die after being fed upon as prey of a heterotroph. When they die, any remaining energy is recorded in the energy sink. They are removed. They will be replaced at the beginning of the next tick.

Heterotrophs may die for several reasons:

- If they are older than the hidden parameter **g-h-DAT** (acronym for Death Age Threshold) they die of old age.
- If they have less energy than their **EPM** attribute (an acronym for Energy Per Move) they die of hunger. **EPM** is calculated as **EPM** = [**g-h-EPM-factor** × **getf** × **g-acceleration** × **g-drop-distance**] – i.e. a portion of the maximum energy that it can hold.
- Or, they can die after being fed upon as prey of another heterotroph.
- And, of course, the mother dies in the course of reproduction by fission, being replaced by two daughters.

The cause of death is recorded in the statistics for the tick, and shown on a graph in Panel 07.

4.2.6.6 Sub-Routine Do-Post-Tick

This sub-routine provides the hook for administrative activities. Here the validity of all agents is checked, usually only during debug activities. Also, here all aggregates that are used in graphs and monitors are calculated.

4.2.7 The Mathematics of MppLab

The very purpose and essence of this model is in the following mathematics. My assumption is that Odum’s speculation about the wide applicability of the MPP is true. My hypothesis is this: If I use the precise analytical mathematics associated with the AM as a generic template to describe the dynamics of digestion, then an ecosystem with an extremely complex trophic web should emerge. This requires that the energy transfer from prey to predator emulates the energy transfer from heavy mass to light mass in the AM in careful detail. Every bite must conform to the time, power and efficiency characteristics of the AM as closely as possible. That is, every bite of energy transferred must be of a precise size appropriate to the attributes of both prey and predator, and the duration of time (one tick), and the current plenitude of the prey. I define plenitude as [**cur-potential** / **max-potential**]. Also the duration of extended feeding events must conform to the discrete tick timeframe of the model. That is to say, there must be no partial ticks, and no partial bites. This will require care, and some tricky formulae.

4.2.7.1 General Equations for OAM operation

See the list of variable names in section 4.1.1. The fundamental equations of interest from Refs B through E are as follows.

Efficiency of 'useful energy' storage (E_u or η) – as defined by Odum and Pinkerton:

$\eta = \frac{W_U}{W_T} = \frac{M_L}{M_H} = \frac{\mathbf{getf}_{pred}}{\mathbf{getf}_{prey}}$	Equ 1

The base time to drop (T_b) for an HOAM that is uncoupled is:

$T_b = \left(\frac{2D}{g}\right)^{1/2}$	Equ 2

Note that this is independent of the masses.

The time to drop (T_d) for an OAM can be expressed in two ways:

$T_d = \left(\frac{2D}{A}\right)^{1/2} = \left(\frac{2DM_t}{gM_d}\right)^{1/2} = \left(\frac{2D[(getf_{prey}) + (getf_{pred})]}{g[(getf_{prey}) - (getf_{pred})]}\right)^{1/2}$	Equ 3
$T_d = \left(\frac{2D}{g}\right)^{1/2} \left(\frac{M_t}{M_d}\right)^{1/2} = T_b \left(\frac{1 + \eta}{1 - \eta}\right)^{1/2}$	Equ 4

The maximum energy (max-potential or W_t) in the prey HOAM prior to release of M_H is:

$W_t = gD(M_H) = gD(getf_{prey})$	Equ 5

The still useful energy transferred to the LH-OAM (W_u) is:

$W_U = DgM_L = Dg(getf_{pred}) = W_t(\eta)$	Equ 6

Waste energy, which becomes waste heat (W_e), is:

$W_e = DgM_d = gD[(getf_{prey}) - (getf_{pred})] = W_t(1 - \eta)$	Equ 7

Note that $W_t = W_U + W_e$. That is to say, the total energy endowment before a ‘run’ of the AM (a feeding event) is equal to the total ‘useful’ energy saved for later (as biomass) plus the total energy expended when the heavy mass smacks down on the floor (expelled as waste heat).

4.2.7.2 The Mathematics of Energy Transfer

f-effect-per-tick-xrg-xfer-in-oam – this is the sub-routine that performs the transfer of energy from prey to predator at a rate of one bite per tick. But, how big should each bite be to be consistent with the operation of the AM. The mathematics involved is a little complicated and the concepts behind this sub-routine are a little tricky, but it is key to my implementation of the AM as a template for other energy transformations. I will explain it first in terms of the AM (using masses), and also translate to digestion terminology (using the **getf** genes).

The prey is in the role of the right-hand side of a standard AM (see Figure 01). The maximum energy that can be stored in such an RH-HOAM is **max-potential** = $W_t = g \times D \times M_H$ where g is acceleration due to gravity (controlled by the slider parameter “**g-acceleration**”); D is the drop distance (controlled by the slider parameter “**g-drop-distance**”), and M_H is the mass of the heavy weight (represented by the proxy **getf** value of the prey). However, the prey may have less than **max-potential** units of energy due to movements, or due to failure to find food for itself, or due to prior feeding by the predator that is currently feeding on it. In the AM the instantaneous rate of transfer of energy is not constant, but varies with time as the mass assembly (the coupled set of masses) accelerates. So, to determine how much energy should be transferred during this tick I need to calculate the proper speed of the mass assembly for this level of energy in the prey, then calculate the portion of the drop time already used, then the amount of energy transferred from that moment for one tick. Clear? This way, the energy transferred mimics the rate of transfer of an appropriate AM for an appropriate duration of time.

Figure 03 – Sizing a Bite

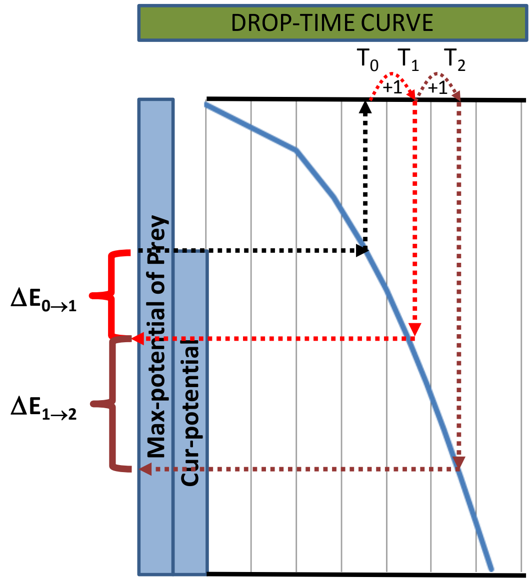


Figure 03 is a pictorial description of the process for calculating the size of a bite. The attribute **max-potential** of the prey determines the drop time (T_d) of the OAM and the acceleration curve. T_d is about 6.4 ticks. The **cur-potential** attribute determines how much time (T_0) of that duration has already effectively passed. I calculate T_0. Then, I add 1 tick to T_0 to get T_1, and I add another tick to that to get T_2. For each of those, I can calculate the level of energy that would be left, and the amount of energy that would be transferred as that tick progresses. I show those bite sizes as $\Delta E_{0\to1}$ and $\Delta E_{1\to2}$. The second bite is calculated as a look-ahead. If there is insufficient energy in the prey to make a complete next bite, the prey is released, the remaining energy in the prey is sent to a sink, and the OAM is dissolved.

Here are the relevant assignments:

- Set M_H as the **getf** value of the prey.
- Set M_L as the **getf** value of the predator.
- Set M_t as $M_H + M_L$.
- Set M_d as $M_H - M_L$.
- Let the maximum energy that the prey can hold be W_t = **max-potential** = $g \times D \times M_H$.
- Let the current amount of energy in the prey be W_C = **cur-potential**.
- Let the numerator be Num = $(W_t – W_C)$.
- Let the denominator be $Den = (M_d/M_t) \times (M_H g^2/2)$
- Let the Odum efficiency be $\eta = E_u = (M_L / M_H)$

$T_d = \sqrt{\frac{2D}{\left[\frac{M_H - M_L}{M_H + M_L}\right] g}} = \sqrt{\frac{2D}{\left[\frac{(getf_{prey}) - (getf_{pred})}{(getf_{prey}) + (getf_{pred})}\right] g}}$	Equ 8
$T_0 = \sqrt{\frac{Num}{Den}} = \sqrt{\frac{(gDM_H) - W_c}{\left[\frac{M_H - M_L}{M_H + M_L}\right] \times \frac{M_H g^2}{2}}} = \sqrt{\frac{\left(gD(getf_{prey})\right) - W_c}{\left[\frac{(getf_{prey}) - (getf_{pred})}{(getf_{prey}) + (getf_{pred})}\right] \times \frac{(getf_{prey}) g^2}{2}}}$	Equ 9
$T_1 = T_0 + 1$	Equ 10
$T_2 = T_0 + 2$	Equ 11

T_d is the total drop time for this OAM. T_0 is that portion of the drop time which has, effectively, already expired (due to the energy already transferred, or otherwise missing from the prey). T_1 and T_2 are the times one and two ticks later, starting from T_0.

Let the amount of energy transferred between time T_0 and T_1 be $\Delta E_{0\to1}$.
Let the amount of energy potentially transferred between time T_1 and T_2 be $\Delta E_{1\to2}$.

$\Delta E_{0\to1} = \left[\frac{M_H - M_L}{M_H + M_L}\right] g[T_1^2 - T_0^2] = \left[\frac{(getf_{prey}) - (getf_{pred})}{(getf_{prey}) + (getf_{pred})}\right] g[T_1^2 - T_0^2]$	Equ 12

$$\Delta E_{1\rightarrow 2} = \left[\frac{M_H - M_L}{M_H + M_L}\right] g[T_2^2 - T_1^2] = \left[\frac{(getf_{prey}) - (getf_{pred})}{(getf_{prey}) + (getf_{pred})}\right] g[T_2^2 - T_1^2]$$

Equ 13

I now have all of the information I need to make the relevant decisions and execute the transfer of energy associated with this one bite. Note that bites escalate in size as the ticks go by.

- Break the packet of energy $\Delta E_{0\rightarrow 1}$ into two parts using Odum's efficiency: waste heat and still-useful energy.
- Transfer the still-useful energy to the predator, and release the waste heat to a heat sink.
- If the predator cannot hold all of the transferred energy, release the excess to an energy sink. It is accounted differently in the statistics.
- If the next packet of energy $\Delta E_{1\rightarrow 2}$ is larger than what the prey actually contains, release the dead prey.

In a normal AM the kinetic energy is not expelled to sink until M_H hits the floor. However, because I want to put new potential into an agent and take it out whenever it is appropriate, I don't want to hold kinetic energy for a moment that may not come in a long while. So, I expel it to sink immediately.

Finally, if the prey is almost emptied, such that another bite would be a partial bite, then the prey is released. The real reason for this little piece of logic is to ensure that there is no noise or bias introduced into my energy transfer statistics due to a mismatch of bite size and energy available in the prey. I want every energy transfer, every bite in every tick, to perfectly represent a transfer consistent with the dynamics of the AM. So, partial bites are not allowed.

The above procedure needs to be calibrated for the ecosystem. For example, if the acceleration due to gravity (g) is set at its real-world value of 9.8 m/s^2 then the drop distance (D) must be extremely large to ensure that a feeding session lasts many ticks. In the real world, digestion of prey by a predator may last seconds (e.g. when an amoeboid eats a bacterium) or weeks (e.g. when a snake eats a goat). A mass can fall a very long way in several weeks. For this ecosystem, in which the **getf** of plant and animal are initially arbitrarily set at 128 and 100 respectively, I put the default values of g at 1 and D at 100.

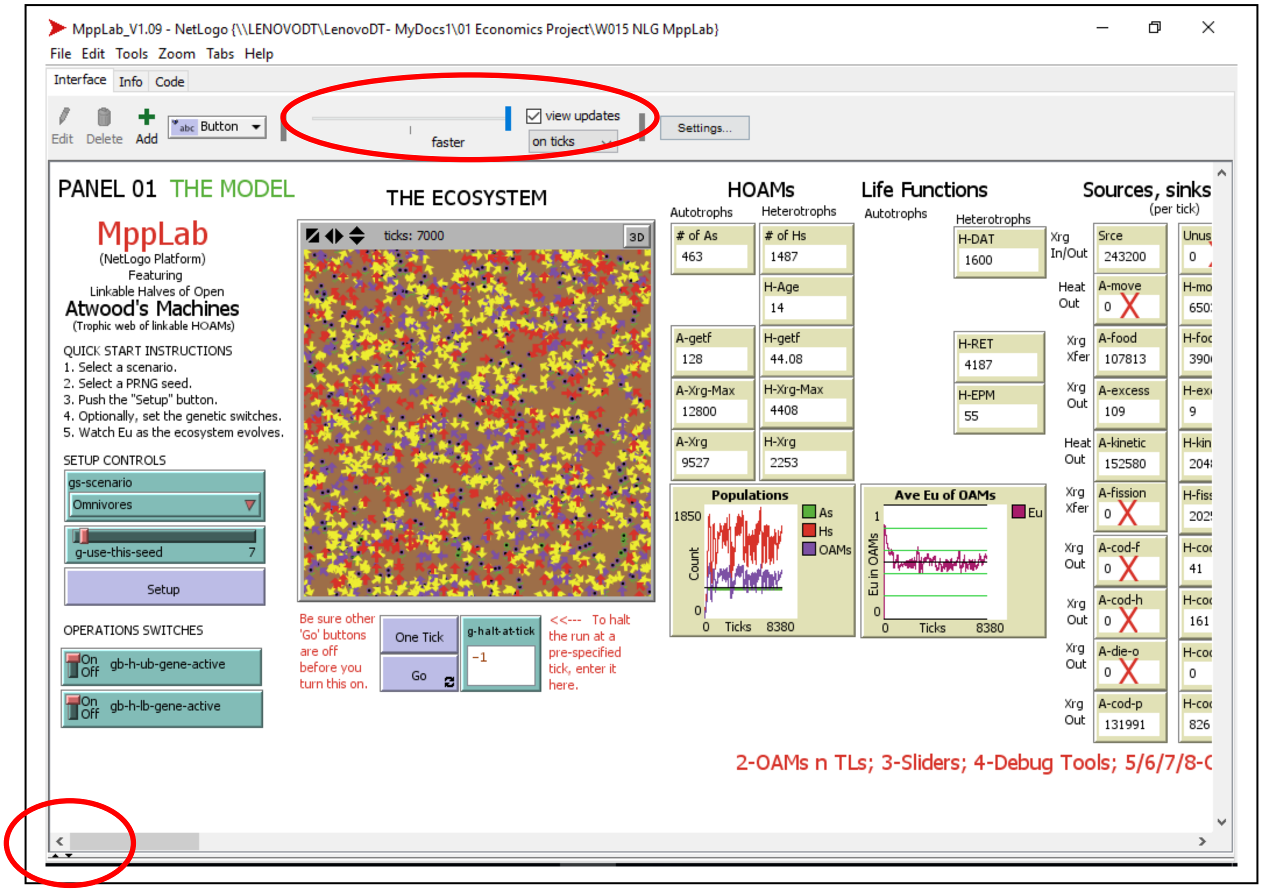
4.2.7.3 Aggregates

It is always difficult to decide how to get visibility into what is happening when there is, in fact, little or nothing visible happening. The best way is in real-time monitors and historical graphs, and NetLogo is pretty good for that. In the background I have placed a large number of aggregators that are updated with statistics throughout a tick. These are used to populate the many monitors in the user interface, and to update the many graphs.

5 Annex A – Descriptions of User Interface Panels

5.1 Panel 01 – The Model

5.1.1 Panel 01a – Main Controls and Monitors


MppLab_V1.09 - NetLogo {\\LENOVODT\LenovoDT- MyDocs1\01 Economics Project\W015 NLG MppLab}

PANEL 01 THE MODEL

MppLab
(NetLogo Platform)
Featuring
Linkable Halves of Open
Atwood's Machines
(Trophic web of linkable HOAMs)

QUICK START INSTRUCTIONS
1. Select a scenario.
2. Select a PRNG seed.
3. Push the "Setup" button.
4. Optionally, set the genetic switches.
5. Watch Eu as the ecosystem evolves.

SETUP CONTROLS
gs-scenario: Omnivores
g-use-this-seed: 7

ticks: 7000

Be sure other 'Go' buttons are off before you turn this on.

g-halt-at-tick: -1

<<--- To halt the run at a pre-specified tick, enter it here.

HOAMs Autotrophs		HOAMs Heterotrophs	
# of As	463	# of Hs	1487
		H-Age	14
A-getf	128	H-getf	44.08
A-Xrg-Max	12800	H-Xrg-Max	4408
A-Xrg	9527	H-Xrg	2253

Life Functions (Heterotrophs): H-DAT 1600; H-RET 4187; H-EPM 55

Sources, sinks (per tick): Srce 243200; A-move 0; A-food 107813; A-excess 109; A-kinetic 152580; A-fission 0; A-cod-f 0; A-cod-h 0; A-die-o 0; A-cod-p 131991

2-OAMs n TLs; 3-Sliders; 4-Debug Tools; 5/6/7/8-C

This is what you see when you first open the model. Things to note are the standard speed controls at the top (slider and tick box), the set of small carets at the bottom left that are used to open and close the NetLogo "Command Center". This is just the left-most part of Panel 01, and the rest is described below. The controls, monitors and displays in this panel are:

- **gs-scenario** – a chooser – used to select one of two scenarios.
- **g-use-this-seed** – a slider – used to select a seed for the Pseudo Random Number Generator (PRNG). If you use the same seed, you will be able to replicate a run. You get a different run for each different seed.
- **Setup** – a button – sets up the model for a new run. Choose the scenario and the seed before pushing this button. There are copies of this button throughout the user interface for convenient access.
- **gb-h-ub-gene-active** – a toggle – enables or disables the use of the ub-genetic-factor attribute (gene) by heterotrophs. When this is on, a predator will exclude from prey those

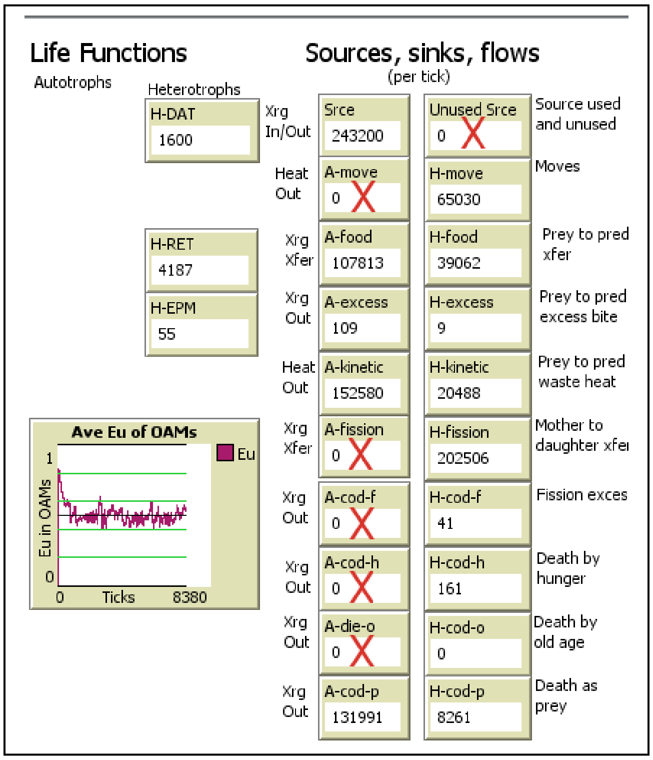
organisms whose **getf** value exceeds the predator's ub-genetic-factor gene value. I.e. if the **getf** attribute of the potential prey is above the predator's ub-genetic-factor value, then the predator will not attempt to eat the prey.

- **gb-h-lb-gene-active** – a toggle – enables or disables the use of the lb-genetic-factor attribute (gene) by heterotrophs. This is a counterpart to the previous gene switch. When this is on, a predator will exclude from prey those organisms whose **getf** value falls below the predator's lb-genetic-factor gene value. I.e. if the **getf** attribute of the potential prey is below the predator's lb-genetic-factor value, then the predator will not attempt to eat the prey.
- **The Ecosystem** – the arena – this shows the green and purple autotrophs, and the red, yellow and purple heterotrophs. See section 4.1.4 for a description of the colours and their interpretation.
- "**One Tick**" and "**Go**" buttons – buttons – See section 4.2.6 for a description of these buttons.
- **g-halt-at-tick** – an input field – enter a number here, and the model will automatically cease a run at this tick.
- **HOAMS** – There is a set of monitors in two columns and a graph beneath them:
    - **Autotrophs** – the left column of monitors is about autotrophs:
        - **# of As** – a monitor – the number of autotrophs currently alive.
        - **A-getf** – a monitor – the average value of the **getf** of autotrophs.
        - **A-Xrg-Max** – a monitor – the average value of maximum energy containable.
        - **A-Xrg** – a monitor – the average value of contained energy.
    - **Heterotrophs** – the right column of monitors is about heterotrophs:
        - **# of Hs** – a monitor – the number of heterotrophs currently alive.
        - **H-Age** – the average age of the heterotrophs.
        - **H-getf** – a monitor – the average value of the **getf** of heterotrophs.
        - **H-Xrg-Max** – a monitor – the average value of maximum energy containable.
        - **H-Xrg** – a monitor – the average value of contained energy.
    - **Populations** – a line graph – four lines showing the maximum number of autotrophs allowed (black), the actual number of autotrophs (green), the number of heterotrophs (red) and the number of OAMs formed (purple). An OAM is an feeding event in which a heterotroph is eating another HOAM (another organism, either plant or animal).
- **Life Functions** – There is a set of monitors in one column and a graph beneath it:
    - **Autotrophs** – the left column of monitors is about autotrophs – there are no entries.
    - **Heterotrophs** – the right column of monitors is about heterotrophs. These are the life function parameters brought over from PSoup and ModEco:
        - **H-DAT** – the average value of the Death Age Threshold – a vestigial monitor for a parameter that is held constant.
        - **H-RET** – a monitor – the average value of Reproductive Age Threshold – a number that varies from agent to agent, calculated based on **getf**.
        - **H-EPM** – a monitor – the average value of Energy Per Move – a key parameter that varies from agent to agent, calculated based on **getf**.
    - **Ave Eu of OAMs** – a line graph – this is a key output of the model. Odum predicted that ecosystems would function at a level of efficiency that is intermediate in value (see

section 4.1.2.) Since this model is based on the AM, and since the AM has peak power at $\eta = 0.5$, it is good to see that this graph often settles on 0.5 for steady-state operation. However, there is a lot of complexity here. While the average Eu settles at 0.5, the distribution of Eu values varies greatly. See the histogram "Efficiency of OAMs" in Panel 05.

5.1.2 Panel 01b – Sources, Sinks, Flows

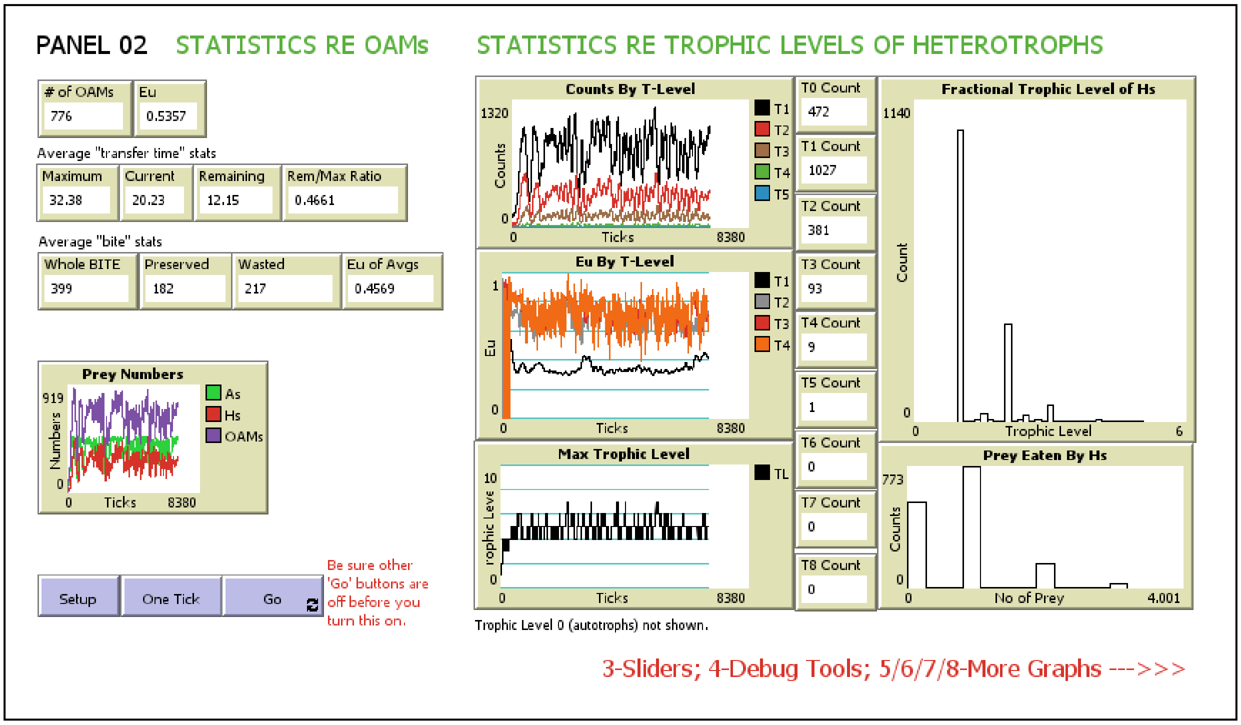
I could not fit all of Panel 01 into one page. This is the overflow. These monitors all track the flows and stores of energy in the ecosystem. There are two columns of monitors, one for the population of autotrophs, and one for the heterotrophs. The meaning is the same for both columns so I will only describe them once. Some monitors do not apply, have a large red X on them, and can be ignored. These are not averages, but system-wide aggregates (totals) collected per tick:



H-DAT 1600			
	Xrg In/Out	Srce 243200	Unused Srce 0 ✗	Source used and unused
	Heat Out	A-move 0 ✗	H-move 65030	Moves
H-RET 4187	Xrg Xfer	A-food 107813	H-food 39062	Prey to pred xfer
H-EPM 55	Xrg Out	A-excess 109	H-excess 9	Prey to pred excess bite
	Heat Out	A-kinetic 152580	H-kinetic 20488	Prey to pred waste heat
	Xrg Xfer	A-fission 0 ✗	H-fission 202506	Mother to daughter xfer
	Xrg Out	A-cod-f 0 ✗	H-cod-f 41	Fission exces
	Xrg Out	A-cod-h 0 ✗	H-cod-h 161	Death by hunger
	Xrg Out	A-die-o 0 ✗	H-cod-o 0	Death by old age
	Xrg Out	A-cod-p 131991	H-cod-p 8261	Death as prey

- **Srce** – energy flow into the ecosystem per tick through the autotrophs.
- **H-Move** – energy expended by heterotrophs on the move exhausting EPM units of energy each.
- **A/H-food** – energy transferred prey to predator as still useful energy.
- **A/H-excess** – energy that was unsuccessfully transferred and sent to sink as energy.
- **A/H-kinetic** – waste heat sent to sink, produced due to inefficiency in the process of digestion. Efficiency in an OAM – ($\mathbf{getf}_{pred}$)/($\mathbf{getf}_{prey}$).
- **H-fission** – energy transferred mothers to daughters.
- **H-cod-f** – energy transferred to sink due to inability of daughters to receive a full allotment from the mother due to downgraded **getf**.
- **H-cod-h** – energy transferred to sink due to death by hunger of agents having a small but insufficient remnant of energy.
- **H-cod-o** – energy transferred to sink due to death by old age.
- **H-cod-p** – energy transferred to sink due to death of prey – incomplete last bite is discarded.

5.2 Panel 02 – Statistics Re OAMs, Trophic Levels


PANEL 02 STATISTICS RE OAMs STATISTICS RE TROPHIC LEVELS OF HETEROTROPHS

# of OAMs 776
Eu 0.5357

Average "transfer time" stats
Maximum 32.38
Current 20.23
Remaining 12.15
Rem/Max Ratio 0.4661

Average "bite" stats
Whole BITE 399
Preserved 182
Wasted 217
Eu of Avgs 0.4569

T0 Count 472
T1 Count 1027
T2 Count 381
T3 Count 93
T4 Count 9
T5 Count 1
T6 Count 0
T7 Count 0
T8 Count 0

Be sure other 'Go' buttons are off before you turn this on.

Trophic Level 0 (autotrophs) not shown.

3-Sliders; 4-Debug Tools; 5/6/7/8-More Graphs --->>>

There are three groups of items in the panel. At the bottom left is a standard set of control buttons – Setup, One Tick, and Go. Read sections 4.2.5 and 4.2.6 for a description of their uses.

The group on the left addresses OAMs:

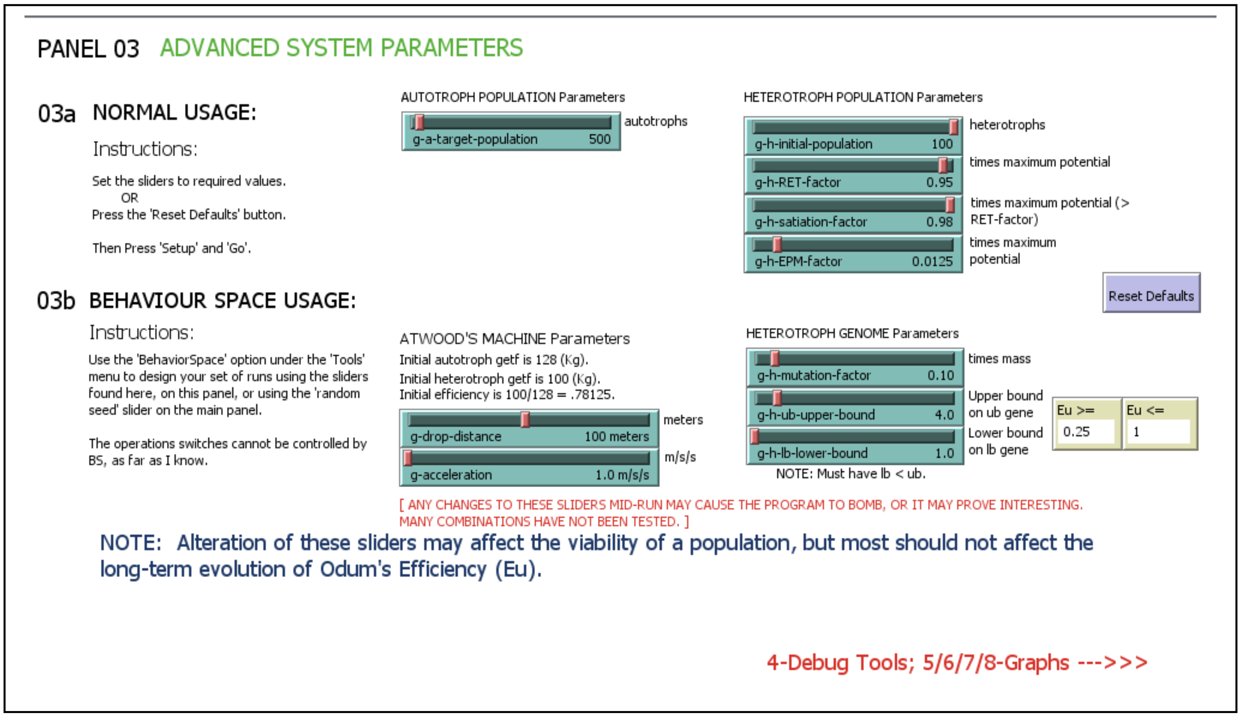
- **# of OAMs** – a monitor – the number of feeding events currently happening.
- **Eu** – a monitor – the average efficiency of all currently active OAMs; expected to remain close to 0.5 (see section 4.1.1). **This is a key monitor to watch.**
- Average “transfer time” stats – average statistics about “drop times” pulled from LH-HOAMS:
    - **Maximum** – a monitor – the maximum drop time.
    - **Current** – a monitor – the drop time already used.
    - **Remaining** – a monitor – the drop time not yet used.
    - **Rem/Max Ratio** – a monitor – average of this ratio, across all OAMs.
- Average “bite” stats – statistics about bite numbers, and effects:
    - **Whole BITE** – a monitor – Average size of one bite.
    - **Preserved** – a monitor – Average portion of bite transferred successfully.
    - **Wasted** – a monitor – Average portion of bite exhausted as waste heat.
    - **Eu of Avgs** – a monitor – (Average Preserved) / (Average Whole BITE). Compare this with Eu (above).
- **Prey numbers** – a line graph – with three historic lines: number of autotrophs that are currently being eaten (green), number of heterotrophs currently being eaten (red), and

number of OAMs (purple), being the sum of the other two.

The group on the right addresses trophic levels. I can track up to nine trophic levels, but may only display data for 4-5. Higher trophic levels are less common and tend to produce a lot of zeros. This data may be difficult to interpret, but is, nevertheless, rather interesting. See section 4.2.6.3 to see how trophic level of a heterotroph is calculated.

- **Counts By T-Level** – a line graph – Trophic level T0 is plants and is not plotted. Five trophic levels are tracked using the rounded values of the fractional trophic level stored in each heterotroph.
- **Eu By T-Level** – a line graph – Each functioning OAM has an associated efficiency, the average of which is plotted here. The value $\eta = 1$ is plotted in blue, and can be seen at the top of the graph. The value $\eta = 0.5$ would be in the middle of the graph, and there are only rare intrusions into that part of the graph. T1 (black) settles at a stable value around 0.35, and the others are volatile, hanging around at value 0.70. Very interesting. Causes uncertain!
- **Max Trophic Level** – a line graph – the largest trophic level across all heterotrophs per tick. It seems to ride at 4, with regular increases to 5, 6 or seven.
- **Counts of agents at each trophic level** – monitors – T0 through T8, nine monitors in all.
- **Fractional Trophic Level of Hs** – a histogram – The monitors display the numbers by rounded trophic level, but this histogram can present similar data for the fractional values.
- **Prey Eaten by Hs** – a histogram – a heterotroph has eaten an integral number of organisms. This shows how many have eaten none, one, two, three, or more.

5.3 Panel 03 – Advanced System Parameters


PANEL 03 ADVANCED SYSTEM PARAMETERS

03a NORMAL USAGE:

Instructions:

Set the sliders to required values.
OR
Press the 'Reset Defaults' button.

Then Press 'Setup' and 'Go'.

03b BEHAVIOUR SPACE USAGE:

Instructions:

Use the 'BehaviorSpace' option under the 'Tools' menu to design your set of runs using the sliders found here, on this panel, or using the 'random seed' slider on the main panel.

The operations switches cannot be controlled by BS, as far as I know.

AUTOTROPH POPULATION Parameters

g-a-target-population 500 autotrophs

HETEROTROPH POPULATION Parameters

g-h-initial-population 100 heterotrophs
g-h-RET-factor 0.95 times maximum potential
g-h-satiation-factor 0.98 times maximum potential (> RET-factor)
g-h-EPM-factor 0.0125 times maximum potential

ATWOOD'S MACHINE Parameters
Initial autotroph getf is 128 (Kg).
Initial heterotroph getf is 100 (Kg).
Initial efficiency is 100/128 = .78125.

g-drop-distance 100 meters meters
g-acceleration 1.0 m/s/s m/s/s

HETEROTROPH GENOME Parameters

g-h-mutation-factor 0.10 times mass
g-h-ub-upper-bound 4.0 Upper bound on ub gene
g-h-lb-lower-bound 1.0 Lower bound on lb gene
NOTE: Must have lb < ub.

Eu >= 0.25
Eu <= 1

[ANY CHANGES TO THESE SLIDERS MID-RUN MAY CAUSE THE PROGRAM TO BOMB, OR IT MAY PROVE INTERESTING. MANY COMBINATIONS HAVE NOT BEEN TESTED.]

NOTE: Alteration of these sliders may affect the viability of a population, but most should not affect the long-term evolution of Odum's Efficiency (Eu).

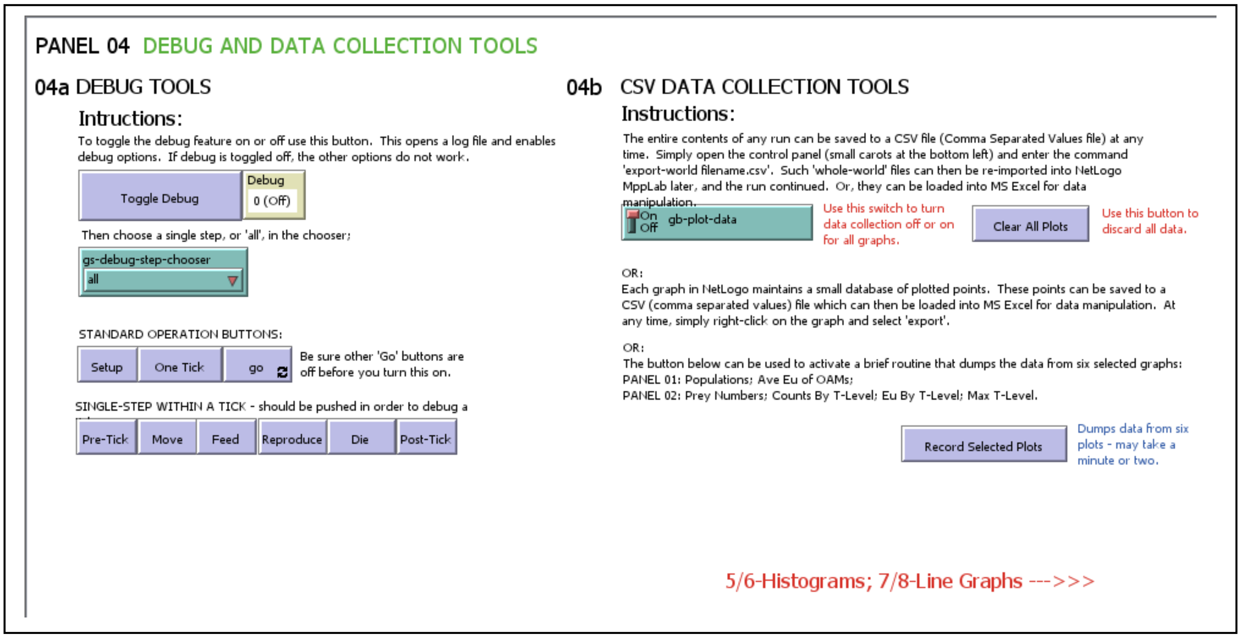
The controls in this panel fall into four groups, and a few odd additions. The "Reset Defaults" button resets all sliders in this panel to default values, and also change the random seed and the two switches in panel. The sliders are:

- Autotroph Population Parameters
  - **g-a-target-population** – a slider – this sets the maximum target population for autotrophs. At the beginning of each tick, if the population of autotrophs falls below this number, the number is immediately brought back to target, and the appropriate load of energy is placed into each.
- Heterotroph Population Parameters
  - **g-h-initial-population** – a slider – this sets the initial population for heterotrophs. It is only used during Setup.
  - **g-h-RET-factor** – a slider – this sets the **RET** attribute of each heterotroph when initialized. **RET** = **max-potential** × **g-h-RET-factor**. This factor must be less than the satiation-factor.
  - **g-h-EPM-factor** – a slider – this sets the **EPM** attribute of each heterotroph when initialized. **EPM** = max-potential × **g-h-EPM-factor**.
  - **g-h-satiation-factor** – a slider – this is used to determine the level of energy above which the heterotroph is no longer hungry and no longer hunts. This factor must be greater than **g-h-RET-factor**.
- Heterotroph Genome Parameters:
  - **g-h-mutation-factor** – a slider – this determines the maximum size of the delta in each mutating gene during reproduction. Three genes mutate: **getf**, lb-genetic-factor and ub-genetic-factor. The maximum delta for each is this factor times (1, 2.5 and 2.5) respectively for each gene type. The actual delta is some random portion of these maxima. To cause the rate of evolution to slow down, set this slider to a lower value, or vise versa.
  - **g-h-ub-upper-bound** – a slider – this is a system-wide upper bound on the gene size in each heterotroph. It is an upper bound on the "upper bound" gene.
  - **Eu >=** - a monitor – indicates the effect on the system-wide efficiency of the ub parameter.
  - **g-h-lb-upper-bound** – a slider – this is a system-wide upper bound on the gene size in each heterotroph. It is a lower bound on the "lower bound" gene.
  - **Eu <** - a monitor – indicates the effect on the system-wide efficiency of the lb parameter.
- Atwood's Machine Parameters – used to tune the operation of the OAMs. These control the "transfer time" stats as seen in Panel 02. The "remaining drop time" needs to sit at about 10 or more ticks for the AM to remain applicable as a model of digestion. If it drops down to 1 or 2 ticks, the results would be very unpredictable.
  - **g-drop-distance** – a slider – this sets the drop time of an OAM.
  - **g-acceleration** – a slider – this sets the acceleration due to gravity that controls the AM dynamics. A value of 1 is strongly recommended, but experimentation with other values is interesting.

5.4 Panel 04 – Debug and Data Collection Tools

Panel 04 has two sub-panels. All of the controls and monitors in this panel were built during the verification process. I decided to leave them as active controls for two reasons:

- Others may wish to change the model in various ways, and these tools may prove useful when the code is being re-verified by other programmers.
- Even if you are not a coder, these debug tools may give you a view inside the operations of the model that is not available via the various monitors and displays in the user interface.



PANEL 04 DEBUG AND DATA COLLECTION TOOLS

04a DEBUG TOOLS

Intructions:

To toggle the debug feature on or off use this button. This opens a log file and enables debug options. If debug is toggled off, the other options do not work.

Then choose a single step, or 'all', in the chooser;

STANDARD OPERATION BUTTONS:

Be sure other 'Go' buttons are off before you turn this on.

SINGLE-STEP WITHIN A TICK - should be pushed in order to debug a

04b CSV DATA COLLECTION TOOLS

Instructions:

The entire contents of any run can be saved to a CSV file (Comma Separated Values file) at any time. Simply open the control panel (small carots at the bottom left) and enter the command 'export-world filename.csv'. Such 'whole-world' files can then be re-imported into NetLogo MppLab later, and the run continued. Or, they can be loaded into MS Excel for data manipulation.

Use this switch to turn data collection off or on for all graphs.

Use this button to discard all data.

OR:
Each graph in NetLogo maintains a small database of plotted points. These points can be saved to a CSV (comma separated values) file which can then be loaded into MS Excel for data manipulation. At any time, simply right-click on the graph and select 'export'.

OR:
The button below can be used to activate a brief routine that dumps the data from six selected graphs:
PANEL 01: Populations; Ave Eu of OAMs;
PANEL 02: Prey Numbers; Counts By T-Level; Eu By T-Level; Max T-Level.

Dumps data from six plots - may take a minute or two.

5/6-Histograms; 7/8-Line Graphs --->>>

5.4.1 Panel 04a – Debug tools

These controls are organized into seven groups as follows:

- **Group 1 – "Toggle Debug" button, and associated monitor.** The code contains many calls to a routine called "to LOG-TO-FILE" which prints a stream of data to a log file as the model runs. Let's call it the log data stream. This button turns on the collection and presentation of the log data stream. This button MUST be pushed to turn on logging **before** any other debug actions are taken. If not, unpredictable things may happen, and the program may show errors relating to file accessibility or similar things. The monitor indicates whether a file is open or not. Use this toggle to enable a T-junction by which the data stream is sent both to the log file, but also to the NetLogo standard "Command Centre". This special window is available via a small set of very small caret triangles in the bottom left corner. Use the expand caret (pointing upwards) to open the command centre window, then use the two-headed arrow to expand it to fill half of the screen. Then the debug data stream will be visible in the command centre. See the graphic for Panel 01 to see the location of these tiny carets (section 5.1).
- **Group 2 – gs-debug-step-chooser.** Use this chooser to select which sub-routine is to produce a log data stream.

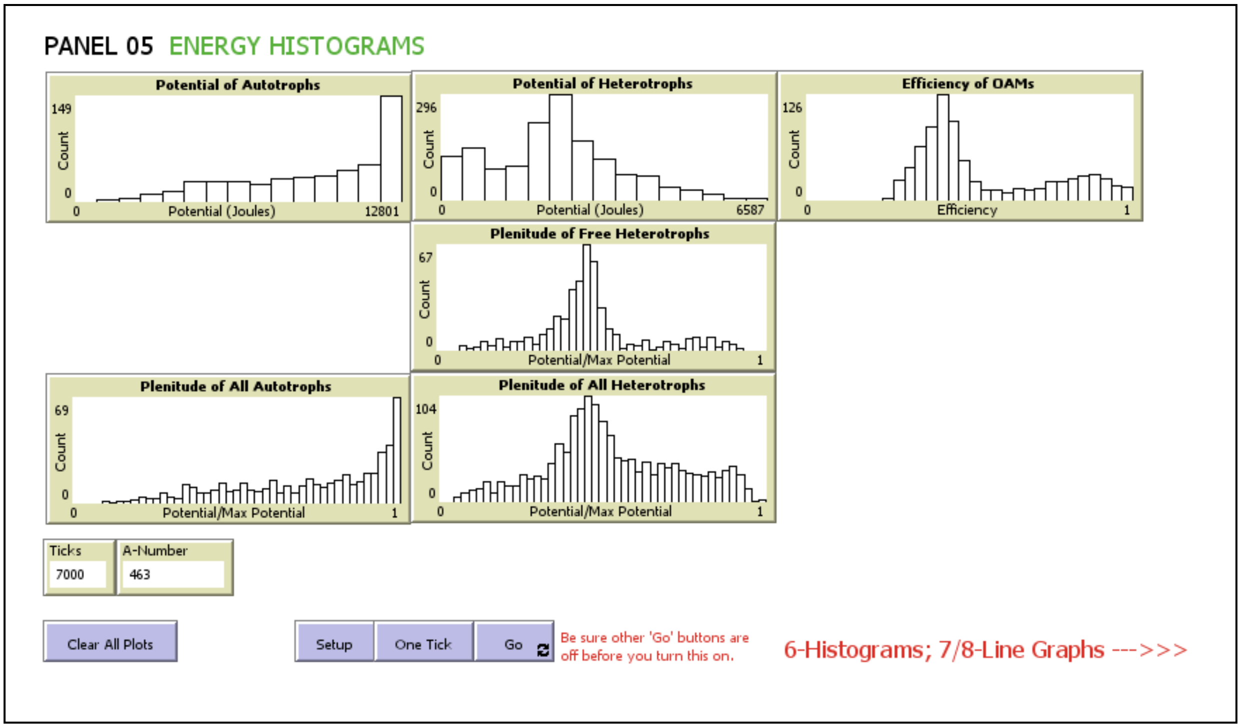
- **Group 3 – Setup, One Tick and Go buttons.** These are copies of three buttons described previously. They occur in each panel. Read sections 4.2.5 and 4.2.6 for detailed information.
- **Group 4 – Single step buttons.** It is possible to step through a tick one sub-routine at a time. Use this, together with the chooser in Group 2, to focus on a single sub-routine at a time.

5.4.2 Panel 04b – CSV Data Collection Tools

Most of this sub-panel is taken up with generic instructions on how to use the built-in NetLogo exportation tools. There are two groups of additional controls:

- **[db-plot-data]** – this is a Boolean switch that toggles the data plotting function for all graphs on and off. If you want to see the graphs develop and/or you want to dump the data to CSV files later, this must be on. If you want to skip through the transient behaviour of the model, this can be off. It is recommended it be left on, as the best most interesting behaviour is when the model is in its transient phase.
- **Record Selected Plots** – This button is used to send the data contained in the seven selected plots to CSV files, to be read into MS Excel later for analysis. The seven plots indicated in the text are the sources of the data for the seven CSV files.

5.5 Panel 05 – Energy Histograms



Ticks 7000

A-Number 463

Be sure other 'Go' buttons are off before you turn this on.

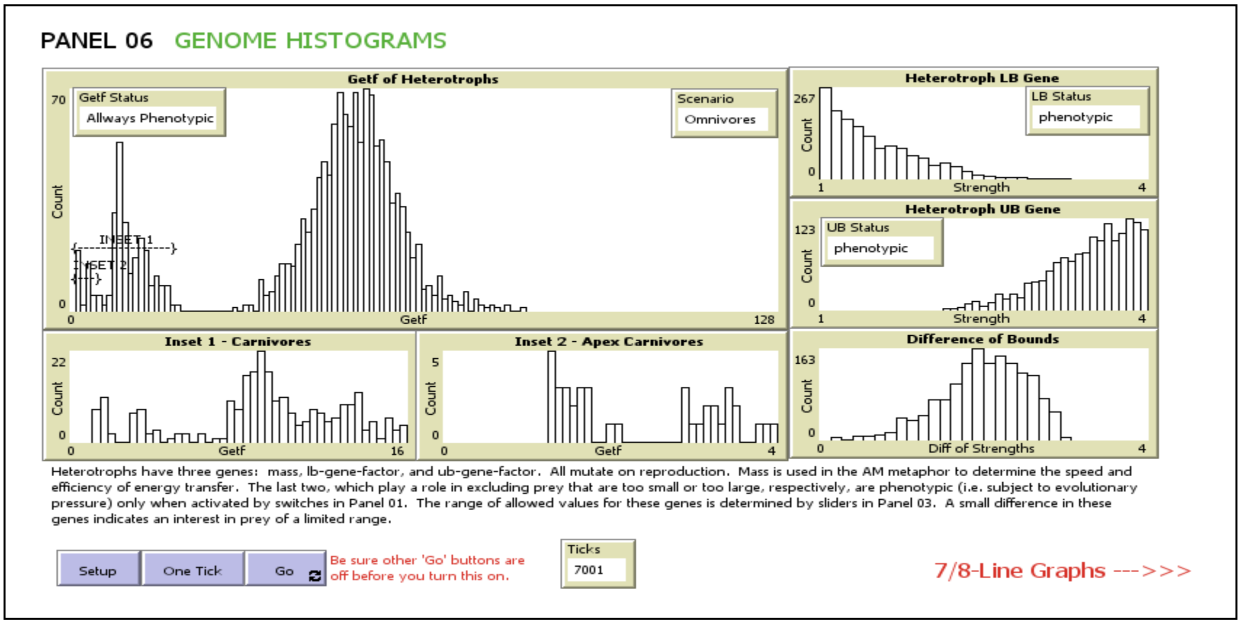
This panel has the standard set of control buttons, a couple of simple monitors, and six histograms. For information about the use of the buttons, read sections 4.2.5 and 4.2.6. The two monitors are:

- **Ticks** – a monitor – simply the number of ticks executed so far. Since this panel lacks any line graphs, it is useful to know the number of ticks.
- **A-Number** – a monitor – the number of autotrophs currently alive.

This six histograms are:

- **Potential of Autotrophs** – a histogram – all agents hold energy as potential energy. Autotrophs are given a full allotment when initialized. They lose energy when heterotrophs snack upon them. This graph shows the distribution of energy among the autotrophs.
- **Potential of Heterotrophs** – a histogram – this shows the distribution of energy among the heterotrophs.
- **Efficiency of OAMs** – a histogram – **THIS IS A KEY GRAPH**. Each active OAM (i.e. each active feeding event) is characterized by an efficiency rating calculated as η = (**getf**$_{pred}$)/(**getf**$_{prey}$). The MPP, as implemented based on the mechanics of the AM, should cause a system to evolve towards an efficiency of 0.5. (See sections 4.1.1, and Panel 01). Indeed, the average efficiency does quickly approach a value of 0,5 and stay there. However, this more detailed histogram indicates that the distribution is not nicely spread about the value of 0.5. In fact, there is a huge bulge between 0.25 and 0.5, with a small hump between 0.5 and 1.0. The **g-h-ub-upper-bound** and **g-h-lb-lower-bound** sliders control the end points of the distribution. The shape in the middle is fascinating to consider. Also see the line graph "Eu By T-Level" in Panel 02.
- **Plenitude of Free Heterotrophs** – a histogram – each heterotroph currently not in an OAM has a fraction of full energy. Those with almost a full complement of energy have high plenitude. Those with **RET** energy reproduce immediately via fission, and the daughters have a "plenitude" of **RET** / 2, or about 0.48.
- **Plenitude of all Autotrophs** – a histogram – essentially the same data as "Potential of Autotrophs". This is a vestigial graph.
- **Plenitude of All Heterotrophs** – a histogram – each heterotroph has a fraction of its full complement of energy, called its plenitude. This includes all heterotrophs, when in an OAM or not.

5.6 Panel 01 – Genome Histograms



Heterotrophs have three genes: mass, lb-gene-factor, and ub-gene-factor. All mutate on reproduction. Mass is used in the AM metaphor to determine the speed and efficiency of energy transfer. The last two, which play a role in excluding prey that are too small or too large, respectively, are phenotypic (i.e. subject to evolutionary pressure) only when activated by switches in Panel 01. The range of allowed values for these genes is determined by sliders in Panel 03. A small difference in these genes indicates an interest in prey of a limited range.

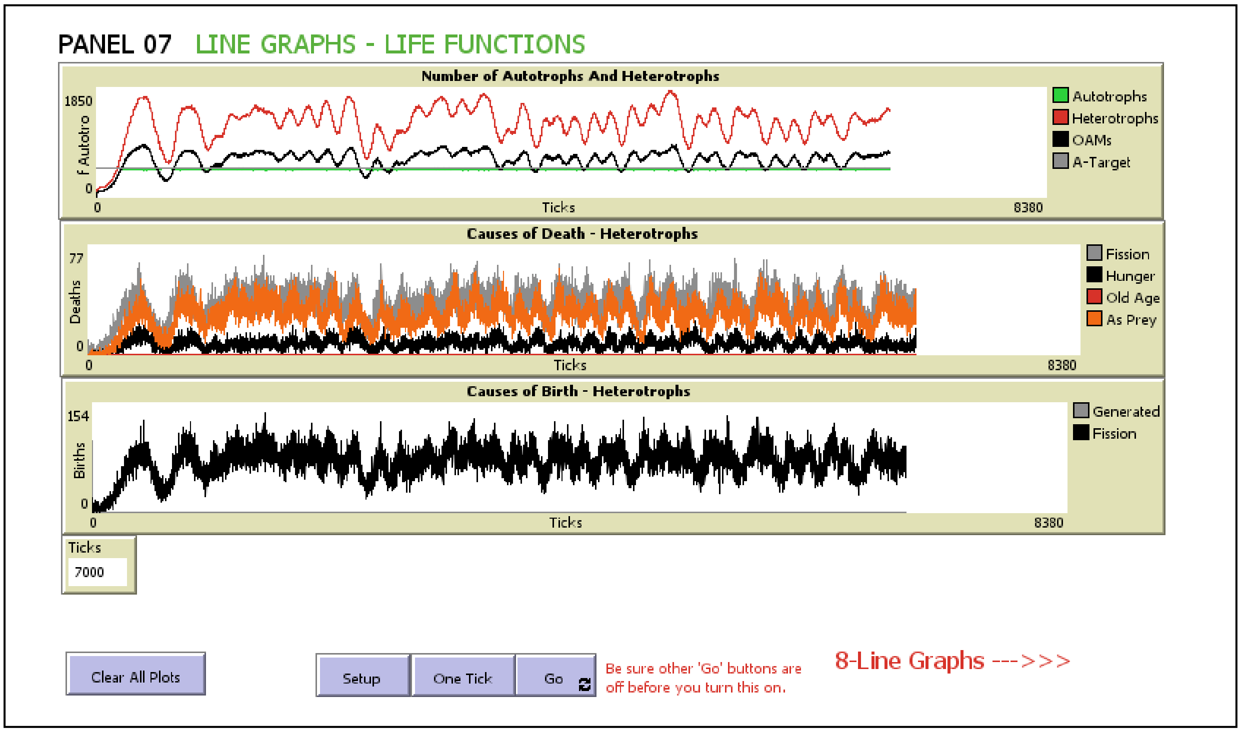
Be sure other 'Go' buttons are off before you turn this on.

This panel has a standard set of control buttons with a monitor.

It also displays six histograms that are fascinating to watch as the ecosystem evolves:

- **Getf of Heterotrophs** – a histogram – The autotrophs (plants) have a **getf** = 128 (not shown). **Getf** is an acronym for genetic energy transfer factor, and it determines the ability of one agent to eat another. The two following histograms are part of this data set, and form insets. This histogram covers **getf** values from 0 to 128. Inset 1 covers **getf** values from 0 to 16. Inset 2 covers values from 0 to 4. The trophic levels appear as domes or waves that spring up around the **getf** = 80 mark, and march leftwards. The apex carnivores are located to the left of inset 2.
- **Inset 1 – Carnivores** – a histogram – heterotrophs with **getf** < 32 are unable to consume plants and must therefore exist as carnivores.
- **Inset 2 – Apex Carnivores** – a histogram heterotrophs with a **getf** < 8 are unable to consume anything other than other carnivores. I.e. all herbivores are inedible to them. The apex carnivores are among these agents.
- **Heterotroph LB Gene** – a histogram – The LB and UB genes control the range of **getf** values that a heterotroph can consider as suitable prey. These genes are subject to mutation and evolutionary pressure, within the limits set by the LB and UB parameters set in Panel 03. We see that, when they are set at 1 and 4 respectively, the LB and UB genes are grouped within those bounds. But they exhibit a distribution of values within that range that implies a condition of maximized entropy (a power law distribution?).
- **Heterotroph UB Gene** – a histogram – the partner of the histogram for the LB gene, described immediately above.
- **Difference of Bounds** – a histogram – this plots the difference between the UB and LB genes, for all heterotrophs. It also exhibits a distribution characteristic of maximum entropy.

5.7 Panel 07 – Line Graphs – Life Functions



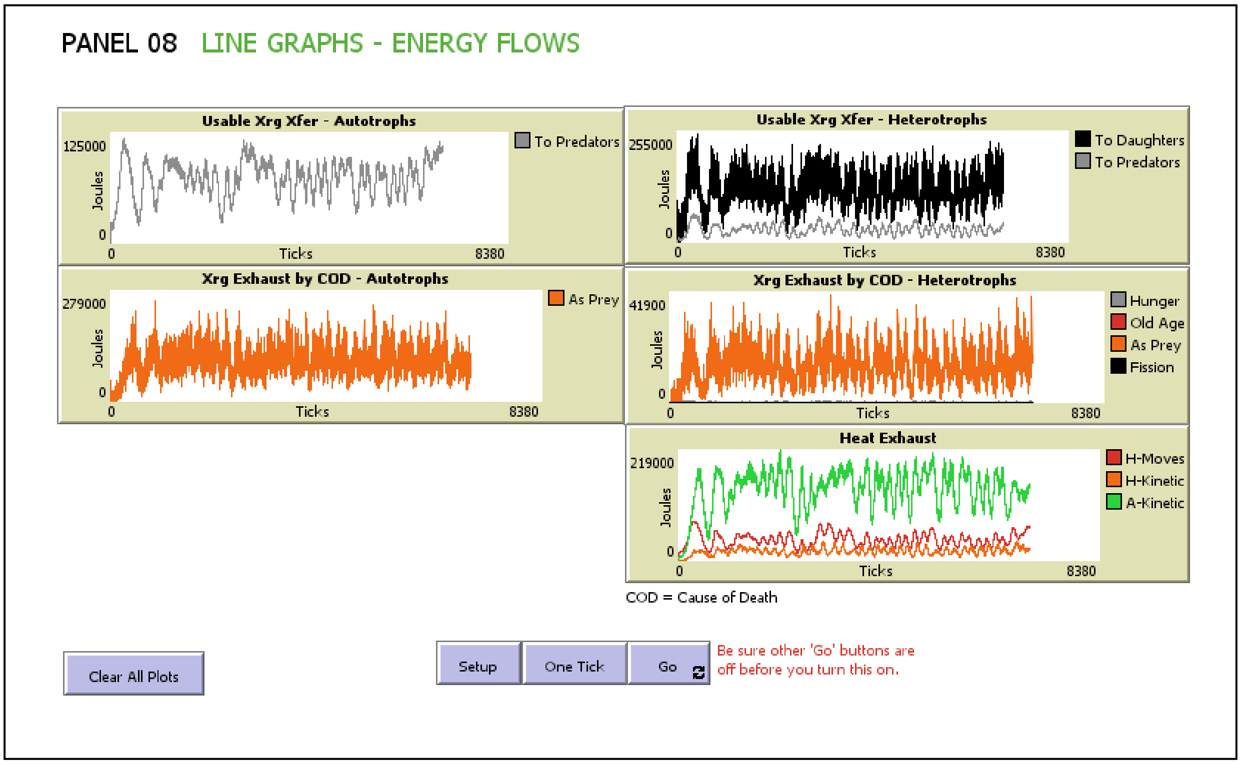
Ticks: 7000

Be sure other 'Go' buttons are off before you turn this on.

This panel contains some standard buttons and monitors, and three line graphs as follows:

- **Number of Autotrophs and Heterotrophs** – a line graph – There are four lines: the target population for autotrophs (grey); the number of autotrophs (green); the number of heterotrophs (red); and the number of active feeding events or OAMs (black).
- **Causes of Death – Heterotrophs** – a line graph – There are four lines indicating death by fission (grey); hunger (black); old age (red) and as prey (orange).
- **Causes of Birth – Heterotrophs** – a line graph – There are two lines indicating birth by initial generation (grey) and fission (black).

5.8 Panel 08 – Line Graphs – Energy Flows



COD = Cause of Death

Be sure other 'Go' buttons are off before you turn this on.

This panel contains some standard buttons and monitors, and five line graphs as follows:

- **Usable Xrg Xfers - Autotrophs** – a line graph – this is the energy transferred to herbivores.
- **Usable Xrg Xfers - Heterotrophs** – a line graph – this is the energy transferred to daughters and predators.
- **Xrg Exhaust by COD – Autotrophs** – a line graph – This is a companion to the first line graph above it. It shows the waste heat produced by herbivores.
- **Xrg Exhaust by COD – Heterotrophs** – a line graph – This is a companion to the second line graph above. It shows the waste heat produced by heterotrophs for a variety of causes of death. Death as expired prey is the dominant channel of such loss of energy. Most of this is due to a partial last bite that the predator must forego.
- **Heat Exhaust** – a line graph – this shows three channels by which waste heat is released: due to movement by heterotrophs at a rate of EPM per move (red); due to inefficient consumption of heterotrophs (orange); and due to inefficient consumption of autotrophs (green).

6 Annex B – Related Models

In my study of the dynamics of sustainable economic systems I have developed a number of models that explore various aspects of systems that evolve to steady state.

6.1 PSoup

Written in C++, this model explores the nature of evolutionary forces in a purely biophysical economy. Based on a previous model called "Simulated Evolution" by Michael Palmiter, **PSoup** places a number of bugs into a primordial soup and tracks their development. Offering scenarios at many levels, it demonstrates a variety of effects of evolution from simple development of effective food search patterns in the lowest levels, to development of the five main senses, genetic cross-over and sexual reproduction. The amazing observation, for me, coming out of this model was the speed and certainty of the development of complexity. Whenever a complex option was made available in the potential genetic space of the bugs, it was quickly explored and very complex interactions between a cohort of highly varied genotypes would emerge. I also learned, from this model, that a constant flow of energy was a necessary condition for a sustainable biophysical economy, and a necessary pre-cursor for the emergence of complexity.

Many concepts from Dr Palmiter's model have been carried over into most of the models described below. In particular, the **DAT**, **RET** and **EPM** parameters from **PSoup** have been implemented in **MppLab** to manage the life functions of the heterotrophs, and the dominant steady-state search pattern called the "dodger" has been implemented to control their movements. Each heterotroph has eight cells around it, and must "decide" in each tick in which of the eight cells it is most likely to find a heterotroph. The best search behaviour is one in which 80% of the time the agent simply steps straight forward, but 10% of the time it turns 45° to the left before stepping forward, and 10% of the time is turns 45° to the right before stepping forwards. A stochastic selection of the three options leads to good performance. I did not want to confuse the effects of **PSoup**-like evolution of search patterns with the **MppLab**-like evolution of **getf** genes, so I have implemented a non-evolving dodger search pattern as a baseline behaviour here.

6.2 ModEco

There are two versions of **ModEco**. Written in C++, the first-written version of **ModEco** is a hybrid economy consisting of a **PSoup**-like biophysical economy joined to a financial economy. In **ModEco** every transfer of matter and/or energy from agent to agent is facilitated by a reciprocal transfer of money. Again, as long as there is a consistent flow of energy, the economy evolves towards a steady-state biophysical condition. However, sustainability of the the financial economy has proven to be a very elusive property. It was discovered that economic sustainability was only possible if profit and loss were eliminated from the financial sector, and in its place an extremely precise regime of recognition and preservation of biophysical value be instituted. The scenario that exhibited sustainability under this harsh condition was called, somewhat jokingly, the "Perpetual Motion Machine", or "The PMM". In more technical terms, the equilibrium achieved in the PMM is an unstable equilibrium that must be maintained by continuous external intervention.

6.3 ModEco and the PMM

Written in NetLogo, this model is a replication of the C++ model, but only of the single scenario called "The PMM". Again, three observations are clear from this model, and from the two previous models, when looking back over the common characteristics. First, the steady-state condition of any of these models demonstrates clear evidence of entropy rising to a maximal value, and then staying there. The distributions of alleles (in **PSoup**), or of wealth (in **ModEco** and The PMM) develop and then hold shapes characteristic of maximal entropy. Second, the emergence of complex economic organizations and inter-agent interactions seems to be an inevitable part of the evolution towards such a sustainable configuration. Third, and perhaps most important, the steady state can only be maintained by programming techniques that address the inherent instability of the equilibria associated with those steady states.

6.4 EiLab

Written in C++, this model enables a rather detailed investigation of the phenomenon of rising entropy in economic models. Inspired by the work of Dragulescu and Yakovenko (2000), it implements a capital exchange model in which I could study the origins and impact of entropy production in such models. **EiLab** stands for "Entropic Index Laboratory". I like this "laboratory" approach, because I can study many scenarios, generate lots of data, and then study the data. There were a few key observations coming out of this model. First, the study of entropy in such models is highly analogous to the study of entropy by Gibbs in his development of the concepts of "Gibbs' Free Energy". Second, in spite of the similarity to entropy, as Gibbs understood it, the entropy being investigated is not thermodynamic entropy, but a purely mathematical abstraction of it which is produced by a logical system executing logical processes distinct from chemical or heat-related effects. Third, the emergence of maximum entropy distributions was inevitable in these models, but the emergence of complexity was not, implying that there may be two distinct sets of conditions causing (a) the production of entropy, and (b) the production of complexity.

6.5 CmLab

Written in NetLogo, this model is another "capital exchange" model in the tradition of those of Benatti, Dragulescu and Yakovenko as found in the BDY models. That is to say, CmLab models the exchange of capital (of money) along with its stores and flows, but does not model the exchange of physical goods or services, nor does it model stores and flows of matter or energy. Whereas **PSoup** contains no financial subsystem, and **ModEco** is a hybrid biophysical / economic system, **EiLab** and **CmLab** contain no biophysical subsystem. This is a continuation of my study of sustainability of economic systems. The purpose of this model is to explore the analogy between the first law of thermodynamics (the conservation of energy) and the conservation of money under the regimes imposed by double-entry book-keeping and fractional reserve banking. My point would be that cash is "sufficiently-well conserved" under these two regimes that it presents an effective basis for the emergence of a purely financial version of those stochastic phenomena that are well-known and well-studied in thermodynamics. I would argue that the first law of thermodynamics is an accounting law, and the phenomenon of conservation of money is a purely financial counterpart to it. The production of thermodynamic entropy would also then incline us to look for a similar purely financial phenomenon, which is what was found in the BDY models, and further demonstrated in **EiLab -**

Model I.

6.6 OamLab

Written in NetLogo, this model is one of three developed to explore more fully the phenomenon of production of complexity in economic systems. This is a continuation of my study of sustainability of economic systems. It is the more simple senior member of a trio of models in which I explore the "Maximum Power Principle" as first proposed by A. J. Lotka and as developed by H. T. Odum. Both men argued that there is a need for a "fourth" law of thermodynamics, required to explain the persistent emergence of complexity. However, I found that Odum's ideas on this topic are difficult to decipher for a number of reasons. Ultimately, I found it necessary to restate his MPP as a set of three closely-related falsifiable hypotheses, and only then could I design models to test those hypotheses. This is the first of three such models designed to test the validity of my restatement of the MPP. The prediction was that this model would evolve to a stationary state in which the average efficiency was 0.5. Strangely, the stationary state average efficiency was 0.618, and not 0.5. This lead to the discovery of a dispute between Odum and Silvert in the early 1980s over which of the two numbers was the correct prediction. On further investigation, it turns out that there are, in fact, two valid answers, and both results are easily validated as consistent with the MPP. So, I consider that model, simple though it be, as supporting the hypotheses of the MPP.

6.7 MppLab

Written in NetLogo, this is the middle member of the trio of models exploring the validity of the MPP. Within an arena of freely roaming organisms (HOAMs) a competition for persistence ensues in which hunger and old age is the probable end of 50% of the population, and successful reproduction is the probable end of the rest. All biomechanical and biological feeding events have their dynamics modeled on the action of the AM, and in this way those aspects of time-regulation are modeled. This has very exciting results, as it is able to model a trophic web, in action, with from 6 to 10 trophic levels all operational at once. I have made a YouTube video using this model.

6.8 TpLab

Written in NetLogo, this model is intended to explore the hypothesized phenomenon that I am calling "Teleological Pruning". I argue that the beliefs and practices of any society are "pruned" and shaped by the same evolutionary pressures that shape our genes. Our societies are therefore shaped to conform to and agree with the effects of the Maximum Power Principle. This would explain why consumerism has emerged as the most destructive and persistent form of economic activity in our highly complex modern global economy. Our global economy has become the most powerful economy ever to have existed (in terms of energy consumed per year, and possibly in terms of energy consumed per person per year). Our beliefs (e.g. the necessity of endless economic growth) and our practices (materialism and consumerism) are now aligned with this destructive social trend. In **TpLab**, social beliefs are pitted against the effects of evolutionary pressures, with interesting effects.

7 Annex C – Sample Debug Output

7.1 To Setup

```
observer: "MppLab_Log_170402_241213.txt opened."
observer: ""
observer: "Do-setup: Debug on; tick = 0"
observer: ""
observer: "  Do-set: ATWOOD MACHINE PARAMETERS (Sliders):"
observer: "  Do-set: g-drop-distance - 100"
observer: "  Do-set: g-acceleration - 1"
observer: ""
observer: "  Do-set: AUTOTROPH POPULATION PARAMETERS (Sliders):"
observer: "  Do-set: g-a-target-population - 50"
observer: ""
observer: "  Do-set: INTERNAL AUTOTROPH PARAMETERS:"
observer: "  Do-set: g-a-initial-getf - 128"
observer: ""
observer: "  Do-set: HETEROTROPH POPULATION PARAMETERS:"
observer: "  Do-set: g-h-initial-population - 10"
observer: "  Do-set: g-h-initial-getf - 100"
observer: "  Do-set: g-h-DAT - 1600"
observer: "  Do-set: g-h-mutation-factor - 0.1"
observer: "  Do-set: g-h-satiation-factor - 0.98"
observer: "  Do-set: g-h-ub-upper-bound - 4"
observer: "  Do-set: g-h-lb-lower-bound - 1"
observer: "  Do-set: Scenario number - 1"
observer: "  Do-set: Scenario name - Omnivores"
observer: "  Do-set: Random seed - 7"
(autotroph 4): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 45): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 47): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 25): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 19): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 30): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 37): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 18): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 32): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 23): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 10): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 2): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 5): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 43): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 17): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 20): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 33): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 11): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 8): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 46): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 15): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 34): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 1): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 27): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 42): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 40): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 13): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 39): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 49): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 35): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 0): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 24): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 7): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 3): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 9): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 12): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 21): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 44): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 28): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 22): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 16): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 29): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 36): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 38): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 31): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 26): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 6): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 48): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 41): "  Do-pre-tick: A(max-xrg) - (12800)"
(autotroph 14): "  Do-pre-tick: A(max-xrg) - (12800)"
(heterotroph 56): "  Do-xxx: H(max-xrg,RET,EPM) - (10000,9500,50)"
(heterotroph 58): "  Do-xxx: H(max-xrg,RET,EPM) - (10000,9500,50)"
(heterotroph 51): "  Do-xxx: H(max-xrg,RET,EPM) - (10000,9500,50)"
(heterotroph 55): "  Do-xxx: H(max-xrg,RET,EPM) - (10000,9500,50)"
```

```
(heterotroph 57): "  Do-xxx: H(max-xrg,RET,EPM) - (10000,9500,50)"
(heterotroph 50): "  Do-xxx: H(max-xrg,RET,EPM) - (10000,9500,50)"
(heterotroph 54): "  Do-xxx: H(max-xrg,RET,EPM) - (10000,9500,50)"
(heterotroph 52): "  Do-xxx: H(max-xrg,RET,EPM) - (10000,9500,50)"
(heterotroph 59): "  Do-xxx: H(max-xrg,RET,EPM) - (10000,9500,50)"
(heterotroph 53): "  Do-xxx: H(max-xrg,RET,EPM) - (10000,9500,50)"
observer: "  Do-xxx: All aggregates updated."
```

7.2 To do-pre-tick

```
observer: ""
observer: "Do-pre-tick: Debug on.; tick = 0"
observer: "  Do-pre-tick: heterotrophs aged."
observer: "  Do-pre-tick: Halt at tick - -1"
observer: "  Do-pre-tick: Current tick - 1"
observer: "  Do-pre-tick: Routine completed."
observer: "  Do-xxx: All aggregates updated."
```

7.3 To do-move

```
observer: ""
observer: "Do-move: Debug on; tick = 1"
(heterotroph 55): "  Do-move: H(heading,xrg,die-flag) - (315,9188,0)"
(heterotroph 56): "  Do-move: H(heading,xrg,die-flag) - (135,7205,0)"
(heterotroph 58): "  Do-move: H(heading,xrg,die-flag) - (270,6842,0)"
(heterotroph 52): "  Do-move: H(heading,xrg,die-flag) - (90,6307,0)"
(heterotroph 54): "  Do-move: H(heading,xrg,die-flag) - (180,6272,0)"
(heterotroph 59): "  Do-move: H(heading,xrg,die-flag) - (225,6792,0)"
(heterotroph 50): "  Do-move: H(heading,xrg,die-flag) - (135,8301,0)"
(heterotroph 53): "  Do-move: H(heading,xrg,die-flag) - (45,6971,0)"
(heterotroph 51): "  Do-move: H(heading,xrg,die-flag) - (315,9807,0)"
(heterotroph 57): "  Do-move: H(heading,xrg,die-flag) - (180,7251,0)"
observer: "  Do-move: procedure completed"
observer: "  Do-xxx: All aggregates updated."
```

7.4 To do-feed

```
observer: ""
observer: "Do-feed: Debug on; tick = 1"
(heterotroph 54): "  Do-feed: Hunting."
(heterotroph 51): "  Do-feed: Satiated!  Not hunting."
(heterotroph 59): "  Do-feed: Hunting."
(heterotroph 59): "  Do-feed: Captured - (autotroph 32)"
(heterotroph 59): "    Do-feed: self - (heterotroph 59)"
(heterotroph 59): "    Do-feed: rh-hoam - (autotroph 32)"
(heterotroph 59): "    Do-feed: light-mass - 100"
(heterotroph 59): "    Do-feed: heavy-mass - 128"
(heterotroph 59): "    Do-feed: Eu-in-oam - 0.78125"
(heterotroph 59): "    Do-feed: mass-total - 228"
(heterotroph 59): "    Do-feed: mass-diff - 28"
(heterotroph 59): "    Do-feed: max-rh-potential - 12800"
(heterotroph 59): "    Do-feed: cur-rh-potential - 10303"
(heterotroph 59): "    Do-feed: numerator - 2497"
(heterotroph 59): "    Do-feed: mass-assembly-ratio - 0.12280701754385964"
(heterotroph 59): "    Do-feed: factor - 64"
(heterotroph 59): "    Do-feed: denominator - 7.859649122807017"
(heterotroph 59): "    Do-feed: time - 17.824103363543585"
(heterotroph 59): "    Do-feed: time-plus-one - 18.824103363543585"
(heterotroph 59): "    Do-feed: delta-rh-potential - 288.0420458550007"
(heterotroph 59): "    Do-feed: delta-rh-potential - 288.0420458550007"
(heterotroph 59): "    Do-feed: delta-lh-potential - 225.03284832421932"
(heterotroph 59): "    Do-feed: delta-oam-kinetic - 63.00919753078139"
(heterotroph 59): "    Do-feed: (ticks,epm,bite) - (1,50,225)"
(heterotroph 59): "    Do-feed: available-room - 3208"
(heterotroph 59): "    Do-feed: excess-potential - 0"
(heterotroph 58): "  Do-feed: Hunting."
(heterotroph 58): "  Do-feed: Captured - (autotroph 15)"
(heterotroph 58): "    Do-feed: self - (heterotroph 58)"
(heterotroph 58): "    Do-feed: rh-hoam - (autotroph 15)"
(heterotroph 58): "    Do-feed: light-mass - 100"
(heterotroph 58): "    Do-feed: heavy-mass - 128"
(heterotroph 58): "    Do-feed: Eu-in-oam - 0.78125"
(heterotroph 58): "    Do-feed: mass-total - 228"
(heterotroph 58): "    Do-feed: mass-diff - 28"
(heterotroph 58): "    Do-feed: max-rh-potential - 12800"
(heterotroph 58): "    Do-feed: cur-rh-potential - 7827"
(heterotroph 58): "    Do-feed: numerator - 4973"
(heterotroph 58): "    Do-feed: mass-assembly-ratio - 0.12280701754385964"
(heterotroph 58): "    Do-feed: factor - 64"
(heterotroph 58): "    Do-feed: denominator - 7.859649122807017"
(heterotroph 58): "    Do-feed: time - 25.154034396664315"
(heterotroph 58): "    Do-feed: time-plus-one - 26.154034396664315"
(heterotroph 58): "    Do-feed: delta-rh-potential - 403.26341788440743"
(heterotroph 58): "    Do-feed: delta-rh-potential - 403.26341788440743"
(heterotroph 58): "    Do-feed: delta-lh-potential - 315.04954522219333"
(heterotroph 58): "    Do-feed: delta-oam-kinetic - 88.2138726622141"
(heterotroph 58): "    Do-feed: (ticks,epm,bite) - (1,50,315)"
```

```
(heterotroph 58): "    Do-feed: available-room - 3158"
(heterotroph 58): "    Do-feed: excess-potential - 0"
(heterotroph 52): "  Do-feed: Hunting."
(heterotroph 55): "  Do-feed: Hunting."
(heterotroph 55): "  Do-feed: Captured - (autotroph 44)"
(heterotroph 55): "    Do-feed: self - (heterotroph 55)"
(heterotroph 55): "    Do-feed: rh-hoam - (autotroph 44)"
(heterotroph 55): "    Do-feed: light-mass - 100"
(heterotroph 55): "    Do-feed: heavy-mass - 128"
(heterotroph 55): "    Do-feed: Eu-in-oam - 0.78125"
(heterotroph 55): "    Do-feed: mass-total - 228"
(heterotroph 55): "    Do-feed: mass-diff - 28"
(heterotroph 55): "    Do-feed: max-rh-potential - 12800"
(heterotroph 55): "    Do-feed: cur-rh-potential - 9758"
(heterotroph 55): "    Do-feed: numerator - 3042"
(heterotroph 55): "    Do-feed: mass-assembly-ratio - 0.12280701754385964"
(heterotroph 55): "    Do-feed: factor - 64"
(heterotroph 55): "    Do-feed: denominator - 7.859649122807017"
(heterotroph 55): "    Do-feed: time - 19.673336742185565"
(heterotroph 55): "    Do-feed: time-plus-one - 20.673336742185565"
(heterotroph 55): "    Do-feed: delta-rh-potential - 317.11069685961877"
(heterotroph 55): "    Do-feed: delta-rh-potential - 317.11069685961877"
(heterotroph 55): "    Do-feed: delta-lh-potential - 247.74273192157716"
(heterotroph 55): "    Do-feed: delta-oam-kinetic - 69.36796493804161"
(heterotroph 55): "    Do-feed: (ticks,epm,bite) - (1,50,247)"
(heterotroph 55): "    Do-feed: available-room - 812"
(heterotroph 55): "    Do-feed: excess-potential - 0"
(heterotroph 56): "  Do-feed: Hunting."
(heterotroph 50): "  Do-feed: Hunting."
(heterotroph 53): "  Do-feed: Hunting."
(heterotroph 57): "  Do-feed: Hunting."
observer: "  Do-feed: procedure completed"
observer: "  Do-xxx: All aggregates updated."
```

7.5 To do-reproduce

```
observer: ""
observer: "Do-reproduce: Debug on; tick = 1"
(heterotroph 51): "  Do-reproduce: H(age,xrg,oam-flag,rep-flag) - (221,9807,0,1)"
(heterotroph 51): "  Do-reproduce: Heterotroph Ma - 51"
(heterotroph 61): "  Do-xxx: H(max-xrg,RET,EPM) - (10000,9500,50)"
(heterotroph 61): "  Do-reproduce: Heterotroph Dx - 61"
(heterotroph 61): "  Do-reproduce: Mutate getf - H(old,delta,new) - (100,7,107)"
(heterotroph 61): "  Do-xxx: H(max-xrg,RET,EPM) - (10790,10251,53)"
(heterotroph 61): "  Do-reproduce: Mutate lb gene - H(old,delta,new) - (1,-0.0103,1.0103)"
(heterotroph 61): "  Do-reproduce: Mutate ub gene - H(old,delta,new) - (4,0.1887,3.8113)"
(heterotroph 61): "  Do-reproduce: endowment-offered - 4903"
(heterotroph 61): "  Do-reproduce: energy accepted - 4903"
(heterotroph 61): "  Do-reproduce: endowment-rejected - 0"
(heterotroph 60): "  Do-xxx: H(max-xrg,RET,EPM) - (10000,9500,50)"
(heterotroph 60): "  Do-reproduce: Heterotroph Dx - 60"
(heterotroph 60): "  Do-reproduce: Mutate getf - H(old,delta,new) - (100,-8,92)"
(heterotroph 60): "  Do-xxx: H(max-xrg,RET,EPM) - (9219,8758,46)"
(heterotroph 60): "  Do-reproduce: Mutate lb gene - H(old,delta,new) - (1,-0.0050,1.005)"
(heterotroph 60): "  Do-reproduce: Mutate ub gene - H(old,delta,new) - (4,0.223,3.777)"
(heterotroph 60): "  Do-reproduce: endowment-offered - 4903"
(heterotroph 60): "  Do-reproduce: energy accepted - 4903"
(heterotroph 60): "  Do-reproduce: endowment-rejected - 0"
observer: "  Do-reproduce: procedure completed"
observer: "  Do-xxx: All aggregates updated."
```

7.6 To do-die

```
observer: ""
observer: "Do-die: Debug on; tick = 1"
(heterotroph 51): "  Do-die: A(age,xrg,oam,cod) - (221,9807,0,2)"
(heterotroph 51): "Do-die: H(xrg,die-flag,cod) - (9807,1,2)"
observer: "  Do-die: procedure completed"
observer: "  Do-xxx: All aggregates updated."
```

7.7 To do-post-tick

```
observer: ""
observer: "Do-Post-tick: Debug on; tick = 1"
observer: "  Do-xxx: All aggregates updated."
observer: "  Do-post-tick: procedure completed."
```