Pricing Issues

ModEco V2.04α

Working Paper

By: Orrery Software
Author: Garvin H Boyle

Date: 04 Mar 2014
Revised: 05 Apr 2014

TABLE OF CONTENTS

Pricing Issues – ModEco V2.04α – Working Paper

1 Purpose

1.1 A ModEco Economy

This note is not intended to include a detailed description of ModEco. However, in order to understand the contents, a brief description of the model economies, and the application interface will be necessary.

In brief, ModEco is a laboratory in which a user can design a simple economy, watch it run on the desktop, and capture large quantities of data as it runs. The goal is to study sustainable economics consistent with the laws of conservations of mass and energy, and to determine the necessary and sufficient conditions for sustainability. However, to date, the only type of sustainable economy achieved has very rigidly controlled prices. Such rigid (and physically unachievable) pricing lacks applicability, and so is the opposite of what one would want to model in a realistic sustainable economy.

A ModEco-based economy has two sub-systems: a biophysical sub-system in which mass and energy are conserved, agents eat, work, and live to procreate, or die by starvation or old age; and an economic sub-system in which money is conserved, and all exchanges of goods and services between agents are reciprocated by an exchange of cash. There is one cash crop: wheat. Agents negotiate prices on every economic transaction. Agents are physically restricted to living and working within a 2-dimensional commuting area, and cannot foresee what negotiation strategy other agents will use when negotiating prices, and so have imperfect knowledge of the economic system, but act in a self-interested fashion.

1.2 ModEco Interface

ModEco has two economic engines. The first, referred to as the cyclic engine, implements a closed cycle for both mass and energy. The second is referred to as the linear engine, as it implements a linear flow of energy from Sun through biosphere to Sink. It is with the cyclic engine that the perpetual motion machine (or PMM - the only sustainable economy achieved so far in this project) was constructed. This working paper specifically addresses issues in the cyclic economic engine of version 2.04α of ModEco for those economies other than the PMM. In those economies, in which profit-taking activities are allowed, inflation, deflation or other evolved characteristics of the economy ultimately cause collapse of the economy and death of all agents.

There are two economic engines. The cyclic engine is the topic of this note. The PMM, Sparse and Crowded scenarios will be discussed.

Much of what is discussed in this note is accessible via a set of toolbars with buttons, as shown.

Left panel; Right panel; Status panel; Notes panel; Next Frmr; Next Wrkr; Initiation Wizard; Economic Wizard; View Agent Data; View Agent Genes; View Contact Technical; ReSet; Advance One Function; Advance One Tick; Go Slow Motion; Go; Stop.

1.3 Purpose of This Note

So, the purpose of this article is to describe the pricing techniques in as much detail as possible, in the search for a clue as to why all non-PMM ModEco-based model economies fail to be sustainable. It is hoped I can find an improved approach to modeling price negotiations from which will emerge a stable sustainable economy. I am looking for a mechanism which emerges out of the local actions of agents that do not have global knowledge of the economy. I believe I need to identify the changes in system characteristics that cause the collapse, and introduce a suitable mechanism of negative feedback that is corrective, and so enables sustainability at some stationary state in the economy's state space.

2 Approach

This is a vexing problem that has dogged me for a couple of years now. This working paper is written in an attempt to break the creative logjam. So, my approach in writing this paper is:

- First, to describe in some detail exactly how the pricing mechanisms work in ModEco,
- And then to explore the implications for stable or unstable price behavior,
- And finally, to explore design changes that might lead to a model of a very simple sustainable agrarian society functioning with money as a medium of exchange.

3 How Pricing Mechanisms Work in ModEco

It was certainly naïve of me, and probably unwise, but in my original design plan I decided to model market negotiations in what I thought was a fairly simple way. I conceived of a model economy in which agents met, negotiated a price, and completed a transaction to the apparent benefit of both, or failed to come to agreement on a price, and departed with a deal unmade, also to the apparent benefit of both. This concept is fundamental to the notion of a free-market economy. I conceived of a means whereby these negotiations would have pricing strategies based in genes that were subject to variation and Darwinian selection. I hoped that agents would evolve effective pricing strategies, and the economy would be stable.

My original conception also involved many commodities and many types of goods and services. However, in my struggle to find just one single simple sustainable economy, I have had to abandon much of that energizing early vision, and seek simplicity. Nevertheless, I could not abandon the concept of free-market negotiations.

When any model is being designed, the designer must decide what facets of reality are to be modeled, and what facets are to be ignored or abstracted away. My real goal in the ModEco project is to convince myself that it is possible to model a sustainable economy based on the exchange of money in a free-market economy that respects the laws of physics. So, to meet that goal, some kind of price negotiation mechanism is a necessary element that must not be "abstracted away".

That being the case, over the past two years I have looked for ways to model free-market negotiations. So, my goal is to find the absolutely most simple (most abstracted) model I can produce that is sustainable, but also realistic enough to convince me that sustainable free-market economies are in fact possible.

That being said, what I am about to describe IS NOT, I must admit, very simple.

3.1 Definitions

Frmrs and Wrkrs – These are the mortal agents that are born and die, that work and eat, and that negotiate prices for goods and services. Together, they are called consumers.

Immortals – The materiel manager (MMgr) and the estate manager (EMgr) are two non-local and immortal agents that provide services distantly analogous to central government.

Goods and services – In ModEco goods are based on the various forms of a single base commodity – food – which appears in four forms in four different types of stores. Whole food comes in two forms, depending on the nature of the intended use by an agent. The food may be in a Frmr's inventory store, awaiting sale to a consumer, or in a consumer's supply store, awaiting consumption. The two other forms of food cannot really be called food, at all, in any practical sense. They are waste mass (produced on consumption of the food) and recycled mass (produced by the MMgr when waste mass is converted; to be placed back on a Frmr's field to be included in the next harvest of whole food). The only service that can be purchased is farm labour. A Frmr can hire up to four Wrkrs per tick to work on the farm. The Wrkr is, essentially, selling the energy contained in his body to harvest food into the Frmr's inventory. In the cyclic economic engine, in version 2.04A, it is considered that the energy goes directly from the Wrkr's body into the food in the Frmr's inventory. So, in essence, the Wrkr is selling energy to the Frmr, in the act of performing a service. In this sense, we can consider energy to be a commodity that Wrkrs sell to Frmrs. This is, of course, not directly analogous to reality, but I would contend it is a reasonable abstraction of an extreme economic biophysical reality. A Wrkr cannot expend more energy in a day than he brings into the economic system by working. In an extreme case, if the Wrkr brings the same energy into the economic system as he/she expends, you can model this as an energy cycle. The only way consumable energy comes into the economy is via the expenditure of the energy of Frmrs and Wrkrs. When a farm is working at full efficiency, the Frmr expends 16 Eu, and four Wrkrs expend four Eu each, for a total of 32 Eu per Tick. They bring in a harvest of 160 Eu, for an EROEI of 5. The Frmr or Wrkr that participates in the greater share of the flow of this energy thrives, while the others do not. Labour can be considered a service. Energy can be considered a good. When I say "good" or "goods" in the following, I mean to include energy sold as a service.

Units of measure – There are several units of measure that are somewhat arcane. I did this to abstract away the issues of scale or modeling accuracy, preferring to focus on issues of dynamic changes. The units of measure used in ModEco are:

- **Ticks** – Discrete units of time. Positive integer.
- **MbMEu (MEu)** – Metabolism-based mass-energy units. Used to measure food. Positive real number. Commodities measured in MEu are inventory and supplies.
- **MbMu (Mu)** – Metabolism-based mass units. Used to measure fiber in food. Positive real number. Commodities measured in Mu are recycled mass and waste mass.
- **MbEu (Eu)** – Metabolism-based energy units. Used to measure energy in food, and energy within the bodies of Frmrs and Wrkrs. Positive real number. Commodities measured in Eu are Frmr's energy and Wrkr's energy.
- **Dollars ($)** – Financial units. Used to measure cash, IValue, MValue, and KValue (see below). Real number, usually positive, but the MMgr is allowed to hold negative cash (go into debt) as a form of economic support.

--- FOCUS ON CURRENT DATA ---
Units: Metabolism-based MEus, Mus, Eus and $

FRMR AGGREGATES -	Units	IValue	MValue
Cash $:	na	na	39559.0
Energy Eus:	858.0	6864.0	7019.3
Recycled Mus:	1548.0	3096.0	3294.3
Inventory MEus:	2238.3	22383.0	22403.8
Supply MEus:	2448.0	24480.0	24479.4
Waste Mus:	32.0	64.0	62.6
Totals:	na	56887.0	96818.4

WRKR AGGREGATES -			
Cash $:	na	na	21481.5
Energy Eus:	1594.0	12752.0	13093.2
Supply MEus:	1301.7	13017.0	13136.2
Waste Mus:	136.0	272.0	281.5
Totals:	na	26041.0	47992.5

MMGR AGGREGATES -			
Cash $:	na	na	22799.5
Mass Mus:	736.0	1472.0	1441.4
Totals:	na	1472.0	24240.9

EMGR GRANT AGGREGATES -			
Cash $:	na	na	-0.0
Gt Mass Mus:	0.0	0.0	0.0
Gt Energy Eus:	0.0	0.0	0.0
Totals:	na	0.0	0.0

TOWNSHIP OVERALL AGGREGATES -	Units	IValue	MValue
Cash $:	na	na	83840.0
Energy Eus:	8440.0	67520.0	68219.8
Mass Mus:	8440.0	16880.0	16991.9
Totals:	na	84400.0	169051.7

This is one of the several kinds of balance sheet available in real time, tick-by-tick, as the economy develops.

Intrinsic value (IValue) – All goods have intrinsic value, measured in dollars. This intrinsic value is arbitrarily determined when the economy is first established, and it is based on the metabolic needs of the agents (for energy and mass, as found in food). When an agent does a day's work, they consume a day's worth of goods containing a day's ration of mass and energy. The pay received for that work determines the value of the food consumed for the performance of the work. The IValue does not change over time. Similar to the base year price of a price index, it marks the dollar value of goods at time zero.

Monetary value (MValue) – All goods have monetary value, measured in dollars. This is the weighted average cost of the goods held in a single store (see "stores" below). Goods are maintained in stores, and when a good is added to a store, or removed from it, pro-rated MValues are also added or removed. A momentary unit value can be assigned to any store by dividing the MValue by the quantity of the contained goods. This is used as the basis of monetary price quotes. The MValue of an almost empty store is volatile, but the MValue of an amply-supplied store is quite stable, each successful quote having only a slight effect on overall weighted average value. For more information on the production and use of price quotes, see the section below.

Market value (KValue) – Similar to monetary value, but different. All goods have market value, measured in dollars. This is the value of the contents of the store, when evaluated using the most recently negotiated unit price that was based on a price quote derived from this store. The KValue of a store is very volatile, changing completely with each successful quote.

Intrinsic unit price – Determined by dividing the IValue of goods held within a single store by the quantity. This unit price is maintained in the store, and used as the base value for producing 'intrinsic' price quotes. This unit price is fixed on initiation and does not ever change. This models the concept of the equivalence of price and value.

Monetary unit price – Determined by dividing the MValue of goods held within a single store by the quantity. This unit price also is maintained in the store, and used as the base value for producing 'monetary' price quotes. This price changes slowly over time. This models the concept of fair return on investment, as selling price is based on costs of acquisition.

Market unit price – Determined as the most recent price quote used successfully in a related transaction. The Market price is inserted into the store of the selling agent and also the store of the buying agent, and the goods held in both stores have a change of KValue of the store on insertion. This is used as the base value for producing 'market' price quotes. This unit price models the concept of opportunistic pricing, as one lucky deal with outlier prices leads to follow-on attempts for similarly extreme prices.

Stores – Agents keep quantities of goods in "stores". The word "goods" here means mass (measured in Mu), energy (measured in Eu) or food (measured in MEu). Stores are logical structures which record quantities of the goods as they come in and leave, together with the amount of IValue, MValue and KValue associated with the goods, and the associated unit price. Before a transaction, the appropriate unit price is used as a base price (see below) in the preparation of a quote. During a successful transaction, amounts of goods, values and unit prices are adjusted in both the supplying and receiving stores.

MassEnergy Stores – Whole food is kept in a special kind of store in which the value of the mass and energy is accounted both as a combined good, and as separate goods. The MValue of the energy is the weighted average of the MValue of all incoming/outgoing energy, as if this was an independent energy store. It's the same for mass. I am not 100% certain that this enables the independent pricing of mass and energy in the system, but I think it does.

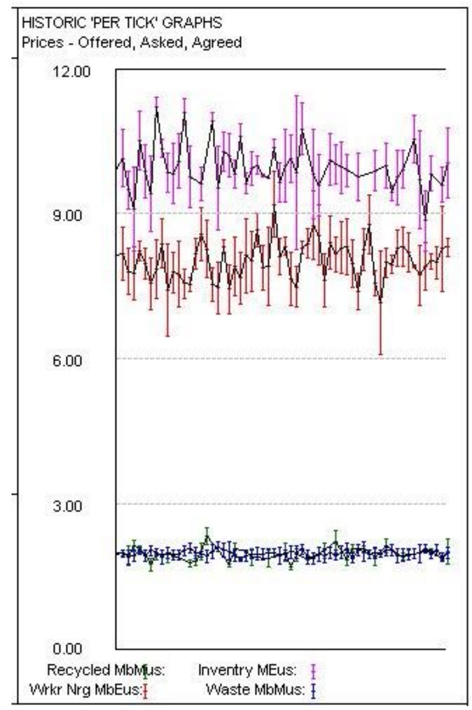
Commuting area – The set of 25 lots immediately surrounding the abode of a Frmr or Wrkr is its commuting area. A Frmr may only hire Wrkrs within its commuting area, using a device euphemistically called the Frmrs union list. A Wrkr, similarly, may only work on farms within its commuting area, using its own union list. Consumers may only purchase supplies from Frmrs each within their own commuting area, using supply lists, and Frmrs may only sell inventory to consumers each within their own commuting area using customer lists. The union lists, supplier lists, and customer lists are, in this implementation, associated with a lot, and do not move when agents move, one of each type of list per lot. This technique provides a 2-dimensional space effect in the economy. Immortal agents are not restricted to this kind of locality, do not occupy any lot, and are able to deal with any mortal agent at any time.

Quote – A quote is a part of price negotiations. Both buyer and seller issue quotes, then a price is agreed upon or the deal is cancelled. An agent constructs a quote U by consulting the relevant store to find a "negotiation base unit price", and then multiplying that by a strategic factor obtained from a price gene. $U = b(\mu+x\sigma)$

3.2 Overview of Economic Negotiations

In its most simple terms a transaction occurs when agents meet, negotiate a price, determine a quantity, and exchange cash for goods or services. Each of these four steps entails a lot of detail, as follows:

- **Agents meet** – Each mortal agent can only make deals with other mortal agents within its commuting area. They can however, make deals with immortal agents, when needed, unrestricted by commuting area. The MMgr and EMgr do not have any locality, and can make deals with any mortal agent when needed. To avoid systematic bias, the order in which initiating agents can make a deal is randomized, and then these agents are randomly paired with potential responding agents. For example, in the "Sell Inventory" phase of a standard tick, the order of the list of all Wrkrs is first randomized. Then, if a Wrkr (as a consumer) needs to purchase supplies from a Frmr's inventory, a list is made of all Frmrs in the Wrkr's commuting area having such goods. The list of potential suppliers is put into random order. Then, one-by-one, the Wrkr attempts to make a deal until one is made, or until the list is exhausted.
- **Agents negotiate a price** – In each transaction, there is a potential seller, and a potential buyer. Simultaneously, the buyer and seller issue a quote. If the seller is asking more than the buyer is bidding, there is no deal because both are unhappy, and the initiating agent moves on to the next potential deal. But, if the seller is asking less than the buyer is bidding, then a deal is struck, the agreed price is the average of the bid and ask price. Both have struck a beneficial deal, and there is happiness all around. The means by which a quote is produced by an agent is discussed in a separate section below (see below).



A real-time chart showing average values for offered, asked, and agreed prices for four types of transaction.

- **Agents determine a quantity** – All transactions have an upper limit, or quota, on the quantity of a good or service. Under normal circumstances (e.g. between two wealthy and well-supplied agents) the quantity exchanged is at full quota. However, the seller may have less than quota to sell. Or, the buyer may have less cash than needed to purchase quota at the agreed price. The transaction proceeds at the agreed price, but the quantity is reduced, and less cash paid. However, this is complicated by the action of the EMgr, one of the immortal agents. In his role of distributing estate assets of dead agents to “deserving but needy agents”, the EMgr can top up the assets of any agent to address under-quota-deficiencies, but only once a price is agreed upon. In some cases, this gets VERY complicated. But the outcome is always the same. Some quantity of the good is exchanged for some cash such that the agreed-upon price is respected, and the quantity is less than or equal to the appropriate quota.

- **Agents exchange cash for goods** – The amount of cash is extracted from the cash store of the buyer and injected into the cash store of the seller. The quantity of goods is extracted from the appropriate store of the seller (together with its associated IValue and MValue), the MValue of the transferred goods is adjusted using the agreed upon price, and the quantity is injected into the store of the buyer (together with associated IValue and adjusted MValue). The details of the extraction and injection processes are in a separate section outlined below. The KValue of both stores is then adjusted using the agreed upon price.

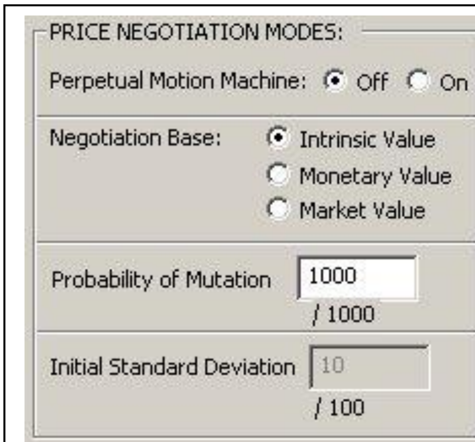
3.3 Extraction and Injection of Goods

3.3.1 Localism

The stores contain an agent’s memory of past successful economic transactions. They do not contain any memory of failed transactions. This memory is available to them in the form of base unit prices that they get by querying an appropriate store, of their own, when preparing a price quote.

The concept behind this process is limited knowledge of agents, sometimes called localism. In the type of model called a cellular automaton, local rules cause the emergence of global system behaviours and characteristics, and I am trying to implement a similar concept in ModEco. I do not believe that any economic actors in a real-world economy have perfect knowledge of the market, nor do I think that most actors always make decisions in accordance with the limited knowledge they may possess. Localism has at least three dimensions within ModEco. In ModEco, agents:

- have imperfect spatial knowledge and effect (due to the limitations of the commuting areas), being unable to conduct commerce with distant agents;
- have limited market knowledge, being able to query base prices from their own stores, but unable to query the base prices of other agents; and
- often make sub-optimal decisions based on the knowledge they do have, being subject to random variations in their quotes, even when knowing the base prices of their own stores.



The user can choose one of three options for extracting a base price from a store, based on intrinsic value, monetary value, or market value.

3.3.2 Trans-generational memory

So, in the spirit of localism, an agent has very limited sensing ability and memory. It knows the price and quantity for this transaction, and the value and quantity of its assets, and that’s all. It knows nothing about prices paid by other agents in transactions in which it was not involved, nor anything about trading strategies of other agents. An agent’s trans-generational memory, and the basis of its pricing strategy, is stored in its pricing genes (see more about pricing genes below) that allowed its parent to reproduce successfully, and in the contents of the stores passed on from parent to child.

3.3.3 Pricing modes

There is a user-selectable toggle in the initiation wizard dialog (IWiz button), and in the economic wizard dialog (EWiz button), that enables a user to choose the base price for a quote. The user can choose between the intrinsic, monetary, or market unit price as the basis for preparation of a quote. These three different pricing modes were each implemented, in this "most simple" model, in a search for a realistic model of price negotiations that would lead to sustainability. When I find one that works, the other two may be eliminated from the application. Maybe. For now, there are three.

3.3.4 Some Variable Definitions

A store will contain a quantity (Q) of one of three types of goods: mass, energy, or mass-energy. The type doesn't matter because all are treated the same way. In three out of four types of economic transaction, two stores of identical type are involved: one for the seller and one for the buyer. This is true for buy/sell waste mass, buy/sell recycled mass, and buy/sell inventory. It is not true for hire worker, in which transaction the energy of the Wrkr is effectively incorporated into the food placed in the Frmr's inventory.

While the mathematics of the process of transfer of goods from store to store is simple, there is a wonderful multiplicity of intermediate variables that is confusing. Each variable is introduced with a let statement, but, here is a summary of the variables:

- P_S, P_A, P_B – unit prices – intrinsic, seller, agreed, buyer
- Q_S, Q_T, Q_B – quantities – seller, transferred, buyer
- I_S, I_T, I_B – intrinsic values – seller, transferred, buyer
- M_S, M_T, M_B – monetary values – seller, transferred, buyer
- K_S, K_T, K_B – market values – seller, transferred, buyer

In addition, quantities, intrinsic values, monetary values, and market values change during the transaction, so I use the left subscripts 1 and 2 to indicate before and after, respectively, when needed.

Finally, I use the symbols S_S and S_B to represent the stores of seller and buyer, respectively, involved in the transaction.

3.3.5 Computational Process

Skip this section unless you want the detail. This section is a bit tedious, but the math is simple and the calculations rational. I provide the tedious detail in simple mathematical equations just to be complete, and to be exact. Let Q_S be the quantity in the seller's store S_S, and let Q_B be the quantity in the buyer's store S_B.

Calculation of changed quantities: Let Q_T be the quantity that is transferred from S_S to S_B. So, after the transaction, the quantity that remains in the store S_S is ${}_2Q_S = {}_1Q_S - Q_T$. Similarly, the quantity that is now in the buyer's store is ${}_2Q_B = {}_1Q_B + Q_T$.

Calculation of changed IValues: The intrinsic unit price of the good is determined arbitrarily and hard-coded into the program. It is based on metabolic needs. Let P_I be the intrinsic unit price. The IValue of the contents of S_S, before and after the transaction, can be calculated as ${}_1I_S$

$= P_I \times {}_1Q_S$ and ${}_2I_S = P_I \times ({}_1Q_S - Q_T)$ respectively. Similarly, the IValue of the contents of S_B, before and after the transaction, can be calculated as ${}_1I_B = P_I \times {}_1Q_B$ and ${}_2I_B = P_I \times ({}_1Q_B + Q_T)$ respectively.

Calculation of changed MValues: All goods are loaded into a temporary store called a "bill of lading" when extracted from S_S, and then placed into S_B from the bill. Temporary structural changes, or changes in the prices and values, if needed, are handled in the bill of lading as the transaction progresses. This is an implementation detail which should not have any effect on the pricing issues being examined in this paper.

- Let ${}_1M_S$ and ${}_2M_S$ be the monetary value (MValue) of the contents of store S_S before and after the transaction, respectively. Let ${}_1M_B$ and ${}_2M_B$ be the MValue of the contents of store S_B before and after the transaction, respectively. Let M_T be the monetary value of the quantity Q_T being transferred. Finally, let P_A be the agreed-upon price.
- When Q_T is extracted from S_S, the value of the remaining contents of the store is simply pro-rated downwards. ${}_2M_S = {}_1M_S \times ({}_1Q_S - Q_T)/{}_1Q_S$.
- But M_T, the monetary value of Q_T, changes. It is determined by the agreed price associated with this transaction. $M_T = P_A \times Q_T$.
- Then the new monetary value of the buyers store is ${}_2M_B = {}_1M_B + M_T$.

Calculation of changed unit prices: Some unit prices are adjusted in each store during the transaction, as follows:

- Let the intrinsic unit prices to not change during any transaction. They are fixed.
- Monetary unit prices are computed as weighted averages for the contents of the store:
  - Let ${}_1P_S$ and ${}_2P_S$ be the monetary unit prices of the contents of store S_S before and after the transaction, respectively.
  - Let ${}_1M_B$ and ${}_2M_B$ be the MValue of the contents of store S_B before and after the transaction, respectively.
  - Let ${}_1Q_B$ and ${}_2Q_B$ be the quantity of the contents of store S_B before and after the transaction, respectively.
  - Then ${}_1P_S = {}_1M_S / {}_1Q_S$, and ${}_2P_S = {}_2M_S / {}_2Q_S$.
  - Similarly ${}_1P_B = {}_1M_B / {}_1Q_B$, and ${}_2P_B = {}_2M_B / {}_2Q_B$.
- The market unit price is simply the agreed-upon price that resulted from the negotiations. It is the equivalent of a spot price. This is inserted into S_S and S_B during the transaction.

Calculation of changed KValues: The market value of each store is altered during the transaction using the spot price: as follows:

- Let ${}_1P_S$ and ${}_2P_S$ be the market unit prices of the contents of store S_S before and after the transaction, respectively.
- Let ${}_1K_B$ and ${}_2K_B$ be the KValue of the contents of store S_B before and after the transaction, respectively.
  - Let ${}_1Q_B$ and ${}_2Q_B$ be the quantity of the contents of store S_B before and after the transaction, respectively.
- Then ${}_1K_S = {}_1Q_S * {}_1P_S$, and ${}_2K_S = {}_2Q_S * {}_2P_S$.
- Similarly ${}_1K_B = {}_1Q_B * {}_1P_B$, and ${}_2K_B = {}_2Q_B * {}_2P_B$.

Calculation of agreed unit price: At any time, an agent can check a store to find base price information. A price quote is fabricated by the agent using the base price as a starting point, and then modifying it using the random number generator to select a value drawn from a normal distribution with mean mu and standard deviation sigma. The quote production method is described in more detail below, but here is are the calculations at a high level:

- Let ${}_1P_S$ and ${}_1P_B$ be the base prices of the seller and buyer respectively, prior to the transaction.
- Let $N(\mu_S, \sigma_S)$ and $N(\mu_B, \sigma_B)$ be two random variates, where m and s are the strengths of the appropriate pricing genes of the seller and buyer respectively.
- Let ${}_QP_S$ and ${}_QP_B$ be the unit prices produced as quotes by the seller and buyer respectively.
- Then ${}_QP_S = {}_1P_S * N(\mu_S, \sigma_S)$, and ${}_QP_B = {}_1P_B * N(\mu_B, \sigma_B)$.
- If P_A is the agreed-upon unit price, the $P_A = ({}_QP_S + {}_QP_B) / 2$.

3.3.6 Encoded memory of agents

Although a base price is always based on information the agent has access to on a momentary basis (like, what's in my wallet?), these three valuation techniques in fact incorporate three different kinds of memory or learning:

- Intrinsic value implies permanent memory of universal truth, infinitely accurate, and accessible to all. There is a sense in which this approaches the ideal of a perfect market.
- Monetary value incorporates a less perfect memory. But this memory also passes across the boundary of fission, as the daughter's value their stores according to the MValues passed on by their mothers, and so the memory is trans-generational. The MValue contained in a store is a kind of weighted average derived from all previous transactions of all forebears, with the most recent transactions, and largest transactions, having a larger weight. For wealthy agents, the memory is deep and intergenerational, but for poorer agents is less deep, with less impact across the fission/rebirth boundary.
- Market value contains only immediate memory, based on the most recent spot price.

3.4 Gene-Mediated Pricing Mechanism

3.4.1 Construction of a Quote

When an agent wants to make a quote for a unit price (ask or bid) it does two things:

- First, it establishes a base price, which is retrieved from its appropriate store. This is based on the one of the three user-selectable modes of pricing. For example, if a Wrkr is buying inventory from a Frmr to put into its supply store, the Frmr will get a base price from its inventory store, and the Wrkr will get a base price from its supply store. The price may be P = IValue/Q (intrinsic price), P = MValue/Q (monetary price), or P = KValue/Q (market price).
- Second, it determines a pricing strategy by consulting the unique personal "price gene" associated with the type of transaction under way. This provides it with a multiplier for the base price in the form of a normally distributed variable N(μ, σ), where μ and σ are values drawn from the price gene.

Both seller and buyer generate their unit prices using a value for μ (mu; mean unit price) and σ (sigma; standard deviation of unit price) and the formula:

$$U = b(\mu + x\sigma)$$

where U is the unit price, b is the base price per unit, and x is a randomly generated variate with mean of zero and standard deviation of one. The random variate 'x' is generated using the Box-Muller formula:

$$x = \sqrt{-2 \bullet \ln(x_1)} \bullet \sin(2\pi \bullet x_2)$$

where x_1 and x_2 are random variates with a uniform distribution (as is produced by most computer-based pseudo-random number generators). The complete equation is then

$$U = b\left(\mu + \left\lfloor \sqrt{-2 \bullet \ln(x_1)} \bullet \sin(2\pi \bullet x_2) \right\rfloor \sigma\right).$$

3.4.2 Uses of a quote

When two agents negotiate a price for a commodity, they each consult their appropriate price genes, each produce a "quote", and then, if the quotes are acceptable, settle on a price. Context or prior dealings (except as noted above) are not considered in the preparation of a quote. Gene-mediated prices proceed as follows:

- Each time a materiel unit (Mu, Eu, or MEu) changes hands, the buyer and seller negotiate a price.
- The buyer tries to minimize the price.
- The seller tries to maximize the price.
- An **offer to sell** is called a quote (or an **ask**) and consists of:
  - Type of goods;
  - The basis of the quote (IValue, MValue or KValue);
  - Minimum acceptable price per unit (a unit price).
- An **offer to buy** is called a quote (or an **offer**) and consists of:
  - Type of goods;
  - The same basis of the quote (IValue, MValue or KValue);
  - Maximum acceptable price per unit (a unit price).
- Each agent has a specific price gene for (a) that type of goods, and (b) that type of transaction (buying or selling). The buyer consults its appropriate "buy" gene for that type of goods. The seller consults its appropriate sell gene for that type of goods. Each produces its quote independently of the other.
- If the seller's asking price is less than the buyer's offer, then both will be happy. The agreed-upon price will be the average of the two. However, if the seller's asking price is more than the buyer is willing to pay, the negotiations end, and there is no deal.
- Fifty percent of every generation will fail to procreate. Agents with bad price genes and/or bad base prices will fail to close deals as often as others, will languish, and ultimately will fail to establish a line of descendants.
- On the other hand, agents with good price genes will close advantageous deals more often, will flourish, and their offspring that inherit their genes and wealth will also flourish.
- Over time, the buyers and sellers compete for access to resources, and the best-adapted genes will probably prevail, giving their owners more successful transactions, and greater success in life. The least-well adapted genes will probably fail to complete sufficient transactions, or will result in bad (money-losing) transactions, and the owners will fail to thrive, and die of starvation or old age.
- All grants provided by the EMgr are free of charge, but can only be given as part of a successful gene-mediated transaction. This may dilute the evolutionary pressure on the

genes (as a bad price results in EMgr-provided benefits that counteract the bad price), but failure to negotiate a price, at least, is not rewarded.

3.4.3 Genetic Controls – Deeper into the Price Genes

The values of μ (mu) and σ (sigma) for buying and selling are determined via the price gene. There are eight numbers stored in a price gene as shown in the table at right.

Each agent should have one price gene for each type of transaction in which it is regularly involved, such as buying supply MEu or selling waste Mu. More explicitly, agents have the following genes:

Frmrs:

- Buy Recycled (Mu) gene
- Buy Labour (Eu) gene
- Sell Inventory (MEu) gene
- Buy Supply (MEu) gene
- Sell Waste (Mu) gene

Wrkrs:

- Sell Labour (Eu) gene
- Buy Supply (Mu) gene
- Sell Waste (Mu) gene

MMgr

- Sell Recycled Mass (Mu) gene *
- Buy Waste Mass (Mu) gene*

1.1.1 Table of Price Gene Components

Purpose	Initial Value
Mu – numerator (N >= 0, N an integer)	20
Mu – denominator (D > 0, D an integer)	20
Mu – delta ($\delta > 0$, δ an integer)	1
Mu – Strength = numerator / denominator (S > 0, S a real number)	1.00
Sigma – numerator (N >= 0, N an integer)	10
Sigma – denominator (D > 0, D an integer)	100
Sigma – delta ($\delta > 0$, δ an integer)	1
Sigma – Strength = numerator / denominator (S > 0, S a real number)	0.10

Pricing strategy is encoded in a price gene. Agents consult a price gene to formulate a quote. A quote is a unit price which is normally distributed about a strategically modified base price. When an agent consults a price gene to produce a quote, the strength of Mu is used as a multiplier of the base unit price to determine a strategic but average quote, which is then modified stochastically using the strength of Sigma. Evolutionary pressures should therefore select strongly on the strengths of Mu and Sigma, and less so on the other components.

* Note that the MMgr does not (cannot) evolve because it is immortal and non-reproducing, but nevertheless plays a central role in the economy. I have therefore decided to allow its genes to change in a manner that does not bias the evolutionary pressure on the price genes. The genes of the MMgr have the average values of all of the genes of the agents with which it must deal. So, the “Sell Recycled Mu” genes of the MMgr are the average of the “Buy Recycled Mu” genes of the Frmrs. The “Buy Waste Mu” genes of the MMgr are the average of the “Sell Waste Mu” genes of all consumers, which means all Frmrs and all Wrkrs.

You could say that the MMgr participates in the economy “at the going rate”.

3.4.4 Mutation of Genes

A few words need to be said about the role of the eight components of a price gene and the effects of mutation.

A price gene has two main components, the mean (μ) and the standard deviation (σ), used to generate a random normal variate $x = N(\mu, \sigma)$ via a formula called the “Box-Muller Formula”. Each of μ and σ has the ability to mutate independently of the other. Suppose the “probability of mutation" in the "ModEco Control Panel" is set to 0.125, then upon production of a new daughter during fission, each price gene in each agents has a 0.125 probability of mutation. Since there are six components within the gene that can mutate with equal probability, each component has a further probability of $1/6^{th}$ to mutate.

Each of these main components μ and σ is computed as a positive ratio called the strength. So the "strength" of μ is actually the number used in the Box-Muller formula, and the "strength" of σ is similarly used. These strengths are computed by dividing the correct numerator by the correct denominator.

When it has been decided that one of the six components should change, then:

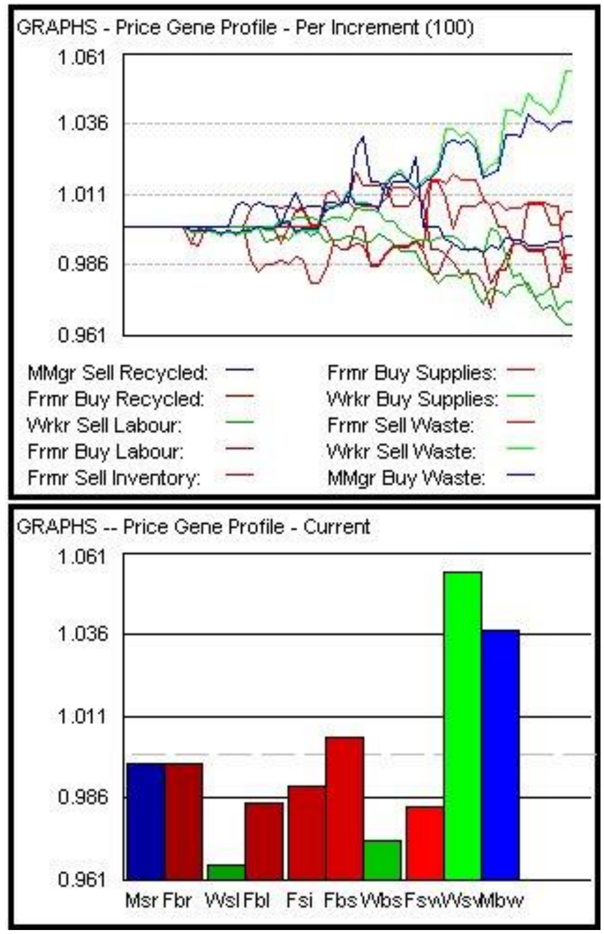
- a numerator changes by the appropriate delta; or
- a denominator changes by the appropriate delta; or
- one of the deltas changes up or down by 1;
- with the condition that the numerator and denominator must remain positive integers;

This particular scheme allows the evolutionary pressures (through selection on phenotypic characteristics) to indirectly explore the space of rational numbers near the assigned initial strength. I say "indirectly" because, the choice of delta is barely phenotypic. If a gene has a large positive delta, then the gene strength will change quickly from generation to generation. Evolutionary pressure will select for larger deltas if and only if rapid change in a gene's strength consistently confers evolutionary advantage, and the favorable mutation in the value of delta is coupled with favorable mutations in other components of the gene dependent on the delta. So, in an application such as PSoup in which sustainability was easily achieved, and in which a run could last millions of ticks (thousands of generations, and thousands of mutations) this scheme enabled a very detailed exploration of values in a gene's phase space. On the other hand, here in ModEco, where a run might collapse within 10 or 20 thousand ticks (10 or 20 generations) the variations in delta would be so subtly phenotypic that there would be little chance for selection to happen.

This issue is explored more completely in the section on Probability of Mutation, below.

I view evolution of a price gene in the following way. We start with two initial ratios, say μ = 10/10 = 1; and σ = 10/100 = 0.10. Call this the initial value (μ_O, σ_O). I assume there is an ideal or optimum price gene (μ_N, σ_N). Any change which makes the agent more competitive (better adapted) will survive and replicate itself. The history of a successful price gene is a series of pairs of ratios which progressively approach the ideal. For this to happen the denominators must be able to get large (providing a more finely grained grid of possible values) and the speed of change needs to adjust appropriately.

The deltas do not play any direct role in the economy, but, rather, govern the rate at which the means and standard deviations can change across generations. In a very long run, the speed at which a gene races towards its ideal value will determine the winner of the race. The delta is the accelerator. Good mutations of the delta will therefore play a large role in determining the nature of the competitive landscape in the mid-term, and in the long run.



Real time output: Top – Historic evolution of average μ, all price genes. Bottom – Snapshot of current values of average μ, all price genes.

The deltas are therefore indirectly phenotypic, as a change will affect the rate of mutation of genes over generations, but will not be acted upon by the environment in this generation. The deltas should increase or decrease by 1 each time they mutate. This means the rate of change of the mu and sigma values is under evolutionary pressure. A good long-term mutation consists of a change in the mu or sigma in the right direction, combined with a change in the associated delta in the right direction, followed, a generation later, by another change in the mu or sigma in the right direction.

Let's look at a few numbers. A Wrkr has three price genes. A Frmr has five price genes. Let's focus on the Wrkr. When a Wrkr goes through fission two daughters are formed. Six new genes are formed and six opportunities for mutation occur (recall each gene is independent). If "p" is the probability of a good mutation (good ratio change or good delta change), then the probability that exactly n good mutations will occur is $C(6, n)\ p^{n}(1-p)^{6-n}$. A reasonable value for "p" is 0.5 times the probability of a mutation happening. Using this formula, P(at least one good mutation happens) is the same as 1 – P(no mutations) $\approx$ 0.32. This looks good, but a good mutation might be undercut by a simultaneous bad mutation. To get a good look at the real probabilities of successful adaptation, we will need to work out a more detailed analysis. This would be useful in finding a formula for the expected speed of evolution towards the ideal gene.

For Wrkrs, n = 3 and k varies from 0 to 3. For Frmrs, n = 5 and k varies from 0 to 5. If we assume that p = 0.125, the default value for the parameter Probability of Mutation, then we can generate a table of probabilities, using the above formula, as follows:

Frmrs, n = 5		Wrkrs, n = 3	
k =	**P =**	**k =**	**P =**
0	0.51291	0	0.66992
1	0.36636	1	0.28711
2	0.10468	2	0.04102
3	0.01495	3	0.00195
4	0.00107		
5	0.00003		

The sum of the probabilities for each type of agent is, of course, equal to 1.

3.4.5 Selection Exposure of Genes

Regular mutations in the genes provide the phenotypic variation in a population of agents that Darwinian selection acts upon. Approximately 50% of every generation must fail to reproduce. Those that fail to reproduce suffer that fate because they have failed to make sufficiently profitable deals associated with one or more of their price genes. So, relatively advantageous genes are good, but the ability of the environment to detect and reward small advantages is dependent on the relative amount of exposure the gene has. An advantageous gene, or a bad gene, with low exposure is of little effect. Random fluctuations mask the relative strengths unless there is high exposure. A mild advantage with high exposure can effectively be rewarded. The level of exposure is similar to the level of resolution of a microscope – with high exposure things are less fuzzy and clearer. The mechanism that gives exposure this quality is the law of large numbers.

```
WORKER - GENETIC MAKEUP
CHROMOSOME 1 (C1) - PRICING GENES:
                           |  BUYING GENES ----------- | SELLING GENES ------------|
                           |  Mean Gene  | StdDev Gene |  Mean Gene  | StdDev Gene |
                  Labour:  |        n/a  |        n/a  |    0.95000  |    0.10000  |
               Sup/Waste:  |    0.84211  |    0.09901  |    1.00000  |    0.10000  |


                           |    Strength |  Numerator  | Denominator |       Delta |
    Sell Labour - Mean:    |    0.95000  |   19.00000  |   20.00000  |   -1.00000  |
                  SDev:    |    0.10000  |   10.00000  |  100.00000  |    1.00000  |

     Buy Supply - Mean:    |    0.84211  |   16.00000  |   19.00000  |   -2.00000  |
                  SDev:    |    0.09901  |   10.00000  |  101.00000  |    1.00000  |

     Sell Waste - Mean:    |    1.00000  |   19.00000  |   19.00000  |   -2.00000  |
                  SDev:    |    0.10000  |   10.00000  |  100.00000  |   -1.00000  |
```

At any time, a user can examine the contents of the price genes for any agent. In this case, a Wrkr exhibits evidence of at least seven mutations of delta values, and an indeterminate number of more strongly phenotypic mutations to the μ and σ values.

If a coin has a 0.503 probability of landing heads, and you flip it ten times counting the heads, you will not be able to detect the small advantage this coin offers in a flip. But if you flip it a million times, the advantage becomes tangible, measurable, and effective for winning games. The number of times the unfair coin is exercised in a game of chance is its exposure.

Most agents cannot reproduce until they are 800 ticks of age (Age Reproduction Threshold = ART = 800 by default). On average, an agent gets cash for 20 MEu for a day's work, and

Gene	Symbol	Average Required Successes Per 5 Ticks	Expected Exposures Per 5 Ticks	Expected Exposures per Generation (800 Ticks)	Counter Gene
Frmr: Buy Recycled (Mu) gene	F:Br	1	2	320	M:Sr
Frmr: Buy Labor (Eu) gene	F:Bl	4	8	1,280	S:Sl
Frmr: Sell Inventory (MEu) gene	F:Si	4	8	1,280	F:Bs W:Bs
Frmr: Buy Supply (MEu) gene	F:Bs	1	2	320	F:Si
Frmr: Sell Waste (Mu) gene	F:Sw	1	2	320	M:Bw
Wrkr: Sell Labor (Eu) gene	W:Sl	1	2	320	F:Bl
Wrkr: Buy Supply (Mu) gene	W:Bs	1	2	320	F:Si
Wrkr: Sell Waste (Mu) gene	W:Sw	1	2	320	M:Bw
MMgr: Sell Recycled (Mu) gene	M:Sr	$10N_F$	$20N_F$	$1600N_F$	F:Br
MMgr: Buy Waste (Mu) gene	M:Bw	$10(N_F+N_W)$	$20(N_F+N_W)$	$1600(N_F+N_W)$	W:Sw F:Sw

N_F is the number of Frmrs; N_W is the number of Wrkrs.

consumes 4 MEu, so they must work once every 5 ticks to break even. But, on average, a quote fails to complete a deal half of the time, so, to survive, an agent must exercise the gene twice as often, on average. This implies that each Wrkr's price gene for job seeking is exercised at least twice per five ticks. By "exercised" I mean used to prepare a quote. For producers (Frmrs) the genes for negotiating wages and selling inventory are exercised four times as often, so that would be 4 successes per 5 ticks, or 8 exercises, on average, per five ticks. For each class of price gene we can draw up a table of expected number of times it is exercised. Each gene class can be represented by a symbol, as shown in the following table. Each gene is exercised in competition with an opposite. For example, when a Wrkr buys supplies from a Frmr, the Frmr uses its "Sell Inventory" price gene, and the Wrkr uses its "Buy Supply" price gene, each to prepare a quote.

The default values of the ModEco parameters are designed to work this way. How do we interpret this? Suppose we consider the W:Sl class of genes. A Wrkr inherits one when it is born. The Wrkr must get work, on average, once per five ticks, to earn enough cash to buy supplies to eat. If on any tick, it has less than 4 MEu of supplies, it dies of starvation. There must be an average influx of 4 MEu per tick. A day's work will buy 20 MEU. Those with good price genes will get work regularly on the first attempt. On average, it will require two attempts (due to the expected 50% "no deal" outcomes). A Wrkr only attempts to purchase supplies when its store of supplies is below threshold. For those with bad price genes, they may try to get work many times per five ticks. But, we know that, on average, there will be 2 exposures, per W:Sl gene, per 5 ticks.

Each such exposure presents an opportunity for the environment (in the form of the Frmrs' F:Si genes of those Frmrs within commuting distance) to reward or punish the agent. So, the relative strength of the W:Sl, when compared to the strength of the F:Si is the phenotype character on which evolution bases its selection. 800 ticks was chosen as the default value for ART because I assume that 320 exposures is a sufficiently high number of exposures to allow the law of large numbers to do its work and reward good genes with rights to reproduce themselves.

4 Implications for Stability

My going-in assumption is that a sustainable economy must be based on stable prices. Stable prices must then be characteristic of an attractor of some sort in the state space of the economy. The above-described approach to pricing often leads to inflation or deflation, and eventual collapse of the economy. I interpret that as there being a probability gradient in the state space that biases movement towards collapse as a final steady state.

A possible solution is revaluation of money from time-to-time, triggered by excessive variance of MValues from IValues. Such an approach is effectively lifting the economy out of the state space and plunking it down into a preferred location. However, that would somehow breach the concept of localism, by which a complex system exhibits macro-level characteristics from local rules. In other words, rather than being intrinsically sustainable, a Big Brother process would watch prices, and rejig them from time to time to make the economy sustainable. Such an approach would seem to me to be in breach of my search for the dynamics of a sustainable free-market economy. A better solution would be to design a negative feedback mechanism such that prices find a pseudo-equilibrium (a stationary state?) and stay there, without superior-level. In other words, alter the probability landscape of the state space.

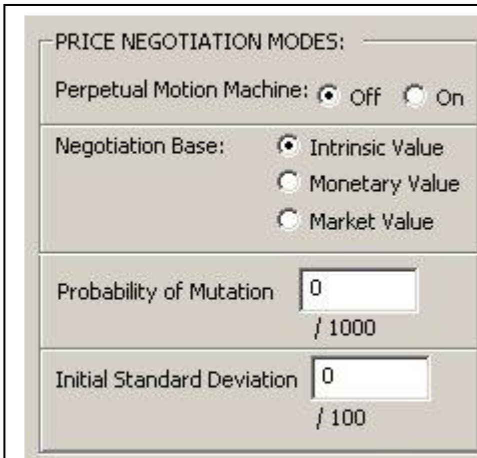
The primary questions to be explored at this point are:

- Is inflation or deflation actually the true cause of collapse?
- Why does this pricing schema lead to inflation or deflation?
- And, how can I design a negative feedback mechanism into the system to stabilize prices at some level, controlling inflation or deflation?

4.1 Other Possible Causes of Instability

I believe I have ruled out other causes through the introduction of a variety of things such as:

- Business factors - preventing overconcentration of wealth within an agent;
- Quotas, reproduction by fission - preventing overconcentration of wealth within one agent, at least, in the short term;
- MMgr – preventing spatial concentration of wealth, by distributing mass equally to all farmers;
- MMgr debt – preventing collapse due to shortage of money in the economy as a whole;
- EMgr – distributing the wealth of dead agents somewhat equally to "needy but deserving agents";
- Discreteness and liquidity, when there are too few agents.



Enforced stable pricing: base price is the intrinsic unit price; μ is fixed at 1.00; σ is fixed at 0.00.

4.2 Price Instability

Price instability can be totally eliminated. If the user selects a negotiation base price based on intrinsic values, sets the probability of mutation of the genes to zero, and sets the initial standard deviation (σ) to zero, then the base unit price in all transactions will be the intrinsic unit price. In the figure to the right, you see those settings from one of the tabs of the IWiz. Note that the "Perpetual Motion Machine" (PMM) switch is shown as off. The three other settings are exactly the settings that would be applied when the PMM switch is turned on. I.e. these are the settings that produce a sustainable economy that has been run as high as 20 million ticks without collapse. There is a great deal of machinery working behind the scenes to ensure that the PMM stays on the knife-edge of the equilibrium point (it's actually some kind of stationary state, but, OK) to be sustainable:

- All variables are maintained with 16 decimal digit precision (32 bits); this means there is often register-generated binary imprecision in the 15^{th} and 16^{th} decimal digit;
- I enforce rounding at the 14^{th} decimal digit or earlier, so I can control the nature of the imprecision that arises due to decimal digit rounding;
- I compare temporary, partial, and local values to expected values with a somewhat higher tolerance for such local errors, mostly as a technique to identify and eliminate programming errors. Every store is checked for logical consistency of all contained variables at almost every step during debug runs. This type of checking is suppressed, as is the custom of most developers, when release code is compiled;
- A baseline value for all conserved quantities (energy, mass, money, intrinsic value) is taken at the initiation of a run. Then at the end of each tick, all conserved quantities are summed, compared to the baseline, and any minor drifting of the conserved quantities is corrected. A pro-rata correction of all MValues and KValues is undertaken to ensure that IValues, MValues and KValues are consistent at the highest level, where conservation laws are enforced. Required corrections are randomly inserted back into the economy to ensure that no bias is put into any one store.

Much of this machinery was introduced into the application with the goal of making the PMM stable and truly conservative. A side effect of these efforts is that I have great confidence that the program is working as expected. I have also built in a number of real-time data access and display capabilities, as part of the process to make the PMM stable, which are of great value in monitoring the activities of the software for non-PMM economies.

The PMM has supreme price stability, due to the above machinery. From the stable configuration that typifies the PMM, there are a few ways to loosen the constraints on prices, while at the same time benefitting from the relative certainty that the economic engine is operating as intended and expected:

- The initial standard deviation can be set to a value between 0 and 1 in increments of 0.01. 10/100 is the recommended value. The base price is still the intrinsic unit price, but an agent's quotes will be a random variable normally distributed around the base price. This change to σ can only be done on initiation via the IWiz. It is suppressed in the EWiz. On initiation, all values of σ in every gene are set to this value. Because the probability of mutation is still set to zero, the base prices will never change. The monetary unit price, the market unit price, and the MValues and KValues of each store will vary, **but these will never be used**. I maintain all three pricing schemes at all times, to allow me to switch pricing modes mid-run. To be clear, the KPrice is the unit price paid/received in the last economic transaction, but the IPrice will be used as the base in the next transaction.

```
ECONOMIC INFORMATION:
 No Of Hires Max: 4; No Of Hires: 4
  Total Value:                     IValue( 28352.725223); MValue( 36370.457508); KValue( 36554.941992)
        Cash:                      IValue(     0.000000); MValue(  8349.555228); KValue(  8349.555228)

      Energy:  Eu(   833.822452); IValue(  6670.579613); MValue(  6444.088927); KValue(  6117.474909)
  Unit Prices:                  ; IPrice(     8.000000); MPrice(     7.728371); KPrice(     7.336664)
    Recycled:  Mu(   491.044050); IValue(   982.088099); MValue(  1004.161708); KValue(  1004.161708)
  Unit Prices:                  ; IPrice(     2.000000); MPrice(     2.044952); KPrice(     2.044952)
   Inventory: MEu(  1113.695708); IValue( 11136.957077); MValue( 11226.410946); KValue( 11940.547952)
  Unit Prices:                  ; IPrice(    10.000000); MPrice(    10.080322); KPrice(    10.721553)
      Supply: MEu(   953.110043); IValue(  9531.100433); MValue(  9313.693287); KValue(  9107.696559)
  Unit Prices:                  ; IPrice(    10.000000); MPrice(     9.771897); KPrice(     9.555766)
       Waste:  Mu(    16.000000); IValue(    32.000000); MValue(    32.547411); KValue(    35.505637)
  Unit Prices:                  ; IPrice(     2.000000); MPrice(     2.034213); KPrice(     2.219102)
 Date Last Supplied: (620)
```

Using the "VA" (View Agent) button, here is the data about the price and value of the stores of a Frmr. Base price is intrinsic; Probability of Mutation is zero; Initial σ is 0.10. Note that the IPrices are rock steady, but the MPrices, KPrices, MValues and KValues vary from the intrinsic values.

- OR – the Probability of Mutation can be set to a value between 0 and 1 in increments of 0.001. 0.125 is the default and recommended value. The base price is still the intrinsic unit price. However, the values of μ and σ are no longer held constant, and will vary as time goes by. σ does not cause bias in pricing, as it will cause a quote to vary equally above and below the base price. μ, on the other hand, does cause bias, and that is its purpose. μ starts with a default value of 1.00 in all genes, but mutation will cause it to rise or to fall. This is the heart of my concept of self-pricing. My expectation was that selling genes would have upwards evolutionary pressure on the value of μ, as sellers want to get more for their produce, and buying genes would have downwards evolutionary pressure on the value of μ, as buyers want to pay less for produce. I expected that those agents that suffered mutations in the wrong direction would be selected out of the population. However, the bias in pricing should happen under evolutionary pressure, over some substantial time, since initial values of delta are zero, and the first alterations in phenotype would not occur until at least the second mutation in a gene. Of course, in a sustainable economy, substantial time is what you want. But regardless of the fact that bias would take a long time to develop in the pricing genes, and would be under the control of Darwinian selection, such economies are inherently unstable in the short term. I cannot explain this.
- OR, the negotiation base price can be set to "monetary" or "market". Both settings are unstable, market prices being more volatile and more unstable than monetary prices, as one might expect.

The PMM automatically sets the probability of mutation to 0.00, the initial standard deviation to 0.00, and the negotiation base price to the intrinsic unit price, as shown in this table. The PMM scenario was run with a crowded scenario, seed = 1. It was user-terminated. The PMM runs "forever" for every seed tested so far, as long as the number of agents is sufficiently above critical mass.

S/N	Settings			Sparse Scenario	Crowded Scenario
	Prob. Of Mutation	**Initial Std Dev**	**Base Price**		
PMM On					
	Zero	Zero	Intrinsic		UT – The PMM – super stable up to 20 million ticks. All quotes are at exactly the base price (since σ is zero), and monetary or market prices do not vary from initial settings.
Legend: "C" = Collapsed; "UT" = User Terminated.					

We have, then, the following table of twelve possible user-determined selections for the pricing mechanism. This table presents the results of 24 trials. Two scenarios were run, each with twelve different settings, as shown. For all runs the random number generator was seeded with 1. The runs are labelled A through L, for reference. While a given setting may collapse or not at different times with different seeds.

Six runs (including the PMM) were user-terminated as it appeared they would run forever, although this assumption may be false. All of the rest collapsed, with the death of all agents. Note that all of the versions of the sparse scenario collapsed. I believe that this is caused by a liquidity problem. There must be enough agents participating in the economy for it to maintain a critical mass during the fluctuations of population. The goal is to have a free-market economy run sustainably. That is, to have an economy with settings H, I, K or L to be stable and not collapse (shaded settings).

S/N	Settings			Sparse Scenario	Crowded Scenario
	Prob. Of Mutation	**Initial Std Dev**	**Base Price**		
PMM Off					
A	Zero	Zero	Intrinsic	C at 505,455 ticks	UT at 1,307,000 ticks
B			Monetary	C at 505,455 ticks	UT at 910,000 ticks
C			Market	C at 50,798 ticks	UT at 422,000 ticks
D		Non-Zero 0.10	Intrinsic	C at 188,576 ticks	UT at 500,000 ticks
E			Monetary	C at 17,973 ticks	UT at 369,000 ticks
F			Market	C at 769 ticks	C at 6,799 ticks
G	Non-Zero 0.125	Zero	Intrinsic	C at 38,820 ticks	C at 26,665 ticks
H			Monetary	C at 3,623 ticks	C at 2,650 ticks
I			Market	C at 3,333 ticks	C at 1.364 ticks
J		Non-Zero 0.10	Intrinsic	C at 67,226 ticks	C at 56,244 ticks
K			Monetary	C at 3,853 ticks	C at 43,463 ticks
L			Market	C at 924 ticks	C at 3,767 ticks
Legend: "C" = Collapsed; "UT" = User Terminated.					

Note that the market-based pricing mechanism is substantially more unstable that the monetary pricing mechanism. Compare B, E, H and K with C, F, I and L respectively. The market-based mechanism was implemented recently, any may still not be completely bug-free. It was added in the hope it might be more stable, but that does not seem to be the case.

Price instability can come from one of two parts of the pricing mechanisms:

- A potential bias from the price genes; or
- A potential bias from the base unit prices used in the preparation of a quote.

By "potential bias" I mean the kind of bias that would cause prices to rise more often than they fall, or vice versa. Bias in the price genes is the easiest to address, so let me discuss that one first.

4.2.1 Potential Bias from the Price Genes

Default Deltas – Bias might come from the default value of delta in the price genes. Originally I set the default value to 1 on initiation. I believed this would not cause bias, because whether the numerator or denominator rises by one, determines whether the strength of a gene rises or falls, respectively. So a rise or fall were equally likely. However the % rise and the % fall are different. I changed the default start value of delta in all genes to 0. This delays any insertion of bias until after a mutation of delta to a non-zero value happens.

Non-zero Deltas – Bias might come from the way a non-zero delta is used to change the strength of genes. As mentioned above, the % rise and the % fall of the strength of μ or σ for any gene are different. So there will be some bias on pricing due to a first round of genetic mutations. A mutation of the numerator of μ or σ causes a larger change upwards than an equally probable mutation of the denominator. There is therefore an intrinsic upward bias towards rising values of these gene strengths. This might be the cause of price instability. A larger of σ will result in a wider spread of possible quotes, so this would not result in a bias on prices, but might be a bias towards more variability, or more volatility in prices. A larger value of μ, on the other hand, would bias price quotes upwards. The question, then, is whether this bias is strong enough to cause a collapse of the economy. How does this migration pressure on prices compare with other potential sources of pressure on prices?

Large Deltas – Delta is essentially an accelerator that is only slightly sensitive to evolutionary pressures, being a latent factor during the life of any agent, and only playing a role during reproduction and mutation. It may be possible that, over an extended period of time, large values of delta are selected for in response to the need for a faster drift rate, but as the gene approaches an ideal value, the large drift rate cannot be easily selected against, and prices drift beyond the ideal values. In other words, a large delta may cause rapid approach to an ideal gene structure, and then promote overshoot. This would cause a bias towards more volatile price quotes, but, due to the fact that delta is applied to either the numerator or denominator, it would have a mild bias upwards on prices.

In conclusion, I think there is the potential for an upwards bias on prices due to the way I have structured the evolution of price genes through mutations. However, I think this bias is small, and develops very slowly over generations as mutations pile on top of mutations. I don't think it can explain the rapid price changes that often occur just before a collapse.

4.2.2 Potential Bias from the Base Unit Prices

I believe the problematic bias comes from this source. I think this source of bias is much stronger, but more difficult to explain.

The memory of previous successful negotiations, within this agent's lifetime, is encoded in the unit prices and values associated with each store of goods. A series of successful deals with escalating extreme prices will bias the agent towards similarly extreme price quotes in following deals. But, by the same means, a series of successful deals with ideal price quotes should bias the agent towards similarly ideal price quotes.

So, I do not understand how there is any bias on prices coming from this source, but I nevertheless believe that this is where the problem comes from.

5 Design Options for Improved Stability

5.1 Change the Mutation Mechanism

Obviously I could remove the upwards bias due to the use of delta in the gene structure. I could simply have two elements in each gene for each of μ and σ, four elements in all per gene, in place of the current eight. So, for example, I could have μ and an absolute associated delta (δ_μ) which is either added to or subtracted from μ. For example, suppose $\mu = 1$ and $\delta_\mu = 0.01$. If μ is to mutate, then on a 50/50 chance, μ goes up or down by exactly δ_μ. Similarly, two numbers could handle σ. I tried this approach in the design of PSoup and found that the rate of genetic movement towards an ideal genotype was far too sluggish in most circumstances, and was often too insensitive, or overly sensitive, to evolutionary pressures. I came up with this gene design currently implemented here in ModEco, in that circumstance, to address those deficiencies, and it worked well there.

So, while this source of upward bias is easy to remove, I don't wish to do it for two reasons. First, I don't think it will solve the problem. And second, I think this design, though somewhat more complicated, will be superior in performance, assuming I can achieve sustainability.

5.2 Change the Base Price Mechanism

The three options for the base price mechanism were each designed in repeated attempts to stabilize the economy while simulating real price negotiations.

As I am somewhat at a loss to determine exactly how the collapse happens, I do not have a good idea to fix it.