UNIVERSITÀ DEGLI STUDI DI TORINO

Fisica dei Sistemi Complessi

Simulation Models for Economics

Policy experiments in a macroeconomic model with endogenous growth

Author:
Marco Pangallo
Date:
12/10/2014

Contents

1 Introduction

Mainstream approach to economics argues that equilibrium is the natural state of the economy, deviations from equilibrium are transitory and unimportant. When economic quantities change over time, for instance in GDP growth, transitions between equilibrium states take place due to exogenous variations of the fundamentals of the economy (in Solow (1956) the production function exogenously changes). On the contrary, complexity economics maintains that economics is endogenously in non equilibrium states: individuals form aggregate patterns, and following a recursive loop they react to those patterns. Fundamental uncertainty and technological innovation make it impossible that a steady state emerges, because of “Brownian motion of decisions” and technologies which are “permanent ongoing generators and demanders of further technologies” (Arthur 2013). Since it seemed impossible to analytically model non equilibrium dynamics, which could involve chaos as well, mainstream economics focused on searching behaviors (preferences, strategies, rules) which were consistent with externally imposed aggregate patterns, such as market clearing and equilibrium uniqueness and efficiency.

In macroeconomics in particular a plethora of assumptions was made to match models with reality. After the neoclassical synthesis, following Lucas' critique (Lucas Jr 1976) macro models looked for microfoundations, assuming a representative individual, rational expectations, perfect markets, and this brought to Real Business Cycles models and neoclassical economics. Incorporating New-Keynesian features of the economy such as price stickiness and imperfections lead to the New Neoclassical Synthesis, grounded in DSGE models. In the “Great Moderation” period such models seemed to work fine, but the current crisis highlighted their limits (Kirman 2010). Whereas some economists think that improvements while keeping DSGE structure could explain what happened and would prove useful in policy analysis (Kocherlakota 2010), other ones claim this is not the case. Fagiolo and Roventini (2012) argue that DSGE models have theoretical, empirical and political economy issues which prevent them to be useful in policy analysis. What they suggest is to turn to Agent Based Computational Economics (ACE), or Agent Based Models. Arthur (2013) acknowledges that using equations allows us to follow economic problems step by step, but the loss in term of realism is invaluable. Fagiolo and Roventini (2012) are well aware of the problems with ACE models, such as over-parametrization, model selection and the role played by initial conditions. However, they suggest that empirical validation could be done by looking at the number of “stylized facts” the model is able to replicate, possibly replacing assumptions in a modular way. This would not be possible in DSGE models, one could not dispense with rationality assumption! Kirman (2012) suggests to start from simple models, which may be analytically tractable, and to complicate them step by step.

My simulation draws many ideas from Dosi, Fagiolo and Roventini (2010). I simulate a 2-sector economy exhibiting endogenous innovation and I study both macro-variables such as GDP as well as micro-variables (e.g. distribution of firm's market shares), looking for stylized facts which could validate my model if they are seen in a wide region of parameters space. If the innovation process goes on, I expect that the economy cannot settle down into an equilibrium state, and keeps growing and changing. I leave out of the simulation everything concerning money, overlooking monetary policies in favor of fiscal and labour market policies. The quantity of money is fixed throughout the simulation. The main difference from Dosi *et al.* (2010) is that I directly simulate every exchange taking place in the economy, rather than using a "quasi" replicator dynamics, being closer to an Agent Based Model (ABM) than to a Monte Carlo simulation, and I simplify some aspects about Schumpeterian features in the firms to focus on the behaviour of consumers/workers[1]. As the authors themselves suggest at the end of their paper, by making use of my simplified model I perform some policy experiments they have not tried.

The rest of this work is structured as follows: Section 2 describes the theoretical model without a specific reference to any programming language. Section 3 describes its implementation in Netlogo[2]: it was necessary to make it more suited to that programming language in order to exploit Netlogo's strengths. I include the code of the most important parts. In Section 4 I explore the parameter space and perform some policy experiments, by making us of R[3] and its package RNetLogo (Thiele *et al.* 2012).

[1]Moreover, as the ABM in the paper is not described in detail, I had to work out several mechanisms to simulate interactions and behaviour of agents and to make some assumption about the evolution of a number of variables.

[2]https://ccl.northwestern.edu/netlogo/

[3]http://www.r-project.org/

2 The model

There are three kind of agents. R&D firms produce capital-goods (machinery) and perform R&D in order to improve the performance of the goods they sell and to cut their production costs. Product firms make final goods using the capital-goods they bought from R&D firms and sell them to their customers. Workers work in the firms and consume only a fraction of their wage. The behaviour of the agents is adaptive. Public expenditure is financed by taxes on firms' profits and workers' wages.

2.1 R&D firms

A number $I = |\mathcal{I}|$ of R&D firms is instantiated and stays in the market until the end of the simulation. Every R&D firm has some attributes: lists containing its workers and its customers (the product firms that bought at least one piece of machinery on previous time step), an integer $M_i(n)$ representing its liquid assets, a real number $f_i(n)$ concerning its market share (the number of orders that firm received with respect to the total of transactions that took place in the R&D firms - product firms market), a tuple (A_i^n, B_i^n), i.e. the two parameters representing the level of technology of each firm at time step n. The cost of producing one machine for firm $i \in \mathcal{I}$ is $c_i(n) = 1/B_i^n$: $c_i(n)$ is the wage payed to firm's workers. By adopting this transaction mechanism no money disappears, all the costs of production are transferred to workers. The highest B_i^n, the lowest is the cost the R&D firm faces. Every R&D firm makes k machines any time step, where $k = \max\{S_i(n-1)+5, S_i(n-1)(1+0.05)\}$ is adaptive with respect to past sales. The price chosen by any R&D firm to sell its machines is $p_i(n) = (1+\mu)c_i(n)$: μ is the markup, constant across R&D firms: they are also homogenous with respect to the fraction ν of their liquid assets they spend in R&D, adding up to $RD_i(n) = \nu M_i(n)$; all the rest of $M_i(n)$ is used to make the k machines[4]. The innovation process is as follows: the R&D firm gets access to a discovery through a draw from a Bernoulli distribution, whose parameter is $\theta_i(n) = 1 - \exp^{-\zeta RD_i(n)}$. Then $A_i^{new}(n) = A_i(n)(1 + x_i^A(n))$, $B_i^{new}(n) = B_i(n)(1+x_i^B(n))$, where $x_i^A(n), x_i^B(n) \sim \Gamma(\alpha, \beta)$, i.e. are drawn from a Gamma distribution with parameters α (shape) and β (rate) and support $[0, \infty)$. The only thing left to R&D firms is to update their list of customers: they start from the list of the product firms that made at least one purchase on the previous time step ($HC_i(n-1)$), and they add $\gamma HC_i(n-1) + 1$ product firms chosen randomly (under the constraint they were not previous customers). γ is a proxy for the degree of perfect information in the market, and this assumption captures the insight that more powerful firms have also more advertising power.

[4]if $p_i(n) \cdot k > M_i(n)$ only a fraction of the machines is produced, if $p_i(n) \cdot k < M_i(n)$ the firm saves money for next time step

2.2 Product firms

A number $J = |\mathcal{J}|$ of product firms is instantiated and stays in the market until the end of the simulation. Product firm have some attributes as R&D firms: a list of workers, market share $f_j(n)$ (which is now the number of goods that product firm sold with respect to the total of transactions that took place in the product firms - consumers market) and liquid assets $M_j(n)$. Peculiar to product firms are a list representing capital stock (a list of h machines whose values are A_i^n's, since B_i^n's only have an effect on the price of the machine and thus can be forgotten), and an integer number $N_j(n)$ (inventories). The product firm can use a machine to produce only one good at any time step. If the desired level of production (to be explained later on) $Q_j(n)$ is such that not all machines are to be used, the product firm starts from the ones with a higher A_i^n. Actually the price it has to pay to produce a consumption good is $c_j(A_i^n, n) = 1/A_i^n$, so A_i^n can be regarded as a measure of the quality of the machine. The desired level of production is again decided on the basis of an adaptive rule: $Q_j^D(n) = 10 + S_j(n-1) + N_j^D(n) - N_j(n-1)$. The desired level of inventories is $N_j^D(n) = 0.1S_j(n-1)$: this way product firms adjust their production to cope with the demand. The 10 in the beginning of the formula is to let them recover from a demand crisis. The problem of product firms is the way they should allocate their liquid assets between production of goods and investments in machinery. Lacking an explanation of this mechanism in the original model, I propose the following strategy: the largest part of liquid assets are allocated to produce $Q_j^D(n)$, a minor part [5] is invested to buy up to $\max\{1, 0.2 \cdot h\}$ new machines [6]. This is also needed because machines have to be replaced after 20 time steps. Finally, if some money is left, it is used to complete the production $Q_j^D(n)$, up to $Q_j(n) \leq Q_j^D(n)$. If some money is still left, it is kept for next time step. Product firms decide which machines to buy according to the following protocol: they check which R&D firms have them in their customer list, they rank R&D firms according to the lowest $p_i(n) + c_j(A_i^n, n)$ (sum of cost and quality), they buy as many machines as they planned by choosing randomly one of the R&D firms in the list. A draw from a $\Gamma(1.2, 0.66)$ distribution, combined with a "floor" function, defines the position in the R&D firms list: it is more likely to choose the most competitive, but sometimes a less competitive R&D firm may be chosen. This prevents, or at least slows down, the formation of a monopolistic market. So far I focused on production plans of product firms. Let's now have a look at their selling strategy. Prices are $p_j(n) = (1 + \mu)c_j(n)$. Since product firms' goods are heterogenous in price, they start selling them from the cheapest ones. Firms profits are $\pi_j(n) =$

[5] the exact amount depends on markup: if investment is larger then profits, firms end up running out of their liquidity, and the result is that the whole economy collapses

[6] in Dosi *et al.* (2010) in footnote 12 the authors mention a "fixed maximum threshold" for the capital growth rate

$\sum_{\text{sold items}} p_j \cdot D_j(n) - \sum_{\text{produced items}} c_j \cdot Q_j(n)$: profits may be negative, but additional production is kept in inventories, and less production will occur next time step. The level of product firm's liquid assets is finally updated: $M_j(n+1) = M_j(n) + \pi_j(n)$. Here $M_j(n)$ is what is left from production and investment.

2.3 Workers/Consumers

A number $N = |\mathcal{N}|$ of workers/consumers is instantiated and stays in the market until the end of the simulation. Workers only have one attribute (i.e. the amount of liquid assets) and they can be employed or unemployed. Labour supply is inelastic: $L^s = N$. Labour demand is adaptive. Firms not only need liquid assets and, in the case of product firms, machinery, but they need also workers. I assume that $\frac{1}{B_i^n}$ workers are needed to produce one unit of output by the R&D firms, $\frac{1}{A_i^n}$ by the product firms. This reflects the technological improvement. At the end of each time step firms hire as many workers as it would be needed to produce the desired amount of output. If next time step the firms need even more workers, they hire them; on the other hand, if they need to fire them, they can fire them with a successful Bernoulli trial with parameter ω. This parameter captures the flexibility of the labour market, and it can be tuned by the government. Firms decide randomly which workers they hire/fire. However there is a problem with this approach. It turned out while implementing the simulation that a realistic labour market would require a balance between technological improvement (i.e. productivity) and level of production: these two quantities should scale together. Actually, if production increases faster than productivity, all workers are to be hired; if it scales slower, most workers are to be fired. Since I wanted to study the effects of several policies on GDP, I devised a simplified labour market imposing a "natural rate of unemployment" in the code, by normalizing labour need of firms to about 90% of labour force. This arbitrary choice allows both to show which firms hire the most workers and to keep some unemployed consumers who can benefit from the policies that target them. All the previously described features of the labour market stand also with this simplification (firms still hire and fire workers). As I already mentioned, all production costs $c_i(n), c_j(A_i^n, n)$ are used in wages. Each firm computes its total costs and divide them equally among its employed workers. The average wages are fixed (since total liquid assets are fixed) but growth is apparent from the sharp increase in production. Consumers spend a fraction $(1-s)$ of their income in consuming goods, and keep a fraction s to increase their liquid assets. When they are unemployed, they consume half of their liquidity to buy goods. To capture imperfect information also in the product firms - consumers market, consumers choose the best price within $J/10$ firms chosen randomly (as if they randomly checked their prices).

2.4 Public sector

The government levies a tax ϕ on both $w(n)$ and $\pi_j(n)$ (R&D funds and wages are not taxed). It can spend its funds in unemployment insurance w^u (it spends a fraction $\delta init)ordirectlybuyingfinalgoods(1-\delta), accordingtothedesiredpolicy.$

3 Simulation

NetLogo's intuitive programming language is specifically designed for Agent Based Models. Its main strength is agentsets (*And every time you use it, the agentset is in a different random order. This helps you keep your model from treating any particular turtles, patches or links differently from any others*[7]). On the other hand, dealing with lists is not trivial. This is why I decided to give up with endowing each agent with lists of other agents, instead I provided them with links to other agents. The main drawback of employing these high level structures is speed: when the number of exchanges in the economy substantially increases, the simulation speed dramatically decreases.

There are three kind of agent breeds: `rdFirms`, `productFirms` and `laborForce`. There are also two link breeds: `supplyRelations` correspond to links between `rdFirms` and `productFirms`, `workRelations` are between firms and `laborForce`. Patches are not important, space is just for illustrative purposes. Rather than listing global and breed specific variables, I describe some procedures and I explain their meaning where needed.

3.1 Setup procedure

```
to setup

clear-all
random-seed Seed

set I HowManyRdFirms
set J HowManyProductFirms
set N HowManyWorkersConsumers
set nu InvestmentInRD
set zeta ChanceNewDiscovery
set alpha 1.2
set beta 1 / TechnologicalDevelopment
set gamma 0.5
set mu Markup
set s SavingRate
set omega FlexibilityLabour
set fi TaxesLevel
set delta SpendingInUnemploymentAid
set totalProduction 0
set governmentAssets 0
```

[7]http://ccl.northwestern.edu/netlogo/docs/programming.html#agentsets

```
[...]

initial-hire

reset-ticks

end
```

The value of most variables is set in the interface; notably, the seed is fixed in order to allow the repeatability of policy experiments and parameters tuning. The names of the variables are in line with those used in Section 2. `initial-hire` is a procedure that links every firm with one worker. `reset-ticks` is in the end of setup procedure, so plots include initial values of the model's variables. Let us also see in detail how `productFirms` are instantiated.

```
create-productFirms J [
        set shape "factory"
        set color yellow
        setxy random-float min-pxcor random-float max-pycor
        repeat 3 [create-supplyRelation-with one-of rdFirms [set color blue]]
        set liquidAssets 100
        set machines []
        set tickOfMachines []
        set workersNeed 1
        set pastSales 0
        set goodsList n-values 10 [1]
        set production 0]
```

They are positioned randomly at the topleft of the View. Every `productFirm` received advertisement only from 3 `rdFirms`, has an empty list of machines, needs just one worker (consistently with `initial-hire`) and is endowed with 10 goods which can be sold at a price of $(1+\mu)\cdot 1$. The graphical choices have been made to make apparent the emergence of top firms, i.e. firms with a high market share, possibly due to better available technology.

3.2 Go procedure

```
to go
  if ticks >= 1000 [stop]
  if ticks > 10 and totalProduction < 2 [stop]
  performResearch
  advertiseMachines
  hireAndFire
  produceMachines
  buyMachines
  produceGoods
  buyGoods
  publicSpend
  tick
```

```
end
```

The second check is to prevent errors due to a complete meltdown of the economy, i.e. empty lists and divisions by zero. I explain in detail the procedures whose implementation has significantly been done using NetLogo features, and which are not a trivial implementation of what is described in Section 2.

3.2.1 advertiseMachines

```
to advertiseMachines
  ask rdFirms [repeat (gamma * (count supplyRelation-neighbors) + 1) [
           let newProductFirms (complement supplyRelation-neighbors productFirms)
           if any? newProductFirms [create-supplyRelation-with one-of newProductFirms
             [set color blue]]
      ]
     ]
end

to-report complement [A B]
  let list-A sort A
  let list-B sort B
  report turtle-set filter [not member? ? list-A] list-B

end
```

Lacking set operations on agentsets, I defined a reporter returning the complement of A with respect to B, i.e. all elements of B which are not part of A. This reporter takes two agentsets as input variables, and it returns another agentset (`turtle-set` is a cast operation on a list given by `filter`). Then R&D firms just create new `supplyRelations` with product firms they did not previously know.

3.2.2 produceMachines

```
to produceMachines
  ask rdFirms [
   ifelse (ticks = 0) [...]
  [let desiredProduction max list (pastSales + 5) ceiling (pastSales * (1 + 0.05))
   ifelse liquidAssets > desiredProduction * unitProductionCost
                  [ set production desiredProduction]
                  [ set production floor (liquidAssets / unitProductionCost)]
   set workersNeed production * unitProductionCost
   let costProduction production * unitProductionCost

   set liquidAssets liquidAssets - costProduction
   set inventories inventories + production
   let workers count workRelation-neighbors
   ask workRelation-neighbors
    [set liquidAssets (liquidAssets + (costProduction / workers)*(1 - fi) )]
   set governmentAssets (governmentAssets + fi * costProduction)]

   set pastSales 0
```

```
  ]
end
```

Two things are worth noticing in this plain piece of code:

- Total liquid assets are kept constant (apart from rounding errors), what R&D firms spend in producing their machines is uniformly used to pay workers' wages. A fraction `fi` is levied by the government.
- `pastSales` are set to zero after the production step. Also product firms behave the same way.

3.2.3 buyMachines

```
to buyMachines
   ask productFirms [if (any? supplyRelation-neighbors with [inventories > 0]) [
     let k max list (pastSales + 5) ceiling (pastSales * (1 + 0.05))
     let bestrdFirms sort-by [[1 / (item 0 technology) + unitProductionCost] of ?1 <
                              [1 / (item 0 technology) + unitProductionCost] of ?2 ]
                            supplyRelation-neighbors with [inventories > 0]
     let sellingrdFirms []
     let investment liquidAssets * mu / 2
     let m 0
     let mincost [unitProductionCost] of min-one-of supplyRelation-neighbors
                                         with [inventories > 0] [unitProductionCost]

     while [k > m and (investment > (1 + mu) * mincost) and
           (any? supplyRelation-neighbors with [inventories > 0])] [
       set mincost [unitProductionCost] of min-one-of supplyRelation-neighbors
                                        with [inventories > 0] [unitProductionCost]
       let listposition floor random-gamma 1.2 0.66
       if listposition >= (length bestrdFirms) [set listposition 0]

       if ([inventories] of (item listposition bestrdFirms) > 0) and
         (investment > (1 + mu) * ([unitProductionCost] of (item listposition bestrdFirms))) [
           set machines lput ([item 0 technology] of (item listposition bestrdFirms)) machines
           set tickOfMachines lput ticks tickOfMachines
           set liquidAssets liquidAssets - (1 + mu) *
                                     ([unitProductionCost] of (item listposition bestrdFirms))
           set investment investment - (1 + mu) *
                                     ([unitProductionCost] of (item listposition bestrdFirms))
           set m m + 1

           set sellingrdFirms lput (item listposition bestrdFirms) sellingrdFirms
           ask (item listposition bestrdFirms) [set inventories inventories - 1
             set pastSales pastSales + 1
             set liquidAssets liquidAssets + unitProductionCost + (1 - fi ) * mu * unitProductionCost
             set governmentAssets governmentAssets + fi * mu * unitProductionCost]
         ]
     ]


     let sellingrdFirmsAgentset turtle-set sellingrdFirms
     ask my-supplyRelations with [not(member? end1 sellingrdFirmsAgentset or
                                  member? end2 sellingrdFirmsAgentset)] [die]
   ]
```

```
43
44     while [(not empty? tickOfMachines) and (ticks - (first tickOfMachines) > 20)] [
45       set tickOfMachines but-first tickOfMachines
46       set machines but-first machines]
47     ]
48
49  end
```

Let us comment the most important commands line by line:

- line 8: investment is constrained by the markup. If it were not, firms would buy too many machines and would run out of their money, thereby being unable to produce consumption goods.
- lines 13 up to 36: product firms keep buying machines until they have bought as many as they want, they have run out of the money they allocated for investment or all R&D firms they know have sold all their machines.
- lines 15 and 16: `mincost` is updated every time. If this operation was not carried out, the simulation could enter in an endless loop due to the fact that the product firm has enough money to buy the cheapest machine, but the cheapest machine has already been bought and is not available anymore.
- lines 17 and 18: selection rule as described in Section 2.2. CDF of Γ function: 0(0.39)1(0.66)2(0.81)3(0.90).
- lines 20 and 21: check that the product firm can afford the chosen machine.
- line 23: `tickOfMachines` is a list used to keep track of when machines were bought. Since machines are heterogenous, all product firms are endowed with a possibly very long list.
- lines 39 up to 41: cancel from advertisement list all R&D firms which the product firm did not buy any good from. This means that some R&D firms have "out fashioned" technology.
- lines 44 up to 46: remove outdated machines.

3.2.4 buyGoods

```
to buyGoods
  ask laborForce [
   ifelse employed = true [let salary liquidAssets - previousStepLiquidAssets
                            set expenditure (1 - s) * salary]
                           [set expenditure liquidAssets / 2 ]

   let knownProductFirms n-of (J / 10) productFirms

   while [any? (knownProductFirms with [not empty? goodsList])] [
```

```
10
11       let chosenProductFirm max-one-of (knownProductFirms with [not empty? goodsList])
12                                        [mean goodsList]
13       let localgoodsList sort ([goodsList] of chosenProductFirm)
14
15       ifelse (1 + mu)*(sum localgoodsList) <= expenditure [
16         set liquidAssets liquidAssets - (1 + mu) * (sum localgoodsList)
17         set expenditure expenditure - (1 + mu) * (sum localgoodsList)
18         ask chosenProductFirm [
19           set pastSales pastSales + length goodsList
20           set liquidAssets liquidAssets + sum goodsList + (1 - fi)* mu * (sum goodsList)
21           set governmentAssets governmentAssets + fi * mu * (sum goodsList)
22           set goodsList []   ]]
23
24       [let temp 0 let m 0
25         while [expenditure > temp and not empty? localgoodsList] [
26           set temp temp + (1 + mu) * (first localgoodsList)
27           set localgoodsList but-first localgoodsList
28           set m m + 1]
29         if m = 1 or m = 0 [stop]
30         set liquidAssets liquidAssets - (1 + mu) *
31         (sum (sublist (sort([goodsList] of chosenProductFirm)) 0 (m - 1)))
32         set expenditure expenditure - (1 + mu) *
33         (sum (sublist (sort([goodsList] of chosenProductFirm)) 0 (m - 1)))
34         ask chosenProductFirm [
35           set pastSales pastSales + m
36           set liquidAssets liquidAssets +
37            (sum (sublist (sort([goodsList] of chosenProductFirm)) 0 (m - 1))) +
38            (1 - fi)* mu * (sum (sublist (sort([goodsList] of chosenProductFirm)) 0 (m - 1)))
39           set governmentAssets governmentAssets + fi * mu *
40            (sum (sublist (sort([goodsList] of chosenProductFirm)) 0 (m - 1)))
41           set goodsList sublist (sort(goodsList)) m (length goodsList)
42         ]]
43
44       ]
45
46        set previousStepLiquidAssets liquidAssets
47        ]
48
49  end
```

- lines 3 up to 5: the spending decisions depend on whether the consumer is employed or unemployed. Since both machines and goods have been produced, first difference between current liquid assets and previous step liquid assets is just the worker's wage.

- lines 11 and 12: the consumer chooses the cheapest product firm by making use of the average cost of the goods on sell. This assumption captures the fact that consumers are just informed about mean features, they do not exactly know and compare all prices because of bounded rationality.

- lines 15 up to 22: in case the consumer can afford to buy all the goods from the chosen product firm, he just buys them and chooses another product firm.

- lines 24 up to 28: otherwise, he understands how many goods he can afford. Updating all features at ones turned out to be faster than updating them good by good.

- line 29: if the `if` condition is satisfied, the consumer has no money to buy more goods.
- lines 30 up to 42: the consumer buys affordable goods. Notice the use of `sublist`, which allows to take off from product firm's goods list only bought items.

4 Data analysis

The list of benchmark parameters is given in Table 1. With these parameters, a typical image appearing on NetLogo's View is that in Figure 1.

Interface input	Symbol	Value
HowManyRdFirms	I	10
HowManyProductFirms	J	40
HowManyWorkersConsumers	N	200
InvestmentInRD	ν	0.04
ChanceNewDiscovery	ζ	0.3
TechnologicalDevelopment	α	1.2
TechnologicalDevelopment	β	200
Extension of advertisement	γ	0.5
Markup	μ	0.2
SavingRate	s	0.3
FlexibilityLabour	ω	0.5
TaxesLevel	ϕ	0.10
SpendingInUnemploymentAid	δ	0.5

Table 1: Benchmark Parameters

If one switches off technological development (i.e. sets $\zeta = 0$) an equilibrium pattern with huge fluctuations emerges (cf. Fig. 2).

However, it is enough to set $\zeta > 0$ and for a wide range of parameters space (to be detailed later on) a stable endogenous growth pattern emerges. It is interesting to look at the distribution of past sales of both R&D and product firms in this context. To get better results, I decided to set $I = 100$, $J = 400$, $N = 2000$ and to turn to R. Netlogo gets open in "headless" mode from R by typing the command `NLStart(nl.path, gui=FALSE)` in the R console, after importing RNetLogo and storing in `nl.path` the address of NetLogo executable. `NLLoadModel(model.path)` opens the model whose address has been stored in `model.path`. `NLCommand("set HowManyRdFirms 100", "set HowManyProductFirms 400", "set HowManyWorkersConsumers`

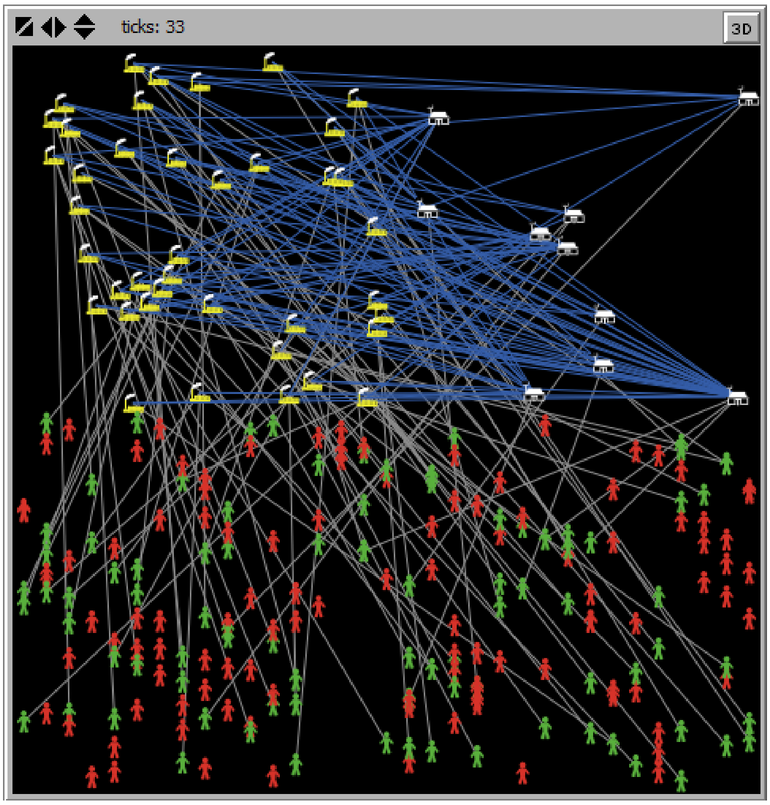
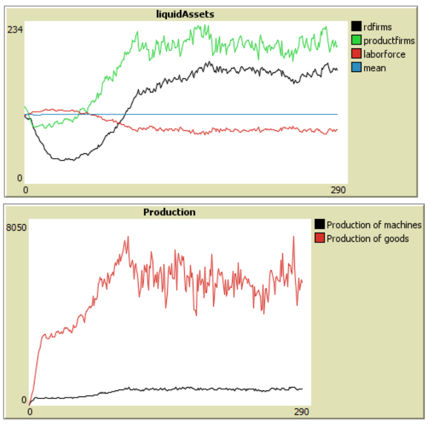
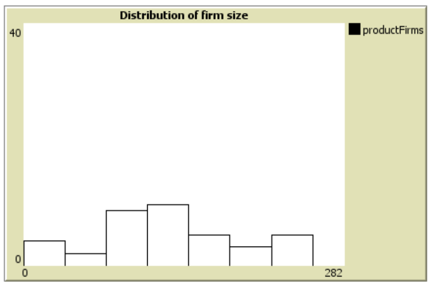


Figure 1: Netlogo's View. Some R&D firms emerge as the dominant ones.



(a) Mean liquid assets of agents and level of aggregate production



(b) Past sales distribution of product firms

Figure 2: Some quantities with technological development switched off

2000", "setup") asks NetLogo to set those values, and then to carry out procedure setup. Notice that the latter has to be carried out at last, as if it would take values from the sliders on Netlogo's interface. The rank-size distribution obtained this way is shown in Figure 3. It can be noticed that past sales do not follow a power law, but indeed exhibit fat tails, as in Dosi *et al.* (2010).

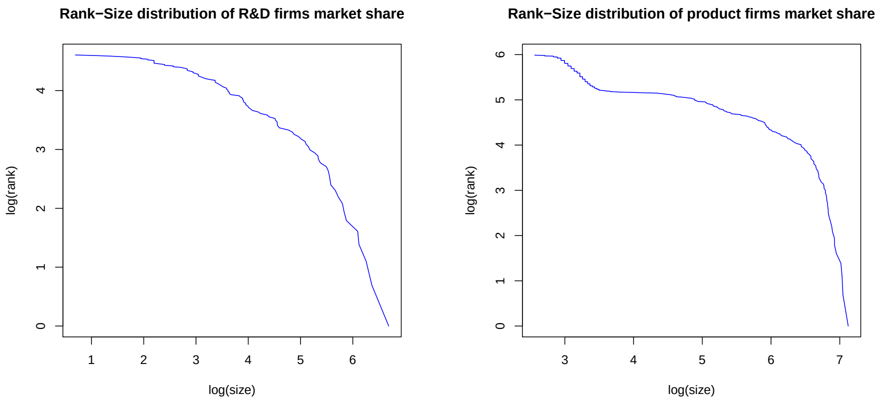


Figure 3: Rank-Size distribution of firms

What can be done next is to see which growth patterns emerge by tuning the several available parameters. I gather them in two groups: Keynesian parameters and Schumpeterian parameters.

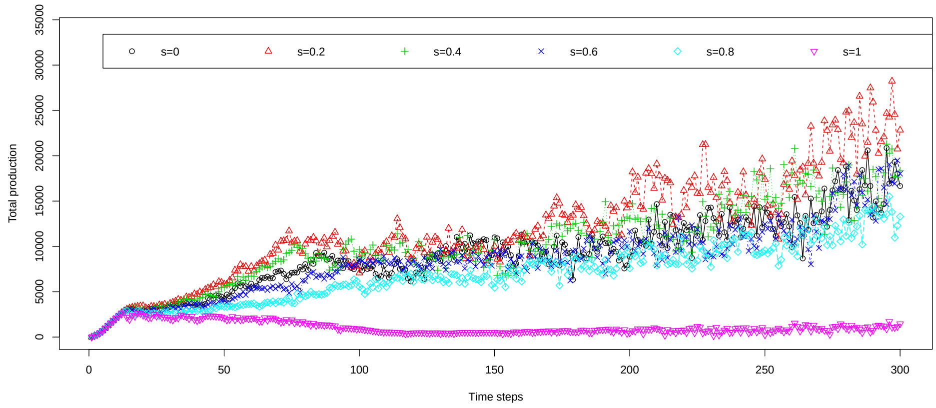
4.1 Keynesian parameters

According to Keynes theory a lower saving rate should imply a higher Keynesian multiplier, hence a stronger economic growth (Blanchard *et al.* 2010). In Figure 4 I plotted the aggregate production against time for several saving rates. Raw data were quite difficult to analyze, so I employed an ARIMA(1,0,1) model in order to get a trend. Using more complicated ARIMA setups made really no difference in the supposed trend.

It can be seen that the highest growth is for $s = 0.2$, and that $s = 1$ implies an economy collapse.

Let us now study the effect of tax level ϕ. In Figure 5 a growth pattern in a restricted range of tax levels is plotted: above $\phi = 0.33$ all tax levels implied zero production after few time steps. This makes sense: if firms do not earn enough, they cannot invest and produce and eventually run out of money.

However it can be seen that a tax level higher than 0 holds a more sustained growth, since government fully spends what consumers may save. This is also in accordance with Dosi *et al.* (2010). In order to get more



Growth at several saving rates

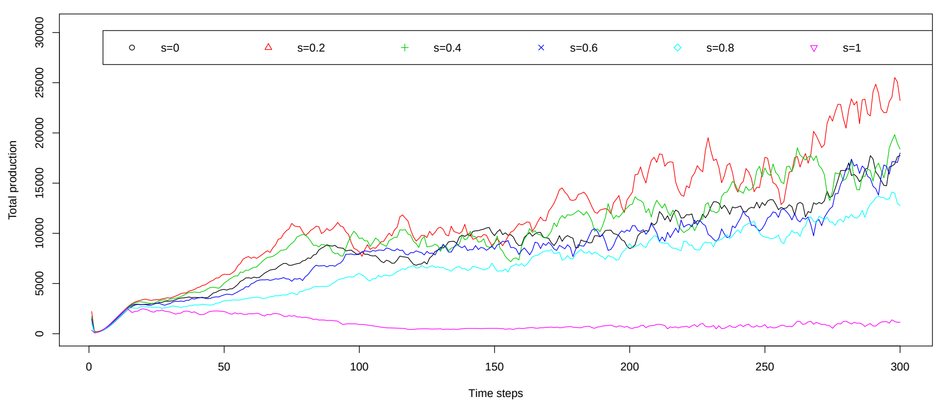


Figure 4: Growth at several saving rates

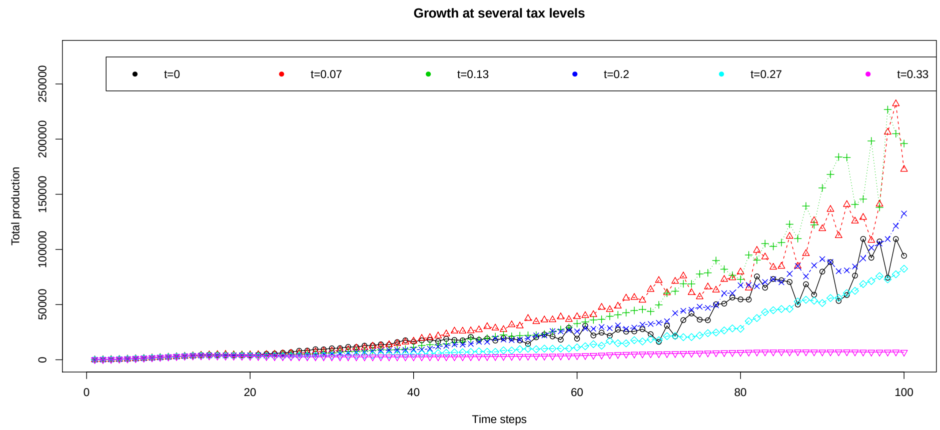


Figure 5: Growth at several tax levels

quantitative measurements, I decided to define average growth rates and volatility as follows:

- Average growth rate: $\bar{r} = \sqrt[t]{\frac{Y_t}{Y_0}} - 1$. Actually $Y_0 = 0$, so Y_1 is used. Also, I decided $Y_t = \frac{1}{5} \cdot \sum_{i=0}^{4} Y_{t-i}$. By taking the mean of the last production levels I ruled out sudden fluctuations.

- Volatility: $v(t) = \sqrt{\frac{1}{t} \cdot \sum_{i=0}^{100} (Y_i - Y_i^{arma})^2}$

In Table 2 I obtained results compatible with plots in Figure 5: it is possible to see that $\phi = 0$ implies a higher volatility than $\phi = 0.2$ or $\phi = 0.27$, which lead to similar final aggregate production. Thus government spending also ensures more stability.

Tax level	Average growth rates	Volatility
0.00	0.069	8361
0.07	0.075	12868
0.13	0.076	12694
0.20	0.070	3427
0.27	0.066	1948
0.33	0.041	101

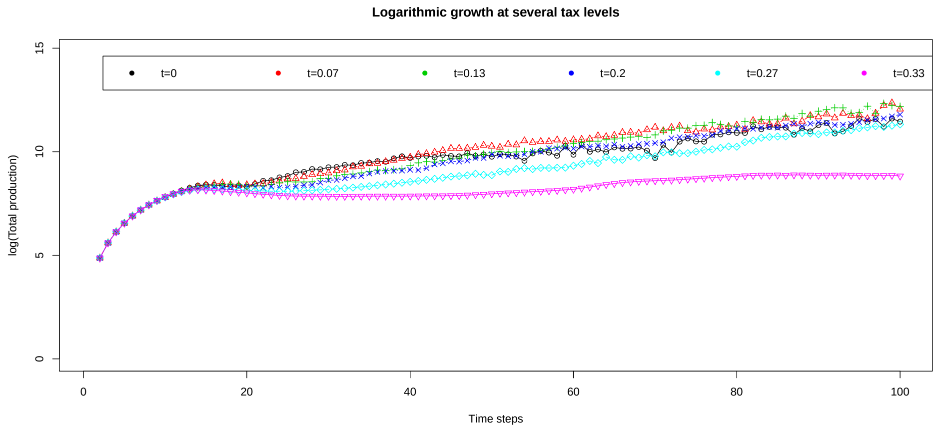
Table 2: Average growth rates and volatility at several tax levels

Finally, since taking a logarithm of total production let me find a linear regime for $t > 20$ (cf. Fig. 6a), I decided to fit growth patterns with exponential curves. Here it is a short piece of R code.

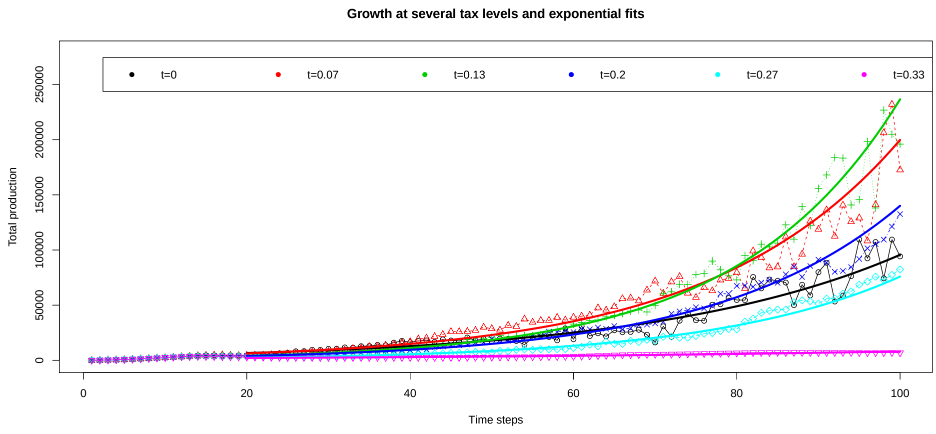
```
exponential.models = list()
```

```
predictions = matrix(nrow = 81, ncol = 6);
volatility = numeric()
for (i in 1:6){
  dataframe <- data.frame(totalProductionMatrix[20:100,i],20:100)
  exponential.models[[i]] <- lm(log(dataframe[[1]]) ~ dataframe[[2]])
  predictions[,i] <- exp(predict(exponential.models[[i]], list(Time=20:100)))
  volatility <- append(volatility,
        sqrt(sum(1/81*(predictions[,i]-totalProductionMatrix[20:100,i])^2)))
}
```

Notice that `lm` method only accepts `data.frame` arguments. In Figure 6b it is possible to see the fit. $\phi = 0.13$ seems to deliver the highest growth.



(a) Log growth



(b) Fit and growth

Figure 6: Growth at several tax levels

Finally, I explored policy parameters such as δ and ω. Results are summed up in Table 3.

It is interesting to see what happens in Table 3b as $\omega = 0$: firms cannot fire their workers, all labor force gets employed and for some reason the

Fraction of spending in unemployment aid	Average growth rates	Volatility
0.0	0.075	12881
0.1	0.073	11148
0.2	0.075	12497
0.3	0.075	13652
0.4	0.071	6856
0.5	0.071	11216
0.6	0.073	13420
0.7	0.075	8142
0.8	0.072	8394
0.9	0.070	7821
1.0	0.068	7050

(a) Change of δ

Flexibility of labour market	Average growth rates	Volatility
0.0	-1.000	73
0.1	0.037	181
0.2	0.042	246
0.3	0.041	258
0.4	0.041	345
0.5	0.044	388
0.6	0.044	402
0.7	0.044	368
0.8	0.043	389
0.9	0.043	345
1.0	0.044	424

(b) Change of ω

Table 3: Change of policy parameters

economy collapses. It seems that a percentage of unemployed people has a stabilizing effect over the economy.

4.2 Schumpeterian parameters

Two parameters related to technological development are available: how much R&D firms invest in research for better machines, how likely it is to make a discovery. Changing β (i.e. the parameter of the gamma distribution) just brings to a increase of technology level, and may slow down the program because growth gets faster.

In Table 4 I show growth rates and volatility for several values of ζ. It is apparent that higher discovery chances mean a more sustained growth, but after $\zeta > 0.4$ growth gets stable: it is almost sure that new discoveries will be made (also provided that $RD_i(n)$ increases as well), and differences may be due to stochastic fluctuations.

Discovery chance	Average growth rates	Volatility
0.0	0.039	250
0.1	0.056	917
0.2	0.064	2297
0.3	0.067	3296
0.4	0.071	4575
0.5	0.069	4095
0.8	0.074	8242
1.0	0.071	NA
2.0	0.072	5476
3.0	0.072	5243

Table 4: Change of discovery chance. For $\zeta = 1.0$ it was not possible to compute ARMA fit, neither with a standard method nor with maximum likelihood

In Figure 7 I show growth rates and a normalized volatility (which would be on a different scale) for changes in ν, the parameter determining how much of R&D firms' liquid assets is invested in R&D.

It is interesting to see that growth drops both if investment is too high or too low. Consistently with other parameter changes, as soon as firms do not produce enough goods and do not make enough profits, they eventually run out of their liquid assets and the economy collapses. Volatility is higher for low investment in R&D: it may be that just some firms are lucky and discover better machines, so stochastic effects do not average out.

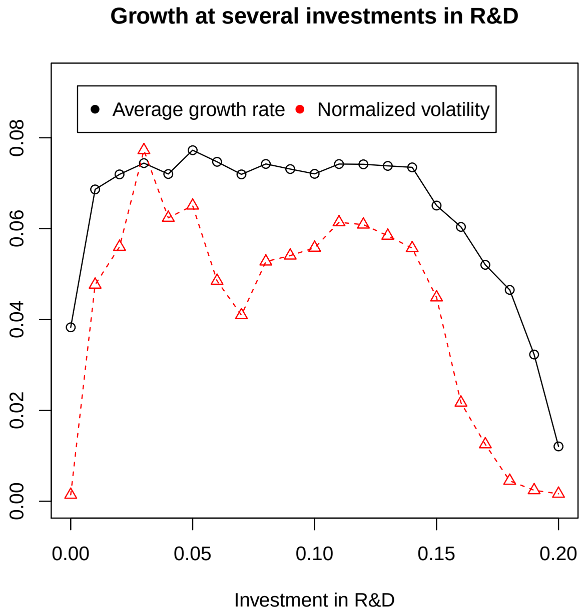


Figure 7: Growth at several investments in R&D

5 Conclusion

In this work I studied a model displaying technological development due to a trial-and-error process performed by R&D firms. This framework lead to a robust growth path in the total production. Without imposing any ex ante consistency requirement but the conservation of total liquid assets, I was able to study an endogenously generated non equilibrium model not treatable with analytical techniques. The behaviour of agents is adaptive and their choices are made with bounded rationality. The implementation of this model has been done using NetLogo, but data analysis has mostly been performed in R. Two salient features validate this model: from an aggregate perspective, it shows a stable growth in a wide parameter region. When it moves into a “collapsed phase”, it is because of unnatural firm decisions: firms would systematically invest more than they can afford. From a micro perspective, the distribution of past sales of both R&D and product firms, though not a power-law, is fat tailed. Firms which were not able to innovate enough only share a small slice of the pie, but some firms act as olygopolists or monopolists in their market. The main findings are the following. Lower saving rates and appropriate tax levels benefit the economy, also in the long run, since the government spends its liquid assets more efficiently. This also contributes to the reduction of production volatility. It does not make much difference the way in which the government spends what it earned from taxes, i.e. unemployment benefits or purchase of final goods, albeit the first seems to harness volatility. Finally, though higher chances of discovery imply a faster growth, investment in R&D should not overcome a certain threshold. Possible improvements of this model are mostly limited by computational capabilities, since the more processes get complex, the more operations have to be done every time step, eventually almost stopping the program as the number of traded goods becomes of the order of 10^6. Since this model is a simplified version of that of Dosi *et al.* (2010), some aspects such as imitation of technologies, adaptive markups or firm replacement could be explored. Moreover, introducing banks could allow to dispense with fixed liquidity assumptions, and a monetary multiplier could be studied. Finally, it could be interesting to focus on interactions between agents in the economy and complicate them, so to obtain a non markovian temporal network (Holme and Saramäki 2012) and to study its properties: would the degree distribution be a power law? How would economic decisions affect network properties?

6 Bibliography

Arthur, W. B. (2013). *Complexity economics: a different framework for economic thought*. In ≪Complexity Economics, Oxford University Press (forthcoming)≫.

Blanchard, O., Amighini, A. and Giavazzi, F. (2010). *Macroeconomia, una prospettiva europea*. Il Mulino.

Dosi, G., Fagiolo, G. and Roventini, A. (2010). *Schumpeter meeting Keynes: A policy-friendly model of endogenous growth and business cycles*. In ≪Journal of Economic Dynamics and Control≫, vol. 34(9), pp. 1748 - 1767.

Fagiolo, G. and Roventini, A. (2012). *Macroeconomic policy in dsge and agent-based models*. In ≪Revue de l'OFCE≫, (5), pp. 67–116.

Holme, P. and Saramäki, J. (2012). *Temporal networks*. In ≪Physics reports≫, vol. 519(3), pp. 97–125.

Kirman, A. (2010). *The economic crisis is a crisis for economic theory*. In ≪CESifo Economic Studies≫, vol. 56(4), pp. 498–535.

— (2012). *Can Artificial Economies Help us Understand Real Economies?*. In ≪Revue de l'OFCE≫, (5), pp. 15–41.

Kocherlakota, N. (2010). *Modern macroeconomic models as tools for economic policy*. In ≪The Region, Federal Reserve Bank of Minneapolis≫, vol. 2010, pp. 5–21.

Lucas Jr, R. E. (1976). *Econometric policy evaluation: A critique*. In *Carnegie-Rochester conference series on public policy*, vol. 1. Elsevier, pp. 19–46.

Solow, R. M. (1956). *A contribution to the theory of economic growth*. In ≪The quarterly journal of economics≫, pp. 65–94.