Danish Research Unit for Industrial Dynamics

DRUID Working Paper No. 96-2

The Nelson and Winter Models Revisited: Prototypes for Computer-Based Reconstruction of Schumpeterian Competition

by
Esben Sloth Andersen
in collaboration with
Anne K. Jensen, Lars Madsen and Martin Jørgensen

April 1996

ISSN 1396-2035

The Nelson and Winter Models Revisited:
Prototypes for Computer-Based Reconstruction
of Schumpeterian Competition

by
Esben Sloth Andersen
in collaboration with
Anne K. Jensen, Lars Madsen and Martin Jørgensen
Department of Business Studies
Aalborg University
Fibigerstraede 4
9220 Aalborg OE
Denmark
Phone +45 98158522
Fax + 45 98156013
E-Mail esa@business.auc.dk

Abstract

The report deals with the reconstruction and further development of the models of industrial dynamics developed by Nelson and Winter and summarised in their famous 1982-book. The basic idea underlying the Nelson and Winter models is that a verbal account of Schumpeterian competition can naturally be transformed into a description of a computational process in which firms not only make short-term production decisions and investment decisions but also performs a search for new technologies. The latter search is successful in a probabilistic manner, and its successes and failures determine an evolutionary process of the industry. Although the simulation models of Nelson and Winter have played a central role the 'take-off' of evolutionary economics, they have never been fully documented and their differences have never been explored. The resulting problems are obvious for students who start from Nelson's and Winter's most famous accounts, but even for researchers with a full collection of the underlying research papers, the situation is quite confusing. The report tries to make things easier by presenting overviews of the structure of Nelson and Winter models as well as fully implemented versions of their simulation models – especially NELWIN78 based on ch. 13 of the 1982-book and NELWIN77 based on ch. 12. The report furthermore presents a computer-based environment (implemented in MAPLE V Rev2) for revision of the models and for analysis of the overwhelming number of data resulting from simulation runs.

Keywords

Evolutionary economic modelling, simulation of industrial dynamics, Schumpeterian competition, Nelson and Winter

JEL classification

L0, L1

ISBN 87-7873-005-8

Preface

This report deals with the reconstruction of the Nelson and Winter (N&W) models of Schumpeterian competition. The reconstructed N&W models will function as a reference point in later phases of a project of the modelling of complex evolutionary dynamics. This project (DRUIDIC) is a part of the Danish Research Unit for Industrial Dynamics (DRUID) and it is performed at the Dept. of Business Studies, Aalborg University.

DRUIDIC means Dynamic Reconstruction of Unfolding Industrial Diversity by Interactive Computing. The end task of the project is to develop a family computer models with special emphasis on vertical industrial dynamics. To obtain this goal the project is divided into several subprojects (project phases), each producing a model that will more or less be used in the next phase of the project:

The starting point is a reconstruction of the N&W models of Schumpeterian competition (the NELWIN models). The main problem is to make the models sufficiently robust and documented. The present report covers this phase. Based on the NELWIN models a related growth model of a labour-based economy will be developed and compared with the original N&W growth model.

The next step is to introduce a production chain and possibilities for firms to specialise in parts of the chain (DRUIDchain). The DRUIDchain model is the starting point for introducing a non-linear production structure that are branching like a tree (the DRUIDtree model). Both the DRUIDchain model and the DRUIDtree model provide starting points for modelling of the evolution of so-called innovation systems.

The DRUIDIC project is lead by Esben S. Andersen (e-mail: esa@business.auc.dk). Anne K. Jensen, Martin Jørgensen and Lars Madsen are project members in the period is February to September 1996. Further work is planned to take place during the next couple of years. The working papers of the project will be published in the DRUID Working Paper series.

Aalborg 28 March 1996

Esben Sloth Andersen

Abbreviations:

DRUID = The Danish Research Unit for Industrial Dynamics.
DRUIDIC = The project on the Dynamic Reconstruction of Unfolding Interfirm Diversity in Industrial Competence.
DRUID# = The DRUIDIC model characterised by #.
N&W = Nelson and Winter.
NELWIN# = The Nelson-and-Winter model published in a paper dated year #.

Contents

1. The family of NELWIN models

The present report deals with the reconstruction and further development of the models of industrial dynamics developed by Nelson and Winter (N&W).

Table 1. A broad definition of the N&W family of models

Model Name	Description	Documentation
1. Theoretical models		
NELWIN68	‘Development and Backwardness in a Two-Technology Evolutionary Model’	Nelson, 1968. N&W, 1982, ch. 10, 235-240, 240-245.
NELWIN71	‘A Particular Model of Economic Selection’	Winter, 1971. N&W, 1982, ch. 6, 144-154.
NELWIN75	‘A Markov Model of Factor Substitution’	N&W, 1975. N&W, 1982, 175-184.
2. Simulation of economic growth		
NELWIN76	‘An Evolutionary Model of Economic Growth’	N, W and Schuette, 1976. N&W, 1982, ch. 9, 209-214.
3. Simulation of Schumpeterian competition		
NELWIN77	‘Dynamic Competition and Technical Progress’	N&W, 1977, 75-76. N&W, 1982, ch. 12, 281-287, 302 f.
NELWIN78	‘Forces Generating and Limiting Concentration under Schumpeterian Competition’	N&W, 1978, 544-546. N&W, 1982, ch. 13.
NELWIN82	‘The Schumpeterian Trade-off Revisited’	N&W, 1982b. N&W, 1982, ch. 14.
NELWIN84	‘Schumpeterian Competition in Alternative Technological Regimes’	Winter, 1984, 317-320, 298-309.
4. Related PhD theses		
SC80	‘The Role of Firm Financial Rules and a Simple Capital Market in an Evolutionary Model of Industry Growth’	Schuette, 1980.
GE82	‘Innovation, Wettbewerb und Evolution: ... Herstellern und Anwendern neuer Produzentengüter’	Gerybadze, 1982.

The basic idea underlying the N&W models is that a ‘verbal account of economic evolution seems to translate naturally into a description of a Markov process – though one in a rather complicated state space.’ (N&W, 1982, 19) At a certain point of time, t, the state of the evolutionary process of an industry is described by e.g. the capital stock and the behavioural rules of each firm. This state is used for determining the short-term behaviour of the industry as well as the new capital stock and the new behavioural rules of each firm at time $t + 1$. It is the shift in behavioural rules which gives the overall evolution the character of a stochastic Markov process.

When this process of state transformation is defined, it is relatively easy to translate it into computer models and simulations. On the other hand, ‘the simulation format does impose its own constructive discipline in the modeling of dynamic systems: the program must contain a complete specification of how the system state at $t + 1$ depends on that at t and on exogenous factors, or it will not run.’ (N&W, 1982, 208 f.)

The N&W models were developed in the 1970s and early 1980s, and they met wide-spread attention when N&W rewrote several earlier articles and formulated a research programme in their book on *An Evolutionary Theory of Economic Change* (1982). The models included in this book were clearly designed to create the outlines of scientific paradigm for evolutionary economics; in this respect they are simple examples of a ‘vastly larger’ class of Markov models (N&W, 1982, 407). To study the potentials of this paradigm, we could turn to the works of some of the followers of N&W – like Silverberg, Dosi, and Orsenigo (1988); Kwasnicki (1992); Chiaromonte and Dosi (1993); Silverberg and Verspagen (1994).

However, the followers tend to move in many directions, so the similarities are not easy to discern.

To come to grips with N&W's suggested paradigm of evolutionary economics, it seems important to revisit N&W's own formulations. The different members of the N&W family of models are mentioned in table 1. We shall especially emphasise the subfamily of models that are implemented on a computer. The reconstructions of these models may be called:

NELWIN76 – presented in ch. 9: 'An Evolutionary Model of Economic Growth'; originally presented in N&W, 1974 and in Nelson, Winter and Schuette, 1976.

NELWIN77 – presented in ch. 12: 'Dynamic Competition and Technical Progress'; originally presented in N&W 1977.

NELWIN78 – presented in ch. 13: 'Forces Generating and Limiting Concentration under Schumpeterian Competition'; originally presented in N&W 1978.

NELWIN82 – presented in ch. 14: 'The Schumpeterian Tradeoff Revisited'; originally presented in N&W, 1982b.

NELWIN84 – presented in Winter, 1984: 'Schumpeterian Competition in Alternative Technological Regimes'.

Although these simulation models have played a central role the 'take-off' of evolutionary economics, they have never been fully documented and their differences have never been fully explored. For instance, their similarities and differences has not been discussed, and the N&W computer programs have apparently disappeared.[1] The resulting problems are obvious for students who start from the N&W book, but even for researchers with a full collection of the N&W papers, the situation is still confusing.

The present report tries to make things easier for present-day evolutionary modellers by presenting reconstructions – called NELWIN models (or shorter NW models). The attempt of reconstructing NELWIN models should be seen as a continuation of Andersen (1994, ch. 4 and Appendix).[2] The report can most conveniently be studied in parallel with the Andersen 1994-book.

2. The basic structure of NELWIN models

2.1. Background

The evolutionary models created through the collaboration of N&W build on their previous individual scientific works. Before their joint work, Winter (1964, 1971) had already made important critiques of the Alchian–Friedman selection argument for profit maximisation while Nelson had been working on the economics of invention, innovation and technical change (Nelson, 1959a, 1959b, 1968). Through their common research endeavour they made an evolutionary synthesis by integrating ideas about:

1. Behavioural patterns and their transmission.
2. Creation of new behavioural patterns.
3. Different types of selection mechanisms.

More specifically, we may say that they combined:

1. Simon's work on rules and satisficing behaviour.
2. Nelson's and other 'Schumpeterian' work on invention and innovation.
3. Alchian's and Winter's work on 'natural selection'.[3]

In the creation of the evolutionary synthesis there is little doubt that the contribution of Simon (and Cyert and March) was of much importance.[4] However, the evolutionary synthesis is a clear example that the whole is more than its constituent parts. Furthermore, we see that the elements were reshaped to fit into their new place in the evolutionary synthesis. Therefore, it is appropriate to present the models as if they were wholly created through N&W's joint work even if they are heavily indebted to several sources.

N&W's evolutionary models are based on the postulate that it is possible to specify the space in which innovative search takes place as well as the way the actual search process takes place. In other words, we postulate a degree of stochastic predictability of most innovative activities. This predictable aspect of economic change may be seen as a result of bounded rationality leading to localised search in the space of (technological and marketing) alternatives.

Generally, we try to imagine the state of a firm at period *t* with respect to, e.g., physical capital and productivity. Together with output rules and functions of factor supply and final demand, this state determines the firm's competitiveness *vis-à-vis* its similarly described competitors and thus its profitability in period *t*. The firm's state in the next period, $t + 1$, is determined by its (simplified) investment rules and by its search rules (and thus search costs) together with the probabilities of finding new rules in the space of alternatives. A newly found alternative will only be included into the new state of the firm if it is judged to increase expected profitability.

This modelling strategy may be summarised in the following way:

1. Define the minimum environmental characteristics, including input and output conditions as well as the spaces in which search for new rules are performed.
2. Define the state of the industry at time *t* as a list of firm states which include physical and informational characteristics as well as behavioural rules and meta-rules.
3. Calculate by means of (1) and (2) the activities of the industry in period *t* as well as the resultant state variables (including possible changes of rules) which characterise the system at the start of period $t + 1$.
4. Make similar calculations for a series of periods and study the evolution of the application of different rules as well as other characteristics of the industry (economy).

By accepting such elements in their model-makers' tool-kit, N&W are imposing upon themselves a certain conception of the evolutionary process. First of all, they apply a population perspective. An 'industry' or an 'economy' is seen as a taxonomic class incorporating a certain degree of variety of processes (and/or products); but the variants must, in principle, be transferable between the different firms. This also implies a certain similarity of the search spaces of the firms, although there may be major differences with respect to the 'distance' to different sources of knowledge.[5] Second, the name of the game is variety-creation and variety-selection within a given economic pattern. In other words, N&W emphasise change which follows 'natural trajectories' within given 'technological regimes' (N&W 1982, 258-262) or 'technological paradigms' (Dosi, 1982) rather than radical change. The latter is suggested in contrast to their models rather than within their models. Third, 'a vast array of particular models can be constructed within the broad limits of the theoretical schema' but the 'enormous generality' of the schema cannot be exploited immediately (N&W, 1982, 19 f). To obtain real understanding about how to handle their powerful family of models, N&W prefer to concentrate on 'very simple examples' and to 'distinguish sharply between the power and generality of the theoretical ideas we employ and the much more limited results that our specific efforts have yielded thus far.' (pp. 20 f)

2.2. The structure of a standard NELWIN model

2.2.1. Overview

As all the other NELWIN simulation models, the standard NELWIN model determines (probabilistically) what happens in each period.[6] In the following we shall describe the steps in this computation. The numbering takes place in accordance with figure 1 which also gives a quick overview over the computational structure.

2.2.2. State variables time t

1. The state of the industry is defined in terms of the size of the physical capital stock (K_{it}) and the productivity of capital (A_{it}) of each of the *n* firms. This state is inherited from the former period.

2.2.3. Short run

Based on the state variables we turn to simple short-run system, i.e., a simplified economic process in the industry whereby output, price and profits of firms are found:

2. The production is characterised by constant returns to scale. The maximum output that can be produced by a firm is thus $Q_{it} = A_{it}K_{it}$. Output of each firm is decided by a full-capacity utilisation rule. This means that actual output is equal to maximum output for each firm. Output of the industry (Q_t) is found by simple aggregation.

3. The aggregate output of the industry faces exogenously given demand conditions that are characterised by unit elasticity, i.e. the same total revenue (D or $Dtot$) is obtained by the industry no matter how much or little it produces. In other words, price adapts to clear the market: $P_t = D / Q_t$

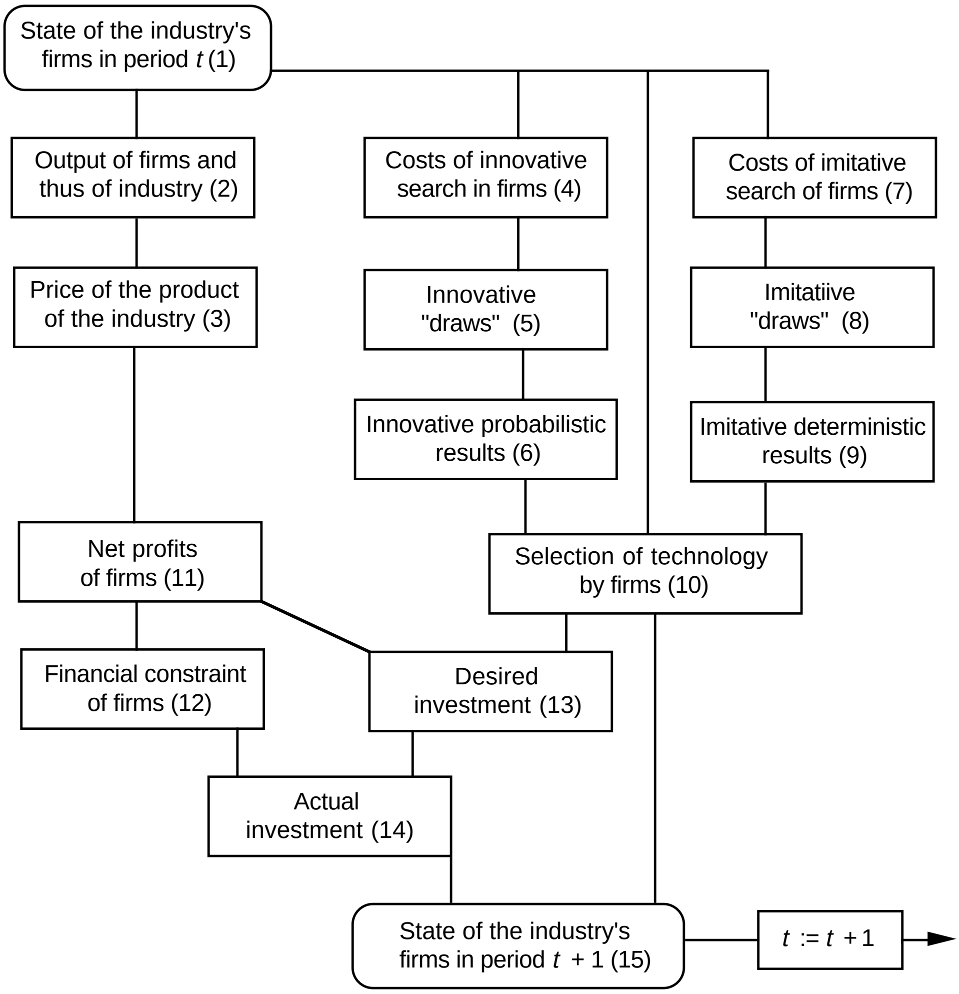

Figure 1. The computational structure of the standard NELWIN model.

2.2.4. New technology

Productivities are specific to individual firms. They reflect knowledge that has a fairly high some degree of appropriability of the results of R&D. Technical change takes the form of process innovations and process imitations that increase the capital coefficient of individual firms $\left(A_{i,t+1} \geq A_{it}\right)$. The processes whereby new production techniques are found and productivity is changed include the following steps:

4. The firms' costs of innovative R&D are found by fixed decision rules that determines them in proportion to the level of physical capital $\left(r_i^{\text{in}} K_{it}\right)$.

5. The firm’s chance of getting a ‘draw’ (an innovation) in the innovative ‘lottery’ (d^{in}) is proportionate to its innovative R&D costs, and it is determined by the exogenously given character of technical change of the industry. It takes the form of a Poisson distribution with a mean number of successes per period determined by the effort of the firm as well as by the appropriability of the technology.

6. An innovative ‘draw’ gives the firm access to another ‘lottery’ that determines the productivity of the innovation. This productivity depends on an exogenously given normal probability distribution. The normal distribution has a mean $\left(\ln\left(A_t^{\text{science}}\right)\right)$ defined by the exponentially growing science-based state-of-the-art. The standard deviation of the distribution is fixed, and the result is transformed back from the log-form to an ordinary productivity.

7. The firms’ costs of imitative R&D are found by fixed decision rules that determines them in proportion to the level of physical capital $\left(r_i^{\text{im}} K_{it}\right)$. The costs are very small, since the industry comes near to pure spill-overs from the innovators.

8. Because of its imitative search effort, each firm gets access to a ‘lottery’. Its probability of obtaining a ‘draw’, i.e. to draw a ticket from the lottery, is proportionate to its imitative search costs but is otherwise determined by exogenous factors (the difficulty of imitation in the particular industry).

9. A 'draw' means that the firm gets access to the best-practice technique and thus the highest productivity level obtained by any firm in the period.

10. The attempts to improve productivity end with a comparison between the productivities obtainable by the technique inherited from the last period and the techniques which may be found by imitative and innovative search. The technique with the highest productivity is chosen. If the technique is changed, it will determine productivity of the next period (disembodied technical change). We thus have the state of technique (production routines) for period t + 1.

2.2.5 New capital

Now we turn to the investment decisions:

11. For each firm we calculate the turnover $\left(P_t Q_{it} = P_t A_{it} K_{it}\right)$ and then find the net profit by deducting the costs elements (which are all measured per unit of physical capital).[7] Taken together variable production costs, capital depreciation,[8] and interest amounts to c per unit of capital, c is assumed to be constant over all periods. The costs of innovative and imitative R&D are determined by fixed decision rules that determines them in proportion to the level of physical capital $\left(r^{\text{in}}, r^{\text{im}}\right)$. Profits per unit of capital are calculated by including R&D costs as ordinary cost elements: $\pi_{it} = P_i A_{it} - \left(c + r^{\text{in}} + r^{\text{im}}\right)$.

12. The maximum investment of a firm is determined by the profits of the present period plus loans from the banks in proportion to the profits. This allows a primitive treatment of the role of banks’ rules in the evolutionary process (see N&W, 1982, 291 ff.).

13. The firm’s desired investment is determined by the unit costs in the next period, a mark-up factor influenced by the market share of the firm, and the rate of depreciation.

14. The actual investment is the minimum of (11) and (12) provided that the result is not negative. The changes in physical capital influences production in the next period.

2.2.6. State variables time t + 1

15. The investment process has no time-lags. The adjusted physical capital stock is available to the industry's firms in period t + 1. By multiplying the capital stock with the new level of productivity, we have the production capacity of the firms of the industry in period t + 1. Similarly, the new productivity is available throughout the innovating or imitating firm.

3. NELWIN78: the setup of experiments

From the viewpoint of a first testing, NELWIN78 is more accessible than the other NELWIN models. The reason is that it is specially designed for making simple hypotheses and tests.[9] We shall explore this characteristic of the model by starting from experiments with the model and gradually including more and more details about the structure of NELWIN78 and the other NELWIN models of 'Schumpeterian competition'. In a later report we shall explore the N&W growth model (NELWIN76).

3.1. Relationships to explore

To define experiments with NELWIN78 we shall, like N&W, take a couple of themes from industrial economics: (1) factors influencing industrial concentration, and (2) factors influencing productivity growth and static efficiency (see e.g. Scherer and Ross, 1990). As our independent variables we shall take (1) industrial concentration and (2) average productivity and the gap between actual and potential production. To explain these variables, we use a set of exogenous variables – partly relating to the basic conditions of technological change in the industry, partly to rules of investment applied by the firms. The basic logic of the exercise is depicted in figure 2.

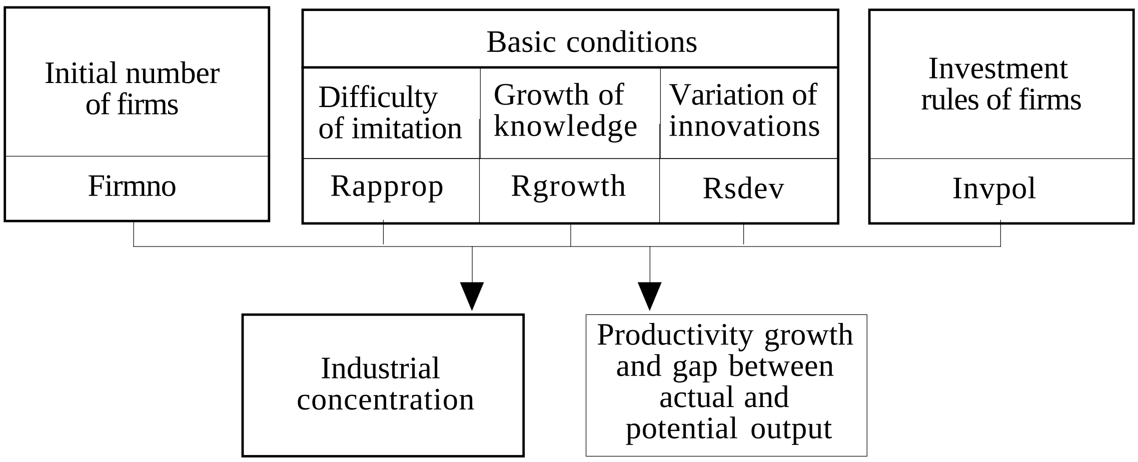

Figure 2: Causality in the experiments with NELWIN78.

NELWIN78 is designed to make easy experiments with these relationships. In the beginning of each run, the user defines each of the 5 exogenous variables (**Firmno**, **Rapprop**, **Rgrowth**, **Rsdev**, **Invpol**) to be high or low. Thus, the set of parameters can be defined by a string of 5 binary digits: 11111 means that all parameters are set to high, 01000 means that all parameters are set to low except the difficulty of imitation (**Rapprop**). Given this information (as well as the number of periods of the simulation), the NELWIN78 computer program performs the simulation. The resulting concentration and productivity conditions can then be studied, and the results can be compared with runs based on other parameter settings.

3.2. Endogenous and exogenous variables

To be more concrete, we need definitions of the endogenous and exogenous variables. First, we need a measure of industrial **concentration**. Here we start from the Herfindahl index:[10]

$$H_t = \sum_{i=1}^{n} s_{it}^2,$$

where n is the number of firms and s_{it} is the market share of firm i in period t. In our studies we (like N&W) apply the reciprocal of the Herfindahl index (called HHs[t] in appendix 2) which gives the number of equally-sized firms that would have the same Herfindahl index as the industry actually have.

Then we turn to **productivity**. In NELWIN78 (figure 3) the state of the industry is defined in terms of the size of the physical capital stock (K_{it}) and the productivity of capital (A_{it}) of each of the n firms. The production of the industry is characterised by constant returns to scale. The maximum output that can be produced by a firm is thus $Q_{it} = A_{it}K_{it}$. Output of each firm is decided by a full-capacity utilisation rule. This means that actual output is equal to maximum output for each firm. Output of the industry (Q_t) is found by simple aggregation. This construction of the model means that a **production gap** (Qgap[t], called AGAP[t] in appendix 2) will normally exist between actual production and production in the case that all firms use best-practice technology:

$$Q_t^{\text{gap}} = \frac{\sum_{i=1}^{n} A_{it}K_{it}}{\sum_{i=1}^{n} A_t^{\max}K_{it}},$$

where A_{it} is the productivity (or output–capital ratio) of firm i in period t, $A_t^{\max}$ is the maximum productivity of any firm in period t, and K_{it} is physical capital of firm i in period t.

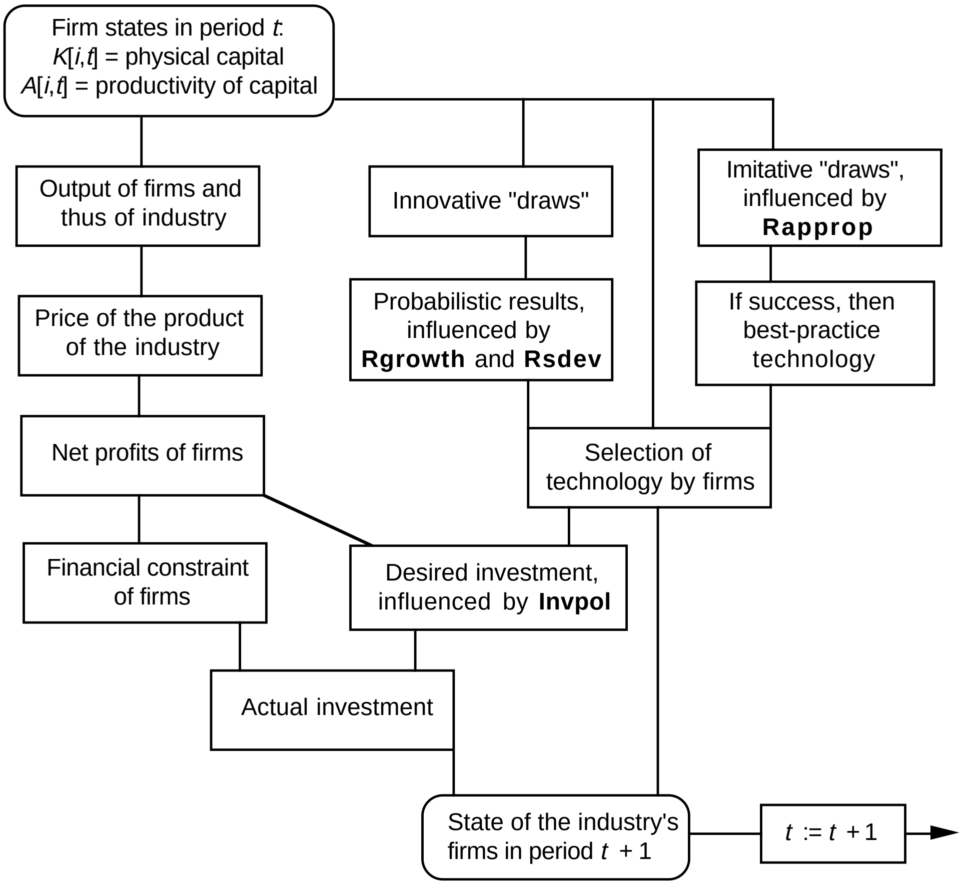

Figure 3. The computational structure of the NELWIN78 model.

To define the exogenous variables. we need to consider the long-term aspects of NELWIN78 (see figure 3). We start by the factors of technological change. Here it should be noted that productivities are specific to individual firms. They reflect knowledge that has some degree of appropriability of the results of R&D. Technical change takes the form of process innovations and process imitations that increase the capital coefficient of individual firms $\left(A_{i,t+1} \geq A_{it}\right)$. The processes whereby new production techniques are found and productivity is changed include the following steps:

The firms' costs of innovative R&D are found by fixed decision rules that determines them in proportion to the level of physical capital. The firm's chance of getting an innovation is proportionate to its innovative R&D costs, and it is determined stochastically by the exogenously given character of technical change of the industry. The productivity of the innovation is also determined by an exogenously given normal probability distribution. The normal distribution has a mean defined by the exponentially growing science-based state-of-the-art (**Rgrowth**). The standard deviation of the distribution (**Rstdev**) is fixed, and the result is transformed back from the log-form to an ordinary productivity.

The firms' costs of imitative R&D are found by fixed decision rules that determines them in proportion to the level of physical capital.[11] Because of its imitative search effort, each firm has a probability of obtaining a success – the probability is proportionate to its imitative search costs but is otherwise determined by exogenous factors (the difficulty of imitation in the particular industry, **Rapprop**). An imitative success means that the firm gets access to the best-practice technique and thus the highest productivity level obtained by any firm in the period.

To understand investment behaviour, we need to return to the short-run functioning of NELWIN78. As already mentioned, firms use a full-capacity rule in the determination of their output. The aggregate output of the industry faces exogenously given demand conditions that are characterised by unit elasticity, i.e. the same total revenue (D) is obtained by the industry no matter how much or little it produces. In other words, price adapts to clear the market: $P_t = D / Q_t$. To determine whether their production capacity should be changed, firms apply mark-up pricing like Cournot oligopolists (see Scherer and Ross, 1990, ch. 6). If the concentration ratio (measured by the Herfindahl index) is high, then firms presuppose a high mark up while the mark up approaches 1 as the degree of concentration decreases.

To see the mechanism, we start by calculating turnover $\left(P_t Q_{it} = P_t A_{it} K_{it}\right)$ and then find the net profit by deducting the costs elements (which include costs of innovative and imitative R&D). In NELWIN78 costs per unit of capital are constant while costs per unit of output decrease as productivity increase. The firm's desired investment is determined by the unit costs and a mark-up factor influenced by the market share of the firm (set by the parameter **Invpol**). The maximum investment of a firm is determined by the profits of the present period plus loans from the banks in proportion to the profits. Actual investment is the minimum of desired and maximum investment.

4. NELWIN78: simulations with varying parameters

4.1. Running NELWIN78 (NW78binary)

NELWIN78 is a part of the DRUIDIC system, a system of computer programs designed for evolutionary-economic modelling The DRUIDIC system is programmed in the MAPLE computer language[12] (designed for symbolic and numerical mathematics), but simple experiments with the NELWIN78 program can be performed without any knowledge of the MAPLE language. A DRUIDIC exercise starts by starting the system (MAPLE) and by starting the specialised DRUIDIC interface – by typing **start();** when an empty MAPLE document has been opened. The DRUIDIC system answers (see the 'dialogue' below); it tells how to get help and abbreviations, and how to turn off the information on the time taken by the simulation.

```
==in=> start();
   =out=>
          DRUIDIC models
          Revision 19 Mar 96
          Type ?info for help - or ?nw78
          Type short; for short names
          Type timeoff; to remove time and bytes
```

Then we call the procedure that implements the NELWIN78 model. This procedure is called **NW78binary** since it is called with a series of binary parameters or 'arguments'. The call has the general form **NW78binary(T, Firmno, Invpol, Rapprop, Rgrowth, Rsdev);**. where **T** is the number of periods, **Firmno** is the number of firms, **Invpol** is a parameter determining the mark-up factor, **Rapprop** is the difficulty of imitation, **Rgrowth** is the exogenous growth of scientific knowledge, and **Rsdev** is the standard deviation of the outcomes of innovative successes. The parameters set by the binary arguments are recorded in table 2. The table also includes information of N&W's choice of size of the parameters for the high and low case, but this information can only be evaluated on the background of the rest of the NELWIN78 model (see below).

Table 2: Parameters reset by the arguments of NW78binary

T	number of periods (quarters of years) in the simulation
	defined by **T**
_[Firms]	number of firms
	defined by **Firmno**
	high = 16, low = 4
_[eta]	lack of aggressiveness in investment strategy
	defined by **Invpol**
	high = 1000, low = 1
_[Rim]	probability of imitative success per unit of capital
	defined by **Rapprop** = difficulty of imitation
	high = 0.005, low = 0.0025
_[phi]	rate of latent productivity increase per quarter
	defined by **Rgrowth**
	high = 0.015, low = 0.005
_[sigma]	standard deviation of log(A) around log(latent productivity)
	defined by **Rsdev**
	high = 0.178, low = 0.018
_[seed]	integer for initialising the random generators
	defined by **seed**
Non-changed parameters:	
_[b]	ratio of external financing to economic profit (= 1)
_[c]	production cost per unit of capital (= 0.16)
_[delta]	depreciation rate per period (quarter) (= 0.03)
_[Dtot]	total revenue of the industry (=64)
_[Rin]	probability of imitative success per unit of capital (= 0.0025)

To sum up: we make a procedure call by a procedure name (short alias: **nw78** referring to the long name: **NW78binary**) followed by 5 binary arguments in parenthesis; finally the command is ended by a semicolon. We start by making a simulation for 20 periods (5 years) with all parameters set to low. After some time the DRUIDIC system answers that the run is complete. Furthermore, it gives information on the run-time of the simulation (in this case 14 seconds) and the number of bytes used in the process (in this case a little lest than 1 MB).

```
==in=> nw78(20,0,0,0,0,0);
   =out=>
           .9919534550

           Info on output 1:   Time:   14.23   Bytes:   937072
```

Now the simulation results are ready, but the data are located in the computer memory (as ‘globally’ accessible variables[13]). They can be inspected by MAPLE’s standard methods or through specialised DRUIDIC procedures (see appendix 1).

4.2. Industry-level analysis

We are now ready for studying the outcome of the simulation at the industry level. To do so we need to know the names used in **NW78** for the industry-level variables. These variables are depicted in table 3.

Table 3: Industry-level variables

AVpr[*t*]	average productivity of firms
HHk[*t*]	reciprocal of the Herfindahl index calculated by capitals
HHs[*t*]	reciprocal of the Herfindahl index calculated by market shares
AVkap[*t*]	average capital of firms
TK[*t*]	total capital
P[*t*]	price of output
AGAP[*t*]	ratio between actual and potential output
Qtot[*t*]	total output of the industry
Rmax[*t*]	maximum productivity of the industry
Rmean[*t*]	latent, science-based productivity of the industry

Let us start by studying production (**Qtot**) and price (**P**) for a 20 period simulation (5 years) – with all parameters set to low. Here we (according to appendix 1) use the procedure

DrawIndVar(variable, periods, pretty) or **driv(variable, periods, pretty)**. The last argument is optional; it tells the system to format the textual output in a 'pretty' way.[14]

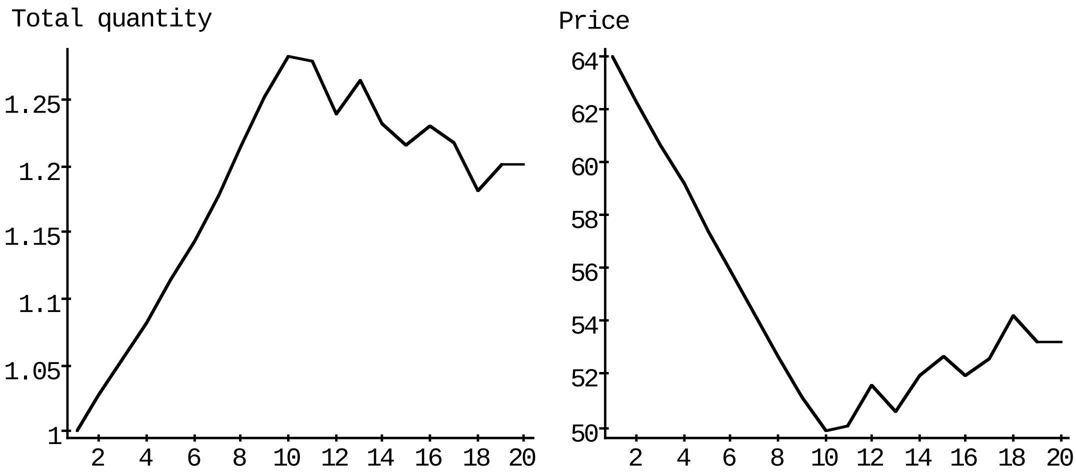

Figure 4. Total quantity and price with low parameter settings.[15]

Figure 4 demonstrates that the industry is expanding in the first half of the simulation while it is later more stable. Because of the unit elasticity of demand (see section 2.2), the price is directly reflecting changes in total output.

We now turn to a plot of the dynamics of the concentration index (the reciprocal Herfindahl index). To find this information we need to know the name used in **NW78** for this index (see table 3). The name is **HHs** for the concentration of market shares and **HHk** for the concentration of capitals.[16]

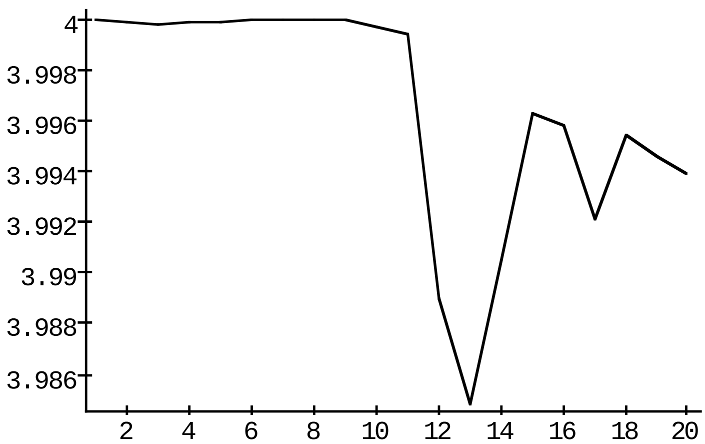

Figure 5. Reciprocal Herfindahl index for a simulation with easy imitation.[17]

Even if the reader has precisely followed the procedure above, the resulting figure will probably differ somewhat from figure 5. The reason is that a random number generator has been used by the program to decide success and failure in innovative and imitative activities. To be able to replicate an experiment precisely, the random number generator must be provided with an initial 'seed'. In the above case the system chose the seed to be 130461246940. But the system can be provided with a positive integer that functions as seed. In the following the seed has been set to 5, which is provided as an extra argument to the procedure:[18]

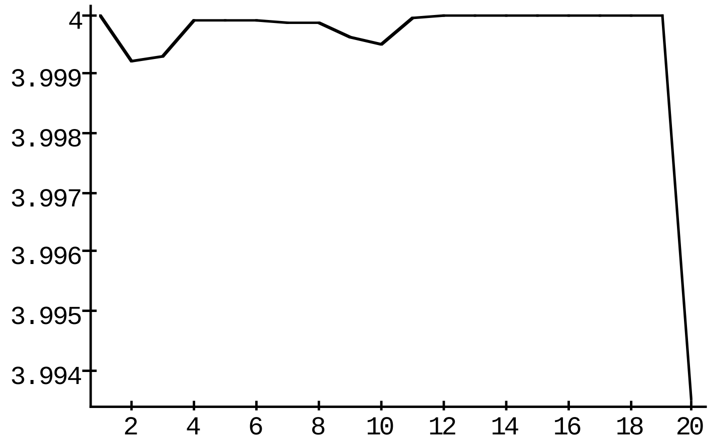

Figure 6. Reciprocal Herfindahl index for another simulation with easy imitation.[19]

We are now ready to make simple experiments. Let us take the case with aggressive investment strategy (low desired mark up) and start with easy imitation. The simulation covers 60 periods (15 years). Information on the resources used by the computation has been included to emphasise that we are now using 2 minutes and 8.5 MB per run. The latter figure indicates the total memory resources used, and some of the resources are reused so the computation does not require 8.5 MB RAM.

```
==in=> nw78(60,0,1,0,0,0,5);
   =out=>
      Info on output 12:   Time:   119.01   Bytes:   8491428
```

After the run we turn to data analysis. In figure 7 we present the results for concentration (as above) and for the productivity gap (the variable **Qgap**).[20]

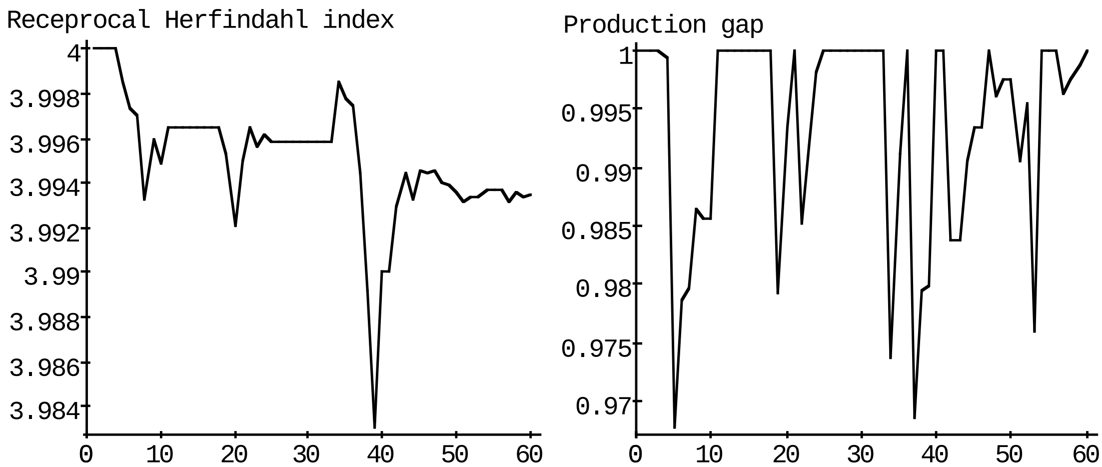

Figure 7. Concentration and production gap for simulations with easy imitation.[21]

Underlying figure 7's changes in concentration and production gap are innovations and imitations, but changes are small. Figure 8 (below) shows the case where imitation is difficult (and appropriability of innovative results is high), and here the long-term picture is different. Now the concentration index drops to about 3.8 while it did not come below 3.98 in the easy-imitation case. Similarly, the production gap index swings around 0.98 while it was clearly higher in the former case.[22]

The simulation results depicted in figure 8 do not demonstrate any 'dramatic' results, and this is due to N&W's choice of a relatively limited variation in the difficulty of imitation. Given their setting, we obtain much more dramatic results if we vary the rate of science-based productivity growth in the case of 16 firms (see figure 9). Here initial setting of parameters is 60 periods (15 years), aggressive investment, low appropriability of innovations, slow productivity growth, and large variation of innovative outcomes:[23]

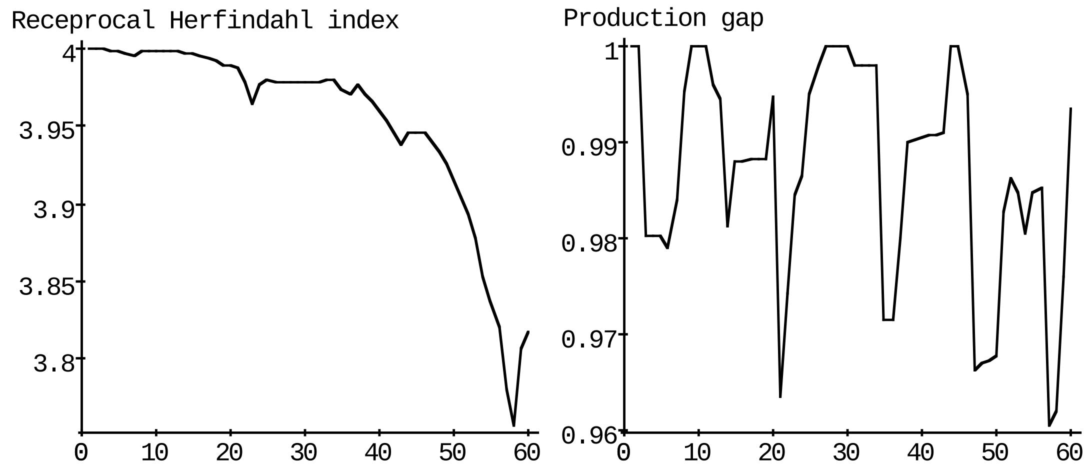

Figure 8. Concentration and production gap for simulations with difficult imitation.[24]

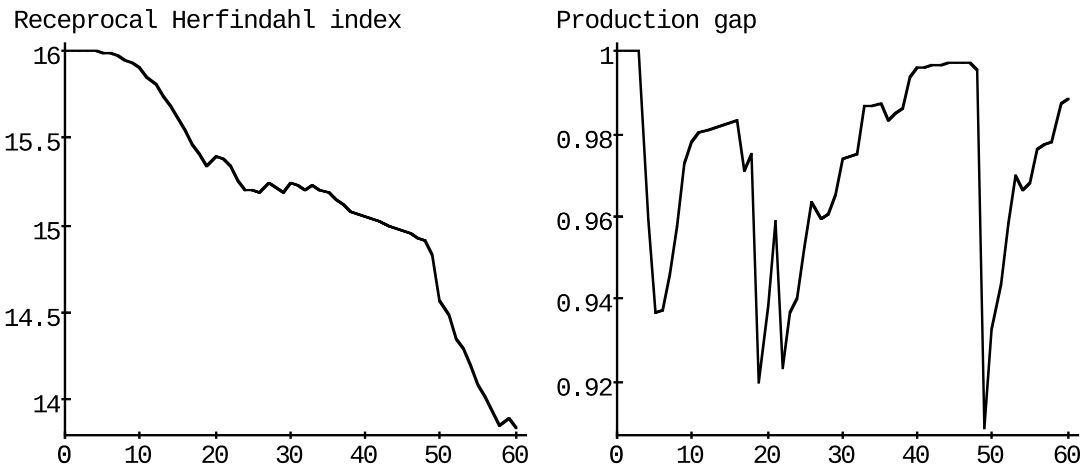

Figure 9. Concentration and production gap with slow productivity growth.[25]

In figure 9 we see how the concentration index falls from 16 to about 14 firms while the production gap swings around 0.95. In figure 10 we have changed the (latent) rate of productivity growth to high. As a result the possibilities of successful innovation increase and this gives a much higher concentration index and a higher difference between actual and potential production. Now the concentration index shows about 10 firms in the end of the simulation while the production gap index swings around 0.9.[26]

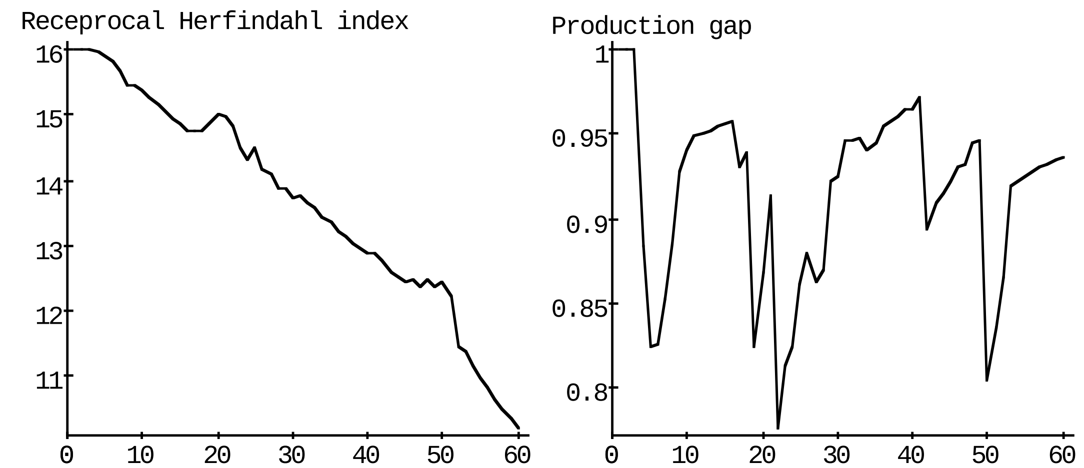

Figure 10. Concentration and production gap for simulations with difficult imitation.[27]

4.3. Firm-level results

A major advantage of N&W-style simulation is that all firm-level variables are (potentially) available for an analysis of the microfoundations of the industry-level evolution. To present the evolution of the firm-level variables, we need the variable names used by NW78. These names are explained in table 4.

Table 4: Firm-level variables

A[*i*,*t*]	productivity of firm *i*
Aim[*i*,*t*]	not available for inspection without change of the program
Ain[*i*,*t*]	not available for inspection without change of the program
Ides[*i*,*t*]	not available for inspection without change of the program
Imax[*i*,*t*]	not available for inspection without change of the program
K[*i*,*t*]	capital of firm *i*
pi[*i*,*t*]	profit of firm *i*
Q[*i*,*t*]	not available for inspection without change of the program
s[*i*,*t*]	market share of firm *i*

First we look as the firm variables for the simulation with difficult imitation.[28]

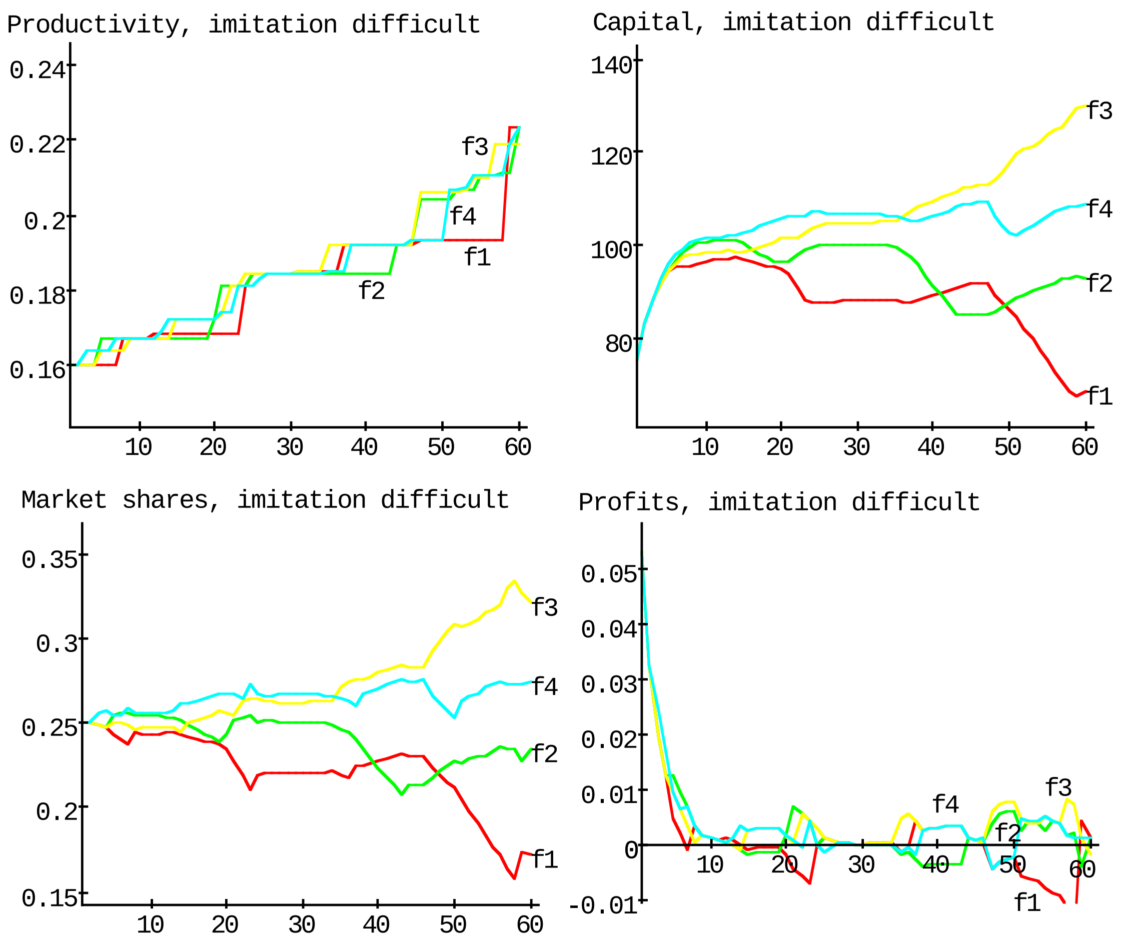

Figure 11. Firm variables for the simulation with difficult imitation.

In figure 11 all 4 firms start with the same productivity (0.16) and the same capital nearly 80). Consequently, they have the same market shares (0.25) and the same profits (about 0.05). Since profits are larger than required by the desired mark up over price, firms expand their capital during the first periods. However, their situation is quickly changing because the success and failure of firms in innovative and imitative activities. Firms 3 and 4 tend to be the lucky ones which means that they increase their productive capacity and thus their output and market share. They also uphold or expand their capital. On the other hand firms 1 and 2 tend to be the unlucky ones with decreasing market shares, under normal profitability and decreasing capital. Especially firm 1 is a ‘laggard’ – except in the very end where it is the first firm to obtain a productivity of more than 0.22. This poor imitative/innovative performance is reflected in market shares, profitability and capital accumulation.

In general figure 11 shows an industry with a relatively high variety of performances of firms. This impression is reinforced when we compare with figure 12 that shows the results for the simulation with easy imitation.[29]

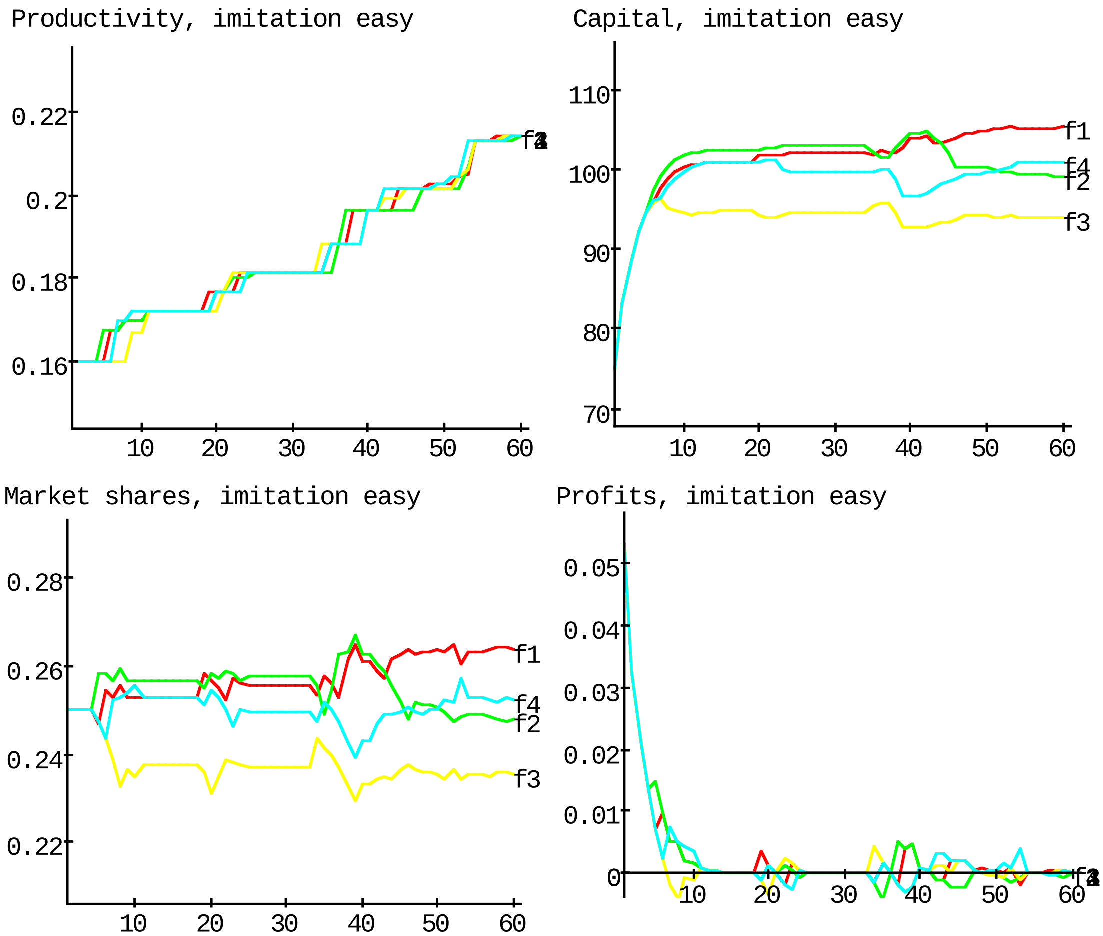

Figure 12. Firm variables for the simulation with easy imitation.

In figure 12 it is difficult to follow the development of productivity and profits. This calls for data in tabular form (see appendix 1).[30]

Table 5: Productivity of 4 firms for selected periods

(a) Imitation is difficult.[31]

—	1	2	3	4
1	.16	.16	.16	.16
10	.168	.168	.168	.168
20	.168	.172	.172	.172
30	.184	.184	.184	.184
40	.192	.184	.192	.192
50	.194	.204	.206	.194
60	.223	.223	.219	.223

(b) Imitation is easy.[32]

—	1	2	3	4
1	.16	.16	.16	.16
10	.170	.170	.167	.172
20	.177	.177	.172	.177
30	.181	.181	.181	.181
40	.196	.196	.196	.196
50	.202	.201	.201	.202
60	.214	.214	.214	.214

Table 5 shows the productivity of each of the 4 firms for selected years in the case of difficult and easy imitation. In the difficult-imitation case all firms have the same productivity in 3 of the selected years while it is 4 in the other case. We also see other differences but it is clear that some of the variety demonstrated by the figures are difficult to see in the tables. The same advantages and limitations of a tabular summary can be seen in table 6 where the profits of firms are recorded.

Table 6: Profits of 4 firms for selected periods, imitation is easy.

(a) Imitation is difficult.

—	1	2	3	4
1	.053	.053	.053	.053
10	.0012	.0012	.0012	.0012
20	-.0019	.0016	.0016	.0016
30	.000058	.000058	.000058	.000058
40	.0031	-.0036	.0031	.0031
50	-.0025	.0059	.0079	-.0025
60	.0012	.0012	-.0020	.0012

(b) Imitation is easy.

—	1	2	3	4
1	.053	.053	.053	.053
10	.0015	.0015	-.0012	.0038
20	.0012	.0012	-.0033	.0012
30	.000040	.000040	.000040	.000040
40	.00079	.00079	.00079	.00079
50	.00066	-.00018	-.00018	.00066
60	.000059	.000059	.000059	.000059

4.4. Statistical analysis (NW78stats)

The data clearly calls for a systematic statistical analysis. This can most easily be obtained by exporting the data (by the procedure DATAexport, see appendix 1) and processing them in a suitable program (e.g. SAS). But this kind of data analysis can also be performed within MAPLE. Here the program NW78stats (see appendix 2) has been used.

Table 7. N&W and NW78 final-period concentration, 101 periods and 4 firms

Experimental condition (binary code)	Reciprocal Herfindahl index N&W's single value (1978, 533)	Reciprocal Herfindahl index Mean of 5 runs with NW78binary
0000	4.000	3.999 (.0015)
0001	3.9995	3.993 (.0011)
0010	3.998	3.994 (.0057)
0011	3.973	3.981 (.0233)
0100	4.000	3.994 (.0106)
0101	3.997	3.994 (.0052)
0110	3.978	3.988 (.0083)
0111	3.998	3.977 (.0212)
1000	3.976	3.961 (0.0446)
1001	3.719	3.894 (0.1189)
1010	3.611	3.526 (0.4149)
1011	3.794	3.557 (0.3436)
1100	3.701	3.780 (0.2309)
1101	3.849	3.553 (0.2253)
1110	2.353	3.378 (0.3997)
1111	2.489	3.303 (0.5070)

In the following we present final-period concentration (measured by the reciprocal Herfindahl index). In each case we make 5 runs with **NW78binary** and take the mean value as well as the standard deviation. These calculations are made by **NW78stats**. The purpose is partly to demonstrate the properties of the model, partly to control how close the model can replicate the behaviour of the results of the model presented by N&W (and Schuette) in 1978. Concerning the latter purpose it should be noticed that we have not been able to include a few, poorly documented aspects of N&W's model.[33] Another problem is that N&W

has only made one run for the 4-firm case and 2 runs for the 16-firm case. This means that random deviations may play an important role (see e.g. the big difference between the two 16-firm runs of the 0101-case in table 8).

In table 7 we present the results for the 4-firm cases. The interpretation of the table presupposes that the specification of the binary definition of parameters from figure 2 and table 2 is consulted. To summarise we may call the codes $B_1B_2B_3B_4$. B_1 is Rapprop = the difficulty of imitation, B_2 is Rgrowth = the science-based growth of the mean of innovative results, B_3 is Rsdev = the constant variability of the research results, B_4 is Invpol = the aggressiveness of investment policy. If all parameters are low (i.e. easy imitation, slow scientific growth, small standard deviation of innovative results and monopolistic restraint with respect to investment (Cournot behaviour). To change the investment behaviour has practically no influence on the concentration ratio (but influences the speed by which it is reached). On the other hand, the change of more binary digits gives higher concentration and higher variability of the outcomes.

In table 7 we see that all runs characterised by $B_1 = 0$, there is a good correspondence between N&W's single run and our mean of 5 runs. The only case where we get a clearly different result is 0110. However, no clear conclusion can be drawn from this deviation.

Table 8. N&W and NW78 final-period concentration, 101 periods and 16 firms

Experimental condition (binary code)	Reciprocal Herfindahl index N&W's 1st value (1978, 533)	Reciprocal Herfindahl index N&W's 2nd value (1978, 533)	Reciprocal Herfindahl index Mean of 5 runs with NW78binary
0000	14.925	15.060	12.42 (5.404)
0001	14.347	14.286	13.94 (.6138)
0010	12.005	12.019	12.94 (0.7313)
0011	12.516	13.514	12.65 (0.9686)
0100	13.072	13.495	13.38 (0.7180)
0101	14.045	10.741	12.82 (0.6223)
0110	10.776	9.579	10.84 (0.9530)
0111	11.050	8.418	10.12 (1.909)
1000	12.937	13.158	12.08 (1.400)
1001	13.550	13.228	11.35 (.9852)
1010	7.429	7.788	8.802 (2.480)
1011	6.361	4.938	7.464 (1.653)
1100	10.893	11.001	9.990 (1.790)
1101	10.091	8.058	8.962 (2.179)
1110	6.150	5.102	6.829 (1.872)
1111	2.856	4.686	4.233 (1.233)

The results of the runs for 16 firms can be seen in table 8. In the first column of this table the codes of the different experiments is given. In the second and third column N&W's results for 2 different runs are recorded, In the fourth column we present the results of the runs with NW78binary. It is obvious that we do not get the same results that N&W. Especially 0000 and 1001 give significantly different results. However, the ranking of the effects of the different settings of the parameters are only changed in these cases.

5. NELWIN77 and 84: varying program elements and parameters

5.1. Combining parameter variation with model/program development

Until now we have followed the usual way of making computer experiments (see the right cycle of figure 13): (1) Start from an existing model and a related computer program, (2) set the parameters of the model, (3) ‘run’ the model, (4) analyse the results. To develop the analysis more systematically we normally (5) find a setting of the parameters that implied a more or less equilibrated state of the model provided that no innovation takes place, (6) study the consequences of changing different parameters, (7) make a statistical analysis that removes the arbitrary character of individual runs influenced by stochastic phenomena.

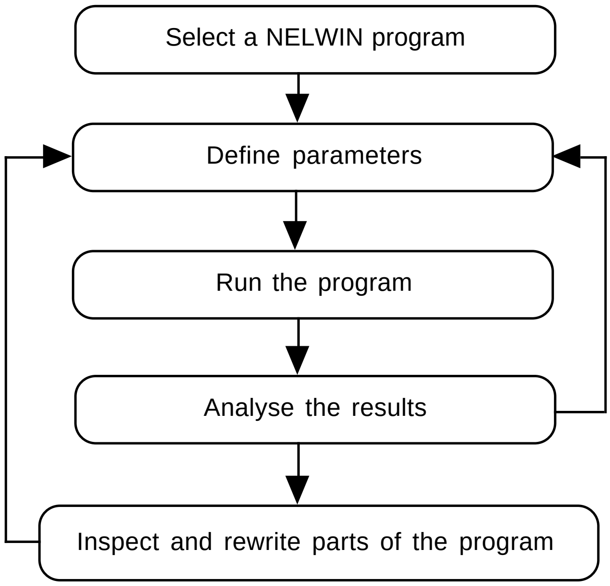

Figure 13. The programming cycles.

These steps are well-understood from, e.g., macroeconomic computer simulation but the crucial step is missing. Often the most obvious reaction is to revise your basic theory and behavioural model rather than just to change the parameters of a given model. But this is normally impossible because of the sheer complexity of the task and for the practical reason that you have to leave the work with the programmed version of the model, reprogram the model (in, say, FORTRAN or PASCAL), recompile the result and hope that the new version will function when you sets its parameters and runs it. The practical problem has, however, been easy to solve for some time. The reason is that an interactive approach to the reprogramming has been developed, not least by the Artificial Intelligence community who has from the very beginning found out that a computer system which can interpret the programming statements continuously and immediately allow the inspection of the results is well-fit for creative analytical work. At the same time it is important that the programs (procedures) can be manipulated by (meta)programs which takes simple programs as ‘data’. This flexibility is well known from the programming language LISP (and the related dialects like SCHEME, cf. Abelson and Sussman, 1985). Much of this flexibility is also found in MAPLE which combines mathematical and modelling power with the interactive approach known from, e.g., Artificial Intelligence experiments. The DRUIDIC programming environment has been designed to take advantage of this fact. In relation to figure 13, we propose to emphasise the left part of the programming cycle.

5.2. Simple simulations with NELWIN77 (NW77param and NW77stats)

Before we turn to a discussion of NELWIN77 and its possible changes (including NELWIN84), it might be helpful to check out the functioning of the program (see appendix 3). The program is called **NW77param** since it is very flexible with respect to making parameter changes. The calling sequence is **NW77param(N, T, par1=x, par2=y, ...)**, where **N** is the number of firms, **T** is the number of periods, **par1** is the name of the first parameter to be changed and **x** is its value, **par2** is the name of the second parameter to be changed and **y** is its value, etc. One of the parameters to change is **seed** which has been discussed earlier. In these runs there is normally a split between firms so that only half are innovators while all firms are imitators.

In the following we present a simulation for 8 firms and 101 periods with difficult imitation (dim=0.5 instead of normally 1.25) .[34] The result should be that innovative firms perform relatively better than firms that are only imitative. The firm-level results of the experiment are depicted in figure 14.

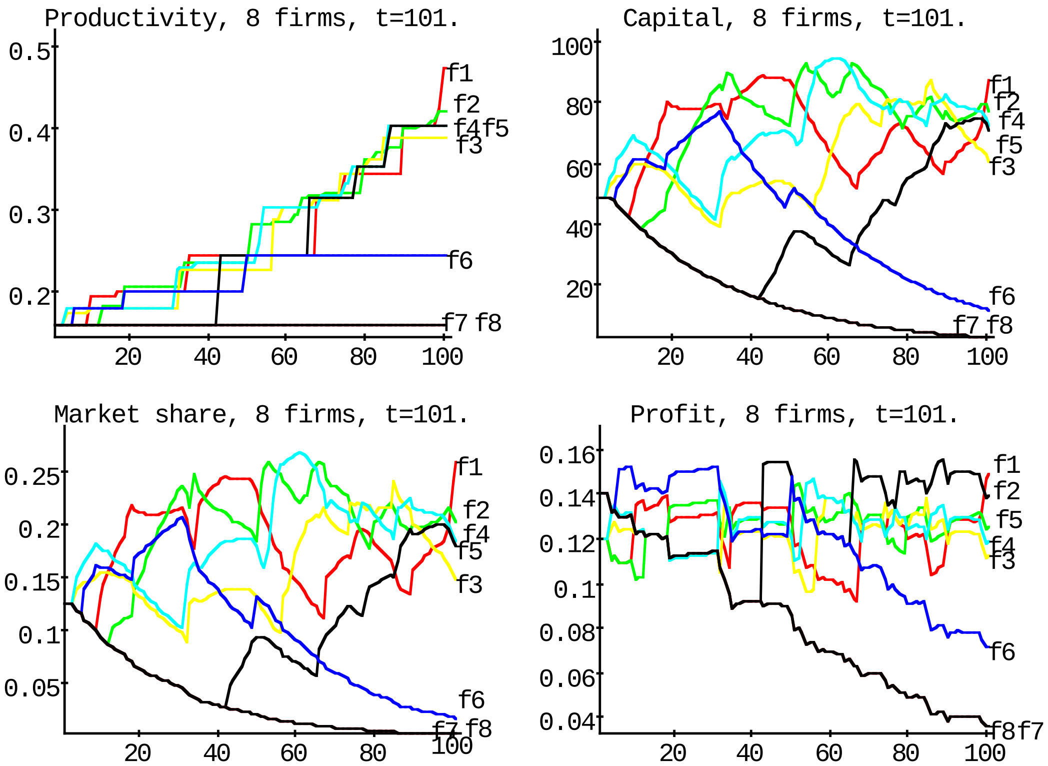

Figure 14. Firm variables for a NELWIN77 simulation with difficult imitation.[35]

As expected it is largely the innovative firms (#1-4) that performs best with respect to profitability in runs with difficult imitation. The reason is that the innovative firms tend to have the highest productivity and the largest incentives and possibilities for expansion. However, one of the imitative firms (#5) has achieved to reach the level of the innovative firms.

Table 9. Final-period data for a run over 101 periods with 8 firms[36]

	PRODUCTIVITY IN PERIOD 101	CAPITAL IN PERIOD 101	PRODUCTIVITY IN PERIOD 101	CAPITAL IN PERIOD 101
	seed = 6		seed = 1	
INNOVATIVE FIRMS:				
FIRM 1	.4735	87.0268	.4080	75.0227
FIRM 2	.4188	77.0003	.4616	99.3675
FIRM 3	.3889	60.7948	.4128	85.4428
FIRM 4	.4037	72.7430	.4240	88.4737
IMITATIVE FIRMS:				
FIRM 5	.4037	70.8304	.16	3.3840
FIRM 6	.2443	11.4799	.2242	9.3942
FIRM 7	.16	2.5454	.3479	14.5641
FIRM 8	.16	2.5454	.16	3.3838
AVERAGE PRODUCTIVITY	.3316	48.1208	.3248	47.3789
STANDARD DEVIATION	.1242	36.1091	.1163	40.3040

The results of the experiment is also recorded in table 9[37] – together with another experiment with the same parameters except that **seed** has been changed. Both runs demonstrate that innovative firms are generally performing better than imitative firms under the given setting of the parameters.

In the following we present some runs with **NELWIN77stats** (see appendix 3). This procedure is called with the following arguments: **NW77stats(R,n,T,SEED)** where **R** is the number of runs for which a mean is calculated, **n** is the number of firms, **T** is the number of periods and **SEED** is the initial integer governing the random number generators.

The aim is to investigate whether our reconstruction of N&W's 1977 model (NELWIN77stats) gives the same results as has been recorded by N&W (1977, 84-85, 1982, 302). As N&W we have chosen to make 5 runs with different seeds over 101 periods (25 years). We shall register the end result in period 101 in the 5 runs. On that background we calculate mean value and standard deviation. Like N&W we make calculations for 4, 8 and 16 firms.[38] The results of our runs as well as N&W's data are recorded in table 10. Here are recorded means and standard deviations (in parentheses) based on final-period characteristics of 5-run samples.

Table 10. N&W and NW77 final-period characteristics, 101 periods and 4, 8 and 16 firms

	N&W (1977, 84-85)	NW77stats	N&W (1977, 84-85)	NW77stats	N&W (1977, 84-85)	NW77stats
	4 firms		8 firms		16 firms	
Herfindahl index for capital	3.6062 (0.3018)	3.854 (0.161)	5.9542 (0.4683)	5.806 (0.697)	7.430 (0.9386)	8.696 (0.699)
Average productivity	0.4032 (0.0245)	0.4231 (0.0138)	0.3776 (0.0136)	0.418 (0.0066)	0.3194 (0.0060)	0.395 (0.0210)
Best-practice productivity	0.4403 (0.0063)	0.442 (0.0127)	0.4468 (0.0013)	0.454 (0.0059)	0.4685 (0.0277)	0.451 (0.0254)
Price	0.4567 (0.0133)	0.444 (0.0137)	0.4124 (0.0064)	0.427 (0.0059)	0.4052 (0.0155)	0.435 (0.0263)

Basically we have obtained the same results as N&W. Exceptions are seen on three issues: average productivity (8 firms), average productivity (16 firms) and Herfindahl index (16 firms). In all cases we obtain larger values than N&W. One reason may be the existence of errors in N&W's choice of technologies.

5.3. The computational structure of NELWIN77

To be able to make inspection and rewriting (see figure 13), we need a program that is not too specialised in relation to parameter experiments (like NW78binary). This means that NELWIN77 (see section 1) is a relatively good choice. The NELWIN77 program is basically an expression of the standard NELWIN model (see section 2 and figure 1). However, to be able to modify the program we need a closer look at its main elements. They are depicted in figure 15.

Basically figure 15 depicts a computational structure that takes a set of state variables for time *t* and transforms them into a set of state variables for time *t* + 1. This process is repeated for as many periods as requested by the user of the program.

The most important problem is which elements of the program are defined as state variables. In the simple NELWIN models only capital and productivity are state variables, and only productivity undergoes an evolutionary transformation. However, N&W (1982, 19) clearly think that this is just a simple example of the 'vast array of particular models can be constructed within the broad limits the theoretical schema' that they are developing. The extendibility of their approach can most clearly be seen when considering changes in the set of state variables (Andersen, 1994, ch. 4). One possibility is to consider behavioural rules as evolving variables (e.g. the propensity to do research (rin) evolves in the model by Silverberg and Verspagen, 1994). Another possibility is consider A[i,t] as a vector of productivities relating to the many different tasks performed by a firm (like in the model proposed in Andersen, 1996a, forthcoming-a, forthcoming-b, and Andersen&Lundvall, 1996).

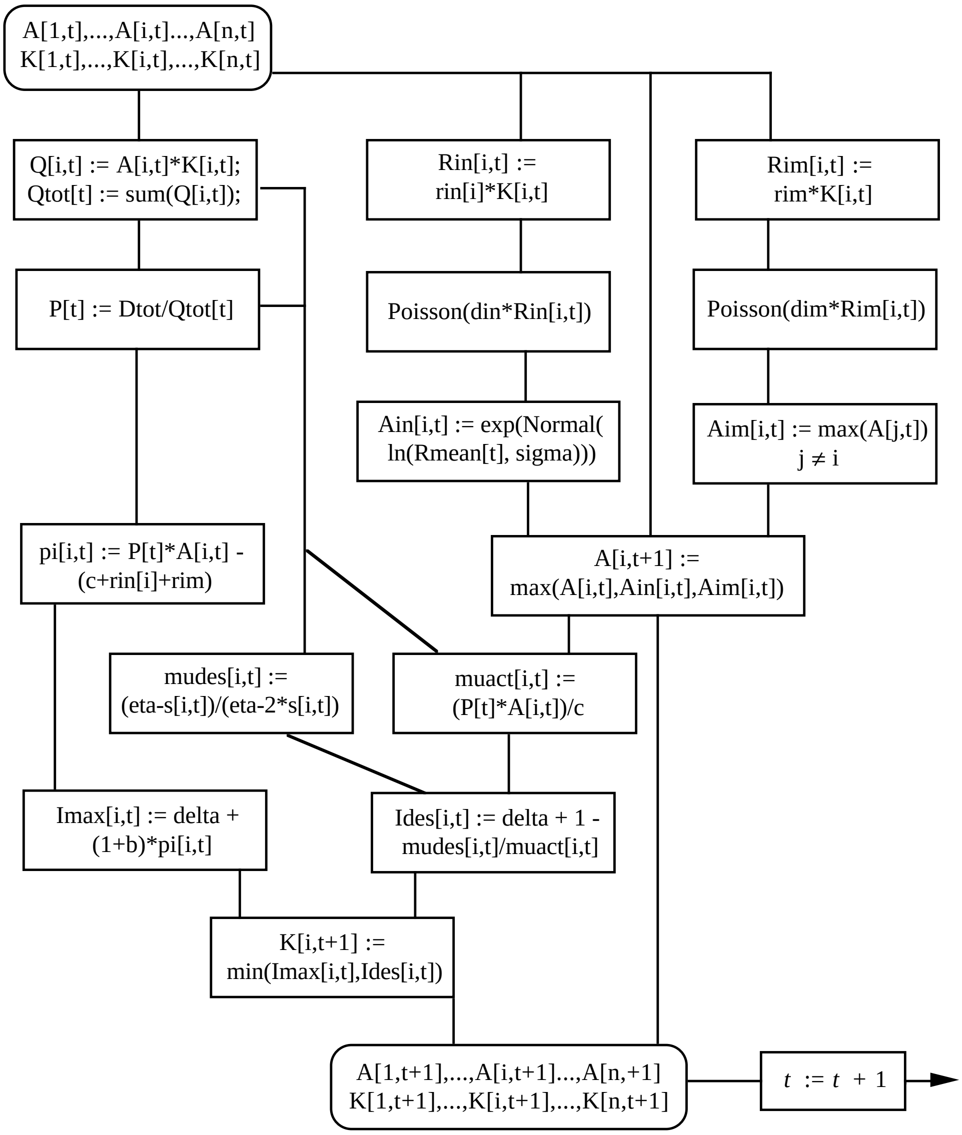

Figure 15. The main computations of the NELWIN77 model.

5.4. Short run: elements and changes

We now turn to the **NELWIN77** and partly the **NELWIN84** models, which will be dealt with in the this sections. First we initialise the state variables. This can most conveniently be made in a way that secures that the industry is in equilibrium unless innovations take place. This problem is dealt with in the **NW77param** program in appendix 3.

Then we come to the short-term market process. Figure 15 can be specified in the following way for the 4-firm case:

```
for i from 1 to 4 do
      Q[i,t] := A[i,t]*K[i,t];                                    (1)
od;
Qtot[t] := sum(Q[i,t], i = 1..4);                                 (2)
P[t]:= 67/Qtot[t];                                                (3)
```

We see (1) that a very simple capacity utilisation rule is used (output = capacity). Demand for the output is characterised by a demand function with unit elasticity, i.e. the same total revenue (=67) is obtained by the industry no matter how much or little it produces (3). The relationship is shown in figure 16.[39]

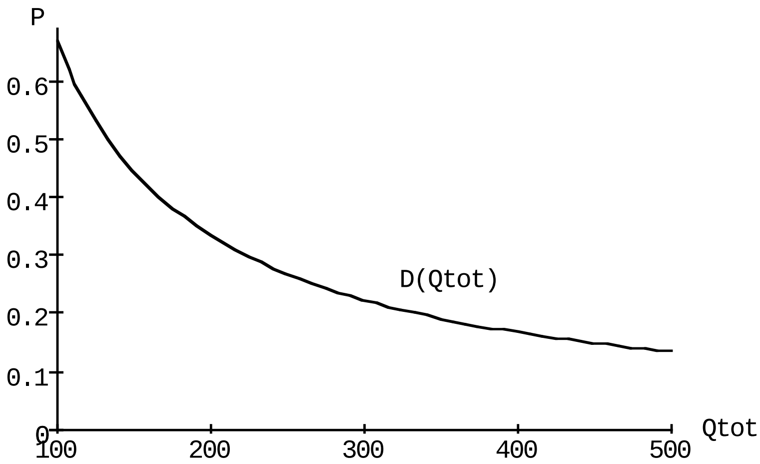

Figure 16. Unit elasticity of demand.

However, we may also want to experiment with cases where total revenue is increasing through time in a logistic manner. The background may e.g. be that here has been a product innovation before the start of the logistic curve. This behaviour of the market for the product of the industry can e.g. be specified as a logistic curve starting from a basic level of demand – is in figure 17.[40]

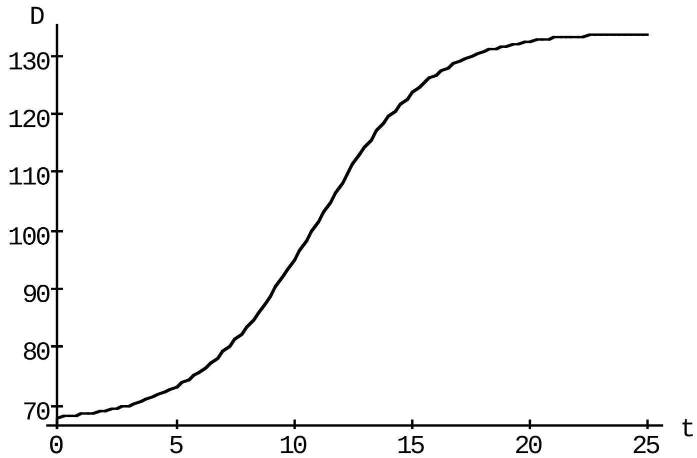

Figure 17. A logistic curve added to a base level of total revenue.

5.5. New technology: elements and changes

In NELWIN77 the determination of the amounts of resources (per unit of capital) that a firm devotes to innovative and imitative R&D are determined by fixed rules. In figure 15 the propensity to do research is specific to each firm although it is constant through time (rin[i]). In NELWIN77 only half of the firms do any innovative R&D while all firms imitate (N&W, 1982, 302). Furthermore, the total amount of R&D expenses is set to a fixed value which reflects the R&D intensity of the industry. Given the total demand of 67, total innovative expenses is set to 4 and total imitative expenses is set to 0.4 (N&W, 1977, 76). The expenses per unit of capital can be found after the overall amount of capital is found. For instance, if total innovative expenses (TRin) and total initial capital (TK[1]) are known, we have:

```
rin := TRin/TK[1];
```

Table 11: MAPLE V Rev2's procedures for random number generation[41]

```
rand(range): With no arguments, a call returns a 12 digit non-negative random
     integer. With an integer range as an argument, the call rand(a..b)
     returns a procedure which, when called, generates random integers
     in the range a..b.
RandUniform(a..b): Uniform distribution on [a,b). [The remember option
     might be applied in a modified version.]
RandNormal(mean, sdev): Normal distribution with mean and standard deviation
     (sdev). I[The remember option might be applied in a modified version.]
RandPoisson(lambda, digits): Random number generator for the Poisson
     distribution where lambda is the mean number of occurrences of an event
     per unit time. The option argument (digits) specifies the number of
     digits. The algorithm is only suited for small lambda.
randmatrix(m, n, entries = f) in the linalg package of MAPLE generates a
     random matrix of dimension m by n. The equation entries = f specifies
     that the function f (with no arguments) is to be used to generate
     the matrix entries.

MAPLE V Rev3 (Windows):
     RandNormal(mean, sdev) is exchanged by normald[mean, sdev]
     RandPoisson(lambda, digits) is exchanged by poisson[lambda, digits]
```

Given these data, the outcome of the search process for the firms during a period can be calculated. However, before we turn to this part of figure 15, it should be noted that to implement the procedures depicting technical change, we make heavy use of the in built (but modifiable) random number generators of MAPLE system. For convenience, we start with a short overview in table 11.

The MAPLE random number generation procedures refers to a global variable, _seed, which is set to the last random number which has been generated.[42] The setting of the parameter is done based in the seed provided as an argument to NELWIN77. Since the generation of random numbers is strictly deterministic, this means that each run with the same initial seed value will give exactly the same result, *ceteris paribus* (i.e. with no parameter change, etc.).[43]

Now we are ready to summarise the steps determining the productivity of firms in period t + 1. We start by the activities related to innovation:

```
Amean[t] := Amean[t-1]*(1 + phi);                                    (1)
NormalDistr := RandNormal(ln(Amean[t]), sigma);                      (2)
Ain[i,t] := 0;                                                       (3)
lambda := din*rin[i]*K[i,t];                                         (4)
Rdraws := RandPoisson(lambda);                                       (5)
Rdrawsno := Rdraws();                                                (6)
if Rdrawsno > 0 then                                                 (7)
    for j from 1 to Rdrawsno do                                      (8)
        Rvalue := exp(Rdistr());´                                    (9)
        Ain[i,t] := max(Ain[i,t], Rvalue);                          (10)
    od;
fi;
```

Equation (1) emphasises that we are dealing with the science-based case where the mean productivity of process innovations in a given period is determined by an exogenously given state of public knowledge. This knowledge is growing exponentially with the growth rate phi. In equation (2) we see that the normal distribution of the innovative results has its mean in the log of the productivity given by public knowledge and a standard deviation that is constant over time. Cumulative technology is an alternative formulation which totally changes the behaviour of the model (see both NELWIN77 and NELWIN84). It is simply obtained by exchanging Amean[t] with A[i,t] as the determinant of the mean of the normal distribution of innovative results.

Before we make use of the normal distribution, we have to find out whether the firm has obtained an innovative draw. These draws are considered as a stochastic Poisson process in time like the processes known from queuing theory (5). We know the average number of innovations per unit of time (and per unit of R&D) and we get actual numbers that show a Poisson distribution. The probability of obtaining an innovation is influenced by the R&D effort (rin[i]*K[i,t]), cf. (4). The lottery is called Rdraws and it may give 0, 1, 2 or more successes (6). For each of the successes[44] the normal distribution defined in equation (2) is consulted in equation (9). The outcome of several innovative successes is the best of them (10).

The specification of process of innovation includes a small extension compared to NELWIN77 and other NELWIN models – namely that more than one outcome of the Poisson lottery is allowed. But there are many other ways of extending the formulations. For the moment we shall, however, turn to imitation which looks like a shortened version of innovation:

```
Amax := max(seq(A[i,t], i = 1..n));                        (11)
lambda := dim*rim*K[i,t];                                   (12)
Rdraw := RandPoisson(lambda);                               (13)
if Rdraw() > 0 then Aim[i,t] := Rmax;                       (14)
else Aim[i,t] := 0;                                         (15)
fi;
```

Here a firm obtain a chance of an imitative ‘draw’ that is proportional to the amount of resources it devotes to imitative R&D (rim[i]*K[i,t]). If the firm is successful, it imitates the leading firm of the industry. This formulation of the process of imitation is obviously very simplistic, and things get even worse when we consider the standard setting of the parameter dim: imitation is very inexpensive and we nearly have a pure spill-over effect from the innovators. Research in which N&W was involved (Levin et al., 1987) has clearly shown that this is not normally the case.

Now we are ready for the choice of the new technology which simply is:

```
A[i,t+1] := max(A[i,t], Ain[i,t], Aim[i,t]);               (16)
```

This means that the new technology (A[i,t+1]) is available all over the firm in the beginning of next period. This is not at all obvious. One problem is whether the productivity of different technologies can be determined precisely. As we have mentioned in a note N&W have included an error mechanism to take care of this problem. But much more importantly is that process innovations will normally reveal their full productivity potentials through a time-consuming process. However, this can be included as has been emphasised by several of the followers of N&W.

5.6. New capital: elements and changes

We now come to the new capital which can be calculated by means of K[i,t], A[i,t+1] and some of the short-run information.

First we study the desired investment which – like in the Cournot story of oligopolistic behaviour – is dependent on the market share of the firm (1). However, the formula chosen in NELWIN77 gives less monopolistic restraint with respect to investment (and thus with respect to quantity) than in the Cournot case (see below). To find the desired investment (4), we also need to know the rate of depreciation and the actual mark-up which the firm expects if the current price and its new technology is used in the next period (3).

```
s[i,t] := Q[i,t]/Qtot[t];                                   (1)
mudes[i,t] := (2-s[i,t])/(2 - 2*s[i,t]);                    (2)
muact[i,t] := (P[t]*A[i,t+1])/c;                            (3)
Ides[i,t] := delta + 1 - mudes[i,t]/muact[i,t];             (4)
```

Then comes the financial conditions for making the investments:

```
pi[i,t] := P[t]*A[i,t] - (c+rim+rin[i]);                    (5)
if pi[i,t] <= 0 then F[i,t] := 0;[45]                       (6)
else F[i,t] := b*pi[i,t];                                   (7)
fi;
Imax[i,t] := delta + pi[i,t] + F[i,t];                      (8)
```

This means that the R&D expenditures are considered as ordinary costs when calculating profits (5). Given positive profits, banks add finance in proportion to profits (6-7). This means that maximum investment can be larger than profits according to the banks’ routines of finance (b), cf. 8.

Now we can sum up the investment decision and calculate the physical capital available in the beginning of the next period:

```
I[i,t] := max(0, min(Ides[i,t], Imax[i,t]));[46]
K[i,t+1] := K[i,t]*(I[i,t] + 1 - delta);
```

One of the discussible aspects of the primitive account of the process of capital accumulation is related to the desired mark-up. The underlying problem is to define a behaviour of firms which depends on their market share, so that large firms do not expand blindly in an industry facing a demand with unit elasticity. N&W has chosen several solutions in their different models of Schumpeterian competition.

NELWIN77 with the above mentioned mark-up method.

```
mudes := (2*eta-s[i,t])/(2*eta-2*s[i,t]);                        (1)
```

NELWIN78: Here we find a possibility of standard Cournot behaviour by setting eta to 1.

```
mudes[i,t] := eta/(eta-s[i,t]);                                  (2)
```

NELWIN84 with a more general method:

```
mudes[i,t] := (eta + (1-s[i,t])*psi)/
                (eta + (1-s[i,t])*psi-s[i,t]);                   (3)
```

Figure 18 shows the different aggressiveness of investment from formula 2 with eta=1 (no aggressiveness) via formula 1 with eta=1 (medium aggressiveness) to formula 3 with eta=1 and psi=2 (rather high aggressiveness).[47]

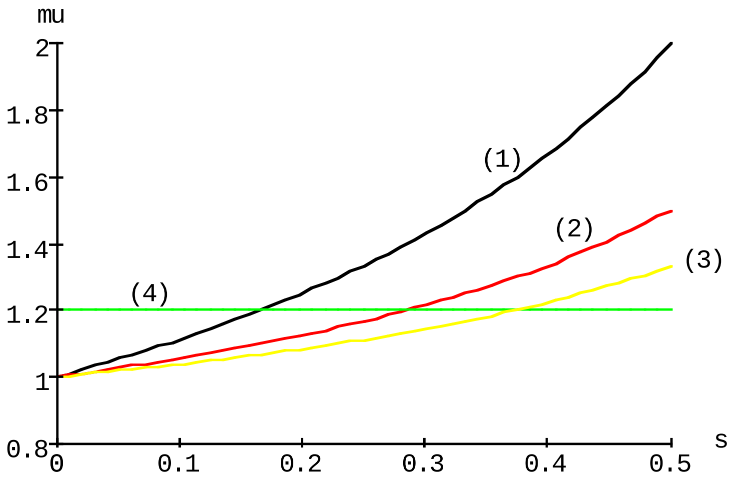

Figure 18. Formulas for desired mark-up.

In figure 18 we have also included a pricelid (4) which reflects the necessary price to break even at the base-level productivity (0.16), namely 0.16/1.2=0.1333, cf. Winter (1984, 220).

Notes

1 A variant of the N&W models is, however, underlying the FORTRAN program published by Schuette (1980, 259-270).

2 Andersen (1994) presents an algorithmic approach to evolutionary economics. One of the purposes of the present report is to make it easier to study and develop the algorithmic approach through computer excersises.

3 This is, of course, a stylised version of the background of the family of models created by N&W. They themselves give generous acknowledgements to these and other 'allies and antecedents of evolutionary theory' (N&W, 1982, 33-45).

4 The readiness of Simon's framework for an evolutionary interpretation is demonstrated by the fact that he had no difficulty in his later integration of Nelson and Winter's ideas into his own framework, see Simon, 1981, 52-57; 1983, ch. 2.

5 The reader should be aware that the empirical relevance of the whole argument is heavily dependent upon the possibility of defining a level of aggregation and a related taxonomy which are not arbitrary constructs of (national) statistical services but which instead reflect important similarities and differences with respect to the factors of the evolutionary process.

6 The general description of the standard NELWIN model fits all the NELWIN models of Schumpeterian competition while the NELWIN76 growth model differ in certain respects (technologies are characterised by productivities of labour and of capital; search is only performed when firms have unsatisfactory profits; etc.).

7 Thus, physical capital functions as the *numéraire* of the system.

8 The only way to reduce productive capacity is through the process of physical depreciation, but this is not depicted by the figure.

9 At the same time its mechanisms of technical change is slightly changed compared to figure 1 and section 2.2.4.

10 N&W, 1982, 312. On the Herfindahl index and other measures of concentration, see e.g. Scherer and Ross (1990, 70-79).

11 The costs are very small, so the industry comes near to pure spill overs from the innovators.

12 The recorded programs are programmed in Maple V Rev2. To use the programs with Maple V Rev3 or Rev4 several changes should be made. For further information see appendix 1 or the WWW pages of the DRUIDIC project: http://www.business.auc.dk/evolution/druidic/druidic.html.

13 In Maple V Rev3 and Rev4 the global variables should be declared as such before they are available (see appendix 1).

14

```
==in=> nw78(20,0,0,0,0,0);
==in=> driv(Qtot,20,pretty); driv(P,20,pretty);
```

15

$$VERSION...[Model = NW78binary\ 1.1.9.b, Note = change\ of\ seed\ is\ included, Date = 22\ Mar\ 96]. PARAMETERS...[Dtot = 64, Firms = 4, Rim = .005000000000, Rin = .002500000000, b = 1, c = .16, \delta = .03000000000, \eta = 1, \phi = .004951000000, \psi = 1, seed = 130461246940, \sigma = .01782240000]$$

16

```
==in=> driv(HHs,20,pretty);
```

17

$$VERSION...[Model = NW78binary\ 1.1.9.b, Note = change\ of\ seed\ is\ included, Date = 22\ Mar\ 96]. PARAMETERS...[Dtot = 64, Firms = 4, Rim = .005000000000, Rin = .002500000000, b = 1, c = .16, \delta = .03000000000, \eta = 1, \phi = .004951000000, \psi = 1, seed = 130461246940, \sigma = .01782240000]$$

18 For instance, figure 6 has been produced by:

```
==in=> nw78(20,0,0,0,0,0,5);
==in=> driv(HHs,20,pretty);
```

19

$$VERSION...[Model = NW78binary\ 1.1.9.b, Note = change\ of\ seed\ is\ included, Date = 22\ Mar\ 96]. PARAMETERS...[Dtot = 64, Firms = 4, Rim = .005000000000, Rin = .002500000000, b = 1, c = .16, \delta = .03000000000, \eta = 1, \phi = .004951000000, \psi = 1, seed = 5, \sigma = .01782240000]$$

20

```
==in=> driv(HHs,60,pretty); driv(Qgap,60,pretty);
```

21

VERSION...[*Model* = *NW78binary 1.1.9.b*, *Note* = *change of seed is included*, *Date* = *22 Mar 96*]. *PARAMETERS*...[*Dtot* = 64, *Firms* = 4, *Rim* = .005000000000, *Rin* = .002500000000, *b* = 1, *c* = .16, δ = .03000000000, η = 1000, ϕ = .004951000000, ψ = 1, *seed* = 5, σ = .01782240000]

22

```
==in=> nw78(60,0,1,1,0,0,5);driv(HHs,60,pretty);driv(Qgap,60,pretty);
```

23

```
==in=> nw78(60,1,1,0,0,1,5);driv(HHs,60,pretty);driv(AGAP,60,pretty);
```

24

VERSION...[*Model* = *NW78binary 1.1.9.b*, *Note* = *change of seed is included*, *Date* = *22 Mar 96*]. *PARAMETERS*...[*Dtot* = 64, *Firms* = 4, *Rim* = .002500000000, *Rin* = .002500000000, *b* = 1, *c* = .16, δ = .03000000000, η = 1000, ϕ = .004951000000, ψ = 1, *seed* = 5, σ = .01782240000]

25

VERSION...[*Model* = *NW78binary 1.1.9.b*, *Note* = *change of seed is included*, *Date* = *22 Mar 96*]. *PARAMETERS*...[*Dtot* = 64, *Firms* = 16, *Rim* = .005000000000, *Rin* = .002500000000, *b* = 1, *c* = .16, δ = .03000000000, η = 1000, ϕ = .004951000000, ψ = 1, *seed* = 5, σ = .05940800000]

26

```
==in=> nw78(60,1,1,0,1,1,5);driv(HHs,60,pretty);driv(Qgap,60,pretty);
```

27

VERSION...[*Model* = *NW78binary 1.1.9.b*, *Note* = *change of seed is included*, *Date* = *22 Mar 96*]. *PARAMETERS*...[*Dtot* = 64, *Firms* = 16, *Rim* = .005000000000, *Rin* = .002500000000, *b* = 1, *c* = .16, δ = .03000000000, η = 1000, ϕ = .01485200000, ψ = 1, *seed* = 5, σ = .178224]

28 Figure 11 is produced by:

```
==in=> nw78(60,0,1,1,0,0,5);
==in=> drv(A,4,60,pretty);drv(K,4,60,pretty);
==in=> drv(s,4,60,pretty); drv(pi,4,60,pretty);
```

29

```
==in=> nw78(60,0,1,0,0,0,5);
==in=> drv(A,4,60,pretty);drv(K,4,60,pretty);
==in=> drv(s,4,60,pretty); drv(pi,4,60,pretty);
```

30 The command for each case is: tapf(A,4,[1,10,20,30,40,50,60],3,pretty);

31

VERSION...[*Model* = *NW78binary 1.1.9.b*, *Date* = *22 Mar 96*, *Note* = *change of seed is included*]. *PARAMETERS*...[*Dtot* = 64, *Firms* = 4, *Rim* = .005000000000, *Rin* = .002500000000, *b* = 1, *c* = .16, δ = .03000000000, η = 1000, ϕ = .004951000000, ψ = 1, *seed* = 5, σ = .01782240000]

32

VERSION...[*Model* = *NW78binary 1.1.9.b*, *Date* = *22 Mar 96*, *Note* = *change of seed is included*]. *PARAMETERS*...[*Dtot* = 64, *Firms* = 4, *Rim* = .005000000000, *Rin* = .002500000000, *b* = 1, *c* = .16, δ = .03000000000, η = 1000, ϕ = .004951000000, ψ = 1, *seed* = 5, σ = .01782240000]

33 For instance, N&W (1978, 545) points out that the ‘actual productivity level in (t + 1) is the largest of the available ones – with occational exceptions generated by a mechanism producing a small random error in the productivity level comparison process.’ We have not been able to find any specifications of this mechanism.

34

```
==in=> NW77param(8,101,dim=0.5,seed=6);
==in=>drv(A,8,101);drv(K,8,101);drv(s,8,101);drv(pi,8,101);
```

35

```
PARAMETERS...   [sigma = 5.0E-2, din = .125, split = true, eta = 1,
```

```
Ainit = .16, phi = 1.0E-2, rim = 9870646756.0E-13, dem = 67, TRim =
.4, TRin = 4.0, rin = 1974129351.0E-11, Seed = 1, psi = 1, seed = 6
, delta = 3.0E-2, b = 1, c = .16, dim = .5]
```

36

```
PARAMETERS...   c = .16, dim = .5, din = .125, eta = 1, phi = 1.0E-2,
rim = 1023454157.0E-12, dem = 67, delta = 3.0E-2, sigma = 5.0E-2, split
 = true, Ainit = .16, rin = 2046908314.0E-11, TRim = .4, Seed = 6, seed=1
TRin = 4.0, psi = 1, b = 1
```

37

```
==in=> tafv([A,K],8,101); taipv([AVpr,SDpr],[101]);
taipv([AVkap,SDkap],[101]);
```

38 Examples of commands are:

```
==in=> NW77stats(5,4,101,6);
==in=> taipv([R_AVpr, AVHHk, R_AVP, R_AVmax],100..101);
```

In the program recorded in appendix 3 the variable names have beed changed.

39

```
==in=> plot(67/Qtot, Qtot=100...500, P=0..0.69);
```

40

```
==in=> logistic := proc(Dmin, Dmax, Dr, T)
            local D, t, plotlist;
            D[0] := Dmin+1;
            for t from 1 to T do
                  D[t] := D[t-1] +
                        Dr*(1-(D[t-1]-Dmin)/(Dmax-Dmin))*(D[t-1]-Dmin);
            od;
            plotlist := seq([t/4,D[t]],t=0..T);
            RETURN([plotlist]);
      end:
==in=> figure16 := logisticPlus(67,134,0.1,100);
==in=> plot(figure16, coords=polar);
```

41 Read more on the generation of (pseudo-) random numbers in Knuth (1981).

42 For the role of such a variable, see Knuth, 1981.

43 Try the following experiments:

```
==in=> rand();   =out=> 427419669081
==in=> rand();   =out=> 321110693270
==in=> _seed := 6; rand();   =out=> 564518014508
==in=> _seed := 6; rand();   =out=> 564518014508
==in=> Rvalue1 := RandNormal(0.16,0.05);
==in=> Rvalue1();   =out=> .1747991246
==in=> Rvalue1();   =out=> .1451923234
==in=> Rvalue2 := RandNormal(ln(0.16),0.05);
==in=> Aln := Rvalue2();     =out=> -1.827656935
==in=> Aresult := exp(Aln);   =out=> .1607898679
==in=> Draws := RandPoisson(0.25);
==in=> Draws();   =out=> 0
==in=> Draws();   =out=> 1
```

44 In N&W's programs and in the programs of appendix 2 and 3 only one 'draw' per period is taken into account.

45 F for external finance.

46 The name for investments (I) presupposes that the standard MAPLE alias for $\sqrt{-1}$ has been switched off (by `alias(I=I);`).

47

```
==in=> plot({1/(1-s),  (2-s)/(2-2*s), (3-2*s)/(3-3*s), 1.2},
             s=0..0.5, mu=0.8..2);
```

References

Abelson, H., and Sussman, G.J. (1985), *Structure and Interpretation of Computer Programs*, McGraw-Hill, Cambridge, Mass. and London.

Andersen, E.S. (1994), *Evolutionary Economics: Post-Schumpeterian Contributions*, Pinter, London.

Andersen, E.S. (1996), An Evolutionary Model of an Industrial Sector with a Varying Degree of Roundaboutness of Production (Abstract), Abstract of a paper proposed for the International Schumpeter Society Conference, Stockholm, 3-5 June, Department of Business Studies, Aalborg University.

Andersen, E.S. (forthcoming-a), 'From Static Structures to Dynamics: Specialisation and Innovative Linkages', in DeBresson, C. (ed.) Economic Interdependence and Innovative Activity: An Input-Output Analysis, Elgar, Aldershot, 331-353.

Andersen, E.S. (forthcoming-b), 'The Evolution of Economic Complexity: A Division-and-Coordination-of-Labour Approach", in Helmstadter, E., and Perlman, M. (eds.), Behavioral Norms, Technological Progress and Economic Dynamics: Studies in Schumpeterian Economics, Michigan University Press.

Andersen, E.S., and Lundvall, B.-A. (1996), National Innovation Systems and the Dynamics of the Division of Labour, DRUID and IKE Group, Department of Business Studies, Aalborg University.

Chiaromonte, F., and Dosi, G. (1993), 'The Micro Foundations of Competitiveness and their Macroeconomic Implications', in Foray, D., and Freeman, C. (eds.), *Technology and the Wealth of Nations: The Dynamics of Constructed Advantage*, Pinter, London, 107-134.

Gerybadze, A. (1982), *Innovation, Wettbewerb und Evolution: Eine mikro- und mesoökonomische Untersuchung des Anpassungsprozesses von Herstellern und Anwendern neuer Produzentengüter*, J.C.B. Mohr (Paul Siebeck), Tübingen.

Kamien, M.I., and Schwarz, N.L. (1982), *Market Structure and Innovation*, Cambridge University Press, Cambridge.

Knuth, D.E. (1981), 'Random Numbers', in Knuth, D.E., *The Art of Computer Programming: Seminumerical Algorithms*, Vol. 2, 2nd edn., Addison-Wesley, Reading, Mass., 1-177

Kwasnicki, W. (1992), *Knowledge, Innovation, and Economy: An Evolutionary Exploration*, Oficyna Wydawnicza Politechniki Wroclawskiej, Wroclaw (revised version will be published by Elgar).

Levin, R.C., Klevorick, A., Nelson, R.R., and Winter, S.G. (1987), 'Appropriating the Returns from Industrial Research and Development', *Brookings Papers on Economic Activity*, Vol. 1987: 3.

Nelson, R.R. (1959a), 'The Economics of Invention: A Survey of the Literature', *Journal of Business*, Vol. 32, 101-27.

Nelson, R.R. (1959b), 'The Simple Economics of Basic Scientific Research', *Journal of Political Economy*, Vol. 67, 297-306.

Nelson, R.R. (1968), 'A "Diffusion Model" of International Productivity Differences in Manufacturing Industry', *American Economic Review*, Vol. 58, 1219-48.

Nelson, R.R., and Winter, S.G. (1974), 'Neoclassical vs Evolutionary Theories of Economic Growth: Critique and Prospectus', *Economic Journal*, Vol. 84, 886-905.

Nelson, R.R., and Winter, S.G. (1977), 'Dynamic Competition and Technical Progress', in Balassa, B., and Nelson, R.R. (eds.), *Economic Progress, Private Values, and Public Policy: Essays in Honor of William Fellner*, North-Holland, Amsterdam, .

Nelson, R.R., and Winter, S.G. (1978), 'Forces Generating and Limiting Concentration under Schumpeterian Competition', *Bell Journal of Economics*, Vol. 9, 524-48.

Nelson, R.R., and Winter, S.G. (1982), *An Evolutionary Theory of Economic Change*, Belknap Press, Cambridge, Mass. and London.

Nelson, R.R., and Winter, S.G. (1982b), 'The Schumpeterian Tradeoff Revisited', *American Economic Review*, Vol. 72, 114-132.

Nelson, R.R., Winter, S.G., and Schuette, H.L. (1976), 'Technical Change in an Evolutionary Model', *Quarterly Journal of Economics*, Vol. 90, 90-118.

Scherer, F.M., and Ross, D. (1990), *Industrial Market Structure and Economic Performance*, 3rd edn., Houghton Mifflin.

Schuette, H.L. (1980), The Role of Firm Financial Rules and a Simple Capital Market in an Evolutionary Model of Industry Growth, Doctoral dissertation, University of M*ichigan.*

Schumpeter, J.A. (1942), *Capitalism, Socialism and Democracy,* Rpt. 1987 with an introduction by T. Bottomore, Unwin, London.

Silverberg, G., Dosi, G., and Orsenigo, L. (1988), 'Innovation, Diversity and Diffusion: A Self-Organization Model', *Economic Journal*, Vol. 98, 1032-1054.

Silverberg, G., and Verspagen, B. (1994), 'Collective Learning, Innovation and Growth in a Boundedly Rational, Evolutionary World', *Journal of Evolutionary Economics*, Vol. 4, 207-226.

Winter, S.G. (1964), 'Economic "Natural Selection" and the Theory of the Firm', *Yale Economic Essays*, Vol. 4, 225-272.

Winter, S.G. (1971), *'Satisficing, Selection and the Innovating Remnant', Quarterly Journal of Economics*, Vol. 85, 237-261 [reprinted in Witt (1993), 76-100].

Winter, S.G. (1984), 'Schumpeterian Competition in Alternative Technological Regimes', *Journal of Economic Behavior and Organization*, Vol. 5, 287-320.

Winter, S.G. (1986), 'Schumpeterian Competition in Alternative Technological Regimes', in Day, R.D., and Eliasson, G. (eds.), *The Dynamics of Market Economies*, North-Holland, Amsterdam, 199-232. [The same as Winter, 1984, but with comments by T.v. Ungern-Sternberg and G. Eliasson]

Appendix 1: The DRUIDIC programming system

A1.1. Programming notation etc.

The following listing of programs (procedures and data structures) for evolutionary-economic experiments is intended to support most of the steps in the analytical process (see figure 13). The DRUIDIC meta-package is automatically loaded together with MAPLE. However, the user needs to type **start();** Then the user chooses one model for closer analysis. At the same time he/she may set certain options. More importantly, the package allows him/her to inspect and rewrite different parts of the program while the data structure and initialisation values are still in the memory of the computer, and he/she may also make experiments which simply consists in redefining the parameter of the (pseudo-) random number generator. It such activities are repeated he/she will see an 'evolving' set of evolutionary-economic packages rather than a (e.g. FORTRAN) program which is facing a strong lock-in to a narrow trajectory.

The linguistic unit in the programming notation is the statement. Statements are separated by ';'. Within a statement parentheses are used to avoid ambiguities or to increase readability. Indentation is used as a partial alternative to parentheses to increase readability. The reason why some programming languages are called imperative is because of the dominant role of the assignment statement which is basically a name which points to a value. The first time a name is used on the left side of an assignment statement, the name is placed in the first column of a 'table of variables' (the 'environment'). In another column the value of the variable is placed. Through a later assignment a statement this value (the content) of a variable may be replaced by the outcome of an evaluation of an expression. In any case, the assignment statement consists of the name of a variable, the assignment sign (':=') and a more or less complex expression.

A definition of a procedure begins with a name, the assignment sign ':=', the reserved word 'proc' and a set of parentheses enclosing none or more names of arguments of the procedure, e.g.:

```
==in=>  RandomNormal := proc(mean, sigma);
```

In this case a function for generating random numbers with two parameters (mean and sigma) which are unspecified at the moment when the procedure is being defined. The definition is given in the body of the definition which goes from the heading already specified to the reserved word "end". In this body the actions taken by the procedure is specified.

There are three types of structures controlling the computations: sequential computation is denoted by the succession of statements each of which is ended by a semicolon; iteration is denoted by '<u>for</u> *i* <u>from</u> *j* <u>to</u> *n* <u>do</u> *y*; <u>od</u>;', where *x* is a logical expression and *n* is an integer number. Selection is denoted '<u>if</u> *x* <u>then</u> *y* <u>else</u> *z*; <u>fi</u>;' or '<u>if</u> *x* <u>then</u> *y*; <u>fi</u>;' (else do nothing), etc.

Such commands are a conspicuous aspect of computer languages. They combine expressions and command structures. An expression is a part of a statement which is supposed to be evaluated by the reader or the computer system in order to provide information for the execution of the statement. The notation is freely using mathematical and statistical expressions. Furthermore, logical expressions, like '$x > y$', are used. In order to evaluate such an expression we first evaluate the two subexpressions 'x' and 'y' and then the expression as a whole. The value of the evaluation of a logical expression is either <u>true</u> or <u>false</u>. The normal equality sign, '=', is only used to denote equality in logical expressions.

The conventions about naming of constants, variables and functions are relatively open. Within definitions of procedures there are often a need for variables with no relevance for the overall computation. These are 'local' variables which in principle may be given local names which should not be confused with the names of those of the 'global' variables, even if they have the same names. To avoid confusion, a list of local names are specified in the beginning of the procedure, while the global names are often written g with an underline, like _seed for the variable specifying the level of help wanted by the user of the system.

The names of variables are often characterised by indexes for firm number and period number. A time-varying variable is denoted x[t] or x.t (t concatenated to x giving the name ‘xt’) and a firm-oriented variable x[i]. In combination we have the variable x for the i’th firm and the t’th period like x.t[i] or x[t,i].

A1.2. Procedures for data analysis

A programming system is not only defined by mechanisms for programming algorithms. The outcome of the computation is data structures in the computer memory or on a hard disk. The problem is to get access to these data in an easy way, so that the results of the analysis can be feed back into the next run on the computer. In principle, this is quite easy. However, MAPLE’s in-built methods of data inspection can be quite cumbersome. To make possible rapid and interactive data analysis, the DRUIDIC system provides several procedures, which are recorded in table 12.

To be able to use the information in table 12, it should be noted:

(1) The level of rounding of figures should be defined by ‘digits’ (e.g. 3).
(2) The text string ‘pretty’ is an optional argument that results in formatted text – but to obtain this one should additionally change ‘output’ from ‘character’ to e.g. ‘small’.
(3) The results (tables and figures) can be saved in a Maple Worksheet for later use, or for transferring to a computer connected to a printer.
(4) The results (tables and figures) can be exported to a program like MS Word through copying and pasting procedures.
(5) DATAexport gives the chance to store information in a form that can be manipulated by MS Excel or MS Word in other ways that given by the procedures.

The DRUIDIC analysis procedures do not only give the tables and figures but also additional information (so that the user knows precisely the underlying model and parameter values). The information can either be given in an unformatted way or in formatted form (after ‘pretty’ and the menu choice format/output/small):

Table 12 : DRUIDIC procedures for data analysis

```
      Help for each procedure can be obtained by typing
      ?procname – e.g. ?driv or ?drs

driv(var,t) = DrawIndVar
      - var is the name of the industry-level variable to be plotted
      - t is the number of periods - beginning with period #1
      - var can only be one of the variable with a table
      - 'pretty' is an optional argument
drs(var1,var2,n,period) = DrawScatter
      - var1 and var2 are the two variables to be plotted
      - n is the number of firms - beginning with firm #1
      - period is the period for which the plot is made
      - var1 and var2 can only be one of the variable with a table
      - 'pretty' is an optional argument
drv(var,n,t) = DrawVariable
      - var is the name of the variable to be plotted
      - n is the number of firms - beginning with firm #1
      - t is the number of periods - beginning with period #1
      - var can only be one of the variable with a table
      - 'pretty' is an optional argument
tafv(varlist,firms,period,digits) = TableFirmVar
      - varlist is a list of variable to be tabulated
      - firms are the number of firms in the table, starting with #1
      - period is the period for which the plot is made
      - varlist must be written as a list, i.e. [x,y,z]
      - digits is the number of digits printed in numbers
      - 'pretty' is an optional argument
taipv(varlist,periodlist,digits) = TableIndPerVar
      - varlist is a list of variables to be tabulated
      - periodlist is a list or a range of periods
      - periodlist can be either [2,4,6,8] or 2..8
      - digits is the number of digits printed in numbers
      - 'pretty' is an optional argument
tapf(var,firms,periodlist,digits) = TablePerFirm
      - var is the firm to be included in the table
      - firms are the number of firms in the table, starting with #1
      - periodlist is a list or a range of periods
      - periodlist can be either [2,4,6,8] or 2..8
      - digits is the number of digits printed in numbers
      - 'pretty' is an optional argument
tapv(varlist,firm,periodlist,digits) = TablePerVar
      - varlist is a list of variables to be tabulated
      - firm is the firm to be included in the table
      - periodlist is a list or a range of periods
      - periodlist can be either [2,4,6,8] or 2..8
      - digits is the number of digits printed in numbers
      - 'pretty' is an optional argument
DATAexport(filename)
      - filename is the name of the file containing the contents of
        a table that has earlier been made by tafv, taipv, tapf or tapv;
        the file has the following pathname ../Maple/lib/DruidData/filename
```

Appendix 2: NELWIN78 programs

The following programs are programmed with MAPLE V Rev2. The programs are available as MAPLE worksheets. To get a copy, send an e-mail to esa@business.auc.dk or check the WWW pages of the DRUIDIC project:
http://www.business.auc.dk/evolution/druidic/druidic.html (from April 96).

The choice of MAPLE as the core language reflects a consideration of the available expressive power and flexibility as well as its orientation towards interactive program development.. MAPLE is a package for making symbolic and numerical mathematics most computer systems. It has been developed since 1980 by researchers at University of Waterloo, Canada, and ETH, Zurich, Switzerland.

To make the programs usable with MAPLE V Rev3 or Rev4, several changes should be made. First of all, the parameters and variables should be declared as global variables. Furthermore, the functions like NormalDistr and Lottery has to be redefined. Ask for help!

NW78binary:= proc(T,Firmno,Invpol,Rapprop,Rgrowth,Rsdev)

```
options `DRUIDIC project 22 Mar 96`;

# MODEL DESCRIPTION
_descr := table();
_descr[Model] := `NW78binary 1.1.9.b`;
_descr[Date] := `22 Mar 96`;
_descr[Note] := `change of seed is included`;

# PARAMETERS
_ := table();
_[b] :=     1;
_[c] :=     .16 ;
_[delta] :=.3*10^(-1);
_[Dtot] :=   64;
_[phi] :=   .1*10^(-1);
_[psi] :=   1;
_[Rin] :=   .25*10^(-2);
_[seed] := _seed;

# VARIABLES
A     := table();
AGAP  := table();
AVcap := table();
AVpr  := table();
HHs   := table();
HHk   := table();
K     := table();
pi    := table();
P     := table();
Rmax  := table();
s     := table();
sK    := table();
SDpr  := table();
SDcap := table();
TK    := table();
Qtot  := table();

# BINARY DEFINITION OF PARAMETERS
if nargs = 7 then _seed := args[7]; _[seed] := args[7] fi;
if not {seq(args[k],k = 2 .. 6)} minus {0,1} = {} then
      ERROR(`we need 6 args: 1 normal and 5 binary (1=high or 0=low)`)
fi;
if Firmno = 1 then n := 16; _[Firms] := 16;
else n := 4; _[Firms] := 4;
fi;
if Invpol = 1 then _[eta] := 1000 else _[eta] := 1 fi;
if Rapprop = 1 then _[Rim] := .25*10^(-2);
else _[Rim] := .5*10^(-2);
```

```
fi;
if Rgrowth = 1 then _[phi] := .14852*10^(-1);
else _[phi] := .4951*10^(-2);
fi;
if Rgrowth = 1 then
      if Rsdev = 1 then _[sigma] := .178224;
      else _[sigma] := .59408*10^(-1);
      fi;
else
      if Rsdev = 1 then _[sigma] := .59408*10^(-1);
      else _[sigma] := .178224*10^(-1);
      fi;
fi;

# INITIALISATION
for i to n do
      A[i,1] := .16 ;
      if n = 4 then
            if Invpol = 1 then K[i,1] := 75.094  else K[i,1] := 100.0  fi
      elif n = 16 then
            if Invpol = 1 then K[i,1] := 23.466  else K[i,1] := 25.0  fi;
      fi;
od;
Rmean[0] := .16 ;

# MAIN LOOP
for t to T do

      # SHORT-RUN MARKET
      for i to n do  Q[i] := A[i,t]*K[i,t] od;
      Qtot[t] := sum(Q['k'],'k' = 1 .. n);
      P[t] := _[Dtot]/Qtot[t];

      # NEW TECHNOLOGY
      Rmax[t] := max(seq(A[i,t],i = 1 .. n));
      Rmean[t] := Rmean[t-1]*(1+_[phi]);
      Rdistr := RandNormal(ln(Rmean[t]),_[sigma]);
      for i to n do
            # INNOVATE
            Ain := 0;
            lambda := _[Rin]*K[i,t];
            Rdraws := RandPoisson(lambda);
            Rdrawsno := Rdraws();
            if 0 < Rdrawsno then
                  Rvalue := exp(Rdistr()); Ain := max(Ain,Rvalue)
            fi;
            # IMITATE
            lambda := _[Rim]*K[i,t];
            Rdraw := RandPoisson(lambda);
            if 0 < Rdraw() then Aim := Rmax[t] else Aim := 0 fi;
            # TECHNO-CHOICE
            A[i,t+1] := max(A[i,t],Ain,Aim);
      od;

      # NEW CAPITAL
      for i to n do
            s[i,t] := Q[i]/Qtot[t];
            muact := P[t]*A[i,t+1]/_[c];
            mudes := _[eta]/(_[eta]-s[i,t]);
            Ides[i] := _[delta]+1-mudes/muact;
            pi[i,t] := P[t]*A[i,t]-_[c];
            if pi[i,t] <= 0 then F[i] := 0 else F[i] := _[b]*pi[i,t] fi;
            Imax[i] := _[delta]+pi[i,t]+F[i];
            I := max(0,min(Ides[i],Imax[i]));
            K[i,t+1] := K[i,t]*(I+1-_[delta]);
      od;

      # STATISTICS
      TK[t] := sum(K[xx,t],xx = 1 .. n);
      for i to n do  sK[i,t] := K[i,t]/TK[t] od;
      HHk[t] := 1/sum(sK[xx,t]^2,xx = 1 .. n);
      HHs[t] := 1/sum(s[xx,t]^2, xx = 1..n);
```

```
      AVpr[t] := sum(A['k',t],'k' = 1 .. n)/n;
      AVkap[t] := sum(K['k',t],'k' = 1 .. n)/n;
      AGAP[t]:=Qtot[t]/(Rmax[t]*TK[t]);

od;
end:
```

NW78stats := proc(R,T, Firmno, Invpol, Rapprop, Rgrowth, Rsdev,SEED)

```
options `LPM/ESA 19 Mar 96`;
_descr := table();
_descr[Model] := `NW78stats 1.1.7.c`;
_descr[Date] := `19 Mar 96`;
_descr[Note] := `Stat`;
Seed := SEED;
for r from 1 to R do
    _seed := Seed + r;
    NW78binary(T, Firmno, Invpol, Rapprop, Rgrowth, Rsdev);
    _HHs[r]:= HHs[T];
    _HHk[r] := HHk[T];
    _AVpr[r] := AVpr[T];
    _AVkap[r] := AVkap[T];
    _AGAP[r]:= AGAP[T];
od;
AVHHs:= sum(_HHs['k'], 'k'=1..R)/R;
AVHHk := sum(_HHk['k'], 'k'=1..R)/R;
AVA := sum(_AVpr['k'],'k'=1..R)/R;
AVK := sum(_AVkap['k'], 'k'=1..R)/R;
AVK := sum(_AVkap['k'], 'k'=1..R)/R;
AVG := sum(_AGAP['k'], 'k'=1..R)/R;
SDHHk := (sum((_HHk['k'] - AVHHk)^2, 'k' = 1..R)/(R-1))^0.5;
SDHHs := (sum((_HHs['k'] - AVHHs)^2, 'k' = 1..R)/(R-1))^0.5;
SDA := (sum((_AVpr['k'] - AVA)^2, 'k' = 1..R)/(R-1))^0.5;
SDK := (sum((_AVkap['k'] - AVK)^2, 'k' = 1..R)/(R-1))^0.5;
SDG := (sum((_AGAP['k'] - AVG)^2, 'k' = 1..R)/(R-1))^0.5;
end:
```

Appendix 3: NELWIN77 programs

NW77param := proc(n,T)

```
local Aim, Ain, Amax, Amean, Aprel, Ides, Imax, lambda, loans, Lottery, muact,
mudes, NormalDistr, param;
options `ESA/AKJ 28 Feb 96`;

# MODEL DESCRIPTION
_descr           := table();
_descr[Model]    := `NELWIN77seed 1.1.7.a`;
_descr[Date]     := `28 Feb 96`;
_descr[Variant]  := `Half innovators; multidraws off.`;

# PARAMETERS
param := {b, c, delta, dem, dim, din, eta,
           phi, rim, rin, seed, sigma, split};
_          := table();
_[Ainit]   := 0.16;
_[b]       := 1;
_[c]       := 0.16;
_[delta]   := 0.03;
_[dim]     := 1.25;
_[din]     := 0.125;
_[dem]     := 67;
_[eta]     := 1;
_[phi]     := 0.01;
_[psi]     := 1;
_[TRim]    := 0.4;
_[TRin]    := 4.0;
```

```
_[Seed]     := 1;
_[sigma]    := 0.05;
_[split]    := true;

# PARAMETER REVISION
if nargs > 2 then
      for i from 3 to nargs do
            if not member(op(1,args[i]), param) then
                  ERROR(`wrong name of parameter`);
            fi;
      _[op(1,args[i])] := op(2,args[i]);
      od;
fi;
_seed       := _[Seed];

# STATE VARIABLES
A           := table();
K           := table();

# OTHER VARIABLES
AVpr  := table();
AVcap := table();
HHs         := table();
HHk         := table();
pi          := table();
P           := table();
s           := table();
sK          := table();
SDpr        := table();
SDcap       := table();
TK          := table();
TQ          := table();

# INITIALISATION
for i from 1 to n do
      A[i,1]    := _[Ainit];
      s[i,1]    := 1/n;
      mudes         := (2*_[eta]-s[i,1])/(2*_[eta]-2*s[i,1]);
      K[i,1]    := _[dem]/(n*_[c]*mudes);
od;
Amean[0] := _[Ainit];
TK[1]  := sum(K['i',1], 'i'=1..n);
_[rim] := _[TRim]/TK[1];
if _[split] then
      if not type(n/2, integer) then
            ERROR(`with split = true, the number of firms must be even`);
      fi;
      _[rin] := (_[TRin]*2)/TK[1];
      for i from 1 to n do
            if i <= n/2 then
                  rin[i] := _[rin];
            else
                  rin[i] := 0.00000001;
            fi;
      od;
else
      _[rin] := _[TRin]/TK[1];
      for i from 1 to n do
            rin[i] := _[rin];
      od;
fi;

# MAIN LOOP
for t from 1 to T do

      # SHORT-RUN MARKET
      for i from 1 to n do
            Q[i] := A[i,t]*K[i,t];
      od;
      TQ[t] := sum(Q[k], k = 1..n);
      P[t]:= _[dem]/TQ[t];
```

```
        # NEW TECHNOLOGY
        Amax := max(seq(A[i,t], i = 1..n));
        Amean[t] := Amean[t-1]*(1 + _[phi]);
        NormalDistr := RandNormal(ln(Amean[t]),_[sigma]);
        for i from 1 to n do
              # INNOVATE
              lambda := _[din]*rin[i]*K[i,t];
              Lottery := RandPoisson(lambda);
              Drawsno := Lottery();
              if Drawsno > 0 then Ain := exp(NormalDistr());
              else Ain := 0;
              fi;
              # IMITATE
              lambda := _[dim]*_[rim]*K[i,t];
              Lottery := RandPoisson(lambda);
              if Lottery() > 0 then Aim := Amax;
              else Aim := 0;
              fi;
              # TECHNO-CHOICE (later: add error to Ain)
              A[i,t+1] := max(A[i,t], Ain, Aim);
        od;

        # NEW CAPITAL
        for i from 1 to n do
              # DESIRED INVESTMENT
              s[i,t] := Q[i]/TQ[t];
              muact := (P[t]*A[i,t+1])/_[c];
              mudes := (2-s[i,t])/(2 - 2*s[i,t]);
              Ides[i] := _[delta] + 1 - mudes/muact;
              # MAXIMUM INVESTMENT
              pi[i,t] := P[t]*A[i,t] - (_[delta]+_[rim]+rin[i]);
              if pi[i,t] <= 0 then loans[i] := 0;
              else loans[i] := 1*pi[i,t];
              fi;
              Imax[i] := _[delta] + pi[i,t] + loans[i];
              # NEW CAPITAL
              Inv := max(0, min(Ides[i], Imax[i]));
              K[i,t+1] := K[i,t]*(Inv + 1 - _[delta]);
        od;

        # INDUSTRY STATISTICS
        TK[t] := sum(K[k,t], k = 1..n);
        AVpr[t] := sum(A[k,t], k = 1..n)/n;
        AVkap[t]:= sum(K[k,t], k = 1..n)/n;
        AVpi[t]:=sum(pi[k,t], k = 1..n)/n;
        sK[i,t] := K[i,t]/TK[t];
        SSpr[t] := (sum((A[k,t] - AVpr[t])^2, k = 1..n)/(n-1))^0.5;
        SSkap[t] := (sum((K[k,t] - AVkap[t])^2, k = 1..n)/(n-1))^0.5;
        SSpi[t] := (sum((pi[k,t] - AVpi[t]^2), k = 1..n)/(n-1))^0.5;
        HHk[t] := 1/sum(sK[k,t]^2, k = 1..n);
        HHs[t] := 1/sum(s[k,t]^2, k = 1..n);

od;
end:
```

NW77stats := proc(R,n,T,SEED)

```
options `ESA/AKJ 28 Feb 96`;
Seed := SEED;
for r from 1 to R do
      Seed := Seed + r;
      NELWIN77param(n,T,'seed'=Seed);
      _HHk[r] := HHk[T];
      _HHs[r] := HHs[T];
      _SDHHk[r] := SSHHk[T];
      _SDHHs[r]:= SSHHs[T];
      _AVpr[r] := AVpr[T];
      _AVpi[r] := AVpi[T];
      _AVkap[r] := AVkap[T];
      _SDpr[r] := SSpr[T];
```

```
      _SDpi[r] := SSpi[T];
      _SDkap[r] := SSkap[T];
od;
HHHk := sum(_HHk['k'], 'k'=1..R)/R;
HHHs := sum(_HHs['k'], 'k'=1..R)/R;
SDHHHk := sdev(seq(_HHk['k'], 'k'=1..R));
SDHHHs := sdev(seq(_HHs['k'], 'k'=1..R));
AAVpr := sum(_AVpr['k'],'k'=1..R)/R;
AAVpi := sum(_AVpi['k'], 'k'=1..R)/R;
AAVkap := sum(_AVkap['k'], 'k'=1..R)/R;
end:
```

Danish **R**esearch **U**nit for **I**ndustrial **D**ynamics

The Research Programme

The DRUID-research programme is organised in 3 different research themes :

- The firm as a learning organisation

- Competence building and inter-firm dynamics

- The learning economy and the competitiveness of systems of innovation

In each of the three areas there is one strategic theoretical and one central empirical and policy oriented orientation.

Theme A: The firm as a learning organisation

The theoretical perspective confronts and combines the ressource-based view (Penrose, 1959) with recent approaches where the focus is on learning and the dynamic capabilities of the firm (Dosi, Teece and Winter, 1992). The aim of this theoretical work is to develop an analytical understanding of the firm as a learning organisation.

The empirical and policy issues relate to the nexus technology, productivity, organisational change and human ressources. More insight in the dynamic interplay between these factors at the level of the firm is crucial to understand international differences in performance at the macro level in terms of economic growth and employment.

Theme B: Competence building and inter-firm dynamics

The theoretical perspective relates to the dynamics of the inter-firm division of labour and the formation of network relationships between firms. An attempt will be made to develop evolutionary models with Schumpeterian innovations as the motor driving a Marshallian evolution of the division of labour.

The empirical and policy issues relate the formation of knowledge-intensive regional and sectoral networks of firms to competitiveness and structural change. Data on the structure of production will be combined with indicators of knowledge and learning. IO-matrixes which include flows of knowledge and new technologies will be developed and supplemented by data from case-studies and questionnaires.

Theme C: The learning economy and the competitiveness of systems of innovation.

The third theme aims at a stronger conceptual and theoretical base for new concepts such as 'systems of innovation' and 'the learning economy' and to link these concepts to the ecological dimension. The focus is on the interaction between institutional and technical change in a specified geographical space. An attempt will be made to synthesise theories of economic development emphasising the role of science based-sectors with those emphasising learning-by-producing and the growing knowledge-intensity of all economic activities.

The main empirical and policy issues are related to changes in the local dimensions of innovation and learning. What remains of the relative autonomy of national systems of innovation? Is there a tendency towards convergence or divergence in the specialisation in trade, production, innovation and in the knowledge base itself when we compare regions and nations?

The Ph.D.-programme

There are at present more than 10 Ph.D.-students working in close connection to the DRUID research programme. DRUID organises regularly specific Ph.D-activities such as workshops, seminars and courses, often in a co-operation with other Danish or international institutes. Also important is the role of DRUID as an environment which stimulates the Ph.D.-students to become creative and effective. This involves several elements:

- access to the international network in the form of visiting fellows and visits at the sister institutions
- participation in research projects
- access to supervision of theses
- access to databases

Each year DRUID welcomes a limited number of foreign Ph.D.-students who wants to work on subjects and project close to the core of the DRUID-research programme.

External projects

DRUID-members are involved in projects with external support. One major project which covers several of the elements of the research programme is DISKO; a comparative analysis of the Danish Innovation System; and there are several projects involving international co-operation within EU's 4th Framework Programme. DRUID is open to host other projects as far as they fall within its research profile. Special attention is given to the communication of research results from such projects to a wide set of social actors and policy makers.

DRUID Working Papers

96-1 **Lundvall, Bengt-Åke:** The Social Dimension of the Learning Economy. (ISBN 87-7873-000-7)

96-2 **Foss, Nicolai J.:** Firms, Incomplete Contracts and Organizational Learning.
(ISBN 87-7873-001-5)

96-3 **Dalum, Bent and Villumsen, Gert**:Are OECD Export Specialisation Patterns 'Sticky?' Relations to the Convergence-Divergence Debate. (ISBN 87-7873-002-3)

96-4 **Foss, Nicolai J:** Austrian and Post-Marshallian Economics: The Bridging Work of George Richardson. (ISBN 87-7873-003-1)

96-5 **Andersen, Esben S., Jensen, Anne K., Madsen, Lars and Jørgensen, Martin:** The Nelson and Winter Models Revisited: Prototypes for Computer-Based Reconstruction of Schumpeterian Competition. (ISBN 87-7873-005-8)

Information for subscribers.

Subscription price for 1996 is 600 DKR (about 20 papers). The rate for single issues is 40 DKR. It is possible to make a commitment to an exchange of papers from related departments or research teams. All correspondence concerning the DRUID Working Papers should be send to:

Mette Madsen
Fibigerstræde 4,
DK-9220 Aalborg OE
Tel. 45 98 15 42 11-2945
Fax. 45 98 15 60 13
E-mail: mm@business.auc.dk