AN ALTERNATIVE ALGORITHM FOR MAPPING STABLE PARAMETERS IN STOCK-FLOW CONSISTENT (SFC) MODELS

Júlio Fernando Costa Santos[*]

Guilherme Jonas da Silva[†]

Resumo

O presente artigo desenvolve um algoritmo de análise de estabilidade através da junção das técnicas de simulação de Monte Carlo, Gauss-Seidel/Newton e Filtros de Variação. Para a comparação de resultados, utiliza-se tanto o algoritmo desenvolvido como uma solução numérico-analítica para o modelo *Benchmark* de Dos Santos e Zezza (2008). Os resultados mostram que há uma sobreposição entre os dois métodos de análise. Tendo em vista que nem sempre é possível ou viável a análise da estabilidade do sistema através de uma solução analítica, o algoritmo apresentado inova ao fornecer uma rota de fuga para o mapeamento de parâmetros estáveis em modelos dinâmicos SFC.

Palavras-chaves: Modelo *Stock-Flow Consistent*; Macrodinâmica Pós-Keynesiana; Algoritmos de Solução Numérica.
Classificação JEL: C53; E12; E17.

Abstract

This article describes the development of a stability analysis algorithm through a combination of Monte Carlo Simulation, Gauss-Seidel/Newton and Variation Filter techniques. In order to compare results, both the new algorithm and a numerical-analytical solution are tested against the Dos Santos and Zezza (2008) benchmark model. The outcome demonstrates overlap between the results from the two approaches. Considering that it is not always possible or feasible to analyze a system's stability through an analytical solution, our algorithm innovates by providing an escape route for mapping stable parameters.

Keywords: Stock-Flow Consistent Model; Post-Keynesian Macrodynamics; Numerical Solution Algorithms.
JEL Classification: C53; E12; E17.

[*] Adjunct Professor of Economics at the Federal University of Uberlândia (UFU). E-mail: julio.costa@ufu.br.
[†] Associate Professor of Economics at the Federal University of Uberlândia (UFU). E-mail: guilhermejonas@yahoo.com.br.

1. Introduction

In the Stock Flow Consistent (SFC) literature, the usual treatment to solve models is to seek a numerical solution to the set of parameters under analysis. In general, these dynamic models can be understood as linear n×n systems, where the current variables are functions of other current and lagged variables.

According to a survey conducted by Caverzasi and Godin (2014), which looked at the current state of the literature regarding post-Keynesian Stock-Flow Consistent modelling (PK-SFC), there are two ways to solve a dynamic economic model: numerically and analytically. The authors also point out that it is possible to solve a model deductively, but this is not considered a proper approach.

Solving a model numerically involves dealing with the following fundamental questions: (i) how to determine the parameter values and the initial value of the endogenous variables; and (ii) how to use the simulation results.

The first question can be answered through one of two methodologies: (a) estimation or (b) calibration. Estimation is the use of a statistical/econometric methodology to determine the parameter values to be used in the model. Calibration, for its part, is the process of determining the parameter and initial stock values using stylized facts or rules of thumb.

The problems that arise from the first methodology relate to the implicit assumption that the parameters are constant over time, which opens the door to the Lucas Critique (1976)[1], although shocks are applied to the model to analyze its general behavior.

The problems related to the second methodology involve questions about the correct way to use it. Two approaches may be found in the literature. The first involves letting the model start from a steady state and then applying a shock. In the second, a base-scenario is drawn with no restriction to convergence to the steady state; from this scenario certain parameters are modified, and the behavior of the model is observed.

[1] In particular, the Lucas Critique (LC) is directed at models that utilize the aggregate behavior of agents without micro-foundations. The problem Lucas outlined is that agents should change their behavior when they change policies and, ultimately, this will affect the set of parameters used in a simulation of aggregated macrodynamic models, which cease to be constant over time. A recent response to the LC in terms of SFC modeling is the incorporation of micro-foundations using the agent-based approach (Agent-Based Models), rather than representative agents, who make choices via intertemporal optimization. A more detailed discussion can be seen in Caiani *et al.* (2016).

The algorithm developed in this article applies the calibration methodology, following an approach in which there is convergence to the steady state. The alternative here involves mapping the desired (or plausible) intervals for the model's endogenous variables and based on these conditions, the attainment of the parameters that will lead the model to this outcome.

According to O'Shea and Kinsella (2010), there is an algorithm for numerical calculation that facilitates the resolution of such models, regardless of their size. This algorithm consists of solving the system for each period *t* using the Gauss-Seidel method and inputting into it, as the best initial guess, the values solved in the last period, *t-1*, for resolution in the current period, *t*. In this sense, econometric software, such as Eviews, and programming languages, such as R and Python, already contain a routine that includes this algorithm[2]. Most of the studies conducted so far use this kind of software to solve their proposed models.

However, calibrating models so that they work well can be an arduous task, since many parameters are difficult to obtain through econometrics (due to a lack of data or a difficulty of estimation) or from previous works. One of the criticisms of SFC models, therefore, is that they do not provide us with general behavior, but rather with behavior specific to the set of parameters used in their calibration.

Authors such as Godin *et al.* (2012) have already made efforts to develop and provide escape routes for authors who wish to make parameter estimations using the linear programming method. However, this remains an open research agenda within this literature.

Throughout sections 2 and 3 of this paper, we therefore present, in the form of a "cookbook", an analytical approach to solving SFC models in discrete time and to assessing their stability in the steady state. Following this, in section 4, we present the algorithms of numerical calculation that help solve these models, generating numerical-analytical answers for both solutions and stability. We call this method M1. In section 5, we present a new, alternative algorithm, which makes it possible to calibrate SFC models by mapping their stability and their economic viability parameters. We call this method M2. Finally, in section 6, we apply M1 and M2 to the "benchmark" model in the literature, demonstrating the convergence of results. Finally, in section 7, we outline our conclusions.

[2] Examples of model simulation routines from the book by Godley and Lavoie (2007) can be found at: http://models.sfc-models.net/.

2. Stability in Stock-Flow Consistent Models (SFC)

The SFC literature has predominantly used discrete-time (difference equation) rather than continuous-time (differential equation) models. Although relevant models are being developed in continuous time, the stability issue discussed in this article is only concerned with discrete-time ones. The reason for this is simple. The literature on continuous time has been widely developed and disseminated in works of applied economics, such as Gandolfo (2009) and Shone (2002). Sections 2 and 3 below were developed in line with Elaydi (2005), Gandolfo (2009), Viana (2012), and Ruggiero and Lopes (1996).

2.1. Stability Analysis in Difference Equations

The presentation of the stability analysis in the difference equations section is divided as follows: linear multidimensional systems; and fixed points and stability in linear systems.

2.1.1. Linear Multidimensional Systems

A linear multidimensional system can be described as follows. There are *N* discrete dynamic variables with *i=1,2,…,N*. Time is also a discrete variable thus: *t=0,1,2,…,T*. We therefore have variables ${x_1}_t, {x_2}_t, \dots, {x_N}_t$ which can be described in the following general way:

$$\begin{aligned} {x_1}_t &= F_1\left({x_1}_{t-1}, {x_2}_{t-1}, \dots, {x_N}_{t-1}\right) \\ {x_2}_t &= F_2\left({x_1}_{t-1}, {x_2}_{t-1}, \dots, {x_N}_{t-1}\right) \\ &\vdots \\ {x_N}_t &= F_N\left({x_1}_{t-1}, {x_2}_{t-1}, \dots, {x_N}_{t-1}\right) \end{aligned} \tag{1}$$

where F_i, i=1,2, … ,N are, in principle, functions of all the variables[3]. In matrix notation, we can define the following N×1 column matrices at time *t*. Thus:

$$v_t = \begin{pmatrix} {x_1}_t \\ {x_2}_t \\ \vdots \\ {x_N}_t \end{pmatrix} \quad \text{e} \quad F(v_t) = \begin{pmatrix} F_1(v_t) \\ F_2(v_t) \\ \vdots \\ F_N(v_t) \end{pmatrix} \tag{2}$$

where v_t is a matrix of the n variables at period *t* and $F(v_t)$ is a column matrix containing their respective functions. Taking the two matrices from equation (2), the general multidimensional model can be described as follows:

[3] In fact, the discrete-time system for SFC models can still take into account the relationships between variables in current time. However, the explanation we provide in this section is simpler.

$$\mathrm{v_t = F(v_{t-1})} \tag{3}$$

Since we are interested in the affine vector function, we have:

$$\mathrm{F(v_t) = A.v_t + B} \tag{4}$$

where the A and B matrix can be defined as:

$$\mathrm{A} = \begin{pmatrix} \mathrm{A_{11}} & \mathrm{A_{12}} & \cdots & \mathrm{A_{1N}} \\ \mathrm{A_{21}} & \mathrm{A_{22}} & \cdots & \mathrm{A_{2N}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{A_{N1}} & \mathrm{A_{N2}} & \cdots & \mathrm{A_{NN}} \end{pmatrix}, \quad \mathrm{B} = \begin{pmatrix} \mathrm{B_{11}} \\ \mathrm{B_{21}} \\ \mathrm{B_{31}} \\ \mathrm{B_{N1}} \end{pmatrix} \tag{5}$$

So, the map can be described as follows:

$$\mathrm{v_t = A.v_{t-1} + B} \tag{6}$$

2.1.1.1. Fixed Points and Stability in Linear Systems:

We have the following fixed points vector (column matrix):

$$\mathrm{v^*} = \begin{pmatrix} \mathrm{x_1}^* \\ \mathrm{x_2}^* \\ \vdots \\ \mathrm{x_N}^* \end{pmatrix} \tag{7}$$

Fixed points can be obtained as follows, provided that the (I-A) matrix is not singular (that is, it has an inverse):

$$\mathrm{v^* = (I - A)^{-1}.B} \tag{8}$$

The general solution of the N-dimensional model is given by:

$$\mathrm{v_t = A^t.[v_0 - v^*] + v^*} \tag{9}$$

where $\mathbf{v_0}$ is the vector of dynamic variables in period $t = 0$.

It is possible to translate the stability of the fixed point to the origin of the N-dimensional space, that is, to the point with coordinates x_i=0, where *i=1,2 , …, N*; making:

$$\mathrm{w_t \equiv v_t - v^*} \tag{10}$$

which means that it can be written more simply (without the constant term), thus:

$$\mathrm{w_t = A.w_{t-1}} \tag{11}$$

The stability of the fixed point at origin therefore depends on the Euclidean distance in the dynamic variables' N-dimensional space which, in turn, is the Euclidean norm of the $\mathrm{w_t}$ vector:

$$\left|\left|\mathrm{w_t}\right|\right| = \sqrt{\mathrm{(w_1)_t^2 + (w_2)_t^2 + \cdots + (w_N)_t^2}} \tag{12}$$

where $(w_i)_t^2 = x_i - x_i^*$, for i=1,2,…,N. The fixed point at the origin $\mathrm{w^*} = 0$ is asymptotically stable if, given initial condition $\mathrm{w_0}$, the subsequent iterations, $\mathrm{A^t.w_0}$, are such that the distances from the origin approach zero when t→∞. Analogously, the origin will be unstable if the distance diverges to infinity when t→∞. Thus, determining the stability of the fixed point requires an analysis of the eigenvalues of matrix A. The square matrix A, (N×N), has eigenvalues ξ given by the roots of the secular equation, which are obtained by:

$$\det(\mathrm{A}-\xi.\mathrm{I})=0 \tag{13}$$

where I is the identity matrix of order N:

$$\mathrm{I}=\begin{pmatrix}1 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1\end{pmatrix} \tag{14}$$

The secular equation is obtained by setting a determinant of order *N* to zero, that is:

$$\begin{pmatrix}\mathrm{A}_{11}-\xi & \mathrm{A}_{12} & \cdots & \mathrm{A}_{1n}\\ \mathrm{A}_{21} & \mathrm{A}_{22}-\xi & \cdots & \mathrm{A}_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ \mathrm{A}_{n1} & \mathrm{A}_{n2} & \cdots & \mathrm{A}_{nn}-\xi\end{pmatrix}=0 \tag{15}$$

Thus, providing an algebraic equation of degree *N* in the form:

$$a_0.\xi^n+a_1.\xi^{n-1}+a_{n-1}.\xi+a_n=0 \tag{16}$$

where a_i, *i=0,1,2,…,n-1* are coefficients dependent on the elements of matrix A. Of course, we can divide the whole equation by $a_0 \neq 0$, such that the coefficient of the first term can be taken as one, without loss of generality.

In the fundamental algebra theorem, an algebraic equation of degree N has N roots, which can be real or complex. Therefore, matrix A will always have N eigenvalues, denoted ξ_i with *i=1,2,…,N*; as well as *N* associated eigenvectors ***u****ᵢ*, such that

$$\mathrm{A}.\mathrm{u}_\mathrm{i}=\xi_\mathrm{i}.\mathrm{u}_\mathrm{i} \tag{17}$$

In principle, to find the eigenvalues of matrix A, we have to solve (7). However, analytical solutions (using radicals) only exist for *N≤4*, while for *N=3* such solutions are already quite complicated and hard to interpret[4]. In practice, explicit calculus requires numerical solutions. One possible solution is to calculate the eigenvalues of the companion matrix.

Generally speaking, there are some real roots and some complex ones, while certain real roots are also complex (multiple or degenerate), which, in general, makes the problem

[4] When $N=2$, we can use Bhaskara's formula to find the polynomial roots. When N=3, we can use Cardano's formula to find the roots, although the expression is much more complicated. When N≥4, we do not have an algebraic expression to find the polynomial roots. Numerical resolution methods therefore become necessary from N>3.

quite complicated. Alternative criteria, such as the Jury Criterion and/or the Schur Criterion, can be used to avoid calculating all the system's eigenvalues. An explanation of these criteria is beyond the scope of this work, so we will only observe the following rules regarding the calculated eigenvalues: If the module of the calculated eigenvalues (complex or real) is less than one, the condition is sufficient for the fixed point to be considered stable. If at least one calculated eigenvalue has a value greater than unity in the module, the fixed point is unstable.

2.1.2. Nonlinear Multidimensional Systems

Our presentation of nonlinear multidimensional systems is divided as follows: fixed points and periodic orbits in nonlinear systems; stability in nonlinear models; hyperbolic points; and solving nonlinear systems using the Newton-Raphson method.

2.1.2.1. Fixed Points and Periodic Orbits in Nonlinear Systems

In many cases, an analysis of linear models also allows us to determine the stability of fixed points in nonlinear models in the way we have done in equation (4), $\mathrm{v}_t = \mathrm{F}(\mathrm{v}_{t-1})$, presented above.

The fixed point of a multidimensional model is a vector of N components that maps onto itself, as presented in equation (8), as follows:

$$\mathrm{v}^* = \mathrm{F}(\mathrm{v}^*) \tag{18}$$

Another concept applicable in more than one dimension is that of periodic orbits. The periodic orbit of period *m*, or the *m*-cycle, is a set of *m* vectors $\{\mathrm{v}_1^*, \mathrm{v}_2^*, \mathrm{v}_3^*, \dots, \mathrm{v}_\mathrm{m}^*\}$, such that one maps the other, cyclically.

$$\begin{gathered} \mathrm{v}_{i+1}^* = \mathrm{F}(\mathrm{v}_i^*) \qquad (i = 1,2,\dots,m-1) \\ \mathrm{v}_1^* = \mathrm{F}(\mathrm{v}_m^*) \end{gathered} \tag{19}$$

Applying these successively to any of the vectors in the *m*-cycle, such as v_m^*, we have:

$$\mathrm{v}_m^* = \mathrm{F}(\mathrm{v}_{m-1}^*) = \mathrm{F}\big(\mathrm{F}(\mathrm{v}_{m-2}^*)\big) = \mathrm{F}\left(\mathrm{F}\big(\mathrm{F}(\mathrm{v}_{m-3}^*)\big)\right) = \mathrm{F}^{[m]}(\mathrm{v}_m^*) \tag{20}$$

so that the elements of an m-cycle are fixed points of the m-th iteration of the $\mathbf{F}(x)$ model.

2.1.2.2. Stability in Nonlinear Models

The fixed-point stability of the nonlinear multidimensional model $\mathrm{v}_{\mathrm{t}+1} = \mathrm{F}(\mathrm{v}_\mathrm{t})$ is investigated by its linearization in the neighborhood of point $\mathbf{v}^*$. We work in the N-

dimensional plane of dynamic variables and around the fixed point, whose coordinates are $(x_1^*, \dots, x_N^*)$, we analyze a hypersphere of radius ϵ, where $\epsilon \ll x_i^*$ is a small number (compared to unity).

To linearize the model, and check whether the fixed point is stable within this neighborhood bounded by ϵ, we expand the N functions $\mathrm{F}_i(\mathrm{v})$ in series. We write the vector in the neighborhood of the fixed point as:

$$\mathrm{v}_t = \mathrm{v}^* + \mathrm{w}_t \tag{21}$$

$$\mathrm{w_t} = \begin{pmatrix} (\mathrm{w_1})_\mathrm{t} \\ (\mathrm{w_2})_\mathrm{t} \\ \vdots \\ (\mathrm{w_N})_\mathrm{t} \end{pmatrix} = \begin{pmatrix} (\mathrm{x_1})_\mathrm{t} - \mathrm{x}_1^* \\ (\mathrm{x_2})_\mathrm{t} - \mathrm{x}_2^* \\ \vdots \\ (\mathrm{x_N})_\mathrm{t} - \mathrm{x}_\mathrm{N}^* \end{pmatrix} \tag{22}$$

such that (19) becomes:

$$\begin{array}{c} (\mathrm{w_1})_\mathrm{t} + \mathrm{x}_1^* = \mathrm{F_1}((\mathrm{w_1})_{\mathrm{t}-1} + \mathrm{x}_1^*, (\mathrm{w_2})_{\mathrm{t}-1} + \mathrm{x}_2^*, \dots, (\mathrm{w_N})_{\mathrm{t}-1} + \mathrm{x}_\mathrm{N}^*) \\ (\mathrm{w_2})_\mathrm{t} + \mathrm{x}_2^* = \mathrm{F_2}((\mathrm{w_1})_{\mathrm{t}-1} + \mathrm{x}_1^*, (\mathrm{w_2})_{\mathrm{t}-1} + \mathrm{x}_2^*, \dots, (\mathrm{w_N})_{\mathrm{t}-1} + \mathrm{x}_\mathrm{N}^*) \\ \vdots \quad = \quad \vdots \\ (\mathrm{w_N})_\mathrm{t} + \mathrm{x}_\mathrm{N}^* = \mathrm{F_N}((\mathrm{w_1})_{\mathrm{t}-1} + \mathrm{x}_1^*, (\mathrm{w_2})_{\mathrm{t}-1} + \mathrm{x}_2^*, \dots, (\mathrm{w_N})_{\mathrm{t}-1} + \mathrm{x}_\mathrm{N}^*) \end{array} \tag{23}$$

Let's also assume that the increments $(w_i)_t = (x_i)_t - x_i^*$ are inside the hypersphere of radius ϵ:

$$\left|\left|(x_i)_t - x_i^*\right|\right| < \epsilon \ll 1 \tag{24}$$

Thus, we can expand each of the *N* components of the F_i vector function, in a power series of *N,* w_i increments, with i going from *1*→*N.* Since ϵ is small enough, we can retain the linear terms and neglect all the others. The following set of coupled equations results from (25):

$$\begin{array}{c} (\mathrm{w_1})_{\mathrm{t}+1} = (\mathrm{w_1})_\mathrm{t}.\left(\frac{\partial \mathrm{F_1}}{\partial \mathrm{x_1}}\right)_{(\mathbf{v}^*)} + (\mathrm{w_2})_\mathrm{t}.\left(\frac{\partial \mathrm{F_1}}{\partial \mathrm{x_2}}\right)_{(\mathbf{v}^*)} + \cdots + (\mathrm{w_N})_\mathrm{t}.\left(\frac{\partial \mathrm{F_1}}{\partial \mathrm{x_N}}\right)_{(\mathbf{v}^*)} + \cdots \\ (\mathrm{w_2})_{\mathrm{t}+1} = (\mathrm{w_1})_\mathrm{t}.\left(\frac{\partial \mathrm{F_2}}{\partial \mathrm{x_1}}\right)_{(\mathbf{v}^*)} + (\mathrm{w_2})_\mathrm{t}.\left(\frac{\partial \mathrm{F_2}}{\partial \mathrm{x_2}}\right)_{(\mathbf{v}^*)} + \cdots + (\mathrm{w_N})_\mathrm{t}.\left(\frac{\partial \mathrm{F_2}}{\partial \mathrm{x_N}}\right)_{(\mathbf{v}^*)} + \cdots \\ \vdots \quad = \quad \vdots \\ (\mathrm{w_N})_{\mathrm{t}+1} = (\mathrm{w_1})_\mathrm{t}.\left(\frac{\partial \mathrm{F_N}}{\partial \mathrm{x_1}}\right)_{(\mathbf{v}^*)} + (\mathrm{w_2})_\mathrm{t}.\left(\frac{\partial \mathrm{F_N}}{\partial \mathrm{x_2}}\right)_{(\mathbf{v}^*)} + \cdots + (\mathrm{w_N})_\mathrm{t}.\left(\frac{\partial \mathrm{F_N}}{\partial \mathrm{x_N}}\right)_{(\mathbf{v}^*)} + \cdots \end{array} \tag{25}$$

where we can use (18) to eliminate the fixed points on both sides of the expressions.

We have a total of N^2 partial derivatives of functions F_i with respect to all dynamic variables x_j, calculated at fixed point v^*. We define the Jacobian matrix as:

$$\mathrm{J}(\mathrm{v}^*) = \begin{pmatrix} \left(\frac{\partial \mathrm{F}_1}{\partial \mathrm{x}_1}\right)_{(\mathrm{v}^*)} & \left(\frac{\partial \mathrm{F}_1}{\partial \mathrm{x}_2}\right)_{(\mathrm{v}^*)} & \cdots & \left(\frac{\partial \mathrm{F}_1}{\partial \mathrm{x}_\mathrm{N}}\right)_{(\mathrm{v}^*)} \\ \left(\frac{\partial \mathrm{F}_2}{\partial \mathrm{x}_1}\right)_{(\mathrm{v}^*)} & \left(\frac{\partial \mathrm{F}_2}{\partial \mathrm{x}_2}\right)_{(\mathrm{v}^*)} & \cdots & \left(\frac{\partial \mathrm{F}_2}{\partial \mathrm{x}_\mathrm{N}}\right)_{(\mathrm{v}^*)} \\ \vdots & \vdots & \vdots & \vdots \\ \left(\frac{\partial \mathrm{F}_\mathrm{N}}{\partial \mathrm{x}_1}\right)_{(\mathrm{v}^*)} & \left(\frac{\partial \mathrm{F}_\mathrm{N}}{\partial \mathrm{x}_2}\right)_{(\mathrm{v}^*)} & \cdots & \left(\frac{\partial \mathrm{F}_\mathrm{N}}{\partial \mathrm{x}_\mathrm{N}}\right)_{(\mathrm{v}^*)} \end{pmatrix} \tag{26}$$

such that (26) can be written in the compact form:

$$\mathrm{w}_{t+1} = \mathrm{J}(\mathrm{v}^*).\mathrm{w}_t \tag{27}$$

which is a linear model in form (7), whose fixed point is the origin in N-dimensional space. Therefore, in order to study the stability of the nonlinear model's fixed-point $\mathbf{v}^*$, it is sufficient to investigate the stability of the origin for the linearized model (27), that is, to study the eigenvalues of the Jacobian matrix, whose elements are constant:

$$J_{ij}(\mathrm{v}^*) = \left(\frac{\partial F_i}{\partial x_j}\right)_{(\mathrm{v}^*)} \tag{28}$$

The fixed-point $\mathbf{v}^*$ will be stable if all the Jacobian matrix eigenvalues have modules less than one. Through the coefficients of the secular equation, satisfied by the Jacobian matrix, and through the Schur criterion, it is possible to determine the fixed-point stability conditions in the linear approximation.

2.1.2.3. Hyperbolic Points

Definition*: A fixed point* v^* *is hyperbolic (or non-degenerate) if all the eigenvalues of the system's Jacobian matrix calculated at that point,* $\mathrm{J}(\mathrm{v}^*)$, *have modules other than one.*

The fixed point v^* will be non-hyperbolic if one or more eigenvalues have modules equal to one, that is, they are located precisely on the unit circle in the complex plane. There is a version of the Hartman-Grobman theorem[5] for discrete models which ensures that, if v^* is a hyperbolic fixed point, the behavior of the solutions in its neighborhood (and therefore its stability) is determined by linearization. If the fixed point is non-hyperbolic, the linearization criterion is not able to tell us whether or not the fixed point is stable, and other methods are required.

[5] This theorem can be found in the appendix.

2.1.3. The Numerical Solution of Nonlinear Systems using the Newton-Raphson Method

One of the problems with analyzing the stability of systems of nonlinear difference equations refers to the calculation of their fixed points.

When we are dealing with a linear case, the calculation is straightforward, using equation (8) presented above (provided that the matrix is singular). However, this does not apply to the nonlinear case. One possible alternative to finding the roots in this case is by following the Newton-Raphson iterative method.

In N-dimensions, according to Ruggiero and Lopes (1996), given a system of nonlinear equations:

$$\mathrm{F}: \mathrm{D} \subset \mathbb{R}^{\mathrm{n}} \rightarrow \mathbb{R}^{\mathrm{n}}, \qquad \mathrm{F} = (\mathrm{f}_1, \dots, \mathrm{f}_{\mathrm{n}})^{\mathrm{T}} \tag{29}$$

The objective is to find solutions for $\mathrm{F}(\mathrm{x}) = 0$, or equally:

$$\begin{gathered} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{gathered} \tag{30}$$

using the following notation:

$$\mathrm{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad e \quad \mathrm{F}(\mathrm{x}) = \begin{pmatrix} f_1(\mathrm{x}) \\ f_2(\mathrm{x}) \\ \vdots \\ f_n(\mathrm{x}) \end{pmatrix} \tag{31}$$

Each function $f_i(\mathrm{x})$ is a nonlinear function on x, $f_i: \mathbb{R}^n \rightarrow \mathbb{R}^n,\ \ i = 1, \dots, n$ and therefore $\mathrm{F}(\mathrm{x})$ is a nonlinear function on x, $\mathrm{F}: \mathbb{R}^n \rightarrow \mathbb{R}^n$. We are assuming that $\mathrm{F}(\mathrm{x})$ is defined on an open set $\mathrm{D} \subset \mathbb{R}^n$ and that it has continuous derivatives on that set. Furthermore, we assume that there is at least one point $\mathrm{x}^* \in \mathrm{D}$, such that $\mathrm{F}(\mathrm{x}^*) = 0$.

Our objective, then, is to use an iterative method which, from starting point $\mathrm{x}^{(0)}$, generates a sequence $\{\mathrm{x}^{(k)}\}$ of vectors and, in a convergence situation, we have:

$$\lim_{k \to \infty} \mathrm{x}^{(k)} = \mathrm{x}^* \tag{32}$$

In iterative methods, it is necessary to establish stopping criteria to accept whether point $\mathrm{x}^{(k)}$ is an exact approximation of x^* or to detect divergence in the iterative process. Since $\mathrm{F}\left(\mathrm{x}^{(\mathrm{k})}\right)$ is a vector of $\mathbb{R}^n$, we confirm that:

$$\left|\left|\mathrm{F}\left(\mathrm{x}^{(\mathrm{k})}\right)\right|\right| < \varepsilon \tag{33}$$

with $||||$ the vector norm and ε the tolerance.

That said, we have the following definition for the Newton-Raphson method:

$$f_i(x) = f_i\left(x^{(k)}\right) + \nabla f_i(c_i)^T.\left(x - x^{(k)}\right) \tag{34}$$

where f_i is a nonlinear function of the $\mathbf{x}$ variables. $\nabla f_i(c_i)^T$ is the gradient vector in c_i transposed. Approximating $\nabla f_i(c_i)^T$ by $\nabla f_i\left(x^{(k)}\right)^T$, $i = 1, \ldots, n$ we have a local linear model for $f_i(x)$ around $x^{(k)}$:

$$f_i(x) \approx f_i\left(x^{(k)}\right) + \nabla f_i\left(x^{(k)}\right)^T.\left(x - x^{(k)}\right),\ i = 1, \ldots, n \tag{35}$$

the local linear model for $F(x)$ around $x^{(k)}$ is therefore:

$$F(x) \approx L_k(x) = F\left(x^{(k)}\right) + J\left(x^{(k)}\right).\left(x - x^{(k)}\right) \tag{36}$$

For the local linear model, the next approximation $x^{(k+1)}$ will be the zero

$$L_k(x) = 0 \Leftrightarrow J\left(x^{(k)}\right).\left(x - x^{(k)}\right) = -F\left(x^{(k)}\right) \tag{37}$$

If we denote $\left(x - x^{(k)}\right)$ by $s^{(k)}$, we have $x^{(k+1)} = x^{(k)} + s^{(k)}$, where $s^{(k)}$ is the solution of the linear system:

$$J\left(x^{(k)}\right)s = -F\left(x^{(k)}\right) \tag{38}$$

Thus, computationally the method requires:

1) An evaluation of the Jacobian matrix at point $x^{(k)}$ and $F\left(x^{(k)}\right)$.

2) If $\left|\left|\mathrm{F}\left(\mathrm{x}^{(\mathrm{k})}\right)\right|\right| < \varepsilon$, stop. Otherwise:

3) The resolution of the linear system $J\left(x^{(k)}\right)s = -F\left(x^{(k)}\right)$ for the point.

4) $\mathrm{x}^{(k+1)} = \mathrm{x}^{(k)} + \mathrm{s}^{(k)}$

3. Analytical-Numerical Stability Analysis

In the two previous sections we presented the method for analyzing the stability of linear and nonlinear systems; we now propose a method that combines the aforementioned analysis with computational numerical methods. In order to analyze the stability of a nonlinear discrete model, we need to perform the following:

1) Find the vector of fixed points.

2) Use the Jacobian matrix at the fixed points vector.

3) Analyze the eigenvalues of the found matrix.

4) If all the absolute values from the eigenvalues are less than one, we have obtained a stable fixed point, otherwise we have an unstable one.

We will demonstrate these steps in the following section, using the consolidated model found in the SFC literature: "the Simplified 'Benchmark', Stock-Flow Consistent Post-Keynesian Growth Model" by Dos Santos and Zezza (2008). The advantage of using this model is due to the following: it is a model for which the authors have already developed an analytical solution. The model has already been discussed in relation to sensitivity analysis methods [Ciuffo and Rosenbaum (2015)] and there is a convergence of analytical efforts in dialogue with it.

4. An Alternative Algorithm for Stability Analysis

In this section, we present an alternative approach to simulate and compare results with the analytical-numerical method for stability analysis. The usual method (M1), presented above, consists of finding the values for the fixed points from a system of difference equations, then calculating the Jacobian matrix at fixed points and the absolute values of the eigenvalues. Finally, through the general stability criterion, the eigenvalues module is used to define whether we are facing a stable or an unstable equilibrium. If this is a two-dimensional model, we can infer through the trace and determinant whether it is a stable fixed point, an unstable point, a saddle point, an unstable focus, or a stable focus.

The alternative method (M2) described in this article consists of a combination of numerical approaches and makes use of Brute Force and Computational Ignorance (BFCI)[6] to demonstrate that there is convergence between the M1 and M2 results.

First, a domain is defined for the set of parameters to be evaluated. Next, the Monte Carlo simulation method is used to randomly generate parameters uniformly distributed within this domain interval. At each simulation *m*, a set *z* of model parameters is generated. This is given by: $\{a_i\}^m, i = 1,2, \dots, z$. From this set, the Gauss-Seidel method[7] is used to solve the system of equations for each period t, where $t = [1,2, \dots, n]$. When t=n, the algorithm stop gives us the final set of endogenous variables $\{y^i_{t=n}\}, i = 1,2, \dots, x$. Note that x must be x≤z and the equations must be linearly independent. From the final endogenous variables, we can obtain the level values and their variations, $\Delta y^i_t, i = 1,2, \dots, x$.

[6] BFCI is programming jargon which means relying on raw computing power. For example, an algorithm that uses BFCI might search through all possible options to find the single best option.

[7] According to O'Shea and Kinsella (2010), this method has been evaluated for linear SFC models. The sufficient condition for convergence is obtained when the matrix is strictly diagonally dominant.

Applying the criterion of having $\Delta y_t^i = 0 \; \forall \; i \; \in \{1,2,\dots,x\}$ in the steady state, a filter was created to select only those parameters for which the condition of $\Delta y_t^i = 0$ was fulfilled. Note that by selecting *ex ante* a *n* value for the final period of calculated *t*, we could have endogenous variables that are on the path of convergence, but which still do not have a value equal to zero. Thus, an alternative way for the filter to be made flexible is to use $\Delta \mathrm{y}_\mathrm{t}^\mathrm{i}$<tolerated, with tolerated≈0.

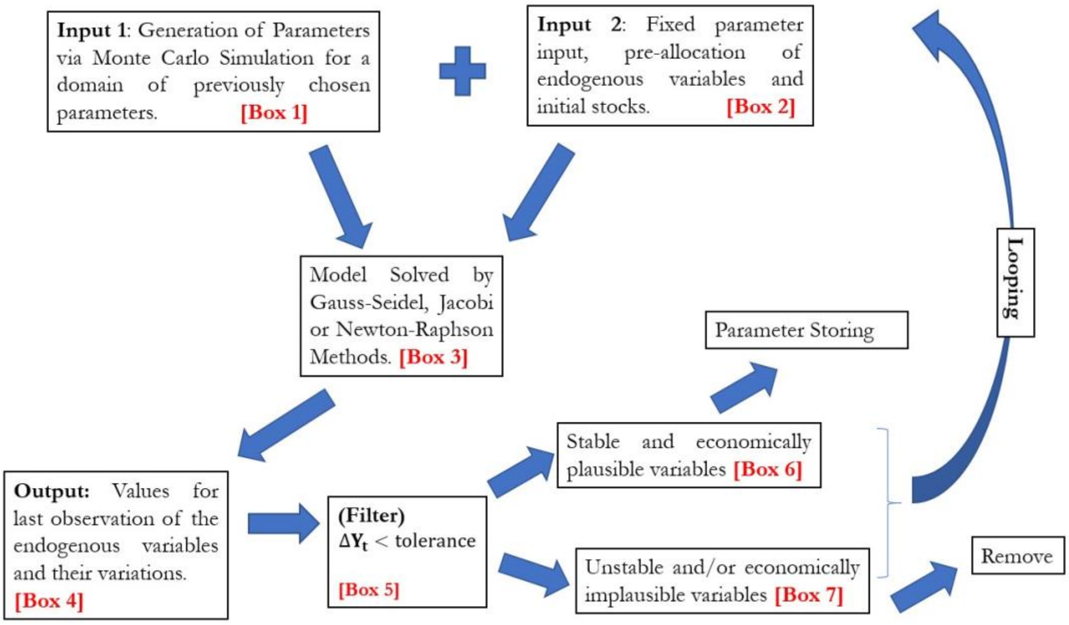
Figure 1, below, outlines the steps of the algorithm. As stated above, first, a random set of parameters is generated within a previously defined domain maintaining uniform distribution, as seen in Box 1. At the same time, other information is also applied, such as the values of the initial stocks and the exogenous parameters/variables to be held as fixed [Box 2]. Using the values provided by Box 1 and 2, the model is solved as a system of linear equations via Gauss-Seidel, or other methods (Jacobi or Newton Raphson, in nonlinear cases) for a given length t [Box 3].

Next, we store the last values of the model state variables, as well as their temporal variations, $\mathbf{\Delta Y_t}$ [Box 4]. In this last row, cases where the temporal variations are smaller than the tolerance value are filtered. If the variation is greater than the tolerance value, we assume that there is no convergence and, therefore, discard the set of parameters. The same applies if convergence is accepted, but the value of the state variable does not make economic sense (wealth or negative GDP, for example) [Box 7].

However, if we accept the convergence criterion and if the final values of the state variables fall within a range of values that make economic sense, the parameter values are stored [Box 6]. Finally, we return to the initial step via looping and exhaustively test the results via Brute Force and Computational Ignorance.

On the other hand, the alternative to the proposed algorithm (M2), which we call (M1), can be summarized as follows: First step, use only the generated parameter values to find the roots of the polynomial that determines the model's fixed points. By determining its roots, we can apply matrix numerical calculus to evaluate the eigenvalues of the Jacobian matrix under the fixed points obtained. If the modulus is less than one, the parameter is considered stable and if it is greater, unstable. A practical programming of the two methods, in MATLAB, may be found in the appendix of this work.

Figure 1- Sketch of the alternative stability algorithm (M2)



Input 1: Generation of Parameters via Monte Carlo Simulation for a domain of previously chosen parameters. [Box 1]
Input 2: Fixed parameter input, pre-allocation of endogenous variables and initial stocks. [Box 2]
Model Solved by Gauss-Seidel, Jacobi or Newton-Raphson Methods. [Box 3]
Output: Values for last observation of the endogenous variables and their variations. [Box 4]
(Filter) $\Delta Y_t <$ tolerance [Box 5]
Stable and economically plausible variables [Box 6]
Unstable and/or economically implausible variables [Box 7]
Parameter Storing
Remove
Looping

Source: Author's own

5. The Dos Santos and Zezza Benchmark (2008)

Initially, the Benchmark model has 35 equations which, if solved simultaneously for a given set of parameters, provide the temporal dynamics of the system. However, the authors show that this model can be more easily analyzed if reduced to a smaller number of equations, that is, it is possible to generate the dynamics of the system using only four equations and calculate the other hidden variables[8] during this reduction.

First, a new equation was generated through the equations that constitute aggregate demand (Consumption, Investment and Public Expenditure), providing the dynamics of the productive capacity utilization curve, which is the same as a normalized IS curve. Secondly, other equations were normalized via inherited capital, which together provided the complete dynamics of the system. These were: public debt, inherited wealth, and investment.

The equation system that gives the short-term dynamics are:

$$g_t = g_0 + (\alpha.\pi + \beta).u_t - \theta_1.il_t \tag{S.1}$$

$$b_t = \frac{[b_{t-1}.(1+ib_{t-1})+\gamma_t - \theta.u_t]}{(1+g_t)} \tag{S.2}$$

[8] 'Hidden' in the sense that the variable has been simplified within other equations. It has not been removed or discarded.

$$vh_t = \frac{[1-(1+\tau_b).(1-\mu)].ib_{t-1}.b_{t-1}+\mu.(1-\theta).\pi.u_t+\{(1-\delta).[1+ib_{t-1}.(1+\tau_b).(1-\mu)]-a\}.vh_{t-1}}{(1+g_t-\delta)} \quad \text{(S.3)}$$

$$u_t = \psi_1.A(il_t) + \psi_1.a.vh_{t-1} \quad \text{(S.4)}$$

and the aggregate parameters are:

$$\psi_1 = \frac{1}{[1-(1-\pi).(1-\theta)-\alpha_1]} \quad \text{(S.5)}$$

$$\alpha_1 = \alpha.\pi + \beta \quad \text{(S.6)}$$

$$A.(il_t) = g_o - \theta_1.(1+\tau_b).ib_t + \gamma_t \quad \text{(S.7)}$$

$$\pi = \tau/(1+\tau) \quad \text{(S.8)}$$

where we have the following parameters: α is the accelerator effect through profits, β is the exogenous accelerator effect, θ is the income tax rate, θ_1 is the interest rate effect on investment loans, γ is normalized public spending, τ is the mark-up rate, τ_b is the bank mark-up, μ is the dividend-earnings ratio, a is the propensity to consume inherited wealth, δ is the percentage of wealth that is allocated to stocks, g_0 is entrepreneurial "animal spirits" in the investment function.

By combining the parameters presented above, we arrive at the Keynesian multiplier given by (S.5), the parameter that captures both the multiplier effect and the accelerator given by (S.6), while the components of aggregate demand that are not sensitive to inherited wealth are given by (S.7). The rate of profit on revenue is given by (S.8).

The endogenous variables are respectively: investment rate, g_t; normalized public debt, b_t; normalized wealth, vh_t and capacity utilization, u_t. The only exogenous variable is the interest rate which remunerates government bonds, *ib*.

The natural way to reduce the system to two equations is to introduce the equations of u_t, given by (S.4) and g_t, given by (S.1) in equations (S.2) and (S.3). In this way, we have a system formed of two equations and now only depending on lagged and non-current variables. These are presented as (S.8) and (S.9) below:

$$b_t = \frac{b_{t-1}.(1+ib_{t-1})+\gamma_t-\theta.[\psi_1.A(il_t)+\psi_1.a.vh_{t-1}]}{1+g_0+(\alpha.\pi+\beta).[\psi_1.A(il_t)+\psi_1.a.vh_{t-1}]-\theta_1.il_t} \quad \text{(S.8)}$$

$$vh_t = \frac{[1-(1+\tau_b).(1-\mu)].ib_{t-1}.b_{t-1}+\mu.(1-\theta).\pi.[\psi_1.A(il_t)+\psi_1.a.vh_{t-1}]+\{(1-\delta).[1+ib_{t-1}.(1+\tau_b).(1-\mu)]-a\}.vh_{t-1}}{1+g_0+(\alpha.\pi+\beta).[\psi_1.A(il_t)+\psi_1.a.vh_{t-1}]-\theta_1.il_t-\delta} \quad \text{(S.9)}$$

However, in order to facilitate the calculations for the following steps, we aggregate the parameters. Thus, (S.8) and (S.9) can be simplified as (S.10) and (S.11) below:

$$b_t = \frac{b_{t-1}.z_1+z_2-z_3.vh_{t-1}}{z_4+z_5.vh_{t-1}} \quad \text{(S.10)}$$

$$vh_t = \frac{s_2+s_1.b_{t-1}+s_5.vh_{t-1}}{z_4+z_5.vh_{t-1}-\delta} \quad \text{(S.11)}$$

where the simplified parameters are described in Table 1 below.

Table 1- Simplified Parameters for the Benchmark Model

Simplified Parameter	Equals
z_1	$(1 + ib_{t-1})$
z_2	$\gamma_t - \theta.\psi_1.A(il_t)$
z_3	$\theta.\psi_1.a$
z_4	$1 + g_0 + (\alpha.\pi + \beta).\psi_1.A(il_t) - \theta_1.il_t$
z_5	$(\alpha.\pi + \beta).\psi_1.a$
s_1	$[1 - (1 + \tau_b).(1 - \mu)].ib_{t-1}$
s_2	$\mu.(1 - \theta).\pi.\psi_1.A(il_t)$
s_3	$\mu.(1 - \theta).\pi.\psi_1.a$
s_4	$(1 - \delta).[1 + ib_{t-1}.(1 + \tau_b).(1 - \mu)] - a$
s_5	$s_3 + s_4$

Knowing that in the steady state we have b^* and vh^* as fixed points, which we have through (S.10) and (S.11), we have the following equations (S.12) and (S.13) which define them:

$$b^* = \frac{b^*.z_1 + z_2 - z_3.vh^*}{z_4 + z_5.vh^*} \tag{S.12}$$

$$vh^* = \frac{s_2 + s_1.b^* + s_5.vh^*}{z_4 + z_5.vh^* - \delta} \tag{S.13}$$

Rearranging the terms in (S.12) and (S.13), after some algebraic operations, substituting one equation for another and after some parameter aggregation, we have the polynomial which, if solved, gives us the roots and thus the values of the fixed points for a set of parameters. One detail worth noting is that we are facing a polynomial of degree three. In the fundamental theorem of algebra there are as many roots as the polynomial degree; there are therefore three possible roots.

$$v_1.vh^{*3} + v_2.vh^{*2} + v_3.vh^* - v_4 = 0 \tag{S.14}$$

Table 2 below shows the values of the polynomial parameters.

Table 2- Simplified Parameters for the Steady State Polynomial

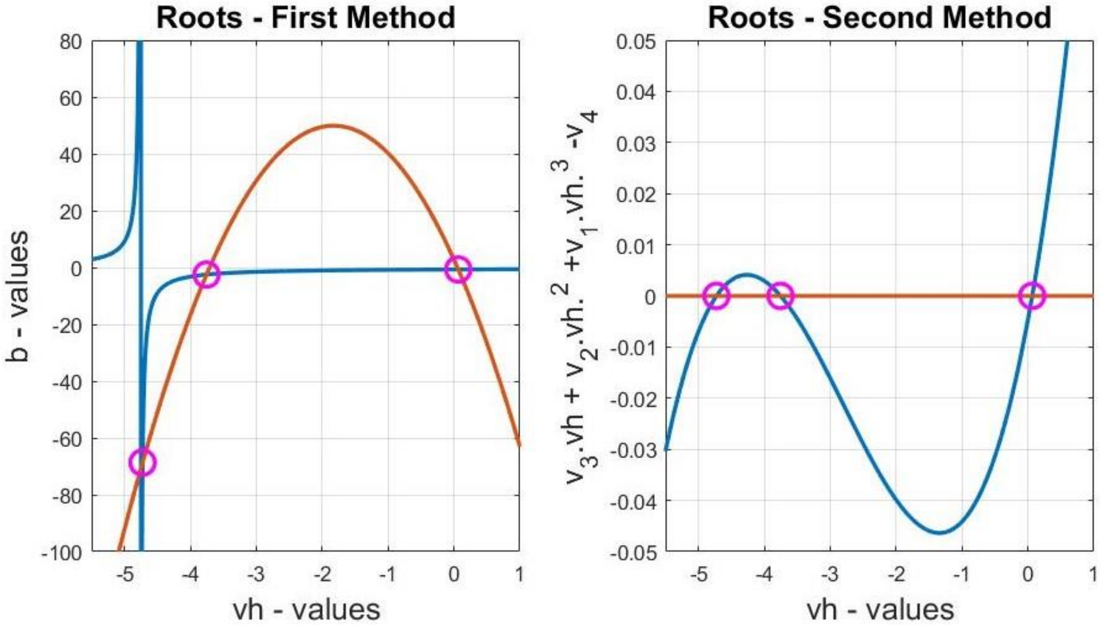
Simplified Parameter	Equals
v_1	$z_5.z_5$
v_2	$[(z_4 - \delta - s_5) + (z_4 - z_1)].z_5$
v_3	$(z_4 - z_1).(z_4 - \delta - s_5) - s_2.z_5 + s_1.z_3$
v_4	$(z_4 - z_1).s_2 + s_1.z_2$

Rearranging S.12 and taking the inverse function of (S.13), we have:

$$b^* = \frac{z_2 - z_3.vh^*}{(z_4 + z_5.vh^* - z_1)} \quad e \quad f(vh^*)^{-1} = \frac{z_5.vh^{*2} - (\delta + s_5 - z_4).vh^* - s_2}{s_1}$$

Through the above two equations and from (S.14) we can plot two graphs that show us that it is possible to find the three roots (if real) for a given set of parameters in the crossing curves (left quadrant) or the zero axis (right quadrant).

Figure 2- Polinomial Roots



Source: Author's own. Note: (1) Roots are marked by purple circles

The importance of finding the roots and, consequently, the fixed-point values for b and vh relates to the way in which the stability of a nonlinear discrete-time system of equations is analyzed. As presented in Section 1.4, we need to evaluate the Jacobian matrix for the values found at the fixed point. Once we have these, all the matrix's eigenvalues are calculated. The sufficient condition for the system's general stability is achieved when all the eigenvalues have modules less than one (if real) and when the square root of the sum of parts a and b is less than one. The system's Jacobian matrix at the fixed point is denoted by:

$$\mathbf{J}(\mathrm{b}^*, \mathrm{vh}^*) = \begin{bmatrix} \left(\frac{\partial \mathrm{vh}}{\partial \mathrm{b}_{-1}}\right)_{(\mathrm{b}^*,\mathrm{vh}^*)} & \left(\frac{\partial \mathrm{vh}}{\partial \mathrm{vh}_{-1}}\right)_{(\mathrm{b}^*,\mathrm{vh}^*)} \\ \left(\frac{\partial \mathrm{b}}{\partial \mathrm{b}_{-1}}\right)_{(\mathrm{b}^*,\mathrm{vh}^*)} & \left(\frac{\partial \mathrm{b}}{\partial \mathrm{vh}_{-1}}\right)_{(\mathrm{b}^*,\mathrm{vh}^*)} \end{bmatrix} \tag{S.15}$$

and through equations (S.10), (S.11), (S.12) and (S.13), we can calculate it in (S.16):

$$\mathrm{J}(\mathrm{b}^*, \mathrm{vh}^*) = \begin{pmatrix} \left(\frac{\mathrm{z}_1}{\mathrm{z}_4+\mathrm{z}_5.\mathrm{vh}^*}\right) & -\left[\frac{\mathrm{z}_3}{\mathrm{z}_4+\mathrm{z}_5.\mathrm{vh}^*} + \frac{\mathrm{z}_5.(\mathrm{z}_2+\mathrm{z}_1.\mathrm{b}^*-\mathrm{z}_3.\mathrm{vh}^*)}{(\mathrm{z}_4+\mathrm{z}_5.\mathrm{vh}^*)^2}\right] \\ \left(\frac{\mathrm{s}_1}{\mathrm{z}_4-\delta+\mathrm{z}_5.\mathrm{vh}^*}\right) & \left[\frac{\mathrm{s}_5}{\mathrm{z}_4-\delta+\mathrm{z}_5.\mathrm{vh}^*} - \frac{\mathrm{z}_5.(\mathrm{s}_2+\mathrm{s}_1.\mathrm{b}^*+\mathrm{s}_5.\mathrm{vh}^*)}{(\mathrm{z}_4-\delta+\mathrm{z}_5.\mathrm{vh}^*)^2}\right] \end{pmatrix} \tag{S.16}$$

In this way, once we have the parameters, we can calculate the fixed points and eigenvalues for the Jacobian matrix over the fixed points. The roots of (S.14) can be numerically evaluated using the *roots.m* function in MATLAB[9].

The economic plausibility criteria for the simulations were as follows:

$$0 < \mathrm{u}^*, \mathrm{g}^* < 1 \qquad \text{(First Condition)}$$

$$0 < \mathrm{vh}^*, \mathrm{b}^* < 10 \qquad \text{(Second Condition)}$$

$$\Delta\mathrm{b}, \Delta\mathrm{vh}, \Delta\mathrm{u}, \Delta\mathrm{g} < 10^{-15} \qquad \text{(Third Condition)}$$

The intuition for each of the three conditions is as follows: first, that both capacity utilization and investment rate are between zero and one hundred percent. The restriction of the former is a logical consequence of a Kaleckian-inspired model, which, by definition, uses some degree of idle capacity in a steady state (either to absorb an unexpected growth in demand or to create a barrier mechanism for the entry of new competitors), while the latter is simply to prevent the investment rate from being absurdly high.

Subsequently, according to the second condition, both normalized wealth and normalized public debt fall within a range between zero and ten. The idea is to create a ceiling to ensure that these sources of wealth and debt do not exceed the economy's fixed capital by more than ten.

Finally, the third condition determines that the state variables must be convergent in the steady state. Since the cut-off time to stop the algorithm is defined *ad-hoc*, the convergence process is defined as true if the variation is less than 10^{-15}. Of course, higher values may be used, depending on author preference.

[9] According to Grasselli and Pelinovsky (2008), in order to find the roots of a polynomial, the existing algorithm in "roots.m" is based on the calculation of the eigenvalues of the companion matrix. This algorithm may be of low accuracy for high degree polynomials (from 8). However, it is the most appropriate numerical method for this case, since Newton-Raphson does not guarantee convergence if the initial kick is not close to the roots.

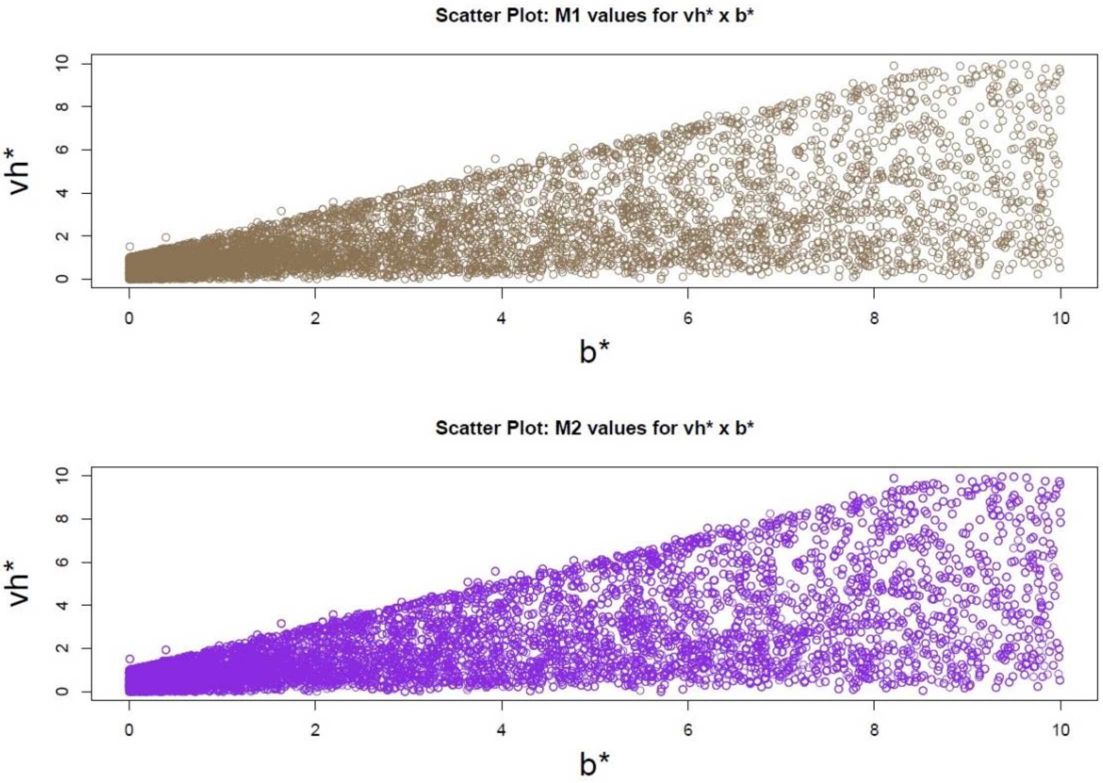
6. Results

In order to compare the results from the two methods, the same random seed was used to generate the M1 and M2 parameters. The experiment can be controlled through this equal generation of parameters, which is why the same set of parameters was used for both methods.

Briefly, for M1, a programming routine was created in MATLAB software to generate the set of parameters, calculate the fixed points, evaluate the Jacobian matrix at the fixed points and, finally, calculate the respective eigenvalues. Parameters for which there was a fixed point with eigenvalue modules below one were filtered. For M2, a programming routine was created in the same language (MATLAB) through which the parameters were generated, the model was solved via Gauss-Seidel, the last values were stored for a previously-defined time horizon and, finally, those where the variations were equal to zero were filtered.

Figure 2 below presents the scatter plots in two quadrants, one for each method. The number of random generated sets was 10^6.

Figure 3- Scatter plots for vh^* and b^* using M1 and M2



Source: Author's own.

The upper quadrant shows the results obtained applying M1 to the sets of stable points, filtered for the ordered pair of roots, with the vh* variable on the horizontal axis, and the b* on the vertical one.

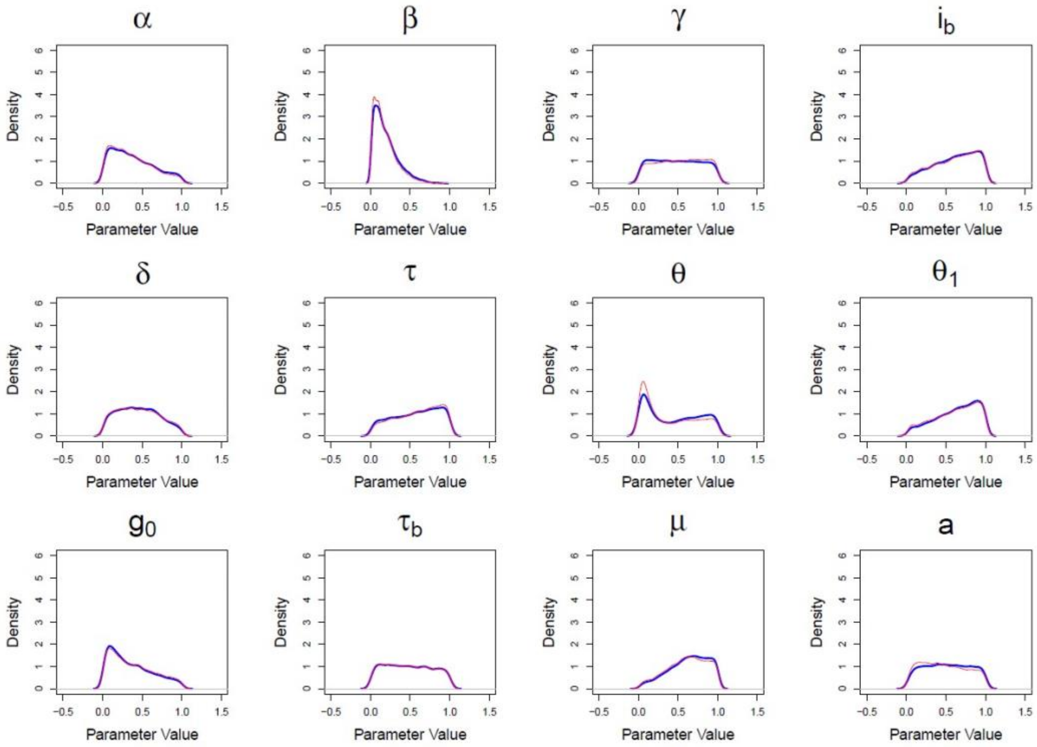
The lower quadrant presents the results obtained using M2 for the final values of the vh and b variables in the last simulation period. The first noteworthy point is that, between the economically plausible intervals (the positive quadrant for both variables), there is an overlap of the areas where the M1 and M2 points meet. This provides the first clue that we are, in fact, dealing with methods that present convergent results.

Nevertheless, to present the probability density distribution (PDF) of the parameters for which the model is stable, a Kernel density function (non-parametric method) was applied to assess whether an overlap of curves would occur. The results are shown in Figure 3 below. We note that the two methods present estimated curves with similar results (which can be visualized in their superposition[10]).

The highlight here is that the main difference between the two methods only occurred in cases where the M1 had a fixed point with three eigenvalue modules below one and when the M2 variations did not converge to zero. This occurred in just 0.145% of the cases; we explored all these in order to understand what was taking place. What we found was that this occurred with stable focus points, that is, there was spiral convergence to the fixed point, but this was due to oscillatory behavior.

[10] When the approach is parametric, statistical tests can be performed to compare whether the distribution differs statically. However, we do not know of a nonparametric method capable of doing this.

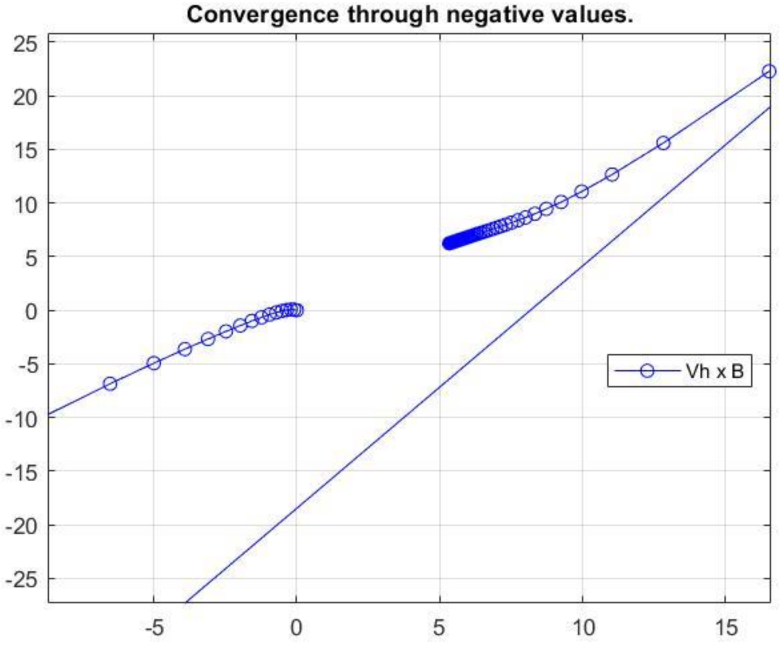
Figure 4- Kernel density estimation for the stable parameters



Source: Author's own. Key: (1) Method 1 in blue; (2) Method 2 in magenta.

Other cases of non-convergence were found in which the variation was very low, but the eigenvalues did not have a module less than one. These findings were either due to limit cycle behavior or unstable focus (with the latter more likely).

Figure 5- Convergence through negative values



Source: Author's own.

Another point worth mentioning is that the algorithm we designed seeks convergence to the equilibrium point, without restricting the mode of convergence. There is no limitation on the values that the variables can obtain during convergence. During this process it is therefore possible for the variables to assume negative values, which is a mathematical possibility, but not an economic one. From an economic point of view, one can reasonably expect that the path between the initial and final points occurs in the strictly positive quadrant, since wealth and public debt normalized by capital stock are necessarily positive. However, since the algorithm filter only takes place at the end point, it is possible for such cases to appear.

The only way to verify whether the chosen parameter set does not display this type of behavior is to verify it case by case. Figure 5 illustrates a typical convergence case where this occurs.

However, given that there are few cases in which the M2 algorithm did not obtain similar results to M1 (less than 1%), the alternative created by the former serves as a new resource for calibrating empirical SFC models, since it is not always easy to obtain data or determine the appropriate econometric technique to calibrate empirical models. As mentioned above, there are certain requirements for this: the model must be able to reach the steady-state or a balanced growth path; and typically the model should not be one that easily contains the periodic orbits that occur in nonlinear dynamics systems (such as the prey-predator model).

7. Conclusion

Most SFC models can be described as systems of difference equations. The traditional approach to finding steady-state solutions in difference equation systems involves first identifying whether the system is linear or nonlinear, and then mapping the fixed points. If the system is linear, the analysis is straightforward, calculating the eigenvalues of the main matrix of coefficients. If it is nonlinear, the system must be linearized and evaluated for the eigenvalues around the steady-state point. However, systems in SFC modeling can become so large as to make traditional stability analysis unfeasible. Given this, the method proposed in this article innovates by offering an alternative approach.

Thus, we developed an algorithm that combines three numerical techniques to map a model's stable parameters. First, a domain is defined for generating random parameter values via Monte Carlo simulation. Next, for each set of generated values, the model is solved and the final values for the system-level variables, and their last period variations, are stored. Finally, the parameters for which there is no more final period variation are separated out, so that the final variables fall within the economically plausible quadrant.

In order to compare the original technique and our alternative one, we applied both techniques, in parallel, to map the stable parameters of the model proposed by Dos Santos and Zezza, widely known in the literature as the "Simplified Benchmark". Our results show convergence in the values obtained by the two techniques when using the non-parametric method of Kernel density. In this way, this work constitutes progress, by providing an escape route for understanding and mapping parameters which may be unfeasible or impossible to map through the traditional method, because the models are too large.

To conclude, we believe that the paper's innovation resides precisely in its demonstration that a relatively simple alternative algorithm maintains equivalence with the local stability analysis routinely performed in post-Keynesian macrodynamic models. However, as is repeatedly stated in the literature on economic dynamics, models with a dimension greater than four present a great deal of difficulty for this type of analysis, whether in continuous or discrete time models. The escape route provided to overcome this difficulty allowed us to map the parameters of an SFC model that make it stable, even when it is high-dimensional.

Furthermore, one could use the same algorithm to investigate parameters that generate different growth and capital accumulation regimes, or even restrict them, in order to find observable values that resemble those of a country's real economy.

References

Blecker, R. (2011). "Open economy models of distribution and growth", In: Hein, E. and Stockhammer, E. A Modern Guide to Keynesian Macroeconomics and Economic Policies, Cheltenham, UK: Edward Elgar.

Caiani, A *et al.* (2016). "Agent based-stock flow consistent macroeconomics: Towards a benchmark model". *Journal of Economic Dynamics and Control*, 69, p. 375-408. https://doi.org/10.1016/j.jedc.2016.06.001

Caverzasi, E.; Godin, A. (2014). "Post-Keynesian stock-flow-consistent modelling: a survey". *Cambridge Journal of Economics*, 39(1), p. 157-187. DOI: https://doi.org/10.1093/cje/beu021

Ciuffo, B.; Rosenbaum, E. (2015). "Comparative numerical analysis of two stock-flow consistent post-Keynesian growth models". *European Journal of Economics and Economic Policies: Intervention*, 12(1), p. 113-134. DOI: https://doi.org/10.4337/ejeep.2015.01.09

Dos Santos, C. H.; Zezza, G. (2008) "A simplified, 'Benchmark', Stock-Flow Consistent Post-Keynesian Growth Model". *Metroeconomia*. 59(3), p.441-478. DOI: https://doi.org/10.1111/j.1467-999X.2008.00316.x

Elaydi, S. (2005) *An Introduction to Difference Equations*. Nova Iorque. Springer.

Lucas, R. E. (1976). "Econometric policy evaluation: A critique". *Carnegie-Rochester Conference Series on Public Policy*, 1, p. 19–46. DOI: https://doi.org/10.1016/S0167-2231(76)80003-6

Kinsella, S.; O'shea, T. (2010). "Solution and Simulation of Large Stock Flow Consistent Monetary Production Models via Gauss Seidel Algorithm". *Journal of Policy Modeling*. DOI: http://doi.org/10.2139/ssrn.1729205

Gandolfo, G. (2009). *Economic Dynamics*. 4th Edition. Springer.

Godley, W.; Lavoie, M. (2007). *Monetary economics: an integrated approach to credit, money, income, production and wealth.* 1ª ed. Nova York: Palgrave Macmillan.

Godin, A.; Aliti, G. T. Kinsella, S. (2012). “Method to Simultaneously Determine Stock, Flow and Parameter Values in Large Stock-Flow Consistent Models”. *Working Paper*. DOI: http://dx.doi.org/10.2139/ssrn.2094996

Grasselli, M.; Pelinovsky, D. (2008). *Numerical Mathematics*. Jones & Bartlett Learning.

Macedo e Silva, A. C. (2008). “Missing Details and Conspicuous Absences: From the Treatise to the "General Theory"”. *Journal of Post Keynesian Economics*, 31(2), p. 325-44. DOI: http://doi.org/10.2753/PKE0160-3477310207.

Macedo e Silva, A. C.; Dos santos, C. H. (2011). “Peering over the edge of the short period? The Keynesian roots of stock-flow consistent macroeconomic models”. *Cambridge Journal of Economics*, 35(1), p. 105-124. DOI: https://doi.org/10.1093/cje/bep083

O’shea, T.; Kinsella, S. (2010). “Solution and Simulation of Large Stock Flow Consistent Monetary Production Models via the Gauss Seidel Algorithm”. *Journal of Policy Modeling, Forthcoming.* DOI: http://dx.doi.org/10.2139/ssrn.1729205

Ruggiero, M. A.; Lopes, V. L. R. (1996). *Cálculo Numérico: aspectos teóricos e computacionais*. 2ª edição. São Paulo: Pearson Makron Books.

Shone, R. (2002). *Economic Dynamics. Phase Diagrams and their Economic Application.* 2nd Edition. Cambridge University Press. New York.

Viana, R. L. (2012). Introdução à Dinâmica Não-Linear e Caos em Economia. Notas de Aula. Universidade Federal do Paraná.