



From Two to One Index Isomorphism in Optimization Program for Quarterly Disaggregation of Annual Times Series

Raïmi Aboudou Essessinou^{1*}, Guy Degla² and Babacar Mbaye Ndiaye³

¹National Institute of Statistics and Economy Analysis (INSAE) / Institute of Mathematics and Physics, University of Abomey-Calavi (UAC), Benin.

²Institute of Mathematics and Physics, University of Abomey-Calavi (UAC), Benin.

³Laboratory of Mathematics of Decision and Numerical Analysis, University of Cheikh Anta Diop, Dakar, Senegal.

Authors' contributions

This work was carried out in collaboration among all the authors. Author RAE designed the study, managed the simulations and wrote the first draft of the manuscript. The mathematical analysis of the models was performed by authors GD and BMN. Furthermore author GD managed the structural consistency and the English language, while author BMN controlled the simulations and the statistical analysis. The three authors managed the literature review, read and approved the final manuscript.

Article Information

DOI: 10.9734/JAMCS/2019/v34i1&230199

Editor(s):

(1) Dr. Leo Willyanto Santoso, Assistant Professor, Department of Informatics, Petra Christian University, Indonesia.

Reviewers:

(1) Olumide Adesina, Olabisi Onabanjo University, Nigeria.

(2) Diab Zouhair, Université Larbi Tébessi, Tébessa – Algérie, Algeria.

(3) Pasupuleti Venkata Siva Kumar, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering & Technology, India.

Complete Peer review History: <https://sdiarticle4.com/review-history/51632>

Received: 18 July 2019

Accepted: 30 September 2019

Published: 12 October 2019

Original Research Article

Abstract

The quarterly disaggregation of an annual economic aggregate, by a mathematical method with a cyclical indicator, gives rise to a problem of minimization to make the quarterly economic aggregate smooth. This involves two indexes for the quarter and the year, which sometimes can make the resolution algorithm less efficient if the problem is large. In this paper we propose a method of indexing quarterly variables based on an isomorphic transformation of a two-index program into a one-index program, in order to minimize the cost of the algorithm of resolution. This method of continuous indexing of variables, applied to national accounts, shows that the algorithm with a single index is more efficient than the algorithm with two indexes when solving the optimization program of the quarterly disaggregation.

Keywords: Isomorphism; indexing variables; optimization; algorithms performance; quarterly disaggregation.

*Corresponding author: E-mail: raimi.essessinou@imsp-uac.org;

1 Introduction

Quarterly disaggregation of macroeconomic aggregates started in the 1950s and several methods have been developed. The mathematical methods using high frequency indicators allow quarterly estimates of the annual series of the economic aggregate by proportional divisions according to a quarterly indicator of economic activity which is related to the economic aggregate (with a strong correlation), also following a fairly smooth quarterly path, according to certain criteria. The statistical relationship between the annualized indicator and the corresponding economic aggregate is then estimated, and it is postulated that this relation observed with the annual data is also valid on the quarterly data. The disaggregation can be challenging due to lack of instrumentation at sufficient resolution, as mentioned by Shiva R., Iyer et al. [1] and Sangjin [2].

Let considering an annual series of arbitrary economic aggregate X , observed over a period of T year and a quarterly time series of an economic activity's indicator I related to X and which is noticed over all quarters of the observation period. Then, the proportionality relation can be applied to each quarterly value of the business cycle indicator. So, having the value $I_{i,y}$ of the business cycle indicator in the quarter i of the year y , to disaggregate the annual national accounts amounts to building a first estimation of the unknown quarterly economic aggregate $X_{i,y}$ using the relationship (1) as formulated Essessinou [3].

$$\hat{X}_{i,y} = \hat{a} \times I_{i,y} + \varepsilon_{i,y} \quad (1)$$

where $y \in \{1,2,3, \dots, T\}$, $i \in \{1,2,3,4\}$, $\varepsilon_{i,y}$ is the error term called smoothing parameter of the estimated quarterly series for quarter i of year y and \hat{a} is the estimated proportional coefficient. The econometric estimation technics of \hat{a} are presented in [4].

Several methods are used to make the series of the estimated quarterly economic aggregate $\hat{X}_{i,y}$ sufficiently smooth. Most of these smoothing methods consist in solving the program (P) below:

$$(P) \left\{ \begin{array}{l} \min_{X_{i,y}} \sum_{y=1}^T \sum_{i=1}^4 (\Delta u_{i,y})^2 \\ Sc \sum_{i=1}^4 u_{i,y} = \hat{u}_y; \forall y = 1,2, \dots, T \end{array} \right.$$

Where

$u_{i,y}$: is a function of the quarterly economic aggregate $X_{i,y}$ and gets in the form $u_{i,y} = f(X_{i,y})$, f being a function defined from R to R which is continuous and at least twice differentiable. Note that $u_{i,y}$ refers to quarter i of year y , so that $u_{1,y}$ is the value for first quarter of the current year y and $u_{4,y-1}$ is the value of the fourth quarter of the previous year where $(y - 1)$.

$\Delta u_{i,y}$: is the consecutive difference of the terms of $u_{i,y}$ defined by:

$$\left\{ \begin{array}{l} \Delta u_{1y} = u_{1y} - u_{4,y-1}; \forall y = 2, \dots, T; \\ \Delta u_{i,y} = u_{i,y} - u_{i-1,y}; \forall i = 2,3,4 \wedge \forall y = 1,2, \dots, T \end{array} \right.$$

\hat{u}_y : is a given value of the observations for the year y balancing the quarterly estimated and annualized data with the observed annual data;

The efficient solvability of the problem (P) needs a variable indexation which must be done judiciously in order to optimize the cost of the numerical resolution algorithm with exponential complexity in case of

simplex method, as the running time may be an exponential function of the dimension of the problem [5], roughly speaking. Indeed, it is always possible to improve the running time of an algorithm without increasing the cost as it is set up by Guy [6], Ji-Bo and Jian-Jun [7] in order to have polynomial running time as raised by Anthony and Shahin [8], Jon through theorem 4.3.1 [9] or theorem 3.1 [10]. Hence, for an algorithm of running time $O(n)$ of the problem (P), one can find another algorithm with reduced execution time $O(\log n)$ which solves the same problem [6]. For example, in the framework of the implementation of a heuristic for the minimization of the cost for a flow problem, the performance of a three-index algorithm has been improved by a two-index algorithm, in terms of running time and the number of solved instances [11]. Moreover, in case of multi-index problem, the running time of an algorithm can be reduced to polynomial time by adopting a parallelism approach as in [12,7].

Thus, the problem that arises is that of choosing an indexation of the associated variables with the sub-annual time series in order to optimize the numerical algorithm of the program (P). In this paper, the two-index program (P) has been transformed into a one-index program by isomorphism, with the aim to minimize the cost of the algorithm for solving the quarterly disaggregation problem.

2 Theoretical Methodology

Let start by recalling that for the quarterly disaggregation by an indirect mathematical method with indicator, we have the following information:

- T : the number of years of observation of the economic aggregate;
- X : the series of economic aggregate given, observed over the entire period of T years;
- I : the series of the quarterly indicator related to the given industry, observed over all quarters of the T years period;

The following indexes are chosen for the parameterizations of the variables:

- $i \in \{1,2,3,4\}$, index of quarters: $i = 1$ for the first quarter, $i = 2$ for the second quarter, $i = 3$ for the third quarter and $i = 4$ for the fourth quarter
- $y \in \{1,2,3, \dots, T\}$, years index;

Thus, the variables X and I for the corresponding quarterly series are indexed by the quarter i and the year y by:

- $X_{i,y}$: the value of the economic aggregate for the quarter i of the year y ;
- $I_{i,y}$: the value of the indicator for the quarter i of the year y .

Definition 1:

On the basis of the year and quarter indexes presented above, the following sets are defined:

- $S_4 = \{1,2,3,4\}$: designate the set of quarterly indexes for each year;
- $S_T = \{1,2,3,4, \dots, T\}$: designate the set of indexes of years;
- $S_{4T} = \{1,2,3,4, \dots, 4T\}$: designate the set of quarter numbers continuously from the first year of observation to the last year of observation T .

S_{4T} is illustrated by the following configuration:

Year y	1				2				3				...	T			
Quarter i	1	2	3	4	1	2	3	4	1	2	3	4	...	1	2	3	4
S_{4T}	1	2	3	4	5	6	7	8	9	10	11	12	...	$4T-3$	$4T-2$	$4T-1$	$4T$

Definition 2:

The double index transformer operator (quarter, year) is defined by:

$$\begin{aligned} S: S_4 \times S_T &\rightarrow S_{4T} \\ (i, y) &\mapsto t = 4(y - 1) + i \end{aligned} \tag{2}$$

We note for each $y \in S_T$, $S_{<4 \times y>} = \{4(y - 1) + i; i \in S_4\}$, the ordered set of quarterly indexes belonging to S_{4T} .

Lemma 1. If T is finite, the subsets $S_{<4 \times y>}$ indexed by y forms a partition of S_{4T} .

Proof:

For $1, 2, 3, \dots, T$, it is trivial to notice that one has:

$$\begin{aligned} S_{<4 \times 1>} &= \{1, 2, 3, 4\}; S_{<4 \times 2>} = \{5, 6, 7, 8\}; S_{<4 \times 3>} = \{9, 10, 11, 12\}; \dots; \\ S_{<4 \times j>} &= \{4j - 3, 4j - 2, 4j - 1, 4j\}; \dots; S_{<4 \times T>} = \{4T - 3, 4T - 2, 4T - 1, 4T\} \end{aligned}$$

Thus, we have:

$$\begin{aligned} S_{4T} &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, \dots, 4T\} \\ &= \{1, 2, 3, 4\} \cup \{5, 6, 7, 8\} \cup \dots \cup \{4T - 3, 4T - 2, 4T - 1, 4T\} \\ &= \bigcup_{y=1}^T \{4y - 3, 4y - 2, 4y - 1, 4y\} \\ S_{4T} &= \bigcup_{y=1}^T S_{<4 \times y>} \quad \square \blacksquare \end{aligned}$$

Theorem 1. If T is finite, the operator S of the definition 2 satisfies the following properties:

- (a)- S is a bijective application
- (b)- S allows the number of quarters to be continuously numbered over the entire length of observation period of length T years.

Proof:

We assume that T is finite.

(a)- S is an application because for everything

$y \geq 1$ et $1 \leq i \leq 4$, (i, y) unique taken in $S_4 \times S_T$, $4(y - 1) + i$ is a unique number belonging to $S_{4T} \subset N$, according to the properties of multiplication and addition in N .

Let (i, y) et (i', y') be two elements of $S_4 \times S_T$ so that $(i, y) = S(i', y')$. So we have $4(y - 1) + i = 4(y' - 1) + i'$, which implies $4(y - y') + (i - i') = 0$.

We know that $y \geq 1$ et $y' \geq 1$ et $1 \leq i, i' \leq 4$. So we have $0 \leq |i - i'| < 4$ and therefore $(i - i')$ can't be multiple of 4 so that $4(y - y') = -(i - i')$. So, the relationship $4(y - y') + (i - i') = 0$ necessarily implies $(y - y') = 0$ et $(i - i') = 0$. Thus $(i, y) = (i', y')$ and we deduce that S is injective by using the definition 1.1.4 in [13].

The surjectivity of S results from the lemma 1. Indeed, $\forall t \in S_{4T}$, there exists at least one subset $S_{<4 \times y>}$ which contains t . Let $y_0 \in S_T$ such that $t \in S_{<4 \times y_0>}$. So there is also $i_0 \in S_4$ such that $t = 4(y_0 - 1) + i_0$.

So we deduce that $\forall t \in S_{4T}$, it exists $(i_0, y_0) \in S_4 \times S_T$ such that $t = S(i_0, y_0)$. Hence, S is surjective according to the definition 1.1.4 in [13].

(b)- The proof is by recurrence on the index year y

For $y = 1$ we have $\mathcal{S}(i, 1) = i, \forall i \in S_4$ et $S_{\langle 4 \times 1 \rangle} = \{i; i \in S_4\} = \{1, 2, 3, 4\}$; quarters are therefore numbered in a continuous and orderly manner 1, 2, 3, 4.

For $y = 2$ one obtain $\mathcal{S}(i, 2) = 4 + i, \forall i \in S_4$ et $S_{\langle 4 \times 2 \rangle} = \{4 + i; i \in S_4\} = \{5, 6, 7, 8\}$.

So for $y = 2$, the quarters are indexed 5, 6, 7, 8 respectively in order; the first quarter of the year $y = 2$ carries the index 5, the second quarter the index 6, the third quarter, the index 7 and the fourth quarter, the index 8.

As $S_8 = S_{4 \times 2} = S_{\langle 4 \times 1 \rangle} \cup S_{\langle 4 \times 2 \rangle} = \{1, 2, 3, 4, 5, 6, 7, 8\}$, then \mathcal{S} allows to index the quarters of the first two years continuously in the following order: 1, 2, 3, 4, 5, 6, 7, 8. This is done in an iterative manner until the order $y = T$.

So, as $S_{4T} = \bigcup_{y=1}^T S_{\langle 4 \times y \rangle}$ et $S_{\langle 4 \times y \rangle} \cap S_{\langle 4 \times y' \rangle} = \emptyset; \forall y \neq y'$, we get that \mathcal{S} indexes quarters continuously in ascending order over the entire observation period and for the quarter i of the year y ; $i = 1, 2, 3, 4$ and $y = 1, 2, 3, \dots, T$. Thus $\mathcal{S}(i, y) = 4(y - 1) + i$ is the continuous index of the quarter i of the year y and then, for each year y , $t = 4y - 3$ is the first quarter, $t = 4y - 2$ is the second quarter, $t = 4y - 1$ is the third quarter and $t = 4y$ is the fourth quarter of the year y . $\square \blacksquare$

Theorem 2 [14]. Let E and F be two vector spaces of finite dimension on R . Then E and F are isomorphic if and only if $\dim(E) = \dim(F)$.

Proof:

See in [14] Theorem 8.2.2.1, p.197

Theorem 3 [14]. Let E and F be two real vector spaces of finite dimensions n and let u be a linear mapping from E to F, that $isu \in L(E, F)$. Then, the following properties are equivalent.

- a. u is an isomorphism
- b. u is injective
- c. u is surjective
- d. u is of rank n .

Proof:

See in [14] Corollary 8.2.2.51, p.199

Remark 1:

The treatment and constructions performed on the basis of the variable X relating to the economic aggregate studied could be applied to variable I relating to the quarterly indicator linked to the economic aggregate. Thus, the final result obtained on the indexing of X is valid for the indexation of I .

Definition 2:

We designate by $\mathcal{X}_{S_{\langle 4 \times y \rangle}} = \{X_{i,y} : i \in S_4\}$, the economic aggregate X values of the four (4) quarters of the year y ; $y = 1, 2, 3, \dots, T$. We notice that $\mathcal{X}_{S_{\langle 4 \times y \rangle}}$ is the set of quarterly values of economic aggregate X indexed by $i = 1, 2, 3, 4$ pour $y = 1, 2, 3, \dots, T$.

We also designate $\mathcal{X}_{S_{4T}} = \{X_t : t = \mathcal{S}(i, y); i \in S_4 \text{ et } y \in S_T\}$, by transposition, all the quarterly values of the economic aggregate X indexed by the elements of S_{4T} .

To facilitate the manipulation of operations on $\mathcal{X}_{S_{<4 \times y>}}$, the following notations are adopted as conventions:

$$(X_{i,y})_{i,y} + (Y_{i,y})_{i,y} = (X + Y)_{i,y} \quad (3)$$

$$\lambda(X_{i,y})_{i,y} = (\lambda X)_{i,y}, \quad \forall \lambda \in \mathbb{R} \quad (4)$$

$$(X_t)_t + (Y_t)_t = (X + Y)_t \quad (5)$$

Theorem 4. If T is finite, the sets $\bigcup_{y=1}^T \mathcal{X}_{S_{<4 \times y>}}$ and $\mathcal{X}_{S_{4T}}$ are isomorphic and there exists a surjective application from S_{4T} to $\mathcal{X}_{S_{4T}}$ which allows to continuously indexing the quarterly values of the economic aggregate over the entire observation period of the annual series.

Proof:

The proof will be done by borrowing an apparent approach to that of the isomorphism factorization theorem.

First, note that $\bigcup_{y=1}^T \mathcal{X}_{S_{<4 \times y>}}$ and $\mathcal{X}_{S_{4T}}$ are two real vector subspaces and

$$\dim \left(\bigcup_{y=1}^T \mathcal{X}_{S_{<4 \times y>}} \right) = \sum_{y=1}^T \dim \left(\mathcal{X}_{S_{<4 \times y>}} \right) = 4T,$$

as $\mathcal{X}_{S_{<4 \times y>}} = \{X_{i,y} : i = 1, 2, 3, 4\}$ implies that $\dim \left(\mathcal{X}_{S_{<4 \times y>}} \right) = 4 ; \forall y = 1, 2, 3, \dots, T$.

Similarly, $\dim \left(\mathcal{X}_{S_{4T}} \right) = \text{card} \{X_t : t = \mathcal{S}(i, y) ; i = 1, 2, 3, 4 \text{ et } y = 1, 2, 3, \dots, T\} = 4T$ and we deduce that $\dim \left(\bigcup_{y=1}^T \mathcal{X}_{S_{<4 \times y>}} \right) = \dim \left(\mathcal{X}_{S_{4T}} \right) = 4T$. The real vector subspaces $\bigcup_{y=1}^T \mathcal{X}_{S_{<4 \times y>}}$ and $\mathcal{X}_{S_{4T}}$ are therefore of the same size $4T$. Now consider the mapping Π defined from

$S_4 \times S_T$ to $\bigcup_{j=1}^T \mathcal{X}_{S_{<4 \times j>}}$ by

$$\begin{aligned} \Pi : S_4 \times S_T &\rightarrow \bigcup_{j=1}^T \mathcal{X}_{S_{<4 \times j>}} \\ (i, y) &\mapsto X_{i,y} \end{aligned}$$

Π allows associating with each time index $(i, y) \equiv (\text{quarter } i, \text{ year } y)$, the values $X_{i,y}$ of the economic aggregate corresponding to the quarter i and the year y . Thus, by definition Π is surjective because any value of the quarterly economic aggregate is associated with a given quarter of a given year.

Then consider the application T_S which is the action of \mathcal{S} on $\bigcup_{j=1}^T \mathcal{X}_{S_{<4 \times j>}}$ defined by:

$$\begin{aligned} T_S : \bigcup_{j=1}^T \mathcal{X}_{S_{<4 \times j>}} &\rightarrow \mathcal{X}_{S_{4T}} \\ (X_{i,y})_{i,y} &\mapsto (X_{t=\mathcal{S}(i,y)})_t \end{aligned}$$

By definition T_S is continuous mapping as renumbering of real variables. It remains to show that T_S is linear.

Let $(X_{i,y})_{i,y}$ and $(Y_{i,y})_{i,y}$ two elements of $\bigcup_{j=1}^T \mathcal{X}_{S_{<4 \times j>}}$, For given (i, y) , let $(\alpha, \beta) \in \mathbb{R}^2$.

We have

$$\begin{aligned} \alpha (X_{i,y})_{i,y} + \beta (Y_{i,y})_{i,y} &= (\alpha X + \beta Y)_{i,y} \in \cup_{j=1}^T \mathcal{X}_{S_{<4 \times j>}} \text{ and} \\ T_S [\alpha (X_{i,y})_{i,y} + \beta (Y_{i,y})_{i,y}] &= T_S [(\alpha X + \beta Y)_{i,y}] \\ &= (\alpha X + \beta Y)_{t=S(i,y)} \\ &= (\alpha X)_{t=S(i,y)} + (\beta Y)_{t=S(i,y)} \\ &= \alpha (X_{t=S(i,y)})_t + \beta (Y_{t=S(i,y)})_t \\ T_S [\alpha (X_{i,y})_{i,y} + \beta (Y_{i,y})_{i,y}] &= \alpha T_S ((X_{i,y})_{i,y}) + \beta T_S ((Y_{i,y})_{i,y}). \end{aligned}$$

So T_S is linear and continuous from $\cup_{y=1}^T \mathcal{X}_{S_{<4 \times y>}}$ to $\mathcal{X}_{S_{4T}}$. Moreover, since the real vector subspaces $\cup_{y=1}^T \mathcal{X}_{S_{<4 \times y>}}$ and $\mathcal{X}_{S_{4T}}$ are of the same finite dimension $4T$, using Theorem 2 and theorem 3, we deduce that T_S is an isomorphism. Therefore, $\cup_{y=1}^T \mathcal{X}_{S_{<4 \times y>}}$ and $\mathcal{X}_{S_{4T}}$ are isomorphic. Furthermore, by Lemma 1 and by transposition $(\mathcal{X}_{S_{<4 \times j>}})_{y=1,2,3,\dots,T}$ forms a partition of $\mathcal{X}_{S_{4T}}$.

Thus, by applying the Theorem 1 by transposition, we deduce that T_S makes it possible to index the quarterly values of the economic aggregate continuously (by renumbering) over the entire observation period of length T years.

The continuous indexing of quarterly series variables is obtained by mapping $\bar{\Pi}$ defined by:

$$\begin{aligned} \bar{\Pi} : S_{4T} &\rightarrow \mathcal{X}_{S_{4T}} \\ t &\mapsto X_{t=S(i,y)} \end{aligned}$$

We then obtain the diagram below:

$$\begin{array}{ccc} S_4 \times S_T & \xrightarrow{\mathcal{S}} & S_{4T} \\ \downarrow \Pi & & \downarrow \bar{\Pi} \\ \cup_{j=1}^T \mathcal{X}_{S_{<4 \times j>}} & \xrightarrow{T_S} & \mathcal{X}_{S_{4T}} \end{array}$$

We have $\bar{\Pi} = T_S \circ \Pi \circ \mathcal{S}^{-1}$ and therefore $\bar{\Pi}$ is surjective as T_S is bijective (isomorphism) Π is surjective and \mathcal{S}^{-1} is bijective. ■

Definition 3:

With the results of Theorem 4, we define on $X_{S_{4T}} = \{X_{i,y}; i \in S_4; y \in S_T\}$, the program (P') of equivalent smoothing by quarterly indexing continues in the form below:

$$(P') \left\{ \begin{array}{l} \text{Min} \sum_{t=2}^{4T} (\Delta u_t)^2 \\ \text{Sc} \sum_{t=4y-3}^{4y} u_t = \hat{u}_y; \forall y = 1, 2, \dots, T \end{array} \right.$$

\hat{u}_y is a given value of the observations for the year y which makes it possible to establish the balance between the estimated and annualized quarterly data and the observed annual data;

u_t is a function of the quarterly economic aggregate X_t ; $u_t = f(X_t)$, f defined previously.

Definition 4:

Let $TP(n)$ be the time used by an algorithm to solve a problem of size n . Let $p(n)$ be the maximum memory actually required for the algorithm. The cost of this algorithm is then defined as follows: $C(n) = p(n) \times TP(n)$.

Remark 2:

Taking into account the cost of an algorithm is intended to take into account both the running time but also the (maximum) number of processors used to obtain this running time [6]. Thus, for the same optimization problem to be solved, an algorithm that uses (asymptotically) less time and memory than another is therefore preferable [6].

Lemma 2. The solving algorithm of (P') provides a more optimal cost with respect to the resolution algorithm of (P).

Proof:

The fundamental difference to be noted in the two programs is the expression of the objective function. In each program, the algorithm will evaluate the objective function differently. The constraints will be evaluated identically.

We denote by F the value of the objective function and by the differential of u . The corresponding algorithms will include for example the following pieces of code for the resolution of the program.

Program (P)	Program (P')
<pre> Obj = 0 For y starting from 1 to T do So = 0 For i starting from 1 to 4 do If i = 1 then Du[1, y] = u[1, y] - u[1, y-1] else Du[i, y] = u[i, y] - u[i-1, y] End (if) Sum[y] = So + Du[i, y] * Du[i, y] i = i + 1 End (for i) F = Obj + Sum[y] y = y + 1 End (for y) Display F </pre>	<pre> Obj = 0 For t starting from 2 to 4T do Du[t] = u[t] - u[t-1] Sum [t] = Obj + Du[t] * Du[t] F = Obj + Sum[t] t = t + 1 End (for t) Display F </pre>

It is clear that the number of code lines corresponding to the program (P') is much smaller than that of the program (P).

Moreover, the complexity of the resolution algorithm is higher in the case of the program (P) than in the case of the program (P'). Indeed, the piece of code for the program (P) seems to perform $(4T - 1)$ differentiations (subtractions) and the same number of places of storage by the machine of the values assigned to the variables $Du[i, y]$. However, in counting, we ignore the many internal calculations that the

computer (the compiler) would do to give the results. A computer usually stores a $4 \times T$ size double-entry array as a simple array of $4T$ length and thus store the $u[i, y]$ elements at position $4(y - 1) + i$ of the table [15]. Thus, in the case of the program (P), each time that one wishes to inform the value of $u[i, y]$, one would need to carry out in addition, a subtraction, a multiplication and an addition. Moreover, when the computer would like to increment the index y (respectively i), it would make a comparison to see if $y > T(i > 4)$.

As one can notice, such a detailed internal analysis of the algorithm by the computer consumes enough resources (time and internal memory) and does not attract. The algorithm of the program (P') therefore seems more optimal than that of the program(P). ■

3 Practical Example on National Accounts

The methodology of continuous indexing is applied on Benin national accounts data.

3.1 Description of the data

In the context of the application, we consider the added value of the " Food industry" branch whose associated variable with the national accounts is marked X . The indicator linked to the accounts of this branch is " Production index of food industries"; the associated variable with this indicator is rated I . We observe the annual series of X over the period 1999-2015, and the quarterly series of I over the period 1st quarter 1999 (1999Q1) till to the 4th quarter 2015 (2015Q4). In total, the annual series is observed over a period of length $T = 17$ years.

Table 1. Data presentation

Year of the period	Value of annual observed X	Two indexes become \implies		One index $t = S(i, y)$	Indicator I	Quarterly value needed for X
		Year index y	Quarter index i			
1999	240.978	1	1	1	72.685	X_1
			2	2	74.477	X_2
			3	3	72.938	X_3
			4	4	74.233	X_4
2000	259.213	2	1	5	74.351	X_5
			2	6	76.686	X_6
			3	7	75.201	X_7
			4	8	77.219	X_8
2001	274.937	3	1	9	79.084	X_9
			2	10	82.117	X_{10}
			3	11	80.779	X_{11}
			4	12	82.843	X_{12}
2002	285.845	4	1	13	84.647	X_{13}
			2	14	85.703	X_{14}
			3	15	84.100	X_{15}
			4	16	87.226	X_{16}
2003	286.773	5	1	17	90.742	X_{17}
			2	18	90.454	X_{18}
			3	19	88.290	X_{19}
			4	20	90.116	X_{20}
2004	293.805	6	1	21	89.554	X_{21}
			2	22	92.504	X_{22}
			3	23	88.976	X_{23}
			4	24	89.893	X_{24}

Year of the period	Value of annual observed X	Two indexes become == >		One index $t = S(t, y)$	Indicator I	Quarterly value needed for X
		Year index y	Quarter index i			
2005	307.373	7	1	25	90.415	X_{25}
			2	26	93.084	X_{26}
			3	27	102.803	X_{27}
			4	28	94.559	X_{28}
2006	315.997	8	1	29	97.486	X_{29}
			2	30	95.216	X_{30}
			3	31	94.622	X_{31}
			4	32	97.147	X_{32}
2007	292.512	9	1	33	97.709	X_{33}
			2	34	98.616	X_{34}
			3	35	101.063	X_{35}
			4	36	102.612	X_{36}
2008	238.604	10	1	37	101.705	X_{37}
			2	38	103.531	X_{38}
			3	39	104.241	X_{39}
			4	40	109.076	X_{40}
2009	228.056	11	1	41	106.365	X_{41}
			2	42	108.408	X_{42}
			3	43	107.512	X_{43}
			4	44	111.710	X_{44}
2010	214.080	12	1	45	107.950	X_{45}
			2	46	113.182	X_{46}
			3	47	112.269	X_{47}
			4	48	112.976	X_{48}
2011	235.140	13	1	49	115.129	X_{49}
			2	50	118.742	X_{50}
			3	51	118.529	X_{51}
			4	52	121.365	X_{52}
2012	228.585	14	1	53	123.291	X_{53}
			2	54	119.979	X_{54}
			3	55	117.119	X_{55}
			4	56	121.825	X_{56}
2013	233.620	15	1	57	127.299	X_{57}
			2	58	125.487	X_{58}
			3	59	120.275	X_{59}
			4	60	125.247	X_{60}
2014	258.753	16	1	61	128.469	X_{61}
			2	62	128.040	X_{62}
			3	63	126.003	X_{63}
			4	64	126.438	X_{64}
2015	254.900	17	1	65	131.836	X_{65}
			2	66	131.331	X_{66}
			3	67	126.041	X_{67}
			4	68	130.138	X_{68}

Source: Benin National Statistics Office (INSAE) data and author's design

We thus obtain:

$$S_{17} = \{1, 2, 3, 4, 5, 6, \dots, 17\};$$

$$S_{68} = S_{4 \times 17} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, \dots, 68\};$$

$$X_{S_{68}} = \{X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, \dots, X_{68}\}.$$

The quarterly values of the X series are indexed by the elements of S_{68} and are obtained as quarterly account data to be manipulated: $X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, \dots, X_{68}$. Similarly, for series I of the quarterly indicator, we obtain as data to manipulate: $I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9, I_{10}, \dots, I_{68}$.

3.2 The explicit model

Consider the problem of quarterly disaggregation by the proportional method of Denton (1971) presented in the IMF handbook [4]. We consider a function $f(x) = ax$ where a is a constant value per period. Hence, let define $a_{i,y} = \frac{1}{I_{i,y}}$ for each value of $I_{i,y}$, $i = 1,2,3,4$ and $y = 1,2,3, \dots, 17$. So, we take $u_{i,y} = \frac{X_{i,y}}{I_{i,y}} = f(X_{i,y})$; $i = 1,2,3,4$ et $y = 1,2,3, \dots, 17$ and we obtain the following program (P1) :

$$(P1) \begin{cases} \underset{X}{Min} \left\{ F(X) = \sum_{y=1}^{17} \sum_{i=1}^4 (\Delta u_{i,y})^2 \right\} \\ Sc \sum_{i=1}^4 X_{i,y} = A_y; \forall y = 1,2, \dots, T \end{cases}$$

$$with \begin{cases} X = (X_{i,y})_{i=1,2,3,4 ; y=1,2,3,\dots,17} \\ u_{i,y} = \frac{X_{i,y}}{I_{i,y}} ; i = 1, 2, 3, 4 \text{ et } y = 1, 2, 3, \dots, 17 \\ u_{1,y} \text{ is the value for first the quarter of the current year } y; \\ u_{4,y-1} \text{ is the value of the fourth quarter forter of the previous year } y - 1; \\ \Delta u_{1y} = u_{1y} - u_{4,y-1} ; \forall y = 2, \dots, T; \\ \Delta u_{i,y} = u_{i,y} - u_{i-1,y} ; \forall i = 2,3,4 \end{cases}$$

y : index of year;

i : index of quarter;

A_y : value of the annual account given for the year y ;

$X_{i,y}$: he quarterly value sought for the account for the quarter i of the year y ;

$I_{i,y}$: the value of the quarterly indicator linked to the account for the quarter i of the year y .

Using the continuous indexation of quarters by the elements of S_{68} , we have $u_t = \frac{X_t}{I_t}$; $t = 1,2,3, \dots, 68$ and obtains the program (P2) which is that of Denton (1971) presented in [4].

$$(P2) \begin{cases} \underset{X}{Min} \sum_{t=2}^{68} \left(\frac{X_t}{I_t} - \frac{X_{t-1}}{I_{t-1}} \right)^2 \\ Sc \sum_{t=4y-3}^{4y} X_t = A_y; \forall y = 1,2, \dots, 17 \end{cases}$$

with

y : year of data observation; $y \in \{1,2, \dots, 17\}$;

t : time in quarter, $t = 1,2,3, \dots, 68$;

A_y : value of the annual account given for the year y ;

X_t : quarterly value sought for the account for the quarter t ;

I_t : value of the quarterly indicator linked to the account of the branch given for the quarter t .

3.3 Algorithms design for simulation

As pointed out in the proof of Lemma 2, the fundamental difference to be noted at the level of the programs (P1) and (P2) lies at the level of the objective functions. Both programs could be solved by any suitable method. It should be noted that the equality constraints in both programs form identical linear systems. Therefore, we limit ourselves here to the evaluation of the objective function involved in finding optimal solutions for each iteration in both cases.

For computational purposes, we have generated from the base information, the variables qbX_{min} and qbX_{max} , relative respectively to the minimum and maximum quarterly values of X by the formulas below:

$$qbX_{min(quarteriofyear)} = Xl = \frac{X_{yearly}}{4} - \frac{(tandarddeviationofX)}{17}$$

$$qbX_{max(quarteriofyear)} = Xu = \frac{X_{yearly}}{4} + \frac{(tandarddeviationofX)}{17}$$

The standard deviation of X is calculated on the series observed over the entire period 1999-2015. Thus, the solution generated for the quarterly series of X is obtained by the relation:

$$X = Xl + (Xu - Xl) \times 5.$$

The algorithms developed for the evaluation of the objective function associated respectively with the program (P1) and (P2) are coded with Matlab and contain the algorithm portions presented in the proof of Lemma 2. The Matlab codes are presented in Algorithm 1 and Algorithm 2 below.

All simulations are performed with Matlab R2008a installed on HP Intel Core i7 personal computer (vPro), 16 Gb of RAM, under the Windows system.

Algorithm 1 [for (P1)]

```

qbX_m=qbX_min(1:17,1:4);
qbX_M=qbX_max(1:17,1:4);
qI = qbl(1:17,1:4);
xl=[qbX_m]; % lower bound
xu=[qbX_M]; % upper bound
x = xl+(xu-xl).*5.0 ;
I = [qI];
u = x./I;
sum1=0;
for y=1:T
    for i=1:4
        if i==1
            if y==1
                Du(1,1)=0;
            else
                Du(y,1)=u(y,1)-u(y-1,4);
            end
        else
            Du(y,i)=u(y,i)-u(y,i-1);
        end
        sum1 = sum1+(Du(y,i))^2
    end
end
fl=sum1;
qX=[x(1:T,1:4)];
fprintf('%f\n', qX);
fprintf('Obj_value = ');
fprintf('%f\n', fl);

```

Algorithm 2 [for (P2)]

```

qbX_m=qbX_min(1:4*17);
qbX_M=qbX_max(1:4*17);
qI = qbl(1:4*17);
xl=[qbX_m]; % lower bound
xu=[qbX_M]; % upper bound
x = xl+(xu-xl).*5.0 ;
I = [qI]';
u = x./I;
sum2=0;
for t=2:4*T
    Du(t)=u(t)-u(t-1);
    Sum2= sum2+(Du(t))^2;
end
f2=sum2;
qX=[x(1:4*T)];
fprintf('%f\n', qX);
fprintf('Obj_value = ');
fprintf('%f\n', f2)

```

4 Results and Discussion

In order to have a good sensibility analysis, each algorithm has been run five times and the results presented in the tables and figures below are analysis in terms of key variables size and performance (execution time and proportion of memory used by Matlab).

4.1 Size of key variables

As the results in Table 2 show, the two algorithms obviously provide the same value for the objective function but different sizes for the key variables.

Table 2. Value of objective function and size of variables

Algorithm 1			Algorithm 2		
Obj_value =16.52272			Obj_value =16.52272		
Name	Size	Bytes	Name	Size	Bytes
I	17x4	544	I	68x1	544
u	17x4	544	u	68x1	544
X	17x4	544	X	68x1	544
Xl	17x4	544	Xl	68x1	544
Xu	17x4	544	Xu	68x1	544

Source: Authors' works

4.2 Performance analysis

As shown in part (a) of Fig. 1 below, overall, Algorithm 1 takes longer for full execution, compared to Algorithm 2. Indeed, from the results in Table 3, for the five (05) performed executions, Algorithm 1 averaged 19.50×10^{-3} seconds per run while Algorithm 2 averaged 17.92×10^{-3} seconds. Thus, from the point of view of execution time, Algorithm 2 is more efficient than Algorithm 1.

Table 3. Evaluation of algorithm performance

N ⁰	Execution time of the algorithm (Second)		Available memory (in MB)		Memory used by Matlab (in MB)		Cost of the algorithm (second x MB)	
	Algo 1	Algo 2	Algo 1	Algo 2	Algo 1	Algo 2	Algo 1	Algo 2
1	0.0232	0.0192	14661	14640	501	501	11.611	9.641
2	0.0197	0.0174	14659	14640	502	501	9.875	8.742
3	0.0188	0.0178	14658	14640	502	501	9.436	8.897
4	0.0178	0.0173	14658	14640	502	501	8.932	8.674
5	0.0181	0.0178	14658	14640	502	501	9.076	8.938
Mean	0.0195	0.0179	14658	14640	501.80	501.00	9.79	8.98

Note: Algo means Algorithm

Moreover, the part (b) of Fig. 1 illustrates that the proportion of memory used by Matlab for the complete execution of Algorithm 1 is relatively higher than that relative to Algorithm 2. Therefore, compared with memory used, Algorithm 2 is more efficient than Algorithm 1 in terms of time.

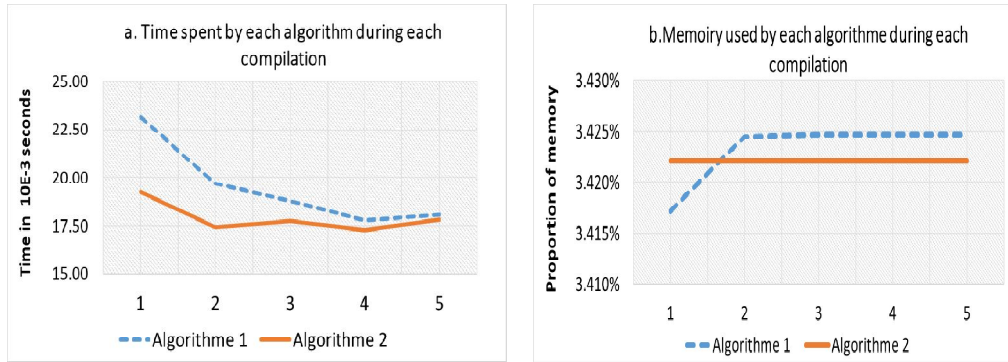


Fig. 1. Compared evolution of time put and memory used by algorithm at each execution

In total, according to Table 3, the cost of the algorithm 1 is higher than that of the algorithm 2. Indeed, on average, the cost (seconds x MB) estimated is 9.79 for the Algorithm 1 and 8.98 for Algorithm 2. We therefore deduce that algorithm 2 is preferable compared to Algorithm 1.

5 Conclusion and Perspective

In this paper, we have presented an indexing approach by building an isomorphism to ensure the performance of algorithms for solving macroeconomic quarterly disaggregation issues. In the light of the results, from the point of view of running time and memory used Algorithm 2 is more efficient than Algorithm 1. Thus, the transformation of the two-index into one-index for the variables by an isomorphism for solving quarterly disaggregation problem, allows it possible to make the simulation algorithm efficient.

It should be underlined that the devised isomorphism yield more efficient algorithms for solving programs with infra annual series, but also to transform a temporal variable into several static variables that are easily manipulated with the algorithms. Thus, a dynamic problem can be transformed into a static problem by the continuous indexing technique. We also believe that this technic can be useful for studying efficiently stochastic or multi-objective case of quarterly disaggregation of annual national accounts.

Competing Interests

Authors have declared that no competing interests exist.

References

- [1] Shiva R. Iyera, et al. Energy disaggregation analysis of a supermarket chain using a facility-model. *Energy and Buildings*, Elsevier. 2015;97:65-76. Available:<https://doi.org/10.1016/j.enbuild.2015.03.053>
- [2] Sangjin Han, et al. Network Support for Resource Disaggregation in Next-Generation Datacenters. Available:<https://people.csail.mit.edu/alizadeh/courses/6.888/papers/disagg.pdf>
- [3] Essessinou RA. Modélisation et optimisation du problème de calage dans la trimestrialisation d'agrégats macroéconomiques. Mémoire de Master II, Option Recherche Opérationnelle. Institut de Mathématiques et de Sciences Physiques (IMSP). Université d'Abomey-Calavi (UAC) – Bénin; 2014. French.

-
- [4] International Monetary Fund: Quarterly National Accounts Manual. 2017 Edition. Pre-Publication. 2018;86-126.
Available:<https://www.imf.org/external/pubs/ft/qna/>
- [5] Jorge Nocedal, Stephen J. Wright. Numerical optimization. Second Edition, Springer Series in Operation Research. 2000;388-389.
- [6] Guy E. Blelloch, Métriques de performance pour les algorithmes et programmes parallèles. French. Available:<http://www.lacim.uqam.ca/~chauve/Enseignement/INF7440/A04/COURS7-8/GUY-metriques.pdf>
- [7] Ji-Bo Wang, Jian-Jun Wang. Single-machine scheduling problems with precedence constraints and simple linear deterioration. Applied Mathematical Modelling. 2015;39:1172–1182. Elsevier.
Available:<http://dx.doi.org/10.1016/j.apm.2014.07.028>
- [8] Anthony Bonato, Shahin Kamali. Approximation algorithms for graph burning. International Conference on Theory and Applications of Models of Computation TAMC 2019: Theory and Applications of Models of Computation. Springer Nature Switzerland AG. 2019;74-92.
- [9] Jon Kleinberg. An approximation algorithm for the disjoint paths problem in even-degree planar graphs. Doctorate of Philosophy in Computer Sciences, Massachusetts Institute of Technologie; 1996.
Available:<https://scholar.google.com/scholar>
- [10] Jon Kleinberg. An approximation algorithm for the disjoint paths problem in even-degree planar graphs.
Available:<https://www.cs.cornell.edu/home/kleinber/focs05-even.pdf>
- [11] Gamst Mette, Jensen PN, Pisinger David, Plum Christian. Two and three-index formulations of the minimum cost multicommodity k-splittable flow problem. European Journal of Operational Research. 2010;202(1):82-89.
Available:<https://doi.org/10.1016/j.ejor.2009.05.014>
- [12] Anna Melchiori, Antonino Sgalambro. Optimizing emergency transportation through multicommodity quickest paths. 18th Euro Working Group on Transportation, EWGT. July 2015, Delft, The Netherlands. ELSEVIER. 2015;14-16.
Available:<https://www.sciencedirect.com>
- [13] Bernard Le Stum. Algèbre linéaire et bilinéaire, Version du 17 Février. 2017;15-18. French.
- [14] Saliou Toure. Algèbre, premier cycle MP1. Universités Francophones. UREF, EDICEF. 1991;197-199. French.
- [15] Ahuja Magnanti Orlin, Network Flow. Pearson New International Edition. 2014;59.

© 2019 Essessinou et al.; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:

The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)

<https://sdiarticle4.com/review-history/51632>