Small-change Computations of Price and Quantity Indices

Mark Horridge, Centre of Policy Studies, October 2003

Abstract

This document compares several methods of computing price and quantity indices, with special reference to the small-change computations used by GEMPACK. Accompanying worked examples, using Excel and GEMPACK, can be downloaded from the CoPS archive at:
www.monash.edu.au/policy/archivep.htm

JEL Codes: C43, E31, O47
Keywords: Price Index, Quantity Index, Divisia

Introduction

Computable general equilibrium (CGE) models implemented using GEMPACK often contain numerous equations defining price and quantity indices, of the following form:

$$p_{tot} = \sum_i S_i p_i$$
$$q_{tot} = \sum_i S_i q_i$$

Above, p_i and q_i are percent changes in individual prices and quantities, and p_{tot} and q_{tot} are percent changes in price and quantity indices. The equations are really differential equations; the changes are understood to be infinitesimal. The S_i are cost or budget shares which are continuously updated (using values of p_i and q_i) during a GEMPACK computation. The percent change in the total nominal expenditure, v_{tot}, is given by:

$$v_{tot} = p_{tot} + q_{tot}$$

The same indices are sometimes computed using flow values rather than shares, giving:

$$V_{tot}p_{tot} = \sum_i V_i p_i \qquad \text{where} \quad V_i \text{ is expenditure on good i}$$
$$V_{tot}q_{tot} = \sum_i V_i q_i \qquad \text{and} \quad V_{tot} = \sum_i V_i$$

or even, somewhat obscurely, as:

$$\sum_i V_i [p_i - p_{tot}] = 0$$
$$\sum_i V_i [q_i - q_{tot}] = 0$$

Although these indices are very convenient to compute in GEMPACK, we may wonder:

- Can we write levels (non-infinitesimal) formula for them?
- How do they relate to other commonly used price and quantity indices?
- How do we use GEMPACK to compute other price and quantity indices?

This document addresses these questions.

Indices used by statisticians

We start by reviewing some well-known indices used to summarize price and quantity changes between an initial time 0 and a final time 1. See Rossiter (2000) for more detail. Some notation:

P_i^0, Q_i^0 initial price and quantity of good i

P_i^1, Q_i^1 final price and quantity of good i

V00 $= \Sigma_i P_i^0 Q_i^0$ = cost at initial prices of initial quantities = initial expenditure

V11 $= \Sigma_i P_i^1 Q_i^1$ = cost at final prices of final quantities = final expenditure

V10 $= \Sigma_i P_i^1 Q_i^0$ = cost at final prices of initial quantities

V01 $= \Sigma_i P_i^0 Q_i^1$ = cost at initial prices of final quantities

VR = V11/V00 = *Value Ratio = (*cost of final bundle)/(cost of original bundle):

The Laspeyres price index is defined as the final/initial ratio of the cost of the **original** bundle, ie:

P^L = V10/V00 *Laspeyres price index*

The Paasche price index is defined as the final/initial ratio of the cost of the **final** bundle, ie:

P^P = V11/V01 *Paasche price index*

We can also write the Laspeyres price index as an initial-share-weighted average of price ratios:

$$P^L = V10/V00 = \Sigma_i P_i^1 Q_i^0 / \Sigma_i P_i^0 Q_i^0 = \Sigma_i [P_i^0 Q_i^0 / V00]\, P_i^1/P_i^0 \quad = \quad = \Sigma_i S_i^0\, [P_i^1/P_i^0]$$

where the S_i^0 are initial value shares. In terms of percentage changes this becomes:

$$p^L = \Sigma_i S_i^0 p_i$$

Note that the above equation is **not** infinitesimal: the p^L and p_i are finite (rather than tiny) percentage changes.

The Paasche index is **not quite** a final-share-weighted average of price changes; instead:

$$P^P = V11/V01 \quad \text{so} \quad P^P.V01 = V11 \quad \text{or} \quad P^P.\Sigma_i P_i^0 Q_i^1 = V11 \quad \text{or} \quad P^P.\Sigma_i P_i^1 Q_i^1\, [P_i^0/P_i^1] = V11$$

or $1/P^P = \Sigma_i S_i^1\, [P_i^0/P_i^1]$ where the S_i^1 are final value shares.

Analogous quantity indices are defined as:

Q^L = V01/V00 *Laspeyres quantity index (at original prices)*

Q^P = V11/V10 *Paasche quantity index (at new prices)*

Again the Laspeyres quantity index has a convenient finite percent change form:

$$q^L = \Sigma_i S_i^0 q_i$$

Table 1 shows a simple numerical example of the above calculations. In Table 1A, initial prices and quantities, and percent changes in both of these, are given—the rest is calculated. The bundle costs V00, V10, V01 and V11 are all we need to take from Table 1A. They are used to calculate the various indices in Table 1B.

Implicit deflators

We might expect that the product of Laspeyres price and quantity indices would equal the Value ratio (V11/V00). Sadly, this is not the case:

$$P^L Q^L = [V10/V00]x[V01/V00] \neq V11/V00$$

The Paasche index suffers the same defect—meaning that is difficult to decompose value changes into price and quantity components. To get around the problem, statisticians construct *implicit deflators*. For example, given a value ratio V (=V11/V00) and a Laspeyres quantity index, we could define an implicit price deflator I by:

$I.Q^L = V$ or $I = V/Q^L$ *Implicit price deflator dual to Laspeyres quantity index*

Column (3) of Table 1B computes implicit indices in this way for each of the four primary indices. It turns out (and can be checked with easy algebra) that the implicit indices dual to Laspeyres are Paasche, and vice versa. In terms of the equation just above:

$$I = V/Q^L = P^P$$

This means that if we know the Laspeyres indices and the value ratio, the Paasche indices are very easy to compute.

Less biased indices

Laspeyres and Paasche price indices diverge because budget shares change over time. However, price indices such as the CPI are politically significant (wages, pensions, or monetary policy may be driven by CPI figures). Hence there is strong pressure to compute a "true" index, which minimizes the effect of changing budget shares. A large literature exists, focused on the search for an optimal index. Such a true index, it is agreed, must lie between the Laspeyres and Paasche indices. Statistical bureaus of the richer countries are now adopting the *Fisher Ideal indices*, defined as:

$P^F = \sqrt{P^L.P^P}$ *Fisher Ideal price index*

$Q^F = \sqrt{Q^L.Q^P}$ *Fisher Ideal quantity index*

These also are shown in Table 1B. The Fisher indices have many properties beloved by theorists. For example, they are implicitly self-dual: their product is equal to the value ratio. Unfortunately, there is one property that Fisher lacks—sometimes called additivity. Laspeyres indices made it possible to construct tables, which compared main economic aggregates from different years in constant prices. Conveniently, we found:

$$GDP_R = C_R + I_R + G_R + X_R - M_R$$

That is, by consistently deflating nominal aggregates, one obtained real aggregates that satisfied the accounting identities obeyed by nominal aggregates. Using the Fisher indices we now find:

$$GDP_R = C_R + I_R + G_R + X_R - M_R + \text{Fisher residual.}$$

Chain-linked and Divisia indices

We could compute a series of 12 monthly Laspeyres price indices, using weights from the start of each month. By multiplying the indices together, we could get an index of price change for the whole year. We could do the same with 12 monthly Paasche indices. We would find that the difference between Laspeyres and Paasche indices, computed using weights from each month, was less than the difference between the two indices computed using merely year-start and year-end data. Indeed, if the full time-path of all prices and quantities was known, we could rebase our weights at arbitrarily short intervals and the Laspeyres and Paasche methods would yield the same number. This price index with continuously varying weights is called the *Divisia index*, with percent-change formula:

$$p^D = \int_0^1 \left(\Sigma_i S_i p_i \right).dt \qquad \textit{Divisia price index}$$

Above, the p_i are percent rates of change of prices and the S_i are continuously changing expenditure shares. The LHS, p^D, is the (finite) percentage change in the price index over the period 0 to 1: it depends not only on the initial and final values of RHS variables, but on the paths they follow through the period.

The Divisia indices for price and quantities have many desirable properties, including the "additivity" that the Fisher index lacks. The only problem is the data requirement—which will rarely be satisfied.

Many other index formulae have been proposed which use only year-start and year-end data. In assessing such formulae, a major criterion is often: how closely do results from the candidate formula approximate the Divisia index[1]? Divisia indices, indeed, are the unattainable ideal towards which practical indices strive.

[1] We need to make some assumptions about the paths of prices and quantities to do the comparison.

We conclude this section by observing that the example calculation in Table 1 uses dispersed price and quantity changes which imply quite large shifts in budget shares. In a real-world case (with smaller share changes), the different methods of calculating indices would yield more similar results.

Price and quantity indices in typical GEMPACK CGE models

We now return to the small change index formulae seen in GEMPACK CGE models, which GEMPACK integrates numerically to compute model solutions:

$$p_{tot} = \sum_i S_i p_i$$

$$q_{tot} = \sum_i S_i q_i$$

These are actually just Divisia indices ! Luckily, the path-following small-change algorithm used by GEMPACK gives a way to evaluate prices, quantities and shares at arbitrarily small intervals, making the Divisia indices a practical proposition rather than a theoretical dream. Why should we bother with indices such as Laspeyres or Fisher which are harder to compute and have fewer desirable properties? There might be two main reasons:

- We may need to report results for CPI or poverty indices which are officially defined (perhaps as Laspeyres indices) and which are important for policy purposes.
- We worry (or our critics rail) about *path dependence*.

Path Dependence

Most CGE models can be represented in the form:

(A) $Y = F(X)$

where Y is a vector of endogenous variables and X is a vector of exogenous variables. That is, the values of Y depend on the values of X. New values of X imply new values of Y, which do not depend on any particular assumptions about the path followed by variables between initial and final solutions[2].

If we start from a set of variable values satisfying (A), and implement small change equations obtained by differentiating the levels system (A), the Gragg and Euler methods used by GEMPACK will converge to correct solutions.

But if we implement small change equations *that have no underlying levels form such as (A)*, then our results will depend on the path followed by variables between initial and final values. The integral formula for a Divisia index is path-dependent in just this way. That might not matter if the time-paths of prices and quantities were those which actually occurred in history. But it does potentially matter if the paths of variable values are purely an artifact of the solution algorithm[3]. Is this a serious drawback of the Divisia indices?

Homothetic Aggregators

In a CGE model formulated in percent or log changes, most equations resembling:

$$p_{tot} = \sum_i S_i p_i \qquad q_{tot} = \sum_i S_i q_i$$

are derived from assumptions of optimization subject to constant-returns to scale production functions—such as CES or nested CES. In this case we have particular rules (demand equations)

[2] This description also applies to "dynamic" models formulated in discrete time if you imagine that the X and Y vectors include variable values for many periods.

[3] For a dynamic CGE model one might argue that the path followed by the GEMPACK solution algorithm actually is the adjustment path of the economy. However, this metaphor can be misleading, unless the model equations are presented as a discrete-time approximation to a system of continuous differential equations—the context in which, outside of GEMPACK and CGE, discussions of the Gragg and Euler solution algorithms are usually couched.

relating the p_i and q_i in a special way, and the above share-weighted indices are **not** path-dependent.

Nevertheless there normally remains a group of indices which do not correspond to an underlying homothetic aggregator function. Examples might include export and investment price indices, and perhaps also the CPI. Results for these variables are potentially dependent on the path followed by the solution algorithm.

Dangling and pervasive indices

Most of the indices mentioned in the previous paragraph are constructed purely for reporting purposes and do not affect other model variables. These *dangling indices* just "hang off" the main model without affecting anything else. For example, while all model results rely on correct calculations of individual export prices and quantities, small errors in measures of the export price index may not matter much.

On the other hand, if a path-dependent index plays an important economic role in the model, all other model results will also be subject to the path-dependence problem. These *pervasive indices* pose a greater problem. For example, if we assume that wages are linked to the CPI, our method of computing the CPI has economy-wide consequences.

The numerical effects of path dependence are very small

So small, in fact, that they are observed only in some special circumstances, described next.

Path dependence hinders exact numerical replication of closure swapping simulations

If our CGE model has a levels form:

(A) $Y = F(X)$

and we have a particular post-simulation set of variable values [X,Y] which satisfy (A), it does not matter if we re-partition variables between exogenous and endogenous groups (so long as all the variable *values* are the same). This allows us to use various artificial closures to extend the flexibility of the model. Naturally exogenous instruments can be allowed to endogenously adjust to meet naturally endogenous targets which are held exogenous for computing purposes[4]. Such devices should not affect our results.

However, the presence of pervasive path-dependent indices will cause results to be (ever-so-slightly) closure dependent. This is because, in a GEMPACK computation, exogenous variables follow a straight line path while endogenous variables will usually follow slightly curved paths. The actual path of each variable will depend on the closure.

Extreme accuracy in simulation results is desirable, not because we believe our results to approximate the real world to the 6th decimal place, but because even small errors and discrepancies often signal serious errors in model formulation and data. Exact numerical replication of results, regardless of the path used to reach those results, is an error-checking tool we are loath to lose.

Summary so far

While Divisia indices are easy to compute and have many desirable properties, their values are slightly path dependent. Indices, such as the Laspeyres or Fisher, that can be represented as levels formulae rather than as integrals, escape this problem. Since the Fisher index, for example, is likely to yield results very close to the Divisia, without being path-dependent, we might occasionally wish to use it instead. Besides, we may wish to report results using official index formulae, which could be based on Laspeyres, Fisher or other forms.

[4] The MONASH model "historical decomposition" is a striking example, requiring closure change for some hundreds of targets and instruments.

Table 1A: Worksheet for index calculations

	old price	old quantity	old value	old share	% change price	% change quantity	new price	new quantity	new price X old quantity	old price X new quantity	new price X new quantity
Good	P0	Q0	V0	S0	%p	%q	P1	Q1	P1.Q0	P0.Q1	P1.Q1
1	1	10	10	0.1	10	20	1.1	12	11	12	13.2
2	1	20	20	0.2	-10	10	0.9	22	18	22	19.8
3	1	30	30	0.3	10	0	1.1	30	33	30	33
4	1	40	40	0.4	-10	-10	0.9	36	36	36	32.4
Total			V00= 100	1					V10= 98	V01= 100	V11= 98.4

Table 1B: Worksheet for index calculations (continued)

		(1) ratio	(2) percent	(3) = 0.9840/(1)	
P Laspeyres = P^L	= V10 / V00	0.9800	-2.0000	1.0041	= Q Paasche
Q Laspeyres = Q^L	= V01 / V00	1.0000	0.0000	0.9840	= P Paasche
P Paasche = P^P	= V11 / V01	0.9840	-1.6000	1.0000	= Q Laspeyres
Q Paasche = Q^P	= V11 / V10	1.0041	0.4082	0.9800	= P Laspeyres
Value ratio	= V11 / V00	0.9840	-1.6000		
P Fisher = P^F	= $(P^L.P^P)^{0.5}$	0.9820	-1.8002	1.0020	= Q Fisher
Q Fisher = Q^F	= $(Q^L.Q^P)^{0.5}$	1.0020	0.2039	0.9820	= P Fisher
Fisher product	= $P^F.Q^F$	0.9840			

Table 2: Results from GEMPACK calculation PQINDEXP

Description	Name	% change
Laspeyres price index	plasp	-2.0000
Paasche price index	ppaas	-1.6000
Fisher price index	pfish	-1.8002
Divisia price index	pDivs	-1.7989
Initial-share-weights price index	p00	-2.4780
Value index	vIndx	-1.6000
Laspeyres quantity index	qlasp	0.0000
Paasche quantity index	qpaas	0.4082
Fisher quantity index	qfish	0.2039
Divisia quantity index	qDivs	0.2025
Initial-share-weights quantity index	q00	-0.4838

Calculating levels indices using GEMPACK

Table 3 shows the TABLO code for a GEMPACK calculation using the more recent "levels" syntax. As can be seen, it is a fairly literal implementation of the formulae presented above—and generates the same numerical results as the Excel worksheet on which Table 1 is based.

GEMPACK automatically translates the levels syntax into small change equations. Hence, there must be a way to write equations for Laspeyres, Paasche, and Fisher indices in the small-change form traditionally used in most larger GEMPACK models.

Table 4 shows the TABLO code for a GEMPACK calculation using small-change or "linear" syntax. The key point is the need for two new vectors of values V10 and V01:

```
Coefficient
 (all,c,COM)  V(c)   # Expenditure values, current prices and quantities #;
 (all,c,COM)  V10(c) # Expenditure values, final prices, original quantities #;
 (all,c,COM)  V01(c) # Expenditure values, original prices, final quantities #;
Update
 (all,c,COM)  V(c)   = p(c)*q(c);
 (all,c,COM)  V10(c) = p(c);
 (all,c,COM)  V01(c) = q(c);
```

Above, V is the vector of current expenditures, which is updated by both price and quantity changes. V10 and V01 have the same initial values as V, but are each updated by just one of price or quantity[5]. The V10 and V01 are used as weights for Laspeyres price and quantity indices. The Paasche indices are computed implicitly, ie, by dividing nominal expenditure change by Laspeyres indices. The Fisher indices are simply the averages of the (log changes in) Laspeyres and Paasche indices.

Results from this percent change calculation are shown in Table 2, and match the Excel results in Table 1B. Also computed are the Divisia indices (very close to the Fisher indices) and the indices derived from the small-change formulae:

$$p00 = \sum_i S_i^0 p_i$$
$$q00 = \sum_i S_i^0 q_i$$

where the S_i^0 are fixed initial value shares. The above formulae resemble the finite-percent-change formulae for the Laspeyres price and quantity indices mentioned above. However they in fact correspond to the levels formula

$$P00 = \prod_i P_i^{S_i^0} \qquad Q00 = \prod_i Q_i^{S_i^0}$$

and are therefore not path-dependent. They are the same as indices derived from the Cobb-Douglas utility function (though prices and quantities in this example do not follow Cobb-Douglas behaviour). Their product is not equal to the value ratio, and in this example the estimates they yield are outliers.

[5] An alternative approach would be to define and update coefficient vectors representing prices and quantities. Still, 2 new values would be needed for each expenditure item.

Table 3: A GEMPACK levels calculation

```
! File:PQINDEXL.TAB - example of levels GEMPACK index computation !

Set COM(C1-C4);

Variable (default=levels);
Equation (default=levels);

Variable
 (all,c,COM)  P(c)  # Prices #;
 (all,c,COM)  Q(c)  # Quantities #;
 (all,c,COM)  V(c)  # Values #;
   V11tot  # Cost at final prices of final quantities #;
   V01tot  # Cost at initial prices of final quantities #;
   V10tot  # Cost at final prices of initial quantities #;

Formula
(initial) V("C1") = 10;
(initial) V("C2") = 20;
(initial) V("C3") = 30;
(initial) V("C4") = 40;
(initial) (all,c,COM)  P(c)=1;
(initial) (all,c,COM)  Q(c)=V(c)/P(c);

Equation E_p_V (all,c,COM)  V(c)=P(c)*Q(c);

Coefficient
(parameter) (all,c,COM)  P0(c)  # Initial prices #;
(parameter) (all,c,COM)  Q0(c)  # Initial quantities #;
(parameter)  V00tot  # Cost at initial prices of initial quantities #;
Formula
(initial) (all,c,COM)  P0(c) = P(c);
(initial) (all,c,COM)  Q0(c) = Q(c);
(initial)              V00tot = sum{c,COM,P0(c)*Q0(c)};

Formula&Equation E_p_V11tot V11tot = sum{c,COM,P(c)*Q(c)};
Formula&Equation E_p_V10tot V10tot = sum{c,COM,P(c)*Q0(c)};
Formula&Equation E_p_V01tot V01tot = sum{c,COM,P0(c)*Q(c)};

Variable
 VIndx # Value index #;
 PLasp # Laspeyres price index #;
 PPaas # Paasche price index #;
 PFish # Fisher price index #;
 QLasp # Laspeyres quantity index #;
 QPaas # Paasche quantity index #;
 QFish # Fisher quantity index #;
Formula&Equation E_p_VIndx VIndx = V11tot/V00tot;
Formula&Equation E_p_PLasp PLasp = V10tot/V00tot;
Formula&Equation E_p_PPaas PPaas = V11tot/V01tot;
Formula&Equation E_p_PFish PFish = [PLasp*PPaas]^0.5;
Formula&Equation E_p_QLasp QLasp = V01tot/V00tot;
Formula&Equation E_p_QPaas QPaas = V11tot/V10tot;
Formula&Equation E_p_QFish QFish = [QLasp*QPaas]^0.5;

Variable
(linear) p_PDivs # Divisia price index #;
(linear) p_QDivs # Divisia quantity index #;
Equation
(linear) E_p_PDivs sum{c,COM,V(c)*[p_PDivs - p_P(c)]}=0;
(linear) E_p_QDivs sum{c,COM,V(c)*[p_QDivs - p_Q(c)]}=0;
```

Table 4: A GEMPACK percent change calculation

```
! File:PQINDEXP.TAB - example of percent change GEMPACK index computation !

Set COM(C1-C4);

Variable (default=linear);
Equation (default=linear);

Coefficient
 (all,c,COM)  V(c)   # Expenditure values, current prices and quantities #;
 (all,c,COM)  V10(c) # Expenditure values, final prices, original quantities #;
 (all,c,COM)  V01(c) # Expenditure values, original prices, final quantities #;

Formula
(initial) V("C1") = 10;
(initial) V("C2") = 20;
(initial) V("C3") = 30;
(initial) V("C4") = 40;
(initial) (all,c,COM) V10(c) = V(c);
(initial) (all,c,COM) V01(c) = V(c);

Variable
 (all,c,COM)  p(c) # Prices #;
 (all,c,COM)  q(c) # Quantities #;

Update
 (all,c,COM)  V(c)   = p(c)*q(c);
 (all,c,COM)  V10(c) = p(c);
 (all,c,COM)  V01(c) = q(c);

Variable
 pDivs # Divisia price index #;
 qDivs # Divisia quantity index #;
 vIndx # Value index #;
 pLasp # Laspeyres price index #;
 qLasp # Laspeyres quantity index #;
 pPaas # Paasche price index #;
 qPaas # Paasche quantity index #;
 pFish # Fisher price index #;
 qFish # Fisher quantity index #;

Equation
 E_pDivs sum{c,COM, V(c)*[p(c)-pDivs]} = 0;
 E_qDivs sum{c,COM, V(c)*[q(c)-qDivs]} = 0;
 E_vIndx             vIndx = pDivs + qDivs;
 E_pLasp sum{c,COM, V10(c)*[p(c)-pLasp]} = 0;
 E_qLasp sum{c,COM, V01(c)*[q(c)-qLasp]} = 0;
 E_pPaas pPaas = vIndx - qLasp;  ! Paasche indices derived implicitly !
 E_qPaas qPaas = vIndx - pLasp;
 E_pFish pFish = 0.5*(pLasp+pPaas);
 E_qFish qFish = 0.5*(qLasp+qPaas);

Variable
 p00 # Initial-share-weights price index #;
 q00 # Initial-share-weights quantity index #;
Coefficient (parameter) (all,c,COM) V00(c) # Initial expenditure values #;
Formula     (initial)   (all,c,COM) V00(c) = V(c);
Equation
 E_p00 sum{c,COM, V00(c)*[p(c)-p00]} = 0;
 E_q00 sum{c,COM, V00(c)*[q(c)-q00]} = 0;
```

Concluding recommendations

For most purposes the Divisia price and quantity indices used in typical models should be retained. They are easy to compute, and have many attractive properties. For example, the Divisia GDP price index turns out to be a share-weighted average of the price indices for absorption, exports and imports. However, the Divisia indices are path-dependent.

We may wish to replicate a particular price index formula used for official purposes. This document has shown how to do this for Laspeyres, Paasche and Fisher price indices.

Where an index is "pervasive", ie, affects agents' behaviour, the method of computing that index will affect values for most other model variables. So all model results would be potentially path-dependent. The numerical effects would normally be small, but might cause a problem if we wished to exactly replicate:

- results computed using a different closure.
- results computed using a different modelling system, eg, GAMS.

To avoid the problem we could replace each pervasive Divisia index with another index based on a levels formula (ie, not path-dependent). For example, if wages were indexed to the CPI, we might define two CPI variables:

- a Divisia CPI used for national accounting purposes; and
- a Fisher CPI used to drive wages.

As long as all Divisia indices in the model were "dangling", other model results would be independent of closure or solution method.

In GEMPACK, computation of levels index formulae usually requires us to define, for each component price/quantity, at least two new coefficients (for example, the V10 and V01 vectors used in the GEMPACK small-change calculation of Table 4). This would seem to be a significant computational overhead in those models which distinguish huge matrices of purchasers' prices. Does the whole database have to be maintained in triplicate[6]? Probably not. Usually the model will already calculate price and quantity indices for nearly all users, which are based on an aggregator function and so have a levels form. These (many fewer) indices could be used as the component prices or quantities used to drive levels indices like the Fisher. So we can compute non-path-dependent indices at modest cost.

References

Rossiter, R., (2000), “Fisher Ideal Indexes in the National Income and Product Accounts,” *Journal of Economic Education*, 31(4), 30-43.

Eichhorn et al. (ed) (1978), *Theory and Applications of Economic Indices*, Physica-Verlag, Wurzburg.

[6] GEMPACK calculations typically store values only for the current values of flows. By contrast, GAMS calculations typically store values for both initial and final values of both prices and quantities—4 times as much. So we see that GAMS modellers (a) calculate all index formulae (except Divisia) rather easily, and (b) tend to avoid large databases and price variables with many subscripts.