PERTURBED POLYNOMIAL PATH METHOD FOR ACCURATELY COMPUTING AND EMPIRICALLY EVALUATING THE COST OF LIVING*

Peter A. Zadrozny

Bureau of Labor Statistics
Division of Price Index Number Research
2 Massachusetts Ave., NE, Room 3105
Washington, DC 20212
e-mail: zadrozny_p@bls.gov

January 20, 2004

Additional key words: numerical solution, econometric evaluation
JEL codes: C32, C43, C53, C63

****
* Incomplete: do not duplicate, circulate, or quote without permission *
****

ABSTRACT

The paper describes and illustrates a perturbed polynomial path method, denoted P^3 and named after related perturbation and parametric path methods, for quickly and accurately computing changes in the real cost of living, Δc_t, as nominal prices change in observation periods t = 1, ..., T, and for computing associated residual quantities, ξ_t, i.e., differences between observed and optimal quantities of goods which maximize a given utility function on a budget line at given nominal prices and income. As prices change, for a given nominal income, a representative consumer moves from one utility-maximizing point to another in combined income and substitution effects. By definition, Δc_t is an adjustment of a consumer's real income, which restricts them to a pure substitution effect, i.e., to be on the same initial indifference curve as prices change. The P^3 method has at least four advantages: (1) The P^3 method is based on kth-order approximating polynomials, computed in h steps, for each observation period, with resulting accuracy of about h^{-k}. (2) The P^3 method can empirically evaluate a price index, however motivated, in terms of sizes of residual quantities of a corresponding utility-maximizing problem. E.g., a geometric-mean index can be evaluated in terms of sizes of residual quantities of a Cobb-Douglas utility function because both the index and the utility function are based on constant expenditure shares. (3) The P^3 method uses only derivatives of a utility function and can compute results even when there are no known explicit closed-form solutions, as illustrated with the tiered CES (TCES) utility function. (4) The P^3 method can account for estimated trend, cycle, and seasonality in residual quantities, by estimating univariate ARMA models of initial residuals and recomputing Δc_t and ξ_t for a modified utility function which incorporates the estimated trend, cycle, and seasonality. The paper illustrates these advantages with Cobb-Douglas and TCES utility functions.

*The paper represents the author's views and does not necessarily represent any official positions of the Bureau of Labor Statistics. Thanks to B. Chen, D. Johnson, and R. Verbrugge for comments and help.

1. Introduction.

The paper describes and illustrates a perturbed polynomial path method, denoted by P^3 and named after related perturbation and parametric path methods (Fleming, 1971; Judd, 1998, 2003; Chen and Zadrozny, 2003), for quickly and accurately computing changes in the real cost of living, Δc_t, as nominal prices change in observation periods t = 1, ..., T, and for computing associated residual quantities, ξ_t, i.e., differences between observed logs of quantities and optimal logs of quantities of goods, which maximize a given utility function on a budget line at given nominal prices and income. As prices change for a given nominal income, a representative consumer moves from one utility-maximizing point to another in combined income and substitution effects. By definition, Δc_t is an adjustment of a consumer's real income which restricts them to a pure substitution effect, i.e., to be on the same initial indifference curve as prices change. Given an initial cost of living index, usually normalized to one, Δc_t converts to level or index form as $c_t = 1 + \Delta c_t$ for t = 1, ..., T. The measure $T \cdot \ln| \sum_{t=1}^{T} \xi_t \xi_t^T / T |$ of residual-quantity sizes allows us to empirically evaluate a computed cost of living or any price index, however motivated, in terms of its corresponding Δc_t computed by the P^3 method. We denote the cost of living by COL and the COL computed by the P^3 method as $COLP^3$.

The P^3 method has at least the following four advantages:

1. Accurately Computing the COL. The paper describes and illustrates the P^3 method for computing a 4th-order polynomial (Taylor series) approximation of an exact Δc_t. Extension to computing a general kth-order polynomial approximation of Δc_t should be evident. The P^3 method divides a unit-length period t into h equal-length segments, computes the kth-order approximation of each segment, sums the h approximations, and, thereby, approximates the exact Δc_t, which is an integral over a path of optimal quantities consumed in period t. $COLP^3$ is numerically accurate to about h^{-k}. We illustrate $COLP^3$ with (k,h) = (4,10), which implies a computed COL with about 10^{-4} accuracy. Larger k and h imply greater accuracy, for example, (k,h) = (6,22) implies about 10^{-8} accuracy. Table 1 depicts minimal values of h for attaining about 10^{-4}, 10^{-8}, and 10^{-16} accuracy for k = 1, ..., 6. If a price index aims to measure COL, its accuracy can be evaluated by comparing it with $COLP^3$ based on the same or nearly the same utility maximizing problem. For example, the accuracy of the geometric-mean

price index can be evaluated by comparing it with COLP^3 based on maximizing a Cobb-Douglas utility function. Current economic discussions consider price indexes based at most on 2nd-order approximations of a utility function, called superlative price indexes (Diewert, 1976), and consider more rudimentary arguments about 1st- and 2nd-order approximation errors (Hausman, 2003).

2. Empirically Evaluating the COL. The empirical validity of a COLP^3 cannot be evaluated by comparing it with the true COL, because the true COL is unobserved. For the same reason, strictly, the empirical validity of a price index being considered cannot be evaluated by comparing it with a "standard" price index. However, a price index can be evaluated indirectly by the sizes of residual quantities of the corresponding COLP^3, based on a corresponding or the same utility function. For example, a geometric-mean price index corresponds to the COLP^3 of a Cobb-Douglas utility function, because both the index and the COLP^3 of a Cobb-Douglas utility function are based on expenditure shares being constant over time. Thus, the empirical validity of a geometric-mean price index can be evaluated by the sizes of residual quantities of the COLP^3 implied by a Cobb-Douglas utility function with the same expenditure shares.

The P^3 method also accurately computes optimal quantities of n purchased and consumed goods (and services), as predicted by maximizing utility at given prices. The optimal or predicted quantities are the log-form nonlinear regression function, $f(p_t)$, in the regression $\ln q_t = f(p_t) + \xi_t$, where $\ln q_t$, $\ln p_t$, and ξ_t denote n×1 vectors of logs of observed quantities of goods, logs of observed prices of goods, and residuals of logs of quantities. The utility function at particular parameter values -- the "model" -- and quantities implied by maximizing utility are jointly empirically validated if their implied residual quantities are sufficiently small. Similarly, a price index is empirically validated if residual quantities of the corresponding utility-maximizing problem are small. For normally distributed residual quantities, -2×log-likelihood function or $L = T \cdot \ln|\hat{\Sigma}|$ measures a model's fit of observed quantities, hence, L measures a model's and predicted quantities's empirical validity, where $\ln|\hat{\Sigma}|$ denotes the log determinant of $\hat{\Sigma} = \sum_{t=1}^{T} \xi_t \xi_t^T / T$, a positive definite (with probability one) estimate of the residual covariance matrix. Adding a penalty for the number of parameters, extends L to an information criterion. Thus, the empirically most valid model and associated predicted quantities should imply the lowest L or information criterion.

The method can be used to better understand differences among common price indexes. For example, recently the official unchained consumer price

index for urban consumers (CPIU) has been significantly above the unofficial chained CPIU (CCPIU). Although, the indexes are based on essentially the same formula, the CCPIU is updated more frequently. The P^3 method can be used as follows to better understand this difference, presumably due to some nonlinearity. First, suppose we have a utility function chosen either a priori, such as Cobb-Douglas with share parameters equal to observed expenditure shares, or by minimizing an *L* measure. Second, suppose we have computed $COLP^3$ using the chosen utility function and the CPIU and CCPIU indexes at the two update frequencies. On average over a sample, only the nonlinear terms should differ between the two indexes. Thus, we can better understand the average divergence between the CPIU and CCPIU indexes over a given time span by studying how the utility function's derivatives and the sample moments of quantities and prices of goods differently affect the nonlinear terms in the indexes. We plan to do this in the future.

3. General Differentiable Utility Functions. The P^3 method uses only derivatives of a utility function and can compute Δc_t for utility functions for which the standard consumer maximization problem has no known explicit closed-form solution. The P^3 method requires only (i) that the utility function is differentiable the desired number of times and (ii) that it satisfies second-order conditions (Mann, 1943) along the computational path. By illustrating results with a tiered constant elasticity of substitution (TCES) utility function, we show that the P^3 method applies to far more general, yet parametrically tractable (parsimonious), utility functions than are commonly used, such as Cobb-Douglas or CES (Arrow et al., 1961).

4. Trend, Cycle, and Seasonality. We can redefine the utility function and reapply the P^3 method to include predictable temporal variations, such as trend, cycle, and seasonality. Suppose we initially compute Δc_t and ξ_t, for t = 1, ..., T, using an invariant utility function. Given initial residual quantities, we can account for their trend, cycle, and seasonality by estimating a vector autoregressive moving-average (VARMA) model, $\xi_t = \hat{\xi}_t + \varepsilon_t$, for t = 1, ..., T, where $\hat{\xi}_t = \Phi(L)\xi_{t-1} + \Theta(L)\varepsilon_{t-1}$, $\Phi(L)$ and $\Theta(L)$ are lag-generating polynomials in positive powers of the lag operator L, and ε_t is an n×1 vector of white-noise disturbances, distributed NIID(0,Σ) (Kaiser and Maravall, 2001). With many goods, estimating a nondiagonal VARMA model, which allows all possible feedbacks among different variables, is infeasible. However, because the residual model needs only to predict one period ahead, even with many goods we can separately estimate a univariate ARMA model for

each residual. Then, we can treat $\hat{\xi}_t$, the predictable part of ξ_t, as part of the utility function and reapply the P^3 method. In the recomputation, we would treat $\hat{\xi}_t$ as exgoneous, like prices, and would fit an interpolating polynomial to its observed discrete-time differences. The initial computation reacts to trend, cycle, and seasonality in prices but the recomputation also reacts to these effects in quantities. Thus, the P^3 method can include effects of predictable temporal variations in prices and quantities, according to an estimated VARMA model of residual quantities. The approach is consistent with Hausman (2003).

The paper proceeds as follows. Section 2 describes the P^3 method in two subsections as a combination of substitution and income effects. Section 3 extends the discussion of section 2 from an invariant utility function to a time-varying utility function and discusses the statistical issues of empirical validation and parameter estimation. Section 4 illustrates the P^3 method with Cobb-Douglas and TCES utility functions and a sample of 10 aggregated goods from data at the Bureau of Labor Statistics. Cobb-Douglas utility functions provide reference cases of closed-form solutions, which can be used to check the accuracy of the P^3 method, and TCES utility functions provide new test cases for the P^3 method, with no known closed-form solutions. Section 5 contains concluding remarks and is followed by an appendix, figures, tables, and references.

2. P^3 Method Computations.

2.1. Preliminary Discussion.

Let $[1,T+1) = \bigcup_{t=1}^{T} [t, t+1)$ denote a continuous time interval divided into T unit-length intervals or periods indexed by their beginning moments, t = 1, ..., T, where $[t,t+1) = \{s | t \leq s < t+1\}$. The following definitions of variables hold both in continuous-time, within a period t, and in discrete time, at the beginning moments t = 1, ..., T. Subscript t denotes variables in discrete time at the beginning moments and argument s or (s) denotes variables in continuous time within periods. The computations apply separately to each period. We concentrate on what happens within period t, think of it as the current period, and think of period t+1 as the next period.

Let $p = (p_1, \ldots, p_n)^T$ and $q = (q_1, \ldots, q_n)^T$ denote n×1 vectors of nominal prices and real quantities of goods (and services) purchased and consumed by a

representative consumer. Then, let p_t denote the vector of nominal prices observed in period t by the consumer and the investigator and let q_t denote the vector of quantities observed in period t by these agents. Throughout, vectors are defined as columns, superscript T denotes transposition, and "quantities" are quantities of goods and services. Each period t, the representative consumer faces the budget line $p_t^T q = e_t$, in nominal terms, where q denotes any n×1 vector of nonnegative quantities and $e_t = p_t^T q_t$ denotes nominal expenditures. Following the standard utility-maximizing theory, we assume the consumer treats e_t as given nominal income and call it such. We say a vector of quantities is optimal if it maximizes a utility function on a budget line.

We consider the budget line as $p_t^T q = e_t c_t$, where c_t is the real COL. For each period t, we think of moving over the period from beginning to end. Let c_t and c_t' be real COL at the beginning and end of period t. Then, given the initial value $c_t = 1$, we want to compute the change in real COL over the period, $\Delta c_t = c_t' - c_t$, caused by price changes over the period. Also, we want to compute corresponding residual quantities, $\xi_t = \ln q_t - \ln \hat{q}_t$, where $\ln q_t$ is the vector of observed logs of quantities and $\ln \hat{q}_t$ is the vector of optimal logs of quantities at current prices and income, p_t and e_t. We convert current and next period's nominal prices to real prices by dividing them by current nominal income, e_t, as $\tilde{p}_t = p_t/e_t$ and $\tilde{p}_{t+1} = p_{t+1}/e_t$, and, correspondingly, convert budget lines from nominal to real terms as, for example, $p_t^T q = c_t e_t$ to $\tilde{p}_t^T q = c_t$.

Figure 1 depicts the P^3 method for two goods and is constructed based on an assumed utility function, current prices, p_t, next period's prices, p_{t+1}, and current quantities, q_t. Each point in figure 1 represents quantities of the two goods. Figure 1 has five utility maximizing points A to E with associated budget lines AA to EE. Point A denotes currently observed quantities, $q_t = (q_{1t}, q_{2t})^T$, and has one indifference curve and three budget lines through it. The indifference curve, denoted AB, passes through points A and B and is given by $u(q) = u(q_t)$, where $u(\cdot)$ is an assumed utility function. The flattest budget line through A, denoted CC, with utility maximizing point C, is given by $\tilde{p}_{t+1}^T q = p_{t+1}^T q_t / e_t$, where $p_{t+1}^T q_t / e_t$ is the Laspeyres price index over period t. The medium-sloped currently-observed budget line through A, denoted EE, with utility maximizing point E, is given by $\tilde{p}_t^T q = 1$. Point A maximizes utility on EE only by the remotest chance. The steepest budget line through A, denoted AA,

which maximizes utility at A is given by $\tilde{p}_t'^{T}q = 1$, where $\tilde{p}_t' = \nabla u(q_t)^T/\nabla u(q_t)q_t$ according to first-order conditions (2.2) and (2.3) Note that $\nabla u(q_t)q_t$ is scalar because $\nabla u(q_t)$ is 1×n. For example, the Cobb-Douglas utility function implies $\tilde{p}_t'$ consists of expenditure shares. Point B on indifference curve AB maximizes utility on budget line BB given by $\tilde{p}_{t+1}^{T}q = \varphi_t(p_{t+1}^{T}q_t/e_t)$, where φ_t is a positive fraction to be computed. Point D maximizes utility on budget line DD at next period's prices and current income, given by $\tilde{p}_{t+1}^{T}q = 1$. Budget lines AA and EE are defined solely by period t information, while budget lines BB to DD are defined by both period t and t+1 information and have slopes given by p_{t+1}.

First, we consider the increase in real income or equivalent decrease in real COL, when the consumer moves from A to D as a result of the hypothetical change in nominal prices, $\Delta p_t' = p_{t+1} - p_t' \neq 0$, for constant nominal income e_t, where p_t' is constructed to make A optimal on AA. We start at point A, which is an optimal consumption point on budget line AA. As the relative price of good one increases from p_t' at AA to p_{t+1} at BB, the optimal consumption point moves from A to B in a pure substitution effect along indifference curve AB. This move causes real income to decrease by $(\varphi_t - 1)(p_{t+1}^{T}q_t/e_t)$ at next period's prices, p_{t+1}.

Although, in general, $0 \leq \varphi_t \leq 1$ and $0 < p_{t+1}^{T}q_t/e_t$, figure 1 indicates $0 < \varphi_t < 1$ and $0 < p_{t+1}^{T}q_t/e_t < 1$. Concavity to the origin of indifference curves, i.e., second-order condition (2.4), causes $0 \leq \varphi_t \leq 1$. A nearly straight-line indifference curve implies a large substitution effect, hence, $\varphi_t \cong 0$ and a maximal decrease in real income when moving from A to B; a very curved indifference curve implies a small substitution effect, hence, $\varphi_t \cong 1$ and a minimal decrease in real income when moving from A to B.

However, for constant nominal income, e_t, nominal price changes also cause the nominal price level, hence, real income to change. Figure 1 indicates that real income increases by $1 - p_{t+1}^{T}q_t/e_t > 0$ when moving from C to D. Thus, for constant nominal income, e_t, price changes $\Delta p_t' = p_{t+1} - p_t'$ cause a net change in real COL of

$$\Delta c_t = \varphi_t(p_{t+1}^{T}q_t/e_t) - 1, \tag{2.1}$$

when moving from A to D, or an equivalent net change in real income of $-\Delta c_t = 1 - \varphi_t(p_{t+1}^T q_t / e_t)$. Although, in general, the net Δc_t could be positive, zero, or negative, figure 1 indicates that net $\Delta c_t = \varphi_t(p_{t+1}^T q_t / e_t) - 1 < 0$, when moving from A to D. Section 2.2 explains the P^3 method in general for computing the initial $\Delta c_t = (\varphi_t - 1)(p_{t+1}^T q_t / e_t)$ or, equivalently, computing φ_t of the pure substitution move from A to B, such that the remaining $\Delta c_t = (p_{t+1}^T q_t / e_t) - 1$ of the pure income move from B to D requires no additional P^3-style computations.

We consider the change in real COL starting from A on budget line AA, caused by the hypothetical price changes $\Delta p_t' = p_{t+1} - p_t' \neq 0$. The same change in real COL and real income results if we start from A as a nonoptimal consumption point on the currently observed budget line EE or from any other budget line through A. This occurs because, regardless of the starting budget line, we start from the same point A and the computed φ_t depends only on the indifference curve AB, which is uniquely defined by A. We start from budget line AA in order to move from A to D along a continuously optimal path as prices change, because the P^3 method is guaranteed to work only when second-order condition (2.5) holds continuously. Thus, the computed Δc_t would remain $\varphi_t(p_{t+1}^T q_t / e_t) - 1$ even if it were computed as caused by the observed price changes, $\Delta p_t = p_{t+1} - p_t$.

If we only want Δc_t, then, we only need to complete the computations discussed in section 2.2. Otherwise, if we also want to evaluate the computed COL's empirical validity, we need to compute residual quantities. In terms of figure 1, the vector of residual quantities is the vector difference A - E on budget line EE. In section 2.3, we discuss computing the residual quantities in a combined pure-substitution and real-income move caused by the price changes $\Delta p_t' = p_t - p_t'$. The move is equivalent to moving in figure 1 from A as optimal on AA to E as optimal on EE. The computation of the residual quantities is a simple modification of the computation of Δc_t discussed in section 2.2.

2.2. Computing the Cost of Living.

The appendix explains definitions and rules of matrix differentiation and should be skimmed or read before continuing. In this section and the next one, we work with derivatives of vector functions of vector arguments, which includes vectorized matrix functions of vectorized matrix arguments, in differential and gradient forms. Differentials are denoted by d and gradients

by ∇. Differentials are defined as usual, but gradients are defined more generally and include the usual definitions of gradients, Jacobians, and Hessians as special cases. All equations are derived in differential form and as few as possible are further converted to gradients. Derived differentials are strictly infinitesimally small, hence, uncomputable and gradients are finite valued, hence, automatically computable. However, we can and do interpret and compute the differentials as finite directional derivatives. We could convert all differentials to gradients and compute all equations in gradient form. Chen and Zadrozny (2003) follow this approach, but especially here, this would result in unnecessary complicated derivations and unnecessary computations with large and sparse Kronecker products and related permutation and vectorization operations (Magnus and Neudecker, 1988). To minimize these complications, we propose computing almost all differentials as directional derivatives, which amounts to computing in the forward mode of automatic or algorithmic differentiation, currently the preferred method for quickly and accurately computing derivatives (Griewank, 2000).

Because, henceforth, all prices are real as defined in section 2.1, for simplicity we drop the tilde which denotes "real" in prices and denote real prices by p_t. We discuss computing the change in the real cost of living, Δc_t, and residual quantities, ξ_t, for each observation period $t = 1, \ldots, T$. However, because the computations are self contained within each period, we proceed with the understanding that "period t" is any of the periods $t = 1, \ldots, T$ and mostly do not repeat this qualification. We also refer to period t as the "current" period and period t+1 as the "next" period.

In the standard problem, the representative consumer maximizes utility, $u(q_t)$, with respect to quantities, q_t, on the budget line, $p_t^T q_t = c_t$, for given prices, p_t, and initial real cost of living, $c_t = 1$. The Lagrangian function of the maximization problem is $u(q_t) + \lambda_t(c_t - p_t^T q_t)$, where λ_t is the budget line's scalar Lagrange multiplier. Totally differentiating the Lagrangian function, setting the result to zero, and imposing $dp_t = 0$ and $dc_t = 0$ because p_t and c_t are given, implies $(\nabla u(q_t) - \lambda_t p_t^T)dq_t + d\lambda_t(c_t - p_t^T q_t) = 0$, where d denotes the differential of a function or variable. Because dq_t and $d\lambda_t$ can assume any nonzero values, the last equation implies the usual first-order conditions,

$$\nabla u(q_t) - \lambda_t p_t^T = 0, \tag{2.2}$$

(2.3) $\quad p_t^T q_t - c_t = 0,$

where $\nabla u(q_t) = [\partial u(q_t)/\partial q_{1t}, \ldots, \partial u(q_t)/\partial q_{nt}]$ is the 1×n gradient row vector of first-partial derivatives of $u(q_t)$. Equations (2.1) and (2.2) can be solved for unique values of q_t and λ_t, at least locally and numerically, for given values of p_t and c_t, if second-order condition (2.5) holds.

So far, we have discussed the standard utility maximizing problem in which real income, c_t, is given. Now, we discuss the problem in which q_t and c_t change in response to price changes, $\Delta p_t' = p_{t+1} - p_t' \neq 0$, such that c_t varies to keep the consumer on the same indifference curve. In section 2.3, we discuss the related problem in which c_t is dropped as a variable because it is identically equal to one. Combining the differentials of (2.2) and (2.3) with $du(q_t) = \nabla u(q_t) dq_t = 0$, which keeps the consumer on the same indifference curve, implies

$$(2.4) \qquad F(x)dy = G(y)dp \quad \text{or} \quad \begin{bmatrix} \nabla^2 u(q) & -p & 0_{1\times 1} \\ -p^T & 0_{1\times 1} & 1 \\ \nabla u(q) & 0_{1\times 1} & 0_{1\times 1} \end{bmatrix} \begin{bmatrix} dq \\ d\lambda \\ dc \end{bmatrix} = \begin{bmatrix} \lambda I_n \\ q^T \\ 0_{1\times n} \end{bmatrix} dp,$$

where $\nabla^2 u(q)$ is the n×n Hessian matrix of second-partial derivatives of $u(q)$ and $F(x)$ is an $(n+2)\times(n+2)$ matrix function of $x = (y^T, p^T)^T$, the $(2n+2)\times 1$ vector of all variables, and $G(y)$ is an $(n+2)\times n$ matrix function of $y = (q^T, \lambda, c)^T$, the $(n+2)\times 1$ vector of variables to be determined. Following statistical section 3, in the problem considered here, we say p is exogenous because prices are the given variables and y is endogenous because it is determined by the problem's solution. If the utility function is differentiable k times, then, $F(x)$ is differentiable k-2 times. $G(y)$ is always differentiable any number of times and has constant first-partial derivatives and zero higher-order partial derivatives. Henceforth, for simplicity, we frequently suppress time subscript t and quantity argument q in the utility function and its derivatives, keeping in mind that x (hence, y) are evaluated at time t.

The variables in x are either observed or can be computed. Given the utility function, prices, and quantities, the first-order conditions (2.2) and (2.3) imply $c = p^T q$ and $\lambda = \nabla u q/c$, so all variables in x are known. Given x,

equation (2.4) is solved uniquely as $dy = H(x)dp$, where $H(x) = F(x)^{-1}G(y)$, if and only if $|F(x)| \neq 0$, where $|\cdot|$ denotes the determinant of a square matrix. Laplace expansion of a determinant and equation (2.4) imply $|F(x)| = \lambda|B(x)|$, where $B(x) = \begin{bmatrix} \nabla^2 u & -p \\ -p^T & 0_{1\times 1} \end{bmatrix}$ is the bordered Hessian of the standard utility maximizing problem with first-order conditions (2.2) and (2.3). The second-order condition of the standard problem is $|B(x)| < 0$ (Mann, 1943). Given $\lambda > 0$, the second-order condition of the present constant-utility problem is

$$|F(x)| = \lambda|B(x)| < 0. \tag{2.5}$$

We say dropped old goods or as yet unpurchased new goods are inactive, because their quantities are zero (or numerically nearly so) and they do not figure in the first-order conditions, but currently purchased and consumed goods are active. Positive prices and quantities of active goods imply positive expenditures, finite and positive marginal utilities, and, hence, a finite and positive Lagrange multiplier, $\lambda = \nabla u^T q/c > 0$. Therefore, at a utility maximizing x, which satisfies second-order condition (2.5), equation (2.4) has the unique solution

$$dy = H(x)dp, \tag{2.6}$$

where $H(x) = F(x)^{-1}G(y)$ is an $(n+2)\times n$ matrix function of x. Dividing dy and dp by ds converts them to the time derivatives dy/ds and dp/ds. Indeed, henceforth, we consider differentials of variables to be time derivatives, but, for simplicity, suppress their ds divisors.

For $s \in [t,t+1)$, we want to compute $\Delta y_t = y_{t+1} - y_t = \int_{s=t}^{t+1} dy(s)$. Let $y(s)$ and $\hat{y}(s)$ denote exact and approximate solution paths of y, for a given price path $p(s)$, for $s \in [t,t+1)$. Some exceptional utility functions, such as CES, have known closed-form solution paths, but for most utility functions which satisfy the second-order conditions, we can only compute an approximate solution path, $\hat{y}(s)$. The implicit function theorem, upon which the P^3 method is based, implies that if the utility function is differentiable $k+3$ times and satisfies the second-order conditions, so that its maxima are interior points, then, the exact solution path is differentiable $k+1$ times, integrates as

$\int_{s=t}^{t+1} dy(s)$, and, for $s \in [t,t+1)$, has the kth-order polynomial (Taylor series) approximation

$$(2.7) \qquad \hat{y}(s) = y_t + \nabla y_t(s-t) + (1/2!)\nabla^2 y_t(s-t)^2 + \ldots + (1/k!)\nabla^k y_t(s-t)^k,$$

where y_t, ..., $\nabla^k y_t$ are (n+2)×1 coefficients to be determined in terms of observed prices and quantities. We treat the true y and p processes and their polynomial interpolates as differentiable continuous-time processes in $s \in [t,t+1)$ and as discrete-time processes in t = 1, ..., T. Ford (1955) discusses connections between continuous and discrete processes and their polynomial interpolates.

Following equation (2.7), we could approximate Δy_t as $\Delta\hat{y}_t = \int_{s=t}^{t+1} d\hat{y}(s) = \nabla y_t + (1/2!)\nabla^2 y_t + \ldots + (1/k!)\nabla^k y_t$, which has the accuracy $\varepsilon = |\Delta y_t - \Delta\hat{y}_t| = (1/(k+1)!)|\nabla^{k+1} y(\theta)|$, for $\theta \in (t,t+1)$ (Apostol, 1974, pp. 241-242). We are interested in the accuracy's order of magnitude in k. With sufficient differentiability of F(x), we can increase k to achieve the desired solution accuracy, although this might lead to a large k with resulting difficulties in deriving, storing, and computing the necessary derivatives. For example, Judd (1998, figure 13.1, p. 483) reports needing $k \geq 10$ to achieve 10^{-6} solution accuracy within about a 40% deviation from a center-point solution with 10^{-14} accuracy. When applying the "pure" perturbation method, we can increase only k to improve accuracy. By adding "polynomial path" to "perturbation," we can also increase h to achieve desired accuracy. Table 1 shows that semi-single-, single-, and double-precision orders of magnitude of accuracy can be achieved with relatively low combinations of k and h. We introduce h by dividing the unit-length time interval of each period t into h subintervals of equal length 1/h, apply a kth-order polynomial approximation to each subinterval, and sum the h approximations.

For each period t and a chosen positive integer h, we partition the period into h subperiods of equal length 1/h as $[t,t+1) = \bigcup_{i=1}^{h} [t_i,t_i+(1/h))$, where $[t_i,t_i+(1/h)) = [t+(i-1)/h,t+(i/h))$ for i = 1, ..., h. For each subperiod $t_i = t_1$, ..., t_h in period t, we compute the y coefficients, ∇y_{t_i}, ..., $\nabla^k y_{t_i}$, recursively and approximate Δy_t as

$$(2.8) \qquad \Delta\hat{y}_t = \sum_{i=1}^{h} \Delta\hat{y}_{t_i} ,$$

$$\Delta\hat{y}_{t_i} = \int_{s=t+(i-1)/h}^{t+(i/h)} d\hat{y}(s) = \nabla y_{t_i} h^{-1} + (1/2!)\nabla^2 y_{t_i} h^{-2} + \ldots + (1/k!)\nabla^k y_{t_i} h^{-k}.$$

Because each term, $\Delta\hat{y}_{t_i}$, has accuracy on the order of $h^{-(k+1)}$, which is denoted by $\varepsilon = O(h^{-(k+1)})$, their sum, $\Delta\hat{y}_t = \sum_{i=1}^{h} \Delta\hat{y}_{t_i}$, has accuracy on the order of h^{-k} or $\varepsilon = O(h^{-k})$. The idea of summing h integrals of low-order polynomials over subintervals of length (b-a)/h instead of integrating a high-order polynomial over the whole interval [a,b] is basic in numerical integration (Judd, 1998, ch. 7, pp. 251-283).

Solution accuracy has the factor $\gamma = 1/(k+1)!)|\nabla^{k+1} y(\theta)|$. Instead of deriving bounds on γ in terms of derivatives of H(x), we can estimate it as follows. Let $\rho(\hat{x}) = ||r(\hat{x})||/||\hat{x}||$ denote the relative accuracy of a computed solution, $\hat{x}$, where $||\cdot||$ denotes a vector norm (Golub and Van Loan, 1996, pp. 52-54), r(x) denotes the column vector of equations (2.2), (2.3), and $u(q) - u_0 = 0$, such that u_0 denotes initial utility. Strictly, $\Delta\hat{x}$ is computed, so that $\hat{x} = \Delta\hat{x} + x_0$, where x_0 is an assumed or given initial value of x. Accuracy can also be expressed as $\varepsilon \leq \gamma h^{-k}$, so that we can estimate γ as the average value of $||r(\hat{x})||h^k$ for various values of h and k. By setting h and k sufficiently large, we can usually obtain an acceptably small $\rho(\hat{x})$ for the computed $\hat{x}$, although we cannot set h and k so large that h^{-k} cannot be stored as a nonzero number on the computer being used. Table 1 depicts minimal values of h for attaining accuracy of orders $h^{-k} = 10^{-4}$, 10^{-8}, and 10^{-16}, for k = 1, ..., 6. Additional numerical errors arise from computing the y coefficients but should be acceptably small or, in the case of inverting F(x), can be made acceptably small by making indifference curves sufficiently curved, i.e., by ensuring that |F(x)| of second-order condition (2.5) is sufficiently below zero.

We now explain the P^3 method for computing the y coefficients, ∇y_{t_i}, ..., $\nabla^k y_{t_i}$, when k = 4. Extension from the 4th-order approximation to the general kth-order approximation should be conceptually straightforward, although the details are tedious. Having stated the approximate y process (2.7), we now derive the computational equations of the P^3 method. We differentiate approximate y process (2.7). We differentiate differential (2.6) of the unknown true y process three more times, to obtain differentials (2.10) of y in terms

of the differentials of H(x) and prices. We derive differentials of H(x) in terms of differentials (2.11) of F(x) and G(y). We state price process (2.15), analogous to approximate y process (2.7), and express its coefficients (2.17) in terms of observed forward-differenced prices (2.18). Finally, we discuss combining these results and computing the coefficients of the approximate y process (2.7), hence, computing Δy_t according to equation (2.8).

For $k = 4$, $s \in [t_i, t_i+h)$, and $t_i = t_1, \ldots, t_h$, differentiating approximate y process (2.7) four times with respect to s implies

$$(2.9) \qquad d\hat{y}(s) = \nabla y_{t_i} + \nabla^2 y_{t_i}(s-t_i) + (1/2)\nabla^3 y_{t_i}(s-t_i)^2 + (1/6)\nabla^4 y_{t_i}(s-t_i)^3,$$

$$d^2\hat{y}(s) = \nabla^2 y_{t_i} + \nabla^3 y_{t_i}(s-t_i) + (1/2)\nabla^4 y_{t_i}(s-t_i)^2,$$

$$d^3\hat{y}(s) = \nabla^3 y_{t_i} + \nabla^4 y_{t_i}(s-t_i),$$

$$d^4\hat{y}(s) = \nabla^4 y_{t_i}.$$

For each $s = t_i = t_1, \ldots, t_h$, we compute the y coefficients, $\nabla y_{t_i}, \ldots, \nabla^4 y_{t_i}$, so that they are equal to the 1st to 4th differentials of true y process (2.6).

Using the product rule of differentiation (6.17) to differentiate equation (2.6) three times, implies

$$(2.10) \qquad d^2y(s) = dH(s)dp(s) + H(s)d^2p(s),$$

$$d^3y(s) = d^2H(s)dp(s) + 2dH(s)d^2p(s) + H(s)d^3p(s),$$

$$d^4y(s) = d^3H(s)dp(s) + 3d^2H(s)d^2p(s) + 3dH(s)d^3p(s) + H(s)d^4p(s),$$

where $H(s) \equiv F(x(s))^{-1}G(y(s))$.

Repeatedly applying the product rule of differentiation to $F(s)H(s) = G(s)$, implies $dF(s)H(s) + F(s)dH(s) = dG(s)$, $d^2F(s)H(s) + 2dF(s)dH(s) + F(s)d^2H(s) = d^2G(s)$, and $d^3F(s)H(s) + 3d^2F(s)dH(s) + 3dF(s)d^2H(s) + F(s)d^3H(s) = d^3G(s)$, where $d^kF(s) \equiv d^kF(x(s))$ and $d^kG(s) \equiv d^kG(y(s))$, for $k = 0, \ldots, 3$. Then, solving for $dH(s)$, $d^2H(s)$, and $d^3H(s)$, implies

$$(2.11) \qquad dH(s) = F(s)^{-1}[dG(s) - dF(s)H(s)],$$

$$d^2H(s) = F(s)^{-1}[d^2G(s) - d^2F(s)H(s) - 2dF(s)dH(s)],$$

$$d^3H(s) = F(s)^{-1}[d^3G(s) - d^3F(s)H(s) + 3d^2F(s)dH(s) + 3dF(s)d^2H(s)].$$

Equations (2.11) are fully recursive. Given utility function $u(\cdot)$, we compute $F(s)$, $G(s)$, and $H(s)$. Then, we compute $dF(s)$, $dG(s)$, and $dH(s)$. Then, we compute $d^2F(s)$, $d^2G(s)$, and $d^2H(s)$. Finally, we compute $d^3F(s)$, $d^3G(s)$, and $d^3H(s)$.

For $k = 1$, 2, and 3, $d^kF(s)$ and $d^kG(s)$ are

$$(2.12) \qquad d^kF(x) = \begin{bmatrix} d^k(\nabla^2 u(q)) & -d^kp & 0_{1\times 1} \\ -d^kp^T & 0_{1\times 1} & 0_{1\times 1} \\ d^k(\nabla u(q)) & 0_{1\times 1} & 0_{1\times 1} \end{bmatrix}, \quad d^kG(y) = \begin{bmatrix} d^k\lambda I_n \\ d^kq^T \\ 0_{1\times n} \end{bmatrix}.$$

The price differentials, d^kp, are given by equations (2.16). The quantity and Lagrange-multiplier differentials, d^kq and $d^k\lambda$, are elements of computed d^ky.

If we multiply differentiate $u(q)$ with respect to q but not further with respect to z, where q is a multiply differentiable funtction of z, then, vectorization rule (6.3) and gradient definition (6.14) imply that, for k and $m \geq 1$, $\text{vec}(d^k(\nabla^m u(q))) = [(\Pi_{k-1} \otimes dq^T) \otimes I_{n^{m-1}\times n}]\nabla^{k+m}u(q)dq$. However, if we multiply differentiate $u(q)$ with respect to z, where q is multiply differentiable with respect to z, then, chain rule (6.16) adds terms in $d^jq \neq 0$, for $j \geq 2$, to $\text{vec}(d^k(\nabla^m u(q)))$. Thus, repeatedly applying product rule of differentiation (6.17) to multiply differentiate $\nabla u(q)$ and $\nabla^2 u(q)$ with respect to time s and using vectorization rule (6.3) implies, for $m = 1$,

$$(2.13) \qquad \text{vec}(d(\nabla u(s))) = \nabla^2 u(s)dq(s),$$

$$\text{vec}(d^2(\nabla u(s))) = [dq(s)^T \otimes I_n]\nabla^3 u(s)dq(s) + \nabla^2 u(s)d^2q(s),$$

$$\text{vec}(d^3(\nabla u(s))) = [dq(s)^T \otimes dq(s)^T \otimes I_n]\nabla^4 u(s)dq(s)$$

$$+ 2[d^2q(s)^T \otimes I_n]\nabla^3 u(s)dq(s) + \nabla^2 u(s)d^3q(s),$$

and, similarly, for m = 2,

$$\text{(2.14)}\quad \text{vec}(d(\nabla^2 u(s))) = \nabla^3 u(s)dq(s),$$

$$\text{vec}(d^2(\nabla^2 u(s))) = [dq(s)^T \otimes I_{n^2}]\nabla^4 u(s)dq(s) + \nabla^3 u(s)d^2 q(s),$$

$$\text{vec}(d^3(\nabla^2 u(s))) = [dq(s)^T \otimes dq(s)^T \otimes I_{n^2}]\nabla^5 u(s)dq(s)$$

$$+ 2[d^2 q(s)^T \otimes I_{n^2}]\nabla^4 u(s)dq(s) + \nabla^3 u(s)d^3 q(s),$$

where $\nabla^k u(s) \equiv \nabla^k u(q(s))$, I_n and I_{n^2} are n×n and n^2×n^2 identity matrices, and $d^k q(s)$ is the top n-dimensional subvector of $d^k y(s)$. The definition of the gradient in the appendix implies $\nabla u(q) = 1\times n = [\partial u(q)/\partial q_1, \ldots, \partial u(q)/\partial q_n]$, $\nabla^2 u(q) = n\times n = [\partial \text{vec}(\nabla u(q))/\partial q_1, \ldots, \partial \text{vec}(\nabla u(q))/\partial q_n]$, $\ldots$, $\nabla^5 u(q) = n^4\times n = [\partial \text{vec}(\nabla^4 u(q))/\partial q_1, \ldots, \partial \text{vec}(\nabla^4 u(q))/\partial q_n]$.

We assume prices follow a 4th-order polynomial process for $s \in [t,t+1)$ and $t = 1, \ldots, T$, analogous to approximate y process (2.7),

$$\text{(2.15)}\quad p(s) = p'_t + \nabla p_t(s-t) + (1/2)\nabla^2 p_t(s-t)^2 + (1/6)\nabla^3 p_t(s-t)^3 + (1/24)\nabla^4 p_t(s-t)^4,$$

with n×1 price coefficients, p'_t, ∇p_t, $\ldots$, $\nabla^4 p_t$. Whereas the price coefficients remain at their initial values, indexed at $t_1 = t$, throughout computations in period t, the y coefficients, ∇y_{t_i}, $\ldots$, $\nabla^4 y_{t_i}$, are indexed by t_i and updated at each iteration $i = 1, \ldots, h$.

From price process (2.15), we require only that it passes through initial and ending prices in period t, i.e., $p(t) = p'_t$ and $p(t+1) = p_{t+1}$, because, in the discrete-time utility-maximizing problem, consumers care only about beginning- and end-of-period prices and do not care about within-period prices. Section 2.1 explained that initial prices, p'_t, are computed to make current observed quantities, q_t, optimal and generally differ from observed prices, p_t. However, as shown in figure 2, we want prices to vary nonlinearly in order to compute nonlinear or higher-order effects of price variations on Δc_t and ξ_t, in this case up to 4th-order effects.

For k = 4 and s ∈ [t,t+1), differentiating price process (2.15) four times with respect to s, implies

$$(2.16)\quad dp(s) = \nabla p_t + \nabla^2 p_t(s-t_i) + (1/2)\nabla^3 p_t(s-t_i)^2 + (1/6)\nabla^4 p_t(s-t_i)^3,$$

$$d^2p(s) = \nabla^2 p_t + \nabla^3 p_t(s-t_i) + (1/2)\nabla^4 p_t(s-t_i)^2,$$

$$d^3p(s) = \nabla^3 p_t + \nabla^4 p_t(s-t_i),$$

$$d^4p(s) = \nabla^4 p_t.$$

For k = 4, differencing price process (2.15) four times with respect to t, we express the price coefficients, ∇p_t, ..., $\nabla^4 p_t$, in terms of forward-differenced prices, Δp_t, ..., $\Delta^4 p_t$, as

$$(2.17)\quad \nabla p_t = \Delta p_t - (1/2)\Delta^2 p_t + (1/3)\Delta^3 p_t - (1/4)\Delta^4 p_t,$$

$$\nabla^2 p_t = \Delta^2 p_t - \Delta^3 p_t + (11/12)\Delta^4 p_t,$$

$$\nabla^3 p_t = \Delta^3 p_t - (3/2)\Delta^4 p_t,$$

$$\nabla^4 p_t = \Delta^4 p_t,$$

such that forward-differenced prices are defined relative to initial "optimal" computed prices, p'_t, and subsequent observed prices, as

$$(2.18)\quad \Delta p_t = p_{t+1} - p'_t,$$

$$\Delta^2 p_t = p_{t+2} - 2p_{t+1} + p'_t,$$

$$\Delta^3 p_t = p_{t+3} - 3p_{t+2} + 3p_{t+1} - p'_t,$$

$$\Delta^4 p_t = p_{t+4} - 4p_{t+3} + 6p_{t+2} - 4p_{t+1} + p'_t.$$

As required, price coefficients set according to forward differences, by equations (2.17) and (2.18), imply price process (2.15) passes through computed initial and observed future prices, p'_t, p_{t+1}, ..., p_{t+4}. Similarly, price coefficients set according to backward or centered differences, respectively, imply the price process passes through computed initial prices and observed past prices or computed initial prices and observed past and future prices. Although setting the price coefficients according to equations (2.17) and (2.18) does achieve a perfect fit of observed prices in period t and neighboring periods, the motivation for setting the price coefficients in this way is numerical, not statistical.

In seven numbered steps, we now discuss the details of computing Δc_t iteratively, for $t_i = t_1, \ldots, t_h$, as the last element of Δy_t. The steps pertain to any period $t = 1, \ldots, T$ and are fully recursive, so that every step can be completed as long as previous steps have been completed.

Step 1: Initialize x_t, prices, and their differentials. For $s = t_1 = t$, compute $x(s) = x_t = (y_t^T, p_t'^T)^T = (q_t^T, \lambda_t, c_t, p_t'^T)^T$, where q_t is observed, following first-order conditions (2.2) and (2.3), $\lambda_t = \nabla u(q_t) q_t$ and $p'_t = \nabla u(q_t)^T/\lambda_t$, and $c_t = 1$. Following equations (2.16), set price differentials as $dp(s) = dp_t = \nabla p_t$, ..., $d^4p(s) = d^4p_t = \nabla^4 p_t$ and, following equations (2.17) and (2.18), compute price coefficients, ∇p_t, ..., $\nabla^4 p_t$, in terms of initial computed and future observed prices, p'_t, p_{t+1}, ..., p_{t+4}.

Step 2: Compute 1st-order y coefficient. Because true and approximate y processes are evaluated at the same times, $s = t_i = t_1, \ldots, t_h$, in all remaining steps y differentials and coefficients are equal, i.e., $d^k y(t_i) = \nabla^k y_{t_i}$. Thus, following equation (2.6), the first equation of (2.9), and the first equations of (2.16) to (2.18), compute

$$H(x_t) = F(x_t)^{-1} G(y_t), \tag{2.19}$$

$$\nabla y_t = H(x_t) dp_t.$$

Step 3. Compute 2nd-order y coefficient. Following the first equations of (2.13) and (2.14), compute

$$\mathrm{vec}(d(\nabla u(q_t))) = \nabla^2 u(q_t) \nabla q_t, \tag{2.20}$$

$$\text{vec}(d(\nabla^2 u(q_t))) = \nabla^3 u(q_t)\nabla q_t,$$

where ∇q_t is the top n-dimensional subvector of ∇y_t. Following equation (2.12), compute

$$(2.21)\qquad dF(x_t) = \begin{bmatrix} d(\nabla^2 u(q_t)) & -dp_t & 0_{1\times 1} \\ -dp_t^T & 0_{1\times 1} & 0_{1\times 1} \\ d(\nabla u(q_t)) & 0_{1\times 1} & 0_{1\times 1} \end{bmatrix}, \quad dG(y_t) = \begin{bmatrix} \nabla\lambda_t I_n \\ \nabla q_t^T \\ 0_{1\times n} \end{bmatrix},$$

where $\nabla\lambda_t$ is the next to last element of ∇y_t. Following the first equation of (2.11), compute

$$(2.22)\qquad dH(x_t) = F(x_t)^{-1}[dG(y_t) - dF(x_t)H(x_t)].$$

Following the second equation of (2.9), the first equation of (2.10), the first and second equations of (2.16) to (2.18), and, using $d^2y(t_1) = \nabla^2 y_t$, compute

$$(2.23)\qquad \nabla^2 y_t = dH(x_t)dp_t + H(x_t)d^2p_t.$$

Step 4: Compute 3rd-order y coefficient. Following the second equations of (2.13) and (2.14), compute

$$(2.24)\qquad \text{vec}(d^2(\nabla u(q_t))) = [\nabla q_t^T \otimes I_n]\nabla^3 u(q_t)\nabla q_t + \nabla^2 u(q_t)\nabla^2 q_t,$$

$$\text{vec}(d^2(\nabla^2 u(q_t))) = [\nabla q_t^T \otimes I_{n^2}]\nabla^4 u(q_t)\nabla q_t + \nabla^3 u(q_t)\nabla^2 q_t,$$

where $\nabla^2 q_t$ is the top n-dimensional subvector of $\nabla^2 y_t$. Following equation (2.12), compute

$$(2.25)\qquad d^2F(x_t) = \begin{bmatrix} d^2(\nabla^2 u(q_t)) & -d^2p_t & 0_{1\times 1} \\ -d^2p_t^T & 0_{1\times 1} & 0_{1\times 1} \\ d^2(\nabla u(q_t)) & 0_{1\times 1} & 0_{1\times 1} \end{bmatrix}, \quad d^2G(y_t) = \begin{bmatrix} \nabla^2\lambda_t I_n \\ \nabla^2 q_t^T \\ 0_{1\times n} \end{bmatrix},$$

where $\nabla^2 q_t$ is the top n-dimensional subvector and $\nabla^2 \lambda_t$ is the next to last element of $\nabla^2 y_t$. Following the second equation of (2.11), compute

$$(2.26) \qquad d^2H(x_t) = F(x_t)^{-1}[d^2G(y_t) - d^2F(x_t)H(x_t) - 2dF(x_t)dH(x_t)].$$

Following the third equation of (2.9), the second equation of (2.10), the first to third equations of (2.16) to (2.18), and, using $d^3y(t_1) = \nabla^2 y_t$, compute

$$(2.27) \qquad \nabla^3 y_t = d^2H(x_t)dp_t + 2dH(x_t)d^2p_t + H(x_t)d^3p_t.$$

Step 5: Compute 4th-order y coefficient and update y. Following the third equations of (2.13) and (2.14), compute

$$(2.28) \qquad \mathrm{vec}(d^3(\nabla u(q_t))) = [\nabla q_t^T \otimes \nabla q_t^T \otimes I_n]\nabla^4 u(q_t)\nabla q_t$$

$$+ 2[\nabla^2 q_t^T \otimes I_n]\nabla^3 u(q_t)\nabla q_t + \nabla^2 u(q_t)\nabla^3 q_t,$$

$$\mathrm{vec}(d^3(\nabla^2 u(q_t))) = [\nabla q_t^T \otimes \nabla q_t^T \otimes I_{n^2}]\nabla^5 u(q_t)\nabla q_t$$

$$+ 2[\nabla^2 q_t^T \otimes I_{n^2}]\nabla^4 u(q_t)\nabla q_t + \nabla^3 u(q_t)\nabla^3 q_t,$$

where $\nabla^3 q_t$ is the top n-dimensional subvector of $\nabla^3 y_t$. Following equation (2.12), compute

$$(2.29) \qquad d^3F(x_t) = \begin{bmatrix} d^3(\nabla^2 u(q_t)) & -d^3p_t & 0_{1\times 1} \\ -d^3p_t^T & 0_{1\times 1} & 0_{1\times 1} \\ d^3(\nabla u(q_t)) & 0_{1\times 1} & 0_{1\times 1} \end{bmatrix}, \quad d^3G(y_t) = \begin{bmatrix} \nabla^3\lambda_t I_n \\ \nabla^3 q_t^T \\ 0_{1\times n} \end{bmatrix},$$

where $\nabla^3 \lambda_t$ is the next to last element of $\nabla^3 y_t$. Following the third equation of (2.11), compute

$$(2.30) \qquad d^3H(x_t) = F(x_t)^{-1}[d^3G(y_t) - d^3F(x_t)H(x_t) + 3d^2F(x_t)dH(x_t) + 3dF(x_t)d^2H(x_t)].$$

Following the fourth equation of (2.9), the third equation of (2.10), the first to fourth equations of (2.16) to (2.18), and, using $d^4y(t_1) = \nabla^4 y_t$, compute

(2.31) $\nabla^4 y_t = d^3H(x_t)dp_t + 3d^2H(x_t)d^2p_t + 3dH(x_t)d^3p_t + H(x_t)d^4p_t.$

Following equation (2.8), compute

(2.32) $\Delta\hat{y}_t = \nabla y_t h^{-1} + (1/2)\nabla^2 y_t h^{-2} + (1/6)\nabla^3 y_t h^{-3} + (1/24)\nabla^4 y_t h^{-4}$

and update y as $y_{t_2} = y_t + \Delta\hat{y}_t$.

Step 6: Update price coefficients, x_{t_2}, and y. For $s = t_2 = t + 1/h$, following equations (2.15) and (2.16), update prices and their differentials, as

(2.33) $p_{t_2} = p'_t + \nabla p_t(1/h) + (1/2)\nabla^2 p_t(1/h)^2 + (1/6)\nabla^3 p_t(1/h)^3 + (1/24)\nabla^4 p_t(1/h)^4,$

$dp_{t_2} = \nabla p_t + \nabla^2 p_t(1/h) + (1/2)\nabla^3 p_t(1/h)^2 + (1/6)\nabla^4 p_t(1/h)^3,$

$d^2p_{t_2} = \nabla^2 p_t + \nabla^3 p_t(1/h) + (1/2)\nabla^4 p_t(1/h)^2,$

$d^3p_{t_2} = \nabla^3 p_t + \nabla^4 p_t(1/h),$

$d^4p_{t_2} = \nabla^4 p_t,$

such that price coefficients ∇p_t, ..., $\nabla^4 p_t$ remain at initially computed values. Set $x_{t_2} = (y_{t_2}^T, p_{t_2}^T)^T$. For $s = t_2$, repeat steps 2 to 5 and update y coefficients to ∇y_{t_2}, ..., $\nabla^4 y_{t_2}$. Following equation (2.32), compute $\Delta\hat{y}_{t_2} = \nabla y_{t_2} h^{-1} + (1/2)\nabla^2 y_{t_2} h^{-2} + (1/6)\nabla^3 y_{t_2} h^{-3} + (1/24)\nabla^4 y_{t_2} h^{-4}$ and update y as $y_{t_3} = y_{t_2} + \Delta\hat{y}_{t_2}$.

Step 7: Repeat steps 2 to 6. For $s = t_3 = t + 2/h$, update prices and their differentials, as

(2.34) $p_{t_3} = p'_t + \nabla p_t(2/h) + (1/2)\nabla^2 p_t(2/h)^2 + (1/6)\nabla^3 p_t(2/h)^3 + (1/24)\nabla^4 p_t(2/h)^4,$

$$dp_{t_3} = \nabla p_t + \nabla^2 p_t (2/h) + (1/2)\nabla^3 p_t (2/h)^2 + (1/6)\nabla^4 p_t (2/h)^3,$$

$$d^2 p_{t_3} = \nabla^2 p_t + \nabla^3 p_t (2/h) + (1/2)\nabla^4 p_t (2/h)^2,$$

$$d^3 p_{t_3} = \nabla^3 p_t + \nabla^4 p_t (2/h),$$

$$d^4 p_{t_3} = \nabla^4 p_t.$$

Set $x_{t_3} = (y_{t_3}^T, p_{t_3}^T)^T$. For $s = t_3$, repeat steps 2 to 5 and update y coefficients to ∇y_{t_3}, ..., $\nabla^4 y_{t_3}$. Compute $\Delta\hat{y}_{t_3} = \nabla y_{t_3} h^{-1} + (1/2)\nabla^2 y_{t_3} h^{-2} + (1/6)\nabla^3 y_{t_3} h^{-3} + (1/24)\nabla^4 y_{t_3} h^{-4}$ and update y as $y_{t_4} = y_{t_3} + \Delta\hat{y}_{t_3}$. Repeat these steps for $s = t_4 = t + 3/h$, ..., $s = t_h = t + (h-1)/h$. At the last step, compute $\Delta\hat{y}_{t_h}$ and pick Δc_t as the last element of $\Delta\hat{y}_{t_h}$.

2.3. Computing Residual Quantities.

We compute $-\exp(\xi_t) = \hat{q}_t - q_t$ exactly like Δc_t, except that we delete all reference to c_t in y, x, F(x), and G(y), because now $c_t \equiv 1$ so that it drops out as a variable. Thus, now $y = (q^T, \lambda)^T$, as before $x = (y^T, p^T)^T$, F(x) is still defined by equation (2.4) but with the last row and column deleted, and G(y) is still defined by equation (2.4) but with the last row deleted. Thus, equation (2.4) now becomes

$$(2.35) \qquad F(x)dy = G(y)dp \quad \text{or} \quad \begin{bmatrix} \nabla^2 u(q) & -p \\ -p^T & 0_{1\times 1} \end{bmatrix} \begin{bmatrix} dq \\ d\lambda \end{bmatrix} = \begin{bmatrix} \lambda I_n \\ q^T \end{bmatrix} dp.$$

After redefining F(x) and G(x) according to equation (2.35), we compute Δy_t in response to $\Delta p_t = p_t - p_t'$ exactly as we computed Δy_t in response to $\Delta p_t = p_{t+1} - p_t'$ in the previous section. Previously, we started at the price-quantity combination p_t' and q_t, corresponding to point A on budget line AA in figure 1, and ended at the price-quantity combination p_{t+1} and q_B, where q_B corresponds to point B on budget line BB in figure 1. Now, we start at the same price-quantity

combination and end at the price-quantity combination p_t and $\hat{q}_t$, which corresponds to point E on budget line EE in figure 1. Previously, we picked Δc_t as the last element of computed Δy_t. Now, we pick $-\exp(\xi_t) = \hat{q}_t - q_t$ as the top n-dimensional subvector of computed Δy_t.

3. Statistical Analysis and Time-Varying Utility Functions.

[to be completed]

4. Illustrative Application.

[to be completed]

5. Conclusion.

[to be completed]

6. Appendix: Matrix Differentiation.

6.1. Definitions of Matrix Derivatives.

Let $A(x) \in \mathbf{D}^k$: $\mathbf{R}^n \to \mathbf{R}^{p\times q}$ be a K-times differentiable p×q matrix function of the n×1 vector x. A(x) could be a function of the matrix $X \in \mathbf{R}^{k\times m}$, such that $x = \mathrm{vec}(X)$, where $\mathrm{vec}(\cdot)$ is the columnwise vectorization of a matrix. We consider derivatives of elements of A with respect to elements of x in three forms: the ∂ or partial derivative form, the d or differential form, and the ∇ or gradient form.

For $k = 1, \ldots, K$ and $i_1, \ldots, i_k \in \{1, \ldots, n\}$, we define $\partial^k_{i_1\cdots i_k} A \in \mathbf{R}^{p\times q}$ by

$$\partial^k_{i_1\cdots i_k} A = \begin{bmatrix} \dfrac{\partial^k A_{11}}{\partial x_{i_1}\cdots\partial x_{i_k}} & \cdots & \dfrac{\partial^k A_{1q}}{\partial x_{i_1}\cdots\partial x_{i_k}} \\ \vdots & & \vdots \\ \dfrac{\partial^k A_{p1}}{\partial x_{i_i}\cdots\partial x_{i_k}} & \cdots & \dfrac{\partial^k A_{pq}}{\partial x_{i_1}\cdots\partial x_{i_k}} \end{bmatrix}, \tag{6.1}$$

as the partial derivative form of k-th order partial derivatives of the elements of A with respect to x_{i_1}, ..., x_{i_k}.

The differential form associated with (6.1) is

$$(6.2) \qquad d^kA = \sum_{i_1=1}^{n} \cdots \sum_{i_k=1}^{n} \partial_{i_1 \cdots i_k} A \cdot dx_{i_1} \cdots dx_{i_k},$$

where the dx_i's are small (strictly, infinitesimal) increments to the elements of $x = (x_1, \ldots, x_n)^T$.

The gradient form associated with (6.1) and (6.2) can now be built up recursively, starting with k = 1. We call matrix representations of kth derivatives of vector function or vectorizations of matrix functions "k-gradients," which generalizes common terminology. For example, for scalar functions the 1-gradient is the gradient (in the usual sense) and the 2-gradient is the Hessian; for vector functions or vectorizations of matrix functions, the 1-gradient is the Jacobian.

We use the following rule for vectorizing matrix products,

$$(6.3) \qquad vec(ABC) = [C^T \otimes A]vec(B),$$

where A, B, and C are matrices conformable to the matrix product ABC and $\otimes$ denotes the Kronecker matrix product (Magnus and Neudecker, 1988, p.30).

6.2. Representations of Matrix Derivatives.

For k = 1, (6.1) and (6.2) become

$$(6.4) \qquad \partial_i A = \begin{bmatrix} \frac{\partial A_{11}}{\partial x_i} & \cdots & \frac{\partial A_{1q}}{\partial x_i} \\ \vdots & & \vdots \\ \frac{\partial A_{p1}}{\partial x_i} & \cdots & \frac{\partial A_{pq}}{\partial x_i} \end{bmatrix},$$

$$(6.5) \qquad dA = \sum_{i=1}^{n} \partial_i A \cdot dx_i.$$

Note that vectorization, summation, and differentiation operations are commutative, i.e., can be applied in any order. Therefore, we vectorize (6.5), to obtain

$$\text{(6.6)} \qquad \mathrm{vec}(dA) = [\partial_1 \mathrm{vec}(A), \ldots, \partial_n \mathrm{vec}(A)]dx,$$

where $dx = (dx_1, \ldots, dx_n)^T$, so that

$$\text{(6.7)} \qquad \mathrm{vec}(dA) = \nabla A \cdot dx,$$

$$\text{(6.8)} \qquad \nabla A = [\partial_1 \mathrm{vec}(A), \ldots, \partial_n \mathrm{vec}(A)].$$

Equations (6.7) and (6.8) relate the ∂, d, and ∇ forms of first-order derivatives of A to each other.

To obtain analogues of (6.7) and (6.8) for k = 2, we differentiate them to obtain

$$\text{(6.9)} \qquad \mathrm{vec}(d^2A) = d(\nabla A)dx,$$

$$\text{(6.10)} \qquad d(\nabla A) = [d(\mathrm{vec}(\partial_1 A)), \ldots, d(\mathrm{vec}(\partial_n A))]$$

$$= \sum_{j=1}^{n} [\partial_j(\mathrm{vec}(\partial_1 A)), \ldots, \partial_j(\mathrm{vec}(\partial_n A))]dx_j.$$

Then, we vectorize (6.10) to obtain

$$\text{(6.11)} \qquad \mathrm{vec}(d(\nabla A)) = \sum_{j=1}^{n} \begin{bmatrix} \partial_j(\mathrm{vec}(\partial_1 A)) \\ \vdots \\ \partial_j(\mathrm{vec}(\partial_n A)) \end{bmatrix} dx_j = \begin{bmatrix} \partial_1(\mathrm{vec}(\partial_1 A)) & \cdots & \partial_n(\mathrm{vec}(\partial_1 A)) \\ \vdots & & \vdots \\ \partial_1(\mathrm{vec}(\partial_n A)) & \cdots & \partial_n(\mathrm{vec}(\partial_n A)) \end{bmatrix} dx.$$

Then, because $\mathrm{vec}(\nabla A) = \begin{bmatrix} \partial_1(\mathrm{vec}(A)) \\ \vdots \\ \partial_n(\mathrm{vec}(A)) \end{bmatrix} = \begin{bmatrix} \mathrm{vec}(\partial_1 A) \\ \vdots \\ \mathrm{vec}(\partial_n A) \end{bmatrix}$, we obtain

$$\text{(6.12)} \qquad \mathrm{vec}(d(\nabla A)) = [\partial_1 \mathrm{vec}(\nabla A), \ldots, \partial_n \mathrm{vec}(\nabla A)]dx.$$

Continuing in this manner for k = 2, ..., K, we obtain

(6.13) $\mathrm{vec}(d(\nabla^{k-1}A)) = \nabla^k A \cdot dx,$

(6.14) $\mathrm{vec}(d^k A) = [(\Pi_{k-1} \otimes dx^T) \otimes I_{pq}]\nabla^k A \cdot dx,$

where $\Pi_{k-1} \otimes dx^T$ denotes k-2 successive Kronecker products of dx^T ($\Pi_0 \otimes dx^T = dx^T$, $\Pi_1 \otimes dx^T = dx^T \otimes dx^T$, ...), and

(6.15) $\nabla^k A = [\partial_1 \mathrm{vec}(\nabla^{k-1}A), \ ..., \ \partial_n \mathrm{vec}(\nabla^{k-1}A)].$

Applied for k = 1, ..., K, (6.15) recusively organizes gradient form derivatives of A up to order K as matrices. Basically, $\nabla^k A$ is the Jacobian matrix of the vectorization of $\nabla^{k-1}A$.

6.3. Differentiation Rules.

Let $A(x) \in \mathbf{D}\colon \mathbf{R}^n \to \mathbf{R}^p$ and $B(y) \in \mathbf{D}\colon \mathbf{R}^p \to \mathbf{R}^q$ be differentiable vector functions (or vectorizations of matrix functions). Let $C(x) \in \mathbf{D}\colon \mathbf{R}^n \to \mathbf{R}^q$ be the differentiable composite vector function C(x) = B(A(x)). Then, the gradient form of derivatives of C(x) is given by the chain rule of differentiation,

(6.16) $\nabla C(x) = \nabla B(A) \cdot \nabla A(x).$

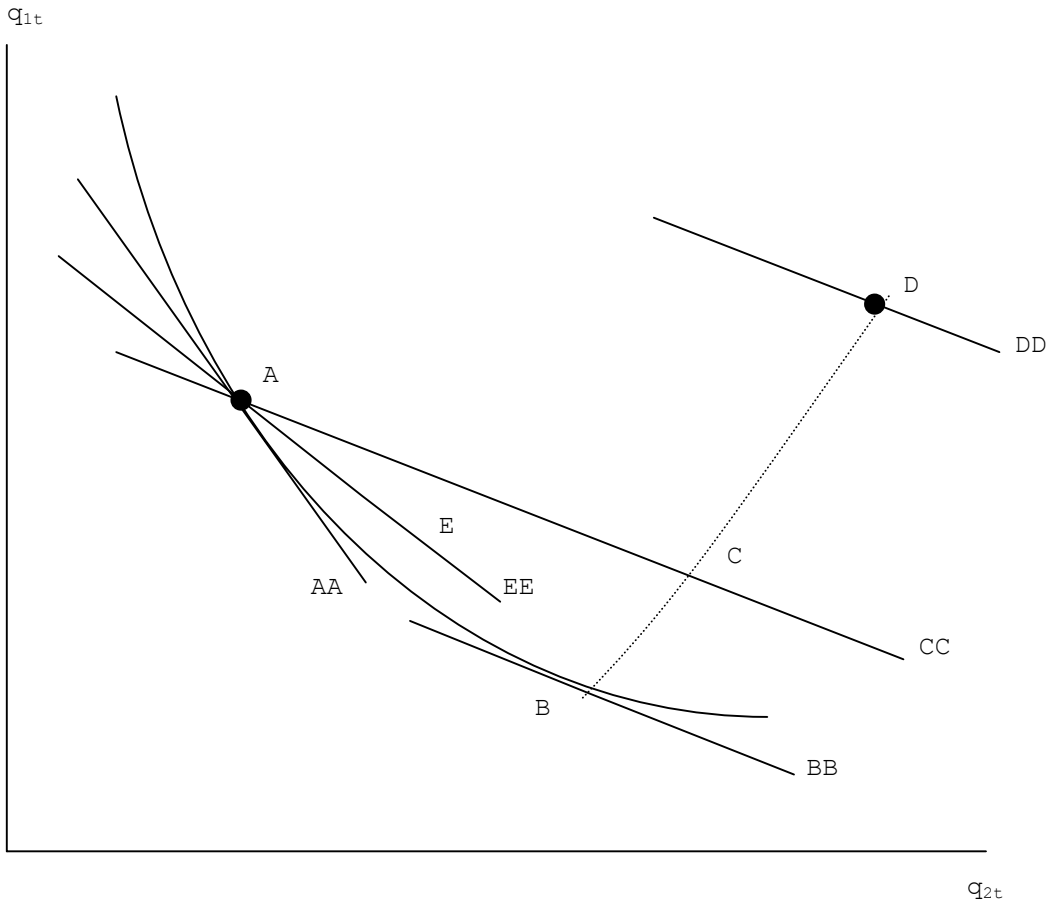
Let $A(x) \in \mathbf{D}\colon \mathbf{R}^n \to \mathbf{R}^{p \times q}$ and $B(x) \in \mathbf{D}\colon \mathbf{R}^n \to \mathbf{R}^{p \times q}$ be differentiable matrix functions conformable to the ordinary matrix product C(x) = A(x)·B(x). Then, the differential form of derivatives of C(x) is given by the product rule of differentiation,

(6.17) $dC(x) = dA(x) \cdot B(x) + A(x) \cdot dB(x).$

Rules (6.16) and (6.17) are quickly proved by elementwise application of the scalar chain rule of differentiation and the scalar product rule of differentiation.

7. Figures and Tables.

Figure 1: Price Substitution and Real Income Effects



Explanation: Figure 1 depicts the current period, t. The points A to E are defined as follows: A is the currently observed consumption point, at current observed quantitites, q_t; B is optimal on budget line BB, on the same indifference curve AB as A, at next period's observed prices, p_{t+1}; C is optimal on budget line CC, through A, at next period's prices; D is optimal on budget line DD, at next period's prices and current nominal income, e_t; E is optimal on observed budget line EE, through A, at current prices and current income. The budget lines are defined as follows: AA is given by $\tilde{p}_t'^{T}q = 1$, for $\tilde{p}_t' = \nabla u(q_t)^T/\nabla u(q_t)q_t$; BB is given by $\tilde{p}_{t+1}^{T}q = \varphi_t(p_{t+1}^{T}q_t/e_t)$, for $0 < \varphi_t < 1$; CC is given by $\tilde{p}_{t+1}^{T}q = p_{t+1}^{T}q_t/e_t < 1$; DD is given by $\tilde{p}_{t+1}^{T}q = 1$; and, EE is given by $\tilde{p}_t^{T}q = 1$.

Figure 2: Example of Polynomial Price Path

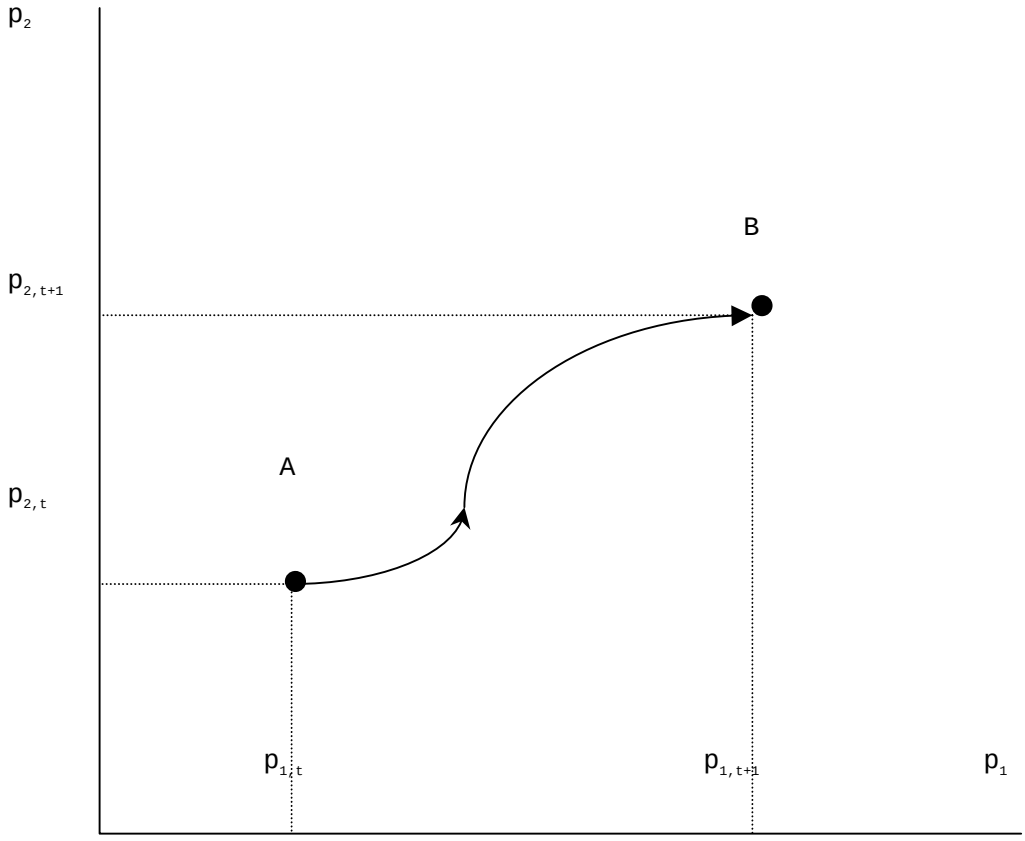


Table 1: Accuracy of Perturbed Polynomial Path Method

Accuracy Order of Magnitude ε	Polynomial Order k	Step Size h^{-1}	Number of Iterations h
Semi-single Precision $\varepsilon = O(10^{-4})$	1	1.00×10^{-4}	10^4
	2	1.00×10^{-2}	10^2
	3	4.55×10^{-2}	22
	4	1.00×10^{-1}	10
	5	1.43×10^{-1}	7
	6	2.00×10^{-1}	5
Single Precision $\varepsilon = O(10^{-8})$	1	1.00×10^{-8}	10^8
	2	1.00×10^{-4}	10^4
	3	2.15×10^{-3}	465
	4	1.00×10^{-2}	100
	5	2.50×10^{-2}	40
	6	4.55×10^{-2}	22
Double Precision $\varepsilon = O(10^{-16})$	1	1.00×10^{-16}	10^{16}
	2	1.00×10^{-8}	10^8
	3	4.64×10^{-6}	215,444
	4	1.00×10^{-4}	10^4
	5	6.31×10^{-4}	1585
	6	2.15×10^{-3}	465

Comments: Given the polynomial order of approximation, k, and the number of iterations or summed integrals, h (both positive integers), the accuracy of the final computed result is about $\varepsilon \cong h^{-k}$ (a positive real number). Alternately, given the desired accuracy, ε, and the polynomial order, k, the required number of summed terms to achieve the desired accuracy is $h = \lceil \varepsilon^{-1/k} \rceil$, where $\lceil x \rceil$ is the ceiling function, i.e., the smallest integer ≥ real number x.

8. References.

Apostol, T.M. (1974), Mathematical Analysis, 2nd ed., Reading, MA: Addison-Wesley.

Arrow, K.J, H.B. Chenery, B.S. Minhas, and R.M. Solow (1961), "Capital-Labor Substitution and Economic Efficiency," Review of Economics and Statistics 43: 225-250.

Chen, B. and P.A. Zadrozny (2003), "Higher Moments in Perturbation Solution of the Linear-Quadratic Exponential Gaussian Optimal Control Problem," Computational Economics 21: 45-64.

Diewert, W.E. (1976), "Exact and Superlative Index Numbers," Journal of Econometrics 4: 115-145.

Fleming, W.H. (1971), "Stochastic Control for Small Noise Intensities," SIAM Journal of Control 9: 473-517.

Ford, L.R. (1955), Differential Equations, New York, NY: McGraw-Hill.

Golub, G.H. and C.F. Van Loan (1996), Matrix Computations, third edition, Baltimore, MD: Johns Hopkins University Press.

Griewank, A. (2000), Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, Philadelphia, PA: Society for Industrial and Applied Mathematics.

Hausman, J. (2003), "Sources of Bias and Solutions to Bias in the Consumer Price Index," Journal of Economic Perspectives 17: 23-44.

Judd, K.L. (1998), Numerical Methods in Economics, Cambridge, MA: M.I.T. Press.

Judd, K.L. (2003), "The Parametric Path Method: an Alternative to Fair-Taylor and L-B-J for Solving Perfect Foresight Models," Journal of Economic Dynamics and Control 26: 1557-1583.

Kaiser, R. and A. Maravall (2001), "An Application of TRAMO-SEATS: Changes in Seasonality and Current Trend-Cycle Assessment" working paper, Research Department, Bank of Spain, Madrid, Spain.

Magnus, J.R. and H. Neudecker (1988), Matrix Differential Calculus with Applications in Statistics and Econometrics, New York, NY: J. Wiley & Sons.

Mann, H.B. (1943), "Quadratic Forms with Linear Constraints," American Mathematical Monthly 7: 430-433.

Schultze, C.L. (2002), ed., At What Price? Conceptualizing and Measuring Cost-of-Living and Price Indexes, Washington, DC: National Academy Press.