TFP Measurement: A Guide

C. Luke Watson[*]

18 June 2018

Abstract

TFP measures everything that contributes to output that is not among the measurable physical factors of production. That is, TFP is a residual. There are two methodologies to recover this residual, each using separate assumptions. First, the "cost-share index" method (CS) is calculated from data without an 'estimation' step and requires a constant returns to scale assumption. This index method is used by the NBER-CES TFP measures. Second, the "production (*or* revenue) function estimation" method (PF) is a structural estimation procedure requiring economic and statistical assumptions. Both procedures are subject to measurement error and data limitations. PF must also deal with simultaneity bias in estimation. This guide describes how to calculate these measures and lists the major assumptions that are required.

Elements of this guide are drawn from

- Van Beveren - J.Econ Surveys (2010)
- Wooldridge - Econ Letters (2009)
- Mollisi & Rovigatti - WP (2017)
- Notes by P.Schrimpf at UBC

[*] Watson: Department of Economics, Michigan State University, email: watso220@msu.edu.

1 Background

1.1 Objective

You want to estimate the productivity of a firm (*or* plant)[1] that is not directly attributable to the measureable factors of production. Assume that there exists a production function for a single product firm producing output quantity Y using a combination of factor quantities labor (L), capital (K), and materials (M). This analysis can be done at the plant, firm, or industry level. I will generically refer to a firm, but the relevant primative is the production function that links inputs to output.

1.2 Production Function and Productivity Terms

Suppose that firms use a Cobb-Douglas production function:[2]

$$Y_{it} = A_{it} K_{it}^{\alpha_K} L_{it}^{\alpha_L} M_{it}^{\alpha_M}$$

The term A_{it} is the Hicksian neutral efficiency level for the firm i in period t. The α_j terms are the factor output elasticities: $\alpha_j = (\frac{\partial Y}{\partial X_j}\frac{X_j}{Y})$. The key econometric property of this model is that the CB function is log-additive (where $x = log(X)$):

$$y_{it} = a_{it} + \alpha_K k_{it} + \alpha_L l_{it} + \alpha_M m_{it}$$

If we are interested in calculating firm specific productivity, then we can decompose a into:

$$a_{it} = \mu + \nu_{it} + e_{it} = \omega_{it} + e_{it}$$

The first term is a mean efficiency level, the second is a firm expected component, and the third term is a nusiance parameter that may contain unforeseeable randomness in production or measurement error. If we have estimates of $\{\hat{\alpha}_j\}$, then we can calculate $\hat{a}$:

$$\hat{a}_{it} = y_{it} - \hat{\alpha}_K k_{it} + \hat{\alpha}_L l_{it} + \hat{\alpha}_M m_{it}$$

Note that, econometrically, $\hat{a}_{it}$ is a residual term. Thus, any econometric problems with estimating an OLS production function will be present for TFP measures.

1.3 A point of Caution

I believe there is a potential for inconsistency over exactly what is measured as productivity. This decomposition of the productivity residual points to some potential for alternative measures. Which part of a is TFP: $\{a, \nu, \omega\}$? How is measurement error or other noise, e, dealt with? How to compare TFP across time or industry?

For example, I believe the Haltiwanger approach[3] is to demean $\{\hat{a}_{it}\}_{i,t}^{N,T}$ by industry and year. I believe this implies that the correct (*or useful*) TFP measure is ν_{it}. Alternatively, in a PF method, ω_{it} is calcuable from the estimation procedure, which could then be demeaned.

[1] Firm refers to ownership; plant refers to the location where production occurs.

[2] The CB production function is by far the leading case; however, there are alternative primatives such as a translog

[3] There is no indication that this should be named after him, but I personally associate this with him and his author group.

For the remainder of this note, ω_{it} refers to firm-level (log) *Total Factor Productivity*. While the statistical error, e, is part of firm efficiency, this should **not** be considered part of TFP. A naive calculation of $\hat{a}_{it}$ (as above) will necessarily include e_{it}. A structural interpretation of e is that this accounts for external productivity factors (e.g., weather) or unanticipated, 'free lunch' productivity that is not *really* due to the firm or its managemnt. Additionally, the error accounts for measurement error in the included variables, so should not be assigned to firm TFP.

1.4 What Data Allows

This next point is a mix between theoretical and inevitable practicalities of estimating TFP. Many firm level datasets do not include quantities nor firm prices for factor or output. Rather, the researcher only has expenditures on inputs and revenues from outputs, and industry level prices / price-deflators. There are two roads to go down.

Strong assuptions about factor and input demand and market structure can make these issues moot. If there is perfect competition with prices set globally in every market, then the industry prices will be firm prices and all firms will sell output at the same price. Thus the researcher can use the deflators and procede as if using quantities.

The other road is to contront this fact and adjust what and how TFP is estimated. The main result is that components of demand (via prices) are now fused in the primary equations of interest. The silver lining is that demand is quite important for firm survival and growth, so the TFP measure that is returned is still of economics value.

A common modeling assumption is to take seriously the market structure and demand in the output market, but assume that firms are all price takers in the factor market. This is most likely for convenience; although, this may be more believable for capital and energy markets.

1.4.1 TFPQ and TFPR

In short, $\mathsf{TFPQ}_{it} = A_{it}$ and $\mathsf{TFPR}_{it} = P_{it}A_{it}$, where P is output price. TFPQ represents technical efficiency in production, while TFPR adds information about the firm's place in the market. High prices may signal better quality or relative monopoly power in a location, which increase the growth of the firm. In most cases, TFPR is what can be estimated.

An issue where this distinction may matter is looking at relatively young firms. Young firms may start because they are a new more technically productive method, but a lack of market presence may lead to lower demand / price. The inverse relationship for these firms in TFPQ and TFPR will obscure some market dynamics.

1.5 From Firms to Industry

To calculate an industry productivity measure, use a weighted sum, where the weights can be based on employment, output, or value-added shares:

$$\mathcal{J} = \sum_{i \in J} \{w(i, J) \cdot \omega_{it}\}$$

2 Method 1: Cost Share Index

This method requires no formal estimation, and is the method that is used for the industry level NBER CES Productivity Measures. However, the documentation to replicate these TFP measures, here, is in an older, separate file that is not linked to on the page.

2.1 Assumptions

If there is constant returns to scale and no input factor adjustment costs, then the industry factor-revenue cost share of a cost-minimizing firm is equal to the factor output elasticity. That is:

$$\mathsf{cs}_{j,it} = \left(\frac{\text{Factor j Wage Bill}}{\text{Revenue}}\right)_{it} = \alpha_j$$

Typically, industry level cost shares are used rather than firm/plant level for estimating firm TFP. Using industry level cost shares implies addition market structure assumptions. FGHW (2017) show that using firm/plant level cost share creates much more dispersion in TFP and is less related to firm/plant survival, which they interpret as implying measurement error at lower levels supercedes potential market structure issues.

Note: in NBER calculation, use two year average $0.5 \times (\mathsf{cs}_{j,it} + \mathsf{cs}_{j,i(t-1)})$ to smooth out differences.

2.2 TFP Calculation

Given $\{\mathsf{cs}_{j,it}\}^{J,N,T}$,TFP can be calculated as

$$\begin{aligned} a_{it} &= y_{it} - \sum_{j \in J} \{\mathsf{cs}_{j,it} \cdot x_{j,it}\} \\ A_{it} &= \mathsf{e}^{a_{it}} \end{aligned}$$

However, NBER calculates industry TFP using a growth rate formula, then integrates to levels:

$$\begin{gathered} \mathsf{d}a_{it} = \mathsf{d}y_{it} - \sum_{j \in J} \{\mathsf{cs}_{j,it} \cdot \mathsf{d}x_{j,it}\} \\ A_{it} = \mathsf{e}^{\left(\ln[A_{i(t-1)}] + \mathsf{d}a_{it}\right)} = A_{i(t-1)} \cdot \mathsf{e}^{(\mathsf{d}a_{it})} \\ \text{where } A_{i(0)} = 1 \text{ and then normalization of } A_{i(t=1997)} == 1 \end{gathered}$$

This difference appears to matter. Using the NBER industry data, the correlation between TFP using the level (above) and growth (below) is 0.76.

The vast majority of the papers summaried use the above version rather than the below . . . I cannot actually think of any paper that uses the second calculation, but I do not want to rule it out.

2.3 Stata Calculate NBER TFP4

```
* read in data
use "naics5811.dta", clear

* rename based on taste
ren naics naics6

* Calculate wage bill and emp of non-production workers
gen nonprodw = (pay - prodw)
gen nonprode = (emp - prode)

* Generate Industry cost share -- note: nominal values
gen sh_Pr = prodw / vship
gen sh_NPr = nonprodw / vship
gen sh_Mat = matcost / vship

* Calculate capital cost share as residual
* --> forces CRS assumption
gen sh_K = 1 - sh_Pr - sh_NPr - sh_Mat
* Should check to see no sh_K>1
sum sh_K,d

* Calculate two-year average cost share
foreach vari of varlist sh_* {
   bys naics6 (year) : gen L_‘vari’ = ‘vari’[_n-1]
   gen y2_‘vari’ = 0.5*(‘vari’ + L_‘vari’)
}

* Use industry deflator to get real materials and output
gen rmat = matcost / pimat
gen rvship = vship / piship

* Calculate log growth rates
foreach vari of varlist nonprode prodh cap rmat rvship {
   bys naics6 (year) : gen dl_‘vari’ = log(‘vari’[_n]) - log(‘vari’[_n-1])
}

* Generate growth rate of TFP
* 1) calc weighted factor contribution
* 2) take difference from true output growth = dl_tfp
gen dl_prY = (y2_sh_Pr*dl_prodh + y2_sh_NPr*dl_nonprode ///
               + y2_sh_Mat*dl_rmat + y2_sh_K*dl_cap)
gen dl_tfp = dl_rvship - dl_prY

* To get tfp levels, set earliest tfp to 1, then calculate forward
gen tfp = 1 if year > 1958
bys naics6 (year) : replace tfp= exp(ln(tfp[_n-1]) + (dl_tfp)) if year>1959
```

```
* NBER normalizes to 1997
gen tfp97 = tfp if year==1997
bys naics6 : egen tfp97_all = mean(tfp97)
replace tfp = tfp / tfp97_all

* Test
corr tfp tfp4
(obs=24857)

      |     tfp     tfp4
-------------------------------
tfp   |   1.0000
tfp4  |   1.0000   1.0000

/*
Note:
NBER current public documentation says it uses non-prod-hours.
It does not.
The earlier documentation clearly states that non-prod-emp is used for TFP.
Prod-hours are used.
If one reruns this procedure but uses prod-emp,
    then the correlation of TFP using prod-hours vs prod-emp is 0.64.
*/

********************************************
**** Compare versions based on TFP calc
********************************************
* Estimate TFP using the level formula rather NBER's growth formula
gen l_tfp_alt = log(rvship) - y2_sh_Pr*log(prodh) - y2_sh_NPr*log(nonprode)
                        - y2_sh_Mat*log(rmat) - y2_sh_K*log(cap)
gen tfp_alt = exp(l_tfp_a)
replace tfp97 = tfp_alt if year==1997
bys naics6 : egen tfp97_all_a = mean(tfp97)
replace tfp_a = tfp_a / tfp97_all_a

* Gen log of the NBER measure, growth rate of level measure
gen l_tfp = log(tfp)
bys naics6 (year) : gen dl_tfp_a = log(tfp_a[_n]) - log(tfp_a[_n-1])

corr tfp tfp_a if year>1960 & year!=1997
* = 0.76
corr l_tfp l_tfp_a if year>1960 & year!=1997
* = 0.43
corr dl_tfp dl_tfp_a if year>1960 & year!=1997
* =0.76
```

3 Method 2: Production Function Estimation

Rather than assume that the factor output elasticities can be calculated from cost shares, production function estimation attempts to estimate these values. The primary assumptions used for this method is that the production function is known, constant, and identical up to parameters up to some level of aggregation. The goal of production function estimation is to estimate consistent factor output elasticities, then do the TFP calculation as in the cost share case.

There are several primary econometric issues in obtaining consistent parameters: Simultaneity Bias, Selection Bias, Market Imperfections & Industry Prices, and Multiproduct Firms.

3.1 Issues in Estimation

Consider estimating $\hat{\omega}_{it} = y_{it} - \left(\hat{\beta}_K k_{it} + \hat{\beta}_L l_{it} + \hat{\beta}_M m_{it}\right)$ using OLS. The following are four fundamental issues in estimating firm level productivites.

3.1.1 Simultaneity Bias

If the firm observes ω *and then* chooses (l, k, m), then there is a simultaneity bias in $\hat{\beta}$. The expected bias of ω requires pinning down several factor correlations. In short, for a 'variable factor' (L, M) intensive firm, simultaneity leads to $\mathsf{Bias}(\omega_{it}) < 0$; for 'fixed factor' (K) intensive, $\mathsf{Bias}(\omega_{it}) > 0$.

3.1.2 Selection Bias

Consider an unbalanced panel of firms with entry and exit. Firms with initially lower ν will be more likely to exit, and positive leading to entry (survivorship). Further, if firms condition factor demand and output on having survived another period and observing ω, then we can expect correlation in (at least) capital investment decisions. An intuition is that a firm with greater ν can survive with less capital. If there is selection bias, then $\mathsf{E}[(\mu + \nu_{it} + e_{it}) \cdot k_{it}] < 0$ and so $\mathsf{Bias}(\beta_K) < 0$.

3.1.3 Market Imperfections & Industry Prices

Market imperfections in the factors or output markets can cause an omitted variable problem when using prices for the imperfect markets. This is only a problem when firm specific quantity data and/or firm specific prices are unavailable (i.e., when using industry level prices).

Real revenue is firm specific price times firm specific output, $R_{it} = P_{it} \cdot Y_{it}$. If firm level price is unavailable, then frequently industry level prices are used, $\bar{P}_{it}$. Thus, using log real revenue as the dependent variable will lead to the following estimation equation (using factor quantities):

$$r_{it} - \bar{p}_{it} = y_{it} + (p_{it} - \bar{p}_{it}) = \beta_K k_{it} + \beta_L l_{it} + \beta_M m_{it} + \omega_{it} + e_{it}$$

Clearly, if $\mathsf{E}[(p_{it} - \bar{p}_{it})x_{it}] \neq 0$, where x is a real factor of production, then bias is introduced. If real factor expenditures are used and firm prices are unobserved, then similar terms can be added to the error term.

If $\bar{p}_{it} - p_{it} > 0$, then output will be greater than average conditional on inputs, so $\mathsf{Bias}(\omega_{it}) > 0$; else, the opposite is true. A similar statement can be made about the input price and TFP bias.

3.1.4 Multiproduct Firms

If firms produce multiple products, then the production function estimation equation is misspecified. If there are multiproduct firms and single product firms producing an good, then depending on the 'synergies' of the production process TFP will be biased.

4 TFP Estimation

From the above, OLS is clearly not practically appropriate. There are, I believe it is fair to say, two strands of this literature: Panel and Semi-Parametric Proxy / Control Functions. Panel methods come in two flavors: Fixed Effect and Dynamic Panel. Control function methods differ based on factor choice timing assumptions of the firm. By far, control function methods appear to be the more prefered of the two; although, this may be because the dynamic panel methods do not seem as well known. I will skip a FE approach because it seems obvious how to implement and because it seems to perform the worst.

4.1 Dynamic Panel: Blundell & Bond (2000)

Consider the model:

$$\begin{aligned} & y_{it} = \beta_k k_{it} + \beta_l l_{it} + (\lambda_t + \eta_i + \nu_{it}) \\ & \nu_{it} = \rho \nu_{i(t-1)} + e_{it} \\ \implies & y_{it} = \beta_k k_{it} - \rho\beta_k k_{i(t-1)} + \beta_l l_{it} - \rho\beta_l l_{i(t-1)} + \rho y_{i(t-1)} + \lambda_t - \rho\lambda_{(t-1)} + (1-\rho)\eta_i + e_{it} \end{aligned}$$

The following can be estimated:

$$y_{it} = \pi_1 k_{it} + \pi_2 k_{i(t-1)} + \pi_3 l_{it} + \pi_4 l_{i(t-1)} + \pi_5 y_{i(t-1)} + \tilde{\lambda}_t + \tilde{\eta}_i + e_{it}$$

With the moment conditions, $s > 0$:

$$\begin{aligned} \mathsf{E}[\ x_{i(t-s)} \cdot (\Delta y_{it} - \pi \Delta x_{it})\ ] &= 0 \\ \mathsf{E}[\ y_{i(t-s)} \cdot (\Delta y_{it} - \pi \Delta x_{it})\ ] &= 0 \\ \mathsf{E}[\ \Delta x_{i,(t-s)} \cdot (y_{it} - \pi x_{it})\ ] &= 0 \\ \mathsf{E}[\ \Delta y_{i,(t-s)} \cdot (y_{it} - \pi x_{it})\ ] &= 0 \end{aligned}$$

If $\pi_2 = -\pi_1\pi_5$ and $\pi_4 = -\pi_2\pi_5$, then a minimum distance procedure can be used to calculate $\{\beta_K, \beta_L, \rho\}$, and then TFP can be calculated.

4.1.1 Stata

Bond has a 2002 paper explaining the method and online provides some code. I have **not** run the following code.

```
/*
id : firm identifier ; year : year
y : log sales ; n : log employment  ; k : log capital stock
y_1 : first lag of y ; yk : (y - k)
*/
tsset id year

* Using 2 year lags as instruments, year dummies as IV in level moments
xi: xtabond2 y n l.n k l.k l.y i.year , gmm(y n k, lag(2 .)) ///
     iv(i.year, equation(level)) robust h(1)

* Using 3 year lags as instruments, year dummies as IV in level moments
xi: xtabond2 y n l.n k l.k l.y i.year , gmm(y n k, lag(3 .)) ///
     iv(i.year, equation(level)) robust h(1)

* Test the parameter constraints
testnl (_b[l.y]*_b[n] = -_b[l.n]) (_b[l.y]*_b[k] = -_b[l.k])
```

Currently, I do not know how to now 'back out' the true production parameters (β) from the estimated (π), but there is / I have a Stata-ado file from a coauthor of Bond that is based on these papers (I would not be surprised if person was the RA).

4.2 Proxy / Control Function Methods

Production function estimation using control functions attempts to model the simultaneity between TFP realization and input choices. Essentially, if unobserved TFP evolves based on specific factor input choices via a markov process, then flexible functions of those specific inputs can be used to control for the simultaneity bias in the coefficients of the *other* inputs. With some consistent estimates of some coefficients and econometric structure, estimates for the remaining inputs can be recovered.

Economic (and thus econometric) choices abouts the timing of input choices and proxy function specificaton constitute the primary differences between the Olley & Pakes (Ecma 1996) – OP; Levinsohn & Petrin (REStud 2003) – LP; Ackerberg, Caves, & Frazer (WP2006, Ecma2015) – ACF; Wooldridge (EL 2009) – Wld.

OP use investment as a proxy; however, the nature of investment choices (can be lumpy and negative) and the method (proxy must be monotonic with unobserved TFP) means that some observations must be dropped. This inspired LP to use material ("intermediate") input choices as the proxy. However, ACF showed that indentification *crucially* depends on timing using examples. The main intuition of ACF is that if materials and labor both choosen in same period as TFP, then there is a collinearity problem. Their solution (seems to me) to be adjust timing assumptions and using labor as a state variable along with capital in proxy function; as in, K chosen in $t-1$, labor in $t-\frac{1}{2}$, and then TFP and M in t.

Woolridge has a short note on the simultaneity issue in LP vs ACF and additionally shows that all three can be recast as a single step (system) GMM problem rather than multi-stage regressions; this allows for optimal weighting matricies and standard errors.[4]

Newer Critques / Issues:

- Collard-Wexler & De Loecker (NBER 2016) include the issue of measurement error in capital.
- Gandhi, Navarro, Rivers (R&R JPE 2017) show that depending on how 'flexible' material inputs are there are still identification issues, and propose how to use other structural moments (FOCs) in estimation.

In the next three subsections, I will detail the OP method, the Wooldridge method, and the stata command `prodest` which can implement OP, LP, ACF, and Wld.

4.3 Olley-Pakes (1996)

Consider the model (materials are just like labor):

$$\begin{aligned} y_{it} &= \beta_k k_{it} + \beta_l l_{it} + \omega_{it} + e_{it} && \text{Prod Function} \\ p(\omega_{i(t+1)} \mid \mathcal{I}_{it}) &= p(\omega_{i(t+1)} \mid \omega_{it}) && \text{Markov TFP evolution} \\ k_{it} &= (1-\delta)k_{i(t-1)} + i_{i(t-1)} && \text{Capital evolution} \\ i_{it} &= I_t(k_{it}, \omega_{it}) && \text{Investment policy func} \end{aligned}$$

If the investment policy is monotonic with TFP, then this can be inverted:

$$\omega_{it} = I_t^{-1}(k_{it}, i_{it})$$

[4] The other three all use bootstrap SE's.

Substitute this into the production function:

$$y_{it} = \beta_k k_{it} + \beta_l l_{it} + I_t^{-1}(k_{it}, i_{it}) + e_{it}$$

Note, that we cannot estimate β_k but it is possible to consistently estimate β_l. Using a flexible function, f, estimate:

$$y_{it} = \beta_l l_{it} + f_t(k_{it}, i_{it}) + e_{it}$$

This yields $\hat{\beta}_l$ and $\hat{f}_{it}$. Note that $\hat{f}_{it} = \beta_k k_{it} + \omega_{it}$. Given the markov assumption:

$$\begin{aligned} \omega_{it} &= \mathsf{E}[\ \omega_{it} \mid \omega_{i(t-1)}\ ] + \xi_{it} = g(\omega_{i(t-1)}) + \xi_{it} \\ \mathsf{E}[\ \xi_{it} \mid k_{it}\ ] &= 0 \end{aligned}$$

We can form the above moment using the estimate of β_l and using NLS estimate β_k:

$$\begin{aligned} y_{it} - \beta_l l_{it} &= \beta_k k_{it} + g(\omega_{i(t-1)}) + \xi_{it} + e_{it} \\ y_{it} - \beta_l l_{it} &= \beta_k k_{it} + g\left(f_{i(t-1)} - \beta_k k_{i(t-1)}\right) + \xi_{it} + e_{it} \end{aligned}$$

An intuition about what helps this problem is that (by assumption) current investment, i_{it}, does not directly affect present production. Part of what causes issues in the LP and ACF 'debate' is that using current material demand as a proxy is problematic because it *does* affect current production.

4.3.1 Selection

OP also include a mechanism to deal with selection (survivorship) in estimation. This method assumes that shut down is permanent. Calculate d_{it} as an indicator for not-shutting-down / existing in period t based on the realizations of ω and ξ. Estimate the probability of existing, P_{it}, using the following model:

$$d_{it} = 1[\xi_{it} \leq \omega^{\star}(k_{it}) - \rho(f_{i(t-1)} - \beta_k k_{i(t-1)})] = h(k_{it}, f_{i(t-1)}, k_{i(t-1)})$$

Now, include $\hat{P}_{it}$ in the second stage (flexible) regression when estimating β_k.

4.3.2 OP in Stata

Included at end due to length.

4.4 Wooldridge

The intuition of Wooldridge's method is that we have the same dependent variable but two different momemnts / instruments.

The production function identities are

$$\begin{pmatrix} y_{it} \\ y_{it} \end{pmatrix} = \begin{bmatrix} \sigma + \beta_l l_{it} + \beta_k k_{it} + k(k_{it}, m_{it})\lambda_1 + e_{it} \\ \theta + \beta_l l_{it} + \beta_k k_{it} + k(k_{i(t-1)}, m_{i(t-1)})\lambda_1 + e_{it} \end{bmatrix} \tag{1}$$

Residuals are:

$$\begin{pmatrix} r_{1,it} \\ r_{2,it} \end{pmatrix} = \begin{bmatrix} y_{it} - (\sigma + \beta_l l_{it} + \beta_k k_{it} + k(k_{it}, m_{it})\lambda_1) \\ y_{it} - \big(\theta + \beta_l l_{it} + \beta_k k_{it} + k(k_{i(t-1)}, m_{i(t-1)})\lambda_1\big) \end{bmatrix} \tag{2}$$

Instruments are:

$$\begin{pmatrix} z_{1,it} \\ z_{2,it} \end{pmatrix} = \begin{bmatrix} (1, l_{it}, k_{it}, k(k_{it}, m_{it})) \\ \big(1, l_{it}, k_{i(t-1)}, k(k_{i(t-1)}, m_{i(t-1)})\big) \end{bmatrix} \tag{3}$$

And the moments are:

$$\mathsf{E}[\ Z'_{it} R_{it}\ ] = 0$$

Note: $k()$ is implemented as polynomial function of its arguments that is linear in parameters.

In this framework, it is possible to include dynamic panel instruments similar to the Bundell & Bond (2000) method by including more lags. The creators of `prodest`, Mollisi & Rovigatti, allow for this in their Stata function, discussed next.

4.5 Stata: prodest

```
*Prodest
prodest depvar [if exp] [in range] ,    ///
 free(varlist) proxy(varlist) state(varlist) method(name)   ///
 [valueadded control(varlist) acf    ///
 id(varname) t(varname) reps(#) level(#) poly(#) seed(#)    ///
 fsresidual(newname ) endogenous(varlist ) opt options]

 * Predict log-residuals = TFP
predict [type] newvarname [if exp] [in range], residuals
```

- free(varlist) : free variable(s). Ln(labour) in OP, LP and ACF.
- state(varlist) : state variable(s). Ln(capital) in OP, LP and ACF.
- proxy(varlist) : proxy variable(s). Ln(investment) in OP, ln(intermediate inputs) in LP and ACF.
- control(varlist) : control variable(s) to be included
- endogenous(varlist) : endogenous variable(s) to be included
- acf : applies the Ackerberg et al. (2015) correction
- valueadded : indicates that depvar is output value added. Default is gross output
- attrition : correct for attrition - i.e. firm exit - in the data
- method : methodology to be used: op (Olley-Pakes), lp (Levinsohn-Petrin), wrdg (Wooldridge) or mr (Mollisi-Rovigatti)
- id(varname) : specifies the panelvar to which the unit belongs. The user can either specify id() or xtset panelvar timevar before launching the command.
- t(varname) : specifies the timevar of the observation. The user can either specify t() or xtset panelvar timevar before launching the command.
- reps(#) : number of bootstrap repetitions
- poly(#) : degree of polynomial approximation for the first stage
- seed(#) : seed to be set before estimation
- fsresiduals(newvarname) : store the first stage residuals (OP and LP only) in newvarname
- translog : use a translog production function for estimation
- level(#) : specifies the confidence level Îś
- optimizer : available optimizers are Nelder Mead (nm), modified Newton-Raphson (nr), Davidon-Fletcher-Powell (dfp), Broyden-Fletcher-Goldfarb-Shanno (bfgs) and Berndt-Hall-Hall-Hausman (bhhh)
- maxiter(#) : maximum number of iterations, default is 10,000
- evaluator(name) : evaluator type
- tolerance sets the tolerance in optimization algorithm

```
* Example:
*   LP method, using water and electricity to proxy, value-added measures,
*    and 3rd order first stage function
prodest lnva, free(lnb lnw) proxy(wat ele) state(lnk) poly(3) ///
   met(lp) valueadded reps(50)
```

5 OP Code

```
/*
Program by Eric Verhoogen, Fall 2014.
Adapted by C.Luke Watson, Summer 2018
*/

#delimit;
set more off;
version 8.0;
set logtype text;
set linesize 160;
set matsize 800;
capture log close;
log using ~/dir/OP_tfp.log, replace;
clear;
set memory 512m;

*** define non-linear program, setting P=1;
capture program drop nlop;
program define nlop;
version 8;
if "‘1’"=="?" {
global S_2 "non-linear estimation, assuming p=1";
global S_1 "B0 BK B1 B2 B3";
** Approximate initial values by regression of yminusbl
on lncapital and polynomial in phihat_lag;
tempvar Y X1 X2 X3;
quietly {;
* use OLS to get starting values;
gen ‘Y’ = ‘e(depvar)’ if e(sample);
gen ‘X1’=‘3’ if e(sample);
gen ‘X2’=‘3’^2 if e(sample);
gen ‘X3’=‘3’^3 if e(sample);
reg ‘Y’ ‘2’  if e(sample);
};
*global B0 = _b[_cons];
*global BK = _b[‘2’];
global B0 = 0;
global BK = 0;
global B1 = 0;
global B2 = 0;
global B3 = 0;
global B4 = 0;
exit;
};
replace ‘1’ = $B0 + $BK*‘2’ + $B1*(‘3’-$B0-$BK*‘4’) + $B2*(‘3’-$B0-$BK*‘4’)^2 +
$B3*(‘3’-$B0-$BK*‘4’)^3;
end;
```

```
*** define non-linear program, allowing both P and h to vary;
capture program drop nlop2;
program define nlop2;
version 8;
if "‘1’"=="?" {
global S_2 "non-linear estimation, interacting (phihat-bk*k-b0) and p";
global S_1 "B0 BK B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15";
global BK = $STARTBK;
global B0 = $STARTB0;
global B1 = $STARTB1;
global B2 = $STARTB2;
global B3 = $STARTB3;
global B4 = 0;
global B5 = 0;
global B6 = 0;
global B7 = 0;
global B8 = 0;
global B9 = 0;
global B10 = 0;
global B11 = 0;
global B12 = 0;
global B13 = 0;
global B14 = 0;
global B15 = 0;
exit;
};
replace ‘1’ = $B0 + $BK*‘2’ + $B1*(‘3’-$B0-$BK*‘5’) + $B2*(‘3’-$B0-$BK*‘5’)^2 +
$B3*(‘3’-$B0-$BK*‘5’)^3 + $B4*‘4’ + $B5*‘4’^2 + $B6*‘4’^3
+ $B7*(‘3’-$B0-$BK*‘5’)*‘4’+
$B8*((‘3’-$B0-$BK*‘5’)^2)*‘4’ + $B9*((‘3’-$B0-$BK*‘5’)^3)*‘4’
+ $B10*(‘3’-$B0-$BK*‘5’)*((‘4’)^2) +
$B11*((‘3’-$B0-$BK*‘5’)^2)*((‘4’)^2) + $B12*((‘3’-$B0-$BK*‘5’)^3)*((‘4’)^2)
+ $B13*(‘3’-$B0-$BK*‘5’)*((‘4’)^3) +
$B14*((‘3’-$B0-$BK*‘5’)^2)*((‘4’)^3) + $B15*((‘3’-$B0-$BK*‘5’)^3)*((‘4’)^3);
end;

* create new variables for white-collar and blue-collar workers together;
gen sales = domsales + exports;
gen emp = emp_wc + emp_bc;
gen hours = hours_wc + hours_bc;
gen wage = (wage_wc*hours_wc + hours_bc*wage_bc)/hours;

* create value-added variable;
gen va = sales - mat - elec - othcosts;

* Drop Problematic Data
drop if va<=0;
drop if hours<=0;
drop if capital<=0;

* create survival variable -- will be useful when running probits below;;
by id (year): gen survivenextyear = indata[_n+1] if year+1==year[_n+1];

* create indicator for whether plant ever exits;
bys id (year) : egen everexit = max(exit)

* create logged variables;
gen lnsales = log(sales);
gen lnva = log(va);
gen lnhours = log(hours);
gen lnhours_wc = log(hours_wc);
gen lnhours_bc = log(hours_bc);
gen lncapital = log(capital);
gen lnwage = log(wage);
gen lnmat = log(mat);
gen lnelec = log(elec);
gen lnothcosts = log(othcosts);
gen lninvest = log(invest);

* create lagged variables;
by id (year): gen lncapital_lag = lncapital[_n-1] if year-1==year[_n-1];
by id (year): gen lnmat_lag = lnmat[_n-1] if year-1==year[_n-1];

* create variables for flexible polynomial in lncapital and lninvest;
gen lncapital2 = lncapital^2;
gen lncapital3 = lncapital^3;
gen lninvest2 = lninvest^2;
gen lninvest3 = lninvest^3;
gen lncapital1lninvest1 = lncapital*lninvest;
gen lncapital2lninvest1 = lncapital2*lninvest;
gen lncapital3lninvest1 = lncapital3*lninvest;
gen lncapital1lninvest2 = lncapital*lninvest2;
gen lncapital2lninvest2 = lncapital2*lninvest2;
gen lncapital3lninvest2 = lncapital3*lninvest2;
gen lncapital1lninvest3 = lncapital*lninvest3;
gen lncapital2lninvest3 = lncapital2*lninvest3;
gen lncapital3lninvest3 = lncapital3*lninvest3;

* create variables for flexible polynomial in lncapital and lnmat;
gen lnmat2 = lnmat^2;
gen lnmat3 = lnmat^3;
gen lncapital1lnmat1 = lncapital*lnmat;
gen lncapital2lnmat1 = lncapital2*lnmat;
gen lncapital3lnmat1 = lncapital3*lnmat;
gen lncapital1lnmat2 = lncapital*lnmat2;
gen lncapital2lnmat2 = lncapital2*lnmat2;
gen lncapital3lnmat2 = lncapital3*lnmat2;
gen lncapital1lnmat3 = lncapital*lnmat3;
gen lncapital2lnmat3 = lncapital2*lnmat3;
gen lncapital3lnmat3 = lncapital3*lnmat3;

* create variables for flexible polynomial in lncapital and lnelec;
gen lnelec2 = lnelec^2;
gen lnelec3 = lnelec^3;
gen lncapital1lnelec1 = lncapital*lnelec;
gen lncapital2lnelec1 = lncapital2*lnelec;
gen lncapital3lnelec1 = lncapital3*lnelec;
gen lncapital1lnelec2 = lncapital*lnelec2;
gen lncapital2lnelec2 = lncapital2*lnelec2;
gen lncapital3lnelec2 = lncapital3*lnelec2;
gen lncapital1lnelec3 = lncapital*lnelec3;
gen lncapital2lnelec3 = lncapital2*lnelec3;
gen lncapital3lnelec3 = lncapital3*lnelec3;

xi i.year;

save ps3_1.dta, replace;

***  Olley-Pakes;

* First Stage

reg lnva lnhours lncapital lninvest lncapital2 lncapital3 lninvest2 lninvest3
lncapital1lninvest1 lncapital2lninvest1
lncapital3lninvest1 lncapital1lninvest2 lncapital2lninvest2 lncapital3lninvest2
lncapital1lninvest3 lncapital2lninvest3 lncapital3lninvest3 if invest>0 & invest~=.;

predict lnvahat;
local blhat = _b[lnhours];
display "blhat = " ‘blhat’;
gen phihat = lnvahat-‘blhat’*lnhours if e(sample);
sort id year;
by id: gen phihat_lag = phihat[_n-1];
gen yminusbl = lnva - lnhours*‘blhat’;


* Second Stage

* assume P=1;

nl op yminusbl lncapital phihat_lag lncapital_lag if invest>0 & invest~=. & year>=2;

local bkhat_1 = $BK;
display "bkhat_1 = " ‘bkhat_1’;

gen tfp_pfix = lnva - lnhours*‘blhat’ - lncapital*‘bkhat_1’;
```

```
* Allow P to vary

* take results from above as starting values;
global STARTBK = $BK;
global STARTB0 = $B0;
global STARTB1 = $B1;
global STARTB2 = $B2;
global STARTB3 = $B3;

*** Notes below from the source code
* nl op2 yminusbl lncapital phihat_lag p_lag lncapital_lag
if invest>0 & invest~=. & year>=2, iter(50000);

* limit to 200 iterations -- use "capture" so that program will continue even though
convergence is not achieved. Can use "break" to get program to move on;

capture nl op2 yminusbl lncapital phihat_lag p_lag lncapital_lag
if invest>0 & invest~=. & year>=2, iter(200);

display "BK = " $BK;
display "B0 = " $B0;
display "B1 = " $B1;
display "B2 = " $B2;
display "B3 = " $B3;

local bkhat_2 = $BK;
display "bkhat_2 = " `bkhat_2';

gen tfp_pfree = lnva - lnhours*`blhat' - lncapital*`bkhat_2';

log close;
```