Taxbenextract: Data Extraction and Shock Simulations Based on the Tax and Benefit Model

Alexandre Desbuquois
ENS Cachan
desbuquois.alexandre@gmail.com

Abstract. The **taxbenextract** function builds upon the Tax and Benefit (TAXBEN) model developed by the OECD. This latter provides detailed information about taxes and transfers received by households whose characteristics can be specified very precisely. It thus allows to simulate potentially targeted shocks for precise individuals characterised by their wage, hours worked, marital status, age, number of kids, among others. This function allows a quick and efficient data extraction. Moreover, it adds new features and further degrees of freedom to the baseline TAXBEN model. The taxbenextract function has been built to make shock simulations easier, but also to make the data extraction process from the TAXBEN model much more efficient and systematic.

Keywords: st0001, taxbenextract, TAXBEN, taxes, benefits, shock simulation

1 Introduction

This article presents the **taxbenextract** function, which builds upon the Tax and Benefit (TAXBEN in what follows) model developed by the OECD. This latter is compatible with the latest version of the TAXBEN model uploaded by early April 2016 and with future versions provided that the structure of the model remains unchanged. It gives very detailed information at the micro level concerning taxes and transfers faced by a given individual. Among others, it provides details about the level of the Income Tax (IT), Social security Contributions paid by the employee (SC) and the employer (SSCR), Housing Benefits (HB), Family Benefits (FB), Unemployment Benefits (UB), Social Assistance (SA), and In-Work benefits (IW). This model allows to extract these policy variables with an unusual high degree of precision. The year, country, marital status, income, number of hours worked, age and number of children are among the information that can be specified to extract the level of taxes and transfers. This model is thus of particular interest, both for private households, and statisticians, economists or even politicians.

By using this function and with a simple line of code, private agents can indeed obtain information about the taxes they have to pay, but also about the benefits they can receive. In a sense, this latter can thus be a precious source of information that could for example be used by modest households to better identify the benefits they are entitled to. It could hence contribute to reinforce the efficiency of some targeted policies.

The TAXBEN model also represents an unvaluable source of information for statisticians, politicians and economists. It can indeed be used to simulate a given policy shock. The comparison of pre and post reform situations then provides a pure accounting perspective. However it does not allow to take into account behavioral responses, that might be of high interest.

The taxbenextract function has been built to facilitate such use of the model, but also to make the data extraction process way easier, efficient and systematic.

2 The TAXBEN model

The TAXBEN model has been developed by the directorate for Employment, Labour and Social Affairs of the OECD. It can be downloaded from
`http://www.oecd.org/els/soc/benefits-and-wages.htm`. The whole model has to be saved into the same folder. The user then faces two different options to use it, namely the user interface and the control file. The second option is designed for experts about the model. Consequently, the user-interface is used by the large majority of individuals.
Most of the taxbenextract function options are contained in the user-friendly user interface[1]. The most important, and certainly confusing option, concerns the choice of the run-type since the model provides seven of them. Basically, these latter implement variations over a given variable, *ceteris paribus*. The following contains a brief explanation of the purpose of each run-type.

- Run-type zero (*long term unemployment*) considers unemployed individuals. It takes variations over the number of months the agent has been in an unemployment state. By default, the number of months unemployed ranges between one and sixty. It thus allows to analyse the evolution of UB and their structure[2] over the unemployment spell.
- Run-type one (*unemployment by wage level*) takes variations over the past wage level of the currently unemployed individual. It thus allows to observe the level of UB depending on the past wage level, and consequently to appreciate UB schemes' generosity[3].
- Run-type two (*by principal hours*) takes variations over the number of hours worked by the principal. The variable associated with the number of hours worked is called *workdayp*. By default, it ranges from zero to eleven. This variable does not directly reflects the number of hours worked, and can seem counter-intuitive. We thus decided to add a variable called *Hours_p*[4] which directly reflects hours worked.
- Run-type three (*by spouse hours*) is identical to run-type two, but takes variations over the number of hours worked by the secondary earner instead of the principal. This run-type hence requires to consider married couples.
- Run-type four (*employment by Average Wage [AW] level*) implements variations over the variable representing the earnings of the principal, everything being equal. Consequently, it allows to take into account different hourly wages.
- Run-type five (*by childcare costs*) is only available after 2003, and implements variations over the *CC_Fee* variable which summarizes childcare costs.
- Run-type six (*by child age increase*) applies variations over childrens' age.

While run-types zero and one are related to a specific labour market status (unemployment), run-types five and six essentially provide information depending on the family situation. For

[1] For further information, the reader can refer to `http://www.oecd.org/els/soc/44434626.pdf`, and to the Documentation folder incorporated into the model.

[2] In the model, UB are decomposed into Unemployment Insurance (UI) and Unemployment Assistance (UA).

[3] The generosity of UB is of course a multidimensional criterion that takes into account the eligibility conditions, the potential sanctions, the level/length of UB, among others.

[4] Some of the key variables of the model contain a subscript, namely "_*p*" or "_*s*". The first one refers to the principal earner, while the second one refers to the secondary earner.

both run-types zero and one, the key variable corresponds to the unemployment benefit level. The former even provides information about the decomposition of UB between Unemployment Insurance (UI) and Unemployment Assistance (UA) over the unemployment spell. Run-types five and six provide details about the influence of childcare cost or children's age over taxes and benefits faced by households.
Run-types two, three and four lead to a key variable that summarizes both taxes and benefits, namely the Marginal Effective Tax Rate (METR). Depending on the selected run-type, this latter measures the additional taxes (or benefits if negative) that have to be paid (received) in case of an earnings' increase equivalent to one percentage point of the AW due to variations in the variable of interest (hours with run-type two or three or hourly wage with run-type four). Let us denote by $\mathcal{P} = \{SC,\ IT,\ HB,\ FB,\ SA,\ IW,\ UB\}$ the set of 'policy variables', by $\mathcal{F}$ the set representing the family situation and the individual's characteristics. It thus contains information about the marital status, the employment status of the potential partner, the individual's age, his/her number of children and their age. Let us then denote by $\mathcal{T}(w,h,\mathcal{F})$ the sum of all taxes and by $\mathcal{B}(w,h,\mathcal{F})$ the one of all benefits. These latter logically depend on $\mathcal{F}$, but also on the hourly wage (w) and on the number of hours worked (h). If we denote by $\mathcal{NT}(w,h,\mathcal{F})$ net taxes, we then have: $\mathcal{NT}(w,h,\mathcal{F}) = \mathcal{T}(w,h,\mathcal{F}) - \mathcal{B}(w,h,\mathcal{F})$.
The METR is then computed differently depending on the run-type.

- With run-types two and three:

$$METR_h(w,h,\mathcal{F}) = \frac{[\mathcal{T}(w,h+\varepsilon,\mathcal{F}) - \mathcal{B}(w,h+\varepsilon,\mathcal{F})] - [\mathcal{T}(w,h,\mathcal{F}) - \mathcal{B}(w,h,\mathcal{F})]}{\varepsilon} \tag{1}$$

  Where ε actually represents by default a one percentage point increment of the average wage for a given country in a given year.

- With run-type four:

$$METR_w(w,h,\mathcal{F}) = \frac{[\mathcal{T}(w+\varepsilon,h,\mathcal{F}) - \mathcal{B}(w+\varepsilon,h,\mathcal{F})] - [\mathcal{T}(w,h,\mathcal{F}) - \mathcal{B}(w,h,\mathcal{F})]}{\varepsilon} \tag{2}$$

Where the subscripts associated to METR just remind the '*direction*' of the derivative.
Interestingly, since the derivative is a linear application, it can be decomposed additively between different components. Let us use subscript B to refer to benefits, and T to taxes. We then have:

- With run-types two and three :

$$METR_h(w,h,\mathcal{F}) = \sum_{p_T,p_B\in\mathcal{P}} \frac{[p_T(w,h+\varepsilon,\mathcal{F}) - p_B(w,h+\varepsilon,\mathcal{F})] - [p_T(w,h,\mathcal{F}) - p_B(w,h,\mathcal{F})]}{\varepsilon} \tag{3}$$

- With run-type four:

$$METR_w(w,h,\mathcal{F}) = \sum_{p_T,p_B\in\mathcal{P}} \frac{[p_T(w+\varepsilon,h,\mathcal{F}) - p_B(w+\varepsilon,h,\mathcal{F})] - [p_T(w,h,\mathcal{F}) - p_B(w,h,\mathcal{F})]}{\varepsilon} \tag{4}$$

Where p_T (resp. p_B) refers to the taxes (benefits) contained in set $\mathcal{P}$.
After this brief overview of the main characteristics of the TAXBEN model, let us now present the taxbenextract function.

3 Taxbenextract

Before using the taxbenextract function, the TAXBEN model must be downloaded from the OECD website: `http://www.oecd.org/els/benefits-and-wages-models.htm`. All the downloaded files have to be saved into the same folder, and you will have to indicate to Stata the associated path by creating a global called *path*.

3.1 Description

Compared to the baseline model, the taxbenextract function provides additional degrees of freedom. For instance, it allows the user to choose the adult's age, but also to choose the one of each children. While the baseline model only relies on the AW or on the Average Production worker Wage (APW), the user has the possibility with the taxbenextract function to choose the reference wage among a much broader set. Indeed, in addition to the AW and APW, the user can select deciles (q10,...,q90), quartiles (q25, q50, q75), the median (q50) and the minimum wage (MW). The function even goes one step further by allowing to take into account different reference wages for each working member of the household. The wage of the principal can for instance be set at the 90^{th} percentile, while the one of the secondary earner can be fixed at the MW. In a sense, this new dimension in the model allows to capture bargaining processes within couples. The function also allows to reach considerable efficiency gains compared to the *user-interface*. It has been designed to make some potential shock simulations easier (see the input/output options).

3.2 Syntax

```
taxbenextract run_type marital_status [ , childnumber(0/1/2/3/4)
   countrylist(string) start(startyear) end(endyear) prefwage(string)
   pwage_level(#) pdays(#) sworks(0/1/2) srefwage(string) swage_level(#) sdays(#)
   tintowork(yes/no) primben(UB/SA) sa(yes/no) ntcp_ee(yes/no) ntcp_er(yes/no)
   childcare(yes/no) ccpt(0/1/2) postcci(yes/no) adage(#) cage1(#) cage2(#)
   cage3(#) cage4(#) uemonth(#) hcost(#) vname(string) isocountrycodes(yes/no)
   graph(yes/no) derivdecompos(yes/no) split(yes/no) input(input's folder name)
   output(output's folder name) save(filename) clear ]
```

3.3 Options

3.3.1 Basic options

The model contains an extremly extensive number of options, reflecting its high flexibility.

`childnumber(string)` specifies the number of kids for whom data have to be extracted. You can specify between zero (0) up to four kids (4). Compared to the user interface, the user can extract simultaneously data for different numbers of kids, for instance by specifying `child(0-4)` or `child(2-3)`.

`countrylist(string)` specifies the country/ies for which data have to be extracted. The user can specify only one country, or as many countries as the model contains. The country codes are

the one of the TAXBEN model (two digit country codes). By specifying `c(all)`, informations can be extracted for all countries for which data are available in the selected years.

`start(startyear)` is the initial year for which data will be extracted. Up to now, data availability starts in 2001 for most countries.

`end(endyear)` indicates the final year for which data will be made available. To extract data for only a given year, indicate the same starting and ending year, or only specify the starting or the final year. If you only use the `start()` (respectively the `end()`) option, the `end()` (resp. the `start()`) option will take the same value by default.

`pwage_level(#)` corresponds to the wage level of the principal. It will necessarily be expressed as a fraction of the choosen reference wage (`prefwage()`). The resulting ratio ranges between 0 and 220 (percent of the reference wage). The default value is 100, meaning that the model considers an individual earning the baseline reference wage.

`pdays(#)` lets the user decide the number of days the principal works per week. This variable ranges between zero and eleven[5]. By default, a full-time worker will be selected (value of five).

`sworks(0/1/2)` can take three different values. The secondary earner can be inactive (value of 0), active and working (value of 1), or available to work (value of 2). The default value is zero.

`swage_level(#)` gives the possibility to choose the wage level of the secondary earner only if this later is declared as active (`swo(1)`). This option is not available with the user-interface. Like for the `pwage_level()` option, values have to be between 0 and 220. Default value is zero. If the user for instance chooses `swa(213)`, the wage level of the secondary earner will represent 213 percent of the selected reference wage.

`sdays(#)` represents the number of days worked by the secondary earner. Of course this option has to take a positive value, between 0 and 11, as for the `pdays()` option. To be consistent, the secondary earner naturally has to be declared as active (`swo(1)`), and with a positive wage (`swa(`*`positive value`*`)`). The default value of this option is zero.

`tintowork(yes/no)` reflects the labour status of the individual both in t and t-1. Being unemployed in t-1 can have a fiscal incidence in t, which makes this option of particular interest to study the extensive margin of labour supply. Default of this option is no.

`primben(UB/SA)` reflects the primary source of benefits, which will of course depend on the income level. Default value is SA if the selected run-type is strictly above one, UB otherwise.

`sa(yes/no)` reflects the fact that only long term unemployed (no), or all individuals for whom relevant conditions are met (yes) receive SA. By default, individuals can receive SA if relevant

[5] A value of 11, which actually reflects eleven days of work per week, can seem unrealistic. However, we kept this level to remain in line with the baseline specification of the model. Note that 11 days of work represent around ninety hours per week, which is not legal in most of OECD countries.

conditions are met.

`ntcp_ee(yes/no)` reflects Non-Tax Compulsory Payments (NTCP) paid by the employee (ee). For further information, see:
`http://www.oecd.org/tax/tax-policy/Non-tax-compuslory-payments_2014.pdf`

`ntcp_er(yes/no)` corresponds to NTCP paid by the employer. Default value is yes for both employees and employers.

`childcare(yes/no)` allows to choose whether the household receives (yes) childcare benefits or not (no). Default value is yes if a positive number of kids is selected, zero otherwise.

`ccpt(0/1/2)` gives the choice between three different configurations, namely full time childcare (0), hypothetical part time (1) and actual part-time (2). In line with the TAXBEN model, default value is zero.

`postcci(yes/no)` permits to choose whether childcare costs will be deducted or not from the gross income. This option can be particularily useful in countries that grant tax exemption(s) due to childcare costs. Default value is no.

`adage(#)` gives the possibility to choose the adult's age. Note that in case of a married couple, both members are constrained to have the same age. This option is not available with the user interface. The default value is forty.

`cage1(#) ... cage4(#)` allows to choose children's age. This option, not available with the user interface can be particularily useful since childcare benefits usually depend on kids' age. Default values are the one of the TAXBEN model and depend on the selected run-type.

`uemonth(#)` allows to choose the length of the unemployment spell. This option, not available with the user interface, is particularily useful with run-type one. Default value is two months.

`hcost(#)` represents the renting cost, as a percentage of the average wage.

`vname(string)` gives the possibility to select only some specific variables in the extraction process. By specifying `vname(UB)`, only the unemployment benefit variable (in addition to extraction-sensitive variables) will be extracted. The vname option can be used as follows to extract the whole set of data provided by default by the TAXBEN model: `vname(details)`. This option is only available if you select one country with the `countrylist(string)` option.

`isocountrycodes(yes/no)` creates directly ISO 3 digit country codes. Default is no.

`graph(yes/no)` provides a graphical representation of some key variables depending on the selected run-type. The graphs will be stored in a *gph* folder that will automatically be created. This latter will contain a sub-folder providing information about the selected run-type that will itself contain sub-sub-folders for each of the selected countries. This option can only be used with

run-types zero to four.

`save(filename)` gives the possibility to save the extracted file. If this option is activated, the file will be saved in an output folder located in the same parent folder as all the .do files used by the function.

3.3.2 Advanced options

The following options require an advanced knowledge of the TAXBEN model. The `prefwage` and `srefwage` options actually provide a large value added to the baseline model. These options allow to take different reference wages for the principal and secondary earners, with wage levels that actually represent key points of the wage distribution in each specific country-year. By default in the TAXBEN model, all earnings are expressed with respect to the AW. The taxbenextract function allows for much more flexibility by allowing the user to choose the AW, but also deciles, quartiles, and the minimum wage. The next two options have essentially been developed to make some simulation exercises easier. Indeed, a user can decide to modify a given parameter in one of the country-year specific dofile to analyse the fiscal influence of a specific shock. The resulting file can then be saved, and compared with the original situation to appreciate the shock.

`prefwage(string)` allows the user to choose a reference wage for the principal. With this option, the user has the choice between the average wage (AW), the wage of the Average Production Worker (APW), the deciles (q10,...,q90) and quartiles (q25, q75) , the median (median or q50), but also the minimum wage (MW). The default choice is the AW.

`srefwage(string)`: same as the prefwage option, but applied to the secondary earner. It is then straightforward to consider a couple with a principal earner with a high wage (e.g. `prefw(q90)`), and a secondary earner with a comparatively low wage (`serfw(MW)`). Note that for some country-year pairs, the selected wage can be missing. For instance, Denmark has no minimum wage. Consequently, if you choose as a reference wage for the principal or the secondary earner MW, some information as the earnings of the individual, the PoAW and the associated variables will be missing.

`derivdecompos(yes/no)`: provides the decomposition of the METR presented in equations **??** and **??**.

`split(yes/no)`: offers a decomposition between principal and spouse for some 'fiscal variables' when this latter is available.

`input(input's folder name)` allows the user to call user-written files, for instance to simulate a given policy shock. The corresponding files must be saved in a folder located into the same super/parent directory as the whole TAXBEN model. For instance, let us say that the user wants to simulate a shock induced by a given parameter of UB schemes in France. The value of the corresponding parameter can be modified in France-specific dofiles. These latter can then be saved in a folder called '*France_UB_reform*', and this name has to be specified in the input(France_UB_reform) option.

`output(output's folder name)` this option permits to create a specific folder to save the results. It will be created within the output folder. This option must be used with the save option, otherwise no folder will be created and the data will not be saved. In line with the example associated with the input option, the user could specify output(Result_UBshock_France).

3.4 Remarks

The advanced options have been created in order to facilitate the simulation of some potential shocks. For instance, the user can decide to modify some parameters in the baseline dofiles located in the PARSPROG folder, and to compare the results with the original files. In such a case, the input-output options are of particular interest first to explain to the taxbenextract function where to find the original and modified files, and then to save them into well-named folders.

In addition to the provision of further degrees of freedom and new functionalities, the taxbenextract function also corrects for some inconsistencies in the user-interface. For instance, while the run-type five can only be executed between 2004 and 2008, the user-interface still provides results with this run-type in 2001. It is also possible to ask for childcare benefits, even for a couple or a single individual without kids. The function corrects this type of inconsistencies and explains to the user why the corresponding configurations cannot lead to consistent results.

This latter also automatically updates. This means that if new countries and/or years are to be added, the function will identify the corresponding years/countries, and will allow the user to obtain the associated information (provided that this latter downloads the corresponding version of the model). Consequently, it does not need to be updated year after year. This automatic update explains why the first run of the model may be relatively more time-consuming.

Some of the aforementioned options are incompatible. For instance, if the user chooses run-type four, and a reference wage level for the principal corresponding to the 90^{th} percentile of the wage distribution, this latter option will be ignored. Indeed, this would lead the wage level of the principal to exceed 220% of AW, a configuration in which the model does not apply anymore. If the user selects run-type 2 with a `pdays` option of 11, this latter option will also be ignored.

Note finally that some combinations of the options can actually lead to unrealistic situations. Imagine for instance that the user chooses an individual working full time at the MW (`prefw(MW)`). Then, using the `pwage_level` option with a value lower than 100 would lead to an individual working at a wage lower than the AW, which may not be possible in some countries. However, this can be seen as a way to capture some activities that may be undertaken informally.

3.5 Some basic examples

Let us first consider a straightforward example. Imagine a 34 years old Hungarian agent wants to know which benefits she/he will receive depending on her/his children's age. This individual is married, with a secondary earner that does not work. She/he earns the minimum wage, works full time, has 2 children of respectively 6 and 4 years old and receives childcare benefits. For her/him to know the evolution of the benefit level she/he will receive as her/his children become older, the taxbenextract function has to be used as follows[6]

[6]Note that before using the function, you will be asked to indicate to stata where the TAXBEN model has been saved through the creation of a global called path. The first use of the function can take some time since it will identify all available countries and the years for which data are available for each of them.

taxbenextract rtype6 married, c(hu) ch(2) s(2003) prefwage(MW) childcare(yes) adage(34) cage1(6) cage2(4) clear

Now, let us consider an agent that would like to work a higher number of hours per week. This individual wants to know if working more would allow her/him to increase her/his net wage significantly, meaning that she/he wants to assess the additional tax she/he may pay when earning more. The run-type to use to answer this type of question is the second one. Since such configuration can be of interest for a broad public, one can extract the associated information for all countries and family situations. This type of extraction makes the function of particular interest for policy makers. Let us take year 2010 for this example. The function has to be used as follows:

```
foreach marital_status in "married" "single"{
    taxbenextract rtype2 'marital_status', c(all) s(2010) child(0-4)
}
```

This function could also be used by a very specific individual. Consider a French household in which both members are working. The principal earner works full-time (5 days per week) with a very high wage corresponding to the 90^{th} percentile of the wage distribution. The secondary earner works 2 days with a potentially varing wage level. This household is also composed of two kids of 2 and 5 years old. In this specific configuration, individuals could question how an increase in the hourly wage earned by one of the member of the couple might affect their taxes and benefits. In this case, the command line would be:

taxbenextract rtype4 married, c(fr) s(2010) e(2013) pdays(5) prefw(q90) swo(1) sdays(2) srefw(MW) ch(2) cage1(2) cage2(5) clear

This function also makes possible an analysis of the extensive margin of labour supply. To make things clear, let us take a very basic example. Consider an Spanish couple with no kids in 2006. Consider first that the secondary earner is inactive, while the principal works full time at the AW. Let us say that this couple wants to increase its overall income. It then faces a basic trade off. The principal can work more at a given hourly wage, or the secondary earner can decide to work. To compare these two configurations, the function can be used as follows:

```
* Consider first a case in which the principal works more
taxbenextract rtype2 married, c(sp) s(2006) ch(0)
* Now consider transition from unemployment to employment
taxbenextract rtype2 married, c(sp) s(2006) ch(0) tintow(yes) swo(1) srefw(MW)
sort PoAW
```

The tax rates in each configuration can now be compared easily.

An economist could also desire to compare the generosity of UB schemes for single individuals over the unemployment spell across different countries (let us say France, the UK, Germany and Spain) for past top-earners. In this case, the taxbenextract function can be used as follows:

taxbenextract rtype0 single, s(2006) e(2007) c(fr uk ge sp) ch(3) childcare(yes) primben(UB) sa(yes) prefwage(q90) gr(yes) clear

The option graph, once activated, even allows to obtain graphical representation of UB's decomposition and evolution over the unemployment spell.

Let us now introduce one method that can be used in order to simulate the effect of a given shock. Imagine for instance than one wonders what would be the effect of a decrease in the level of UB. This is actually a question of high interest. The French government for instance announced in the past months a potential drop of the benefit level for some unemployed individuals with also a potential drop in benefits' duration. Sweden and Belgium recently modified the UB rules and decreased their level and/or duration.
Let us take the french case in 2013. To simulate a drop in UB level, the user can for instance modify the FRPARM13.do file that is located on the PARSPROG folder. Parameters concerning UB are located at the beginning of this file. By changing the values of the variable *ub_rate*, or those of *max_dur*, the user can, in a puely accounting perspective, simulate the effect(s) of the aforementioned reforms. The corresponding file then has to be saved in another folder, with exactly the same name. Importantly, even if the FRPROG13.do is not modified, it also has to be saved, with the same name, in the same folder as the modified FRPARM13.do. The new folder that will be created has to be located in the same parent directory as the whole model. It could for instance be named Reform_UB_FR.
The function can then be used as follows:

** Generate the initial situation*
taxbenextract rtype0 single, c(fr) s(2013) ch(1) primben(UB)
** Simulate the shock*
taxbenextract rtype0 single, c(fr) s(2013) ch(1) primben(UB) input(Reform_UB_FR) output(Effect_UBreform_FR) save(reform_fr_2013)

The fiscal influence of the UB scheme change can then easily be analysed.

4 Conclusion

Based on the highly flexible and unusually rich TAXBEN model, the taxbenextract function makes the data extraction process much more efficient. A simple line of code can be used to extract all available information between 2001 and 2013, for all available countries and different family situations.

This function also corrects some of the inconsistencies in the user-interface, and adds degrees of freedom to the baseline model. With this function, reference wages can even be made heterogeneous within a given household, which adds a considerable value added to the TAXBEN model.

Finally, the function has been designed to make the simulation of some chocks straightforward. In that sense, the augmented version of the TAXBEN model is very useful to appreciate, from a pure accounting perspective, the fiscal effect of some reform projects.

5 Acknowledgments

I dearly thank Sean Gibson for his numerous responses regarding specific questions about details of the TAXBEN model. His answers have been precious in my understanding of the model.

About the authors

Alexandre Desbuquois is a student in economics at the École Normale Suprieure de Cachan (Paris), with a master's degree in economics from the Paris School of Economics. He developed the taxbenextract function after a six months internship at the OECD.
He is currently working as research assistant at the London School of Economics for Professor C. Landais and Dr Johannes Spinnewijn.