

EUROMOD Functions & Parameters I

Olga Rastrigina

UNU-WIDER Training course

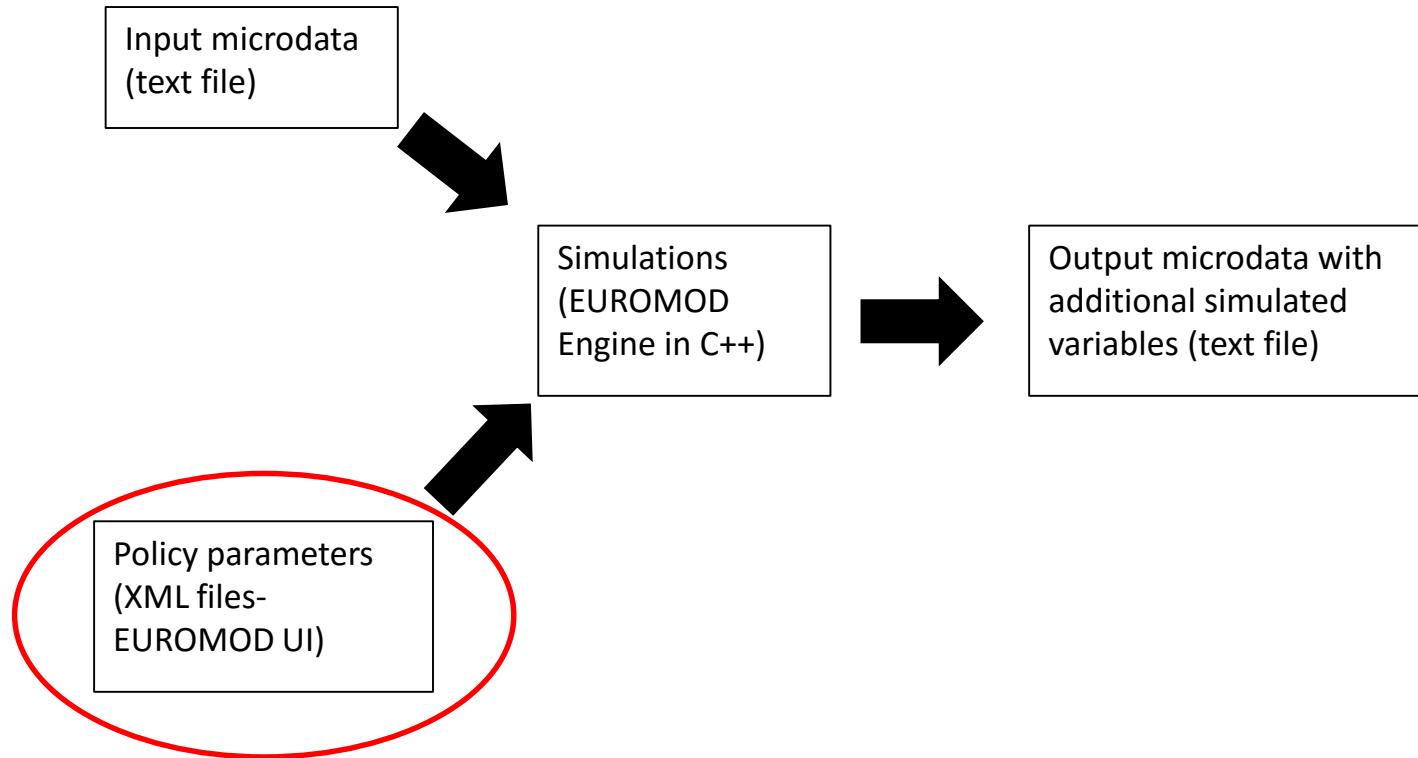
29 February – 4 March 2016, Johannesburg, SOUTH AFRICA



UNITED NATIONS
UNIVERSITY
UNU-WIDER



Structure



EUROMOD functions



EUROMOD functions

Building blocks to implement policies

- Stored in XML
- Manipulated via the User Interface
- Calculations in EUROMOD executable (C++ code)

EUROMOD language

- Flexible
- Harmonised
- Parameterised
- Sufficient
- Transparent and documented (In-built HELP)



UNITED NATIONS
UNIVERSITY
UNU-WIDER



Structure of a function

Switch: **on** / **off** / n/a / **toggle** / **switch**

\mathcal{f}_x ArithOp		on	benefit calculation
formula		nDepChildrenInTU*amount#1	
#_amount	1	200#m	
output_...		bch_s	
TAX_U...		tu_sben_family_sl	There are age limits set for dep. children in TU.

Function name

Parameter names

Parameter values

Type of functions

System functions

- functions used to define some general settings
- E.g. uprating, default values for datasets, etc.

Policy functions

- functions used to implement tax-benefit policies
- E.g. define eligibility, calculate amount, etc.

Special functions

- more advanced functions that perform more complicated tasks
- E.g. loops, change parameters at run-time, etc.
- (not covered in this course)



Manipulating functions

	Function	Parameter	Description
6	sic_sl	on	SIC: Social Insurance Contributions
6.1	SchedCalc	on	employer's contribution
6.1.1	base		
6.1.2	band		
6.1.3	uplim		i.e. switched off
6.1.4	outp		
6.1.5	TAX		
6.2	SchedC		employee's contribution
6.2.1	base		
6.2.2	band		
6.2.3	uplim		i.e. switched off
6.2.4	outp		
6.2.5	TAX		
6.3	SchedC		self-employed contribution
6.3.1	base		
6.3.2	band		
6.3.3	uplim	325#m	
6.3.4	output_...	tscse_s	
6.3.5	TAX_U...	tu_individual_sl	
7

right-click on function or parameter name to activate menu

EUROMOD policies



Policies (I)

Policy is a **block of functions** that complete a policy simulation

It can be manipulated independently:

- switch affects all functions in the policy
- can be copied / moved

Policy names (usually) end with the country acronym

Policies can have any name ...but in practice we use naming conventions

The **comment** explains what the policy intends to simulate

Policies can be:

- common to all countries (ex: defining output)
- country specific (ex: means-tested child benefit)

Order of policies is called the '**spine**'



Policies (II)

policy name

policy switch

policy description

6	▼ sic_sl	on	SIC: Social Insurance Contributions
6.1	▼ SchedCalc	on	employer's contribution
6.1.1	base	yem	
6.1.2	band_rate	1 0.1	
6.1.3	uplim	999999999.99#m	i.e. switched off
6.1.4	output_...	tscer_s	
6.1.5	TAX_U...	tu_individual_sl	
6.2	▼ SchedCalc	on	employee's contribution
6.2.1	base	yem	
6.2.2	band_rate	1 0.05	
6.2.3	uplim	999999999.99#m	i.e. switched off
6.2.4	output_...	tscee_s	
6.2.5	TAX_U...	tu_individual_sl	
6.3	▼ SchedCalc	on	self-employed contribution
6.3.1	base	yse	
6.3.2	band_rate	1 0.13	
6.3.3	uplim	325#m	
6.3.4	output_...	tscse_s	
6.3.5	TAX_U...	tu_individual_sl	

policy made up of 3 functions

Manipulating policies

Simpleland - EUROMOD G1.0 (D:\Home\savram\EUROMOD\EuromodFiles_G1.0\)

Run EUROMOD

Countries

Country Tools Administration Tools Add-Ons Applications Help & Info

Simpleland loaded

AT BE BG CY CZ DE DK EE EL ES FI FR HU IE IT LT LU LV MT NL PL PT RO SE SI SK SL

	Policy	Gr...	SL_demo	Comment
1	Uprate_sl		on	DEF: UPDATING FACTORS
2	ILDef_sl		on	DEF: INCOME CONCEPTS
3	TUDef_sl		on	DEF: ASSESSMENT UNITS
4	yem_sl		on	DEF: Minimum Wage
5	yse_sl		on	DEF: recode negative self-employment income to zero
6	SIC_sl		on	SIC: Social Insurance Contributions
6.1	SIC_sl			employer's contribution
6.1.1				
6.1.2				
6.1.3				i.e. switched off
6.1.4				
6.1.5				
6.2	SIC_sl			employee's contribution
6.2.1				
6.2.2				i.e. switched off
6.2.3				
6.2.4				
6.2.5				
6.3	SIC_sl			self-employed contribution
6.3.1				
6.3.2				
6.3.3				
6.3.4				
6.3.5				
7	bch_sl		on	BEN: Child Benefit
8	tin_sl		on	TAX: Income Tax

right-click on policy name to activate menu

EUROMOD functions: Elig & ArithOp



Function Elig

- Used to implement **conditions**
- Condition is defined in parameter **elig_cond**
- Creates a variable which is equal to 0 or 1 (by default **sel_s**)
- Subsequent functions use this information via parameter **who_must_be_elig**

7.3	Elig		on	Eligibility for employee pension insurance contribution
7.3.1	elig_cond		{dag >= \$tscag18} & {dag <= \$tscag67} & {il_tsceebbs >= \$tsceeeex}	Age between 18 - 67 and contribution base above limit
7.3.2	TAX_UNIT		tu_individual_fi	
7.4	BenCalc		on	Occupational pension contribution
7.4.1	who_must_be_elig		one	
7.4.2	comp_cond	1	{dag < \$tscpiag}	Pension contribution below age limit
7.4.3	comp_perTU	1	il_tsceebbs * \$tsceepirt1	
7.4.4	comp_cond	2	{dag >= \$tscpiag}	Pension contribution at or above age limit
7.4.5	comp_perTU	2	il_tsceebbs * \$tsceepirt2	
7.4.6	result_var		tsceepi_s	Employee pension contribution
7.4.7	output_var		tscee_s	Employee social contributions
7.4.8	TAX_UNIT		tu_individual_fi	

Function ArithOp

- Arithmetical **calculator**
- The parameter **formula** contains some calculations
- The result is stored as output variable in parameter **output_var**

12	tscer_uk		on	SIC: employer national insurance contribution
13	bunct_uk		on	BEN: contributory unemployment benefit (Job Seekers Allowance)
13.1	Elig		on	Eligibility conditions taken from the data plus check of all conditions being satisfied
13.2	Elig		off	Eligibility conditions fully simulated
13.3	ArithOp		on	Base for disregards for Pensions and Earnings
13.3.1	formula		(0 <max> (ypp + boactcm - amount#1)) + (0 <max> (yem + yse - amount#2))	
13.3.2	#_amount	1	\$JSAIncdisPen	
13.3.3	#_amount	2	\$JSAIncdisEarn	
13.3.4	output_var		sin01_s	
13.3.5	TAX_UNIT		tu_individual_uk	
13.4	BenCalc		on	Amount

Parameters



Parameters (I)

Can be:

- Common to several functions / Specific to one function
- Compulsory / Optional

Order of parameters in a function is not important

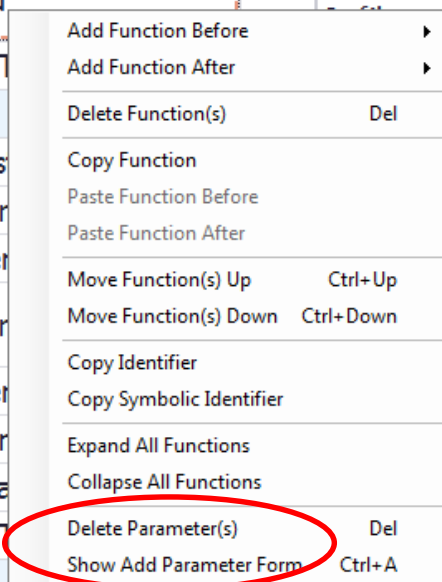
- but order of functions in a policy is!!

Manipulated via context menu (right click)



Parameters (II)

7.3.1	elig_cond	{dag >= \$tscag18} & {dag <= \$tscag67}	insurance contribution
7.3.2	TAX_UNIT	ubs >= \$tsceex}	Age between 18 - 67 and contribution base above limit
7.4	BenCalc		Occupational pension contribution
7.4.1	who_must		
7.4.2	comp_cor	\$tsdpiag}	Pension contribution below age limit
7.4.3	comp_per	* \$tsceepirt1	
7.4.4	comp_cor	\$tsdpiag}	Pension contribution at or above age limit
7.4.5	comp_per	* \$tsceepirt2	
7.4.6	result_var		Employee pension contribution
7.4.7	output_var		Employee social contributions
7.4.8	TAX_UNIT		
7.5	BenCalc		Unemployment contribution
7.6	BenCalc	on	Sickness daily allowance contribution



Parameters (III)


Add Parameters

Elig (order: 3) in policy tscee_fi

Add	Parameter	Replaces	Grp/No	Count	Description
<input type="checkbox"/>	Output_Var				Variable for storing the result of the function. Result of function overwrites the current value of th...
<input type="checkbox"/>	Result_Var				Variable for storing the result of the function. Result of function overwrites the current value of th...
<input type="checkbox"/>	Who_Must_Be_Elig				Function's calculations are carried out if ...- one (one_member): ... one member of the assessme...
<input type="checkbox"/>	Elig_Var				Variable indicating whether a person is "eligible" (see parameter Who_Must_Be_Elig):- zero: per...
<input type="checkbox"/>	Run_Cond				Function is only carried out if the condition is fulfilled. The parameter is intended to be a conditio...
<input type="checkbox"/>	#_LowLim		1	1	Footnote parameter for the further specification of an operand: replaces operand if operand is s...
<input type="checkbox"/>	#_UpLim		1	1	Footnote parameter for the further specification of an operand: replaces operand if operand is hi...
<input type="checkbox"/>	#_Amount		1	1	Footnote parameter for the further specification of an operand: indicates the numeric value of an...
<input type="checkbox"/>	#_Level		1	1	Footnote parameter for the further specification of an operand: indicates an alternative assessm...
<input type="checkbox"/>	#_AgeMin		1	1	Parameter of several queries (e.g. nDepChildrenInTu).
<input type="checkbox"/>	#_DataBasename		1	1	Parameter of query IsUsedDatabase.
<input type="checkbox"/>	#_AgeMax		1	1	Parameter of several queries (e.g. nDepChildrenInTu).
<input type="checkbox"/>	#_N		1	1	Parameter of query IsNtoMchild.
<input type="checkbox"/>	#_M		1	1	Parameter of query IsNtoMchild.
<input type="checkbox"/>	#_Val		1	1	Parameter of query HasMaxValInTu.
<input type="checkbox"/>	#_Income		1	1	Parameter of several queries (e.g. GetPartnerIncome).
<input type="checkbox"/>	#_Info		1	1	Parameter of several queries (e.g. GetPartnerInfo).
<input type="checkbox"/>	#_Unique		1	1	Parameter of query HasMaxValInTu.
<input type="checkbox"/>	#_Adults_Only		1	1	Parameter of query HasMaxValInTu.

☒ Show Common Parameters
☒ Show Footnote Parameters

☒ Description (F5) ☒ Summary (F6)

+ Add  Close

Common **output** parameters

- **Output_var, output_add_var**
- Either output_var or output_add_var must be indicated!
- Exception: func_Elig → sel_s

16	• tscee_el	on	SIC: employee social insurance contributions - private sector (IKA)
16.1	▸ fx Elig	on	Eligibility
16.2	▸ fx BenCalc	on	Contribution base: upper earnings threshold
16.3	▸ fx ArithOp	on	Contribution base
16.4	▸ fx ArithOp	on	Primary pension
16.4.1	formula	sin02_s * 6.67%	
16.4.2	output_var	tsceepi_s	
16.4.3	TAX_UNIT	tu_individual_el	
16.5	▸ fx BenCalc	on	Primary pension: extra for hazardous occupations
16.6	▸ fx ArithOp	on	Supplementary pension
16.6.1	formula	sin02_s * 3%	
16.6.2	output_add_var	tsceepi_s	(add) - same variable used
16.6.3	TAX_UNIT	tu_individual_el	
16.7	▸ fx BenCalc	on	Supplementary pension: extra for hazardous occupations
16.8	▸ fx ArithOp	on	Sickness insurance

Common parameter **TAX_UNIT**

TAX_UNIT defines the **assessment unit** a function refers to:

- Individual (*tu_individual_xx*)
- Household (*tu_household_xx*)
- Various definitions of family units (various names, e.g. *tu_family_xx*)

Compulsory for most policy functions!

7	• bch_sl		on	BEN: Child Benefit
7.1	ArithOp		on	benefit calculation
7.1.1	formula		nDepChildrenInTU* unt#1	
7.1.2	#_amount	1	200#m	
7.1.3	output_var		bch_s	
7.1.4	TAX_UNIT		tu_sben_family_sl	There are age limits s
7.2	Elig		n/a	

Common **eligibility** parameters (I)

who_must_be_elig: calculations of the function are carried out if...

- **one**: one member of the assessment unit is eligible
- **one_adult**: one adult member of the assessment unit is eligible
- **all**: all members of the assessment unit are eligible
- **all_adults**: all adult members of the assessment unit are eligible
- **nobody**: calculations are carried out for each assessment unit (default)

By default eligibility is determined by the variable **sel_s**
(can use other variable in parameter **elig_var**)

- 0: person is not eligible
- 1: person is eligible

Common **eligibility** parameters (II)

12.2	✶ Elig		on	by employees
12.2.1	elig_cond		{lse_s=1}	
12.2.2	TAX_UNIT		tu_individual_ro	
12.3	✶ ArithOp		on	Health insurance contributions paid by the self-employed on income other than intelctual property
12.3.1	who_must_be_elig		one	
12.3.2	formula		0.055*ysetx_s#1	Note: health insurance contribution due on self-employment income even when employed
12.3.3	#_lowlim	1	\$MinWage	Note: lower limit on which contributions payable
12.3.4	lowlim		0	Make sure not negative
12.3.5	result_var		tscsehl_s	note:no health insurance due on intelctual property income unless sole source of income ; (simulated ir contributed hic)
12.3.6	output_add_var		tschl_s	
12.3.7	TAX_UNIT		tu_individual_ro	

Common eligibility parameters (III)

6	• sic_sl		on	SIC: Soci
7	• bch_sl		on	BEN: Chil
7.1	fx ArithOp		on	benefit cal
7.2	fx Elig		n/a	
7.2.1	Elig_Cond		{dag>=80}	
7.2.2	TAX_UNIT		tu_household_sl	
8	• tin_sl		on	TAX: Inco
9	• bsa_sl		on	BEN: Soci
9.1	fx Elig		off	eligibility p

				who_must_be_elig				
idhh	idperson	dag	sel_s	one	one_adult	all	all_adults	nobody
1	11	80	?	?	?	?	?	?
1	12	60	?	?	?	?	?	?
1	13	40	?	?	?	?	?	?
2	21	80	?	?	?	?	?	?
2	22	6	?	?	?	?	?	?
3	31	80	?	?	?	?	?	?
4	41	40	?	?	?	?	?	?
4	42	40	?	?	?	?	?	?

Common eligibility parameters (III)

6	• sic_sl		on	SIC: Soci
7	• bch_sl		on	BEN: Chil
7.1	• ArithOp		on	benefit cal
7.2	• Elig		n/a	
7.2.1	Elig_Cond		{dag>=80}	
7.2.2	TAX_UNIT		tu_household_sl	
8	• tin_sl		on	TAX: Inco
9	• bsa_sl		on	BEN: Soc
9.1	• Elig		off	eligibility p

				who_must_be_elig				
idhh	idperson	dag	sel_s	one	one_adult	all	all_adults	nobody
1	11	80	1	1	1	0	0	1
1	12	60	0	1	1	0	0	1
1	13	40	0	1	1	0	0	1
2	21	80	1	1	1	0	1	1
2	22	6	0	1	1	0	1	1
3	31	80	1	1	1	1	1	1
4	41	40	0	0	0	0	0	1
4	42	40	0	0	0	0	0	1

Common **limiting** parameters

- Lowlim (lower limit)
- Uplim (upper limit)

6.4	ArithOp	on
6.4.1	Formula	yem*0.2
6.4.2	LowLim	50#m
6.4.3	UpLim	200#m
6.4.4	Output_Var	sin01_s
6.4.5	TAX_UNIT	tu_individual_sl

Linking functions



Interactions between functions (I)

The functions interact in three ways:

1. Input:

One function calculates a variable

Subsequent function uses it

ArithOp		on	Tax base
formula		il_tinnaty - il_tinnata	Tax base
lowlim		0	
output_var		i_tinnatb	Tax base
TAX_UNIT		tu_individual_fi	
BenCalc		on	Tax schedule: amount for first band
comp_cond	1	{i_tinnatb > \$tinnatslt01}	
comp_perTU	1	\$tinnatsam01	
output_var		i_tinnagt	State tax: gross tax (i.e., before tax credits)
TAX_UNIT		tu_individual_fi	

Interactions between functions (II)

2. Condition:

One function evaluates a condition

Subsequent function uses it

fx Elig	on	Liable to pay SIC
elig_cond	{il_tscee_base>0}	
TAX_UNIT	tu_individual_ro	
fx SchedCalc	on	
who_must_be_elig	one	
base	(il_tscee_base*12)/yemmy	Adjust base for number of months worked
band_rate	1 0.095	
band_uplim	1 5*\$AvGrossWage	"Upper ceiling 2007, 2011, 2012"
band_rate	2 0	
output_var	tsceepi_s	
TAX_UNIT	tu_individual_ro	

Interactions between functions (III)

3. Addition:

One function calculates a part of a policy

Subsequent function calculates another part

And both parts are added up

✚ SchedCalc	on	Class A: contribution (per month in work)
who_must_be_...	one	
base	int_tscee_base	contribution base
band_rate	1 0.085	8.5% ..
band_uplim	1 356#w	on incomes up to €356
band_rate	2 0.1075	10.75% if income exceeds that limit
simple_prog	yes	i.e. apply relevant rate on the whole income
output_var	tscer_s	
TAX_UNIT	tu_individual_ie	
✚ ArithOp	on	Class A: contribution (average per month)
✚ Elig	on	Class J: eligibility
✚ ArithOp	on	Class J: contribution
who_must_be_...	one	
formula	il_tscee_base * 0.5%	(here unadjusted as proportional anyway)
output_add_var	tscer_s	(add)
TAX_UNIT	tu_individual_ie	

Allocation of result of a function

It is always assigned to the **head of the assessment unit**

For all **other members** of the unit output variable is zero

