NICE 2020 model — code documentation

T. O. Da Costa

Contents

1 Model - Indexes

Symbol Name	Model Name	Short description
c	`country`	Countries included in the model
q	`quantile`	Index of income quantiles
$rrice$	`regionrice`	Regions defined by the RICE model
$rwpp$	`regionwpp`	Regions defined from the World Population Prospects database (UN)
t	`time`	Time index

2 Model - Parameters

N.B. : \$ = 2017 US dollars. pers. = person. yr = year.

Name	**Short description**	**Unit**	**Value**
$\beta 1$	Linear damage coefficient on temperature. This parameter represents the direct impact of the temperature anomaly on the economy	$T^{-\beta 2}$	0.0236
$\beta 1\_KW[t,c]$	Linear damage coefficient on local temperature anomaly for Kalkuhl and Wenz based damage function	T^{-1}	—
$\beta 2$	Power damage coefficient on temperature. This parameter captures the non-linear effects of temperature anomalies on the economy	—	2
$\beta 2\_KW[t,c]$	Quadratic damage coefficient on local temperature anomaly for Kalkuhl and Wenz based damage function	T^{-2}	—
$\beta\_temp[c]$	Temperature scaling coefficients, which translate global temperature anomalies into country-level temperature anomalies	—	—
η	Inequality aversion.	—	1.5 (by default)
$\sigma[t,c]$	Emissions output ratio. This parameter is used for modelling emissions intensity as a function of economic activity	$GtCO_2/(10^6$ \$)	—
$\theta 2$	Exponent of abatement cost function (DICE-2023 value)	—	2.6
$\mu\_input[t,c]$	Input mitigation rate, used with option 3 "country_abatement_rate"	%	—
control_regime	Switch for emissions control regime ; 1 = "global_carbon_tax", 2 = "country_carbon_tax", 3 = "country_abatement_rate"	—	3 (by default)
daily_poverty_line	Daily poverty line	\$/pers/day	2.15

Name	Short description	Unit	Value
damage_elasticity	Income elasticity of climate damages (1 = proportional to income)	—	0.85
depk[t,c]	Depreciation rate on capital	%	—
elasticity	Income elasticity with respect to climate damage, mitigation costs, etc. It can either be `damage_elasticity` or `CO2_income_elasticity`	—	—
elasticity_intercept[t]	Intercept term for estimating income elasticity	—	3.22[1]
elasticity_slope[t]	Slope term for estimating income elasticity	—	-0.2[1]
global_carbon_tax[t]	Global carbon tax	$\$/tCO_2$	—
global_recycle_share[c]	Share of country revenues that are recycled globally in the form of international transfers (1 = 100%)	%	1 (by default)
global_temperature[t] = temp_anomaly[t]	Global average surface temperature excess (above pre-industrial [year 1750] level)	°C	—
income_shares[c,q]	An array of income shares by quantile (where rows represent countries and columns represent quantiles)	—	—
increase_value	The annual tax increase value. By default, it is equal to `tax_start_value`, which means that the tax increases by its initial value each year	$\$/tCO_2$	—
k0[c]	Initial level of capital. Determines the starting point for capital accumulation	10^6 \$/yr	—
l[t,c]	Labor/population. This parameter represents either the population or the available workforce for production of a given country	10^3 pers.	—
list_llmic_flag[c]	Equal to 1 if the country is on the list of low and middle income countries, 0 otherwise	—	—
list_risk_index[c]	Risk index	—	—
local_temp_anomaly[t,c]	Country-level average surface temperature anomaly (above pre-industrial [year 1750] level)	°C	—
lost_revenue_share	Portion of carbon tax revenue that is lost and cannot be recycled (1 = 100% of revenue lost, 0 = no revenue lost)	%	0 (by default)
mapcrrice[c]	Mapping from country index to RICE region index	—	183 countries to 12 regions
mapcrwpp[c]	Mapping from country index to UN WPP region index	—	183 countries to 20 regions
max_study_gdp	Maximum value of GDP per capita observed in elasticity studies	\$/pers.	48892

Name	Short description	Unit	Value
min__study__gdp	Minimum value of GDP per capita observed in elasticity studies	\$/pers.	647
nb__country	Number of countries	—	183
nb__quantile	Number of quantiles	—	10 (by default)
pbacktime[t]	*Backstop price* from DICE 2023	\$/tCO_2	—
quantile__income __shares[t,c,q]	Income shares of quantile	—	—
recycle__share[c,q]	Share of carbon tax revenues recycled to each quantile	—	$\frac{1}{\text{nb\_quantile}}$ (by default)
reference__carbon__tax[t]	Reference carbon tax	\$/tCO_2	—
reference__country__index	Reference country index	—	USA
s[t,c]	Savings rate	%	—
share	Capital's share of production. This parameter is global and affects the distribution of income between capital and labor	%	0.3
switch__dam __abs__llmic__recycle	Boolean, carbon tax revenues are recycled globally in proportion to absolute damage in LLMICs	—	0 or 1
switch__dam __rel__llmic__recycle	Boolean, carbon tax revenues are recycled globally in proportion to population-weighted relative damages in low- and middle-income countries (LLMICs)	—	0 or 1
switch__global __pc__recycle	Boolean, carbon tax revenues are recycled globally on an equal per capita basis	—	0 or 1
switch__recycle	Boolean for recycling carbon tax revenue	—	0 or 1
switch__risk __llmic__recycle	Boolean, carbon tax revenues are recycled according to a population-weighted risk index in LLMICs	—	0 or 1
switch__scope__recycle	Boolean, carbon tax revenues are recycled at national (0) or global (1) level	—	0 or 1
tax__start__value	The initial value of the carbon tax	\$/tCO_2	Depends on the chosen carbon tax pathway
temp__anomaly[t] = global__temperature[t]	Global average surface temperature excess (above pre-industrial [year 1750] level)	°C	—
tfp[t,c]	Total factor productivity	—	—
year__model__end	The end of the model. If it is less than `year_tax_end`, the last tax value is repeated up to this year	yr	2300 (by default)
year__step	The step in years between two tax values	yr	1 (by default)

Name	Short description	Unit	Value
year_tax_end	The last year for which to calculate the tax	yr	2200 (by default)
year_tax_start	The first year of the tax increase	yr	2020 (by default)

3 Model - Variables

Name	Short description	Unit
$\mu[t,c]$	GHG emissions mitigation rate	%
μ_cons$[t,c]$	μ parameter of the lognormal distribution of consumption	—
σ_cons$[t,c]$	σ parameter of the lognormal distribution of consumption	—
$\theta1[t,c]$	Multiplicative parameter of abatement cost function. Equal to ABATEFRAC at 100% mitigation	—
ABATECOST[t,c]	Cost of emission reductions	10^6 \$/yr
ABATEFRAC[t,c]	Cost of emission reductions as a share of gross economic output	%
abatement_cost _dist[t,c,q]	Share of the distribution of abatement costs per quantile	—
C[t,c]	Country consumption	10^6 \$/yr
carbon_tax_dist[t,c,q]	Shares of the distribution of CO2 tax burden per quantile	—
cons_EDE_country[t,c]	Consumption equivalent to equitably distributed well-being in a given country	10^3 \$/pers/yr
cons_EDE_global[t]	Global consumption equivalent to equitably distributed well-being	10^3 \$/pers/yr
cons_EDE_rwpp[t,rwpp]	Consumption equivalent to equitably distributed well-being for WPP regions	10^3 \$/pers/yr
country_carbon_tax[t,c]	CO2 tax rate	\$/tCO_2
country_pc_dividend[t,c]	Total fiscal dividends per person, including all international monetary transfers	10^3 \$/pers/yr
country_pc_dividend _domestic_transfers[t,c]	Fiscal dividends per person from domestic redistribution, i.e., within a country	10^3 \$/pers/yr
country_pc_dividend _global_transfers[t,c]	Tax dividends per person from international transfers	10^3 \$/pers/yr
country_pc_dividend _llmic[t,c]	Tax income per person in LLMIC	10^3 \$/pers/yr
CPC[t,c]	Country level consumption per capita	10^3 \$/pers/yr
CPC_post[t,c]	Country level consumption per capita after recycling	10^3 \$/pers/yr

[1]Results from the meta-regression based on study results to calculate elasticity vs. ln gdp per capita relationship.

Name	**Short description**	**Unit**
CPC_post_global[t]	World consumption per capita after recycling	10^3 \$/pers/yr
CPC_post_rwpp[t,rwpp]	Regional per capita consumption after recycling	10^3 \$/pers/yr
CPC_rwp[t,rwpp]	Regional level consumption per capita	10^3 \$/pers/yr
DAMFRAC[t,c]	Country-level damages as a share of net GDP based on global temperatures	%
damage_dist[t,c,q]	Share of the distribution of climate damage per quantile	—
E_gtco2[t,c]	Country-level total GHG emissions	10^9 tCO2/yr
E_Global_gtco2[t]	Global emissions (sum of all country emissions)	10^9 tCO2/yr
E_Global_gtco2 _rrice[t,rrice]	Regional GHG emissions for the regions defined in the RICE model	10^9 tCO2/yr
E_Global_gtc[t]	Global emissions in units of gigatonnes of carbon, giving compatible units with FAIR	10^9 tC/yr
GLOBAL_ABATEFRAC _full_abatement[t]	Global `ABATEFRAC[t]` in case of full mitigation	%
global_gini_cons[t]	Gini index of world consumption	—
global_poverty _population_cons[t]	Number of people living in poverty in the world, based on consumption	10^3 pers
global_revenue[t]	Carbon tax revenue, derived from the total recycled revenue of all countries	\$/yr
global_pc_revenue[t]	Carbon tax revenue per person, derived from the total recycled revenue of all countries	10^3 \$/pers/yr
gini_cons[t,c]	Gini index of country consumption	—
gini_cons_rwpp[t,rwpp]	Gini index of regional consumption	—
I[t,c]	Investment	10^6 \$/yr
K[t,c]	Capital	10^6 \$/yr
l_rwpp[t,rwpp]	Regional population	10^3 pers.
llmic_population[t]	Total population in low- and middle-income countries	10^3 pers.
LOCAL_ DAMFRAC_KW[t,c]	Country-level damages as a share of net GDP based on local temperatures and on Kalkuhl & Wenz	%
local_temperature[t,c]	Excess temperature at country level (above pre-industrial [year 1750] level)	°C
pc_gdp[t,c] = Y_pc[t,c]	Net GDP per capita after damages and mitigation costs	\$/pers/yr
poverty_population _cons[t,c]	Number of people living in poverty in each country, according to consumption	10^3 pers
poverty_population _cons_rwpp[t,rwpp]	Number of people in poverty at regional level, based on consumption	10^3 pers
poverty_rate_cons[t,c]	Poverty rate in a country, according to consumption. The poverty line is defined as \$ 2.15 per capita per day	%

Name	Short description	Unit
qc_base[t,c,q]	Consumption per quantile per capita before damage, before abatement cost, before tax	10^3 $pers/yr
qc_post _damage_abatement[t,c,q]	Consumption per quantile per capita after damage, after abatement	10^3 $/pers/yr
qc_post_recycle[t,c,q]	Consumption per quantile per capita after recycling the carbon tax to each quantile	10^3 $/pers/yr
qc_post_tax[t,c,q]	Consumption per quantile per capita after subtraction of the carbon tax	10^3 $/pers/yr
qc_share[t,c,q]	Proportion of consumption per quantile per capita	%
qpop[t,c,q]	Population per quantile	10^3 pers
revenue_recycled _global_level[t]	Recycle a share `global_recycle_share` of `tax_revenue` at global level	10^3 $/yr
tax_pc_revenue[t,c]	Carbon tax revenue per capita from country emissions	10^3 $/pers/yr
tax_revenue[t,c]	Carbon tax revenue for a given country	$/yr
tax_values[t]	Array containing the carbon tax values over time, until the last year of tax defined	—
total_c_post_recycle[t,c]	Total consumption per country after recycling	10^3 $/pers/yr
total_tax_pc_revenue[t]	Total carbon tax revenue per person, sum of tax revenues in all countries per person	10^3 $/pers/yr
total_tax_revenue[t]	Total carbon tax revenue, sum of tax revenues in all countries	$/yr
TRANSFRAC_dam _abs_llmic[t,c]	Proportion of damage suffered in the country in relation to global damage (damage costs ratio)	%
TRANSFRAC_dam _rel_llmic[t,c]	Proportion of damage suffered in the country in relation to global damage (share of population-weighted global net output ratio)	%
TRANSFRAC_risk _llmic[t,c]	Proportion of risk based on the population-weighted 2023 risk index in LLMIC	%
updated_quantile _distribution[t,c,q]	An array which contains the updated quantile share distribution for each country, considering the given income elasticity	—
welfare_country[t,c]	Welfare for countries	—
welfare_global[t]	Global welfare	—
welfare_rwpp[t,rwpp]	Welfare in a given WPP region	—
YGROSS[t,c]	Gross output	10^6 $/yr
YGROSS_global[t]	Global gross output, represents the sum of all countries' gross production	10^{12} $/yr
Y[t,c]	Output net of damages and abatement costs	10^6 $/yr
Y_pc[t,c] = pc_gdp[t,c]	Net GDP per capita after damages and mitigation costs	$/pers/yr

Name	Short description	Unit
Y_pc_rwpp[t,rwpp]	Regional per capita output net of abatement and damages	$/pers/yr

4 In theory

The goal is to maximize the following Social Welfare function :

$$W(\text{cons}(c,q,t)) = \sum_{cqt} \frac{l(c,q,t)}{(1+\rho)^t} \frac{\text{cons}(c,q,t)^{1-\eta}}{1-\eta} \tag{1}$$

- W : social welfare
- cons : per capita consumption
- l : population
- η : inequality aversion (marginal utility elasticity)
- ρ : rate of pure time preference
- c : country
- q : quantile
- t : time

5 abatement.jl

This file defines the components needed to calculate the cost of mitigation.

5.1 Indexes

- `country`: Countries included in the model.
- `regionrice`: Regions defined by the RICE model.

5.2 Parameters

5.2.1 Time-dependent

- `pbacktime`: *backstop price* from DICE 2023, in USD2017 per tCO2.
- `global_carbon_tax`: Global carbon tax (USD2017 per tCO2).
- `reference_carbon_tax`: Reference carbon tax (USD2017 per tCO2).

5.2.2 Time and country dependent

- σ : Emissions output ratio, GtCO2 per million of 2017 US dollars .
- `YGROSS`: Gross output, measured in millions of 2017 US dollars per year.
- `s`: Savings rate (%).
- `l`: Labor/population (thousands).
- μ_input : Input mitigation rate, used with option 3 “country_abatement_rate”.

5.2.3 Independent

- η : Inequality aversion.
- $\theta 2$: Exponent of abatement cost function (DICE-2023 value).
- `reference_country_index`: Reference country index.
- `control_regime`: Switch for emissions control regime ; 1 = “global_carbon_tax”, 2 = “country_carbon_tax”, 3 = “country_abatement_rate”.

5.3 Variables

5.3.1 Time and country dependent

- $\theta 1$: Multiplicative parameter of abatement cost function. Equal to ABATEFRAC at 100% mitigation
- `country_carbon_tax`: CO2 tax rate (2017 USD per tCO2)
- μ : GHG emissions mitigation rate ($\mu \in [0; 1]$).
- `ABATEFRAC`: Cost of emission reductions in share of gross output.
- `ABATECOST`: Cost of emission reductions, in millions USD2017 per year.

5.3.2 Time-dependent

- `GLOBAL_ABATEFRAC_full_abatement`: global `ABATEFRAC` in case of full mitigation (*i.e.* $\mu = 1$).

5.4 Equations

5.4.1 Calculating $\theta 1$ for each country

For each country `c` in the `country` set, $\theta 1$ is calculated as follows :

$$\theta 1[t,c] = pbacktime[t] \times \frac{\sigma[t,c] \times 10^3}{\theta 2} \tag{2}$$

The 10^3 factor is used for converting σ into tCO2/USD2017.

Then, depending on the emission control regime specified by `p.control_regime`, different methods are used to calculate the abatement rate μ and the country-specific carbon tax `country_carbon_tax`.

a. Global carbon tax, `control_regime == 1` A global carbon tax is applied to all countries :

$$country\_carbon\_tax[t,c] = global\_carbon\_tax[t] \tag{3}$$

$$\mu[t,c] = min\left(max(0, (\frac{country\_carbon\_tax[t,c] \times \sigma[t,c] \times 10^3}{\theta 1[t,c] \times \theta 2})^{\frac{1}{\theta 2 - 1}}), 1\right) \tag{4}$$

b. Carbon tax by country, `control_regime == 2` Each country's carbon tax is calculated from the reference carbon tax, adjusted according to the country's economic and demographic parameters.

$$\begin{aligned} country\_carbon\_tax[t,c] = reference\_carbon\_tax[t] \times \frac{1 - s[t, reference\_country\_index]}{1 - s[t,c]} \\ \times \left(\frac{YGROSS[t,c] l[t,c]}{YGROSS[t, reference\_country\_index] l[t, reference\_country\_index]}\right)^{\eta} \end{aligned} \tag{5}$$

$$\mu[t,c] = min\left(max(0, (\frac{country\_carbon\_tax[t,c] \times \sigma[t,c] \times 10^3}{\theta 1[t,c] \times \theta 2})^{\frac{1}{\theta 2 - 1}}), 1\right) \tag{6}$$

c. Abatement rate by country, `control_regime == 3` The abatement rate μ is supplied directly by the user ($\mu[t,c] = \mu\_input[t,c]$) and the country's carbon tax is calculated according to this rate.

$$country\_carbon\_tax[t,c] = \frac{\theta 1[t,c] \times \theta 2}{\sigma[t,c] \times 10^3} * \mu^{\theta 2 - 1}[t,c] \tag{7}$$

5.4.2 Calculation of abatement cost, abatement rate and overall abatement rate in the case of full abatement

For each country, the abatement rate `ABATEFRAC` and the abatement cost `ABATECOST` are calculated as follows:

$$ABATEFRAC[t,c] = \theta 1[t,c] \times \mu^{\theta 2}[t,c] \tag{8}$$

$$ABATECOST[t,c] = YGROSS[t,c] \times ABATEFRAC[t,c] \tag{9}$$

$$GLOBAL\_ABATEFRAC\_full\_abatement[t] = \frac{\sum_c (\theta 1[t,c] \times YGROSS[t,c])}{\sum_c YGROSS[t,c]} \tag{10}$$

6 damages.jl

This file calculates the economic damage linked to temperature anomalies, as a share of GDP.

6.1 Indexes

- `country`: Countries included in the model.
- `regionrice`: Regions defined by the RICE model.

6.2 Parameters

6.2.1 Independent

- $\beta 1$: Linear damage coefficient on temperature. This parameter represents the direct impact of the temperature anomaly on the economy.
- $\beta 2$: Power damage coefficient on temperature. This parameter captures the non-linear effects of temperature anomalies on the economy..

6.2.2 Time-dependent

- `temp_anomaly`: Global average surface temperature anomaly (°C above pre-industrial [year 1750]).

6.2.3 Time and country dependent

- `local_temp_anomaly`: Country-level average surface temperature anomaly (°C above pre-industrial [year 1750]).
- $\beta 1\_KW$: Linear damage coefficient on local temperature anomaly for Kalkuhl and Wenz based damage function.
- $\beta 2\_KW$: Quadratic damage coefficient on local temperature anomaly for Kalkuhl and Wenz based damage function.

6.3 Variables

6.3.1 Time and country dependent

- `LOCAL_DAMFRAC_KW`: Country-level damages as a share of net GDP based on local temperatures and on Kalkuhl & Wenz.
- `DAMFRAC`: Country-level damages as a share of net GDP based on global temperatures.

6.4 Equations

$$DAMFRAC[t, c] = \beta 1 \times temp\_anomaly^{\beta 2}[t] \tag{11}$$

$$\begin{aligned} LOCAL\_DAMFRAC\_KW[t, c] = \beta 1\_KW[c] \times local\_temp\_anomaly[t, c] \\ + \beta 2\_KW[c] \times local\_temp\_anomaly^2[t, c] \end{aligned} \tag{12}$$

7 emissions.jl

This file allows to calculate greenhouse gas (GHG) emissions at global and national level.

7.1 Indexes

- `country`: Countries included in the model.
- `regionrice`: Regions defined by the RICE model.

7.2 Parameters

7.2.1 Country dependent

- `mapcrrice`: Mapping from country index to RICE region index. This parameter is indexed by country, linking country data to specific regions in the RICE model.

7.2.2 Time and country dependent

- σ : Emissions output ratio, GtCO2 per million USD2017. This parameter is used for modelling emissions intensity as a function of economic activity.
- `YGROSS`: Gross output, in million USD2017 per year. This parameter represents the economic activity of each country.
- μ : GHG emissions control rate, representing the effectiveness of mitigation policies implemented.

7.3 Variables

7.3.1 Time-dependent

- `E_Global_gtco2`: Global emissions (sum of all country emissions), in GtCO2 per year.
- `E_Global_gtc`: Global emissions in units of gigatonnes of carbon (GtC per year), giving compatible units with FAIR.

7.3.2 Time and country dependent

- `E_gtco2`: Country level total greenhouse gas emissions, in GtCO2 per year.

7.3.3 Time and RICE region dependent

- `E_gtco2_rrice`: Regional GHG emissions for the regions defined in the RICE model, measured in GtCO2 per year.

7.4 Equations

7.4.1 Emissions by country

For each country `c` in the `country` set, CO2 emissions are calculated using the following equation:

$$E_{\mathrm{gtco2}}[t, c] = YGROSS[t, c] \cdot \sigma[t, c] \cdot (1 - \mu[t, c]) \tag{13}$$

7.4.2 Global emissions

Global CO2 emissions in year `t` are calculated as the sum of all countries' emissions:

$$E_{\text{Global\_gtco2}}[t] = \sum_{c} E_{\text{gtco2}}[t, c] \tag{14}$$

7.4.3 Global emissions in Carbon Units (GtC)

Global emissions are then converted into gigatons of carbon (GtC) using the molecular conversion factor :

$$E_{\text{Global\_gtc}}[t] = E_{\text{Global\_gtco2}}[t] \cdot \frac{12.01}{44.01} \tag{15}$$

This conversion is necessary for integration with the FAIR model, which uses emissions in GtC.

7.4.4 Regional emissions for RICE regions

For each `rrice` RICE region in the set of `regionrice` regions, regional CO2 emissions are calculated as the sum of the emissions of the countries belonging to that region:

$$E_{\text{gtco2\_rrice}}[t, rrice] = \sum_{c \in \text{rrice}} E_{\text{gtco2}}[t, c] \tag{16}$$

This process makes it possible to aggregate CO2 emissions at different levels of granularity.

8 grosseconomy.jl

8.1 Index

- `country`: Countries in the model.

8.2 Parameters

8.2.1 Country dependent

- `k0`: Initial level of capital. Determines the starting point for capital accumulation.

8.2.2 Time and country dependent

- `I`: Investment, in millions of 2017 US dollars per year (1e6 USD2017/year).
- `l`: Labor/population (workforce available for production).
- `tfp`: Total factor productivity.
- `depk`: Depreciation rate on capital. Represents the loss of capital value due to wear and tear and obsolescence.

8.2.3 Independent

- `share`: Capital's share of production. This parameter is global and affects the distribution of income between capital and labor.

8.3 Variables

8.3.1 Time-dependent

- `YGROSS_global`: Global gross output, measured in trillions of 2017 US dollars per year (1e12 USD2017/year). Represents the sum of all countries' gross production.

8.3.2 Time and country dependent

- `YGROSS`: Gross output in each country, measured in millions of 2017 US dollars per year.
- `K`: Capital, in millions of USD2017 per year.

8.4 Equations

8.4.1 Capital (K)

The capital in each country is calculated for each period `t` and for each country `c`.

- For the first period (`t` is the first period), capital is initialized to the value `k0` for country `c` :

$$K[t, c] = k0[c] \tag{17}$$

- For subsequent periods, capital is calculated by taking into account the previous year's capital depreciation and the previous year's investment, multiplied by 5 (representing a 5-year period):

$$K[t, c] = (1 - depk[t, c])^5 \cdot K[t - 1, c] + I[t - 1, c] \cdot 5 \tag{18}$$

8.4.2 Calculation of Gross Production by Country (YGROSS)

Gross output by country is calculated for each period `t` and for each country `c` using the Cobb-Douglas production function equation:

$$YGROSS[t, c] = tfp[t, c] \cdot K[t, c]^{share} \cdot l[t, c]^{(1-share)} \tag{19}$$

8.4.3 Calculation of Global Gross Production (YGROSS_global)

Global gross output is calculated for each period `t` by aggregating the gross output of all countries and converting the result into trillions of USD2017 :

$$YGROSS\_global[t] = \frac{\sum_c YGROSS[t, c]}{10^6} \tag{20}$$

9 neteconomy.jl

This file calculates the net savings by taking into account the mitigation costs and damages associated with climate change.

9.1 Indexes

- `country`: Countries included in the model.
- `regionwpp`: Regions defined from the World Population Prospects database (UN).

9.2 Parameters

9.2.1 Country dependent

- `mapcrwpp`: Mapping from country index to WPP region index. This parameter is indexed by country, linking country data to specific regions defined by the UN WPP.

9.2.2 Time and country dependent

- `YGROSS`: Gross output in each country, measured in millions of 2017 US dollars per year.
- `ABATEFRAC`: Cost of emission reductions in share of gross output.

- LOCAL_DAMFRAC_KW: Country-level damages as a share of net GDP based on local temperatures and on Kalkuhl & Wenz.
- s: Savings rate (%).
- l: Labor/population (in thousands).

9.3 Variables

9.3.1 Time and country dependent

- Y: Output net of damages and abatement costs, in millions USD2017 per year.
- C: Country consumption, in millions USD2017 per year.
- CPC: Country level consumption per capita, in thousands USD2017 per person per year.
- Y_pc: Per capita output net of abatement and damages, in USD2017 per person per year.
- I: Investment, in millions of USD2017 per year.

9.3.2 Time and WPP region dependent

- C_rwpp: Regional consumption, in millions USD2017 per year.
- l_rwpp: Regional population (in thousands).
- CPC_rwpp: Regional level consumption per capita, in thousand USD2017 per person per year.
- Y_pc_rwpp: Regional per capita output net of abatement and damages, in USD2017 per person per year.

9.4 Equations

- Regional consumption C_rwpp is initialized to zero for the current time step.

9.4.1 Country loop

For each country c in the country set:

1. **Net Output** calculation: Net output Y is calculated by adjusting gross output YGROSS for mitigation costs ABATEFRAC and damages LOCAL_DAMFRAC_KW according to equation :

$$Y[t,c] = \frac{(1 - ABATEFRAC[t,c])}{(1 + LOCAL\_DAMFRAC\_KW[t,c])} \cdot YGROSS[t,c] \tag{21}$$

2. **Investment** : Investment I is determined as a fraction s of net output Y :

$$I[t,c] = s[t,c] \cdot Y[t,c] \tag{22}$$

3. **Consumption by country** : Consumption C is calculated as net output Y minus investment I, except for the last period where it is equal to consumption in the previous period.

$$C[t,c] = \begin{cases} Y[t,c] - I[t,c] & \text{if } t \text{ is not the last period} \\ C[t-1,c] & \text{otherwise} \end{cases} \tag{23}$$

4. **Consumption per capita** : Per capita consumption CPC is calculated by dividing consumption C by population l.

$$CPC[t,c] = \frac{C[t,c]}{l[t,c]} \tag{24}$$

5. **Net output per head** : Net output per capita `Y_pc` is calculated by adjusting net output `Y` for population `l`, with scaling to convert units.

$$Y\_pc[t,c] = \frac{Y[t,c]}{l[t,c]} \times 10^3 \tag{25}$$

9.4.2 Loop over WPP regions

For each region `rwpp` in the set of regions `regionwpp` :

1. **Country indices** : The indices of the `c` countries belonging to the `rwpp` region are identified.
2. **Regional Consumption**:

$$C\_rwpp[t,rwpp] = \sum_{c \in rwpp} C[t,c] \tag{26}$$

3. **Regional population** :

$$l\_rwpp[t,rwpp] = \sum_{c \in rwpp} l[t,c] \tag{27}$$

4. **Regional per capita consumption** :

$$CPC\_rwpp[t,rwpp] = \frac{C[t,rwpp]}{l[t,rwpp]} \tag{28}$$

5. **Regional net output per capita** :

$$Y\_pc\_rwpp[t,c] = \frac{\sum_{c \in rwpp} Y[t,c]}{\sum_{c \in rwpp} l[t,c]} \times 10^3 \tag{29}$$

10 pattern_scale.jl

File designed to calculate local temperature changes from global mean temperature variations, using country-specific scaling coefficients.

10.1 Index

- `country = Index()`: defines a `country` index for the modelled regions. This index is used for iterating over the different countries in the model.

10.2 Parameters

- β_temp`= Parameter(index=[country])`: declares a β_temp parameter which represents temperature scaling coefficients. These coefficients translate global temperature anomalies into country-level temperature anomalies. They are indexed by country.
- `global_temperature = Parameter(index=[time])`: declares a `global_temperature` parameter which represents the time-dependent global average temperature excess (in °C).

10.3 Variables

- `local_temperature = Variable(index=[time, country])`: declares a `local_temperature` variable which represents the excess temperature at country level (in °C) according to country and year (time-dependent).

10.4 Function

- `function run_timestep(p, v, d, t)`: Defines the `run_timestep` function which is called at each time step of the model. It takes as parameters the parameters (`p`), the variables (`v`), the dimensions (`d`), and the current time step (`t`).

10.5 Equation

For all countries, the local temperature is estimated as the product of the scaling coefficient β_temp and the global mean temperature anomaly at time `t`.

$$\text{local\_temperature}[t, c] = \beta\_temp[c] \cdot \text{global\_temperature}[t] \tag{30}$$

11 quantile_distribution.jl

This file calculates many variables (see *infra*), including per capita consumption, the poverty rate or the Gini index.

11.1 Indices

- `country`: Countries included in the model.
- `regionwpp`: Regions defined from the World Population Prospects database (UN).
- `quantile`: Index of income quantiles

11.2 Parameters

11.2.1 Time and country dependent

- `ABATEFRAC`: Cost of CO2 emission reductions as share of gross economic output.
- `LOCAL_DAMFRAC_KW`: National damages as a share of net GDP calculated from national temperatures.
- `CPC`: Country level consumption per capita, in thousands of USD2017 per person per year.
- `l`: Population of the country (in thousands).
- `Y` : Net production after damage and mitigation costs, in millions USD2017 per year.
- `country_pc_dividend`: Total per capita revenue from the carbon tax, including any international transfers, in thousands of USD2017 per person per year.
- `tax_pc_revenue`: Carbon tax revenue per capita from country emissions, in thousands of USD2017 per person per year.

11.2.2 Other

- `switch_recycle`: Recycling of carbon tax revenues.
- `nb_quantile`: Number of quantiles (10 by default).
- `min_study_gdp`, `max_study_gdp`: Minimum and maximum values of GDP per capita observed in elasticity studies (in dollars per person).
- `elasticity_intercept`, `elasticity_slope`: Intercept and slope terms for estimating income elasticity (time-dependent).
- `damage_elasticity`: Elasticity of income in relation to climate damage (1 = proportional to income).
- `daily_poverty_line`: Daily poverty line, in USD2017.
- `quantile_income_shares`: Income shares of deciles, depending on time, country and quantiles.
- `recycle_share`: Share of carbon tax revenues recycled to each quantile, depending on country and quantiles.
- `mapcrwpp`: Mapping from country index to WPP region index.

11.3 Variables

11.3.1 Time-dependent

- `CPC_post_global`: World consumption per capita after recycling, in thousands of USD2017 per person per year.
- `global_gini_cons`: Gini index of world consumption.
- `global_poverty_population_cons`: Number of people living in poverty in the world (in thousands), based on consumption.

11.3.2 Time and country dependent

- `CO2_income_elasticity`: Income elasticity - price of CO2.
- `pc_gdp`: Net GDP per capita after damages and mitigation costs (in USD2017 per person per year).
- `total_c_post_recycle`: Total consumption per country after recycling, in thousands of USD2017 per person per year.
- `CPC_post`: National consumption per capita after recycling, in thousands of USD2017 per person per year.
- `gini_cons`: Gini index of country consumption (%).
- μ`_cons`$[t, c]$, σ`_cons`$[t, c]$: μ and σ parameters of the lognormal distribution of consumption.
- `poverty_rate_cons`: Poverty rate in a country (%), according to consumption. The poverty line is defined as USD2017 2.15 per person per day.
- `poverty_population_cons`: Number of people living in poverty in each country (in thousands), according to consumption.

11.3.3 Time, country and quantile dependent

- `abatement_cost_dist`, `carbon_tax_dist`, `damage_dist`: Shares of the distribution of abatement costs, CO2 tax burden and climate damage per quantile.
- `qc_base`: Consumption per quantile per capita before damage, before abatement cost, before tax (thousands of USD2017 per person per year).
- `qc_post_damage_abatement`: Consumption per quantile per capita after damage, after abatement (thousands of USD2017 per person per year).
- `qc_post_tax`: Consumption per quantile per capita after subtraction of the carbon tax (thousands USD2017 per person per year).
- `qc_post_recycle`: Consumption per quantile per capita after recycling the carbon tax to each quantile (thousands USD2017 per person per year).
- `qc_share`: Proportion of consumption per quantile per capita (%).
- `qpop`: Population per quantile (thousands).

11.3.4 Time and `regionwpp` dependent

- `CPC_post_rwpp`: Regional per capita consumption after recycling, in thousands of USD2017 per person per year.
- `gini_cons_rwpp`: Gini index of regional consumption (%).
- `poverty_population_cons_rwpp`: Number of people in poverty at regional level (in thousands), based on consumption.

11.4 Equation

The `run_timestep` function calculates various economic indicators for each time step, country and income quantile.

11.4.1 Calculation of GDP per capita ($/pers/year)

$$pc\_gdp[t, c] = \frac{Y[t, c]}{l[t, c]} 10^3 \tag{31}$$

The factor 10^3 allows the conversion in dollars per person.

11.4.2 Population by quantile (in thousands)

$$qpop[t,c,q] = \frac{l[t,c]}{nb\_quantile} \tag{32}$$

11.4.3 Income elasticity - CO2 price

$$\text{CO2\_income\_elasticity}[t,c] = \text{elasticity\_intercept} + \text{elasticity\_slope} \times log(GDP) \tag{33}$$

where :

$$GDP = \begin{cases} pc\_gdp[t,c] & \text{if pc\_gdp[t,c]} \in \text{[min\_study\_gdp, max\_study\_gdp]} \\ min\_study\_gdp & \text{if pc\_gdp[t,c]} < \text{min\_study\_gdp} \\ max\_study\_gdp & \text{if pc\_gdp[t,c]} > \text{max\_study\_gdp} \end{cases} \tag{34}$$

11.4.4 Distribution of Tax Burdens and Mitigation Costs

The distributions of the shares of the carbon tax burden (`carbon_tax_dist`), mitigation costs (`abatement_cost_dist`), and climate damages (`damage_dist`) per quantile are calculated using the `country_quantile_distribution` function.

- The carbon tax burden and mitigation costs are equal, calculated with the function `country_quantile_distribution` (see 14.1.1 for more details).

$$\left.\begin{array}{l}\texttt{carbon\_tax\_dist[t,c,q]} \\ \texttt{abatement\_cost\_dist[t,c,q]}\end{array}\right\} = \texttt{country\_quantile\_distribution}\Big(\texttt{CO2\_income\_elasticity[t,c]}, \texttt{quantile\_income\_shares[t,c,q], nb\_quantile}\Big) \tag{35}$$

- Climate damages are calculated by replacing the income-price elasticity of CO2 by the damage-income elasticity `damage_elasticity` in the above function.

$$\texttt{damage\_dist[t,c,q]} = \texttt{country\_quantile\_distribution}\Big(\texttt{damage\_elasticity}, \texttt{quantile\_income\_shares[t,c,q], nb\_quantile)}\Big) \tag{36}$$

11.4.5 Consumption per Quantile

$$\text{qc\_base}[t,c,q] = \text{nb\_quantile} \times \text{CPC}[t,c] \times \text{quantile\_income\_shares}[t,c,q] \times \frac{1 + \text{LOCAL\_DAMFRAC\_KW}[t,c]}{1 - \text{ABATEFRAC}[t,c]} \tag{37}$$

$$\text{qc\_post\_damage\_abatement}[t,c,q] = max\Big[\text{qc\_base}[t,c,q] - \text{nb\_quantile} \times \text{CPC}[t,c] \times \Big(\text{LOCAL\_DAMFRAC\_KW}[t,c] \times \text{damage\_dist}[t,c,q] - \frac{1 + \text{LOCAL\_DAMFRAC\_KW[t,c]}}{1 - \text{ABATEFRAC}[t,c]} \times \text{ABATEFRAC}[t,c] \times \text{abatement\_cost\_dist}[t,c,q]\Big), 10^{-8}\Big] \tag{38}$$

$$\begin{aligned} \text{qc\_post\_tax}[t,c,q] = \text{qc\_post\_damage\_abatement}[t,c,q] \\ - (\text{nb\_quantile} \times \text{tax\_pc\_revenue}[t,c] \times \text{carbon\_tax\_dist}[t,c,q]) \end{aligned} \tag{39}$$

- If the revenues are recycled, then consumption per post-tax quantile is increased by the product of the total per capita revenue from the carbon tax and the share of its recycling to each quantile:

$$\begin{aligned} \text{qc\_post\_recycle}[t,c,q] = \text{qc\_post\_tax}[t,c,q] \\ + (\text{nb\_quantile} \times \text{country\_pc\_dividend}[t,c] \times \text{recycle\_share}[c,q]) \end{aligned} \tag{40}$$

- If revenues are not recycled, then consumption per post-tax quantile is increased by the product of the carbon tax revenue per capita from each country's emissions and its initial tax burden:

$$\begin{aligned} \text{qc\_post\_recycle}[t,c,q] = \text{qc\_post\_tax}[t,c,q] \\ + (\text{nb\_quantile} \times \text{tax\_pc\_revenue}[t,c] \times \text{carbon\_tax\_dist}[t,c,q]) \end{aligned} \tag{41}$$

11.4.6 Calculation of per capita consumption

- Total consumption per country after recycling:

$$\text{total\_c\_post\_recycle}[t,c] = \sum_q \text{qc\_post\_recycle}[t,c,q] \tag{42}$$

- Per capita consumption after recycling:

$$\text{CPC\_post}[t,c] = \frac{\text{total\_c\_post\_recycle}[t,c]}{\text{nb\_quantile}} \tag{43}$$

- Proportion of per capita quantile consumption (%):

$$\text{qc\_share}[t,c,q] = \frac{\text{qc\_post\_recycle}[t,c,q]}{\text{total\_c\_post\_recycle}[t,c]} \times 100 \tag{44}$$

11.4.7 Calculation of the Gini Index and the Poverty Rate

The Gini index (`gini_cons`) and the poverty rate (`poverty_rate_cons`) are calculated using the quantile distribution of consumption.

- The **Gini Index** is calculated using the Gini function in Julia :

$$\texttt{gini\_cons[t,c]} = \texttt{gini}\Big(\texttt{convert(Vector\{Real\}},\texttt{qc\_share[t,c,:])}\Big) \times 100 \tag{45}$$

- If the Gini index is between 0 and 100, we can calculate σ and μ, taking care to maintain homogeneous units.

$$\sigma_{\text{cons}}[t,c] = \sqrt{2} \times \text{quantile}\left(\mathcal{N}(0,1), \frac{gini\_cons[t,c]/100 + 1}{2}\right) \tag{46}$$

$$\mu_{\text{cons}}[t,c] = \log(CPC[t,c] \times 10^3) - \frac{(\sigma_{\text{cons}})^2}{2} \tag{47}$$

- The **poverty rate** is calculated from a cumulative distribution function of a normal distribution with mean μ`_cons`$[t, c]$ and standard deviation σ`_cons`$[t, c]$, evaluated at the logarithm of the poverty line. It reflects the percentage of the population that falls below the poverty line, according to a normal distribution:

$$\text{poverty\_line} = \text{daily\_poverty\_line} \times 365 \tag{48}$$

$$\text{poverty\_rate\_cons}[t, c] = \text{cdf}\Big(\mathcal{N}(\mu_{\text{cons}}, \sigma_{\text{cons}}), \log(\text{poverty\_line})\Big) \times 100 \tag{49}$$

11.4.8 Regional and global calculations

Country indices c and regional indices $rwpp$ are mapped using `mapcrwpp`.

- Regional per capita consumption after recycling:

$$CPC\_post\_rwpp[t, rwpp] = \frac{\sum_{c \in rwpp}(l[t, c] \cdot CPC\_post[t, c])}{\sum_{c \in rwpp} l[t, c]} \tag{50}$$

- `CPC_post_global` is calculated as the weighted sum of countries' total consumption after recycling, divided by the world population:

$$CPC\_post\_global[t] = \frac{\sum_{c}(total\_c\_post\_recycle[t, c] \cdot l[t, c]/nb\_quantile)}{\sum_{c} l[t, c]} \tag{51}$$

- Regional Gini index:

$$\begin{aligned}&\texttt{gini\_cons\_rwpp[t, rwpp]} =\\ &\texttt{gini}\Big(\texttt{convert}\Big(\texttt{Vector\{Real\},vec(qc\_post\_recycle[t, country\_indices, :])}\Big),\\ &\qquad\texttt{convert}\Big(\texttt{Vector\{Real\}, vec(qpop[t, country\_indices, :])}\Big)\Big) \times 100\end{aligned} \tag{52}$$

- Global Gini index:

$$\begin{aligned}&\texttt{global\_gini\_cons[t]} = \texttt{gini}\Big(\texttt{convert}\Big(\texttt{Vector\{Real\},vec(qc\_post\_recycle[t, :, :])}\Big),\\ &\qquad\texttt{convert}\Big(\texttt{Vector\{Real\}, vec(qpop[t, :, :])}\Big)\Big) \times 100\end{aligned} \tag{53}$$

- The population in poverty (`poverty_population_cons_rwpp`) for a given region is the sum of the populations in poverty of the countries in that region, cleaned of missing values:

$$\text{poverty\_population\_cons\_rwpp}[t,\ rwpp] = \sum_{c \in rwpp} \text{poverty\_population\_cons}[t,\ c] \tag{54}$$

- Population living in poverty at global level :

$$\text{global\_poverty\_population\_cons}[t] = \sum_{c} \text{poverty\_population\_cons}[t,\ c] \tag{55}$$

12 revenue_recycle.jl

This component models the recycling of carbon tax revenues across different countries and quantiles. It is essential for studying mitigation policies and their impact on reducing poverty and inequality. `revenue_recycle.jl` is used by `nice_v2_module.jl` and interacts with `quantile_distribution.jl` and `welfare.jl`.

12.1 Index

- `country` : Countries included in the model.

12.2 Parameters

12.2.1 Time and country dependent

- `Y`: Net production, in millions of USD2017 per year.
- `country_carbon_tax`: Carbon tax by country ($/tCO2).
- `LOCAL_DAMFRAC_KW`: National damage as a percentage of net GDP calculated from national temperatures.
- `E_gtco2`: Industrial carbon emissions (GtCO2 per year).
- `l`: Country population (in thousands).
- `tax_pc_revenue`: Carbon tax revenue per capita from the country's emissions, in thousands of USD2017 per person per year.

12.2.2 Country dependent

- `global_recycle_share`: Share of country revenues that are recycled globally in the form of international transfers (1 = 100%).
- `list_llmic_flag`: Equal to 1 if the country is on the list of low and middle income countries, 0 otherwise.
- `list_risk_index`: Risk index.

12.2.3 Independent

- `switch_recycle`: Recycling of carbon tax revenues.
- `lost_revenue_share`: Portion of carbon tax revenue that is lost and cannot be recycled (1 = 100% of revenue lost, 0 = no revenue lost).
- `switch_recycle`: Boolean for recycling carbon tax revenue.
- `switch_scope_recycle`: Boolean, carbon tax revenues are recycled at national (0) or global (1) level.
- `switch_global_pc_recycle`: Boolean, carbon tax revenues are recycled globally on an equal per capita basis (1).
- `switch_dam_rel_llmic_recycle`: Boolean, carbon tax revenues are recycled globally in proportion to population-weighted relative damages in low- and middle-income countries (LLMIC) (1).
- `switch_dam_abs_llmic_recycle`: Boolean, carbon tax revenues are recycled globally in proportion to absolute damage in LLMICs (1).
- `switch_risk_llmic_recycle`: Boolean, carbon tax revenues are recycled according to a population-weighted risk index in LLMICs (1).
- `nb_country`: Number of countries.

12.3 Variables

12.3.1 Time-dependent

- `total_tax_revenue`: Total carbon tax revenue (USD2017/year), sum of tax revenues in all countries.
- `total_tax_pc_revenue`: Total carbon tax revenue per person per year (in thousands of dollars per person per year), sum of tax revenues in all countries per person per year.
- `global_revenue`: Carbon tax revenue, derived from the total recycled revenue of all countries (in USD2017 per year).
- `global_pc_revenue`: Carbon tax revenue per person per year, derived from the total recycled revenue of all countries (in thousands of dollars per person per year).
- `llmic_population`: Total population in low- and middle-income countries (in thousands).
- `revenue_recycled_global_level`: Sum for all countries of `tax_revenue` recycled at a `global_recycle_share`. Mesured in thousands of USD2017 per year.

12.3.2 Time and country dependent

- `TRANSFRAC_dam_rel_llmic`: Proportion of damage suffered in the country in relation to global damage (share of global net GDP lost per person).
- `TRANSFRAC_dam_abs_llmic`: Proportion of damage suffered in the country in relation to global damage (share of global net GDP lost per person in LLMIC).
- `TRANSFRAC_risk_llmic`: Proportion of risk based on the population-weighted 2023 risk index in LLMIC.
- `country_pc_dividend_llmic`: Tax income per person in LLMIC, in thousands of USD2017 per person per year.
- `tax_revenue`: Carbon tax revenue for a given country, in USD2017 per year.
- `tax_pc_revenue`: Carbon tax revenue per person for a given country, in thousands of USD2017 per person per year.
- `country_pc_dividend`: Total fiscal dividends per person, including all international monetary transfers, in thousands of USD2017 per person per year.
- `country_pc_dividend_domestic_transfers`: Fiscal dividends per person from domestic redistribution, i.e. within a country, in thousands of USD2017 per person per year.
- `country_pc_dividend_global_transfers`: Tax dividends per person from international transfers, in thousands of USD2017 per person per year.

12.4 Equation

The `run_timestep` function calculates the proportion of income redistributed according to various criteria, including relative and absolute damage or the risk index, specifically for low- and middle-income countries (LLMIC).

It begins by identifying the indices of countries considered to be poor (LLMIC) and calculates the total population of these countries at time `t` :

$$\text{llmic\_population}[t] = \sum_{c \in \text{LLMIC}} l[t, c] \tag{56}$$

12.4.1 Relative and absolute damages and risk index for LLMICs

- If the country `c` is not in the list of poor countries, all three variables are set to zero:

$$\left.\begin{aligned} &\text{TRANSFRAC\_dam\_rel\_llmic}[t, c] \\ &\text{TRANSFRAC\_dam\_abs\_llmic}[t, c] \\ &\text{TRANSFRAC\_risk\_llmic}[t, c] \end{aligned}\right\} = 0 \tag{57}$$

- Otherwise, for the first time step, each redistribution fraction for LLMICs is initialised as the inverse of the total number of poor countries:

$$\left.\begin{aligned} &\text{TRANSFRAC\_dam\_rel\_llmic}[t = 1, c] \\ &\text{TRANSFRAC\_dam\_abs\_llmic}[t = 1, c] \\ &\text{TRANSFRAC\_risk\_llmic}[t = 1, c] \end{aligned}\right\} = \frac{1}{\text{Number of LLMICs}} \tag{58}$$

- For the following time steps, we have :
  - Relative damage (`TRANSFRAC_dam_rel_llmic`): maximum between local damage (`LOCAL_DAMFRAC_KW`) and 0, multiplied by the population of the country (`l`), divided by

the sum of this product over all LLMICs.

$$\text{TRANSFRAC\_dam\_rel\_llmic}[t,c] = \frac{\max\Big(\text{LOCAL\_DAMFRAC\_KW}[t,c],0\Big)\cdot l[t,c]}{\sum_{c\in LLMIC}\max\Big(\text{LOCAL\_DAMFRAC\_KW}[t,c],0\Big)\cdot l[t,c]} \tag{59}$$

– Absolute damage (`TRANSFRAC_dam_abs_llmic`): the operation is the same, substituting population `l` for gross output `Y`.

$$\text{TRANSFRAC\_dam\_abs\_llmic}[t,c] = \frac{\max\Big(\text{LOCAL\_DAMFRAC\_KW}[t,c],0\Big)\cdot Y[t,c]}{\sum_{c\in LLMIC}\max\Big(\text{LOCAL\_DAMFRAC\_KW}[t,c],0\Big)\cdot Y[t,c]} \tag{60}$$

– Risk index (`TRANSFRAC_risk_llmic`): product of the risk index of country `c` (`list_risk_index[c]`) by its population `l`, divided by the sum of this product over all LLMICs.

$$\text{TRANSFRAC\_risk\_llmic}[t,c] = \frac{list\_risk\_index[c]\cdot l[t,c]}{\sum_{c\in LLMIC} list\_risk\_index[c]\cdot l[t,c]} \tag{61}$$

12.4.2 Carbon tax revenue

$$\text{tax\_revenue}[t,c] = \big(E\_gtco2[t,c]\times\text{country\_carbon\_tax}[t,c]\times 10^9\big) \times (1-\text{lost\_revenue\_share}) \tag{62}$$

Emissions are in GtCO2 and `country_carbon_tax` in USD2017/tCO2, hence the normalisation by 10^9.

$$\text{tax\_pc\_revenue}[t,c] = \frac{tax\_revenue[t,c]}{l[t,c]\times 10^6} \tag{63}$$

`tax_revenue` is in USD2017/year and population in thousands of inhabitants, hence the normalisation by 10^{-6}.

$$\text{total\_tax\_revenue}[t] = \sum_c \text{tax\_revenue}[t,c] \tag{64}$$

$$\text{total\_tax\_pc\_revenue}[t] = \frac{\text{total\_tax\_revenue}[t]}{\sum_c l[t,c]\times 10^6} \tag{65}$$

$$\text{global\_revenue}[t] = \sum_c\Big(\text{tax\_revenue}[t,c]\times\text{global\_recycle\_share}[c]\times\text{switch\_scope\_recycle}\Big) \tag{66}$$

$$\text{global\_pc\_revenue}[t] = \frac{\text{global\_revenue}[t]}{\sum_c(l[t,c])\times 10^6} \tag{67}$$

12.4.3 Dividends according to Boolean values

`switch_recycle = 1` No income recycling implies no dividends.

$$\left.\begin{array}{l}\text{country\_pc\_dividend\_domestic\_transfers}[t,c]\\ \text{country\_pc\_dividend\_global\_transfers}[t,c]\end{array}\right\} = 0 \tag{68}$$

`switch_recycle==1, switch_scope_recycle==0` Revenues from the carbon tax are recycled at a national level:

$$\text{country\_pc\_dividend\_domestic\_transfers}[t,c] = \frac{tax\_revenue[t,c]}{l[t,c] \times 10^6} \tag{69}$$

$$\text{country\_pc\_dividend\_global\_transfers}[t,c] = 0 \tag{70}$$

`switch_recycle==1, switch_scope_recycle==1` Carbon tax revenues are recycled at a global level:

$$\text{country\_pc\_dividend\_domestic\_transfers}[t,c] = (1-\text{global\_recycle\_share}[c]) \times \frac{tax\_revenue[t,c]}{l[t,c] \times 10^6} \tag{71}$$

$$\text{revenue\_recycled\_global\_level}[t] = \sum_c (\text{tax\_revenue}[t,c] \times \text{global\_recycle\_share}[c]) \times 10^{-3} \tag{72}$$

Booleans change the way dividends are calculated.

- `switch_pc_global_recycle = 1`; Carbon tax revenues are recycled globally on an equal per capita basis.

$$\text{country\_pc\_dividend\_global\_transfers}[t,c] = \frac{\text{revenue\_recycled\_global\_level}[t]}{\sum_c (l[t,c]) \times 10^3} \tag{73}$$

- `switch_dam_rel_llmic_recycle = 1`; carbon tax revenues are recycled globally in proportion to population-weighted relative damages in low- and middle-income countries (LLMIC). In LLMICs:

$$\text{country\_pc\_dividend\_global\_transfers}[t,c] = \frac{\text{revenue\_recycled\_global\_level}[t,c] \times \text{TRANSFRAC\_dam\_rel\_llmic[t,c]}}{l[t,c] \times 10^3} \tag{74}$$

  Otherwise:

$$\text{country\_pc\_dividend\_global\_transfers}[t,c] = 0 \tag{75}$$

- `switch_dam_abs_llmic_recycle = 1`; carbon tax revenues are recycled globally in proportion to absolute population-weighted damages in low- and middle-income countries (LLMIC). In LLMICs:

$$\text{country\_pc\_dividend\_global\_transfers}[t,c] = \frac{\text{revenue\_recycled\_global\_level}[t] \times \text{TRANSFRAC\_dam\_abs\_llmic[t,c]}}{l[t,c] \times 10^3} \tag{76}$$

  Otherwise:

$$\text{country\_pc\_dividend\_global\_transfers}[t,c] = 0 \tag{77}$$

- `switch_risk_llmic_recycle = 1`; carbon tax revenues are recycled according to a population-weighted risk index in LLMIC. In LLMICs:

$$\text{country\_pc\_dividend\_global\_transfers}[t,c] = \frac{\text{revenue\_recycled\_global\_level}[t] \times \text{TRANSFRAC\_risk\_llmic}}{l[t,c] \times 10^3} \tag{78}$$

  Otherwise:

$$\text{country\_pc\_dividend\_global\_transfers}[t,c] = 0 \tag{79}$$

12.4.4 Total dividends

Total dividends per person are written as:

$$\text{country\_pc\_dividend}[t, c] = \text{country\_pc\_dividend\_domestic\_transfers}[t, c] + \text{country\_pc\_dividend\_global\_transfers}[t, c] \quad (80)$$

13 welfare.jl

This file calculates economic welfare and consumption equivalent to equitably distributed welfare (EDE) as a function of inequality aversion (η). `welfare.jl` is loaded by `nice_v2_module.jl` and uses data provided by `revenue_recycle.jl` and `quantile_distribution.jl`.

13.1 Indexes

- `country`, `regionwpp`, `quantile` : Indexes representing countries, regions and income quantiles respectively.

13.2 Parameters

- `qc_post_recycle`: Per capita quantile consumption after redistribution of the recycling tax (in thousands of USD2017 per person per year). It depends on time, country and quantile.
- η: Inequality aversion.
- `nb_quantile`: Number of quantiles.
- `l`: Population (in thousands). This parameter is time and country dependent.
- `mapcrwpp`: Correspondence of the country index with the WPP regions index (country dependent).

13.3 Variables

- `cons_EDE_country`, `cons_EDE_rwpp`, `cons_EDE_global`: Consumption equivalent to equitably distributed well-being (in thousands of USD2017 per person per year) for countries, regions and globally.
- `welfare_country`, `welfare_rwpp`, `welfare_global`: Welfare for countries, regions and globally.

13.4 Equations

13.4.1 Case where $\eta \neq 1$

For each country (`c`), EDE consumption and welfare are calculated as follows:

EDE consumption by country

$$cons\_EDE\_country[t, c] = \left(\frac{1}{nb\_quantile} \sum_q (qc\_post\_recycle[t, c, q]^{(1-\eta)}) \right)^{\frac{1}{(1-\eta)}} \quad (81)$$

EDE consumption by region For each region, the country indices c are mapped using `mapcrwpp`.

$$cons\_EDE\_rwpp[t, rwpp] = \left(\frac{\sum_{c \in rwpp} (l[t, c] \cdot cons\_EDE\_country[t, c]^{(1-\eta)})}{\sum_{c \in rwpp} l[t, c]} \right)^{\frac{1}{(1-\eta)}} \quad (82)$$

Overall EDE consumption

$$cons\_EDE\_global[t] = \left(\frac{\sum_c (l[t, c] \cdot cons\_EDE\_country[t, c]^{(1-\eta)})}{\sum_c l[t, c]} \right)^{\frac{1}{(1-\eta)}} \quad (83)$$

Welfare by country

$$welfare\_country[t,c] = \left(\frac{l[t,c]}{nb\_quantile}\right)\sum_q \left(\frac{qc\_post\_recycle[t,c,q]^{(1-\eta)}}{(1-\eta)}\right) \tag{84}$$

Welfare by region

$$welfare\_rwpp[t,rwpp] = \sum_{c\in rwpp} welfare\_country[t,c] \tag{85}$$

Global welfare

$$welfare\_global[t] = \sum_c (welfare\_country[t,c]) \tag{86}$$

13.4.2 Case where $\eta = 1$

For each country (`c`), EDE consumption and welfare are calculated using the logarithm :

EDE consumption by country

$$cons\_EDE\_country[t,c] = \exp\left(\frac{1}{nb\_quantile}\sum_q \log(qc\_post\_recycle[t,c,q])\right) \tag{87}$$

EDE consumption by region

$$cons\_EDE\_rwpp[t,rwpp] = \exp\left(\frac{\sum_{c\in rwpp} l[t,c]\cdot\log(cons\_EDE\_country[t,c])}{\sum_{c\in rwpp} l[t,c]}\right) \tag{88}$$

Total EDE consumption

$$cons\_EDE\_country[t] = \exp\left(\frac{\sum_c l[t,c]\cdot\log(cons\_ED\_country[t,c])}{\sum_c l[t,c]}\right) \tag{89}$$

Welfare by country

$$welfare\_country[t,c] = \left(\frac{l[t,c]}{nb\_quantile}\right)\sum_q \log(qc\_post\_recycle[t,c,q]) \tag{90}$$

Welfare by region

$$welfare\_rwpp[t,rwpp] = \sum_{c\in rwpp} welfare\_country[t,c] \tag{91}$$

Global welfare

$$welfare\_global[t] = \sum_c (welfare\_country[t,c]) \tag{92}$$

14 helper_functions.jl

14.1 Functions

14.1.1 `country_quantile_distribution`

Objective The `country_quantile_distribution` function calculates the distribution of quantile shares for a given country, based on a given income elasticity. This function is useful for analysing the impact of climate damage, CO_2 mitigation costs, or CO_2 related tax burdens on different segments of the population within a country.

Arguments of the function

- `elasticity`: Income elasticity with respect to climate damage, CO_2 mitigation costs, CO_2 tax burdens, etc. This must be a real number.
- `income_shares`: An array of income shares by quantile (where the rows represent the countries and the columns the quantiles)
- `nb_quantile`: Number of quantiles, specified as an integer.

Process

1. **Application of elasticity**: For a given country, income elasticity is applied to income shares per quantile by raising each income share to the power of the elasticity.

$$\text{scaled\_shares}[t, c, q] = \text{income\_shares}^{\text{elasticity}}[t, c, q] \tag{93}$$

2. **Allocate an empty array**: An empty array `updated_quantile_distribution` is allocated to store the distribution across quantiles resulting from the application of the elasticity.
3. **Calculation of updated distributions**: The function iterates over each quantile to calculate the updated distribution. For each quantile, the updated share is calculated by dividing the scaled income share for that quantile by the total sum of scaled income shares for all quantiles.

$$\text{updated\_quantile\_distribution}[t, c, q] = \frac{\text{scaled\_shares}[t, c, q]}{\sum_{q=1}^{\text{nb\_quantile}} \text{scaled\_shares}[t, c, qq]} \tag{94}$$

Return The function returns the array `updated_quantile_distribution`, which contains the updated quantile share distribution for each country, reflecting the impact of income elasticity on the distribution of economic costs or damages.

14.1.2 `linear_tax_trajectory`

This function calculates a linear carbon tax trajectory. It assumes that a carbon tax is $0 in the first period and increases linearly until the end of the specified period.

Function arguments

- `tax_start_value` (Real): The initial value of the carbon tax (in 2017 US dollars per tonne of CO2).
- `increase_value` (Real): The annual tax increase value. By default, it is equal to `tax_start_value`, which means that the tax increases by its initial value each year.
- `year_tax_start` (Int64): The first year of the tax trajectory. The tax starts at 0 in this year and increases to `tax_start_value` the following year.
- `year_tax_end` (Int64): The last year for which to calculate the tax.
- `year_step` (Int64): The step in years between two tax values. The default is 1, indicating an annual increase.
- `year_model_end` (Int64): The end of the model. If it is less than `year_tax_end`, the last tax value is repeated up to this year. By default, it is set to 2300.

Description The `linear_tax_trajectory` function generates a carbon tax trajectory which starts at $0 and increases linearly each year according to the `increase_value` specified, from year `year_tax_start + 1` to year `year_tax_end`. After `year_tax_end`, the tax value remains constant until `year_model_end` if specified.

```julia
function linear_tax_trajectory(;tax_start_value::Real, increase_value::Real=tax_start_value,
 year_tax_start::Int64, year_tax_end::Int64, year_step::Int64=1, year_model_end::Int64=2300)

for t in [year_tax_start+1:year_step:year_tax_end]
    tax_values = tax_start_value + increase_value * (t-(year_tax_start+1))

    full_tax_path = [0; tax_values; fill(tax_values[end], year_model_end - year_tax_end)]
```

```
    return full_tax_path
end
```

For $t \in$ [year_tax_start $+ 1$: year_step : year_tax_end]:

$$\text{tax\_values}[t] = [\text{tax\_start\_value} + \text{increase\_value} \times (t - (\text{year\_tax\_start} + 1)) \tag{95}$$

$$\text{full\_tax\_path} = [0; \text{tax\_values}; \text{fill}(\text{tax\_values}[\text{end}], \text{year\_model\_end} - \text{year\_tax\_end})] \tag{96}$$

Return The function returns a vector containing the complete trajectory of the carbon tax, from the start year to the end year of the model.

Example of use

```
linear_tax_path = linear_tax_trajectory(tax_start_value=10, increase_value=5,
year_tax_start=2020, year_tax_end=2050, year_step=1, year_model_end=2100)
```

14.1.3 `save_nicev2_results`

Description This function creates a directory of files to store the results of a model, dividing the model output by global, regional and quantile levels. It also manages the recycling of CO2 tax revenues.

Arguments

- `m::Model` : A version of the NICE model.
- `output_directory::String` : The path to the directory where the model results will be saved.
- `recycling_revenue::Bool=true` : Indicates whether CO2 tax revenues are recycled (`true`) or not (`false`).
- `recycling_type::Int64=0` : The type of revenue recycling.
- `result_year_end::Int64=2100` : The end year for the results.

Process The function begins by creating sub-directories to store the results, with and without revenue recycling. The type of revenue recycling is determined by the `recycling_type` argument, which influences the name of the sub-folder.

The function then saves different types of results in the appropriate directories:

- **Global Output**: Includes files such as global temperature, global gross GDP, CO2 emissions, etc.
- **Regional Output**: Includes files such as consumption per capita per region, net GDP per capita per region, etc.
- **Output by Country**: Includes files such as gross GDP per country, consumption per country, population per country, etc.
- **Output by Quantile**: Includes files such as CO2 tax distribution, per capita consumption, consumption after damage and abatement, etc.

Example of use

```
m = load_modele_nice() # Hypothetical function to load a NICE model
output_directory = "path/to/folder/results"
save_nicev2_results(m, output_directory, revenue_recycling=true, recycling_type=1)
```

14.1.4 `get_global_mitigation`

This function calculates the global CO2 mitigation rate for a given model with a specific mitigation policy, compared with a reference scenario with no CO2 policy (business as usual, BAU).

Arguments

- `m_policy` (`Model`): A model representing the economy or environmental system with a CO2 mitigation policy applied.
- `m_bau` (`Model`): A model representing the reference scenario (business as usual) without a CO2 mitigation policy.

Description The function performs the following operations:

1. **Calculates the global industrial CO2 emissions** for the `m_policy` model which includes a specific mitigation policy.

$$\text{global\_emissions\_policy} = \sum(\text{m\_policy}[:\text{emissions}, :\text{E\_Global\_gtco2}], \text{dims} = 2) \tag{97}$$

2. **Calculation of global CO2 emissions without policy (BAU)**: Calculates global industrial CO2 emissions for the `m_bau` model, which represents a scenario without a CO2 emissions mitigation policy.

$$\text{global\_emissions\_base} = \sum(\text{m\_bau}[:\text{emissions}, :\text{E\_Global\_gtco2}], \text{dims} = 2) \tag{98}$$

3. **Calculation of the mitigation rate**: Determines the global mitigation rate by comparing the emissions with mitigation policy to the emissions of the BAU scenario.

$$\text{global\_mitigation\_rates} = \frac{\text{global\_emissions\_base} - \text{global\_emissions\_policy}}{\text{global\_emissions\_base}} \tag{99}$$

Return

- `global_mitigation_rates`: A vector containing global CO2 mitigation rates, calculated as the relative reduction in emissions compared with the BAU scenario.

Example of use

```
# Creating the m_policy and m_bau models
m_policy = Model(...) # Model with mitigation policy
m_bau = Model(...) # BAU model

# Calculation of global mitigation rate
mitigation_rates = get_global_mitigation(m_policy, m_bau)
```

15 nice_v2_module.jl

This file defines the `create_nice_v2` function, which creates a versison of the NICE 2020 simulated model by coupling the FAIRv2.0 model to a simple damage function.

15.1 Loading packages and data

The module uses several Julia packages for its operations:

- `Mimi`: A framework for developing and analysing AMIs.
- `MimiFAIRv2`: A specific implementation of the FAIR model within the Mimi framework.
- `Statistics`: Provides basic statistical functions.
- `Random`: Functions for generating random numbers.
- `Distributions`: A package for defining and working with probabilistic distributions.
- `Inequality`: Used to analyse inequalities in data or model results.

Economic data is loaded from an external `parameters.jl` file, located in the `data` folder relative to the module.

15.2 Mimi Model Auxiliary Functions and Components

The module includes auxiliary functions from the `helper_functions.jl` file. The following model components are loaded from the `components` folder:

- `gross_economy.jl`: Defines how the gross economy works before abatement costs or damages are taken into account.
- `abatement.jl`: Manages emission abatement costs and strategies.
- `emissions.jl`: Calculates emissions based on various economic and abatement factors.
- `pattern_scale.jl` : Compute local temperature values.
- `damages.jl`: Estimates the economic damage caused by climate change.
- net_economy.jl‘: Calculates the net savings after taking into account damage and abatement costs.
- `revenue_recycle.jl`: Processes the recycling of revenues generated by taxes or emission permits.
- `quantile_distribution.jl` : Analyses the distribution of impacts or costs across different quantiles of the population.
- `welfare.jl`: Evaluates general welfare, taking into account economic and environmental aspects.

15.3 `create_nice_v2()` function

The function is based on the Mimi-FAIRv2 model with an emissions and radiative forcing scenario defined according to SSP2-45, between 2020 and 2300. 12 RICE regions (`["US", "EU", "Japan", "Russia", "Eurasia", "China", "India", "MidEast", "Africa", "LatAm", "OHI", "OthAsia"]`) and 10 quantiles are created, `c` is defined as the symbol representing the countries (`countries`).

The emissions, abatement and gross economy components are added before the FAIR carbon cycle, and this model (FAIR) is coupled with scaling, regional damages, net production, income recycling, quantile recycling and welfare valuation components, in that order.

By default, the absence of income recycling is defined, and a population matrix is set (time, country). We define the mapping of countries to the RICE and WPP regions and the number of quantiles that will be used.

15.4 Initial conditions

Initial conditions are updated for different environmental cycles in the FAIR model. Each block of code relates to a specific cycle and updates the parameters associated with that cycle.

15.4.1 The `update_param!` function

The `update_param!` function is used for updating model parameters. It generally takes four arguments:

1. `m`: The model to which the parameter belongs.
2. The name of the cycle for which the parameter is being updated.
3. The name of the parameter to be updated.
4. The new parameter value.

15.4.2 Aerosol cycle and temperature

```
update_param!(m, :aerosol_plus_cycles, :aerosol_plus_0, init_aerosol[!,:concentration])

update_param!(m, :aerosol_plus_cycles, :R0_aerosol_plus,
```

```
                Matrix(init_aerosol[!, [:R1, :R2, :R3, :R4]]))
update_param!(m, :aerosol_plus_cycles, :GU_aerosol_plus_0, init_aerosol[!,:GU])
```

Initial concentration (`aerosol_plus_0`), radiative response (`R0_aerosol_plus`), and radiative forcing (`GU_aerosol_plus_0`) of aerosols are updated. The same is done for methane (CH4), carbon dioxide (CO2), fluorinated gases, substances controlled by the Montreal Protocol, and nitrous oxide (N2O). Finally, the initial temperature is defined.

15.4.3 Gross economy

We connect the population parameter `l` to the gross economy, and do the same with total factor productivity, the rate of capital depreciation, initial capital `k0` and the share of income allocated to capital (here `share` is set to `0.3`).

15.5 Abatement : `abatement.jl`

15.5.1 `set_param!` function

The `set_param!` function is used for defining or modifying the value of a specific parameter for a given component of the model.

- The mitigation control regime is defined (1: Global carbon tax, 2: Carbon tax per country, 3: Mitigation rate per country), which is set to 3 by default.
- Set the global carbon tax (`global_carbon_tax`) and the reference carbon tax (`reference_carbon_tax`) to zero for each time period.
- The USA is defined as the reference country.
- Initialise μ to 0 in order to prepare the model to receive specific mitigation rate values for each country and time period.
- Set $\theta_2 = 2.6$ and $\eta = 1.5$.
- The cost of the CO2 removal technology over time is specified with `full_pbacktime`.
- The values of the emission rate σ and the savings rate s are set.
- The population parameter `l` is connected to the mitigation component, so that the value of `l` is the same throughout the model.

15.6 CO2 emissions : `emissions.jl`

We map countries to rice regions with `mapcrrice`, and update the emission rate σ.

15.7 Temperature : `pattern_scale.jl`

Update the β temperature dimensioning parameter according to CMIP projections.

15.8 Damages : `damages.jl`

The damage coefficients $\beta 1$ (= 0.0236), $\beta 2$ (= 2), $\beta 1\_KW$, and $\beta 2\_KW$ are defined.

15.9 Net economy : `net_economy.jl`

We map countries to WPP regions with `mapcrwpp`, update the savings rate s, and connect the population parameter `l`.

15.10 Income recycling : `revenue_recycle.jl`

- Assume a damage-income elasticity of 0.85 (`damage_elasticity_data = 0.85`)
- By default, the proportion of countries' revenues that are recycled globally in the form of international transfers is set at 100% for all countries.
- By default, carbon tax revenues are recycled globally (`switch_scope_recycle = 1`), and all other Booleans are disabled. The share of revenue lost (`lost_revenue_share`) is set to 0.
- Lists of low- and middle-income countries and risk index are defined.

15.11 Quantiles : `quantile_distribution.jl`

- The minimum (647 USD2017 per capita) and maximum (48892 USD2017 per capita) GDP per capita used in elasticity studies are defined.
- Update the intercept and slope parameters for elasticity, which define the relationship between GDP per capita and demand elasticity, essential for modelling how economic changes affect consumption.
- The elasticity of damage and income distribution is configured.
- The share of income recycling for a quantile is uniformly defined as the inverse of the total number of quantiles.
- The poverty line is set at 2.15 USD2017/day.
- The demographic parameters of population and correspondence between countries and WPP regions are connected. The Boolean for activating income recycling (`switch_recycle`) is also connected.

15.12 Welfare : `welfare.jl`

Similarly, we set certain values (number of quantiles and inequality aversion $\eta = 1.5$) and connect the demographic parameters.

15.13 Connection between components

The general syntax for creating a connection is :

```
connect_param!(model_name, :component_requiring_value => :name_of_required_value,
:component_calculating_value => :name_of_calculated_value)
```

- `model_name` : The name of the model, in this case m.
- `:component_requiring_value` : The component which requires a value from another component.
- `:name_of_required_value` : The name of the value required by the component.
- `:component_calculating_value` : The component which calculates the required value.
- name_of_calculated_value‘ : The name of the value calculated by the component.

15.14 Components

- `I` is calculated by `neteconomy` and is used by `grosseconomy`.
- `YGROSS` is calculated by `grosseconomy` and is used for `abatement`, `emissions` and `neteconomy`.
- μ is calculated by `abatement` and used for `emissions`.
- `E_Global_gtc` is calculated by `emissions`, is required under the name `E_co2` and is used for `co2_cycle`.
- `T` is calculated by `temperature`, is required under the name `global_temperature` and is used for `pattern_scale`.
- `T` is calculated by `temperature`, is required under the name `temp_anomaly` and is used for `damages`.
- `local_temperature` is calculated by `pattern_scale`, is required under the name `local_temp_anomaly` and is used for `damages`.
- `ABATEFRAC` is calculated by `abatement` and is used for `neteconomy`.
- `LOCAL_DAMFRAC_KW` is calculated by `damages` and is used for `neteconomy`, `revenue_recycle`, and `quantile_recycle`.
- `E_gtco2` is calculated by `emissions` and is used for `revenue_recycle`.
- `country_carbon_tax` is calculated by `abatement` and is used for `revenue_cycle`.
- `Y` is calculated by `neteconomy` and used for `revenue_recycle` and `quantile_recycle`.
- `CPPC` is calculated by `neteconomy` and used for `quantile_recycle`.
- `country_pc_dividend` is calculated by `revenue_recycle` and is used for `quantile_recycle`.
- `tax_pc_revenue` is calculated by `revenue_recycle` and is used for `quantile_recycle`.
- `qc_post_recycle` is calculated by `quantile_recycle` and is used for `welfare`.

16 main_runs.jl

This file is the main entry point for running simulations of the NICE 2020 model. It orchestrates the simulation process by loading the necessary components, initialising the model and running the simulations. This file depends on `nice_v2_module.jl` to create a version of the simulated model and on `helper_functions.jl` for other useful functions and to save the results.

It requires the use of the Mimi (to develop and run integrated climate impact assessment models) and MimiFAIRv2 (implementation of the FAIR model within the Mimi framework) packages.

16.1 Creating the base model

`base_model = nice_v2_module.create_nice_v2()` creates a versison of the simulated model by calling the `create_nice_v2` function in the `nice_v2_module` module. This function configures the model with specific economic and climate data, as well as dimensions such as countries, RICE regions, and income quantiles.

- Model dimensions: The variables `nb_steps`, `nb_regions`, `nb_country`, and `nb_quantile` are used to store the number of time steps, regions, countries, and income quantiles in the model. This information is extracted using the dim_keys function with the appropriate dimensions (`:time`, `:regionrice`, `:country`, `:quantile`).
- Sharing of recycled carbon tax revenues: The `recycle_share` matrix is defined to represent the share of recycled carbon tax revenues that each region-quantile pair receives. It is initialised by normalising the value of each country quantile to $\frac{1}{\text{nb quantile}}$.
- Carbon taxation paths :

1. Global CO2 tax trajectory: `global_co2_tax` is defined using the `linear_tax_trajectory` function in the `nice_v2_module`, which creates a linear CO2 tax trajectory from the year 2020 to 2200, starting at 5 currency units per tonne of CO2 and increasing by 8 units each year.
2. Reference CO2 tax trajectory: In a similar way, `reference_co2_tax` defines a CO2 tax trajectory with different starting and increasing values (8.5 and 21, respectively), over the same period.

16.2 Results to be exported

16.2.1 Climate model without mitigation measures

- **Model initialization**: A version of the simulated model `nice_v2` is created without a CO2 mitigation policy. This version is obtained from the `nice_v2_module` by calling the `create_nice_v2()` function. This model includes user-defined specifications but does not implement a CO2 mitigation policy.
- **Parameter update**:
  - The emission control regime is set to `3` using the `update_param!` function. This corresponds to a country-by-country abatement rate regime, as opposed to a global or country-by-country carbon tax.
  - The μ*_input* parameter is updated to be an array of zeros, with dimensions corresponding to the number of steps (`nb_steps`) and the number of countries (`nb_country`). This indicates that no CO2 mitigation measures are applied in the base model.
- **Run the model** : The model is run using the `run(bau_model)` function. This simulates the base model without a CO2 mitigation policy and calculates the corresponding results.
- **Saving results** : The results of the base model are saved to a specified directory (`output_directory_bau`) using the `save_nicev2_results` function in the `nice_v2_module`. The `revenue_recycling` option is set to `false`, indicating that revenue recycling is not taken into account in this scenario.

16.2.2 No redistribution of income generated by environmental policies to households.

- `switch_recycle = 0`: Here, the `switch_recycle` variable is set to `0`, meaning that the income recycling functionality is deactivated (`OFF`).

16.2.3 Documentation of carbon tax scenarios for a 2-degree budget

Simulation of climate policy scenarios aimed at keeping global warming below 2 degrees Celsius.

A. Without revenue recycling The absence of revenue recycling makes it possible to isolate the effect of the carbon tax on emissions without the potential offsetting effects of income redistribution.

i. Global carbon tax without revenue recycling This scenario simulates the effect of a global carbon tax without income recycling. The aim is to assess the impact of a uniform global tax policy on CO2 emissions, without redistributing the revenues generated by this tax to households or businesses.

1. **Initialise the model**: Create a basic version of the `nice_v2` model using the `create_nice_v2()` function in the `nice_v2_module` module.
2. **Scenario configuration** :
   - Set the emission control regime to `1`, corresponding to a global carbon tax.
   - The global carbon tax parameter is updated with the `global_co2_tax` variable.
   - Deactivate revenue recycling using the `switch_recycle` variable.
3. **Run the model**: Simulate the model using the `run()` function.
4. **Save results** : The results are saved in the specified directory without the option to recycle income.

ii. Benchmark US carbon tax without revenue recycling This scenario explores the impact of a carbon tax applied differently in different countries, with a reference tax imposed in the United States, without revenue recycling.

1. **Model initialization**: Creation of a new instance of the `nice_v2` model.
2. **Scenario configuration**:
   - The emissions control regime is set to `2`, indicating a country-specific carbon tax.
   - US reference carbon tax parameter updated with `reference_co2_tax` variable.
   - Revenue recycling deactivated.
3. **Run the model**: Simulate the model for this specific scenario.
4. **Save results**: Results are saved without the revenue recycling option.

B. With total recycling of revenues The aim is to evaluate how recycling the revenues generated by the carbon tax can influence economic and environmental results, particularly in the context of limiting global warming to 2 degrees Celsius.

Configuration of revenue recycling

- `switch_recycle = 1`: Activates the recycling of carbon tax revenues to households.
- `switch_scope_recycle = 0`: Indicates that revenue recycling takes place at national level (`0`) rather than at global level (`1`).
- The other variables (`switch_global_pc_recycle`, `switch_dam_rel_llmic_recycle`, `switch_dam_abs_llmic_recycle`, `switch_risk_llmic_recycle`) are deactivated (`0`), meaning that income recycling does not follow these specific criteria (recycling per person, limited to poor countries, proportional to damage in cost/person or recycling by population-weighted risk index in low- and middle-income countries).

Models launched

- A national carbon tax (`2`, with `1: "global_carbon_tax", 2: "country_carbon_tax", 3: "country_abatement_rate"`), with revenue recycling at national level (`switch_scope_recycle = 0`) and the US tax level as a reference.

- A global carbon tax (1), with revenue recycling at national level (switch_scope_recycle = 0).

C. With partial revenue recycling

i. Recycling of 5% of carbon tax revenues, redistributed in proportion to relative damages, weighted by population in low-income countries (LLMIC). Carbon tax revenues are recycled on a global scale (switch_scope_recycle = 1), to households (switch_recycle = 1) in poor countries according to a relative damage criterion (switch_dam_rel_llmic_recycle = 1), at a rate of 5% (global_recycle_share = 0.05).

ii. Recycling of 5% of carbon tax revenues, redistributed in proportion to absolute damage (in dollars per capita) in low-income countries (LLMIC). Carbon tax revenues are recycled on a global scale (switch_scope_recycle = 1), to households (switch_recycle = 1) in proportion to the damage suffered (in dollars per capita) in low- to middle-income countries (switch_dam_abs_llmic_recycle = 1), at a rate of 5% (global_recycle_share = 0.05).