MIT License

Copyright (c) 2023 JimRuzicka

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Version: 9-12-2023

**Development and running an *ECOTRAN* model:**

This GitHub repository includes the code and the re-gridded ROMS driver files used for analysis in Barceló et al. *in review*, “Non-linear and alternating spatial effects of climate change on the Northern California Current Ecosystem: Insights from a climate and ecophysiology linked end-to-end ecosystem model”.

Repository contents:

1. The ECOTRAN code set for the CAFA Northern California Current project. (C++ files include a version compiled for Mac OS and a raw, uncompiled version to allow compiling on other systems)
2. The excel VisualBasic file (.xlsm) with the mass-balanced *Ecopath* NCC food web and a .csv file with the full food web parameter set. In sub-folder: NCC\_FoodWeb/
3. A compressed zip file containing 1 pre-processed ROMS driver file for the GFDL ESM, on the Northern California Current *ECOTRAN* spatial grid. File is in sub-folder: ROMS\_driverFiles\_NCCgrid/PrePrep\_GFDL\_1980\_09112023.mat. See Pozo Buil et al. (2021) for a full description of the original ROMS product.

The master Matlab script to run ECOTRAN is ECOTRANdynamic\_NCC\_ROMS\_09122023.m. The user will need to change file directory calls to the directory structure of their local computer. This README file describes the general structure of an *ECOTRAN* model, a catalog list of the functions call (Section 5), and a brief description of the major steps in the master script.

The main output result of an *ECOTRAN* simulation is the variable re\_Y describing the production rate of each functional group within each of 15 geographic sub-regions at each time-step (structured as a 3D matrix: time X group X sub-region).

**Organization of this README file:**

1. Brief description of *ECOTRAN* models

2: Defining food web parameters

3: Calculation of the main *ECOTRAN* variables EnergyBudget and ConsumptionBudget

4: Generate Monte Carlo food webs

5: Time-dynamic simulations

6: References

This is a very brief practical guide to the *ECOTRAN* end-to-end modeling code. See also the comments within the code files themselves. It is being provided for others to use, adapt to their own model studies, and to expand.

Please also see in the references section these other publications that have used *ECOTRAN*: (Chiaverano et al., 2018; de Haast et al., 2018; Robinson et al., 2014; Robinson et al., 2015; Ruzicka, Brink, et al., 2016; Ruzicka et al., 2012; Ruzicka, Daly, et al., 2016; Ruzicka, Steele, Ballerini, et al., 2013; Ruzicka, Steele, Gaichas, et al., 2013; Steele & Ruzicka, 2011).

NOTE: Please be aware that this code suite is frequently being revised for organizational clarity, robustness, functionality, and error correction. There may be errors that I have not caught or that appear in model scenario situations that I have never tried.

Please address questions to Jim Ruzicka ([james.ruzicka@noaa.gov](mailto:jim.ruzicka@oregonstate.edu)), and I will do my best to provide clarifications.

**1: Brief description of *ECOTRAN* models**

*ECOTRAN* end-to-end (physics-to-fisheries) models usually consist of a food web model and a physical model. There may be more than one food web representing different physically connected sub-regions. The food web model includes all the trophic interactions from primary producers through top trophic level predators and fishing fleets, nutrient dynamics, and detritus dynamics. The physical model describes the physical transport of nutrients, plankton, and detritus between subregions. The physical model may also describe temperature conditions. The geometry of an *ECOTRAN* model may be variable from a single regional box, to a 1-dimensional stack of vertically connected boxes, to a 2-dimensional cross-shelf model, to a 3-dimensional model with several physically connected food webs across a range of latitudes, longitudes, and depths. Model sub-regions may also be connected by migration of living groups.

The food web used by *ECOTRAN* are typically (but not necessarily) founded upon an *ECOPATH* food web model (Christensen & Walters, 2004; ecopath.org) that represents a snapshot in time of energy flow from primary producers to top predators and fleets.

*ECOTRAN* models are based on the transformation of the solution for a system of linear equations describing predation pressure upon all members of a food web, such as solved by the *ECOPATH* software package, into a donor-driven trophic matrix ***Acg*** that maps the fate of all production by groups ***g*** through the food web to consumers ***c*** (Steele & Ruzicka, 2011; Steele, 2009):

(eq. 1)

Where consumption matrix ***Qgc*** defines the rate that each producer ***g*** is consumed by each consumer ***c***, diet matrix ***Dgc*** is the fraction by biomass of each producer ***g*** within the diet of each consumer ***c***, ***qc*** is the total consumption rate of consumer ***c***, and term is the total grazing or predation pressure upon each producer ***g*** by all consumers ***c***. Consumption matrix ***Qgc*** may be solved using *ECOPATH* techniques. Trophic matrix ***Acg*** is expanded to include nutrient and detritus pools and account for the ultimate fate of all biomass consumed by group ***g*** among its predators, among nutrient and detritus pools via feces and metabolic nitrogenous waste excretion, or to detritus as senescence. A model expressed in this format can readily be used to quantify the consequences of changes to community composition (Chiaverano et al., 2018), changes in nutrient subsidies (Treasure et al., 2015), changes in oceanographic regime through coupling with physical models (Ruzicka, Brink, et al., 2016), changes in fishery management policy (Ruzicka et al., 2019), or physiological or diet changes to any functional group.

The *ECOTRAN* state variable ***Xg,d*** is the production rate of each group ***g*** in each spatial subregion ***d*** (Steele & Ruzicka et al., 2011). In time-dynamic simulations (e.g., Ruzicka et al., 2018), the production rate at each time-step (mmole N m-3 d-1) is calculated as the balance of ingestion input, predation losses, metabolic costs, feces and senescence losses, and net physical fluxes (Ruzicka et al., 2016):

(eq. 2)

Term ***rg*** = intrinsic growth rate of group ***g*** (production/biomass, d-1), ***teg*** = transfer efficiency of biomass input to next higher trophic level. Because ***Acg*** accounts for the fate of all ingestion by ***g*** to predators, to feces, to metabolism, and to senescence, ***teg*** = 1 for all groups except for unconsumed, non-metabolized detritus. Term = total ingestion intake rate by ***g***, = total predation rate upon ***g***, ***ig*** = external input rate of each food web driver group ***g*** (nitrate and ammonium), = net physical transport or migration rate of ***g*** across model subregion and depth zone boundaries into subdomain ***d***, = any optionally forced biomass change rate, and = any optionally defined emigration rate (negative for immigration into the subregion). Consumption matrix ***Qgc*** in eq. 2 is recalculated for each group ***c*** at each time-step as:

. (eq. 3)

Term is a temperature response scaling factor for the ingestion rate of group ***c***in subregion***d****.* This is a dome-shaped scaling curve following the Thornton-Lessem approach (Thornton & Lessem 1978) laid out by Megrey et al. (2007). This relationship is effectively the convolution of two sigmoid curves, representing the increasing segment (gcta) and decreasing segment (gctb) of the temperature dependence function. Thornton-Lessem curves are defined by eight parameters: four temperature reference points (***te1***, ***te2***, ***te3***, ***te4***) and four ingestion rate reference points (fractions of maximum ingestion) associated with each temperature (***xk1***, ***xk2***, ***xk3***, ***xk4***). Using notation from Megrey et al. (2007):

***tt5 = 1 / (te2 - te1)*** (eq. 4a)

***t5 = tt5 · log[xk2 · (1.0 - xk1) / (0.02 · xk1)]*** (eq. 4b)

***t4 = exp[t5 \* (Ttimeseries - te1)]*** (eq. 4c)

***tt7 = 1 / (te4 - te3)*** (eq. 4d)

***t7 = tt7 · log[xk3 · (1.0 - xk4) / (0.02 · xk4)]*** (eq. 4e)

***t6 = exp[t7 · (te4 - Ttimeseries)]*** (eq. 4f)

***gcta = (xk1∙t4) / [(1.0+xk1∙(t4-1.0))]*** (eq. 4g)

***gctb = (xk4∙t6) / [(1.0-xk4∙(t6-1.0))]*** (eq. 4h)

***λ = gcta ∙ gctb*** (eq. 4i)

Temperature effects on metabolic rates are incorporated by scaling flows to nitrogenous waste pools ***c*** in trophic matrix ***Acg***. Scaling factor (***S***) is an exponential function of temperature defined by the metabolic rate at reference temperature ***TREF*** and the metabolic rate at a temperature 10℃ higher:

(eq. 5)

where and are the metabolic rates at the reference temperature and at temperature ***T*** of the current time point, respectively. Parameter is the slope of the exponential relationship. By default, ***Q10*** = 2.36, based on Clarke (2004).

Term ***Fcg*** is the functional response term that scales the consumption rate of producer ***g*** by consumer ***c*** (Steele & Ruzicka, 2011; Steele, 2009):

. (eq. 6)

Term ***mg*** is a functional response modifier characteristic of each prey type, and ***qc*** is the ingestion rate of consumer ***c*** at time ***t***. Prey–predator relationships are strictly donor-driven when ***mg*** = 0 (***Fcg*** = 1); that is, the consumption rate of group ***c*** is a constant proportion of producer abundance (actually, producer ingestion rate **)** as defined by . When = 1, the consumer half-saturation ingestion rate equals the initial condition consumption rate that defines the mass-balanced food web. At higher , the consumption rate begins to scale linearly with consumer abundance (actually, ingestion rate ) at ***t*.** This functional response model is analogous to the arena foraging model used by *Ecosim* (Christensen & Walters, 2004; Steele, 2009) in that the response term is a function of consumer abundance rather than producer (prey) abundance.

Physical flux rates of nutrients, plankton, and detritus groups between subregions are described as losses from source subregion ***s*** and gains to destination subregion ***d***:

(eq. 7)

where = net flux of group ***g*** into sub-region ***d***, = net volume flux from subregion ***s*** to subregion ***d***, = biomass density of ***g*** in subregion ***s***, = flux of biomass ***g*** between subregions ***s*** and ***d***, and = volume of destination subregion ***d***. Losses from source subregion ***s*** () are negative.

**2: Defining model parameters**

*ECOPATH*-style mass-balance models are set up within Excel. Excel *VisualBasic* code provides the essential set of *ECOPATH* algorithms needed define a food web model directly without needing to use the complete *ECOPATH* software package. This was made available to us from Kerim Aydin (NOAA AFSC).

Please see the included Northern California food web parameter files, **NCC2\_09032022.xlsm** and **NCC2\_09032022.csv**.

The first five tabs are where the standard ECOPATH parameters are defined: *MAIN*, *Diets*, *Detritus*, *Fishing*, *Discards*.

The next five tabs are where various parameter uncertainty terms are defined for use in generating Monte Carlo models (if wanted, its optional).

The next two tabs are where the results from *ECOPATH* are returned: *MainOutputs*, *Mortalities*.

The final three tabs are for the definition of parameters specific to *ECOTRAN*. 1) *EcotranType* tab is where the type of each functional group is defined by number code, and where aggregated groups can be defined for automated aggregation. 2) *EcotranRecycling* tab is where the fates of detritus and NH4 excretion are defined. There are two detritus types: feces & non-predation mortality. Each detritus type goes to one of two terminal pools: surface or benthic. Note that there may be many detritus pools defined as functional groups in *ECOPATH*, but ultimately, all detritus that is not consumed will wind up in either the terminal pelagic or the terminal benthic detritus pool. Similarly, there are two NH4 pools: pelagic NH4 and benthic NH4. This tab is also where nitrate and ammonium production is partitioned between primary producers (scroll to the left) 3) The *FunctionalResponse* tab is where functional response parameters are defined. There is room for expansion in this tab to include many optional functional response parameters; as of now, code only uses the first column.

After all food web parameters are set up in the excel workbook, there is a big red button to run *ECOPATH* algorithms on the *MAIN* tab (scroll to the right to see the button) to evaluate the thermodynamic balance of the food web. A food web is in “balance” when none of the model’s model’s living groups suffer more predation than their production rate allows. Please see the *ECOPATH* manual for details on the parameters needed to build and evaluate the balance of a food web model (Christensen et al., 2005).

A second big red button on the *MAIN* tab of the excel *VisualBasic* workbook writes the entire food web to a .csv file for use by *ECOTRAN* in the Matlab scripting language (e.g., **NCC2\_09032022.csv**). Note that if Excel *VisualBasic* gives an error during the save process (due to outdated versions of *VisualBasic*), the open and formatted model parameter file can be saved as a .csv file manually.

OPTIONAL: After a balanced model has been generated, there is code available to generate and evaluate Monte Models (see section “4: Generate Monte Carlo food webs”)

**3: Assembly of the main *ECOTRAN* variables EnergyBudget & ConsumptionBudget**

The balanced *ECOPATH*-style food web that was built in the excel *VisualBasic* .xlsm file is exported directly to a .csv file. We now switch to the *MATLAB* scripting language and workspace ([www.mathworks.com](http://www.mathworks.com)).

To run *ECOTRAN* static or dynamic models, making food web plots, or calculating various model metrics, a common set of steps is followed in the *MATLAB* scripts. The following description applies to script ECOTRANdynamic\_NCC\_09122023 where many of these steps are combined. See also section “5: Time-dynamic simulations” for more details and a full list and description of all the functions that are called.

**Step 1 –** The first step is to define the food web parameter file to read, load the parameters into *MATLAB* memory, and automatically prepare parameter variables in *ECOTRAN* format.

For example, the mass-balanced Northern California Current food web was constructed in *VisualBasic* file **NCC\_11242020.xlsm** and the parameters were exported as .csv file **NCC\_11242020.csv**. In the script, the parameters may be loaded into memory from the .csv file or directly from the *VisualBasic* file:

[dat] = f\_readEwEcsv\_10pp\_07072021(**NCC\_11242020.csv**');

Model parameter values (EwEResult) and parameter uncertainty levels (PEDIGREE, as coefficients of variation) are prepared in the script as:

[EwEResult, PEDIGREE] = f\_AggregateBiologicalModel\_02052021(dat);

f\_AggregateBiologicalModel\_02052021 has capacity to aggregate functional groups. But aggregated parameter sets should again be entered into the *VisualBasic* file to confirm mass balance and then treated as a new food web. Functional groups do not need to be aggregated beyond the level at which the *ECOPATH* food web was defined, but the function f\_AggregateBiologicalModel\_02052021 is still needed in order to prepare the *ECOPATH* parameters for later construction of the *ECOTRAN* model.

Step 1a – This step allows the user to specify several simulation run options:

* Define whether to use a C++ or *Matlab* version of the differential equation solver. (only the C++ solver is current).
* Define whether to allow independently defined food web parameter sets for each model subregion or to use the same food web parameter set for all subregions.
* Define whether the model is driven from the physical flux input of nutrients from defined boundary concentrations or from externally provided primary producer biomass and production rates (as from ROMS-NEMUSC biogeochemical model output)
* Define the driver used to calculate initial production rates. (e.g., from ROMS-NEMUSC biogeochemical model output)
* Define the form of the functional response model
* Specify whether or not to generate a new set of Monte Carlo food web parameters, load a pre-generated set of Monte Carlo food webs, or not use alternate food webs
* Specify the physical model to use
* Specify which temperature response functions to apply

**Step 2 –** The second step is to build the end-to-end *ECOTRAN* model:

[ECOTRAN] = ECOTRANheart\_09032021(EwEResult, MonteCarloStore);

Where variable MonteCarloStore may be a stack of previously generated random food web parameter sets. Usually, at this stage MonteCarloStore = [], an empty variable. See section “4: Generate Monte Carlo food webs” for more information on the construction of randomly generated alternate food webs.

The ECOTRANheart\_09032021 code returns the ECOTRAN structure variable. The ECOTRAN structure variable has a lot of information in it, but the main terms are the EnergyBudget, BioenergeticBudget, ProductionBudget, ConsumptionBudget, and five fate\_ variables. These variables describe the static, balanced state of the average trophic exchanges living functional groups and abiotic pools (nutrients, detritus) over time:

EnergyBudget (***Acg***)--- this is the heart of *ECOTRAN*. This matrix defines the fate of all energy (biomass) that enters a functional group box. Groups as producers run across columns. Groups as consumers run down the rows. Each column sums to 1. Note that if there is net immigration of a group into the model domain (emigration fraction is negative), then the entries in all other rows of that column will be greater to include distribution of immigration fate.

* Any group may be referred to in this text as either producer (***g***) or a consumer (***c***) depending upon your frame of reference within the food web.
* Primary producers are treated as consumers of nutrients (Consumer 1 in the example matrix below).
* Fleets are treated as any other predator – except that the column sum of fisheries do not sum to 1. Fleets remove production from ecosystem and only return a portion of that take as discard contribution to detritus.
* Bacteria can be included explicitly as a defined functional group, or bacteria can be defined implicitly (as in the example below) as flow from detritus to NH4 pools.
* Unconsumed pelagic detritus flows to benthic detritus. Column sum of benthic detritus does not need to sum to 1 – as this and fisheries are the ultimate loss pathways out of ecosystem. (alternatively, benthic detritus column sums to 1 but transfer efficiency is < 1)
* Advection losses and gains are not accounted for within the EnergyBudget. These gains & losses are taken or added directly to production rate estimates during model runs. [NOTE: sinking & migration gains and losses also need to be accounted for outside of the EnergyBudget]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | NO3 | pelagic NH4 | benthic NH4 | Consumer 1 | Consumer 2 | Fishery 1 | Fishery 2 | pelagic detritus | benthic detritus |
| NO3 |  |  |  |  |  |  |  |  |  |
| pelagic NH4 |  |  |  |  | 0.05 |  |  | 0.08 |  |
| benthic NH4 |  |  |  |  | 0.2 |  |  |  | 0.25 |
| Consumer 1 | 1 | 1 | 1 |  |  |  |  |  |  |
| Consumer 2 |  |  |  | 0.42 |  |  |  |  |  |
| Fishery 1 |  |  |  | 0.3 | 0.05 |  |  |  |  |
| Fishery 2 |  |  |  | 0.05 | 0.3 |  |  |  |  |
| pelagic detritus |  |  |  | 0.2 |  |  |  |  |  |
| benthic detritus |  |  |  | 0.05 | 0.2 | 0.1 | 0.03 | 0.82 |  |

Table 1. Generic format of an *ECOTRAN* EnergyBudget. Groups act as consumers ***c*** when considering rows. Groups act as producers ***g*** when considering columns.

BioenergeticBudget --- summary of the fate of all energy (biomass) that enters a functional group box. Basically, a simplified, 3-row version of the EnergyBudget. The sum of each column = 1, except for fleets & benthic detritus):

row 1 = production

row 2 = feces

row 3 = metabolism

ProductionBudget --- a more detailed breakdown of row 1 (production) in the BioenergeticBudget. The sum of each column = production.

row 1 = predation (total consumption of each group ***g*** going to all its consumers)

row 2 = eggs (total consumption of each group ***g*** going to eggs, gametes, or live births)

row 3 = senescence (total consumption of each group ***g*** going to senescence “other mortality”)

row 4 = ba (total consumption of each group ***g*** going to biomass accumulation)

row 5 = em (total consumption of each group ***g*** going to emigration)

ConsumptionBudget --- a more concise description of the fate of all consumption given in the EnergyBudget, but also includes biomass aggregation (***ba*** = forced, predetermined biomass growth) and ***em*** (emigration). Each column sums to 1.

row 1 = feces (total consumption of each group ***g*** that is not assimilated and is defined as feces)

row 2 = metabolism (total consumption of each group ***g*** that is excreted as metabolic waste (NH4))

row 3 = eggs (total consumption of each group ***g*** going to eggs, gametes, or live births)

row 4 = predation (total consumption of each group ***g*** going to all its consumers)

row 5 = senescence (total consumption of each group ***g*** going to senescence “other mortality”)

row 6 = ***ba*** (total consumption of each group ***g*** going to biomass accumulation). Can be positive or negative.

row 7 = ***em*** (total consumption of each group ***g*** going to emigration). Can be positive or negative.

fate\_metabolism --- defines which nutrient pool does metabolic waste flow to

fate\_eggs --- defines where does reproductive production flow to

fate\_feces --- defines which detritus pool does non-assimilated waste flow to

fate\_senescence --- defines which detritus pools dead bodies flow to

fate\_predation --- defines which predators predation pressure flows to and in what proportions

**4: Generate Monte Carlo food webs**

(Please note that Monte Carlo analyses were not used in the CAFA ecophysiology project)

Alternate, randomly-generated food webs can be produced from the original mass-balanced “type” model parameter set and defined levels of uncertainty for each parameter. There is also a function for generating alternative predator-prey functional response parameters. See example script ECOTRANdynamic\_NCC\_09122023 and section “5: Time-dynamic simulations” for more details.

**Step 3** - A set of Monte Carlo models may be directly from the EnergyBudget variable (***Acg***). The first step is to define uncertainty values (Coefficients of Variation) for all model parameters and from that information calculate the uncertainty for every cell within ***Acg***. This is done in function f\_E2Epedigree\_08042020.

The second step is to generate a stack of random EnergyBudget matrices. This is done in code f\_E2E\_MonteCarlo\_08042020. Parameter uncertainties (“pedigrees”) may either be defined individually within the *VisualBasic* .xlsm file (e.g., **NCC\_11242020.xlsm**), or they may defined as default values more generally by parameter type in the code.

To briefly summarize what happens here. We first establish uncertainty values about each of the values of the ConsumptionBudget for each model group. These terms define the fate of total consumption: feces, metabolism, egg production, predation, senescence, biomass accumulation, and emigration into/out of model domain. The uncertainties about some of these terms may be well-defined (e.g., assimilation efficiency for feces production) and some are very poorly-defined (e.g., predation and senescence). We then define the uncertainty parameters for all members of the EnergyBudget matrix. The EnergyBudget is essentially a more detailed version of the ConsumptionBudget. The EnergyBudget matrix further breaks down the predation term for each producer group into individual predators that eat that producer (or primary producers as consumers of a nutrient pool). It also breaks down feces and senescence fates into various detritus pools. Once uncertainties (CV) are defined for the ConsumptionBudget and the EnergyBudget, a random version is generated for both of these matrices. The elements of the random ConsumptionBudget are adjusted so that column totals for every model group is equal to ONE. This adjustment is done hierarchically so that well-defined terms (e.g., feces, eggs, metabolism) are minimally adjusted while poorly-defined terms (e.g., predation and senescence) can be more broadly adjusted. Some individual terms may be greater than 1 or even negative (e.g., biomass accumulation, emigration, predation), while physiological terms terms must be between 0 and 1, but the total of each column must sum to ONE. A random EnergyBudget matrix is then generated and the individual elements within the EnergyBudget matrix are adjusted to match the random ConsumptionBudget. For example, all consumer elements in any given column of the EnergyBudget matrix are scaled so that their sum is equal to the predation sum of the ConsumptionBudget. The sum of each column of the EnergyBudget matrix is then also equal to ONE.

The theory behind this method and why it is fast. As each element of the EnergyBudget matrix is mathematically determined by various defined physiological, diet, predation, senescence rate, population growth, and emigration terms and we have an uncertainty value (CV) defined for each of these terms, then we can calculate the uncertainty level for each element within the EnergyBudget matrix (see notes in code file for rules of adding and multiplying uncertainty terms and references). Because the sum of each column within the EnergyBudget matrix sums to one, the model is in thermodynamic balance (see Steele, 2009). This makes the method 1 technique fast. We do not generate a random ECOPATH-style (EwE) model from individual parameters and skip having to evaluate whether it is in mass-balance (all ecotrophic efficiency terms <= 1).

NOTE: You may control whether or not a stack of Monte Carlo models is generated, or have the option to load a pre-generated stack of Monte Carlo models by setting control switches at Step 1a in the ECOTRANdynamic code:

switch\_MonteCarlo = 'MonteCarlo\_build'; generate (and optionally save) a stack of MonteCarlo food webs

switch\_MonteCarlo = 'MonteCarlo\_load' load a saved stack of MonteCarlo food webs

switch\_MonteCarlo = 'MonteCarlo\_TypeModel'; use NO MonteCarlo food webs

NOTE: the top of a stack of Monte Carlo food webs is the “type” model as defined by the parameters in the *VisualBasic* .xlsm file.

**5: Time-dynamic simulations**

This section presents a catalog description of each subfunction that is called, followed by a step-by-step description of the processes that occur in each step.

See section “1. Brief description of *ECOTRAN* models” for a description of the equations. The physical model is described briefly below in terms of the functions used; see also Ruzicka, Brink, et al. (2016) for a description of a 2-dimensional upwelling physical forcing version.

**All time-dynamic *MATLAB* functions, in the order they are called:**

**Main script m-file:**

ECOTRANdynamic\_NCC\_09122023 This calls all required sub-functions to run a complete simulation. (NOTE: I prefer to run the script by cutting and pasting into the Matlab workspace window, each step of the ECOTRANdynamic script, step-by-step, in order)

**Main script sub-functions:**

f\_readEwEcsv\_10pp\_07072021 read the VisualBasic (*ECOPATH*) mass-balanced food web parameters from a .csv file as variable dat. NOTE: use for models with up to 10 (or less) primary producers

f\_AggregateBiologicalModel\_02052021 prepare *ECOPATH* parameter set for work in the *ECOTRAN* code. NOTE: you can aggregate functional groups here, if wanted

f\_calcEE\_12292020 calculate Ecotrophic Efficiency of each functional group to evaluate mass balance

f\_VarianceDivision\_12132018 calculate the variance of one variance term divided by another variance term

f\_VarianceMultiplication\_12132018 calculate the product of two variance terms

ECOTRANheart\_09032021 generate an *ECOTRAN* model

f\_ECOfunction\_09032021 return a single *ECOTRAN* model from 1 "type" *ECOPATH* parameter set or from 1 randomly generated Monte Carlo parameter set

f\_RedistributeCannibalism\_11202019 remove cannibalism terms on diagonal of the diet matrix *Dpc*. NOTE: Cannibalism is directed to additional metabolism and feces production (equivalent to the cannibalism fraction of diet). This is the mechanism for reduced group Transfer Efficiency that J Steele pointed out is required if cannibalism is to be removed from diet matrix.

f\_calcEE\_12292020 calculate Ecotrophic Efficiency of each functional group to evaluate mass balance

f\_calcPredationBudget\_12102019 for each for each producer *p*, calculate the fraction of total predation going to each consumer *c*

f\_E2Epedigree\_08042020 calculate the uncertainty for every cell within the EnergyBudget (*Acp*) from pre-defined uncertainty values for all parameters

f\_VarianceDivision\_12132018 calculate the variance of one variance term divided by another variance term

f\_VarianceMultiplication\_12132018 calculate the product of two variance terms

f\_E2E\_MonteCarlo\_08042020 generate a stack of random EnergyBudget matrices (*Acp*) by drawing from a normal or a uniform distribution about each element of the EnergyBudget. The first model in the stack is the “type” model generated from the VisualBasic mass-balanced food web parameter set.

**Physical model functions:**

f\_OrdinalDate Calculate the ordinal date from date format '01-Jan-1998'. The ordinal date is day of year with January 1 of any year = 1

f\_ECOTRANphysics\_NCC\_ROMS\_08152023 Use ROMS output to drive the *ECOTRAN* model through time. Sub-functions define the *ECOTRAN* spatial grid and aggregate ROMS volume flux output and biogeochemical model output to the *ECOTRAN* grid.

f\_ROMS\_GridPrep\_NCC\_08152023 Map ROMS grid to *ECOTRAN* grid

f\_ROMS\_FluxPrep\_NCC\_08152023 express ROMS fluxes in DESTINY<--SOURCE format on *ECOTRAN* grid

f\_CompactFluxTimeSeries\_11182019 compact physical flux time-series, arranged as 3D matrix (time X source box X destiny box). Also, eliminate source & destiny information for subregions that never communicate

f\_UnCompactFluxTimeSeries\_12112019 UnCompact a previously compacted physical flux time-series to provide IMPORT & EXPORT fluxes for each box and the domain as a whole

f\_calcNetFlux\_12112019 calculate net flux into and net flux out of each model box and across outer domain boundaries

f\_EvaluateFluxBalance\_11262021 examine for flux time-series for imbalances IN & OUT of individual boxes and IN & OUT of the overall domain

f\_LightIntensity\_12112020 Light intensity parameters: instantaneous (W/m2), daily mean averaged across 24 h (W m^-2 h^-1), & daily integrated (W m^-2 d^-1) solar raditation at ocean surface; (vertical vector: num\_t X 1). (Not actually used when driven by ROMS output)

f\_ECOTRANmigration\_NCC\_02222022 area of overlap of neighboring model sub-domains (for ROMS). (NOTE: this only provides the area of contact between spatial domain boxes. This information is not used by this version of the code package, but null variables are still needed for code to run)

f\_DVMsinusoid\_08122021 calculate diel vertical migration (DVM) flux rates between all model domain boxes for each functional group at each time point. Uses a daily sinusoidal migration pattern. (NOTE: Not defined for the NCC model, but null variables are still required to run the code)

**Physiological temperature response functions:**

f\_physiology\_Q10\_12082022 Calculate temperature-dependent Q10 metabolic rate scaling factors for living groups

f\_TLparameterization\_08182023 Prepare the eight required Thornton-Lessem parameters for each living group

f\_ThorntonLessem\_12012022 Calculate temperature-dependent Thornton-Lessem temperature scaling factors for ingestion rate

**Main script sub-functions:**

f\_FunctionalResponse\_MonteCarlo\_05132021 prepare array of vulnerability terms and allows for generation of random functional response terms within a predefined uncertainty level

f\_InitialProductionRates\_02012022 calculate initial consumption rate *q* conditions

f\_WebProductivity\_03272019 calculate consumption rates *q* of all groups under a given driver (e.g., NO3 or primary production); also accounts for defined rates of group production export when running static scenarios. NOTE: despite the function name, consumption rates are calculated, not production.

**Model solver**: The model is run by solving the system of Ordinary Differential Equations (ODE) for each functional group at each time step. See equations in Section “1. Brief description of *ECOTRAN* models”. The solver is coded in C++.

f\_PrepMexODE\_09192022 prepare ECOTRAN variables for using C++ ODE solver mex function (mex = *Matlab* EXecutable function). Pack parameters and drivers along proper dimensions

f\_unspoolMATRIX\_04282020 linearize (vectorize) multidimensional matrices up to 4D for use in C++.

mex\_ECOTRANode\_09182022.mexmaci64 use MATLAB-executable C++ function tosolve the ODE for solution to functional group consumption rates at each time-point; default is for reflective boundary conditions. NOTE: this is compiled for Mac. NOTE: uncompiled code file is mex\_ECOTRANode\_09182022.cpp

**These are the steps that take place in the time dynamic script, ECOTRANdynamic\_NCC\_09122023:**

**Step 1** - Identify and load the *ECOPATH* mass-balanced food web parameter set. Define the .csv model filename. This is the mass-balanced food web constructed using K. Aydin’s *VisualBasic* version of the *ECOPATH* algorithms and exported to .csv format. Models from other sources may be used, but they need to be arranged into the same column format as produced by VisualBasic file (e.g., as produced by **NCC\_11242020.xlsm**, see Section “2: Defining food web parameters”). Note: the file directory paths will need to be updated for the local computer.

Also, there are options to choose for running the model that must be made at this step:

1. switch\_MonteCarlo Generate a stack of random food web models, load a stack of Monte Carlo food web models, or don’t use Monte Carlo food webs at all.
2. switch\_FunctionalResponse constant, default non-linear, or other user-defined non-linear functional response parameters.
3. switch\_Q10 activate or deactivate Q10 metabolic scaling response to temperature
4. switch\_ThorntonLessem activate or deactivate Thornton-Lessem ingestion rate scaling response to temperature
5. switch\_SubModel Choose whether to use the same food web for all spatial sub-regions or to use independently defined webs for each spatial sub-region.

Two functions are called in this step: f\_readEwEcsv\_10pp\_07072021.m reads the *VisualBasic* (EwE) mass-balanced food web model from a .csv file and f\_AggregateBiologicalModel\_02052021.m prepares this file for work in the ECOTRAN code.

NOTES:

* The file directory path to each food web model will need to be updated for the local computer.
* Trophic aggregation of functional groups can be done by the function f\_AggregateBiologicalModel\_02052021.m, but it is not necessary to further aggregate functional groups. IT IS still necessary to run this aggregation function in any case in order to format parameter dimensions to work in *ECOTRAN*. I advise that if automated trophic aggregation is done, the aggregated model be entered into a new *VisualBasic* file, proofed for mass-balance, and exported as a new .csv file.

**Step 2** - Prepare the end-to-end *ECOTRAN* model (E2E). This is the heart of *ECOTRAN* and calls function: ECOTRANheart\_09032021. (See section “3: Calculation of the main *ECOTRAN* variables EnergyBudget and ConsumptionBudget”).

**Step 3** – Optionally, generate or load Monte Carlo models as defined be Step 1. Uses functions f\_E2Epedigree\_08042020 and f\_E2E\_MonteCarlo\_08042020. (See section “4: Generate Monte Carlo food webs”)

**Step 4** – Internal variable bookkeeping.

**Step 5** - Define the transfer efficiencies for each group. Because the EnergyBudget defines the fate of all consumption (rather than only of production), we define the transfer efficiencies to all be 1 EXCEPT for the terminal benthic detritus group in static model analyses. This is a poorly defined term for any system. By practice, we define terminal benthic detritus transfer efficiency to be 0.1; the model is fairly insensitive to all but extreme values. You may adjust TENH4 for food webs run within a given physical setting so that model runs have appropriate *f*-ratios under reflective (or other) ocean boundary conditions.

**Step 6** - Define the physical geometry of the model and prepare physical advection, mixing, temperature, any biogeochemical driver products, and detritus sinking rates.

At step 6a: Define several required parameters:

-- Information for ROMS drivers (for details of the ROMS driver used in the CAFA ecophysiology project, please see Pozo Buil et al., 2021)

-- directory containing ROMS netcdf files

-- Names of netcdf files containing information defining the ROMS grid: depth\_levels\_trimmed.nc and wc12\_grd.nc.

-- List of files representing ROMS output for individual years (e.g. wc12\_avg\_gfdl\_1980\_trimmed.nc)

-- the time range of the run is defined by start and end date.

-- the time-step dt is defined as fraction of 24 hour day (e.g., 24/24 for a 1-day step)

At step 6b: Call the main function that prepares the ROMS physical flux, temperature, and primary production drivers for *ECOTRAN*, f\_ECOTRANphysics\_NCC\_ROMS\_08152023.

-- sub-function f\_ROMS\_GridPrep\_08152023 re-maps the ROMS spatial grid onto the coarser *ECOTRAN* grid. Within the code of this sub-function, the user defines the boundaries of the *ECOTRAN* grid north-to-south by latitude, cross-shelf by bathymetry range, and vertically by meters from the surface.

-- sub-function f\_ROMS\_FluxPrep\_08152023 calculates daily horizontal physical fluxes through the *ECOTRAN* grid faces as the sum of the fluxes through overlapping and fractional fluxes through partially overlapping ROMS and *ECOTRAN* cell faces. Vertical fluxes are re-calculated as required to conserve volume in each *ECOTRAN* grid cell. Temperature and primary production values are calculated as volume-weighted means of the volume overlap of ROMS and *ECOTRAN* grid cells.

-- Additional sub-functions are called to compact the ROMS variable time-series (f\_CompactFluxTimeSeries\_11182019) and to check for conservation of grid cell volume (f\_EvaluateFluxBalance\_11262021).

-- Sub-function f\_LightIntensity\_12112020 provides several estimates of light intensity time-series, but these values are not used when ECOTRAN is driven by primary production rates provided by coupled ROMS-biogeochemical model output.

At step 6c: f\_ECOTRANmigration\_NCC\_02222022 calculates the shared boundary areas between neighboring boxes. These are needed when migration fluxes are calculated. NOTE: Migration is not included in the CAFA ecophysiology study, and these values are all set to 1.

Steps 6d and 6e: prepare physical flux variables for the differential equation solver.

Steps 6f: Define detritus sinking rates (m d-1).

At step 6g: option to define diel vertical migration rates of any functional group here in terms of sub-domain flux velocities at dawn and at dusk. These parameters are used by function f\_DVMsinusoid\_08122021. NOTE: this is not used for the NCC model and all values from the DVM function are set to 1 as a default

**Optional Step 6** **version** – Pre-prepared ROMS time-series for the *ECOTRAN* grid can be loaded directly here rather than having to be repeated each time the model is run. NOTE: Un-comment this version of Step 6 to activate and re-comment the full version of Step 6.

**Step 7** -There is no Step 7 in the CAFA ecophysiology study

**Step 8** - Prepare the external driver time-series. In the CAFA ecophysiology study, *ECOTRAN* is driven by primary production from the ROMS-biogeochemical model output.

**Step 9** - Define functional predator-prey relations. This step also allows for Monte Carlo generation of alternate FunctionalResponseParams via the function f\_FunctionalResponse\_MonteCarlo\_05132021.

NOTE: FunctionalResponseParams is a function of the CONSUMER (not the producer) and needs to be aligned with the consumer ROWS in *ECOTRAN* (not producer columns).

NOTE: to change one half-saturation constant for individual groups, change FunctionalResponseParams([group row number(s)]). A value of 0 is purely donor-driven, a value of 1 is analogous to the non-linear *Ecopath-with-Ecosim* default.

NOTE: (optional) Define Michaelis-Menten functional nutrient uptake rates for primary producers. Michaelis-Menten mechanics are not required. As a default, they are not used.

**Step 10** - calculate physiology temperature response scalers (See Section “1. Brief description of *ECOTRAN* models”).

-- Metabolic rates are scaled exponentially with temperature: f\_physiology\_Q10\_12082022. The Q10 scaling factor is applied in step 13d.

-- Consumption rates follow a dome-shaped temperature response (Thornton-Lessem). Sub-function f\_TLparameterization\_05182023 prepares the eight required Thornton-Lessem parameters for each living group. Sub-function f\_ThorntonLessem\_12012022 calculates temperature-dependent Thornton-Lessem temperature scaling factors for each fish group’s ingestion rate

**Step 11** - This step makes a required adjustment terminal benthic detritus. Removal (sequestration) of terminal benthic detritus is accounted for via emigration (em) and not as senescence. Add senescence term in the ConsumptionBudget (row 5) to the emigration term (row 7) and set the senescence term to 0. Also, (optional) make alternate changes to ConsumptionBudget for scenario testing.

**Step 12** – Define individual sub-region box types to activate/deactivate trophic relationships based on spatial sub-region or depth zone. NOTE: for CAFA ecophysiology study, trophic relationships are active in all boxes.

**Step 13** -

**sub-step 13a** - Cycle through each Monte Carlo model by selecting one *ECOTRAN* EnergyBudget matrix (***Acg***) and ConsumptionBudget matrix at a time. (NOTE: deactivated for the CAFA ecophysiology study. Only the original, defined food web parameters are used.)

**sub-step 13b** - Define individual sub-region food webs. There are two options: use the same food web parameter set for each sub-regions or define different food webs for different sub-regions. (NOTE: for the CAFA ecophysiology project, the same food web is used for all sub-regions.)

**sub-step 13c** - calculate fate\_predation variable for the current model (i.e. budget predation pressure between predators).

**sub-step 13d** - Build a time-series of varying physiologies, bioenergetic budgets, and predation vs senescence mortality rates via changes to the ConsumptionBudget matrix. Also, the Q10 metabolic rate scaler is applied to flows to nitrogenous waste pools at this step.

**sub-step 13e** - (optional) make scenario changes to ConsumptionBudget terms at specific time points and spatial boxes.

**sub-step 13f** - pack parameters needed for the ODE solver into structure variable ODEinput.

**Step 14** - Define initial conditions as rates of biomass (nitrogen) flowing into each model group (i.e. ingestion rates) and pack into ODEinput structure variable. The CAFA ecophysiology project uses primary production from ROMS-biogeochemical model output to drive *ECOTRAN*. Step 14 calls sub-function f\_InitialProductionRates\_11202019.

**Step 15** - Solve the dynamic model as defined by a system of ordinary differential equations (ODE).

OPTION 1: solve the dynamic model using C++ odeint solver. Call matlab-executable function mex\_ECOTRANode\_08222022.mexmaci64.

OPTION 2: - solve the dynamic model using *MATLAB* ODE solver ode23t. Call ODE function f\_ECOTRANode\_08202022.

NOTE: trial-and-error suggests ode23t has a bit better performance than using ODE45

**sub-step 15b and 15c** - unstack output of ODE solver as store\_ProductionRates to retrieve spatial box information and save re\_Y

**Step 16** - save results.

**6: References**

Chiaverano, L. M., Robinson, K. L., Tam, J., Ruzicka, J. J., Quiñones, J., Aleksa, K., Hernandez, F. J., Brodeur, R. D., Leaf, R., Uye, S., Decker, M. B., Acha, M., Mianzan, H. W., & Graham, W. M. (2018). Evaluating the role of large jellyfish and forage fishes as energy pathways, and their interplay with fisheries, in the Northern Humboldt Current System. *Progress in Oceanography, 164*, 28-36.

Christensen, V., & Walters, C. J. (2004). Ecopath with Ecosim: methods, capabilities and limitations. *Ecological Modelling, 172*, 109-139.

Christensen, V., Walters, C. J., & Pauly, D. (2005). *Ecopath with Ecosim: a User’s Guide*. Fisheries Centre University of British Columbia

Clarke, A. (2004). Is there a universal temperature dependence of metabolism? *Functional* *Ecology 18*, 252-256.

de Haast, J., Treasure, A., Ruzicka, J., & Moloney, C. (2018). A donor-driven approach to modelling anchovy-sardine dominance shifts in the southern Benguela ecosystem. *Journal of Marine Systems, 188*, 117-132.

Megrey, B. A., Rose, K. A., Klumb, R. A., Hay, D. E., Werner, F. E., Eslinger, D. L., & Smith, S. L. (2007). A bioenergetics-based population dynamics model of Pacific herring (*Clupea harengus pallasi*) coupled to a lower trophic level nutrient–phytoplankton–zooplankton model: Description, calibration, and sensitivity analysis. *Ecological Modelling 202*, 144–164.

Pozo Buil, M., Jacox, M. G., Fiechter, J., Alexander, M. A., Bograd, S. J., Curchitser, E. N., Edwards, C. A., Rykaczewski, R. R., & Stock, C. A. (2021). A dynamically downscaled ensemble of future projections for the California Current system. Frontiers in Marine Science 8: 612874:18pp.

Robinson, K. L., Ruzicka, J. J., Decker, M. B., Brodeur, R. D., Hernandez, F. J., Quiñones, J., Acha, E. M., Uye, S.-I., Mianzan, H., & Graham, W. M. (2014). Jellyfish, forage fish, and the world’s major fisheries. *Oceanography, 27*(4), 78-89.

Robinson, K. L., Ruzicka, J. J., Hernandez, F. J., Graham, W. M., Decker, M. B., Brodeur, R. D., & Sutor, M. (2015). Evaluating energy flows through jellyfish and gulf menhaden (*Brevoortia patronus*) and the effects of fishing on the northern Gulf of Mexico ecosystem. *ICES Journal of Marine Science, 72*(8), 2301–2312. <https://doi.org/doi>: 10.1093/icesjms/fsv088

Ruzicka, J. J., Brink, K. H., Gifford, D. J., & Bahr, F. (2016). A physically coupled end-to-end model platform for coastal ecosystems: Simulating the effects of climate change and changing upwelling characteristics on the Northern California Current ecosystem. *Ecological Modelling, 331*, 86–99.

Ruzicka, J. J., Brodeur, R. D., Emmett, R. L., Steele, J. H., Zamon, J. E., Morgan, C. A., Thomas, A. C., & Wainwright, T. C. (2012). Interannual variability in the Northern California Current food web structure: changes in energy flow pathways and the role of forage fish, euphausiids, and jellyfish. *Progress in Oceanography, 102*, 19-41.

Ruzicka, J. J., Daly, E. A., & Brodeur, R. D. (2016). Evidence that summer jellyfish blooms impact Pacific Northwest salmon production. *Ecosphere, 7*(4). <https://doi.org/10.1002/ecs2.1324>

Ruzicka, J. J., Kasperski, S., Zador, S., & Himes-Cornell, A. (2019). Comparing the roles of Pacific halibut and arrowtooth flounder within the Gulf of Alaska ecosystem and fishing economy. *Fisheries Oceanography, 28*, 576-596.

Ruzicka, J. J., Steele, J. H., Ballerini, T., Gaichas, S. K., & Ainley, D. G. (2013). Dividing up the pie: whales, fish, and humans as competitors. *Progress in Oceanography, 116*, 207-219.

Ruzicka, J. J., Steele, J. H., Gaichas, S. K., Ballerini, T., Gifford, D. J., Brodeur, R. D., & Hofmann, E. E. (2013). Analysis of energy flow in US GLOBEC ecosystems using end-to-end models. *Oceanography, 26*(4), 82-97. <http://dx.doi.org/10.5670/oceanog.2013.77>.

Ruzicka, J. J., Steele, J. S., Brink, K. H., Gifford, D. J., & Bahr, F. (2018). Understanding large-scale energy flows through end-to-end shelf ecosystems - the importance of physical context. *Journal of Marine Systems, 187*, 235-249.

Steele, J., & Ruzicka, J. J. (2011). Constructing end-to-end models using ECOPATH data. *J. Mar. Syst., 87*(3-4), 227-238.

Steele, J. H. (2009). Assessment of some linear food web methods. *Journal of Marine Systems, 76*, 186–194. <https://doi.org/doi:10.1016/j.jmarsys.2008.05.012>

Thornton, K. W., & Lessem, A. S. (1978). A Temperature Algorithm for Modifying Biological Rates. *Transactions of the American Fisheries Society* *107*, 284-287.

Treasure, A., Ruzicka, J., Moloney, C., Gurney, L., & Ansorge, I. (2015). Land-sea interactions and consequences for sub-Antarctic marine food webs. *Ecosystems, 18*, 752-768. <https://doi.org/DOI>: 10.1007/s10021-015-9860-2