**ECOTRAN**

Version: 5-16-2023

**Development of ECOTRAN models and their analysis:**

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)) or Dylan Gomes ([dylan.ge.gomes@gmail.com](mailto:dylan.ge.gomes@gmail.com)), and we will do our best to provide clarifications.

**Organization of the summary:**

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: Alternate food web scenarios

6: Time-dynamic simulations

7: References

**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 ***Acp*** that maps the fate of all production by groups ***p*** through the food web to consumers ***c*** (Steele & Ruzicka, 2011; Steele, 2009):

(eq. 1)

Where consumption matrix ***Qpc*** defines the rate that each producer ***p*** is consumed by each consumer ***c***, diet matrix ***Dpc*** is the fraction by biomass of each producer *p* 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 ***p*** by all consumers ***c***. Consumption matrix ***Qpc*** may be solved using *ECOPATH* techniques. Trophic matrix ***Acp*** is expanded to include nutrient and detritus pools and account for the distribution of all consumption by group ***p*** between its consumers, between nutrient and detritus pools via feces and ammonium 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.

In time-dynamic simulations with *ECOTRAN* (e.g., Ruzicka et al., 2018), the change in biomass inflow for each functional group ***p*** within model subregion ***d*** via consumption, predation, physical flux, or migration is:

(eq. 2)

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

(eq. 3)

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

. (eq. 4)

Term ***mp*** is a functional response modifier characteristic of each prey type, and ***qc*** is the consumption rate of consumer ***c*** at time ***t***. Prey–predator relationships are strictly donor-driven when ***mp*** = 0 (***Fcp*** = 1); that is, the consumption rate of group ***c*** is a constant proportion of producer abundance (actually, consumption 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, consumption 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.

**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 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 f\_ECOTRANdynamic\_NCC2\_12082022 where many of these steps are combined. See also section “6: 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 **NCC2\_09032022.xlsm** and the parameters were exported as .csv file **NCC2\_09032022.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('**NCC2\_09032022.csv**');

[dat] = f\_readKpath\_05052022('**NCC2\_09032022.xlsm**');

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 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 (***Acp***)--- 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 (***p***) 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 ***p*** 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 ***p*** going to all its consumers)

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

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

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

row 5 = em (total consumption of each group ***p*** 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 ***p*** that is not assimilated and is defined as feces)

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

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

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

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

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

row 7 = ***em*** (total consumption of each group ***p*** 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**

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 f\_ECOTRANdynamic\_NCC2\_12082022 and section “6: Time-dynamic simulations” for more details.

**Step 3** - A set of Monte Carlo models may be directly from the EnergyBudget variable (***Acp***). 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 ***Acp***. 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., **NCC2\_09032022.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 1 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: Alternate food web scenarios**

In addition to alternate ecosystem physical drivers, alternate food web structure “what if” scenarios can be run within a time-dynamic simulation:

* Change biomass and consumption rate of one or more consumer groups
* Change predation pressure on one or more specific producer groups
* Change fishery group landings and/or discard rates

**SCENARIO STEP** - Generate a food web structural scenario. There are many options here. One option is a scaling up of the abundance of one group (e.g., baleen whales) and their consumption of model resources at the expense of all other consumer groups that eat the same things (with indirect effects along all trophic chains).

First, define the consumer group(s) to be modified (modify\_consumer in the code).

Second, define the amount they should be re-scaled by (ScaleFactor). A value > 1 means increase flow to modify\_consumer (and reduced flow to offset\_consumer). A value < 1 means decreased consumption by modify\_consumer. Do not use a negative ScaleFactor.

Third, define “offset” consumers (those consumers that get to eat more or eat less prey because of the forced changes to modify\_consumer.

Fourth, define target producers. You can define whether the modify\_consumer group is changing its consumption of just one, a few, or all of its prey groups.

The food web (EnergyBudget) is modified by code f\_ScenarioGenerator\_10212020.m. All Monte Carlo versions of the food web are similarly modified.

Results may be expressed as the change in group production in the scenario food web relative to the base food web: (Pscenario - Pbase) / Pbase. (Actually, model simulations results are returned as changes in biomass inflow ***q*** to each group)

NOTE: There will be a warning given for an individual Monte Carlo model in the case that one or several prey groups cannot support the changed abundance of the modified consumer:

“-->WARNING in MC-26: maximum scaler exceeded for target producer(s): 14 15”

The modified models will still be in thermodynamic balance but the target change to the consumer biomass will not be realized (because there is not enough food available to support that amount of change).

**6: Time-dynamic simulations**

f\_ECOTRANdynamic\_NCC2\_12082022 is a callable function version of the main script for running time-dynamic simulations. This code calls all required supporting functions and driver files. Use ECOTRANdynamic\_NCC2\_TEMPLATE\_12082022 to define alternate model and scenario settings and make multiple calls to the main script (f\_ECOTRANdynamic\_NCC2\_12082022) for batch processing.

For parallel processing of Monte Carlo models, use ECOTRAN\_DynamicCode\_NCC2\_PARALLEL\_05152023 to define alternate model and scenario settings and make multiple calls to the main script the function f\_ECOTRANdynamic\_NCC2\_PARALLEL\_05152023.

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 (for more detail, see the main text of the Gomes et al. manuscript and Ruzicka, Brink, et al. (2016).

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

**Main script m-file:**

f\_ECOTRANdynamic\_NCC2\_12082022 callable version of the main script

f\_ECOTRANdynamic\_NCC2\_PARALLEL\_05152023 main script for parallel processing

**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\_readKpath\_05052022 (Alternative: read the VisualBasic (*ECOPATH*) mass-balanced food web model directly from the VisualBasic Excel file (.xlsm) as variable dat

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.

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.

f\_ScenarioGenerator\_10212020 use to build food web restructuring scenarios (see section “5: Alternate food web scenarios”)

**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\_NCC2\_upwelling\_05142023 define 2-dimensional,, cross-shelf, 5-box model structure and return Advection and Mixing exchanges between boxes. NOTE: specify physical flux time-series to use: 'Brink\_BUI', 'ERD\_CUTI', 'ERD\_AVG\_CUTI', 'ERD\_CONST', or 'Fake\_Upwelling')

calcur\_res.mat dataset with monthly mean salinity, temperature, and NO3+NO2 concentrations; from NH-Line off central Oregon coast (See Ruzicka et al. Journal of Marine Systems 187 (2018) 235-249)

f\_prep\_ERD\_CUTI\_SELECTYEAR\_10132022 process ERD CUTI [https://mjacox.com/upwelling-indices/] timeseries for upwelling flux cropped to datestart:dateend time period (m2s-1). As an input, this function can trim the above timeseries out of “ALL” years (1988-2021), any individual year (e.g., “2009”), or will generate an average timeseries (one year in length) of all years (1988-2021) with an input of “AVG”. If “AVG” is selected, this one average year will be repeated for the duration of datestart:dateend; the same is true for any individual year, such as “2009”.

f\_smooth moving mean smoothing over provided smoothing\_window (in days before and after time-point)

f\_CompactFluxTimeSeries\_11182019 compact physical flux time-series when arranged as 3D matrix (time X source box X destiny box)

f\_UnCompactFluxTimeSeries\_12112019 un-compact a flux time-series to provide IMPORT & EXPORT fluxes for each spatial box and for the overall spatial domain

f\_CalcNetFlux\_12112019.m calculate net flux into and net flux out of each model spatial domain box and across outer domain boundaries

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

round2 round to a specified number of decimal places

f\_LightIntensity\_12112020 solar light intensity for given location and time. Instantaneous (W m-2), daily mean averaged across 24 h (W m-2h-1), & daily integrated (W m-2d-1) solar radiation at ocean surface

f\_ECOTRANmigration\_NCC\_2D\_09042022 Prepares information needed to calculate migration rates for groups. NOTE: this only provides the area of contact between spatial domain boxes.

f\_CompactFluxTimeSeries\_11182019 compact physical flux time-series when arranged as 3D matrix (time X source box X destiny box)

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]

**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**: There are two options for running the model by solving the system of Ordinary Differential Equations (ODE) at each time step

**C++ solver option**:

f\_PrepMexODE\_08212022 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\_08222022.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\_08222022.cpp

mex\_ECOTRANode\_08222022.mexw64 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 Windows. NOTE: uncompiled code file is mex\_ECOTRANode\_08222022.cpp

**MATLAB solver option**:

f\_ECOTRANode\_2D\_08202022 solve the ODE for solution to functional group consumption rates at each time-point; default is for reflective boundary conditions

f\_PhysicalFlux\_intraODE\_09092019 calculate biomass fluxes IN & OUT of each spatial box and across domain boundaries (mmoles N m-3d-1)

**These are the steps that take place in the time dynamic code:**

**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 **NCC2\_09032022.xlsm**). Note: the file directory path will need to be updated for the local computer.

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

1. switch\_ODEsolver choose C++ or *MATLAB* ODE solver
2. switch\_FunctionalResponse constant, default non-linear, or other defined non-linear functional response parameters.
3. switch\_PhysicalModel switch between physical models
4. 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. (NOTE: set in ECOTRANdynamic\_NCC2\_TEMPLATE\_12082022 or ECOTRAN\_DynamicCode\_NCC2\_PARALLEL\_05152023)
5. switch\_FoodWebScenario Choose whether or not to force changes to food web before dynamic run. (NOTE: set in ECOTRANdynamic\_NCC2\_TEMPLATE\_12082022 or ECOTRAN\_DynamicCode\_NCC2\_PARALLEL\_05152023)
6. 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. (NOTE: set in ECOTRANdynamic\_NCC2\_TEMPLATE\_12082022 or ECOTRAN\_DynamicCode\_NCC2\_PARALLEL\_05152023)
7. switch\_INITIALproduction Several ways to calculate or to load pre-defined initial conditions for all model groups(NOTE: set in ECOTRANdynamic\_NCC2\_TEMPLATE\_12082022 or ECOTRAN\_DynamicCode\_NCC2\_PARALLEL\_05152023)

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.

**SCENARIO STEP -** Force changes to the food web structure (See section “5: Alternate food web scenarios”)

**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, and sinking rate time-series.

At step 6a: Set several required parameters:

-- 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., 3/24 for 3-hour step or 24/24 for a 1-day step)

-- the CUTI upwelling latitude. (set in ECOTRANdynamic\_NCC2\_TEMPLATE\_12082022 or ECOTRAN\_DynamicCode\_NCC2\_PARALLEL\_05152023)

-- local directory paths to the nutrient data file calcur.mat and to CUTI datasets [https://mjacox.com/upwelling-indices/] [https://www.mjacox.com/wp-content/uploads/2023/05/CUTI\_daily.csv]

At step 6b: There is a function specific to each physical setting called at this step (e.g., f\_ECOTRANphysics\_NCC2\_upwelling\_05142023).

At step 6c: f\_ECOTRANmigration\_NCC\_2D\_09042022 calculates the shared boundary areas between neighboring boxes. Needed to calculate migration fluxes in terms of biomass densities. NOTE: values are all set to 1 for the NCC model

At step 6d: physical and migration flux variables between are compacted to eliminate flux information for boxes that never have fluxes. NOTE: at step 6f, there is alternate coding depending on whether you are using the *MATLAB* ordinary differential equation solver or the C++ ordinary differential equation solver. Choice of differential equation solver is set in Step 1.

At step 6g: define the sinking rates of any functional group here (m d-1).

At step 6h: 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

**Step 7** – Assign default place-holder values for physiology modification terms.

**Step 8** - Prepare the external\_driver time-series. This is the external input (of nutrients) that enters the model domain via advection and/or mixing. Regardless of how efficient an ecosystem is recycling nutrients, there will need to be an external input of a driver (e.g., nutrients) to offset various losses from the system (e.g., detritus sequestration, fishing losses).

NOTE: Under the reflective boundary assumption, only specified groups (NO3) are advected into the model domain, and the code eliminates advection of non-NO3 groups (or of empty water) into the domain. The reflective boundary assumption is that biomass of non-NO3 groups is the same on either side of model domain outer boundaries.

**Step 9** - Define functional predator-prey relations. This step 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 linear, a value of 1 is analogous to the non-linear Ecopath-with-Ecosim default.

**Step 10** – (optional) Define Michaelis-Menten functional predator-prey relations for primary producers. NOTE: Michaelis-Menten mechanics are not required. As a default, they are not used.

**Step 11** - apply model-tuning adjustments to the ConsumptionBudget. Here detritus recycling rates and the nitrification rate are defined.

**Step 12** - Make adjustments to non-predation terms in the ConsumptionBudget. These are weight-specific growth & consumption rates for each group, and they can vary with time if wanted.

**sub\_step 12a** - Make 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.

**sub\_step 12b** - (optional) Make alternate changes to ConsumptionBudget for scenario testing.

**sub\_step 12c** - Activate these code lines for use with phytoplankton Michaelis-Menten nutrient uptake kinetics and/or for linking with canned NPZD plankton model time-series.

**Step 13** - Define individual sub-regional food webs. Definition of food web types in individual sub-regions was originally handled by activating or deactivating individual trophic links in the EnergyBudget and ConsumptionBudget terms (by multiplying by values of 1 or 0). This can also be done through a deliberate definition of food webs for individual sub-regions (see sub-step 14b).

**Step 14** - Cycle through each Monte Carlo model by selecting one ECOTRAN EnergyBudget matrix (*Acp*) and ConsumptionBudget matrix at a time.

**sub-step 14a** - initialize solution time variable, loop through each Monte Carlo model, and select the current Monte Carlo model

**sub-step 14b** - set food web models for each sub-region. There are two options: use the same food web for all sub-regions or use different food webs for each sub-region (see STEP 1).

**sub-step 14c** - calculate fate\_predation variable for the current model (i.e. budget predation pressure between predators).

**sub-step 14d** - build time-series for varying physiologies (or time-series for non-varying physiologies).

**sub-step 14e** - (optional) make scenario changes to ConsumptionBudget or EnergyBudget terms at specific time points, durations, and spatial boxes.

**sub-step 14f** - pack parameters needed for the ODE solver into structure variable ODEinput.

**sub-steps 12i & 12j**- define initial conditions as biomass (nitrogen) flowing into each model group (i.e. consumption rates) and pack into ODEinput structure variable. There are two options:

METHOD 1: use for driving initial model conditions with primary production defined by ECOPATH mean condition parameterization, P = [(P/B) \* B]

Both methods call function f\_InitialProductionRates\_11202019.

**Step 15** - Define initial conditions as biomass (nitrogen) flowing into each model group (i.e. consumption rates) and pack into ODEinput structure variable. There are three options:

METHOD 1: use for driving initial model conditions with primary production defined by *ECOPATH* parameters, P = [(P/B) \* B]

METHOD 2: use for driving initial model conditions based on physical flux of external nutrients

METHOD 3: use for driving initial model conditions with values loaded from an external file. Usually used for cases where different food webs are used for each model sub-region

**Step 16** - 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 16b/c** - unstack output of ODE solver as store\_ProductionRates to retrieve spatial box information and save re\_Y

**Step 17** - save results and loop to next Monte Carlo model.

**7: 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

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.

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>

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