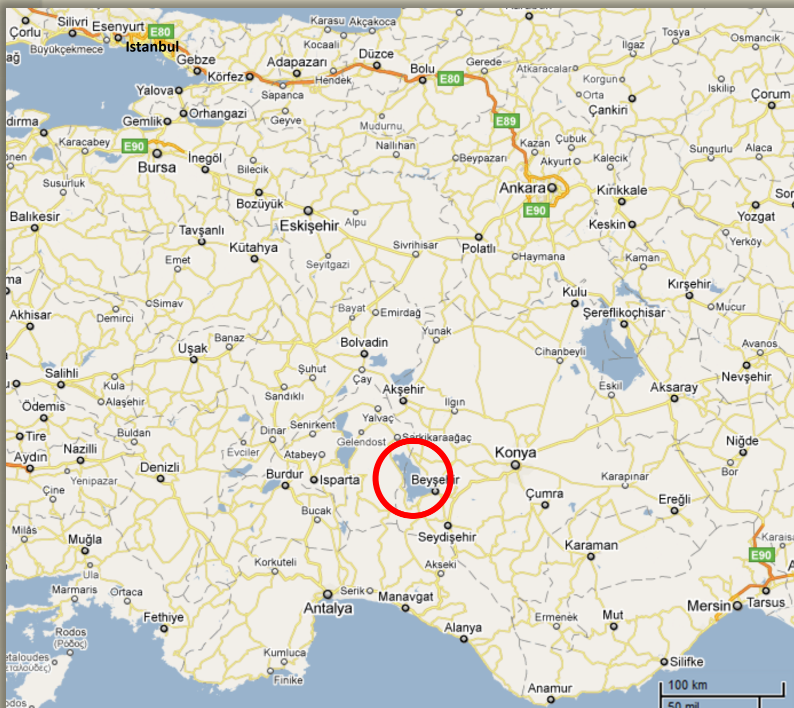
Quick background

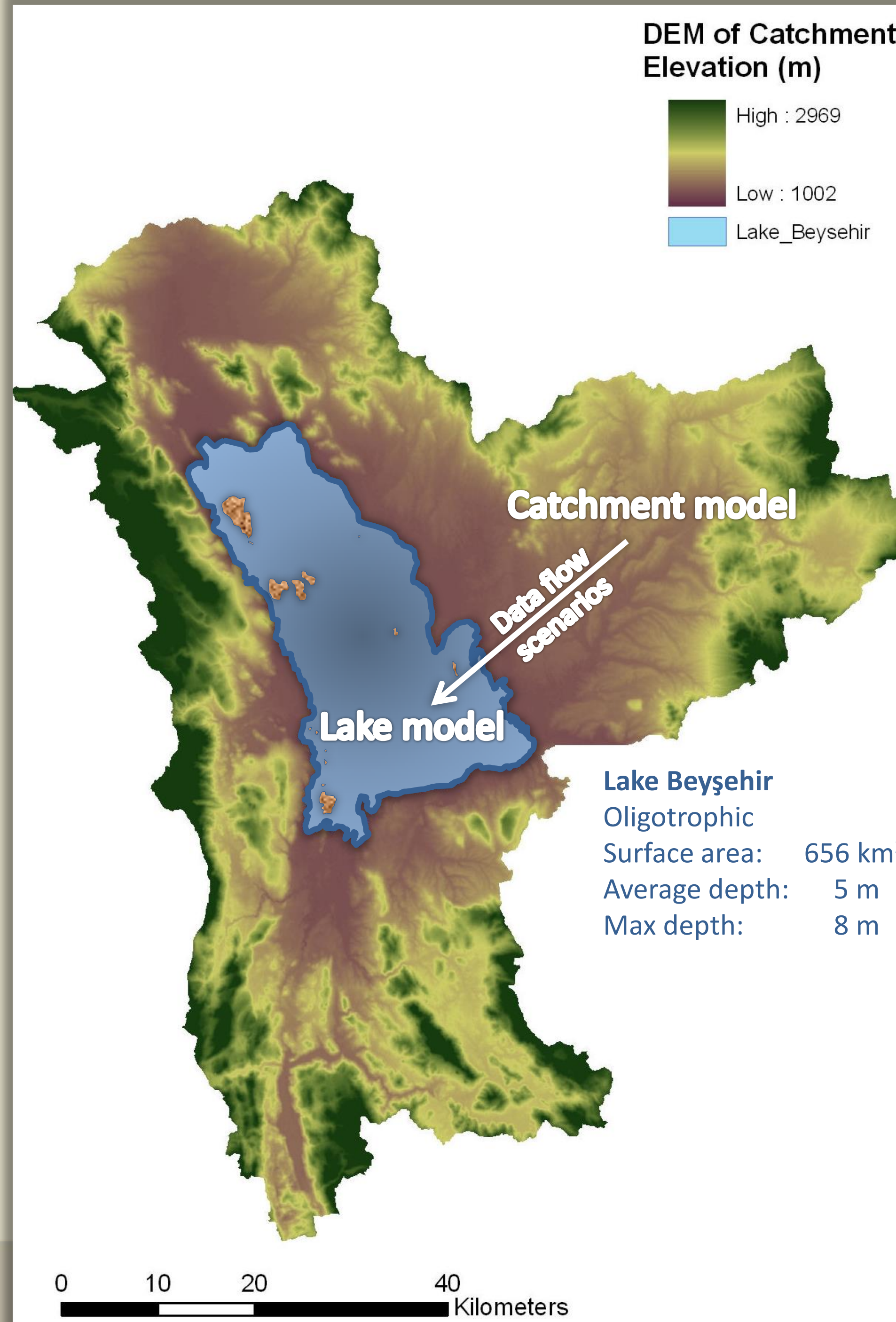
VOLTRES:

Data is being collected, and a series of mathematical models are applied to better understand and quantify the effects of future climate on the hydrology, ecosystem and fisheries of Volta Lake

Department of Bioscience, Aarhus University, Denmark

Typical project example: Modelling in REFRESH WP5





Lake Beyşehir

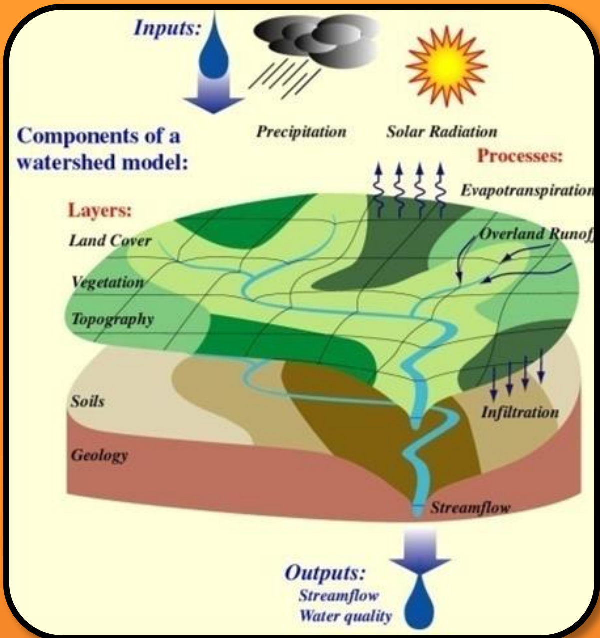
Oligotrophic

Surface area:	656 km^2
Average depth:	5 m
Max depth:	8 m

CATCHMENT MODEL

SWAT: Soil and Water Assessment Tool

Developed at the Grassland, Soil and Water Research Laboratory , Texas A&M University



Input:

- Meteorology (daily averages)
  - Precipitation
  - Solar radiation
  - Air temperature
  - Wind speed
  - Relative humidity
- Topography, *Digital Elevation Model (DEM)*
- Land use, land/cropping practises, point sources
- Soil types
- Stream (and lake) networks

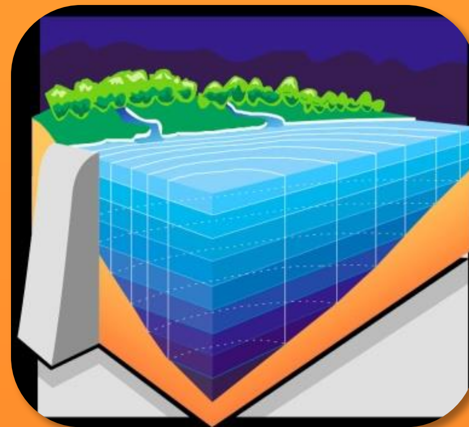
Output:

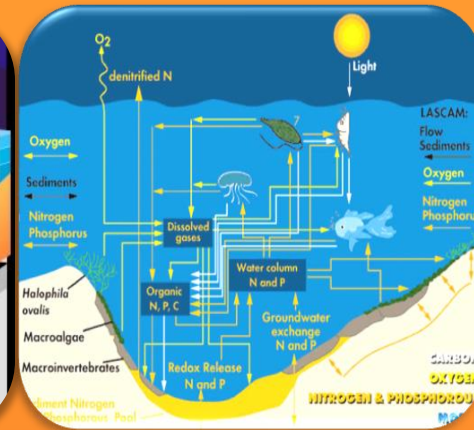
- stream flow
- nutrient concentrations in streams (and lakes)

LAKE WATER QUALITY MODEL

ELCOM-CAEDYM (example):

Estuary and Lake COmputer Model – Computational Aquatic Ecosystem DYnamics Model
Developed at the Centre for Water Research, University of Western Australia





Input:

- Meteorology (daily or hourly averages)
  - Precipitation
  - Solar radiation
  - Air temperature
  - Wind speed
  - Relative humidity
  - (cloud cover)
- Morphology
- Inflow volumes and nutrient concentrations
- Outflow volumes

Output:

- physical dynamics
  - water level
  - temperature distribution
  - circulation/mixing patterns (velocity fields)
- water quality (flexible)
  - dissolved oxygen and nutrient concentrations
  - phytoplankton biomass
  - zooplankton biomass etc. etc…

Ecological Modelling of Large Lakes

DANIDA/CSIR joint workshop – Ghana June 2015
Karsten Bolding and Dennis Trolle

Workshop goals:
Understand classification of models and when to use what.
Make participants able to configure, run and analyze complex mechanistic coupled hydrodynamic bio-geochemical models.

- Participants should learn how to utilize a complex model for scenario simulations (with focus on climate change effects).
- Participants will face, and learn to overcome, the typical file formatting issues, when setting up complex models and using these for scenario simulations.
- Participants will learn basic data handling to extract and present key findings from model simulations.

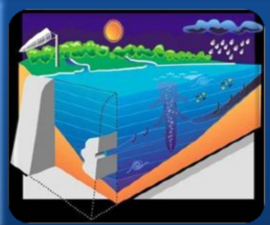
Hands-on:
Use a 1D aquatic ecosystem model to quantify the effects of climate change on the ecosystem of Volta Lake.

- Investigate forcing/input parameters – meteorology and inflow/outflow.
- Investigate sensitivity to meteorological forcing – wind-factor and air temperature.
- How to make the model setup more realistic and improve (hopefully) simulations.

Programme of today

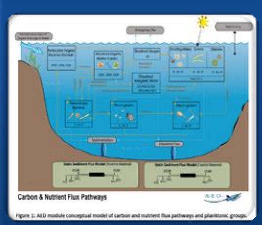
Dennis Trolle

- General introduction to mathematical models



Karsten Bolding

- Software installation
- Model configuration



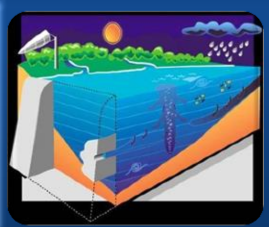
Karsten Bolding & Dennis Trolle

- Exercises

Outline

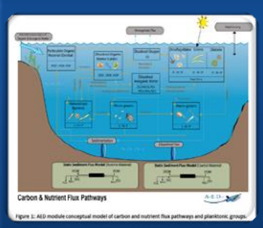
Background

- What are mathematical models and how to use them?
- Software installation and configuration



1D model setup for Volta Lake using GOTM/FABM

- Introduction
- Input
- Configuration

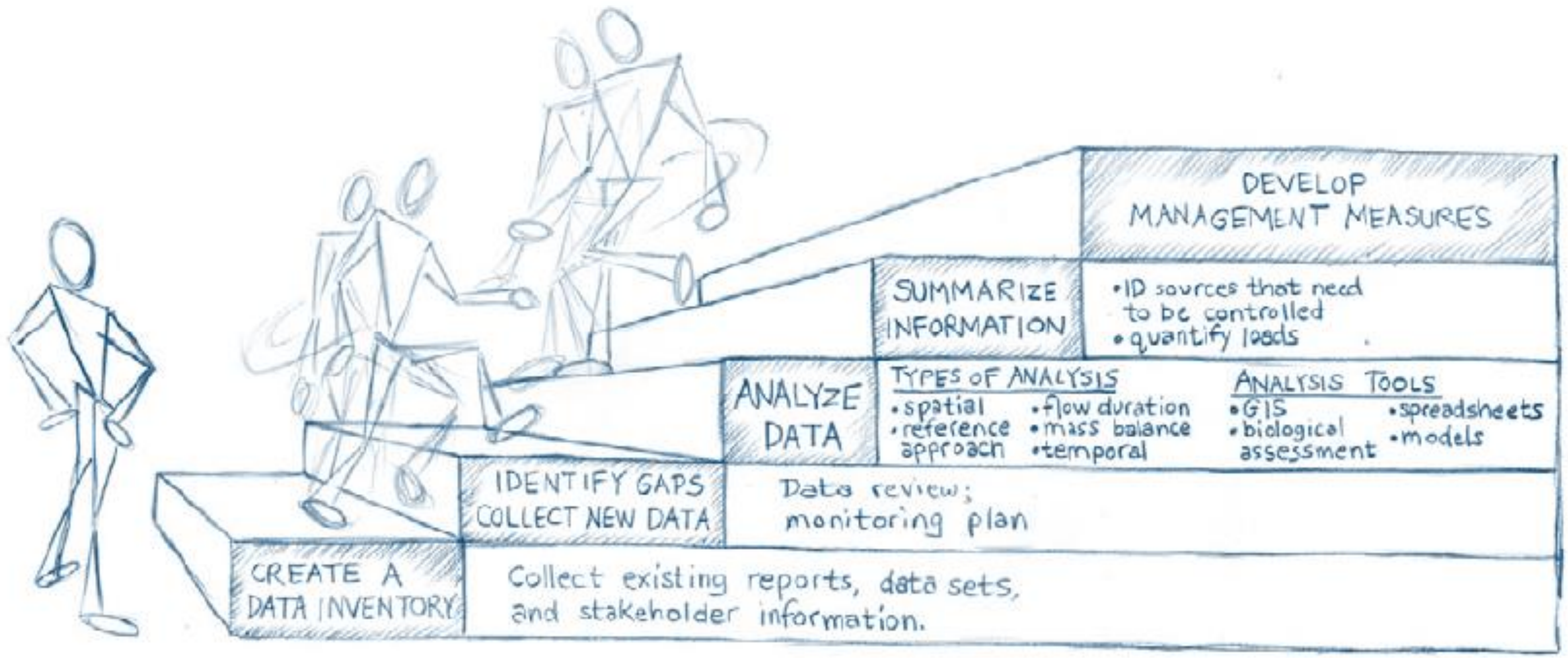


The biogeochemical model: FABM-ERGOM

- Introduction
- Input
- Configuration
- Running GOTM-FABM-ERGOM simulations
- Output/visualization/extracting data
- Assignments

Models can play a key role in management of natural ressources

Management Plans can only be developed if we have data and some kind of models for analysing state and pressures



Introduction to mathematical modelling

What is a mathematical model?

- Models are typically simplified representations of a realworld system (e.g., lake ecosystems, landscapes etc.)
- Models are formulated in equations and/or computer code
- Models are intended to mimic essential features of a system while leaving out inessentials

"Everything (Models) should be made as simple as possible, but not simpler"
(adapted from a quote about theories attributed to Albert Einstein, 1879-1955)

"Essentially, all models are wrong, but some are useful"
(George Box, 1919-2013)

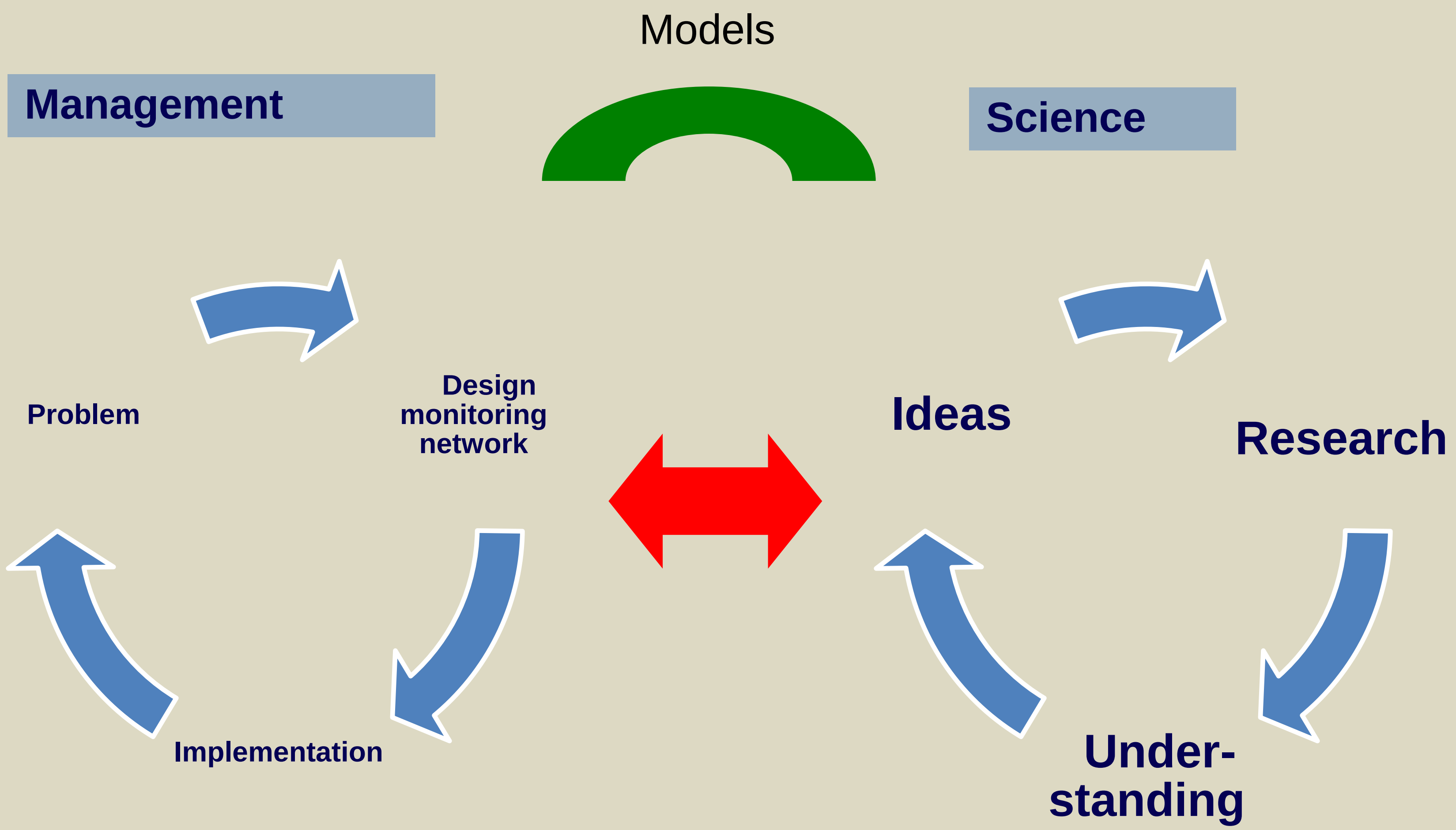
Why use mathematical models?

- Models can be developed and used for **aiding scientific understanding**
  - e.g., hypothesis testing (or creation), where models may especially be useful when they *fail to fit data*, and quantification of processes and interactions

- Models can be used as **virtual experimental laboratories**
  - in some cases realistic experimenting is not possible in the real world (for example due to scale or costs)

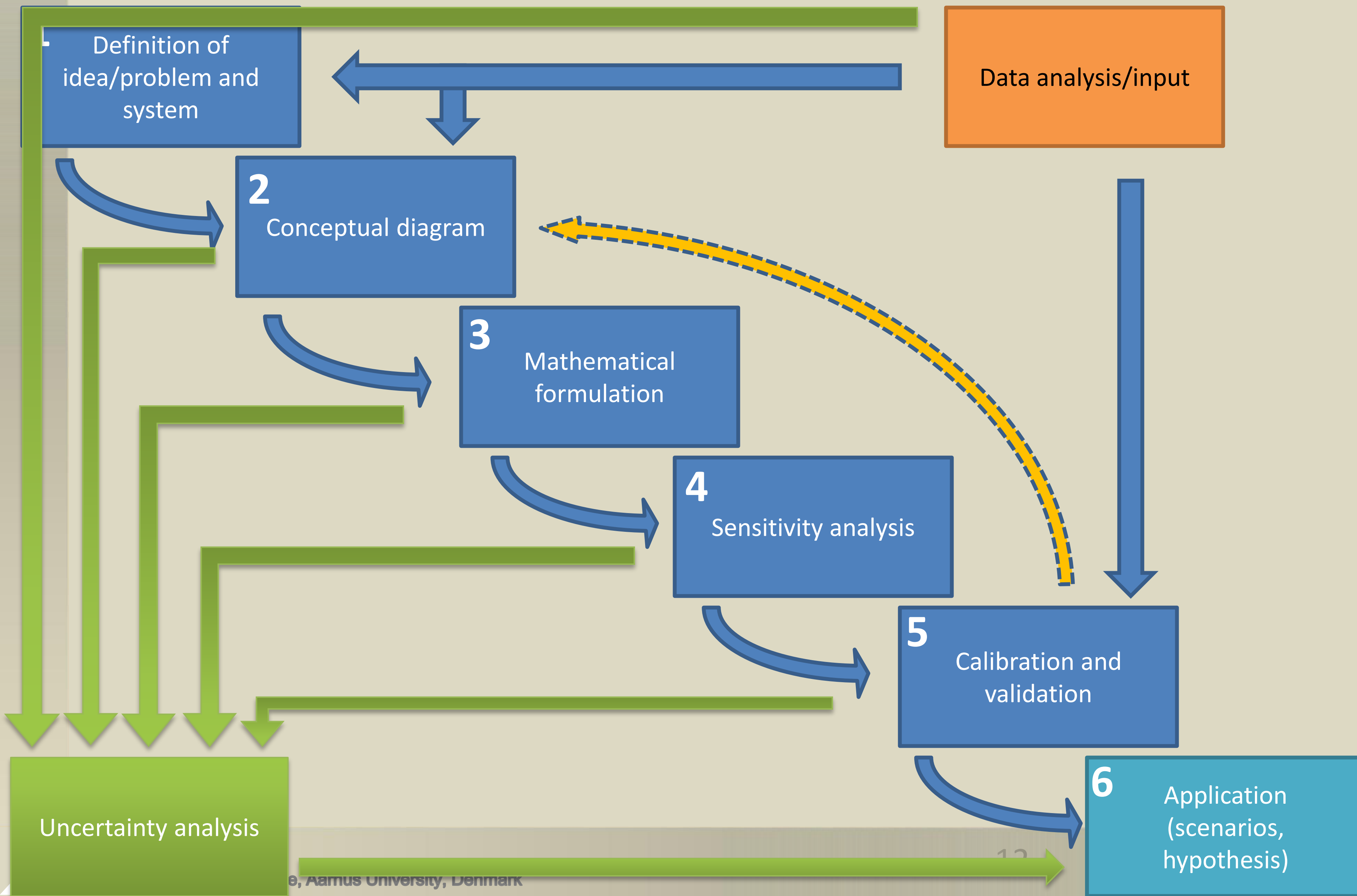
- Models can be used to **manage real world problems** based on scientific understanding
  - Restoration efforts (e.g., for lakes: changing external/internal nutrient loading, changing water level, changing ecosystem balance e.g. through biomanipulation etc.)
  - Changing climate

Models also provide a way for science to support decision making in management



The modelling process

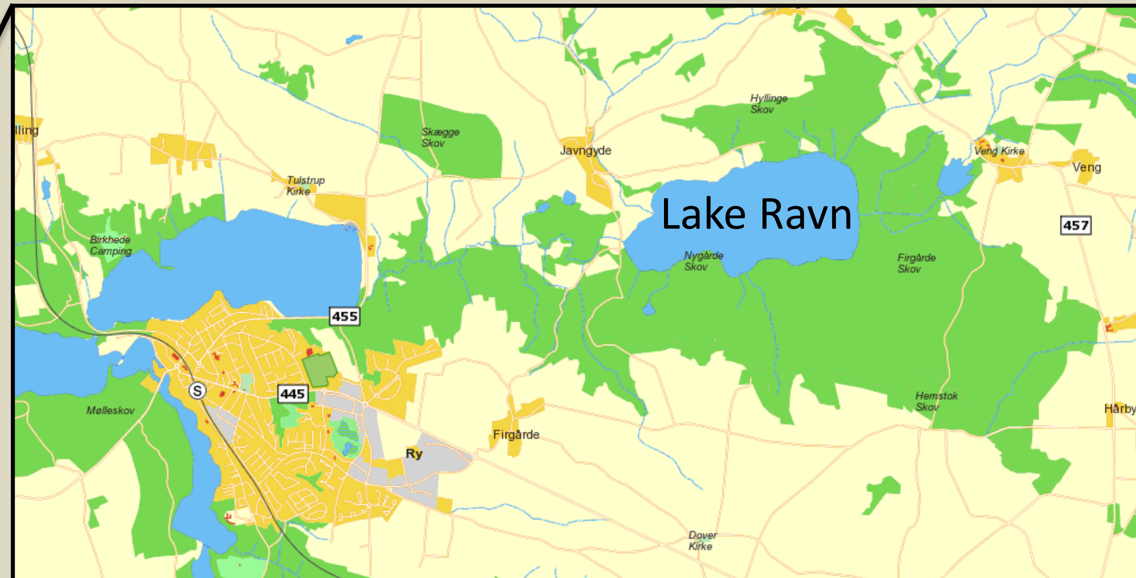
A series of steps are taken to implement an idea first into a conceptual model, and then into a quantitative model



An example...

1 Definition of idea/problem and system

System and problem:
"Lake Ravn is too dirty,
we need to do something."



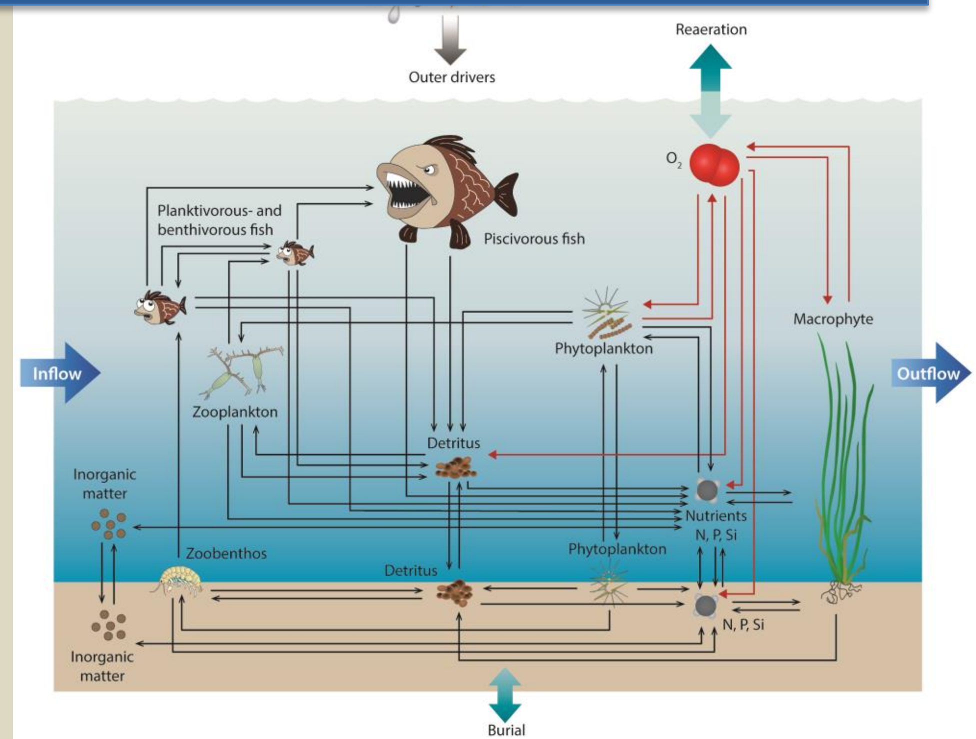
A model will help estimate the needed actions and specifically how much external nutrient loads must be reduced.

An example…

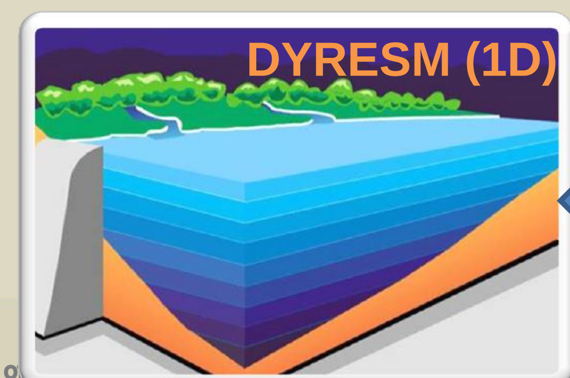
2

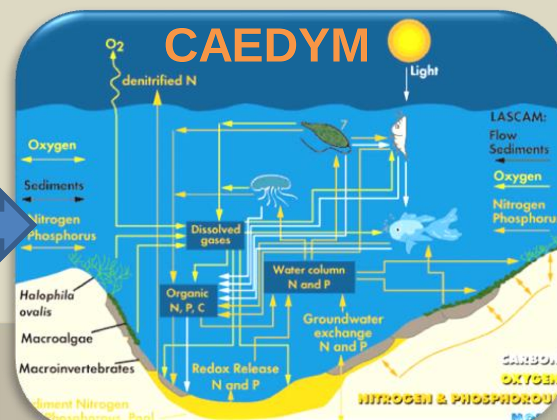
Conceptual diagram

“The conceptual model represents
our ideas about how the system works”

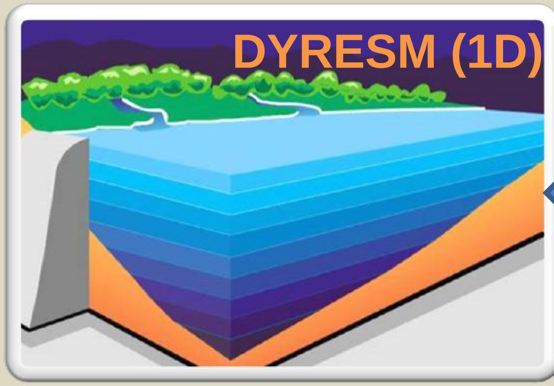


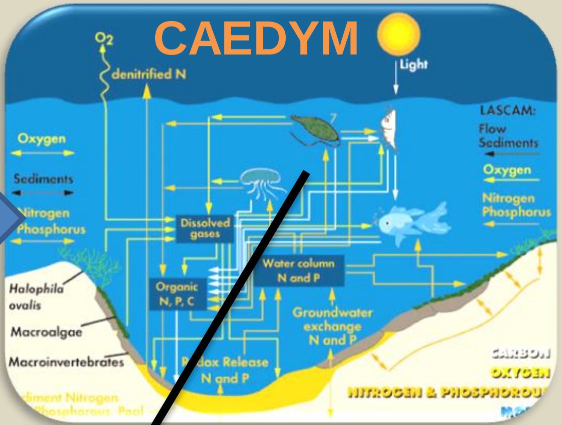
In this case, we could simply modify an already existing ecosystem model (no point to re-invent the wheel) to suit our needs for Lake Ravn





An example...

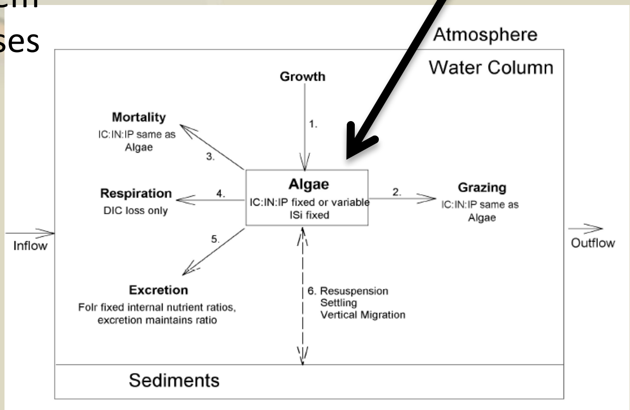




3 Mathematical formulation

Equations can be formulated to describe observed system responses, or to describe individual rates of the processes included in the system

The equations can be studied mathematically either analytically or translated into computer code to obtain numerical solutions for changes in state variables



$$f(N)_a = \frac{NH_4 + NO_3}{NH_4 + NO_3 + K_{N_a}}$$

$$f^{T1}(T) = \vartheta^{T-20} - \vartheta^{k(T-a)} + b$$

Modelling of actual growth rate of phytoplankton (example):

$$\mu_{g_a} = \mu_{MAX_a} \min\left[f(I)_a, f(N)_a, f(P)_a, f(Si)_a^{*}, f(C)_a^{**}\right] f_{A_a}^{T1}(T)$$

$$f(I) = 1 - \exp\left(\frac{-I}{I_k}\right)$$

An example...

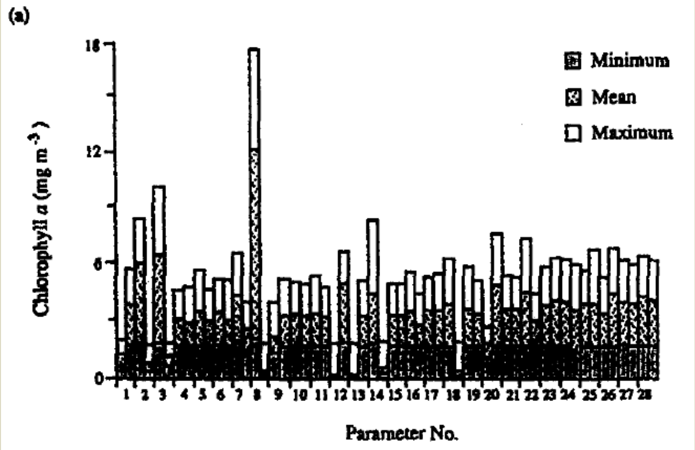
4
Sensitivity analysis

"Sensitivity analysis is the process of determining the significance of one or a combination of parameters in relation to an objective function or a model output"

(this is important because parameters represent processes)

Has typically been done and published several times for an already existing model. However, in the ideal world, this is repeated for each specific model case study (as parameter sensitivity to some extent is case specific). Here example from:

S.G. Schladow, D.P. Hamilton / Ecological Modelling 96 (1997) 111–123



Sensitive model parameters with respect to mean concentration, vertical distribution and temporal distribution of chlorophyll *a* and dissolved oxygen concentrations. Definitions of parameters are given in Table 1 of Part I

	Chlorophyll *a*	Dissolved oxygen
Mean concentration	G_P, k_r, k_m, IP_{min}, UP_{max}, K_P	IP_{min}, K_P, k_b
Vertical distribution	ϑ_P, IP_{min}, UP_{max}, K_P, ρ_P	k_b
Temporal distribution	IP_{min}, UP_{max}, UN_{max}, K_p, k_b	

16

An example…

5
Calibration and validation

"**Calibration** is adjusting model parameters with the purpose of achieving the best simulation match with observations"

This may be achieved by a combination of visual inspection and optimization of an objective function,

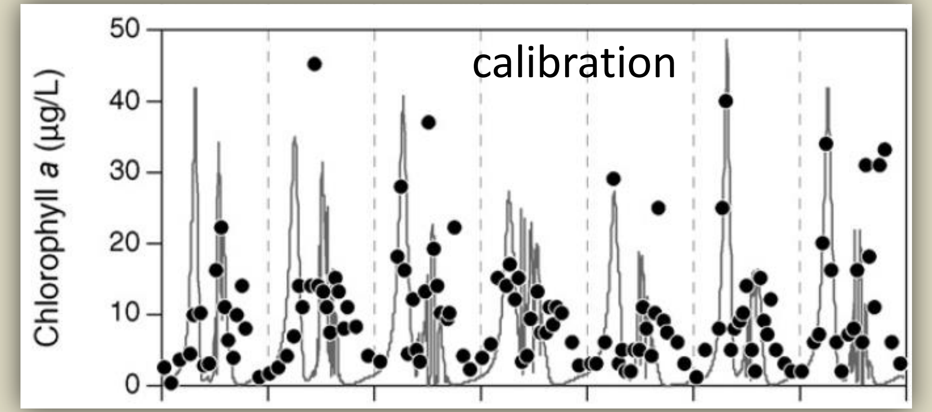
for example, minimize the relative absolute error:

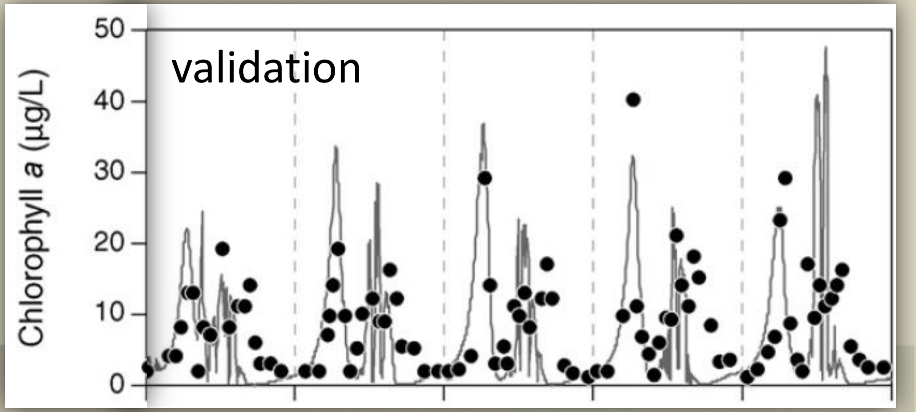
$$RE(Chla) = \frac{\sum_{i=1}^{n} \left| Chla_{simulated} - Chla_{observed} \right|}{\sum_{i=1}^{n} Chla_{observed}}$$

5
Calibration and **validation**

"**Validation** is the process of testing the calibrated parameters with an independent set of data (in time and/or space)"

Model adjustment to Lake Ravn case



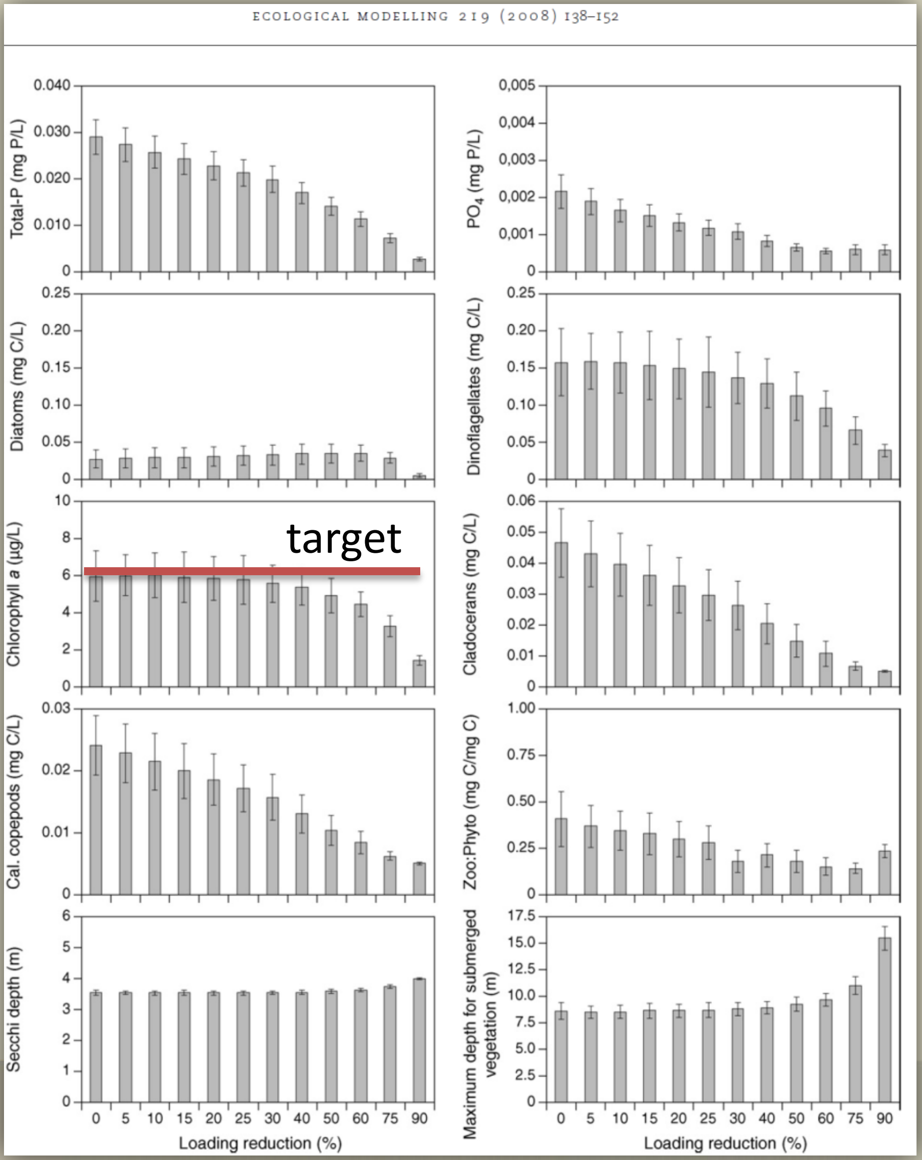


An example...

6 Application (scenarios, hypothesis)

Model scenarios are very useful in testing “what if” questions

Model scenario simulations

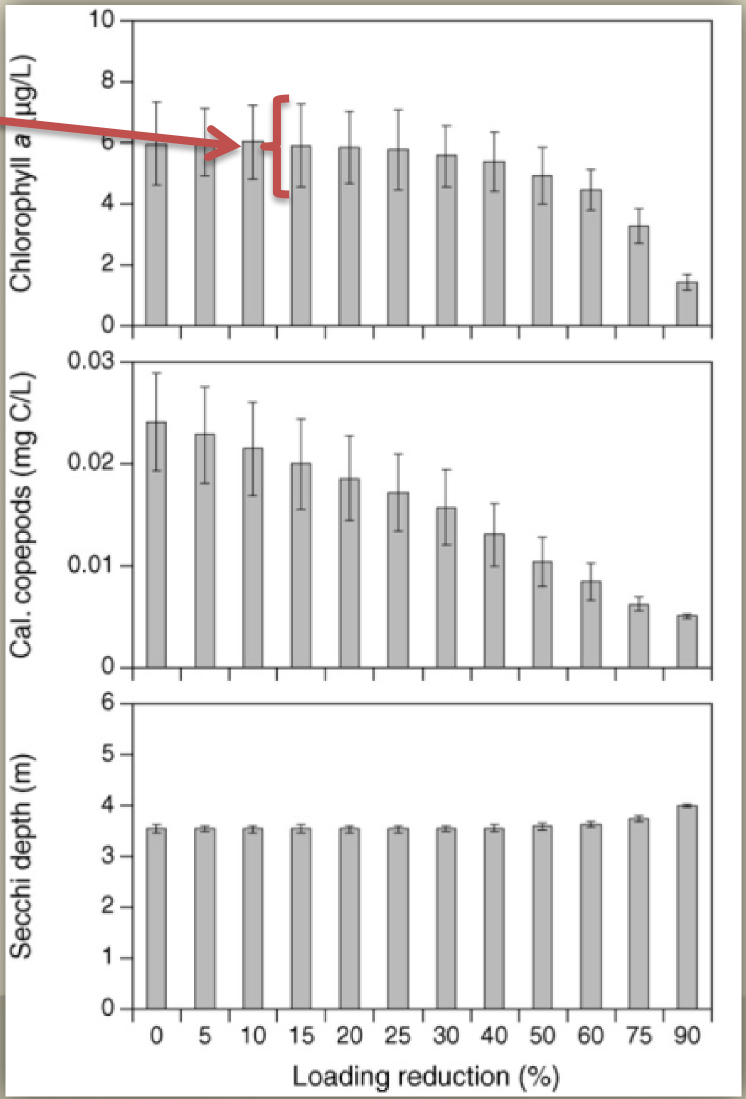


An example…

Uncertainty analysis

Model scenario simulations

"Uncertainty analysis is the process of determining the impact of various uncertainties on the model outputs"

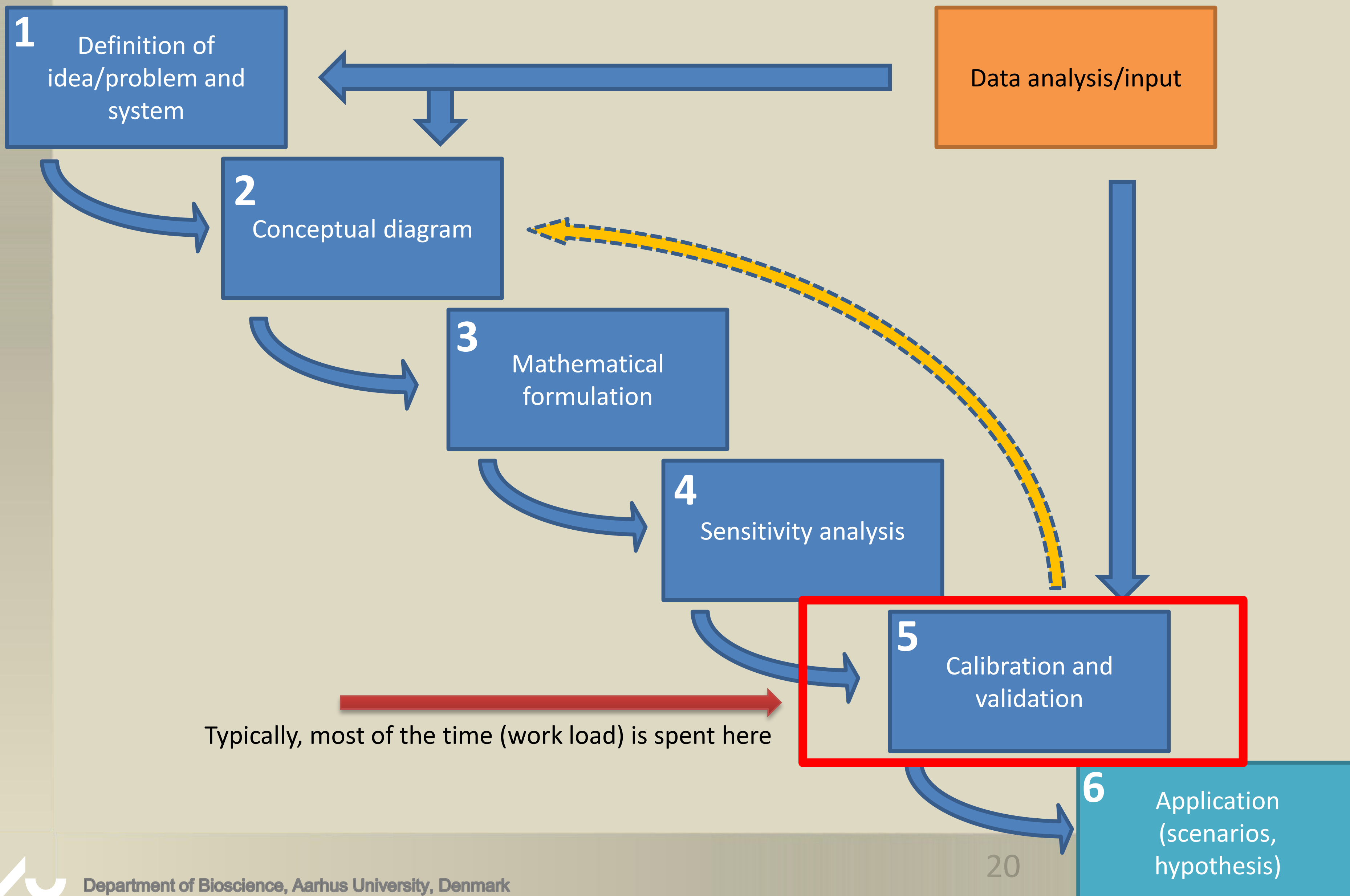


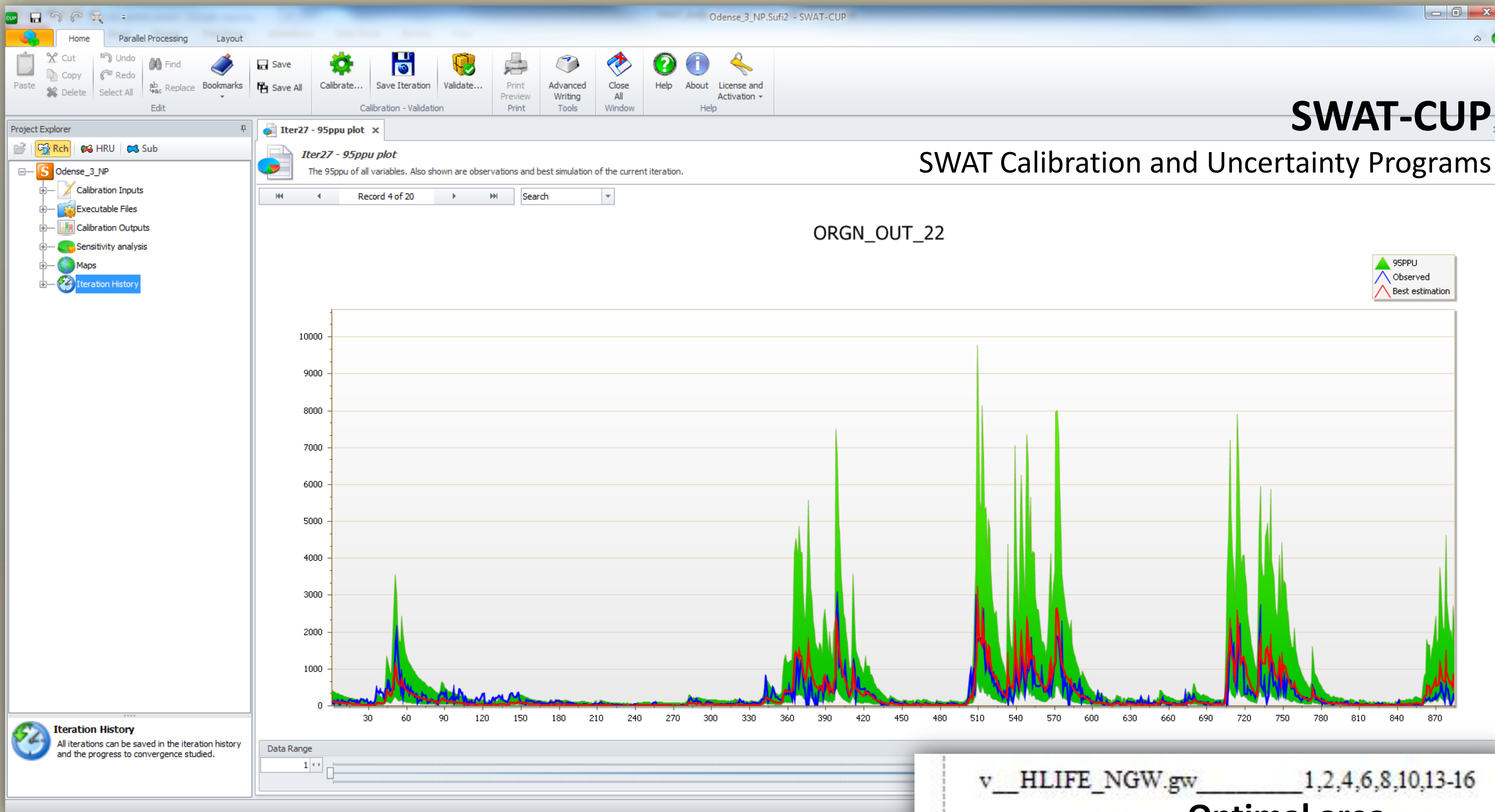
Uncertainties can be classified into categories:

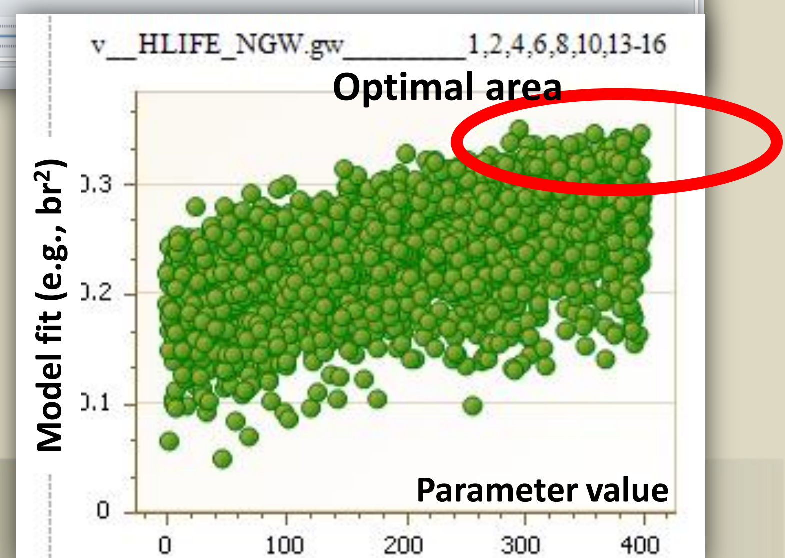
- **Context and framing**, i.e. at the boundaries of the system to be modelled.
- **Input uncertainty** in terms of external driving forces (e.g., runoff and nutrient loads) and data used for calibration.
- **Model structure uncertainty** is the conceptual uncertainty due to incomplete understanding and simplified descriptions of modelled processes as compared to reality.
- **Parameter uncertainty**, i.e. the uncertainties related to parameter values (and non-uniqueness).
- **Model technical uncertainty** is the uncertainty arising from computer implementation of the model, e.g., due to numerical approximations, resolution in space and time, and bugs in the software.

From: J.C. Refsgaard et al. / Environmental Modelling & Software 22 (2007) 1543-1556

The modelling process



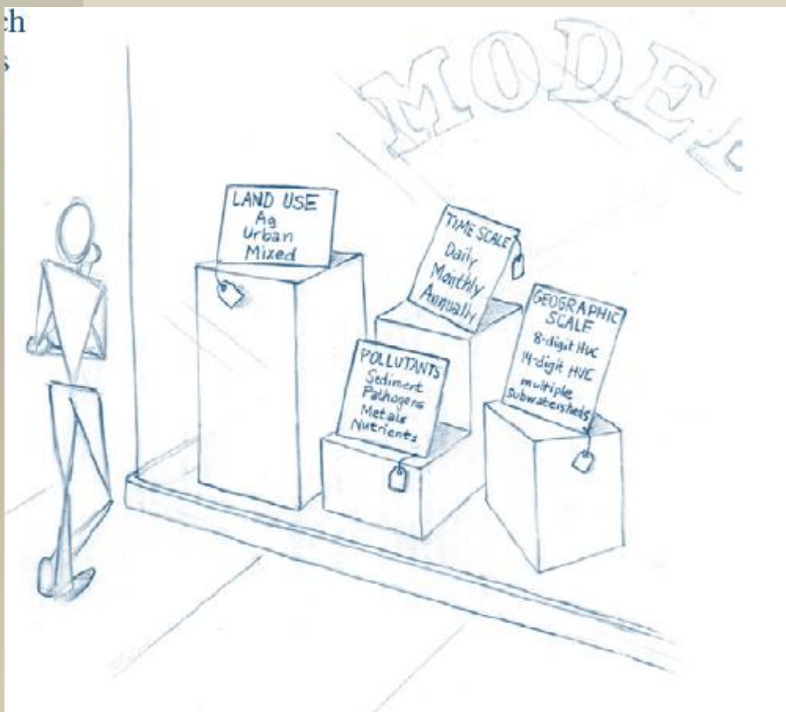




Tools for automatic calibration, sensitivity analysis, and uncertainty analysis are available for some models, but not all.

(and not all techniques take the concept of non-uniqueness into account)

What kind of models should you choose?



Relevance Considerations

- ✓ The model can represent the land uses and processes that are most important in your watershed.
- ✓ The model predicts the pollutants you're concerned about.

Credibility Considerations

- ✓ Model validations have been published in a peer-reviewed journal.
- ✓ The model is in the public domain, and the source code is available on request.

Usability Considerations

- ✓ Documentation, training, and support are available.
- ✓ The model can be run with data that are generally available or data that can be obtained with reasonable effort.
- ✓ The model and user interface are reliable and thoroughly tested.

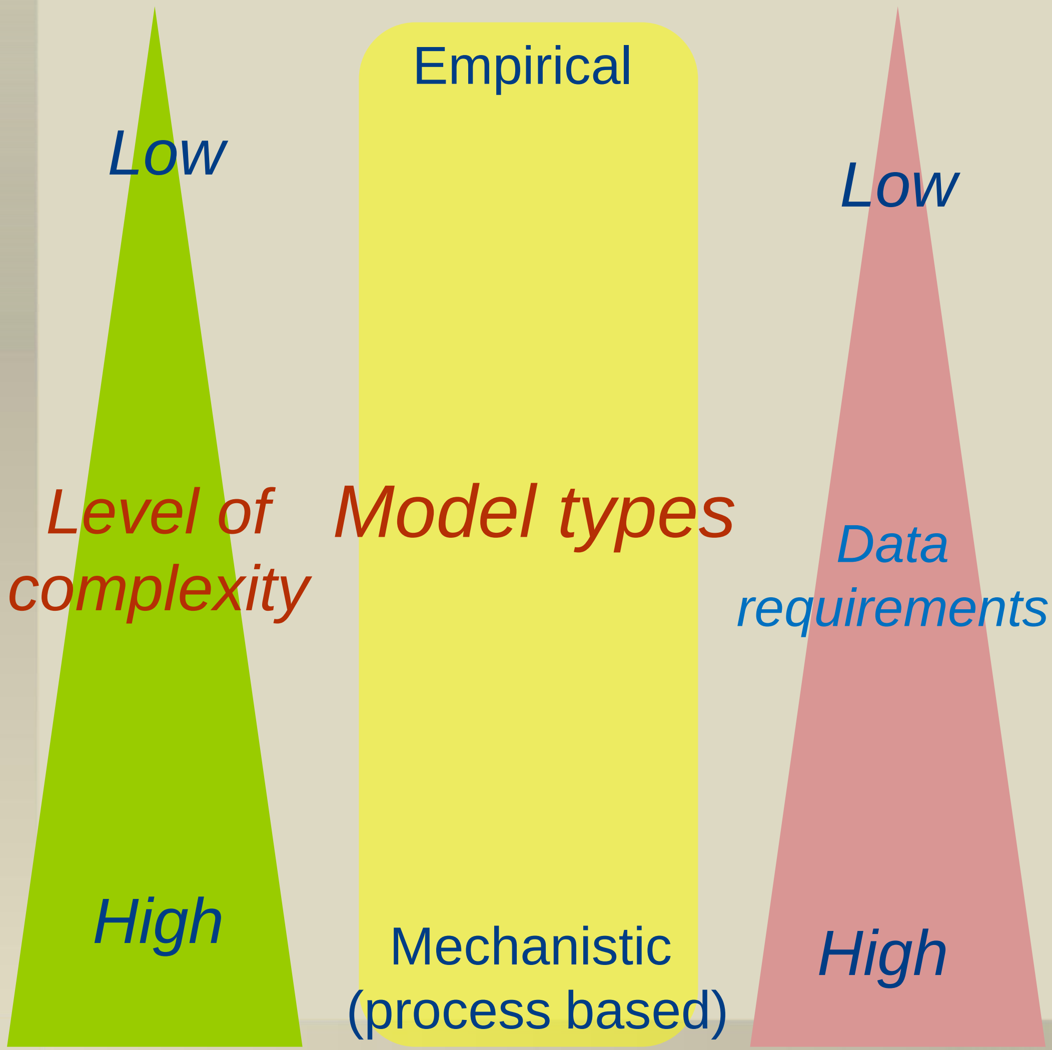
Utility Considerations

- ✓ The model or supplemental tools are able to predict the likely water quality impacts of the land use or management changes you are considering in your watershed plan.

Model classifications

- Models differ widely in complexity (and data requirements) and may be classified in a number of ways

---



Runoff coefficients

Lumped models (e.g., NAM)

Semi-distributed models, based on curve number and linear reservoir (e.g., SWAT)

Fully distributed 3D models (e.g., MikeShe, MODFLOW)

Model classifications

- Models may be classified in a number of ways

Model classification	Empirical	Mechanistic
Empirical (statistical, correlative) versus **Mechanistic** (process based)	Empirical models are usually regression based. They provide a (usually simple) mathematical relationship among a set of measured variables.	Mechanistic model are based on mathematical formulation of processes, that together attempts to describe how a system work.

Model classification	Static	Dynamic
Static (steady state) versus **Dynamic**	Static models are at an equilibrium (or steady state) with forcing data.	In dynamic models, state variables may change with respect to time.

Model classifications cont.

Model classification	Deterministic	Stochastic
Deterministic versus **Stochastic**	If a model contains no random variables, it is considered deterministic. Predictions by a deterministic model under a specifik set of conditions are always exactly the same.	If a model contains one or more random variables, it is considered stochastic. Stochastic model predictions under a set of specified conditions are not always exactly the same, since random variables within the model potentially can take on different values each time the model is solved/run.

Model classification	Simulation	Analytical
Simulation (numerical) versus **Analytical**	Models that must be solved numerically (as many ecological models) are simulation models.	Models that may be solved in a mathematically closed form are analytical. Examples are regression models and simple differential equations

Model classifications cont.

Model classification	Quantitative	Qualitative
Quantitative versus **Qualitative**	Quantitative models lead to a detailed, numerical prediction about model responses.	Qualitative models lead to general descriptions (e.g., indexes) about model responses.

In the literature, authors will typically only list one (or a few) of these model classifications, when describing the properties of a model.

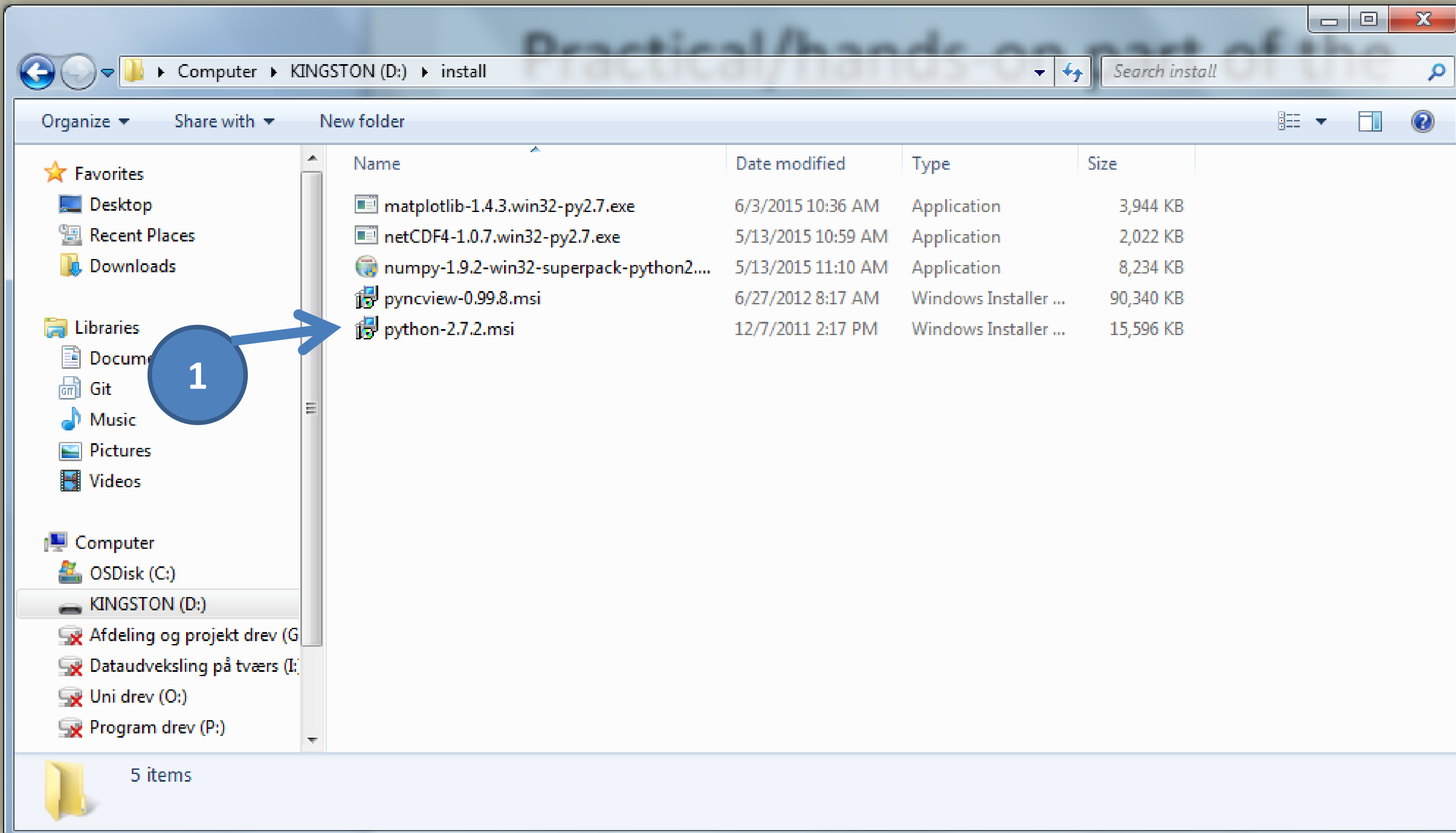
Practical/hands-on part of the workshop

- Copy Ghana folder from USB-disk to your Desktop (8GB)
- We will go through the process of software installation
  - A few different general purpose tools
  - Model specific tools
- We will make experiments with a semi-realistic 1D Volta Lake set-up
  - Learn about necessary input - and configuration files
  - Run the model
  - Evaluate results
  - Modify forcing functions
  - Compare different model simulations

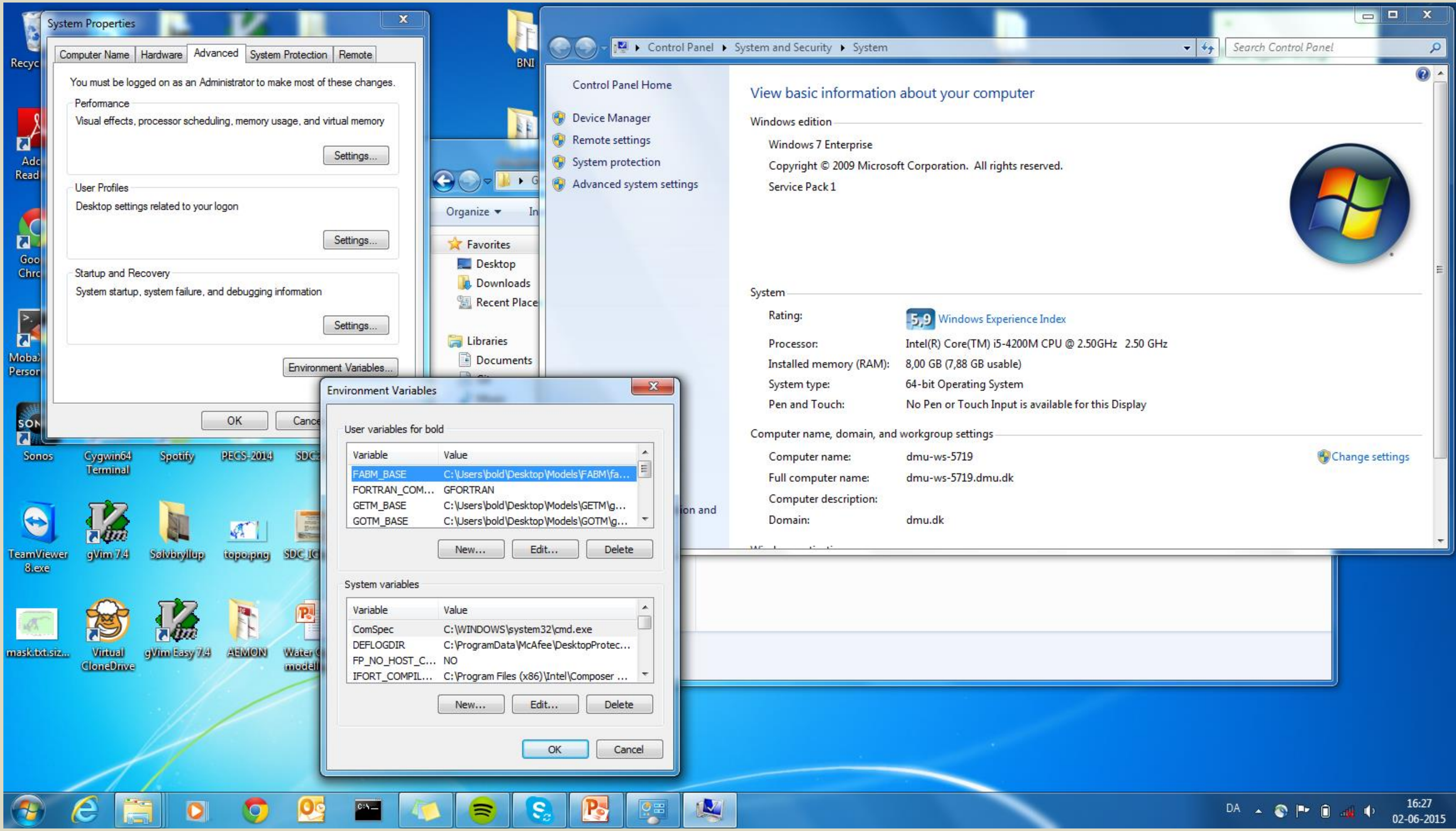
Tools and setup files

- Tools (different models – different tools ☺)
  - Python – scripting language – www.python.org
  - Python packages for data handling and visualization
    - numpy, matplotlib, NetCDF4 reader
  - Pyncview.py – flexible advanced plotting package build with python
- Volta Lake set-up
  - GOTM executable
  - Master configuration file – *volta_lake.xml*
  - Forcing files
  - Coupling files
  - FABM configuration for ERGOM

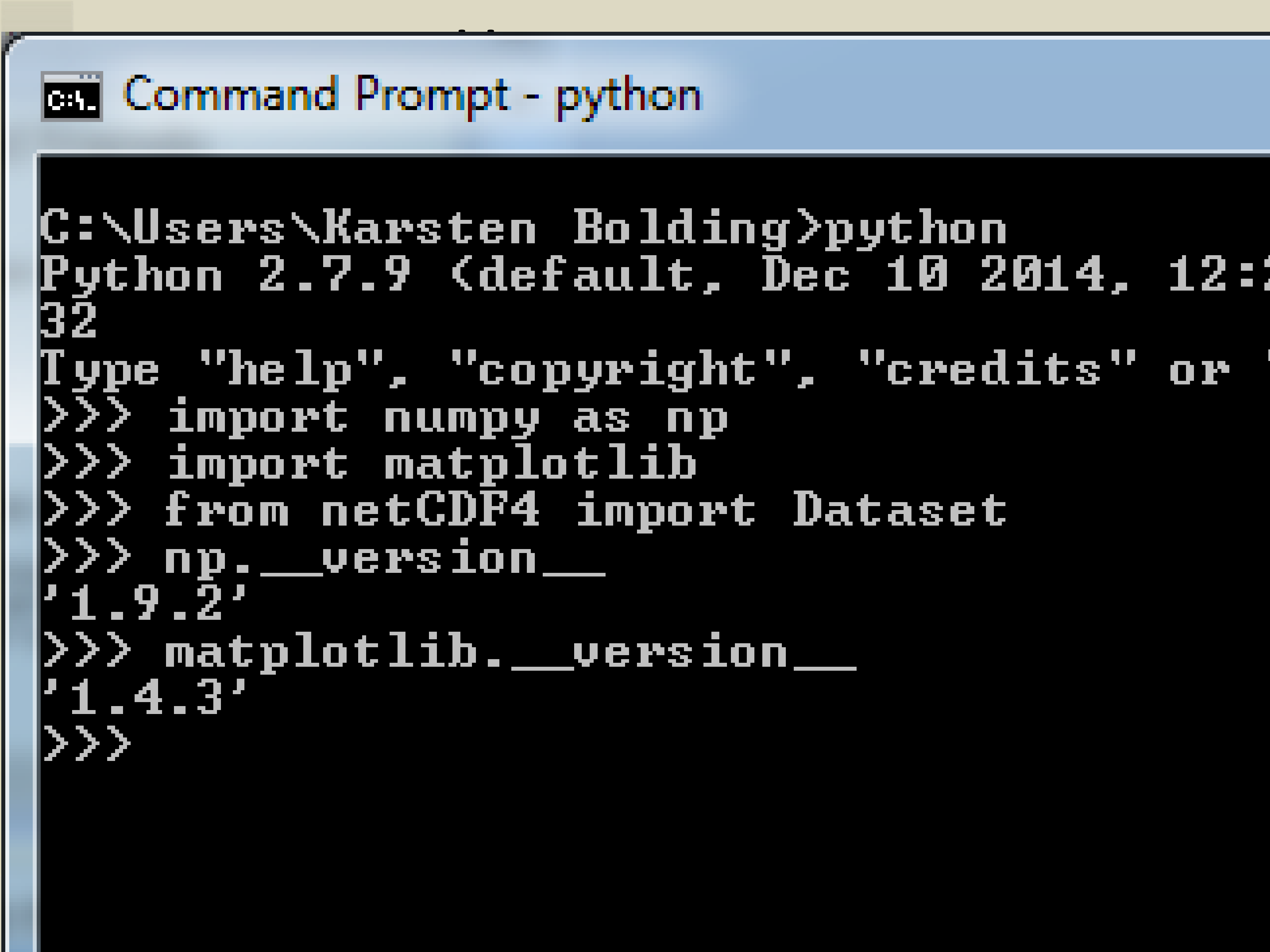
Python and friends



On Windows7



Department of Bioscience, Aarhus University, Denmark

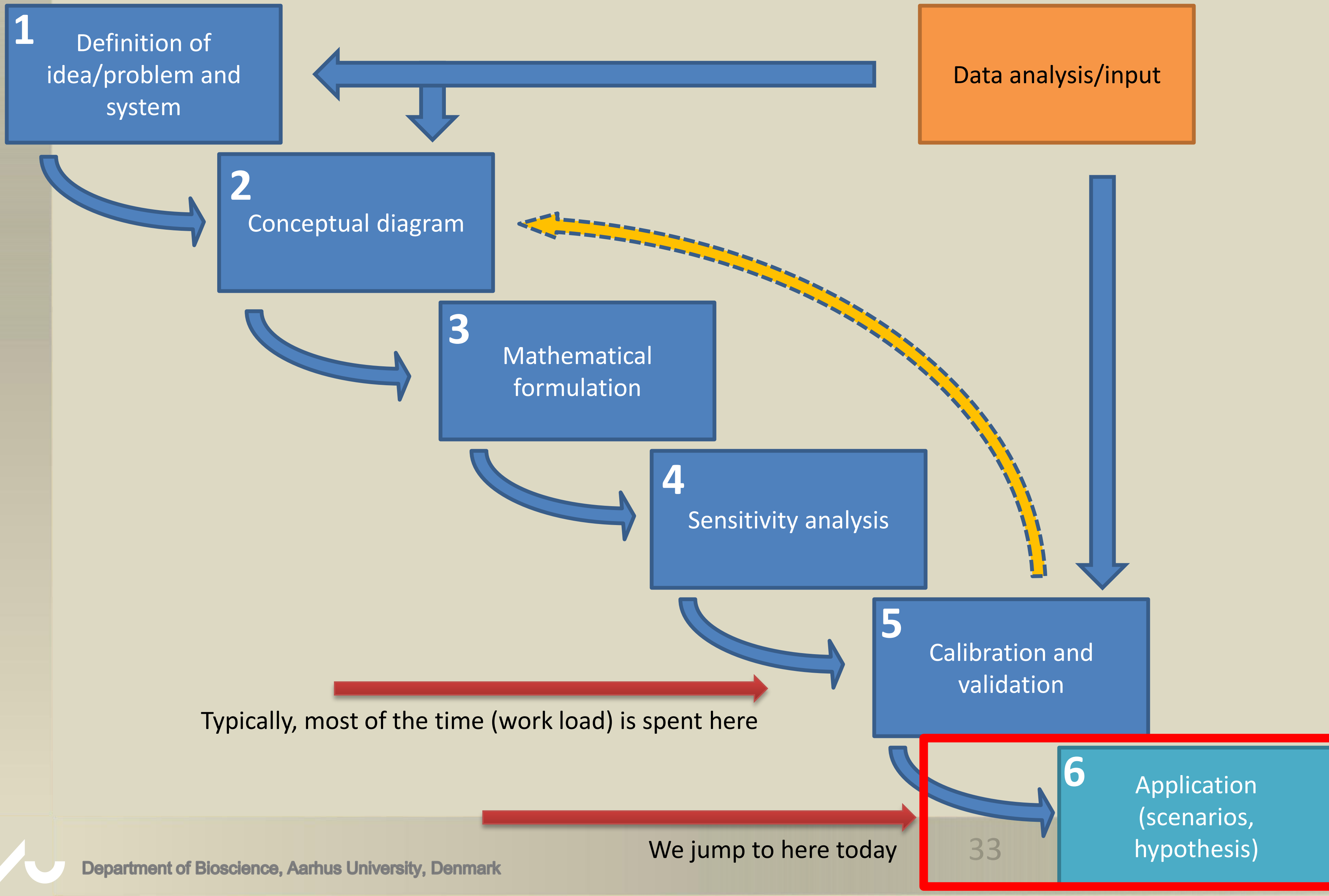

Command Prompt - python
C:\Users\Karsten Bolding>python
Python 2.7.9 (default, Dec 10 2014, 12:
32
Type "help", "copyright", "credits" or
>>> import numpy as np
>>> import matplotlib
>>> from netCDF4 import Dataset
>>> np.__version__
'1.9.2'
>>> matplotlib.__version__
'1.4.3'
>>>

GOTM configuration files

- Volta Lake set-up in Ghana\BIOS\Lakes\volta_lake\GOTM
  - bin\
    - gotm.exe (model) and editscenario.py (configuration tool)
  - volta_lake\
    - .bat files
    - volta_lake.xml
    - .dat files
    - gotm_fabm.nml
    - fabm.yaml

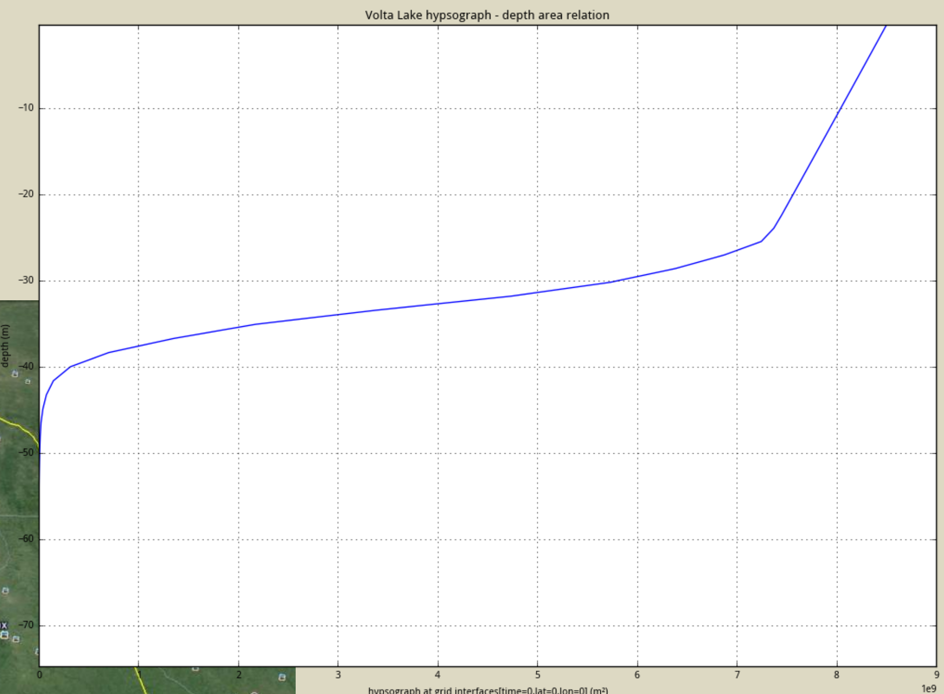
The modelling process

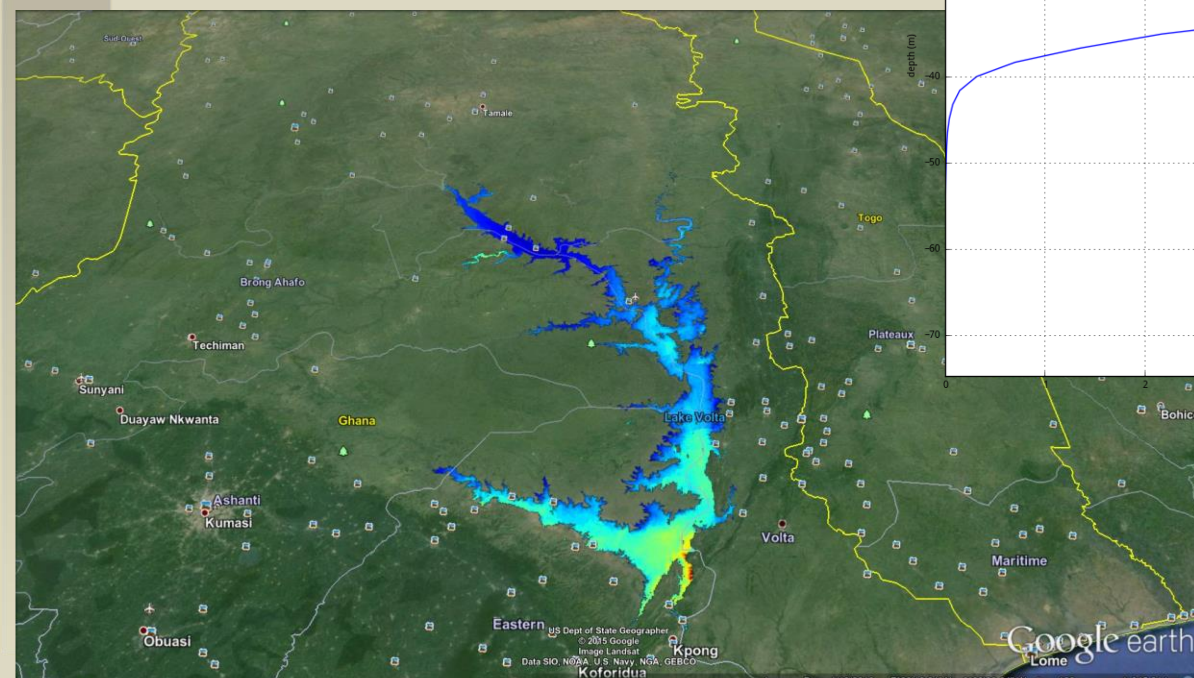
Climate change will degrade water quality – how much do we need to do?



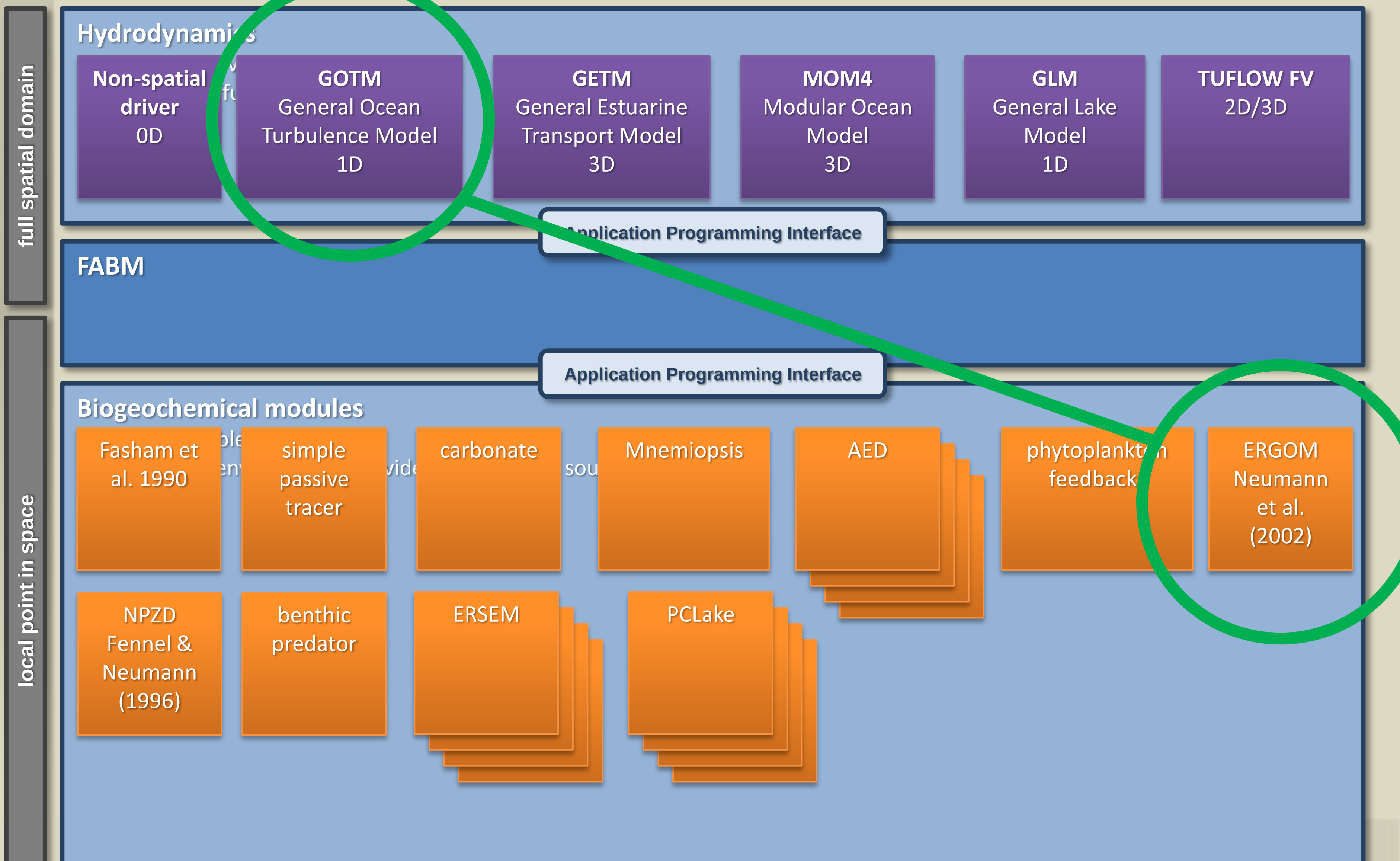
Volta Lake 1D model

- **<u>Lake Volt has a surface area of 8500 km^2</u>**
- Very irregular coast line – 1D (?)
- Meteorological forcing from NCEP (Deborah)
- Hypsograph show very small deep area
- We will run a 14 month simulation
- All necessary material on USB-stick





Workshop will be based on GOTM-FABM-ERGOM



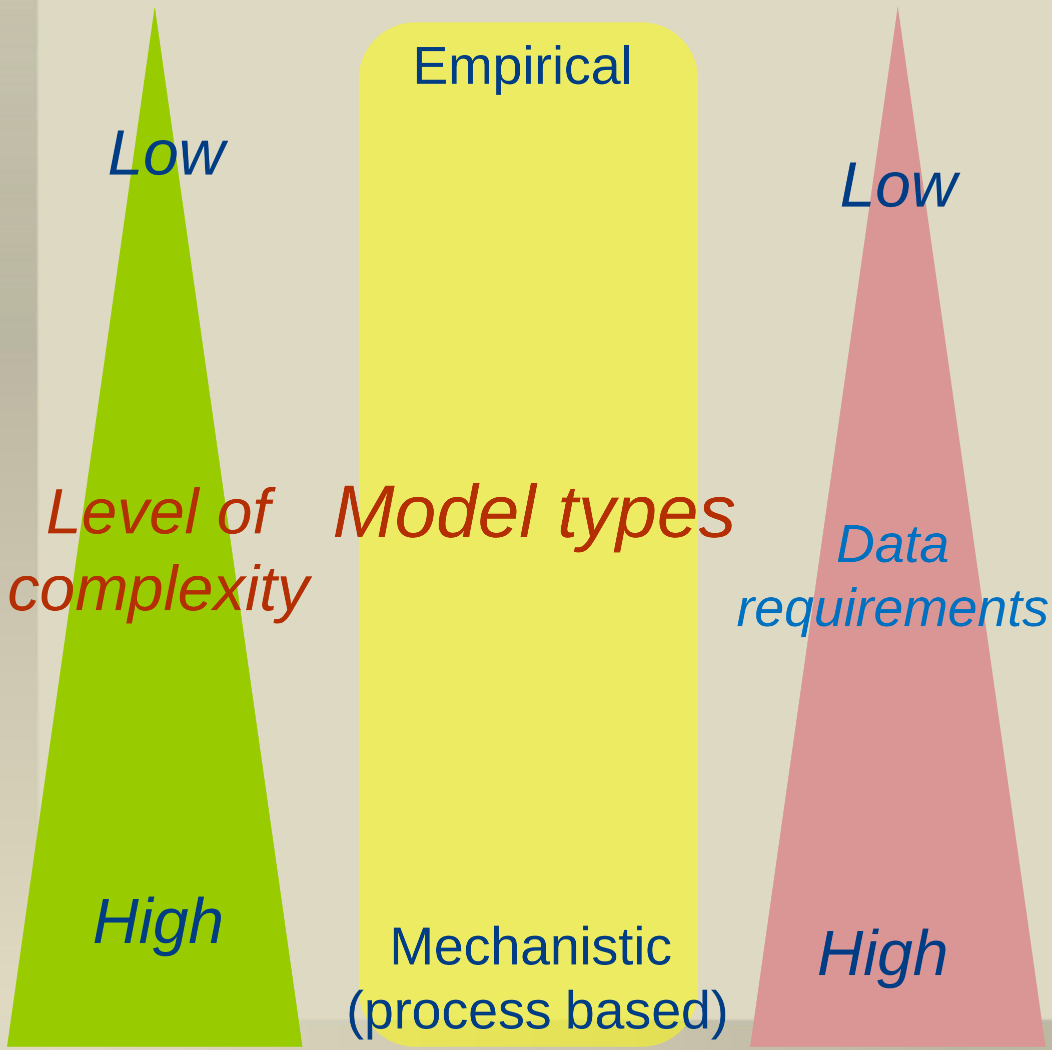
Introduction to GOTM-FABM-ERGOM

One-dimensional aquatic ecosystem model

LECTURE BY DENNIS TROLLE & KARSTEN BOLDING
DEPARTMENT OF BIOSCIENCE
FACULTY OF SCIENCE AND TECHNOLOGY, AARHUS UNIVERSITY, DENMARK

Model classifications

- Models differ widely in complexity (and data requirements) and may be classified in a number of ways



Runoff coefficients

Lumped models (e.g., NAM)

Semi-distributed models, based on curve number and linear reservoir (e.g., SWAT)

Fully distributed 3D models (e.g., MikeShe, MODFLOW)

Background to the models

ERGOM (Ecological ReGional Ocean Model) – biogeochemical model
ERGOM simulates the C, N, P, DO and Si cycles including inorganic nutrients, organic matter, multiple phytoplankton groups and a zooplankton group. ERGOM was originally developed at the Leibniz Institute for Baltic Sea Research, Warnemuende, Germany, by Thomas Neumann and Wolfgang Fennel.

GOTM (General Ocean Turbulence Model) – 1D hydrodynamic driver
Is a one-dimensional hydrodynamic model. It accounts for the effect of inflows/outflows, vertical mixing and surface heating and cooling, including the effect of ice cover.

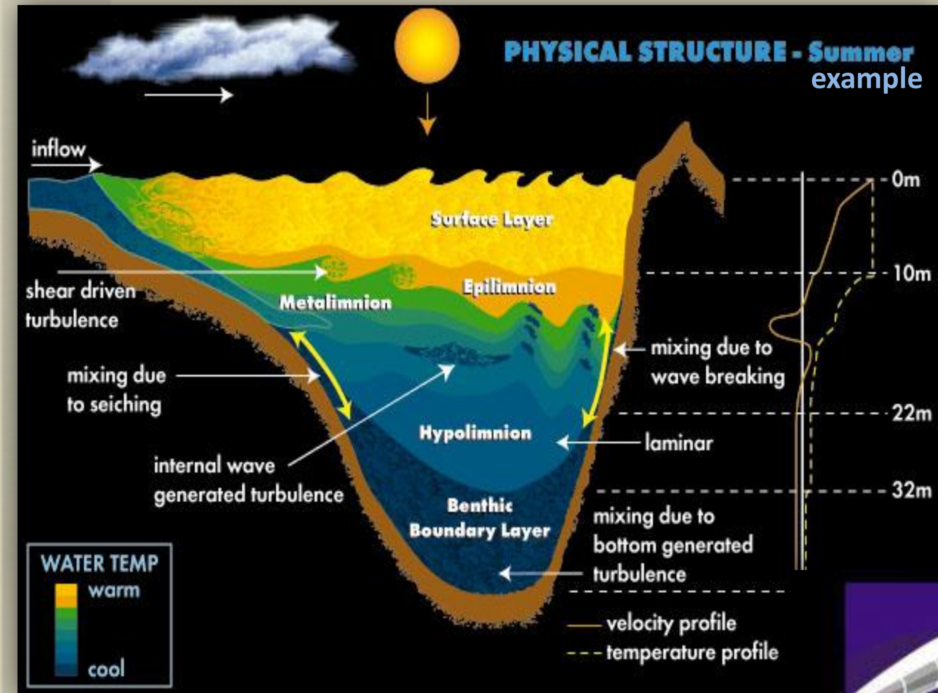
GETM (General Estuarine Transport Model) – 3D hydrodynamic driver
GETM is a 3D hydrodynamic model that operates on a structured, curvilinear grid.

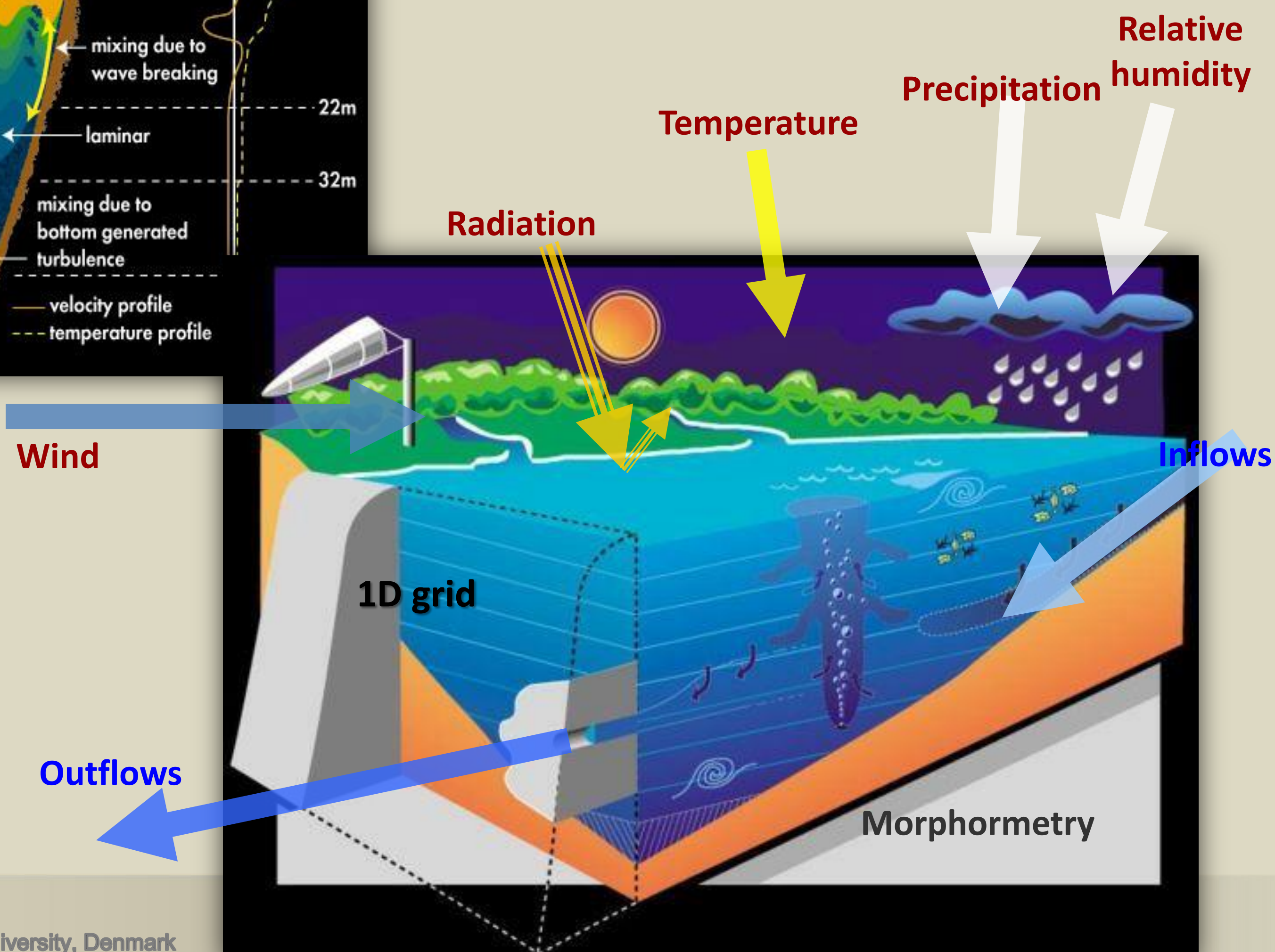
FABM (Framework for Aquatic Biogeochemical Models)
FABM is **NOT** a biogeochemical model itself, but FABM facilitates and eases the integration of several different biogeochemical models with several physical driver models (ranging 0D to 3D).

Application of GOTM-FABM-ERGOM

- 1D hydrodynamics model coupled with a complex ecological model (originally developed for marine environments but also able to run for freshwaters)
- Suitable for lakes where the forces acting to destabilize the water column (wind stress, surface cooling or plunging inflows) do not act over prolonged periods of time (calculate Lake Number to check for thermocline tilting)
- Good for long-term scenarios (years to decades)
- See also:
- http://fabm.net/
- http://gotm.net/
- http://ergom.net/

GOTM – conceptual model





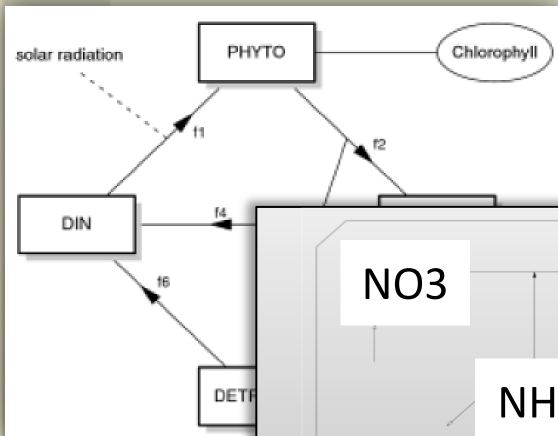
http://gotm.net/

Conceptual ecosystem models of different complexity

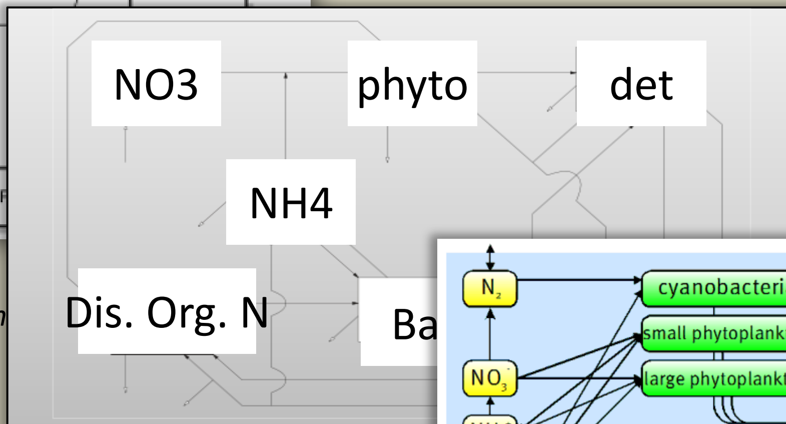
4 state variables

Increasing complexity

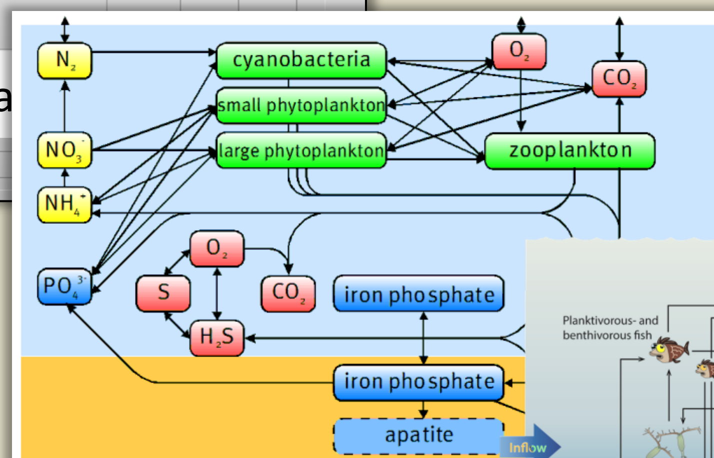
> 100 state variables



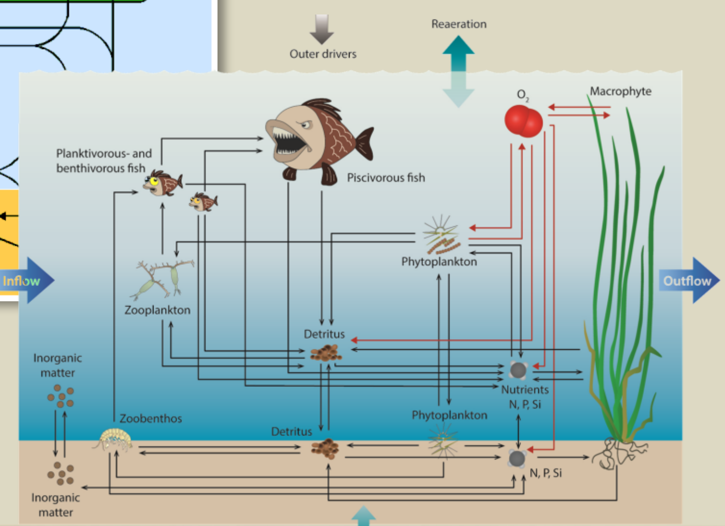
NPZD model
Fennel & Neuman



Fasham,
Ducklow, McKelvie (1990)

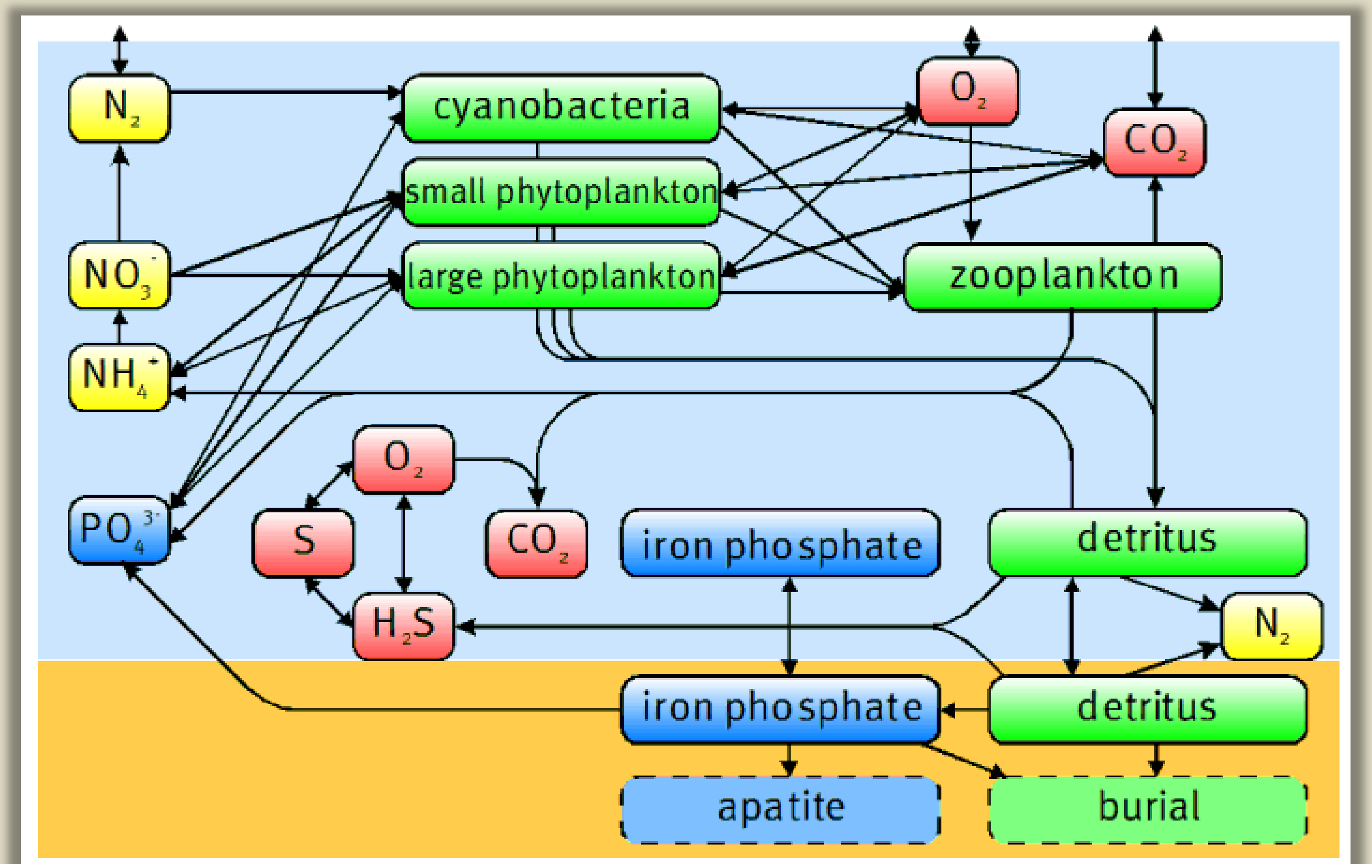


From http://www.ergom.net
Neumann et al. 2002; Neumann 2000



PCLake, originally by Jan Janse 1995,
Further developed by AU (Fenjuan Hu, Ph.D.)

ERGOM – conceptual model

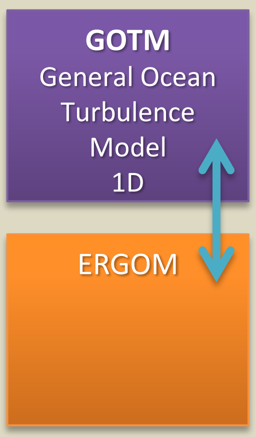


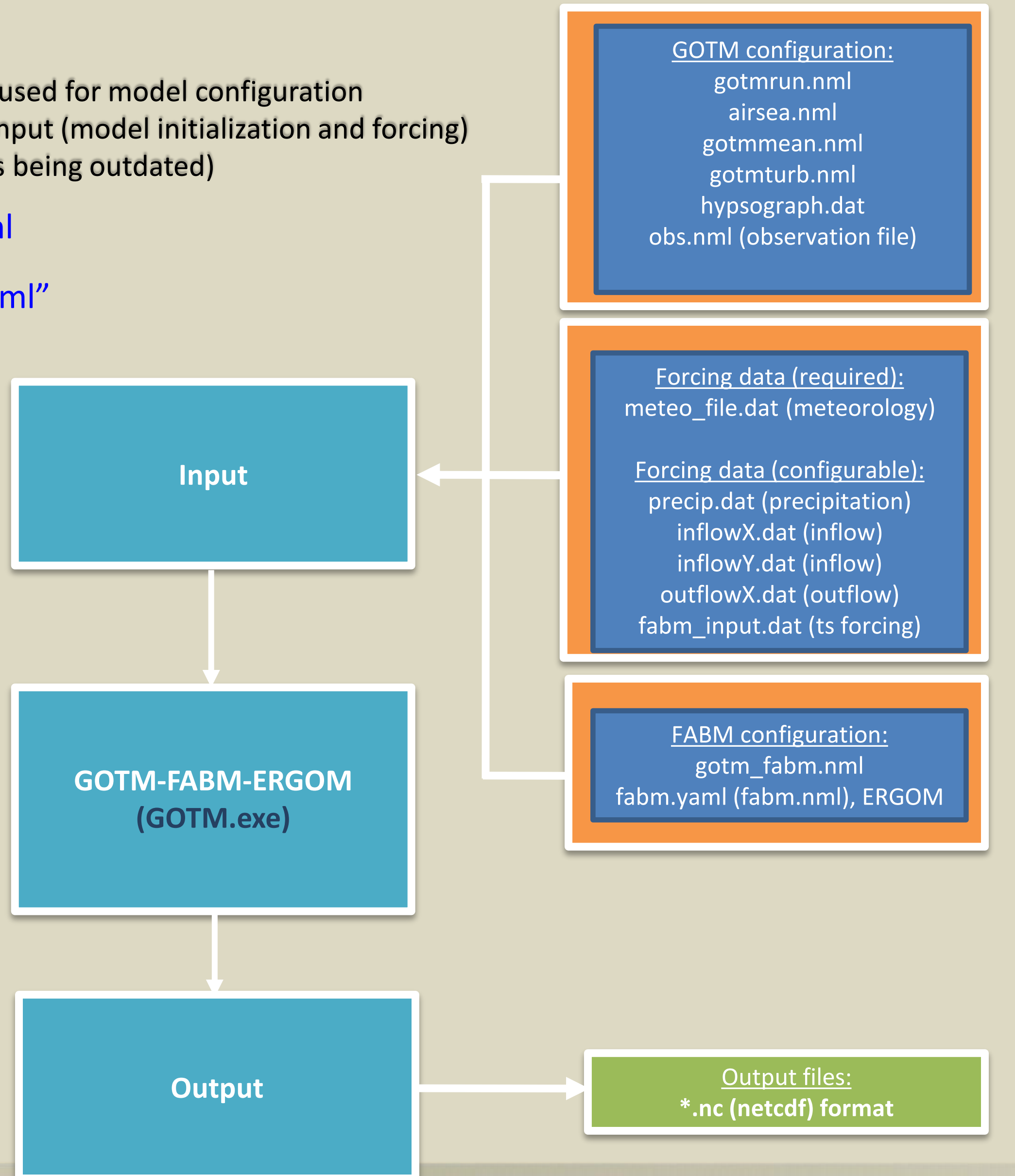
From http://www.ergom.net
Neumann et al. 2002; Neumann 2000

GOTM-FABM-ERGOM simulation engine

<u>Generally:</u>
Namelist (.nml) and YAML (.yaml) files are used for model configuration
Ascii (e.g., .dat or .txt) are used for model input (model initialization and forcing)
Output is provided in netcdf format (Ascii is being outdated)

Note: namelist files (*.nml) and yaml can be opened in any text editor.
The files include "namelist" and "yaml" formatting"

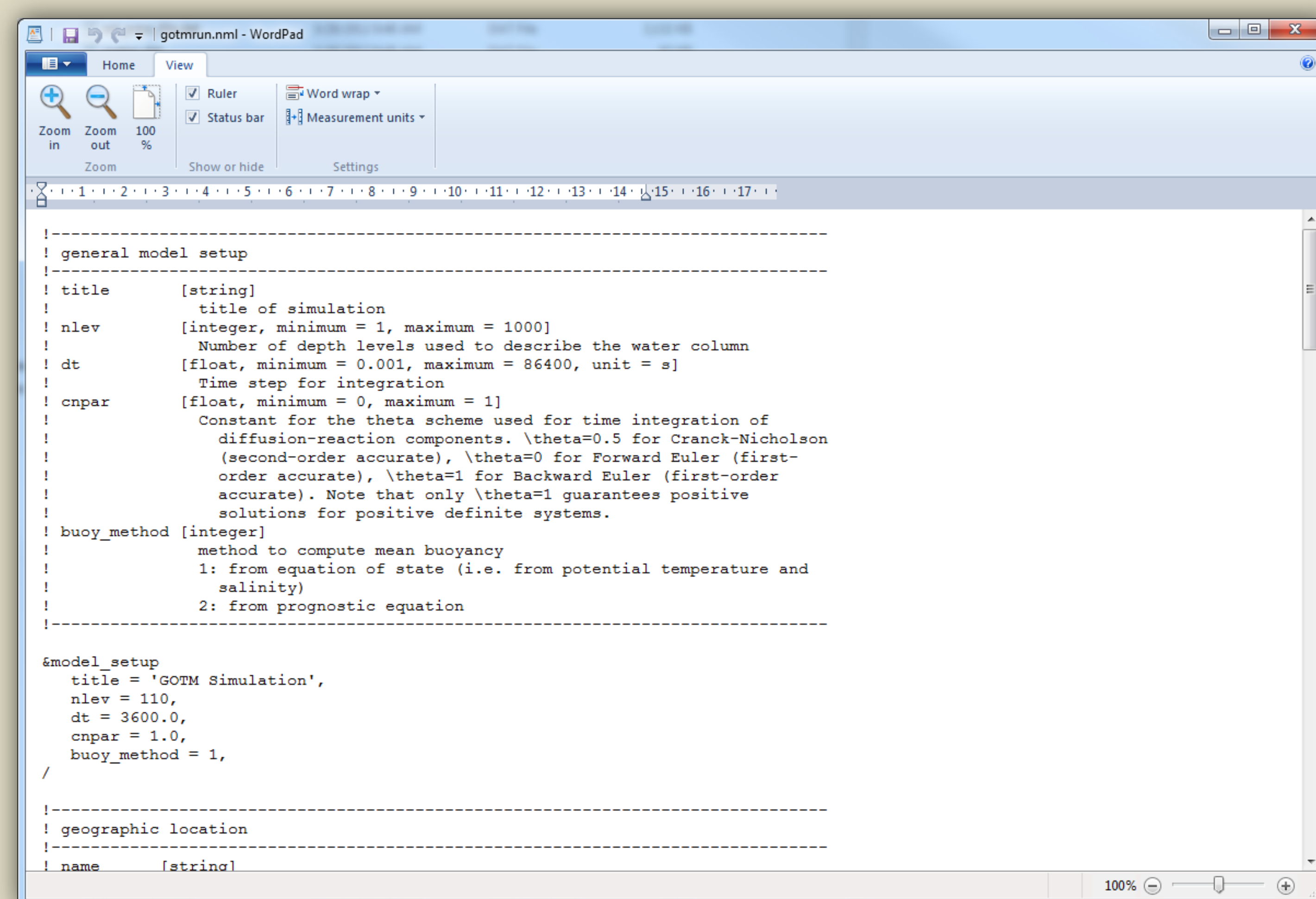




GOTM namelist configuration files

- gotmrun.nml
  - Basic runtime information
  - e.g., simulation period
- airsea.nml
  - Meteorological forcing definition
  - e.g., meteo filename
- gotmmean.nml
  - Mean flow specifications
  - e.g., grid definitions (typically not modified except for grid zoom options)
- gotmturb.nml
  - Turbulence set up
  - e.g., selection of turbulence closure model (typically not modified)
- hypsograph.dat
  - Hypsographic information (depth-area relations)
  - Mostly used for lake applications
- obs.nml (observation file)
  - Physical time series data (scalar and profile)
  - e.g., surface temperature and temperature profiles (can be used for relaxation and/or validation against model output)

gotmrun.nml

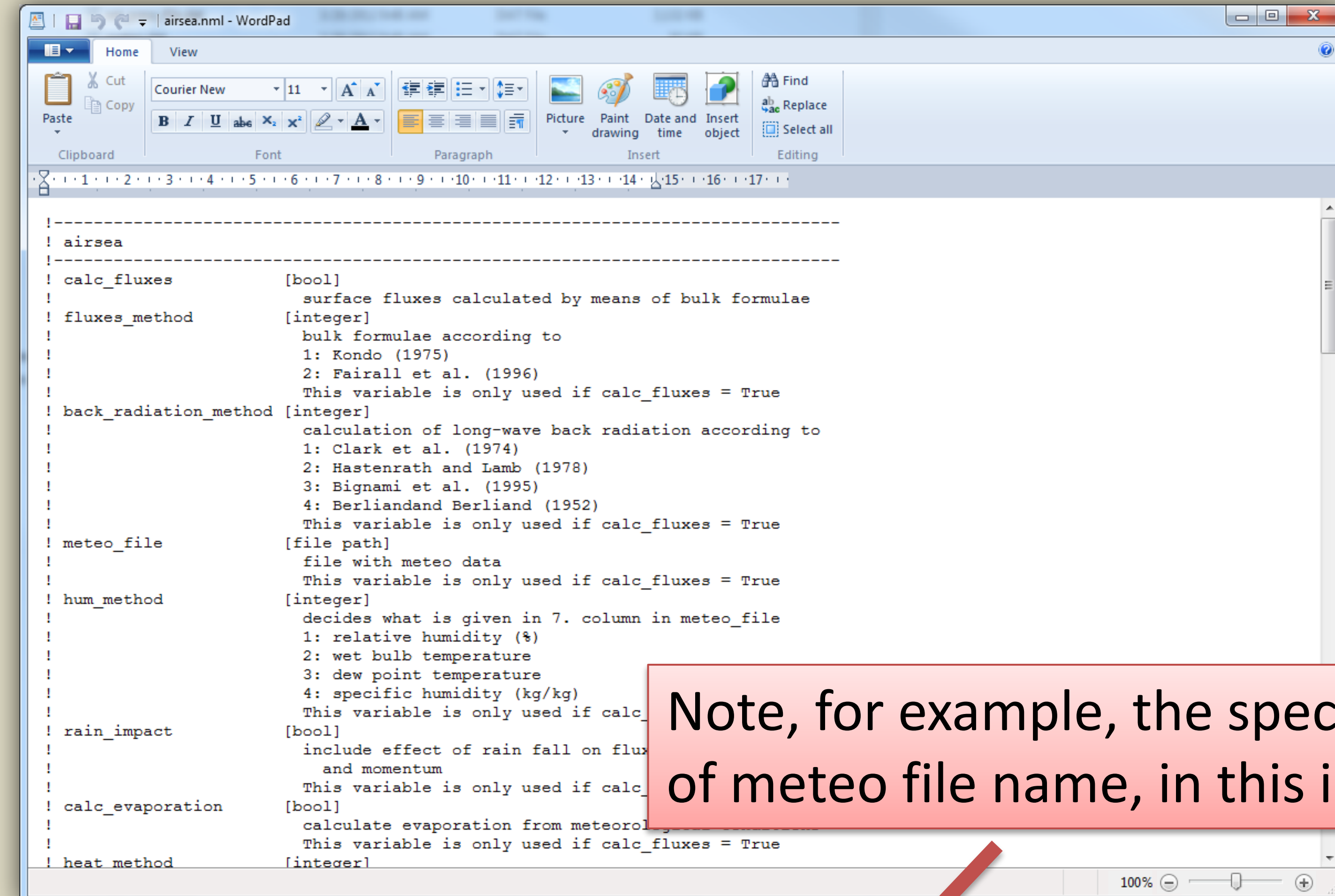



```
!-------------------------------------------------------------------------------
! general model setup
!-------------------------------------------------------------------------------
! title       [string]
!               title of simulation
! nlev        [integer, minimum = 1, maximum = 1000]
!               Number of depth levels used to describe the water column
! dt          [float, minimum = 0.001, maximum = 86400, unit = s]
!               Time step for integration
! cnpar       [float, minimum = 0, maximum = 1]
!               Constant for the theta scheme used for time integration of
!                 diffusion-reaction components. \theta=0.5 for Cranck-Nicholson
!                 (second-order accurate), \theta=0 for Forward Euler (first-
!                 order accurate), \theta=1 for Backward Euler (first-order
!                 accurate). Note that only \theta=1 guarantees positive
!                 solutions for positive definite systems.
! buoy_method [integer]
!               method to compute mean buoyancy
!               1: from equation of state (i.e. from potential temperature and
!                 salinity)
!               2: from prognostic equation
!-------------------------------------------------------------------------------

&model_setup
   title = 'GOTM Simulation',
   nlev = 110,
   dt = 3600.0,
   cnpar = 1.0,
   buoy_method = 1,
/

!-------------------------------------------------------------------------------
! geographic location
!-------------------------------------------------------------------------------
! name       [string]
```

<u>Configuration in relation to:</u>

general model setup (e.g., depth and number of vertical layers), geographic location, duration of run, format for output and filename(s), equation of state

airsea.nml



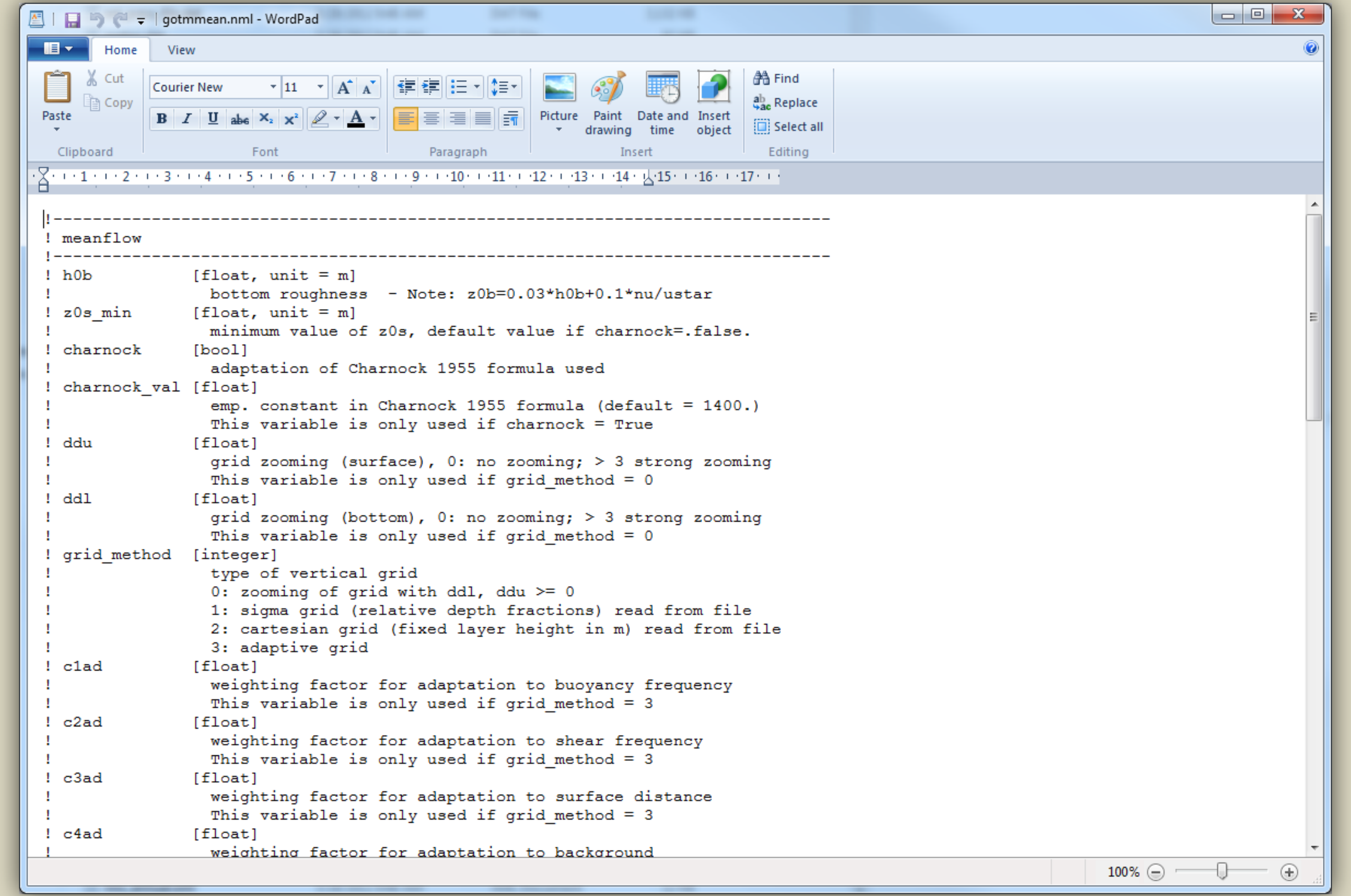

```
!-------------------------------------------------------------------------------
! airsea
!-------------------------------------------------------------------------------
! calc_fluxes           [bool]
!                         surface fluxes calculated by means of bulk formulae
! fluxes_method         [integer]
!                         bulk formulae according to
!                         1: Kondo (1975)
!                         2: Fairall et al. (1996)
!                         This variable is only used if calc_fluxes = True
! back_radiation_method [integer]
!                         calculation of long-wave back radiation according to
!                         1: Clark et al. (1974)
!                         2: Hastenrath and Lamb (1978)
!                         3: Bignami et al. (1995)
!                         4: Berliandand Berliand (1952)
!                         This variable is only used if calc_fluxes = True
! meteo_file            [file path]
!                         file with meteo data
!                         This variable is only used if calc_fluxes = True
! hum_method            [integer]
!                         decides what is given in 7. column in meteo_file
!                         1: relative humidity (%)
!                         2: wet bulb temperature
!                         3: dew point temperature
!                         4: specific humidity (kg/kg)
!                         This variable is only used if calc_
! rain_impact           [bool]
!                         include effect of rain fall on flux
!                           and momentum
!                         This variable is only used if calc_
! calc_evaporation      [bool]
!                         calculate evaporation from meteorol
!                         This variable is only used if calc_fluxes = True
! heat_method           [integer]
```

Note, for example, the specification of meteo file name, in this input file

Configuration in relation to:

air-sea interactions, like heat and momentum flux calculations,
scaling factors (e.g., for wind input), type of humidity input (four different options),
file specifications (e.g., **meteo_file name**)

gotmmean.nml



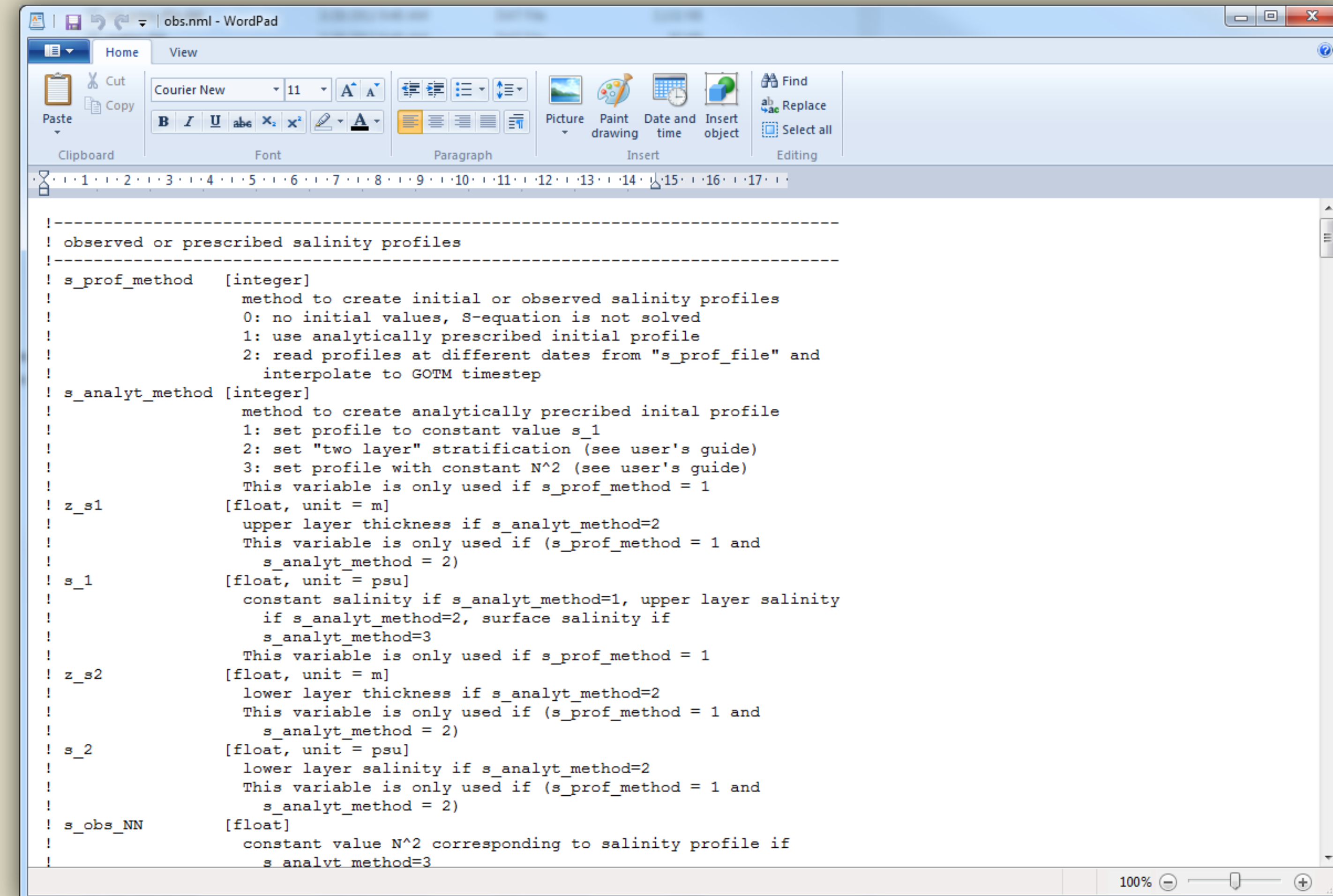
```
!-------------------------------------------------------------------------------
! meanflow
!-------------------------------------------------------------------------------
! h0b           [float, unit = m]
!                 bottom roughness  - Note: z0b=0.03*h0b+0.1*nu/ustar
! z0s_min       [float, unit = m]
!                 minimum value of z0s, default value if charnock=.false.
! charnock      [bool]
!                 adaptation of Charnock 1955 formula used
! charnock_val  [float]
!                 emp. constant in Charnock 1955 formula (default = 1400.)
!                 This variable is only used if charnock = True
! ddu           [float]
!                 grid zooming (surface), 0: no zooming; > 3 strong zooming
!                 This variable is only used if grid_method = 0
! ddl           [float]
!                 grid zooming (bottom), 0: no zooming; > 3 strong zooming
!                 This variable is only used if grid_method = 0
! grid_method   [integer]
!                 type of vertical grid
!                 0: zooming of grid with ddl, ddu >= 0
!                 1: sigma grid (relative depth fractions) read from file
!                 2: cartesian grid (fixed layer height in m) read from file
!                 3: adaptive grid
! c1ad          [float]
!                 weighting factor for adaptation to buoyancy frequency
!                 This variable is only used if grid_method = 3
! c2ad          [float]
!                 weighting factor for adaptation to shear frequency
!                 This variable is only used if grid_method = 3
! c3ad          [float]
!                 weighting factor for adaptation to surface distance
!                 This variable is only used if grid_method = 3
! c4ad          [float]
!                 weighting factor for adaptation to background
```

<u>Configuration in relation to:</u>

calculation grid specifications (and grid zoom options),

physical parameters (e.g., gravity, heat capacity), molecular viscosity and diffusion

gotmturb.nml



```
!-------------------------------------------------------------------------------
! observed or prescribed salinity profiles
!-------------------------------------------------------------------------------
! s_prof_method   [integer]
!                   method to create initial or observed salinity profiles
!                   0: no initial values, S-equation is not solved
!                   1: use analytically prescribed initial profile
!                   2: read profiles at different dates from "s_prof_file" and
!                     interpolate to GOTM timestep
! s_analyt_method [integer]
!                   method to create analytically precribed inital profile
!                   1: set profile to constant value s_1
!                   2: set "two layer" stratification (see user's guide)
!                   3: set profile with constant N^2 (see user's guide)
!                   This variable is only used if s_prof_method = 1
! z_s1            [float, unit = m]
!                   upper layer thickness if s_analyt_method=2
!                   This variable is only used if (s_prof_method = 1 and
!                     s_analyt_method = 2)
! s_1             [float, unit = psu]
!                   constant salinity if s_analyt_method=1, upper layer salinity
!                     if s_analyt_method=2, surface salinity if
!                     s_analyt_method=3
!                   This variable is only used if s_prof_method = 1
! z_s2            [float, unit = m]
!                   lower layer thickness if s_analyt_method=2
!                   This variable is only used if (s_prof_method = 1 and
!                     s_analyt_method = 2)
! s_2             [float, unit = psu]
!                   lower layer salinity if s_analyt_method=2
!                   This variable is only used if (s_prof_method = 1 and
!                     s_analyt_method = 2)
! s_obs_NN        [float]
!                   constant value N^2 corresponding to salinity profile if
!                     s_analyt_method=3
```

Configuration in relation to:

selection of turbulence closure model, and parameterization of turbulence models

hypsograph.dat

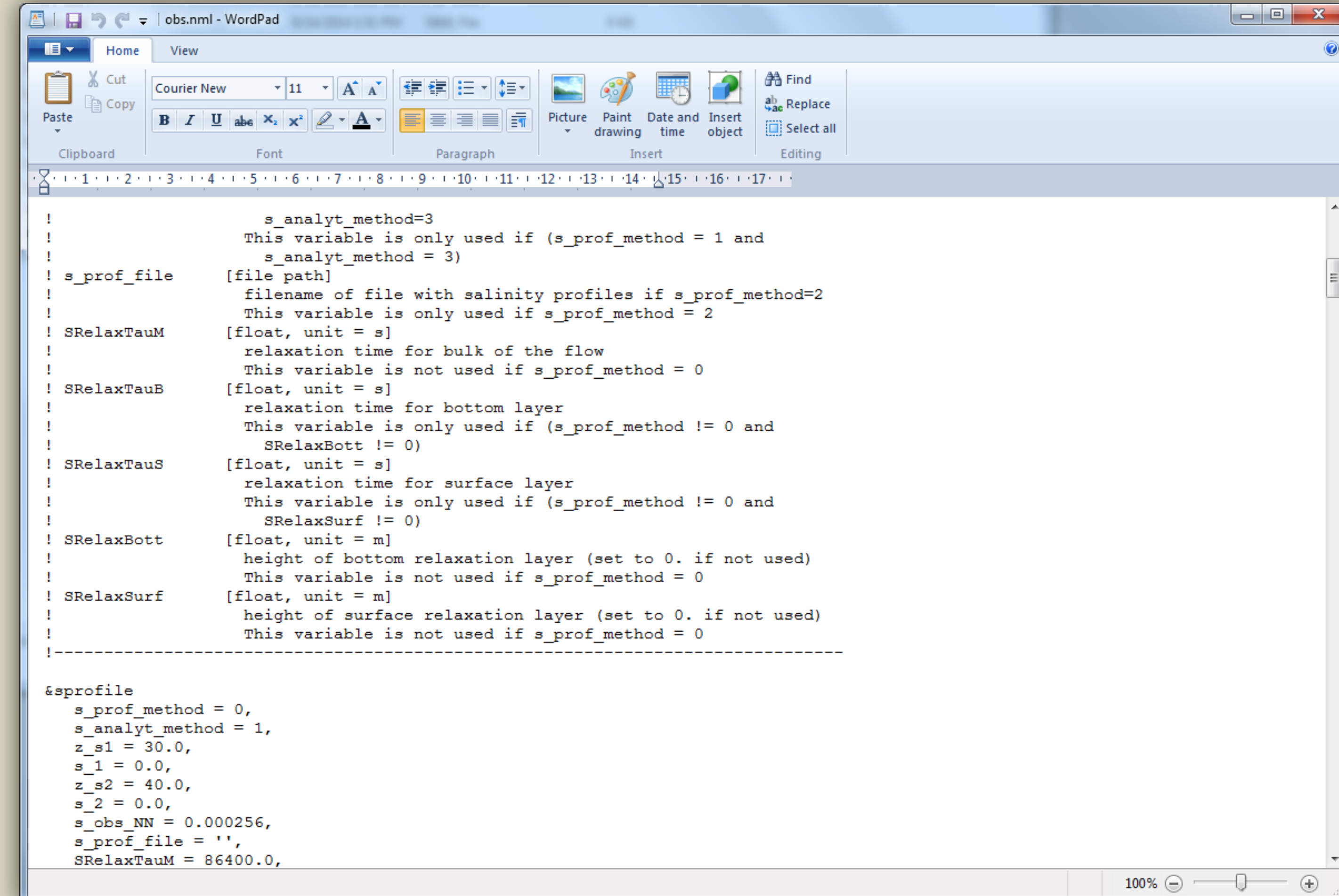
hypsograph.dat - WordPad

```
! first line specifies 36 data points,
! and reading format according to case(3)
! column one is height/depth (m) and column two is area (m2)
!   case(1)  ! surface ref, read from bottom
!   case(2)  ! surface ref, read from surface
!   case(3)  ! bottom ref, read from bottom
!   case(4)  ! bottom ref, read from surface
36       3
0.0   0
0.7   1541
1.5   16921
2.5   45021
3.5   88521
4.5   136421.002
5.5   180321.004
6.5   241721.006
7.5   318621.008
8.5   469321.008
9.5   555021.005
10.5  646421.007
12.5  684421.007
13.5  719321.009
14.5  752721.011
15.5  787521.01
16.5  823121.008
17.5  858721.006
18.5  898921.007
19.5  942821.009
20.5  989321.009
21.5  1036421.007
22.5  1077721.006
23.5  1144921.003
24.5  1206421.003
25.5  1267021.001
26.5  1321221.002
```

<u>Configuration in relation to:</u>

depth-area relations, number of data points and format specification (four options)

obs.nml



```
!                    s_analyt_method=3
!                  This variable is only used if (s_prof_method = 1 and
!                    s_analyt_method = 3)
! s_prof_file     [file path]
!                  filename of file with salinity profiles if s_prof_method=2
!                  This variable is only used if s_prof_method = 2
! SRelaxTauM      [float, unit = s]
!                  relaxation time for bulk of the flow
!                  This variable is not used if s_prof_method = 0
! SRelaxTauB      [float, unit = s]
!                  relaxation time for bottom layer
!                  This variable is only used if (s_prof_method != 0 and
!                    SRelaxBott != 0)
! SRelaxTauS      [float, unit = s]
!                  relaxation time for surface layer
!                  This variable is only used if (s_prof_method != 0 and
!                    SRelaxSurf != 0)
! SRelaxBott      [float, unit = m]
!                  height of bottom relaxation layer (set to 0. if not used)
!                  This variable is not used if s_prof_method = 0
! SRelaxSurf      [float, unit = m]
!                  height of surface relaxation layer (set to 0. if not used)
!                  This variable is not used if s_prof_method = 0
!-------------------------------------------------------------------------------

&sprofile
   s_prof_method = 0,
   s_analyt_method = 1,
   z_s1 = 30.0,
   s_1 = 0.0,
   z_s2 = 40.0,
   s_2 = 0.0,
   s_obs_NN = 0.000256,
   s_prof_file = '',
   SRelaxTauM = 86400.0,
```

<u>Configuration in relation to:</u>
observed/prescribed salinity profiles, observed/prescribed potential temperature profiles, external pressure gradients, internal pressure gradients, **<u>light extinction method</u>**, vertical advection options, sea surface elevations, wind induced waves options, observed or prescribed velocity profiles, turbulent dissipation rate profiles

Department of Bioscience, Aarhus University, Denmark

FABM namelist configuration files

- gotm_fabml.nml
  - Basic runtime information where FABM can be turned on/off
- fabm.yaml
  - Selection of biogeochemical models/modules to run (e.g., ERGOM, ERSEM, AED, PCLake)
  - Initialization of FABM model(s) (e.g., ERGOM) and model parameterization
  - Additional to fabm.yaml, some biogeochemical models may also include extra .nml files for model parameterization (e.g., phytoplankton.nml, zooplankton.nml)

gotm_fabm.nml

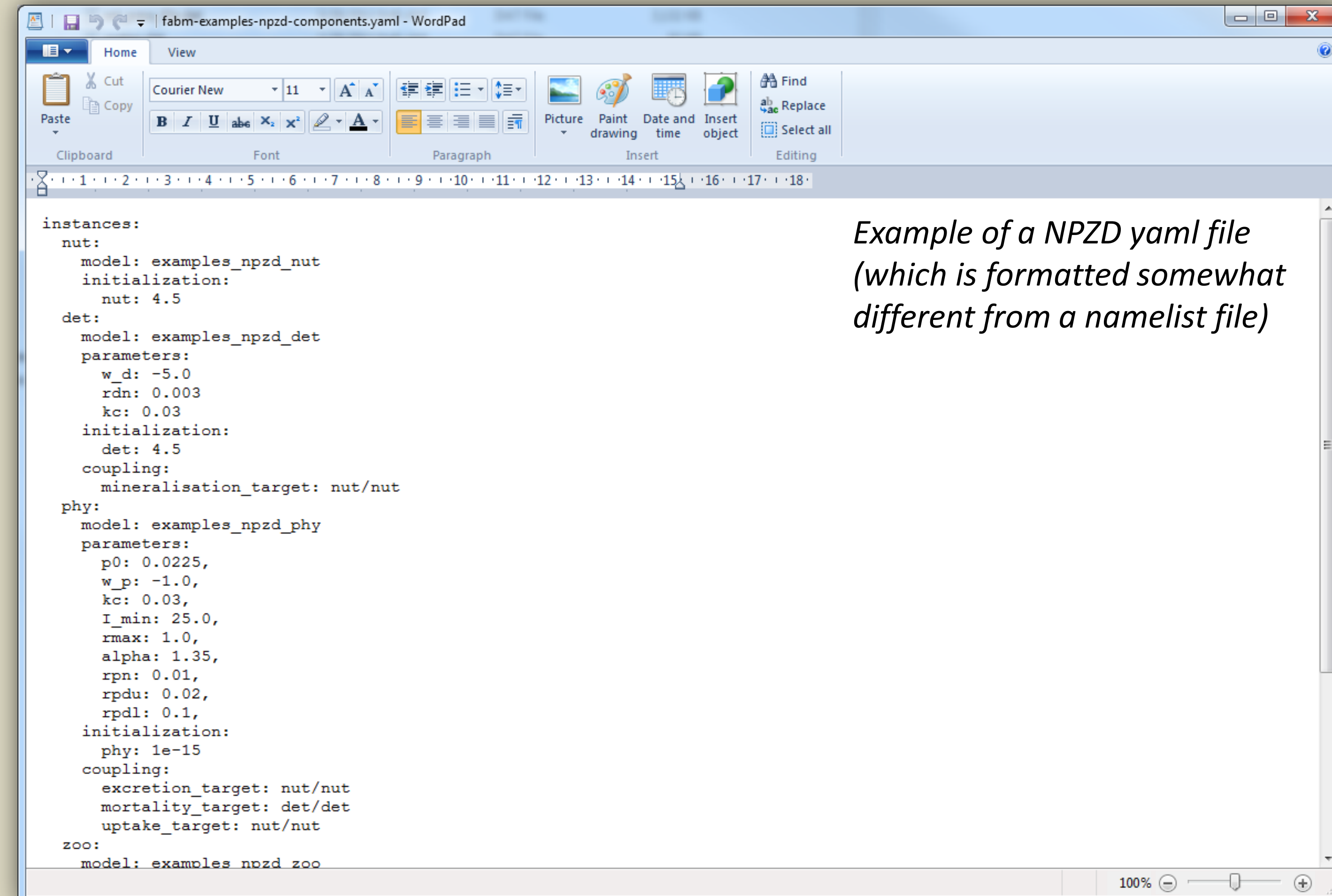
```
!-------------------------------------------------------------------------------
! gotm_fabm_nml: namelist that controls the GOTM-FABM coupler
!-------------------------------------------------------------------------------
! fabm_calc          [bool]
!                     use the Framework for Aquatic Biogeochemical Models
! cnpar              [float]
!                     Cranck-Nicholson parameter for vertical diffusion
!                     This variable is used only if fabm_calc = True
! w_adv_discr        [integer]
!                     advection scheme for vertical motion
!                     1: first-order upstream
!                     2: not coded yet
!                     3: third-order polynomial
!                     4: TVD with Superbee limiter
!                     5: TVD with MUSCL limiter
!                     6: TVD with ULTIMATE QUICKEST
!                     This variable is used only if fabm_calc = True
! ode_method         [integer]
!                     ODE scheme for source and sink dynamics
!                     1: first-order explicit (not positive)
!                     2: second-order explicit Runge-Kutta (not positive)
!                     3: fourth-order explicit Runge-Kutta (not positive)
!                     4: Patankar (first-order, not conservative)
!                     5: Patankar-RK (second-order, not conservative)
!                     6: Patankar-RK (does not work, not conservative)
!                     7: Modified Patankar (1st-order, conservat., posit.)
!                     8: Modified Patankar-RK (2nd-order, conservat., posit.)
!                     9: Modified Patankar-RK (does not work, conservat.,
!                       posit.)
!                     10: Extended Modified Patankar (1st-order, conservat.,
!                       posit.)
!                     11: Extended Modified Patankar-RK (2nd-order, conservat.,
!                       posit.)
!                     This variable is used only if fabm_calc = True
! split_factor       [integer, minimum = 1]
```

Configuration in relation to:

whether fabm should be on/off, ODE method, bioshade feedback on/off, repair_state on/off

Department of Bioscience, Aarhus University, Denmark

fabm.yaml



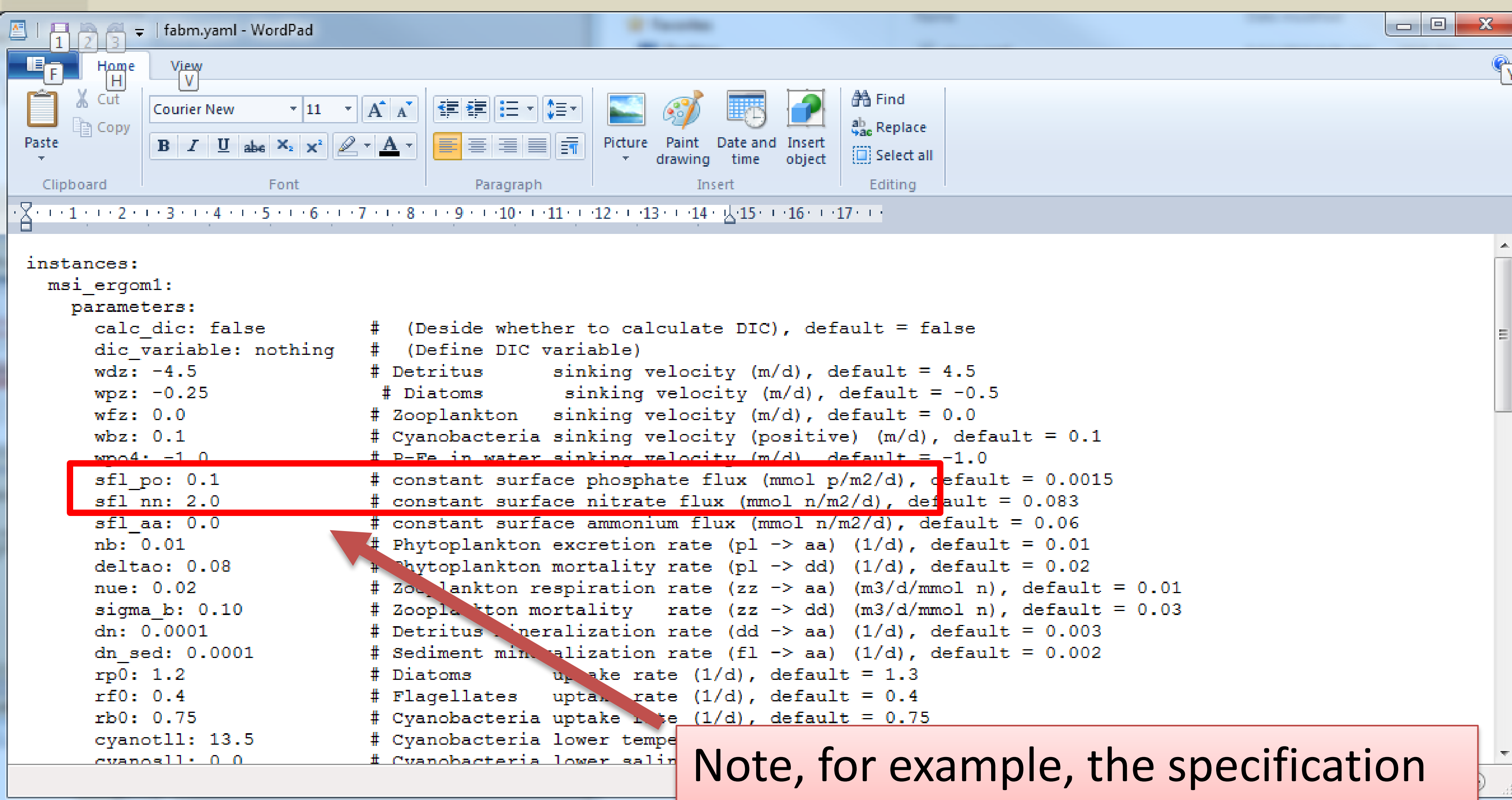
```yaml
instances:
  nut:
    model: examples_npzd_nut
    initialization:
      nut: 4.5
  det:
    model: examples_npzd_det
    parameters:
      w_d: -5.0
      rdn: 0.003
      kc: 0.03
    initialization:
      det: 4.5
    coupling:
      mineralisation_target: nut/nut
  phy:
    model: examples_npzd_phy
    parameters:
      p0: 0.0225,
      w_p: -1.0,
      kc: 0.03,
      I_min: 25.0,
      rmax: 1.0,
      alpha: 1.35,
      rpn: 0.01,
      rpdu: 0.02,
      rpdl: 0.1,
    initialization:
      phy: 1e-15
    coupling:
      excretion_target: nut/nut
      mortality_target: det/det
      uptake_target: nut/nut
  zoo:
    model: examples_npzd_zoo
```


Example of a NPZD yaml file (which is formatted somewhat different from a namelist file)

<u>Configuration in relation to:</u>

List of coupled biogeochemical models to run plus configuration of each of these, (e.g., ERGOM initial conditions, ERGOM model parameterization)

ERGOM (fabm.yaml)



```yaml
instances:
  msi_ergom1:
    parameters:
      calc_dic: false          #  (Deside whether to calculate DIC), default = false
      dic_variable: nothing    #  (Define DIC variable)
      wdz: -4.5                # Detritus      sinking velocity (m/d), default = 4.5
      wpz: -0.25                # Diatoms       sinking velocity (m/d), default = -0.5
      wfz: 0.0                 # Zooplankton   sinking velocity (m/d), default = 0.0
      wbz: 0.1                 # Cyanobacteria sinking velocity (positive) (m/d), default = 0.1
      wpo4: -1.0               # P-Fe in water sinking velocity (m/d), default = -1.0
      sfl_po: 0.1              # constant surface phosphate flux (mmol p/m2/d), default = 0.0015
      sfl_nn: 2.0              # constant surface nitrate flux (mmol n/m2/d), default = 0.083
      sfl_aa: 0.0              # constant surface ammonium flux (mmol n/m2/d), default = 0.06
      nb: 0.01                 # Phytoplankton excretion rate (pl -> aa) (1/d), default = 0.01
      deltao: 0.08             # Phytoplankton mortality rate (pl -> dd) (1/d), default = 0.02
      nue: 0.02                # Zooplankton respiration rate (zz -> aa) (m3/d/mmol n), default = 0.01
      sigma_b: 0.10            # Zooplankton mortality   rate (zz -> dd) (m3/d/mmol n), default = 0.03
      dn: 0.0001               # Detritus mineralization rate (dd -> aa) (1/d), default = 0.003
      dn_sed: 0.0001           # Sediment mineralization rate (fl -> aa) (1/d), default = 0.002
      rp0: 1.2                 # Diatoms       uptake rate (1/d), default = 1.3
      rf0: 0.4                 # Flagellates   uptake rate (1/d), default = 0.4
      rb0: 0.75                # Cyanobacteria uptake rate (1/d), default = 0.75
      cyanotll: 13.5           # Cyanobacteria lower tempe
      cyanosll: 0.0            # Cyanobacteria lower salin
```

Note, for example, the specification of external nutrient loads here

Initialization and forcing data files

Two basic types:
scalar data and profile data with fixed formats (all in ASCII files). Time information is specified as YYYY-MM-DD HH:MM:SS and is always in the first two columns, remaining columns are data in free format. Interpolation to model time step is done by GOTM. Time can be non-equidistant and is assumed to be in GMT.

Scalar data:

- meteo_file.dat
- precip.dat
- Inflows/Outflows
  - Black Volta, White Volta, Oti
  - Volta

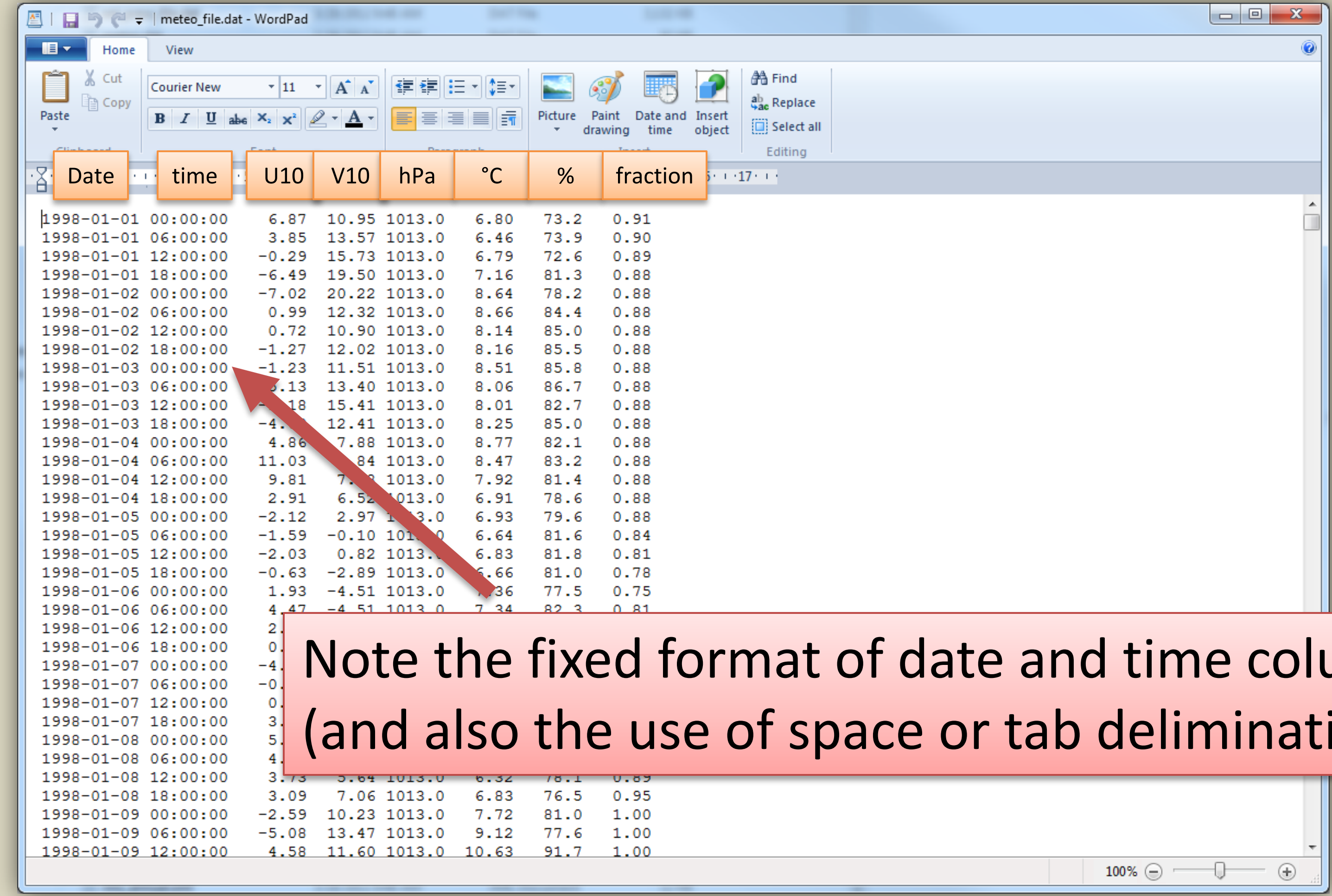
Water balance from: *Volta Basin Water Balance, M. Andreini et. Al. 2000.*

Profile data:

- e.g., t_prof_file.dat (temperature profiles)

Department of Bioscience, Aarhus University, Denmark

meteo_file.dat



Date	time	U10	V10	hPa	°C	%	fraction
1998-01-01	00:00:00	6.87	10.95	1013.0	6.80	73.2	0.91
1998-01-01	06:00:00	3.85	13.57	1013.0	6.46	73.9	0.90
1998-01-01	12:00:00	-0.29	15.73	1013.0	6.79	72.6	0.89
1998-01-01	18:00:00	-6.49	19.50	1013.0	7.16	81.3	0.88
1998-01-02	00:00:00	-7.02	20.22	1013.0	8.64	78.2	0.88
1998-01-02	06:00:00	0.99	12.32	1013.0	8.66	84.4	0.88
1998-01-02	12:00:00	0.72	10.90	1013.0	8.14	85.0	0.88
1998-01-02	18:00:00	-1.27	12.02	1013.0	8.16	85.5	0.88
1998-01-03	00:00:00	-1.23	11.51	1013.0	8.51	85.8	0.88
1998-01-03	06:00:00	[illegible]	13.40	1013.0	8.06	86.7	0.88
1998-01-03	12:00:00	[illegible]	15.41	1013.0	8.01	82.7	0.88
1998-01-03	18:00:00	[illegible]	12.41	1013.0	8.25	85.0	0.88
1998-01-04	00:00:00	4.86	7.88	1013.0	8.77	82.1	0.88
1998-01-04	06:00:00	11.03	[illegible]	1013.0	8.47	83.2	0.88
1998-01-04	12:00:00	9.81	[illegible]	1013.0	7.92	81.4	0.88
1998-01-04	18:00:00	2.91	6.52	[illegible]	6.91	78.6	0.88
1998-01-05	00:00:00	-2.12	2.97	[illegible]	6.93	79.6	0.88
1998-01-05	06:00:00	-1.59	-0.10	[illegible]	6.64	81.6	0.84
1998-01-05	12:00:00	-2.03	0.82	[illegible]	6.83	81.8	0.81
1998-01-05	18:00:00	-0.63	-2.89	1013.0	[illegible]	81.0	0.78
1998-01-06	00:00:00	1.93	-4.51	1013.0	[illegible]	77.5	0.75
1998-01-06	06:00:00	4.47	-4.51	1013.0	7.34	82.3	0.81
1998-01-06	12:00:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1998-01-06	18:00:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1998-01-07	00:00:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1998-01-07	06:00:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1998-01-07	12:00:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1998-01-07	18:00:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1998-01-08	00:00:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1998-01-08	06:00:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1998-01-08	12:00:00	3.73	5.64	1013.0	6.32	78.1	0.89
1998-01-08	18:00:00	3.09	7.06	1013.0	6.83	76.5	0.95
1998-01-09	00:00:00	-2.59	10.23	1013.0	7.72	81.0	1.00
1998-01-09	06:00:00	-5.08	13.47	1013.0	9.12	77.6	1.00
1998-01-09	12:00:00	4.58	11.60	1013.0	10.63	91.7	1.00

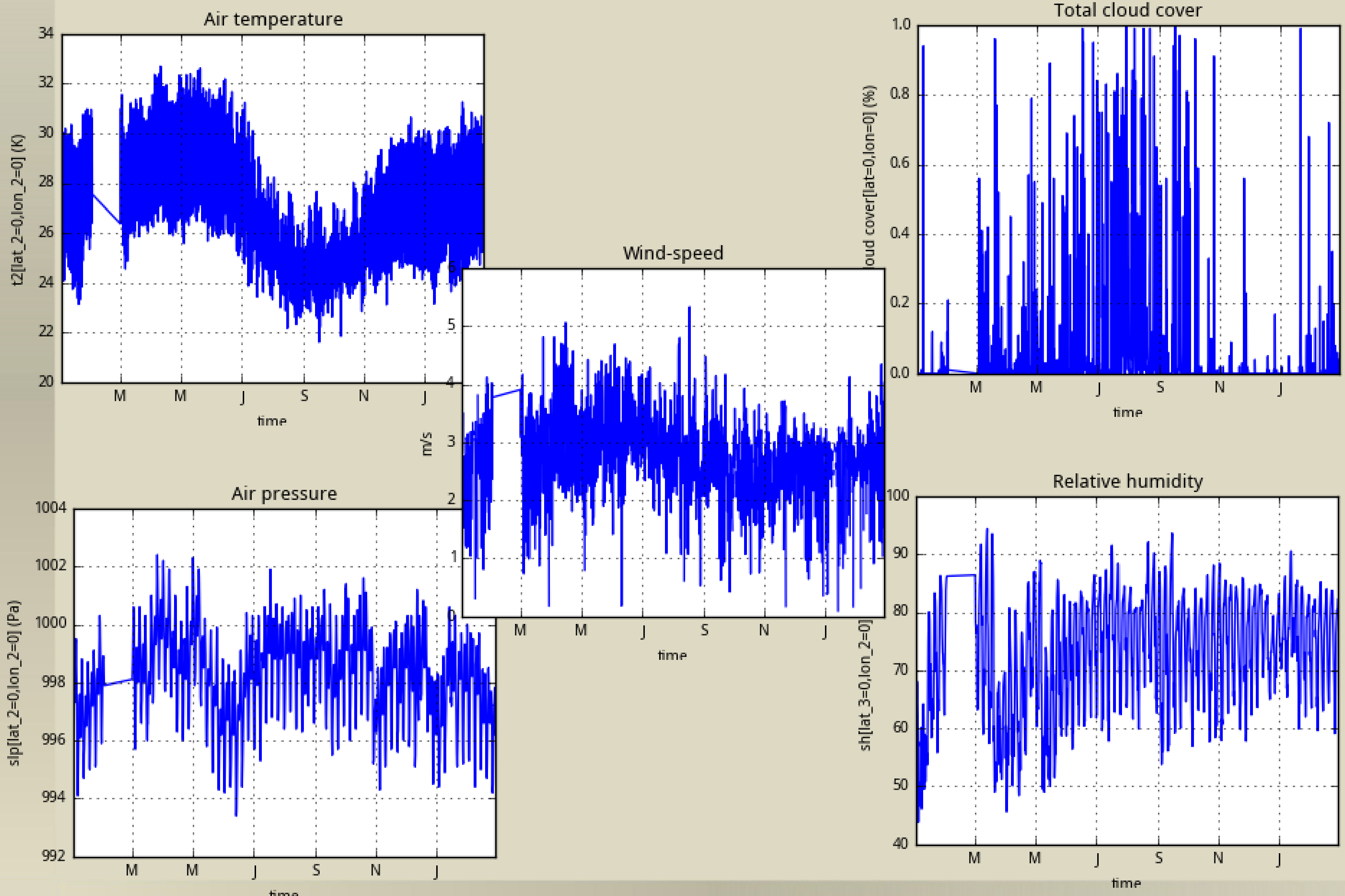
Note the fixed format of date and time columns (and also the use of space or tab delimination)

<u>Forcing data from left to right:</u>

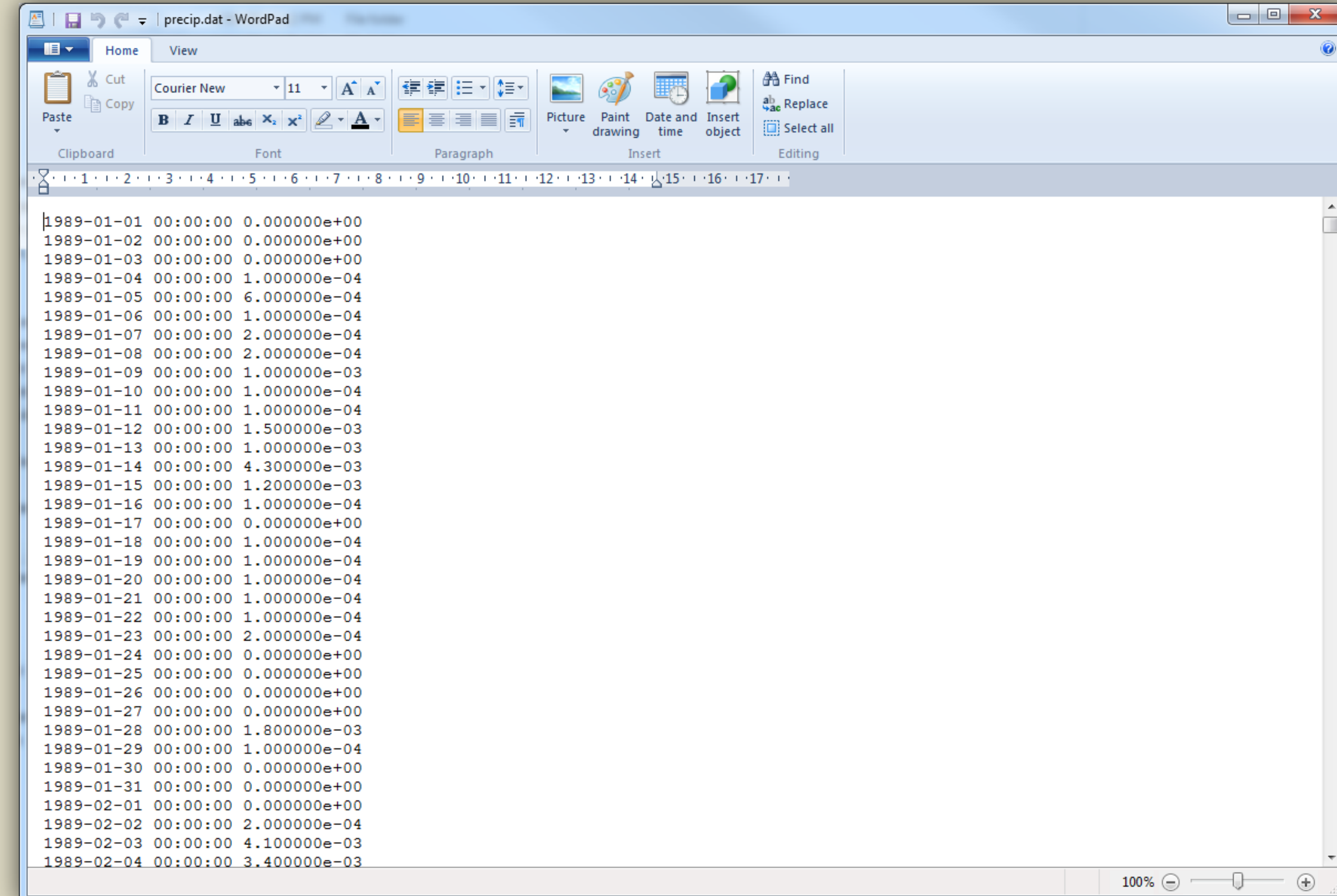
date/time, U10, V10 (10 m wind component in m/s), air pressure (hPa),
T2 (two meter air temperature in °C or K), humidity (e.g., in %), cloud cover (fraction)

Note: if only wind speed is available and not direction/components, then either U10 or V10 may be set to 0.

Meteo data


Air temperature
t2[lat_2=0,lon_2=0] (K)
time
Total cloud cover
oud cover[lat=0,lon=0] (%)
time
Wind-speed
m/s
time
Air pressure
slp[lat_2=0,lon_2=0] (Pa)
time
Relative humidity
sh[lat_3=0,lon_2=0]
time

precip.dat



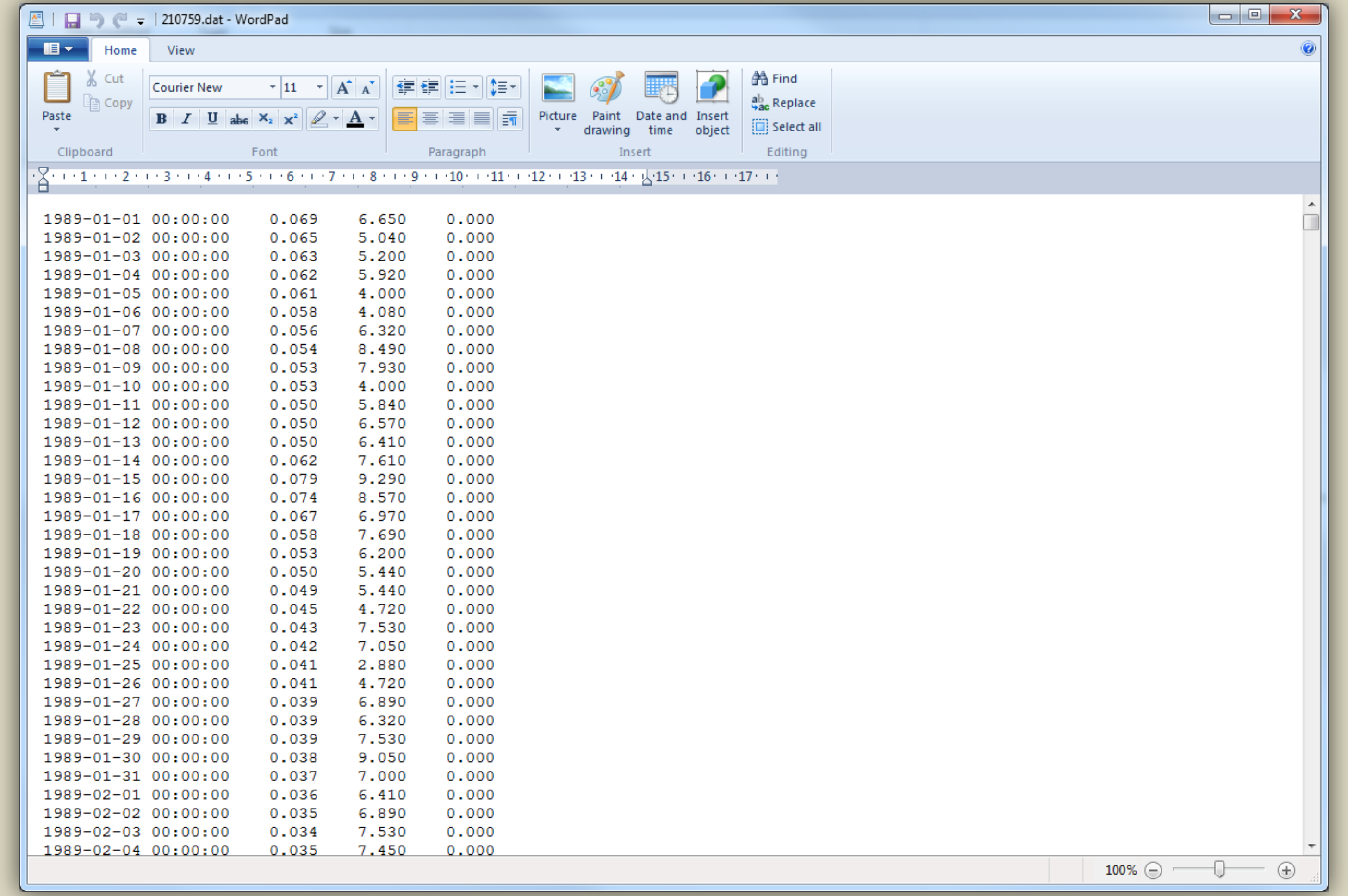
```
1989-01-01 00:00:00 0.000000e+00
1989-01-02 00:00:00 0.000000e+00
1989-01-03 00:00:00 0.000000e+00
1989-01-04 00:00:00 1.000000e-04
1989-01-05 00:00:00 6.000000e-04
1989-01-06 00:00:00 1.000000e-04
1989-01-07 00:00:00 2.000000e-04
1989-01-08 00:00:00 2.000000e-04
1989-01-09 00:00:00 1.000000e-03
1989-01-10 00:00:00 1.000000e-04
1989-01-11 00:00:00 1.000000e-04
1989-01-12 00:00:00 1.500000e-03
1989-01-13 00:00:00 1.000000e-03
1989-01-14 00:00:00 4.300000e-03
1989-01-15 00:00:00 1.200000e-03
1989-01-16 00:00:00 1.000000e-04
1989-01-17 00:00:00 0.000000e+00
1989-01-18 00:00:00 1.000000e-04
1989-01-19 00:00:00 1.000000e-04
1989-01-20 00:00:00 1.000000e-04
1989-01-21 00:00:00 1.000000e-04
1989-01-22 00:00:00 1.000000e-04
1989-01-23 00:00:00 2.000000e-04
1989-01-24 00:00:00 0.000000e+00
1989-01-25 00:00:00 0.000000e+00
1989-01-26 00:00:00 0.000000e+00
1989-01-27 00:00:00 0.000000e+00
1989-01-28 00:00:00 1.800000e-03
1989-01-29 00:00:00 1.000000e-04
1989-01-30 00:00:00 0.000000e+00
1989-01-31 00:00:00 0.000000e+00
1989-02-01 00:00:00 0.000000e+00
1989-02-02 00:00:00 2.000000e-04
1989-02-03 00:00:00 4.100000e-03
1989-02-04 00:00:00 3.400000e-03
```

<u>Forcing data from left to right:</u>

date/time, precipitation (m/s)

Note: precip_factor in airsea.nml can be used to scale from e.g., m/d (set this value to 1 if precipitation is provided in m/s)

Inflow format

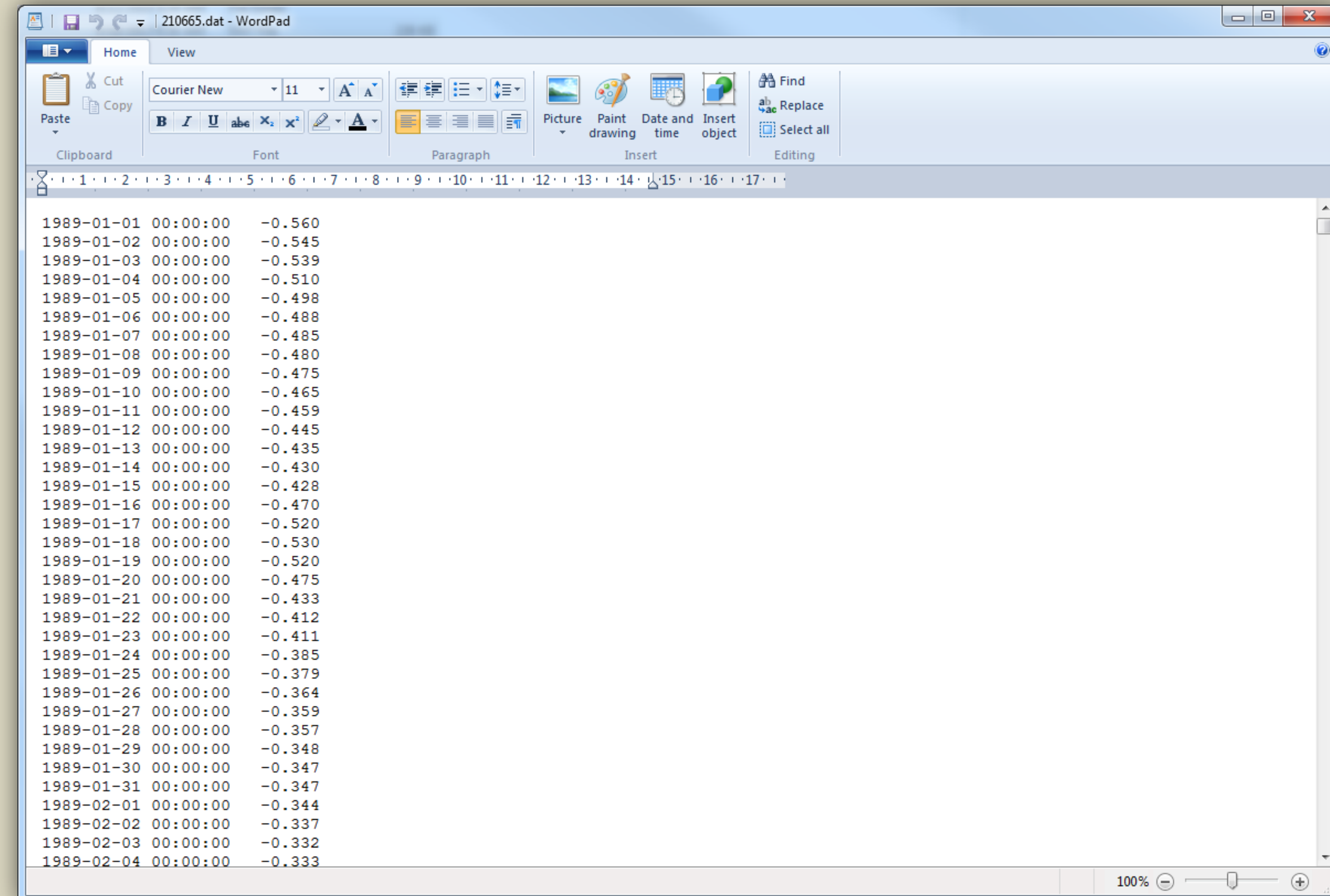


1989-01-01	00:00:00	0.069	6.650	0.000
1989-01-02	00:00:00	0.065	5.040	0.000
1989-01-03	00:00:00	0.063	5.200	0.000
1989-01-04	00:00:00	0.062	5.920	0.000
1989-01-05	00:00:00	0.061	4.000	0.000
1989-01-06	00:00:00	0.058	4.080	0.000
1989-01-07	00:00:00	0.056	6.320	0.000
1989-01-08	00:00:00	0.054	8.490	0.000
1989-01-09	00:00:00	0.053	7.930	0.000
1989-01-10	00:00:00	0.053	4.000	0.000
1989-01-11	00:00:00	0.050	5.840	0.000
1989-01-12	00:00:00	0.050	6.570	0.000
1989-01-13	00:00:00	0.050	6.410	0.000
1989-01-14	00:00:00	0.062	7.610	0.000
1989-01-15	00:00:00	0.079	9.290	0.000
1989-01-16	00:00:00	0.074	8.570	0.000
1989-01-17	00:00:00	0.067	6.970	0.000
1989-01-18	00:00:00	0.058	7.690	0.000
1989-01-19	00:00:00	0.053	6.200	0.000
1989-01-20	00:00:00	0.050	5.440	0.000
1989-01-21	00:00:00	0.049	5.440	0.000
1989-01-22	00:00:00	0.045	4.720	0.000
1989-01-23	00:00:00	0.043	7.530	0.000
1989-01-24	00:00:00	0.042	7.050	0.000
1989-01-25	00:00:00	0.041	2.880	0.000
1989-01-26	00:00:00	0.041	4.720	0.000
1989-01-27	00:00:00	0.039	6.890	0.000
1989-01-28	00:00:00	0.039	6.320	0.000
1989-01-29	00:00:00	0.039	7.530	0.000
1989-01-30	00:00:00	0.038	9.050	0.000
1989-01-31	00:00:00	0.037	7.000	0.000
1989-02-01	00:00:00	0.036	6.410	0.000
1989-02-02	00:00:00	0.035	6.890	0.000
1989-02-03	00:00:00	0.034	7.530	0.000
1989-02-04	00:00:00	0.035	7.450	0.000

<u>Forcing data from left to right:</u>

date/time, discharge (m^3/s), temperature (°C) and salinity (PSU)

Outflow format

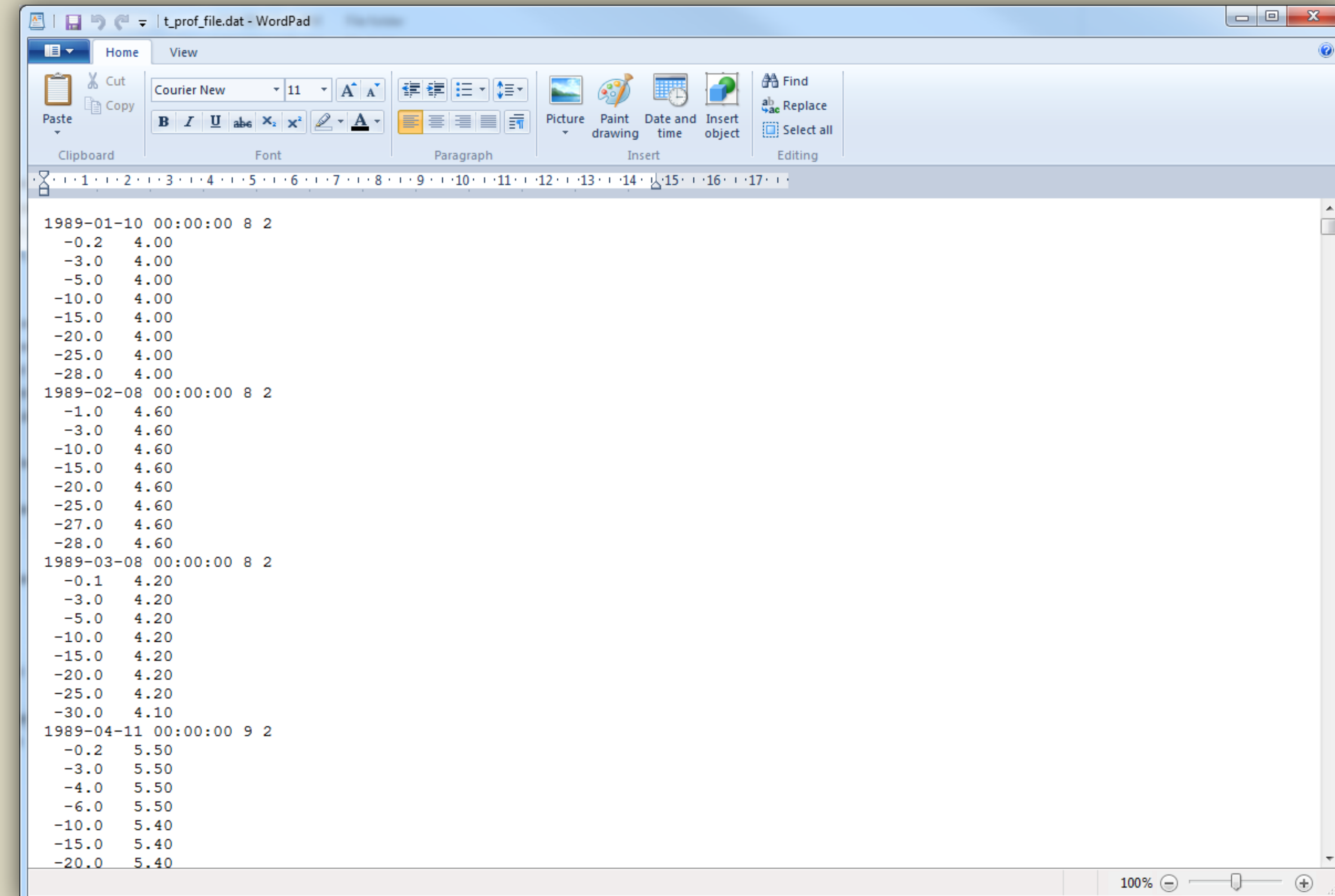



```
1989-01-01 00:00:00   -0.560
1989-01-02 00:00:00   -0.545
1989-01-03 00:00:00   -0.539
1989-01-04 00:00:00   -0.510
1989-01-05 00:00:00   -0.498
1989-01-06 00:00:00   -0.488
1989-01-07 00:00:00   -0.485
1989-01-08 00:00:00   -0.480
1989-01-09 00:00:00   -0.475
1989-01-10 00:00:00   -0.465
1989-01-11 00:00:00   -0.459
1989-01-12 00:00:00   -0.445
1989-01-13 00:00:00   -0.435
1989-01-14 00:00:00   -0.430
1989-01-15 00:00:00   -0.428
1989-01-16 00:00:00   -0.470
1989-01-17 00:00:00   -0.520
1989-01-18 00:00:00   -0.530
1989-01-19 00:00:00   -0.520
1989-01-20 00:00:00   -0.475
1989-01-21 00:00:00   -0.433
1989-01-22 00:00:00   -0.412
1989-01-23 00:00:00   -0.411
1989-01-24 00:00:00   -0.385
1989-01-25 00:00:00   -0.379
1989-01-26 00:00:00   -0.364
1989-01-27 00:00:00   -0.359
1989-01-28 00:00:00   -0.357
1989-01-29 00:00:00   -0.348
1989-01-30 00:00:00   -0.347
1989-01-31 00:00:00   -0.347
1989-02-01 00:00:00   -0.344
1989-02-02 00:00:00   -0.337
1989-02-03 00:00:00   -0.332
1989-02-04 00:00:00   -0.333
```

<u>Forcing data from left to right:</u>

date/time, outflow volume (m^3/s)

profiledata.dat




```
1989-01-10 00:00:00 8 2
  -0.2    4.00
  -3.0    4.00
  -5.0    4.00
 -10.0    4.00
 -15.0    4.00
 -20.0    4.00
 -25.0    4.00
 -28.0    4.00
1989-02-08 00:00:00 8 2
  -1.0    4.60
  -3.0    4.60
 -10.0    4.60
 -15.0    4.60
 -20.0    4.60
 -25.0    4.60
 -27.0    4.60
 -28.0    4.60
1989-03-08 00:00:00 8 2
  -0.1    4.20
  -3.0    4.20
  -5.0    4.20
 -10.0    4.20
 -15.0    4.20
 -20.0    4.20
 -25.0    4.20
 -30.0    4.10
1989-04-11 00:00:00 9 2
  -0.2    5.50
  -3.0    5.50
  -4.0    5.50
  -6.0    5.50
 -10.0    5.40
 -15.0    5.40
 -20.0    5.40
```

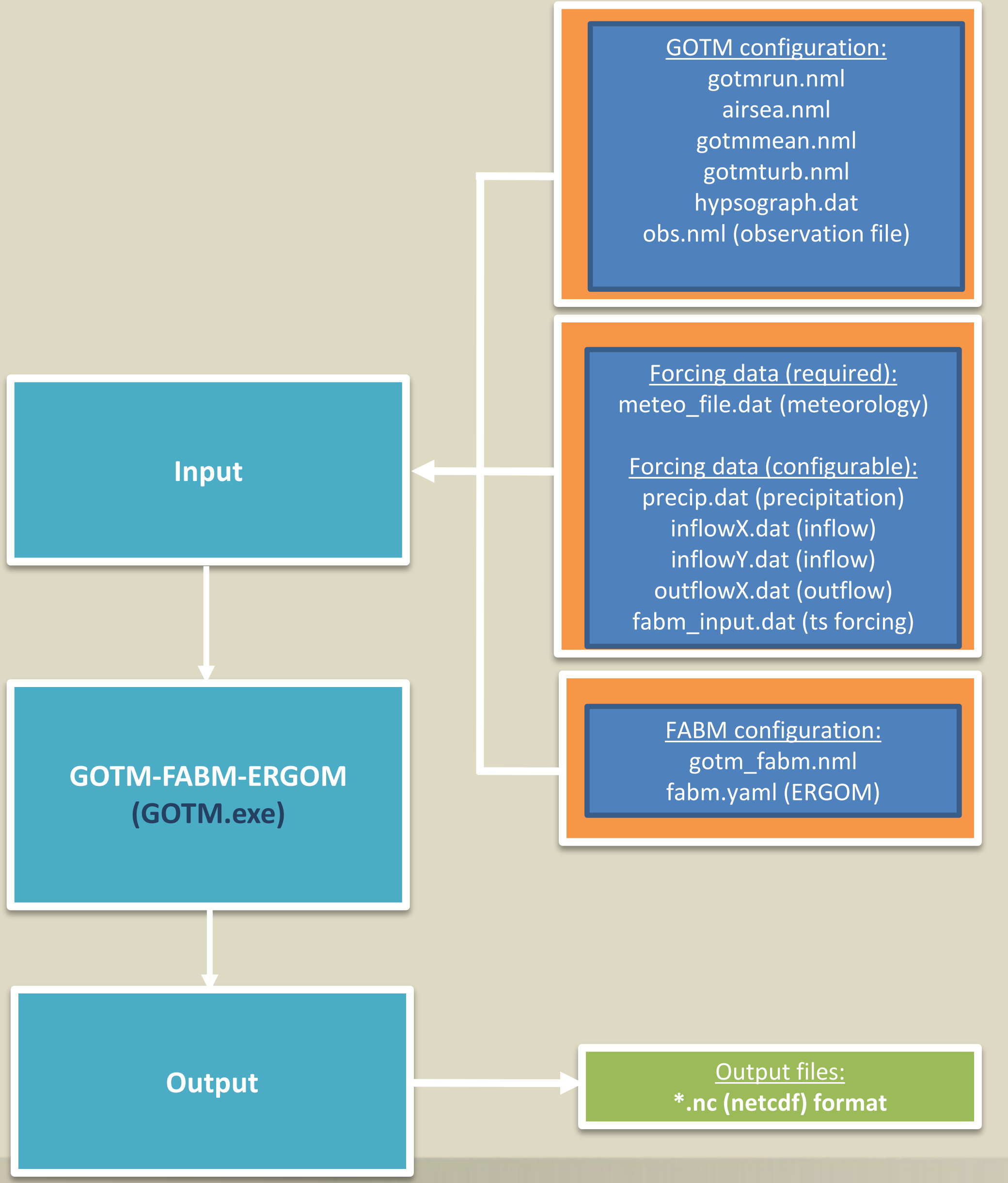
Profile data for initialization and/or relaxation :

First line: date/time and “number of data points” and “read order/case”

Following lines: depth/height and data value

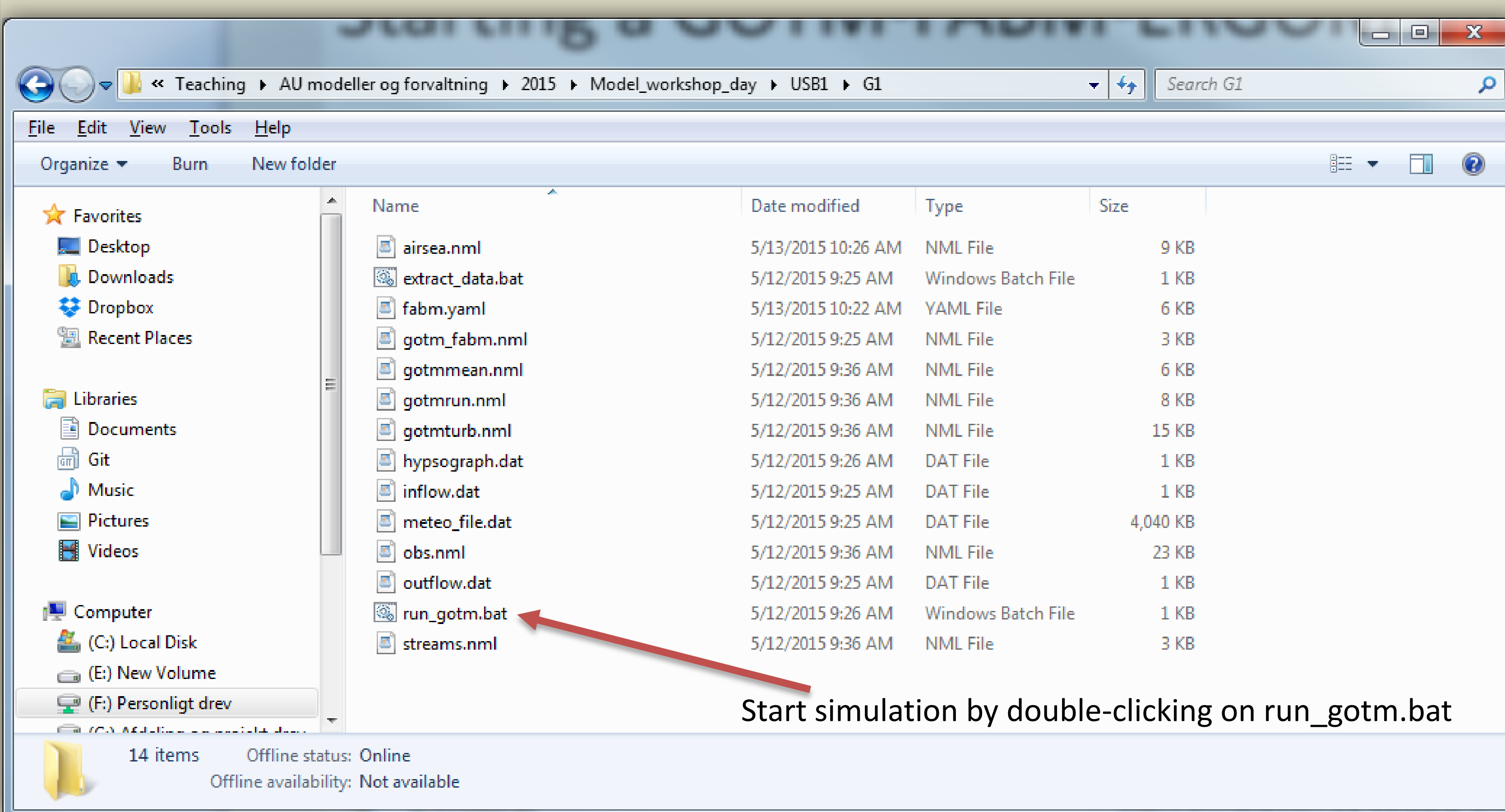
Note: vertical interpolation is done automatically

GOTM-FABM-ERGOM simulation engine

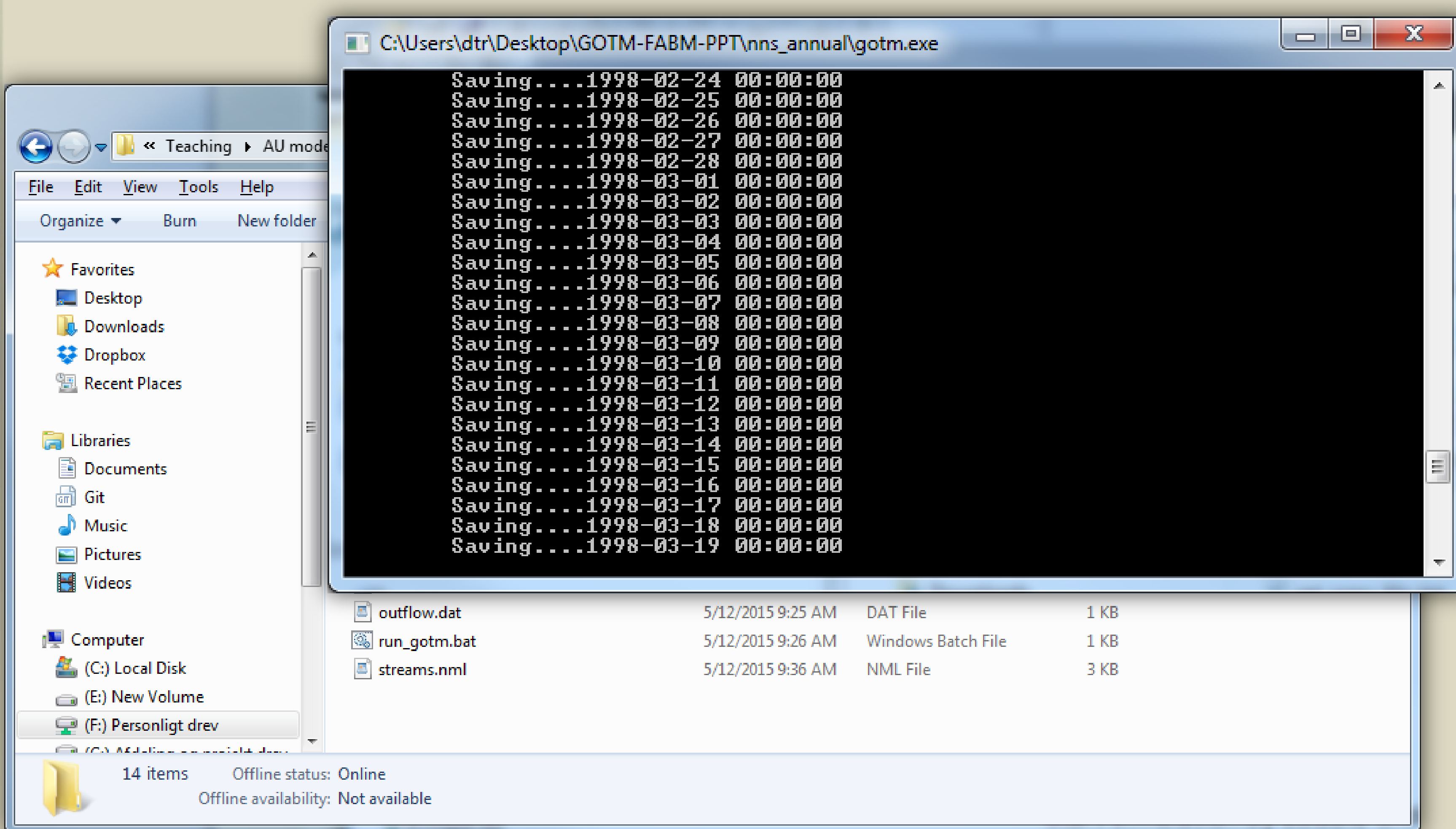


Starting a GOTM-FABM-ERGOM simulation

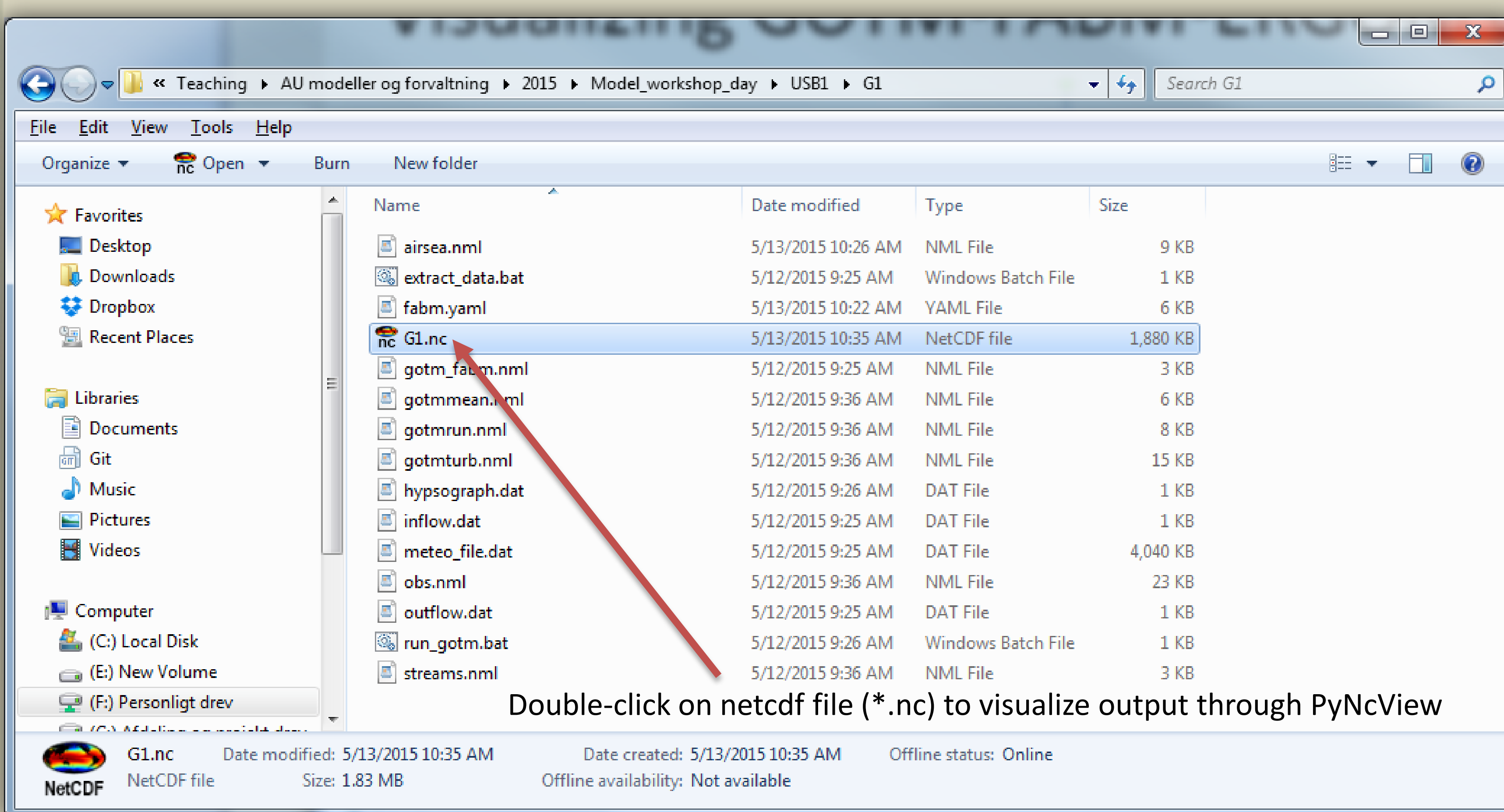
- Go to folder with model input files…



Starting a GOTM-FABM-ERGOM simulation

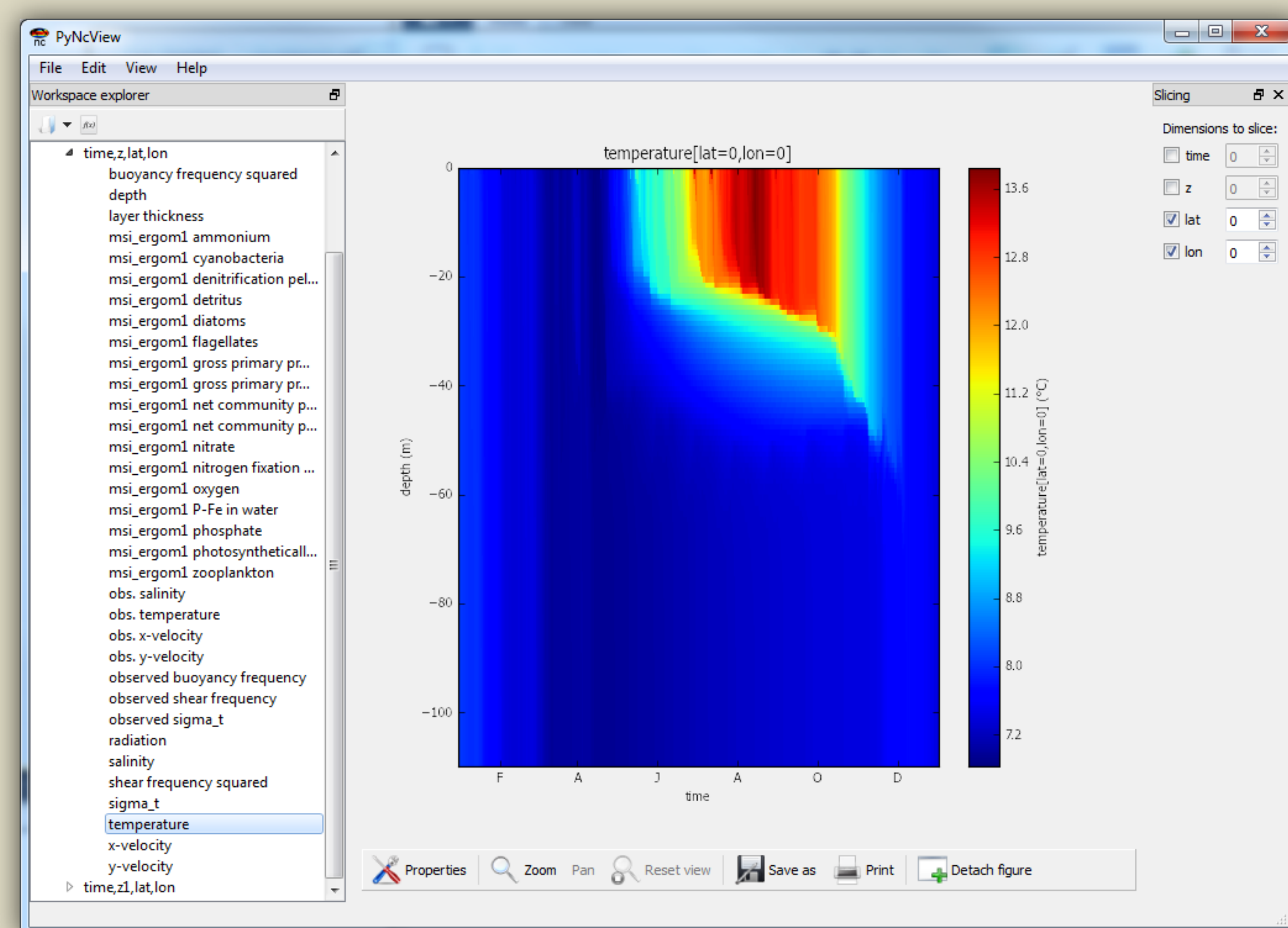


Visualizing GOTM-FABM-ERGOM output



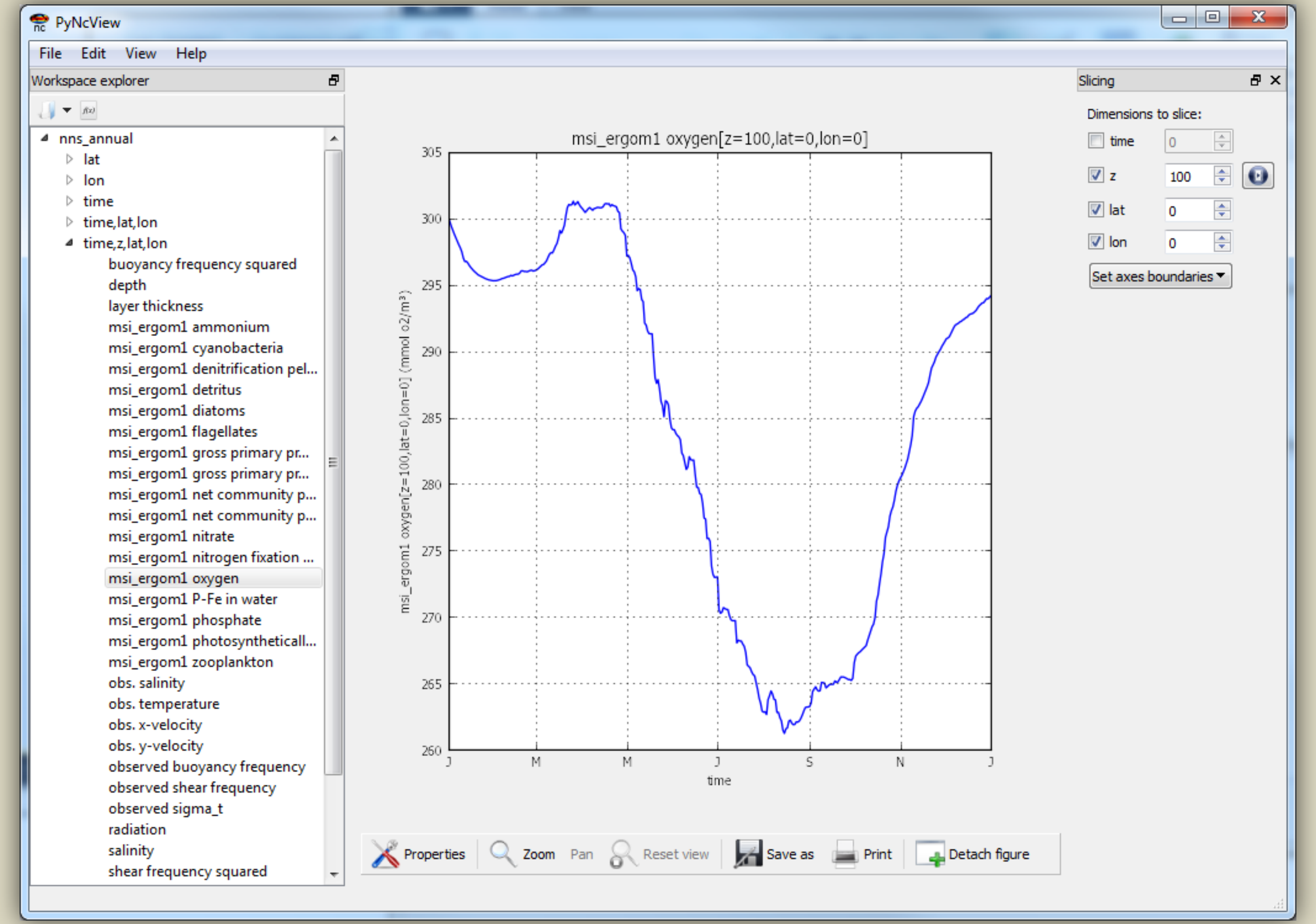
Double-click on netcdf file (*.nc) to visualize output through PyNcView

Visualizing GOTM-FABM-ERGOM output



Counter plots of physical and biogeochemical variables

Visualizing GOTM-FABM-ERGOM output



Line plots of physical and biogeochemical variables
(either as a function of time or a function of depth)

Department of Bioscience, Aarhus University, Denmark

Now you are ready to start using GOTM-FABM-ERGOM but remember:

- Model set up: 5%
- Calibration/validation: 90%
- Scenario simulations: 5%

Assignments - 1

- Use GOTM to run an annual Volta Lake simulation using present day conditions
  - Open log.txt file in editor – check if run went OK
  - Hint – rename output file for future use
- Modify the meteo-forcing
  - Add 3 degrees to air temperature
  - Add 5% to each of the wind components
  - Hint – use Excel for data-manipulation
- Make future scenario simulation
  - Rename output file

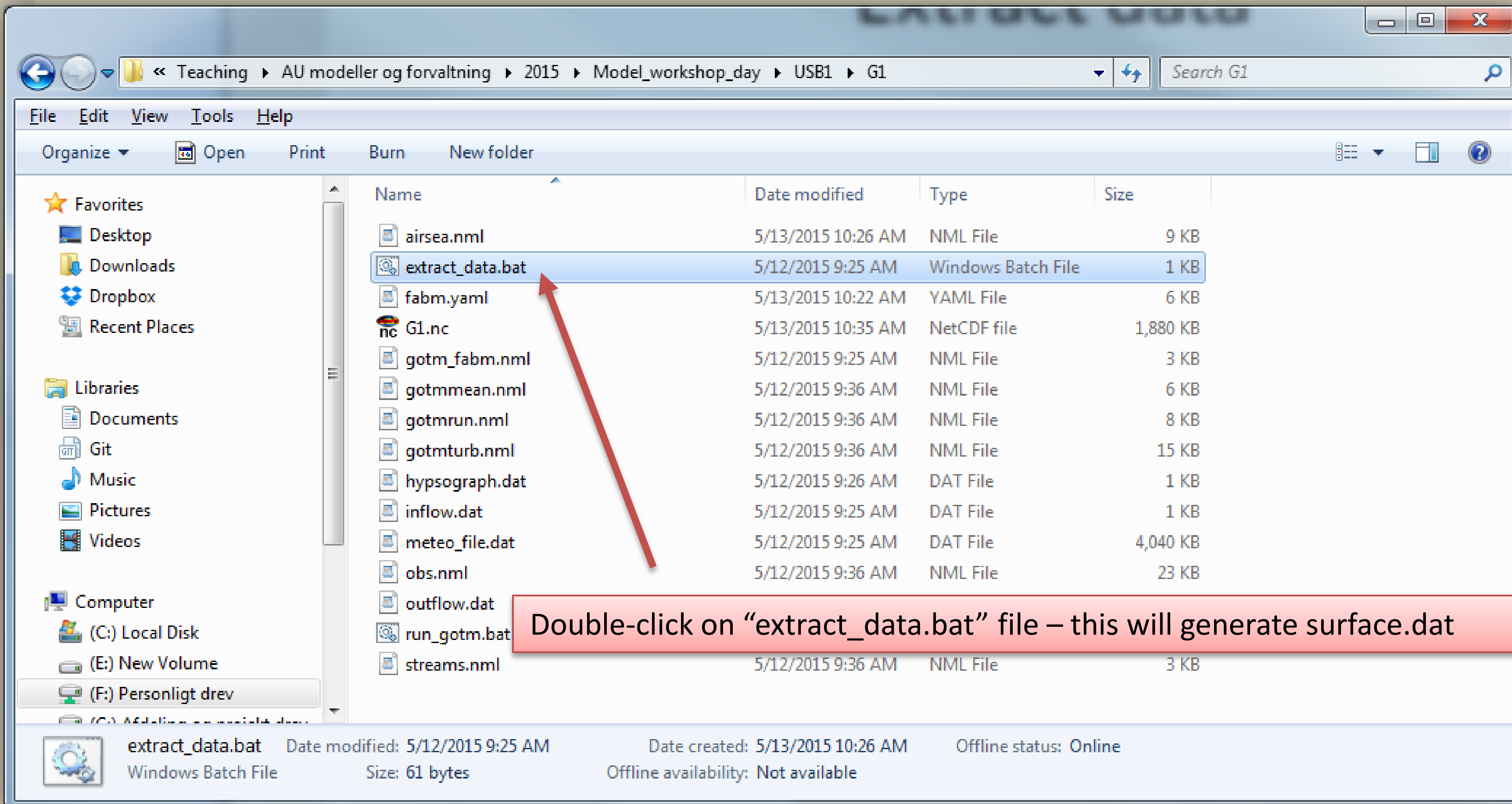
Assignments - 2

- Evaluate heat-budget
  - Double-click on .nc file
- Plot and describe surface and bottom temperature
- Plot difference between surface and bottom
  - use hyper-slabbing
  - volta_lake['temp'][:,1,0,0]
- Evaluate temperature difference between present day and future scenario.

Assignments - 3

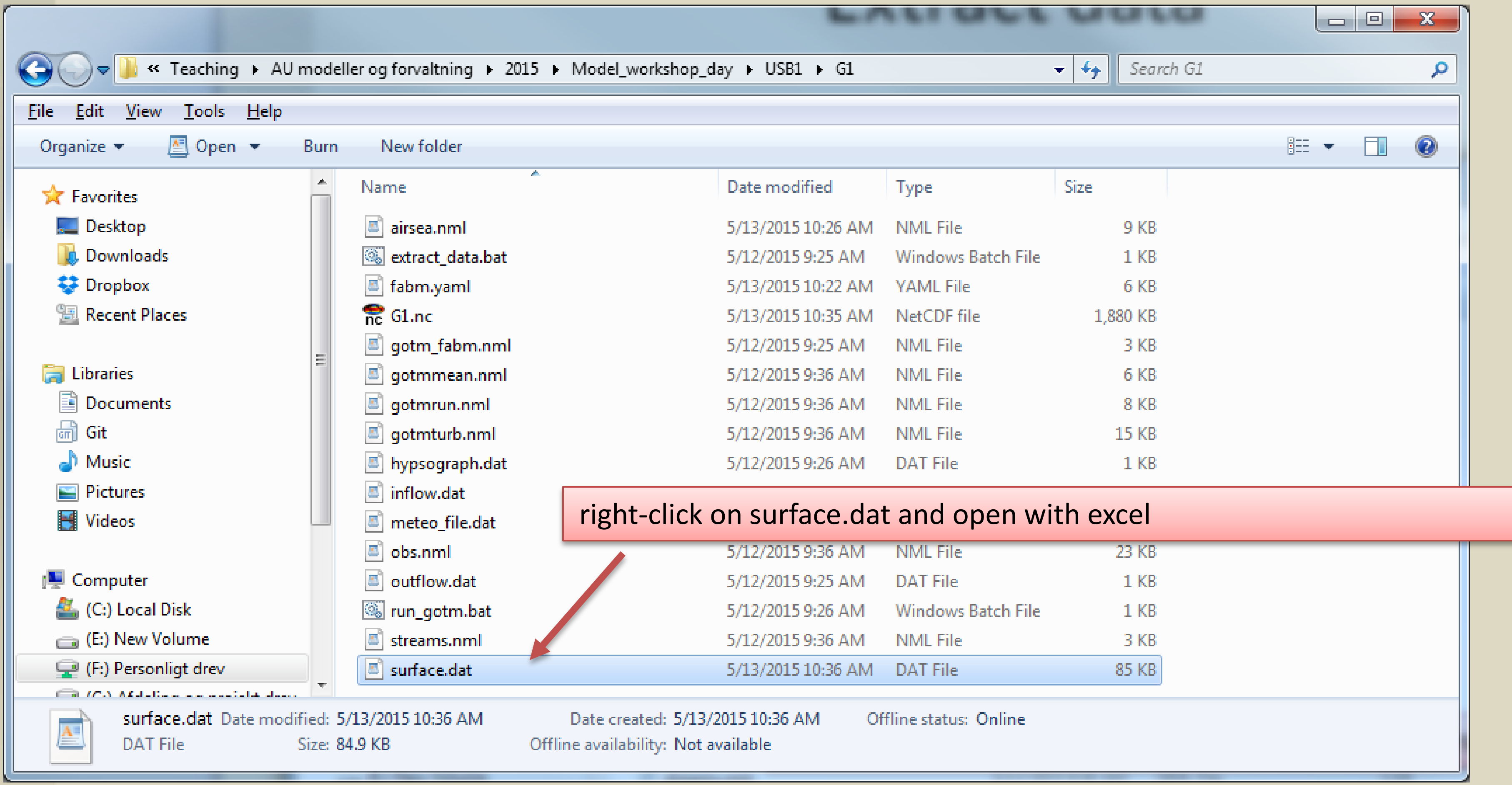
- Evaluate heat-budget
- Plot and describe surface and bottom temperature
- Plot difference between surface and bottom
- Evaluate temperature difference between present day and future scenario.

Extract data

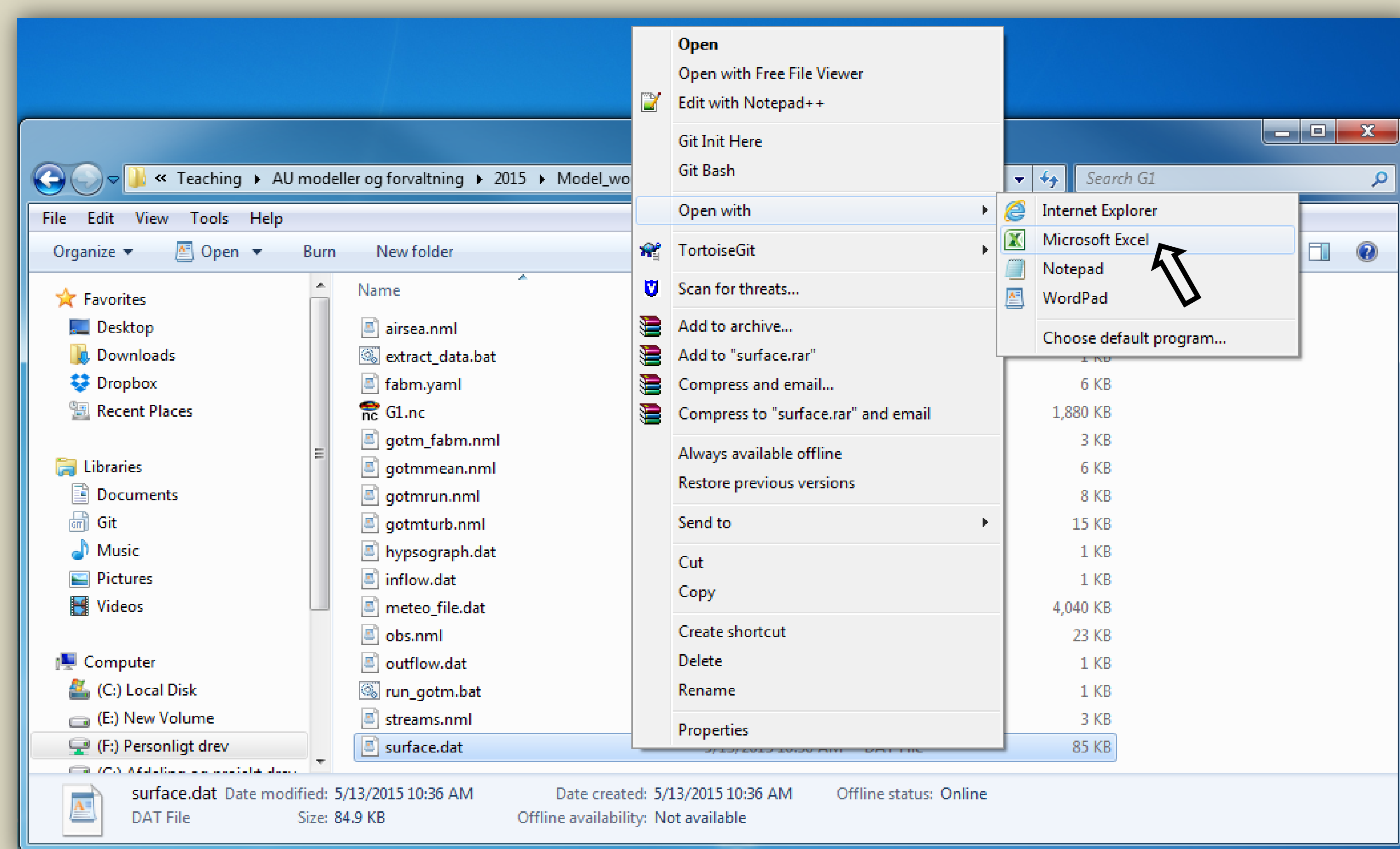


This will extract data from the surface layer of your model simulation (and generate “surface.dat”)

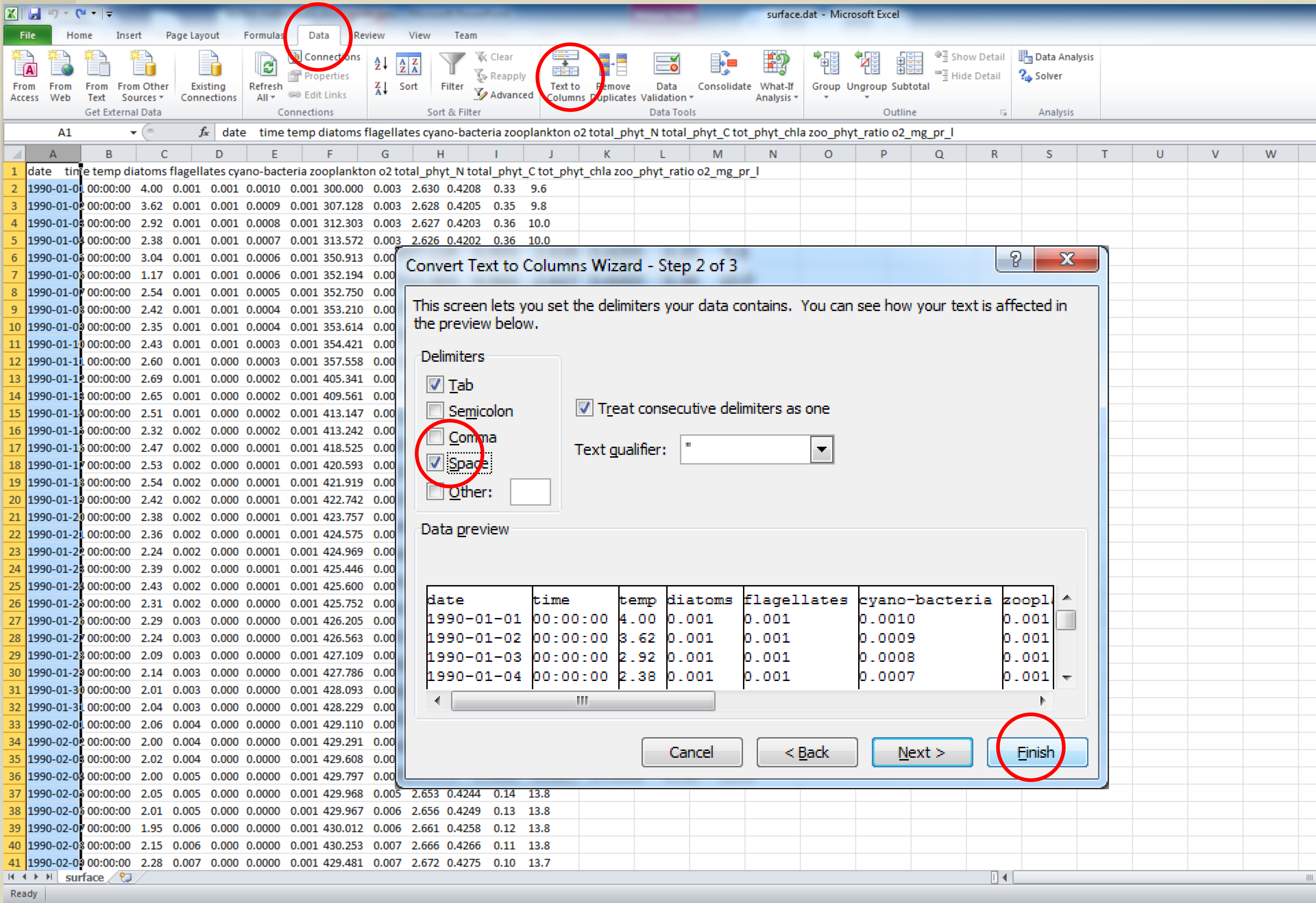
Open file with data



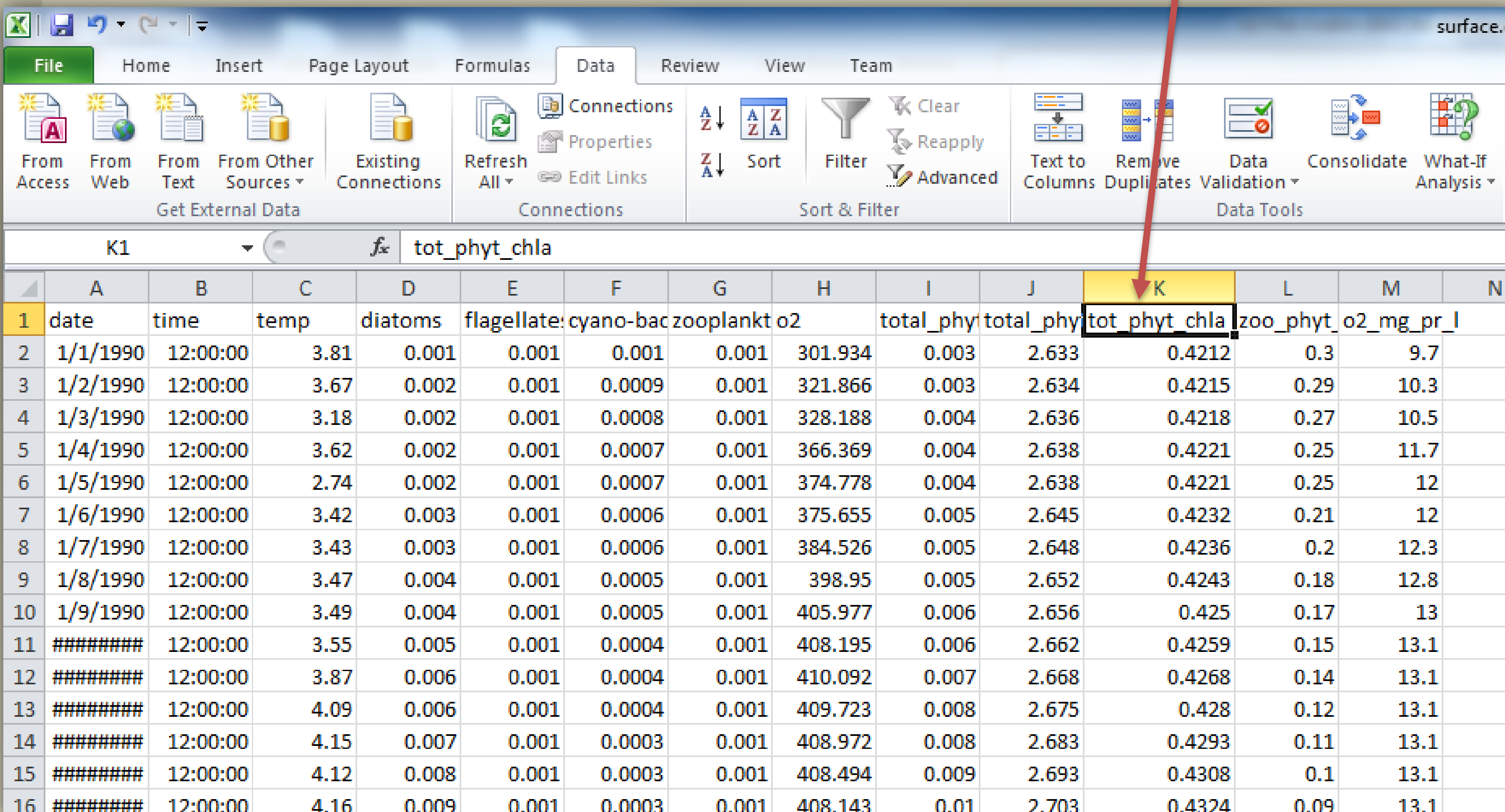
Open file (surface.dat) for example in excel



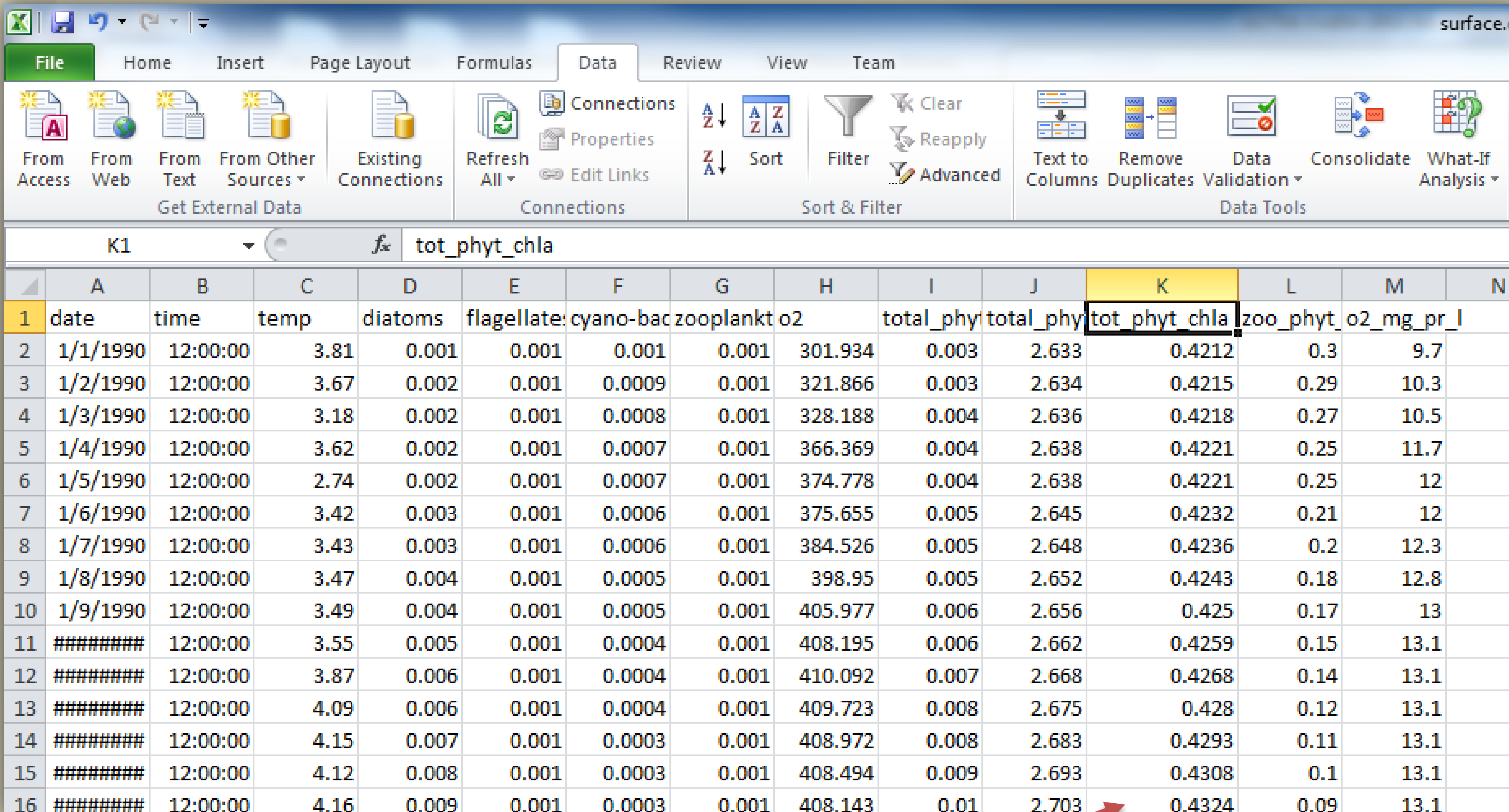
Format data columns…



The column that we are mostly interested in for this exercise is column "K", which contain total phytoplankton concentration in µg chla/L



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	date	time	temp	diatoms	flagellate:	cyano-bac	zooplankt	o2	total_phyt	total_phy	tot_phyt_chla	zoo_phyt_	o2_mg_pr_l	
2	1/1/1990	12:00:00	3.81	0.001	0.001	0.001	0.001	301.934	0.003	2.633	0.4212	0.3	9.7	
3	1/2/1990	12:00:00	3.67	0.002	0.001	0.0009	0.001	321.866	0.003	2.634	0.4215	0.29	10.3	
4	1/3/1990	12:00:00	3.18	0.002	0.001	0.0008	0.001	328.188	0.004	2.636	0.4218	0.27	10.5	
5	1/4/1990	12:00:00	3.62	0.002	0.001	0.0007	0.001	366.369	0.004	2.638	0.4221	0.25	11.7	
6	1/5/1990	12:00:00	2.74	0.002	0.001	0.0007	0.001	374.778	0.004	2.638	0.4221	0.25	12	
7	1/6/1990	12:00:00	3.42	0.003	0.001	0.0006	0.001	375.655	0.005	2.645	0.4232	0.21	12	
8	1/7/1990	12:00:00	3.43	0.003	0.001	0.0006	0.001	384.526	0.005	2.648	0.4236	0.2	12.3	
9	1/8/1990	12:00:00	3.47	0.004	0.001	0.0005	0.001	398.95	0.005	2.652	0.4243	0.18	12.8	
10	1/9/1990	12:00:00	3.49	0.004	0.001	0.0005	0.001	405.977	0.006	2.656	0.425	0.17	13	
11	########	12:00:00	3.55	0.005	0.001	0.0004	0.001	408.195	0.006	2.662	0.4259	0.15	13.1	
12	########	12:00:00	3.87	0.006	0.001	0.0004	0.001	410.092	0.007	2.668	0.4268	0.14	13.1	
13	########	12:00:00	4.09	0.006	0.001	0.0004	0.001	409.723	0.008	2.675	0.428	0.12	13.1	
14	########	12:00:00	4.15	0.007	0.001	0.0003	0.001	408.972	0.008	2.683	0.4293	0.11	13.1	
15	########	12:00:00	4.12	0.008	0.001	0.0003	0.001	408.494	0.009	2.693	0.4308	0.1	13.1	
16	########	12:00:00	4.16	0.009	0.001	0.0003	0.001	408.143	0.01	2.703	0.4324	0.09	13.1	



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	date	time	temp	diatoms	flagellate	cyano-bac	zooplankt	o2	total_phyt	total_phy	tot_phyt_chla	zoo_phyt_	o2_mg_pr_l	
2	1/1/1990	12:00:00	3.81	0.001	0.001	0.001	0.001	301.934	0.003	2.633	0.4212	0.3	9.7	
3	1/2/1990	12:00:00	3.67	0.002	0.001	0.0009	0.001	321.866	0.003	2.634	0.4215	0.29	10.3	
4	1/3/1990	12:00:00	3.18	0.002	0.001	0.0008	0.001	328.188	0.004	2.636	0.4218	0.27	10.5	
5	1/4/1990	12:00:00	3.62	0.002	0.001	0.0007	0.001	366.369	0.004	2.638	0.4221	0.25	11.7	
6	1/5/1990	12:00:00	2.74	0.002	0.001	0.0007	0.001	374.778	0.004	2.638	0.4221	0.25	12	
7	1/6/1990	12:00:00	3.42	0.003	0.001	0.0006	0.001	375.655	0.005	2.645	0.4232	0.21	12	
8	1/7/1990	12:00:00	3.43	0.003	0.001	0.0006	0.001	384.526	0.005	2.648	0.4236	0.2	12.3	
9	1/8/1990	12:00:00	3.47	0.004	0.001	0.0005	0.001	398.95	0.005	2.652	0.4243	0.18	12.8	
10	1/9/1990	12:00:00	3.49	0.004	0.001	0.0005	0.001	405.977	0.006	2.656	0.425	0.17	13	
11	########	12:00:00	3.55	0.005	0.001	0.0004	0.001	408.195	0.006	2.662	0.4259	0.15	13.1	
12	########	12:00:00	3.87	0.006	0.001	0.0004	0.001	410.092	0.007	2.668	0.4268	0.14	13.1	
13	########	12:00:00	4.09	0.006	0.001	0.0004	0.001	409.723	0.008	2.675	0.428	0.12	13.1	
14	########	12:00:00	4.15	0.007	0.001	0.0003	0.001	408.972	0.008	2.683	0.4293	0.11	13.1	
15	########	12:00:00	4.12	0.008	0.001	0.0003	0.001	408.494	0.009	2.693	0.4308	0.1	13.1	
16	########	12:00:00	4.16	0.009	0.001	0.0003	0.001	408.143	0.01	2.703	0.4324	0.09	13.1	

Average value for entire column is easily derived using Excel function (“average”)