Computer Science in Ocean and Climate Research

Lecture 7: Spin-up – Computation of Steady States

Prof. Dr. Thomas Slawig

CAU Kiel
Dep. of Computer Science

Summer 2020

Contents

Contents

Recall: Mathematical formulation of a climate model

- Spatial discretization already done ...

$\rightsquigarrow$ Initial value problem (IVP) for a system of ordinary differential equations (ODEs)

$$\dot{y}(t) = f(y(t), t), \quad t \geq 0,$$

with initial value

$$y(t_0) = y_0.$$

- Fully discrete version: discretize also time, approximation $y_k \approx y(t_k)$.

$\rightsquigarrow$ General time-stepping (Euler, improved Euler ...):

$$\left.\begin{array}{rcl} y_{k+1} & = & y_k + \Delta t \Phi(y_k, t_k) \\ t_{k+1} & = & t_k + \Delta t \end{array}\right\} k = 0, 1, \ldots, n-1.$$

Steady States

- Climate system is time-dependent
- Periodic forcing (annual and daily cycles)
- Internal variability

$\rightsquigarrow$ There are no **stationary solutions**, i.e., solutions that satisfy

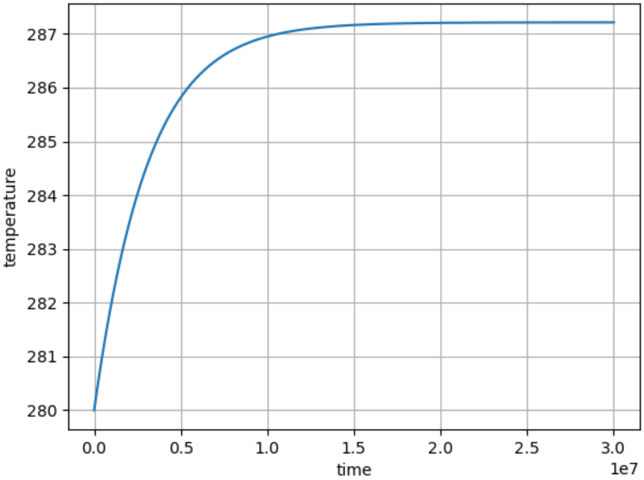
$$\dot{y}(t) = f(y(t), t) = 0, \qquad \text{for all } t \geq 0,$$

- i.e., solutions y that are constant w.r.t. time and satisfy

$$f(y, t) = 0 \qquad \text{for all } t \geq 0.$$

- However, for some investigations it makes sense to look for stationary solutions:
  - Simplified models, that do not model the daily and annual cycles, often: annual time-steps.
  - Parameter studies and model-data comparison using climatological (averaged) data.

Example stationary solution: 0-D energy balance model



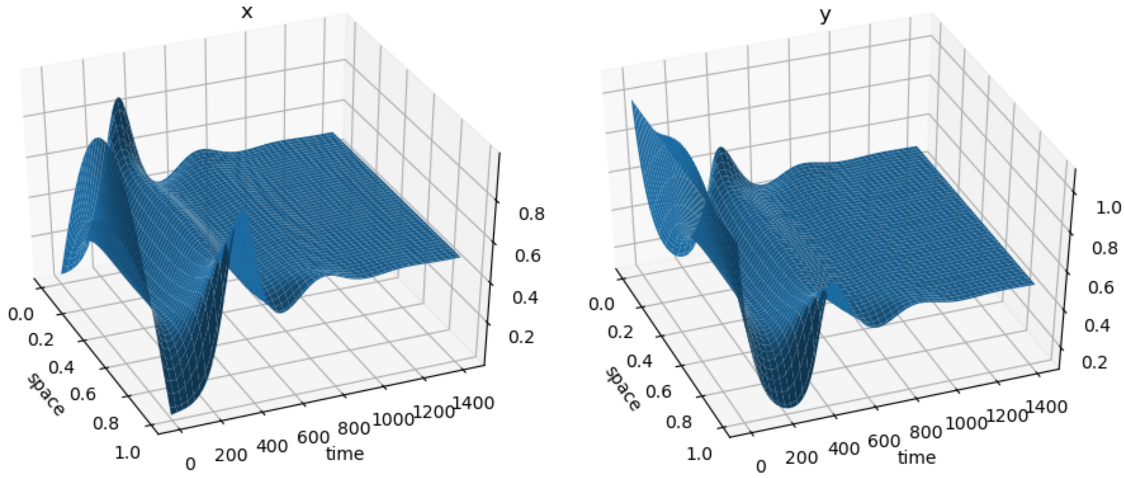
Steady States

- Also the predator-prey model with diffusion

$$\left.\begin{array}{rcl} f_1(x_k, y_k) & = & Dx_k + x_k * (\alpha - \beta y_k - \lambda x_k) \\ f_2(x_k, y_k) & = & Dy_k + y_k * (\delta x_k - \gamma - \mu y_k) \end{array}\right\} \quad k = 0, 1, \ldots,$$

  tends to a stationary solution for many parameter settings.
- Diffusion levels out the differences.
- How can we compute a stationary state?
- In many cases, we can just compute the time-stepping as long as the difference between two time-steps is getting small.
- This is called a **spin-up**.

Example: Predator prey model with diffusion



Pseudo-time-stepping or Spin-up

- Assume we have a model

$$\dot{y}(t) = f(y(t), t), \quad t \geq 0,$$

  and a time-stepping scheme

$$y_{k+1} = y_k + \Phi(y_k, t_k), \quad k = 0, 1, \dots$$

- We want to compute a stationary solution.
- We call the following iteration a **spin-up** or a **pseudo-time-stepping scheme**:
  1. Choose an initial state y_0 and a desired accuracy $\epsilon > 0$.
  2. For $k = 0, 1, \dots$: Compute

$$y_{k+1} = y_k + \Phi(y_k, t_k)$$

  until $\|y_{k+1} - y_k\| < \epsilon$.

Pseudo-time-stepping or spin-up

- Mathematical framework: spin-up = fixed-point iteration:
- Convergence, if the iteration function $G : \mathbb{R}^n \to \mathbb{R}^n$ is a contractive mapping, i.e.,

$$\exists L < 1 : \quad \|G(y) - G(z)\| \leq L\|y - z\| \quad \text{for all } y, z \in \mathbb{R}^n$$

- ... or $\|G'(y)\| \leq L < 1$ for all $y \in \mathbb{R}^n$.
- Generally, this is difficult to show.
- Advantages and disadvantages of a spin-up:

\+ Original time-integration can be used $\rightsquigarrow$ no implementation effort.

– Generally not clear: Does the algorithm converge?

– Is there a unique solution?

– Does the solution depend on the initial guess y_0?

– Slow convergence: Example: ocean model, takes 3'000-10'000 years model time.

Contents

Steady annual cycles

- Periodic forcing with annual cycles:

$\rightsquigarrow$ It makes sense to look for (annually) periodic solutions.

- They satisfy

$$y(t+T) = y(t), \quad \text{for all } t \geq 0,$$

where $T =$ one year.

- In the time-discrete model: Assume one year needs K steps. We have a nested loop:
- For $\ell = 0, 1, \ldots$:

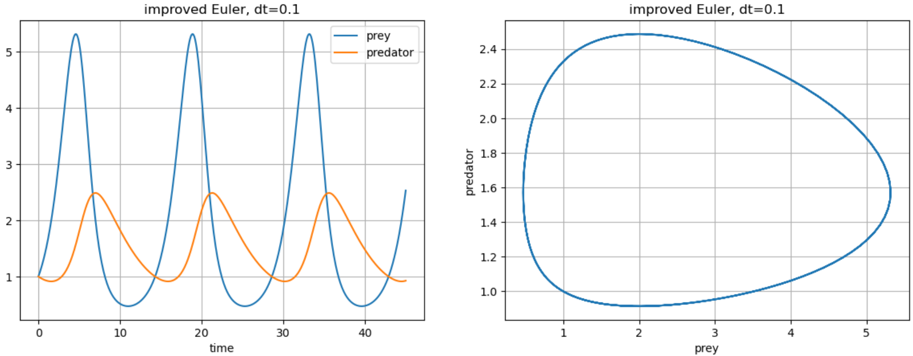
For $k = 0, \ldots, K-1$: Compute

$$y_{\ell K+k+1} = y_{\ell K+k} + \Phi(y_{\ell K+k}, t_{\ell K+k})$$

until $\|y_{(\ell+1)K} - y_{\ell K}\| < \epsilon$.

- Different stopping criterion: check after K steps (1 year).

Example periodic solution: predator-prey without diffusion

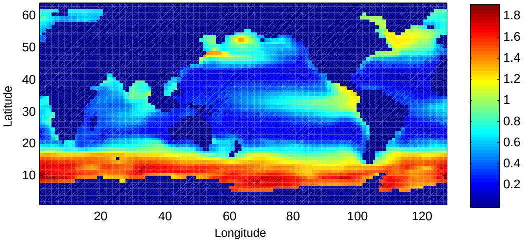


- Here: no spin-up, because solution is directly periodic.
- Different case: Ocean circulation, takes 3’000 to 10’000 years to reach periodic state.

Contents

Example periodic solution: 3-D marine ecosystem

- Extension of predator-prey model.
- Marine ecosystem: nutrients and plankton in the ocean circulation.
- Transport and diffusion coming from ocean circulation model, ...
- ... in our case: pre-computed matrices T, D.
- Explicit transport, horizontal diffusion and ecosystem model.



nutrient distribution at the surface

- Implicit vertical diffusion (small vertical resolution $\rightsquigarrow$ small time-step for explicit method).
- Time-stepping with operator-splitting, y_k contains nutrients, plankton, ...

$$\text{explicit step:} \qquad y_{k+\frac{1}{2}} = y_k + \Delta t \left(T y_k + D_{hor} y_k + q(y_k) \right),$$

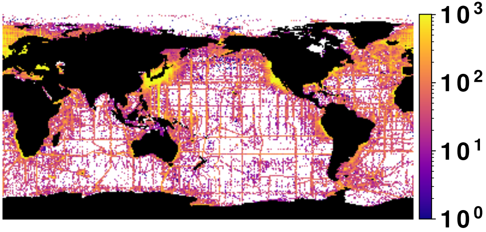
$$\text{implicit step:} \qquad \text{solve } \left(I - \Delta t D_{vert}\right) y_{k+1} = y_{k+\frac{1}{2}}, \quad k = 0, \ldots, n-1,$$

Example periodic solution: 3-D marine ecosystem

- Spatial resolution 2.8125° : 128×64 horizontal boxes $\times 15$ vertical layers: $\approx 52'000$ boxes.
- Full matrices cannot be stored.

$\rightsquigarrow$ Need for sparse matrices and special library.

- PETSc: Portable Extendable Toolkit for Scientific Computing (C, C++).
- Therein: spatial parallelization with distributed memory and MPI already realized.
- Thus, hidden from the user.
- Different ecosystem models.

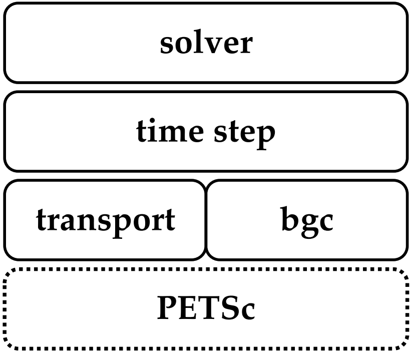


number of available measurement data

- Aim: Parameter optimization: Find best parameter for given climatological data.
- Optimization means: hundreds of model evaluations.

$\rightsquigarrow$ Simulation has to be fast.

Example 3-D marine ecosystem: code structure




```
main
├→ InitWithFilePath
│   ├→ GeometryInit
│   ├→ LoadInit
│   ├→ BGCInit
│   ├→ TransportInit
│   ├→ TimeStepInit
│   └→ SolverInit
├→ Solver
│   └→ TimeStepPhi
│       └→ TimeStepPhiStep
│           ├→ BGCStep
│           └→ TransportStep
└→ Final
    ├→ SolverFinal
    ├→ TimeStepFinal
    ├→ TransportFinal
    ├→ BGCFinal
    ├→ LoadFinal
    └→ GeometryFinal
```

Contents

Computation on a Linux High Performance Computing (HPC) cluster

University description of the HPC cluster:
The NEC HPC-Linux-Cluster is part of a hybrid NEC high performance system at the University Computing Centre. It consists of a NEC SX-Aurora TSUBASA vector system with a theoretical peak performance of 137,6 TFlops and the scalar Linux-Cluster with in total 6192 cores and a theoretical peak performance of 404.4 TFlops. Both systems can be accessed by the same front end and have access to a shared 5 PB global file system. Batch computations are managed by an overall batch system (NQSV).

Computation on a Linux High Performance Computing (HPC) cluster

- We use the NEC-HPC Linux Cluster of Kiel University.
- Front end NEC HPC 124Rh-1
  - 4 systems each with
  - 2 Intel Xeon Gold 6130 (Skylake-SP) processors (2.1 GHz)
  - 32 cores per node and 768 GB main memory
- NEC Linux-Cluster batch nodes
  - 172 nodes each with 2 Intel Xeon Gold 6130 (Skylake-SP) processors (2.1 GHz)
  - 32 cores per node and 192 GB main memory
  - 18 nodes each with 2 Intel Xeon E5-2680v3 (Haswell-EP) processors (2.5 GHz)
  - 24 cores per node and 128 GB main memory
  - Connection: EDR infiniband
- Operating system: Red Hat Linux (Release 7.4)

Access to an HPC cluster

- The following steps are typical for the use of an HPC cluster.
- User account:
  - For the use of an HPC system an extra validation is required.
- System access:
  - After an interactive login, the user is automatically redirected to one of the available front ends.
  - Access is only possible via a SSH-connection established within the internal network of the university.
- Computations on the HPC cluster should be primarily performed in batch mode.
- Only short interactive operations (e.g. compilation, test of scripts or programs) should be performed on the front end.

Compilation of a simulation software on an HPC cluster

- We use the 3-D marine ecosystem simulation software described above as example.
- The software is called Metos3D (Marine Ecosystem Toolkit for Simulation and Optimization in 3D).
  - The software can be found on `https://metos3d.github.io`
  - It can be installed on the HPC cluster using the command provided on the above website.
- To compile the software on the HPC system, we need three software packages
  - C/Fortran compiler
  - MPI compiler (parallel computing)
  - PETSc library.
- They are installed on the HPC system and have just to be loaded before compilation:

```
module load intel17.0.4 intelmpi17.0.4 petsc3.7.6intel
```

- Then the simulation package with one exemplary ecosystem model (here: N-DOP model) can be compiled:

```
metos3d simpack N-DOP
```

Compilation of a simulation software on an HPC cluster

- The compilation generates an executable named
  `metos3d-simpack-N-DOP.exe`
- An option file with simulation parameters has to be provided. In the model directory, here
  `.metos3d/model/model/N-DOP/option`
  there is a file `test.N-DOP.option.txt` that defines a short test run.
- We can perform this test run with:
  `./metos3d-simpack-N-DOP.exe model/N-DOP/option/test.N-DOP.option.txt`
- To run a longer simulation, we have to start a batch job.

Running a simulation using a batch system

- Longer runs shall not be performed on the login node or front end, but submitted using a batch system.
- There different of these systems, depending on the actual HPC system used.
- We have to write a batch file.
- This is a text file that contains all necessary information
  - on the software
  - the used libraries (i.e., modules in the HPC system)
  - output files
  - optional: choice of queue (there usually exist different queues for long/shorter runs or for special user groups)
  - desired or needed ressources (processors, memory and time).
- The batch file is then submitted using the `qsub` command on the frond end.
- For a longer run, the option file of the simulation software has to be modified accordingly.
- Detailed descriptions are usually provided on the website of the HPC system.

Example batch file (batch system: NQSII)

```
#!/bin/bash
#PBS -T intmpi
#PBS -b 1
#PBS -l cpunum_job=16
#PBS -l elapstim_req=02:00:00
#PBS -l memsz_job=10gb
#PBS -N test-N-DOP
#PBS -o test-N-DOP.out
#PBS -j o
#PBS -q clexpress
cd $PBS_O_WORKDIR
module load intel17.0.4 intelmpi17.0.4 petsc3.7.6intel
mpirun $NQSII_MPIOPTS -np 16 ./metos3d-simpack-N-DOP.exe option-file.txt
qstat -f ${PBS_JOBID/0:}
```

Example batch file

- This is a bash shell script:
  `#!/bin/bash`
- Type of process:
  `#PBS -T intmpi`
- Number of nodes:
  `#PBS -b 1`
- Number of requested cores per nodes:
  `#PBS -l cpunum_job=16`
- Needed wall-clock time:
  `#PBS -l elapstim_req=02:00:00`
- Main memory required per node:
  `#PBS -l memsz_job=10gb`
- Name of batch job:
  `#PBS -N test-N-DOP`

Example batch file

- Files for standard-output:
  `#PBS -o test-N-DOP.out`
- Write standard-error to standard-output:
  `#PBS -j o`
- Requested batch class (queue):
  `#PBS -q clexpress`
- Change to working directory:
  `cd $PBS_O_WORKDIR`
- Load modules
  `module load intel17.0.4 intelmpi17.0.4 petsc3.7.6intel`
- Run programm:
  `mpirun $NQSII_MPIOPTS -np 16 ./metos3d-simpack-N-DOP.exe option-file.txt`
- Give queue status:
  `qstat -f ${PBS_JOBID/0:}`

What is important

- It is often necessary or desired to compute a steady (stationary or periodic) solution of climate models ...
- ... either for parameter studies, model-data comparison, or model parameter optimization.
- This can be done by running a time integration until no (big) changes from time-step to time-step (or year to year for anually periodic states) appear.
- This procedure is called spin-up or pseudo-time-stepping.
- Mathematically, this is a fixed-point iteration.
- It might be slow and take thousands of years model time.
- But it can be easily performed using the time integrator used for transient model runs.
- Simulation on an HPC system is different from one running on a simple PC or workstation.
- The needed software has to be installed on the HPC system.
- Actual runs have to be submitted using a batch system.