

PMPP 2015/16



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Example Project



(Preliminary) Course Schedule

you are here



12.10.2015	Introduction to PMPP
13.10.2015	Lecture Example Project, CUDA Programming 1
19.10.2015	Lecture CUDA Programming 2
20.10.2015	Lecture CUDA Programming 3
26.10.2015	Introduction Final Projects, Exercise 1 assigned
27.10.2015	Questions and Answers (Q&A)
02.11.2015	Lecture, Final Projects assigned, Ex. 1 due, Ex. 2 assigned
03.11.2015	Questions and Answers (Q&A)
09.11.2015	Lecture, Exercise 2 due
10.11.2015	Lecture
16.11.2015	Questions and Answers (Q&A)
17.11.2015	Questions and Answers (Q&A)
23.11.2015	1 st Status Presentation Final Projects
24.11.2015	1 st Status Presentation Final Projects (continued)



(Preliminary) Course Schedule

30.11.2015

01.12.2015

07.12.2014

08.12.2014

14.12.2014

15.12.2014

Christmas Break

11.01.2015 2nd Presentation Status Final Projects (1)

12.01.2015 2nd Presentation Status Final Projects (2)

...

01.02.2015

02.02.2015

08.02.2015 final Presentation Status Final Projects (1)

09.02.2015 final Presentation Status Final Projects (2)



- all exercises and projects will be solved using NVIDIA CUDA on Linux systems
 - all exercises and projects will run on the HHLR
 - **register in TuCAN IMMEDIATELY, i.e., by October 13th so that we can create your accounts**
 - **fill in and sign account form and return it to us (ASAP)**
- also possible but not really recommended to run CUDA on your own system
 - no support provided
 - all exercises and projects must run on HHLR for grading

Nutzungsordnung des Hochleistungsrechners der TU Darmstadt Nutzung durch Studierende im Rahmen einer Lehrveranstaltung



1. Präambel

Diese Nutzungsordnung legt fest, nach welchen Regeln der Hochleistungsrechner von Studierenden im Rahmen einer Lehrveranstaltung der TU Darmstadt benutzt werden darf.

Der Hochleistungsrechner steht über das des Wissenschaftlerinnen und Wissenschaftlern der TU Darmstadt und den Wissenschaftlerinnen und Wissenschaftlern anderer Universitäten zur Verfügung. Wissenschaftliches Rechnen ist gestattet, sofern die eingesetzte Software dies erlaubt. Jegliche rein kommerzielle Nutzung ist untersagt. Bestandteil dieser Nutzungsordnung sind die Bestimmungen in der [Allgemeinen Benutzungsordnung für die Informationsverarbeitungs- und Kommunikations-Infrastruktur](#) (1) der TU Darmstadt.

Ein Verstoß gegen diese Nutzungsordnung kann zum Entzug der Nutzungsberechtigung führen.

Alle ausgefüllten Nutzungsanträge (dieses Formular) müssen vom Veranstalter gemeinsam mit dem gesonderten Nutzungsantrag zur Lehrveranstaltung beim HRZ eingereicht werden. Für Fragen stehen wir Ihnen gerne auch per E-Mail: hhlr@tu-darmstadt.de zur Verfügung.

2. Aufgaben des Hochschulrechenzentrums der TU Darmstadt

2.a. Betrieb des Hochleistungsrechners

Der Hochleistungsrechner wird vom Hochschulrechenzentrum (HRZ) der Technischen Universität Darmstadt betrieben. Zu den Aufgaben gehören die Zuteilung von Rechenzeit, Hilfestellung bei der Softwareinstallation und das Einrichten von Nutzerkonten.

Die Zuteilung der Rechenzeit erfolgt nach festgelegten Regeln (siehe Abschnitt 3.c) ohne Ausgleichsanspruch. Für den Fall, dass Rechenzeit entfällt (z.B. wegen einer Systemwartung) oder durch Abbruch eines Rechenjobs auf Grund eines Fehlers, gilt es als „Wartungszeit“.

2.b. Speicherung von Login-Daten

Das HRZ speichert ausschließlich Login-Daten der Nutzer/-innen. Diese sind: a) Personen- und Projekt-Informationen (Name, Matrikelnummer, E-Mail-Adresse, etc.). Diese Daten dienen allein der internen Verwaltung und sind für die Zwecke des § 13 Abs. 5 DSGVO zu beachten.

2.c. Speicherung von Arbeitsdaten *Bitte BETA-Hinweis unten beachten!

Ein Ziel ist ausschließlich die Weiterbildung im Rahmen der unter 5 angegebenen Veranstaltung. Nutzer (z.B. persönliche Bilder oder E-Mails etc.) auf der Infrastruktur des HPC-Clusters unter der Adresse [home_na<TU-ID>](#) wird für Accounts im Rahmen der Lehrveranstaltung nicht gespeichert.

Sollte der/die Nutzer/-in bereits ein reguläres Nutzer-Konto (z.B. im Rahmen der Lehrveranstaltung) eingerichtet haben, so wird das bestehende Konto genutzt. Für die hier beantragte Veranstaltung wurde ein gesondertes Konto eingerichtet.

2.d. Ende der Nutzungsberechtigung

Beim Auslaufen oder beim Entzug der Nutzungsberechtigung werden die Nutzer-Daten ([home_na<TU-ID>](#)) des Antragstellers/der Antragstellerin gelöscht.

3. Aufgaben der Nutzer/-innen

3.a. Allgemeine Nutzungsbedingungen

Das HPC-Cluster des Hochschulrechenzentrums ist ausschließlich für die Nutzung im Rahmen der beantragten Lehrveranstaltung zur Verfügung. Eine anderweitige Nutzung ist untersagt.

3.b. Umgang mit dem Nutzer-Konto

Für die Nutzung des Hochleistungsrechners ist ein Nutzer-Konto erforderlich. Das Nutzerkonto gehört zum ID-System der TU Darmstadt.

Das Passwort setzen Sie selbst. Es muss mindestens 8 Zeichen lang sein und aus Groß- und Kleinschreibung, Ziffern und Sonderzeichen bestehen. Die Angabe des Passworts ist verpflichtend. Die Angabe des Passworts ist verpflichtend. Die Angabe des Passworts ist verpflichtend.

3.c.

Die Nutzer/-innen sind verpflichtet, sich a) unerlaubten Zugang zu Daten anderer Nutzer/-innen zu verschaffen, b) sich die Nutzung des Hochleistungsrechners für Zwecke zu misbrauchen, die dem Zweck der Nutzung widersprechen. Das kann zum Entzug der Nutzungsberechtigung führen.

Die Nutzer/-innen sind verpflichtet, die Nutzungsbedingungen der Lehrveranstaltung zu befolgen. Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzer/-innen sind verpflichtet, die Nutzungsbedingungen der Lehrveranstaltung zu befolgen. Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

Die Nutzungsbedingungen der Lehrveranstaltung sind auf der Website des Hochschulrechenzentrums (2) bekanntgegeben.

s. Antrag auf Nutzung des Hochleistungsrechners im Rahmen einer Lehrveranstaltung

Informationen über den/die Antragsteller/-in

Dieses Formular finden Sie online auch unter „Nutzerantrag für Studierende (Lehrveranstaltung)“ auf unserer Webseite http://www.hhlr.tu-darmstadt.de/hhlr/lichtenberg/zugang/lichtenberg_zugang.de.jsp.

Nachname, Vorname:

E-Mail:

TU-ID (auch Externe):

Universität/wiss.Einrichtung:

Titel der Lehrveranstaltung:

Semester / Jahr:

TU-Darmstadt
Grundlagen der Informatik 3 (GdI 3)
Wintersemester 2013/14

Die maximale Laufzeit der Nutzungsberechtigung richtet sich nach dem Nutzungsantrag für die Lehrveranstaltung (aber max. 6 Monate).

Ich bestätige hiermit die Richtigkeit meiner Angaben und verpflichte mich zur Einhaltung dieser Nutzungsordnung. Ich bin mit der Verarbeitung meiner personenbezogenen Daten (nach Abschnitt 2.b) einverstanden.

Ort, Datum

Unterschrift

(Wird vom HRZ ausgefüllt)

Wurde genehmigt und eingerichtet: ja/nein

Laufzeit des Nutzerkontos: - siehe Hauptformular des/der Lehrenden -

Datum

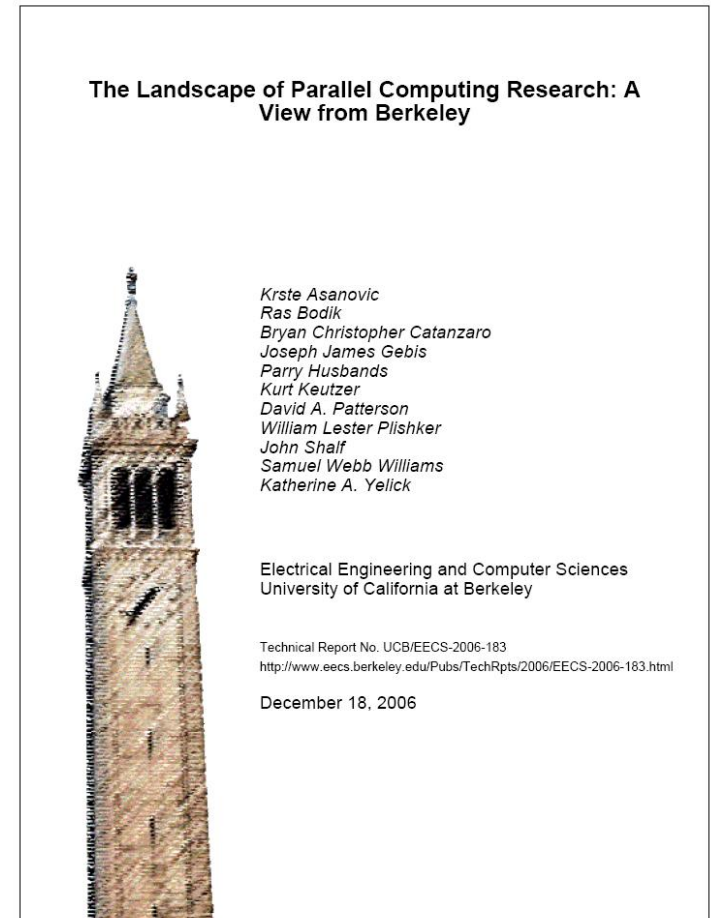
Unterschrift

Anmerkungen:

Informal English translation available in Moodle.

Why PMPP?

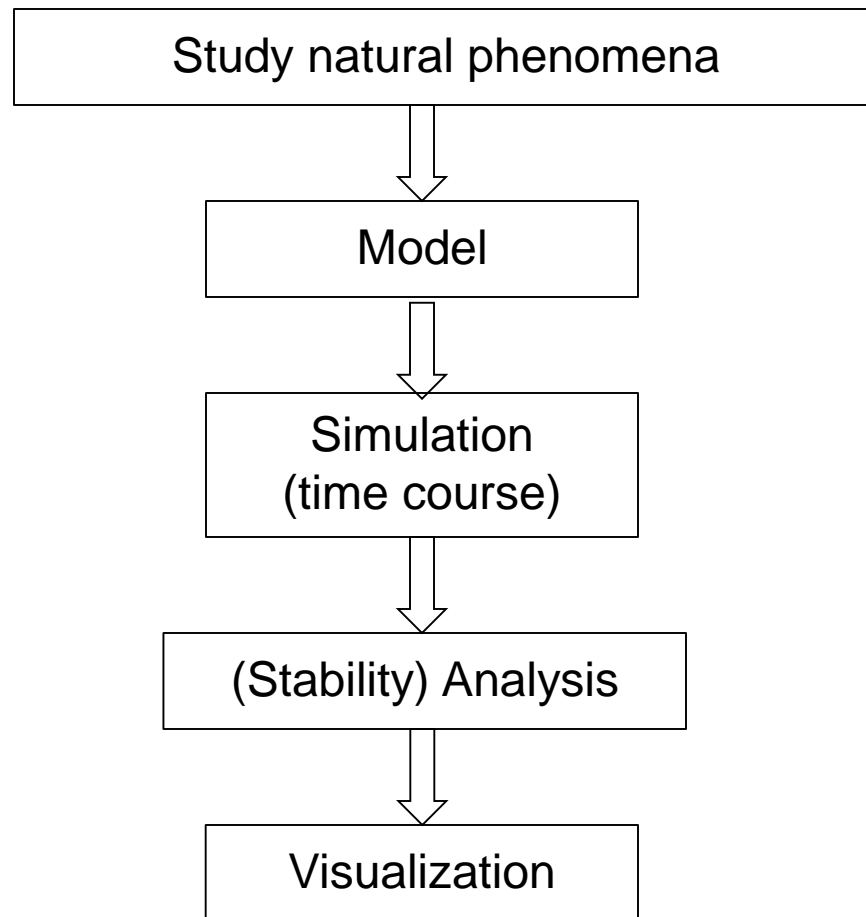
- discussion inspired by
The Landscape of Parallel
Computing Research: A View from
Berkeley
by Asanovic et al., December 18,
2006
Technical Report UCB/EECS-
2006-183
- read Sections 1-3 of the Berkeley
Report
[Asanovic et al. 2006]
 - available online or on the course
webpage
 - you are welcome to read the
whole document



Massively-Parallel Simulation of Biochemical Systems

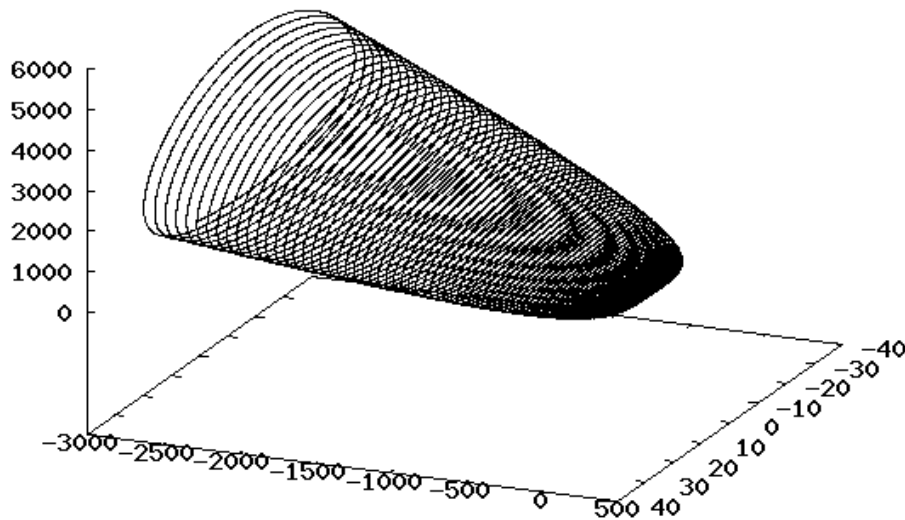
Jens Ackermann Paul Baecher Thorsten Franzel
Michael Goesele Kay Hamacher

published at GI 2009 Workshop on
Massively Parallel Computational Biology on GPUs

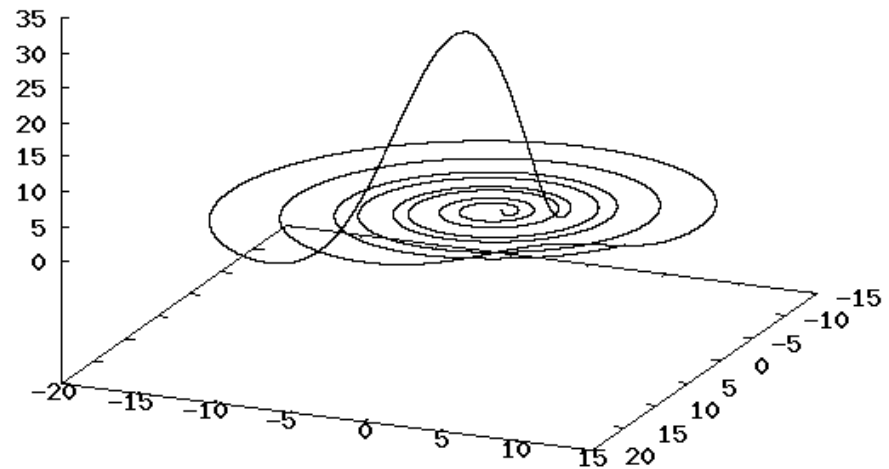


Parameter Dependency

- Behavior of a single system can be highly dependent on the parameters/conditions:

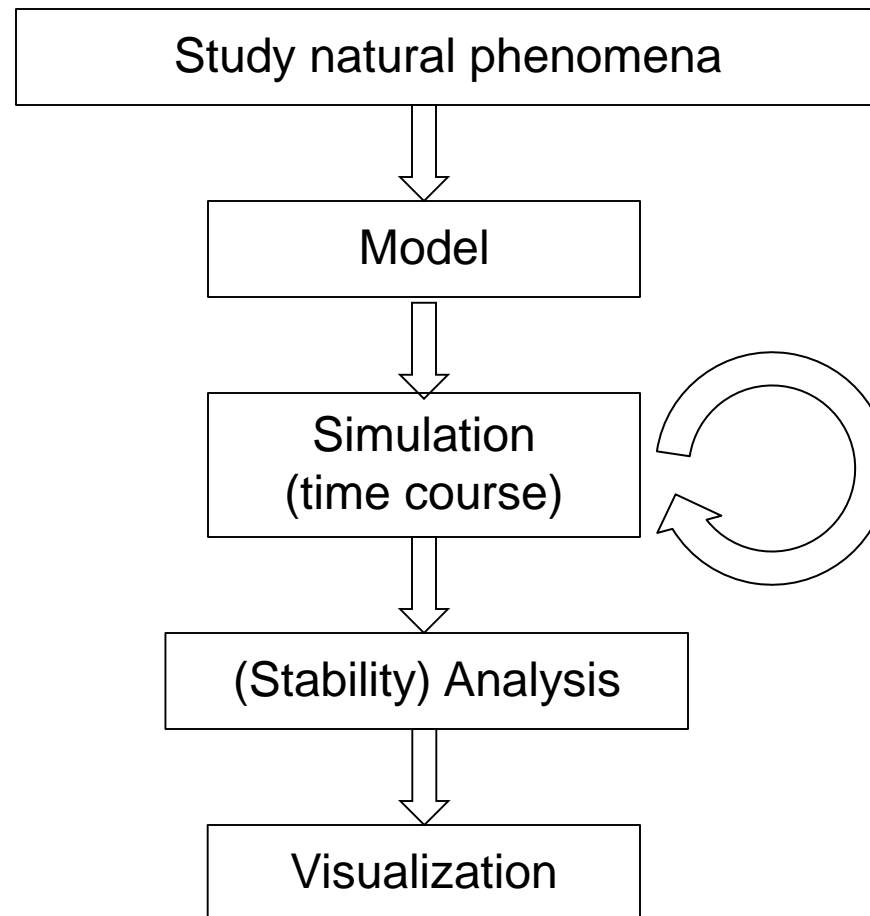


$$(a,b,c) = (0.2, 0.0, 0.0)$$

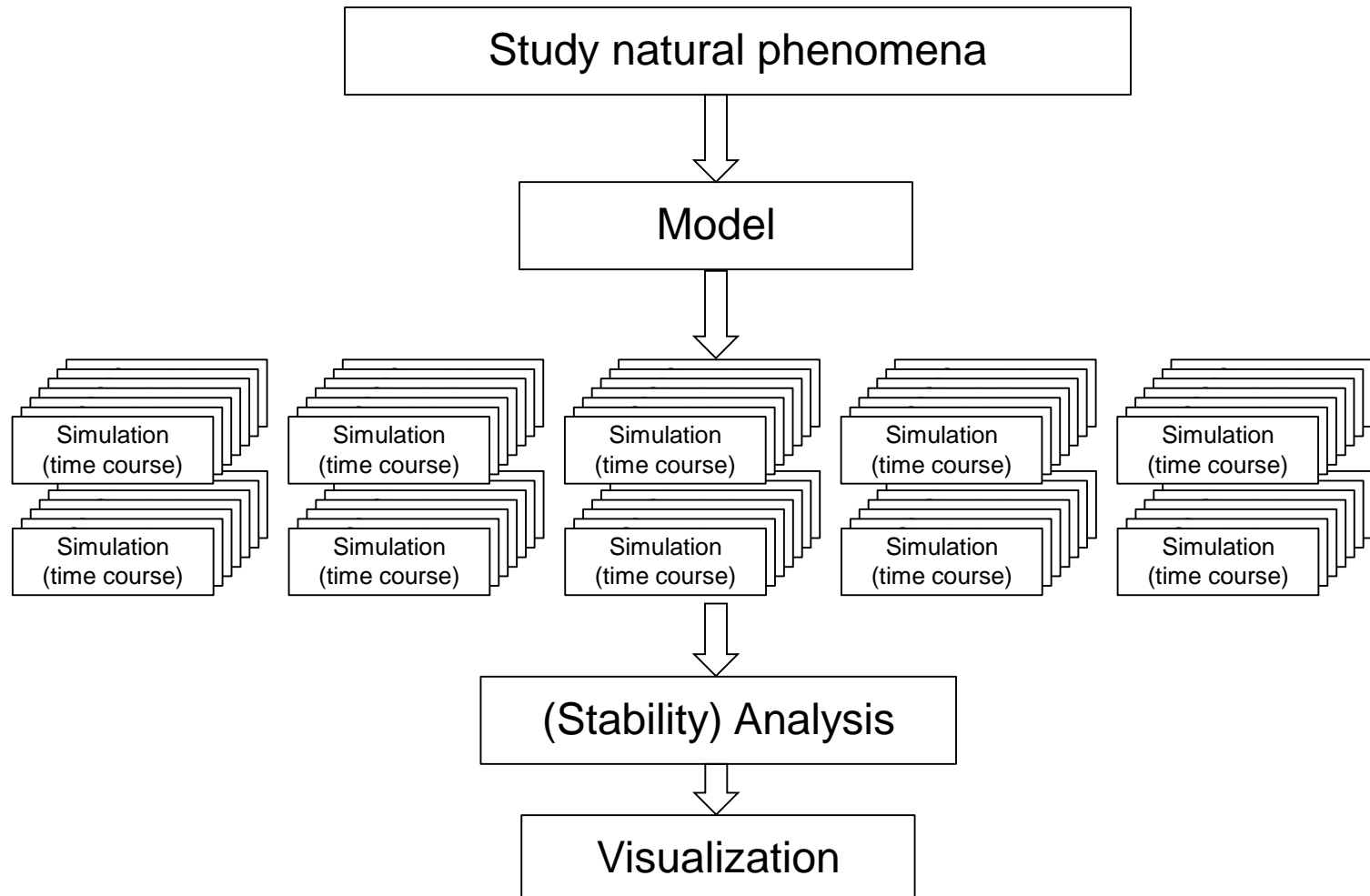


$$(a,b,c) = (0.15, 0.2, 10.0)$$

Multiple Simulations



Multiple Simulations



- Automated pipeline from models in high-level description language to CUDA simulation.
- Considerable speedup for simple model.
 - New possibilities for the analysis.
 - No need for expensive clusters.

Simulation Software Packages

- [SOSlib]: outdated, not massively parallel
- [BioNessie]: large-scale grid computing → HPC clusters are expensive
- [Copasi]: sequential parameter scans, export model to C

„You have to be careful [..], COPASI will run 100 time course simulations during this two-dimensional scan which can take a long time.“

Systems Biology Markup Language (SBML)

- XML based language → machine-readable
- Covers different kinds of models: chemical reactions, metabolic pathways, ...
- There exists a multitude of tools for modeling and analysis.
→ SBML as common ground for data inter-change.
- Databases available.

```
<species id="ATP" compartment="cell"/>
...
<parameter id="temperature" value="0.5"/>
...
<rateRule variable="ATP">
  <math xmlns="...">
    <apply><times/>
      <ci>temperature</ci>
      <ci>ADP</ci>
    </apply>
  </math>
</rateRule>...
```

Biochemical System

Time dependent variables

$$(x_1(t), \dots, x_n(t))$$

Initial values

$$(x_1(t_0), \dots, x_n(t_0))$$

System of (autonomous) ODEs

$$dx_1/dt = f_1(p, x(t))$$

...

$$dx_n/dt = f_n(p, x(t))$$



$$dx/dt = f(p, x(t))$$

Fixed parameters

$$(p_1, \dots, p_m)$$

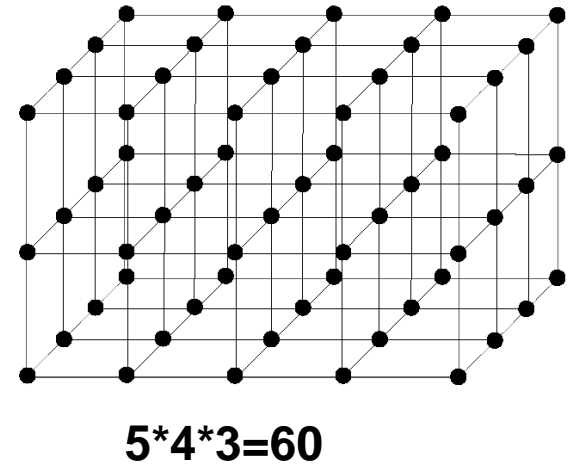
Time

t



Brute-force Parameter Scan

- Sample the m-dimensional space of all possible parameter sets
- For each parameter set p :
 - Evolve ODEs by integrating f .
 - Analyze emerging time series.



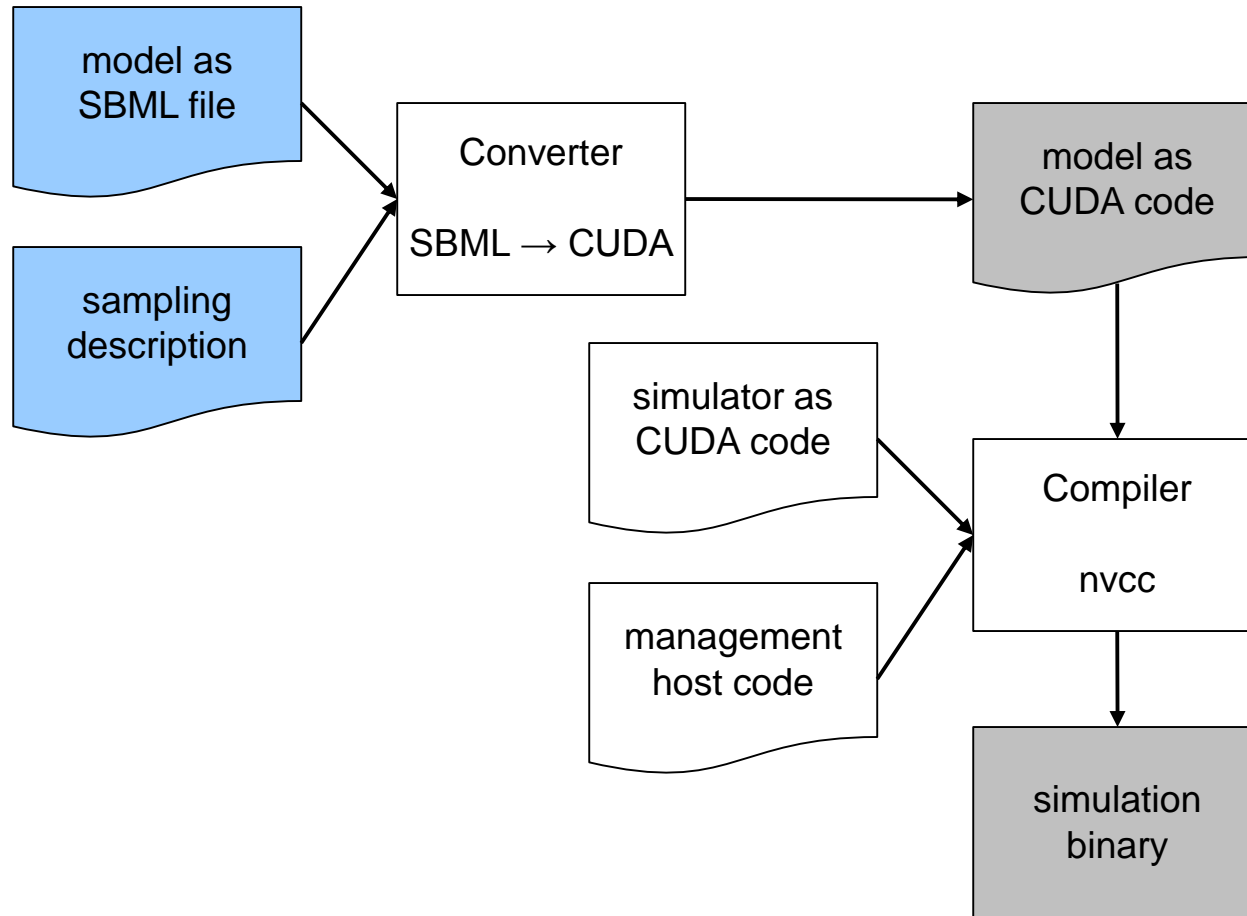
Problem:

Combinatorics → many parameter sets even for small grid sizes.

Solution:

embarrassingly parallel → use CUDA

Our Pipeline



CUDA Implementation Issues

- One thread = one parameter setting.

```
__device__ float f(float *x, uint i) { ... }  
...  
for (uint t = 0; t < T; t++) {                               // time steps  
    for (uint i = 0; i < n; i++) {                             // coordinates  
        x[i] = x[i] + step_size * f(x, i);                   // Euler  
        result[thdOffset + t * n + i] = x[i];  
    }  
}
```

Global
memory
write!

- Euler method needs to evaluate the model function f.
- Lots of data. Storing all time series would slow things down.

Challenges for SBML Conversion

- Function f needed:

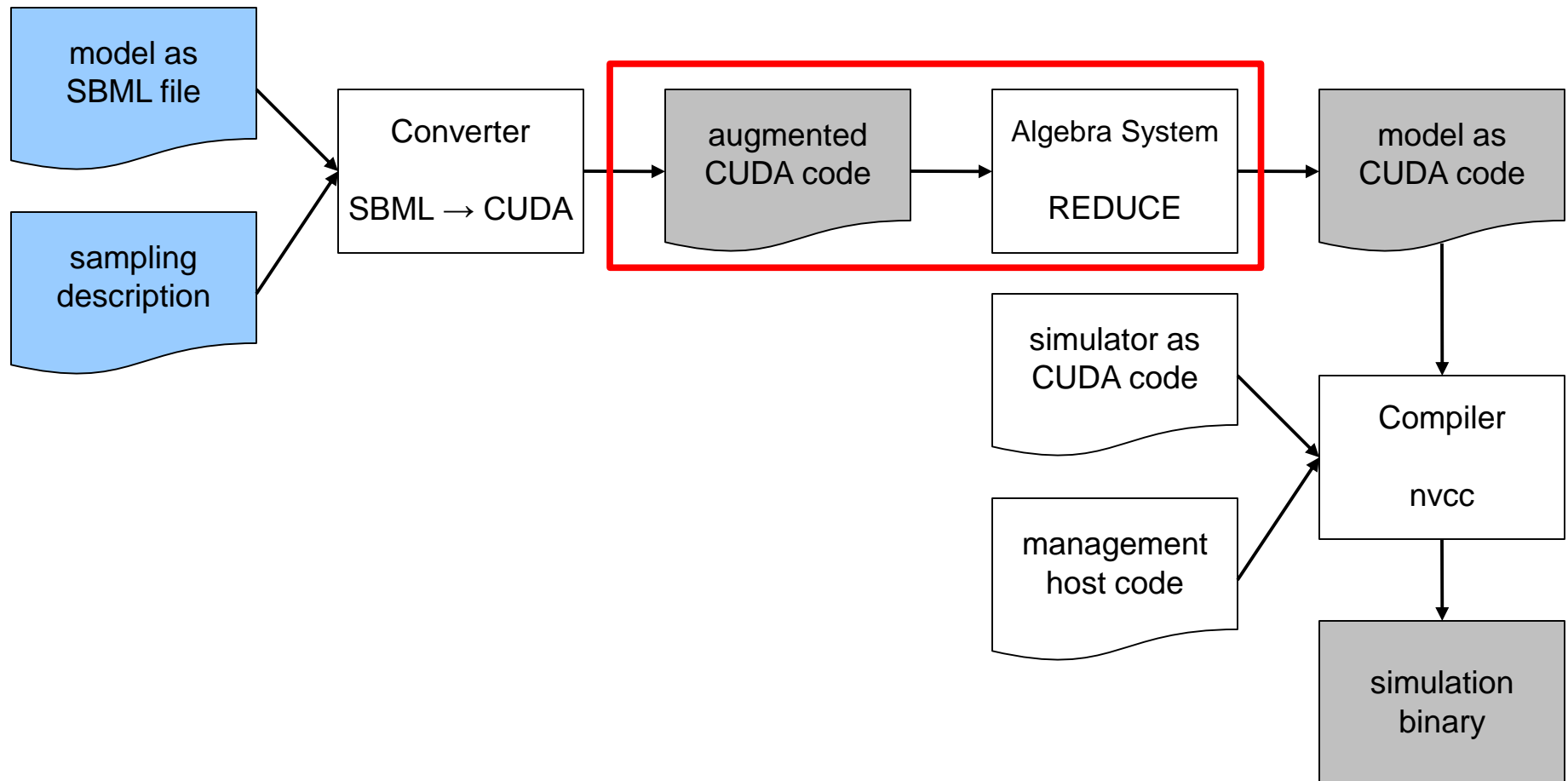
- Extract from SBML model.



- Convert f to CUDA code and compile into simulation binary.
- Too much data:
 - Store only characteristics, e.g., Lyapunov numbers.
 - Computation needs Jacobian of f .
 - Computer Algebra System to automatically differentiate f .

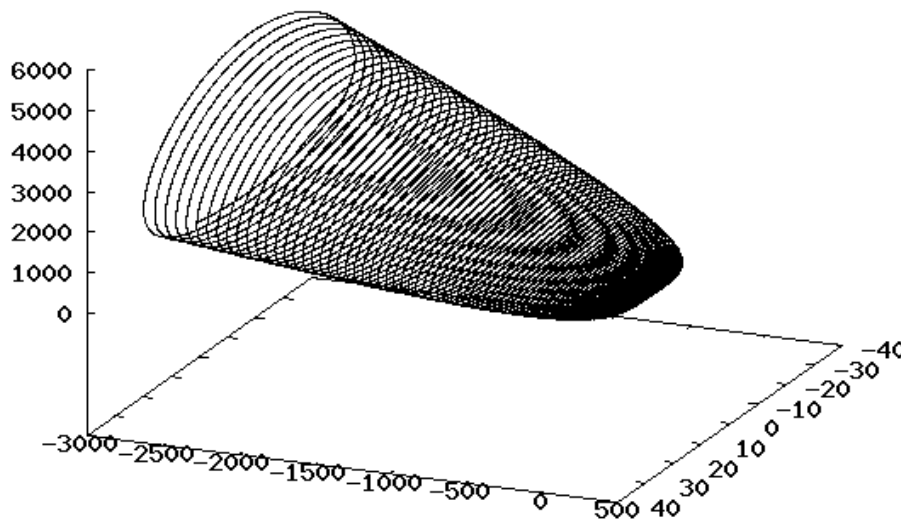


Our Pipeline (final)

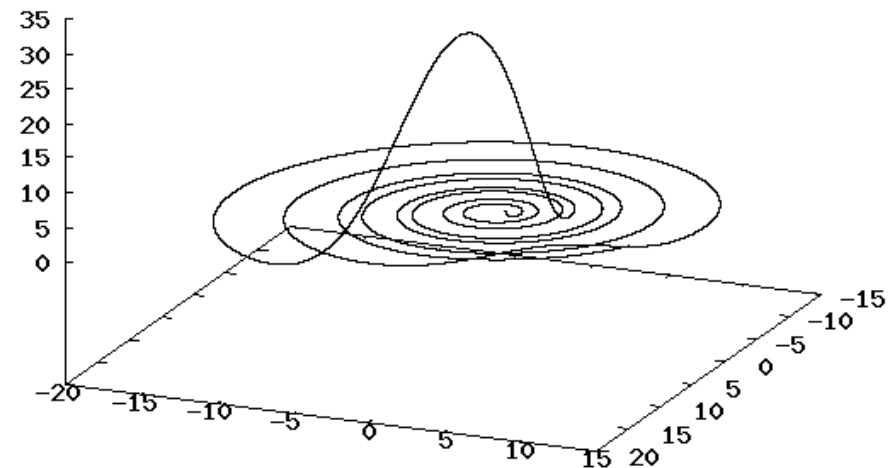


Proof of Concept: Roessler Attractor

- [Roessler, 1976], [Samardzija, 1989]
- 3 variables, 3 parameters.
- SBML model is "human readable".



$$(a,b,c) = (0.2, 0.0, 0.0)$$



$$(a,b,c) = (0.15, 0.2, 10.0)$$

Results

- 3 variables, 3 parameters, 10^6 parameter sets:

GPU/CPU	#MPs	threads/block	Time [10 ³ s]
9800 GX2	1x16	256	12.22
9800 GX2	2x16	256	6.13
9800 GX2	2x16	320	5.55
Xeon CPU	-	serialized	~280

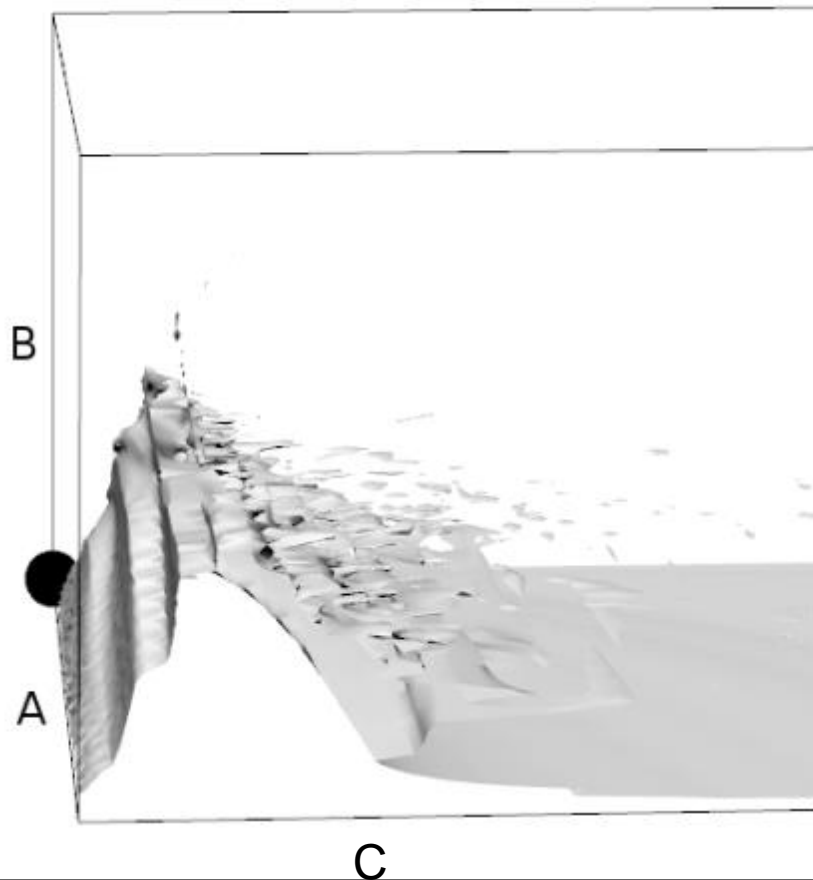
 Speedup ~ 47

Preliminary results.!

19 variables, 48 parameters  Speedup 27

Possible Visualization

- Isosurface extraction in the parameter space:



Conclusion

- Parameter scanning is important for understanding complex system behavior.
 - Ideally suited for parallel computation.
 - Potentially high performance gain if it fits to the architecture.
- Automatic SBML to CUDA conversion connects modeling tools to fast, parallel simulations.



Corresponding Publication

Google

Scholar



Bearbeiten



Exportieren ▾



Michael Goesele

Massively-Parallel Simulation of Biochemical Systems.

[\[PDF\] von emis.de](#)

Autoren Jens Ackermann, Paul Baecher, Thorsten Franzel, Michael Goesele, Kay Hamacher

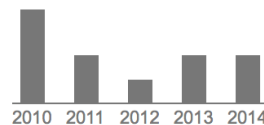
Publikationsdatum 2009/9/29

Konferenz GI Jahrestagung

Seiten 739-750

Beschreibung Abstract: Understanding biological evolution prompts for a detailed understanding of the realized phenotype. Biochemical and gene regulatory dynamics are a cornerstone for the physiology of the cell and must therefore be regarded as one of the major aspects of such a phenotype. Experimental insight into molecular parameters is, however, hard to come by. Model development therefore requires computational parameter estimation. At the same time, design of cellular dynamics is highly efficient when done in-silico. We therefore ...

Zitate insgesamt **Zitiert von: 11**



Google Scholar-
Artikel [Massively-Parallel Simulation of Biochemical Systems.](#)
J Ackermann, P Baecher, T Franzel, M Goesele... - GI Jahrestagung, 2009
[Zitiert von: 11](#) - [Ähnliche Artikel](#) - [Alle 7 Versionen](#)

