2012 Basic Numerical Analysis

Course Guidance

Intended Audience

- Grad. students who do *not* major in computational physics
- Those who are not familiar with scientific computing/programming
- Those who'd like to be able to do numerical integration, matrix inversion etc.
- Those who do *not* dislike programing

Course plan

Part 1 (Oct. 2 - Nov. 13, taught by Yoshida)

Overview, programming basics,
ODEs, PDEs, multi-dimension

Part 2 (Nov. 20 - Jan. 15, by Prof. Shimizu)

matrix operation、eigen-value
problems、Monte-Carlo method,
parallel computing

Important!

Unit and grade:

A problem set is given in each part,
i.e., 2 reports, 50 points each.

Grade given out of 50 + 50 = 100 points.

Lecture 1

- Large-scale computer simulations
- Programming basics
- Example programs
- Root finding

Useful textbooks

シリーズ現代の天文学 14巻
「シミュレーション天文学」 日本評論社
富阪幸治 他編

「パソコンによるシミュレーション物理」
朝倉書店 矢部孝 他著

Numerical Recipes

Press, Teukolsky, Vetterling, Flanerry 著

Numerical simulations

- Complex phenomena in 2D/3D
  fluid, material, quantum mechanics, bio
- Compare theoretical models with experiments/observations
- Visualize what cannot be easily seen

K computer of RIKEN

計算ノードの数 82944

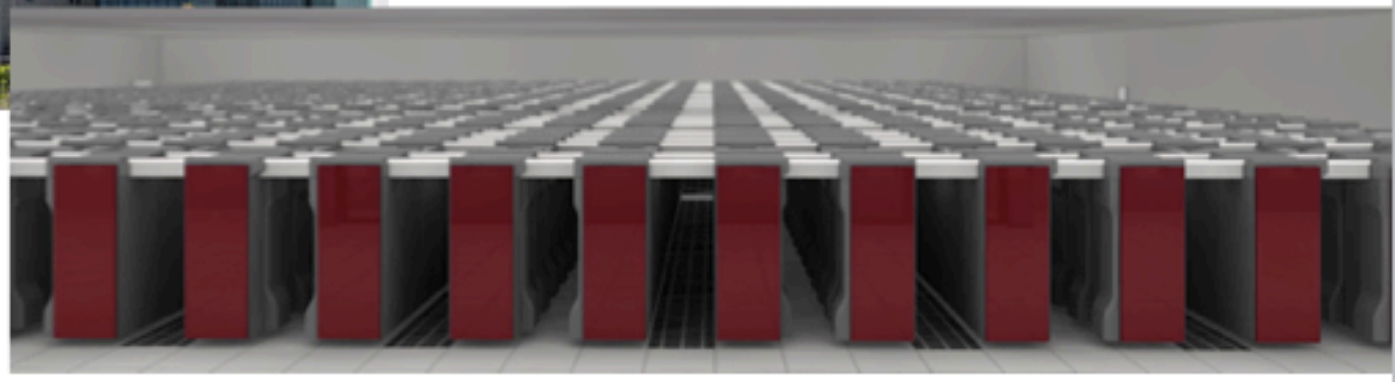
コアの数 > 640,000

ピーク演算速度　11.3 Pflops

メモリ > 1Pbytes (16GB per node)

ネットワークデータ転送 5GB/sec (双方).

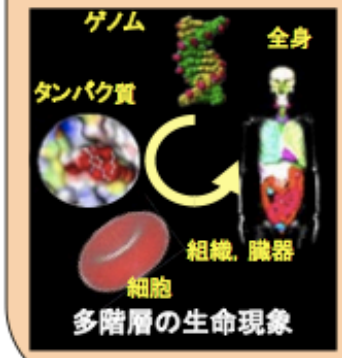
6次元トーラス構造



「京」の戦略分野

予測する生命科学・医療および創薬基盤

新薬の開発

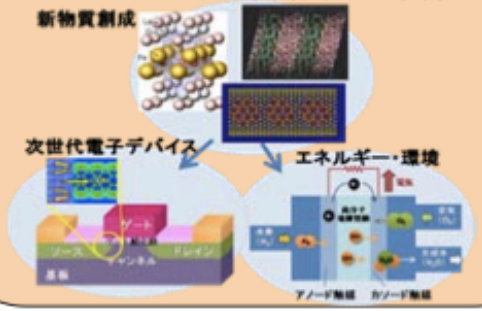


生体分子から細胞、臓器、全身にわたる多階層の生命現象を予測し、副作用のない革新的な医薬品が開発できる。

新物質・エネルギー創成

新デバイスとエネルギーの開発

新物質・新現象の探索を基盤とし、次世代電子デバイス開発の指針を与え、クリーンエネルギーの生成の高効率化に資する。



防災・減災に資する地球変動予測

台風の進路や集中豪雨の予測

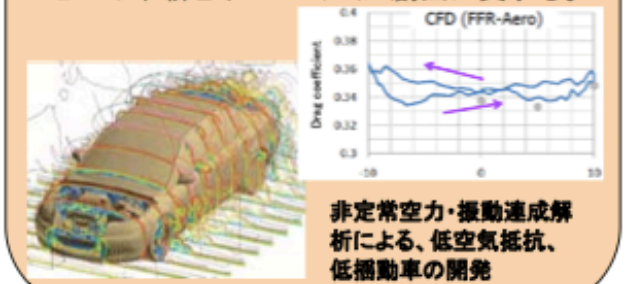
NICAMによる全球3.5kmシミュレーション

全球雲解像モデルにより台風の進路や集中豪雨の高精度予測が可能となり、効果的な防災・減災対策に資する。

次世代ものづくり

設計プロセスの革新

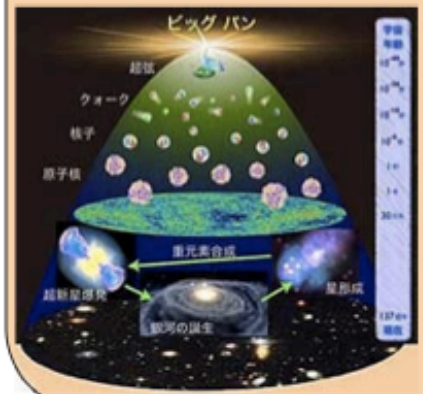
独創的要素技術の創造、組合せ最適化、丸ごと性能評価を可能とし、ものづくりプロセスの革新とイノベーション創出に資する。



非定常空力・振動連成解析による、低空気抵抗、低振動車の開発

物質と宇宙の起源と構造

物質の起源と宇宙の構造形成

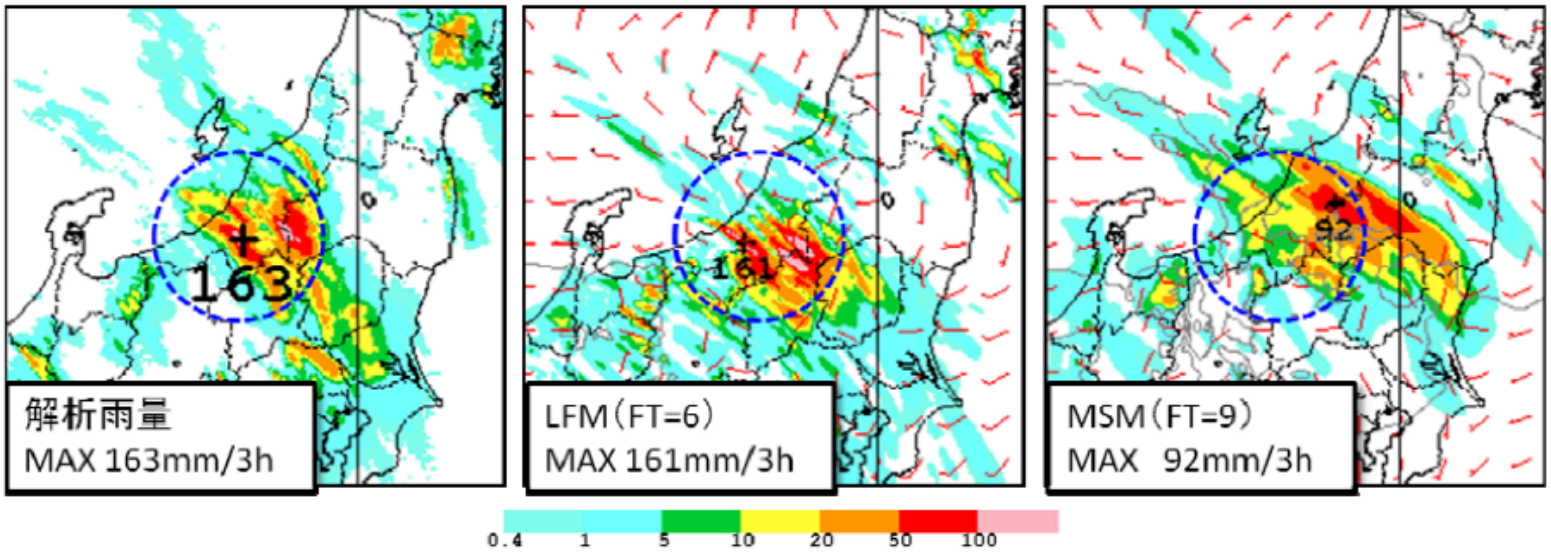


ビッグバンに始まる宇宙において、極微の素粒子から元素合成、そして星・銀河形成に至る物質と宇宙の起源と構造を統一的に解明する。

Large-scale simulations

Weather forecast

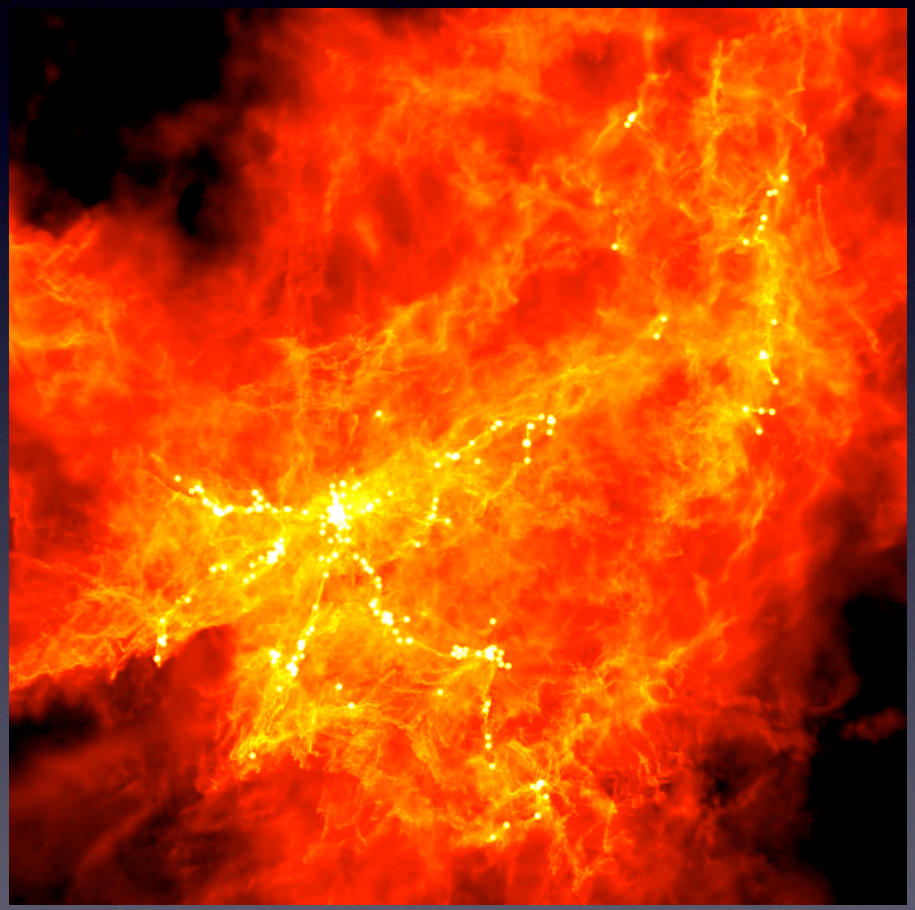
2011 Fukushima-Niigata storm



2km per cell resolution

(data/measurements are needed at this resolution,)

Astrophysics

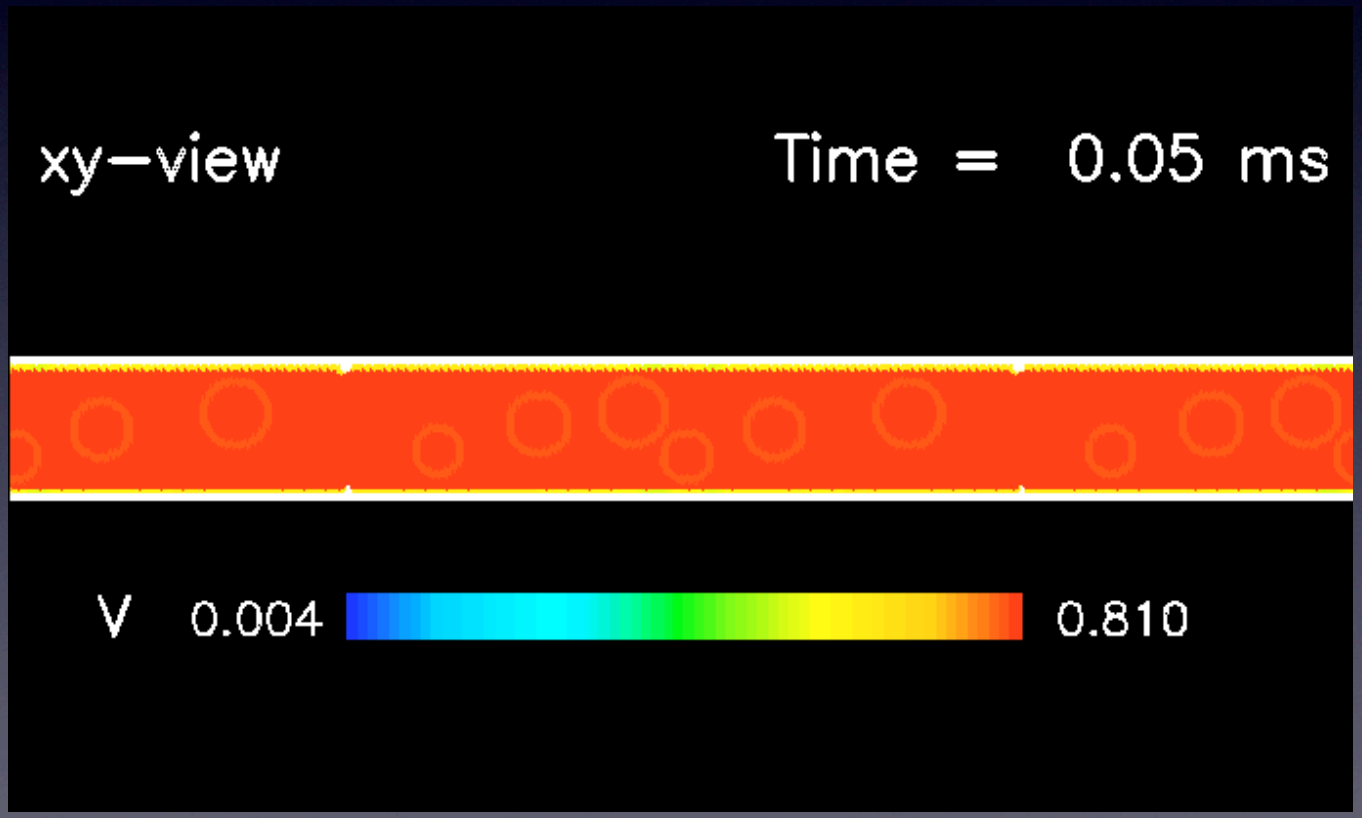


Star formaion in a

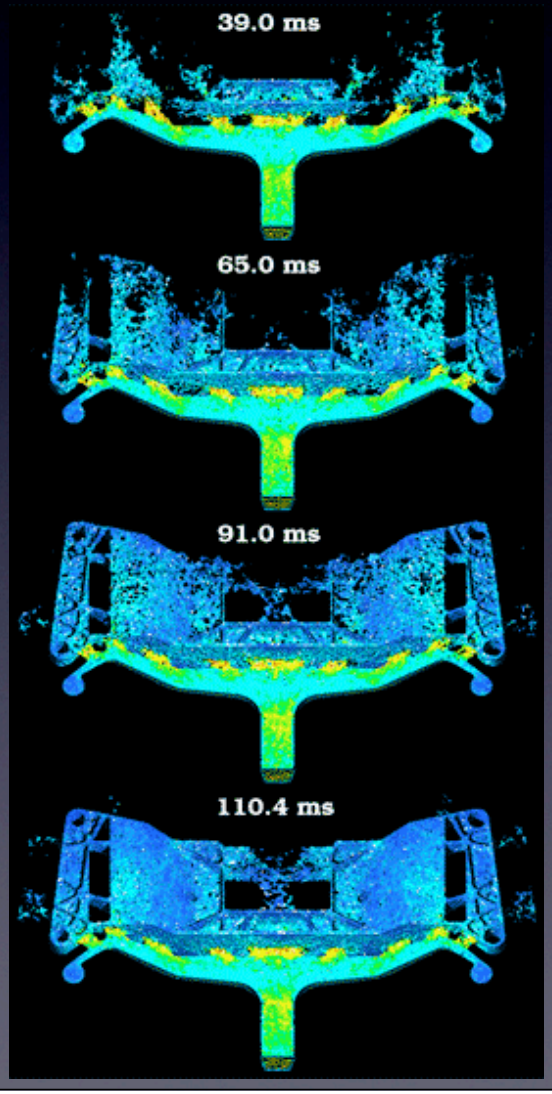
turbulen gas cloud

Fluid mechanics

Blood flow



Engineering

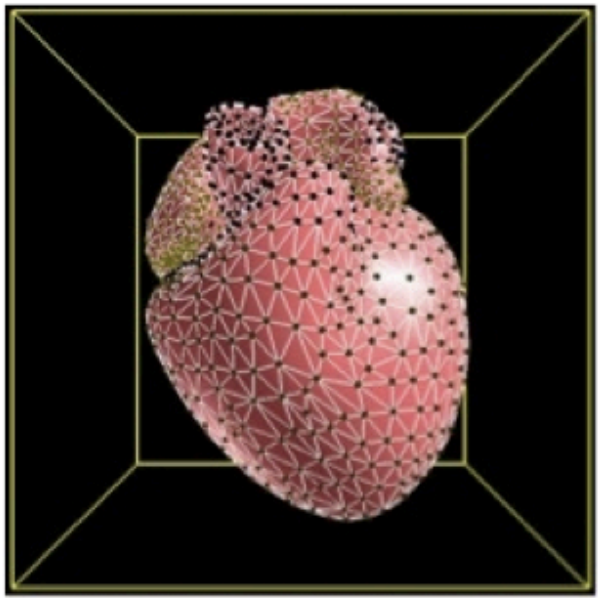


Viscous fluid flow into

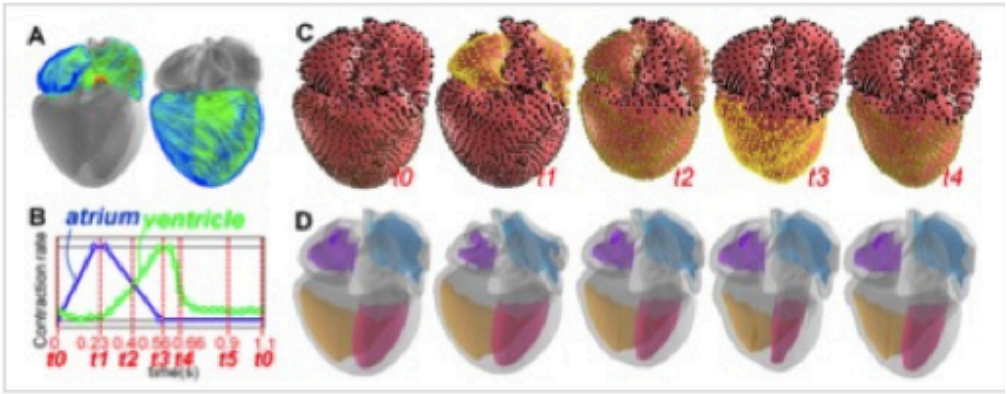
a mold. A particle simulation.

Medical sciences

心臓の運動と血液循環



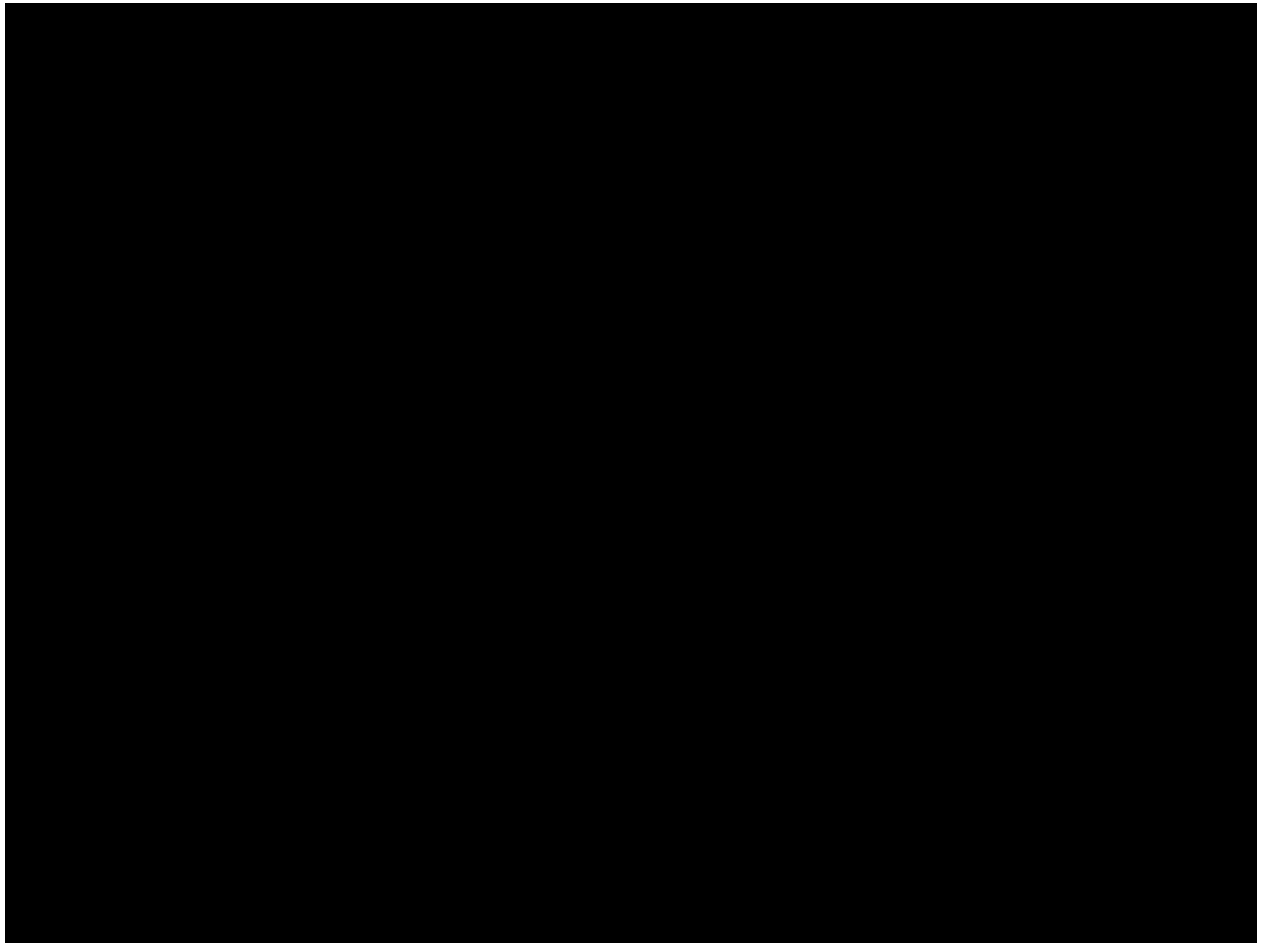
心臓を7000個の部分に分け、
心筋細胞の働きを再現



国立循環器病研究センター、理化学研究所、東大、滋賀医科大

Pandemic

ロスアンゼルスで１０人のH5N1型感染者発生



休校などの措置をとらなかった場合、およそ90日で9000万人が感染

In this class, “simulation” refers to, mostly, finding numerical solution(s) for a system that is described by well-established governing equations (such as Navier-Stokes, Newton’s eq.)

There are other types of “simulations” used in TV news, media etc which are not necessarily the outcomes of scientific computing.

Programming in C

Suppose you'd like to

Find a root of $3x^3 + 4x^2 - 6x + 1 = 0$,

Intgerate $r^2 \log r$ over [0.1, 1.3],

or advance $\partial_t f + \partial_x f = 0$ in time.

“Solutions”

- Forget about it. Find another prom.
- Read a textbook
- Use a software (e.g. mathematica)
- Write your own program

  The last one is what this course is meant for.

Basic structure

```
#include <stdio.h>

#include <math.h>

void factorial();

int main()

{

    double a, b, c, d;

    a = 3.24;  b=1.59;

    c = sqrt(b * b - 4.0 * a);

    return 0;

}
```

基本 Cプログラム

```
#include <stdio.h>

#include <math.h>

void factorial();

int main()

{

    double a, b, c, d;

    a = 3.24;  b=1.59;

    c = sqrt(b * b - 4.0 * a);

    return 0;

}
```

Library headers

function declaration

main part

variable declaration

operations

Variables

In C, one can write

a, x, c14, volume, electron_mass, TEMP

as variables.

Their types are

整数 int,

実数(浮動小数点数) float, double

文字列 char

Data type

integer constants　0　2　3245　76892

floating point numbers

0.01　3.2　4.678e8

characters（strings)

“world”　“Tokyo 113”　“080-3921”

Variable declaration

C言語では変数を用いる前に宣言しなくてはならない

```
int main()
{
   int  i, j, k;
   double a, b, c, d;
   i = 3;
   a = 2.3;  b=1.59; c=0.05;
   c = sqrt(b * b - 4.0 * a - i * c);
    return 0;
```

ここ

変数宣言

しかし、必ずしもプログラム先頭部でなくとも、途中で（ただし使用前に）宣言してもよい。

```
int main()
{
   double a, b, c, d;
   a = 2.3;  b=1.59; c=0.05;
   c = sqrt(b * b - 4.0 * a);
   d = integrate_fx();

   double factor = 3.4;     ここ
   a = factor * d;
```

式と文 (expression, statement)

expressions

```
a + b
x = y
a = b + c
x <= y
```

など

statements

```
a = 6;
a = b + c;
++j;
{
  pi = 3.14;
  area = pi * r * r;
}
```

など

四則演算

+, -, *, /

C言語には指数演算子が無く、
math libraryとして pow が
定義されている。たとえば
２の４乗は pow(2, 4)

演算の優先度

a - b/c + d　は a - (b/c) + d
のように処理される。

increment と decrement

b = a++;　は
b = a; a = a+1;　と同じ。
b = ++a;　は
a = a+1; b=a;　と同じ。

比較演算子

less than	<
less than or equal to	<=
greater than	>
greater than or equal to	>=

等号演算子

equal to	==
not equal to	!=

代入演算子

+=, -=, *=, /=

i += 6 は　i = i + 6 と同じ。他も同様。

条件演算子

なになに？〇〇：□□

例　(j < 5)？ 12 : -6

j が５より小さい（ j<5 が真）とき12を返し、
j が５より大きい (j < 5 が偽)とき　-6を返す。
クイズ)　j=5 のときは？

応用　k = (i < 0) ? n : m

関数ライブラリ

#include <math.h>

を文頭におくと、以下の関数を使える

acos(d), asin(d), atan(d)

cbrt(d), sqrt(d)

cos(d), sin(d),

cosh(d),sinh(d),tanh(d),

exp(d), log(d), log10(d), pow(d1,d2)

また、いくつかの定数も定義されている:

M_PI, M_SQRT2, M_E など

関数の定義と呼び出し

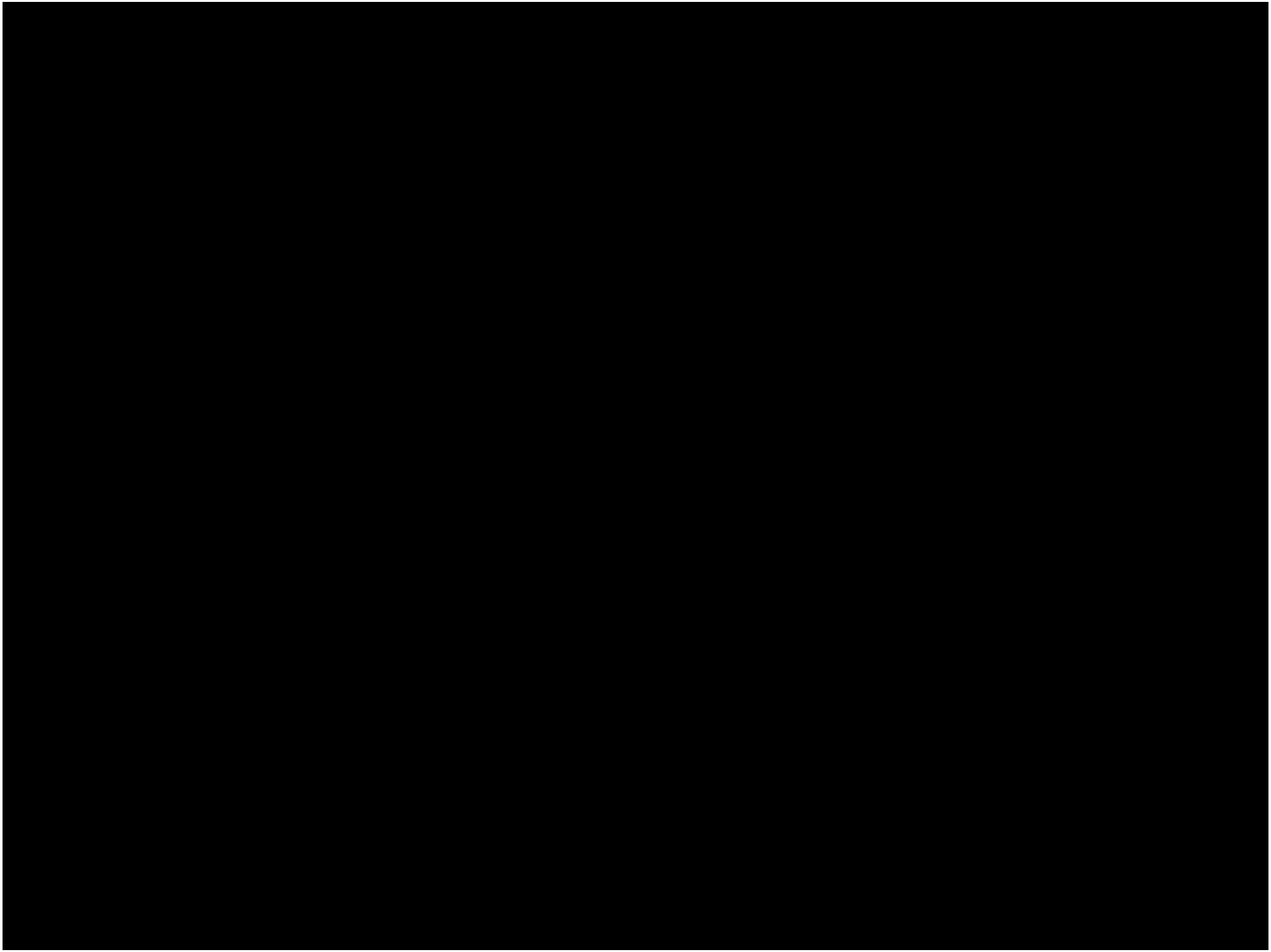
関数を使うには、宣言と定義が必要

```
double factorial(int n);

int main()

{

   int  i;   double fac;

   for(i=0; i <10; i++)

        fac = factorial(j);

        cout<<fac<<endl;
```

関数の定義と呼び出し

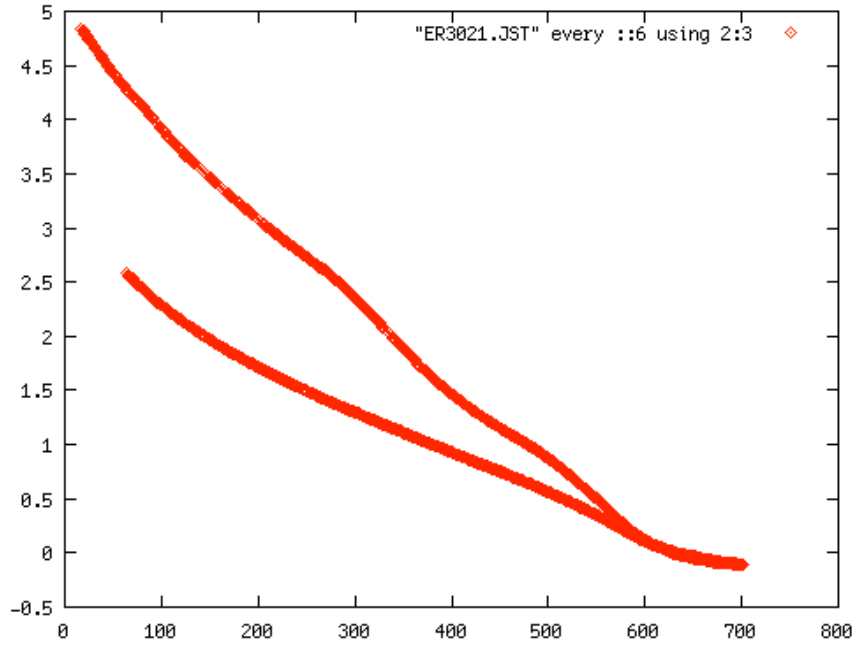
関数を使うには、宣言と定義が必要

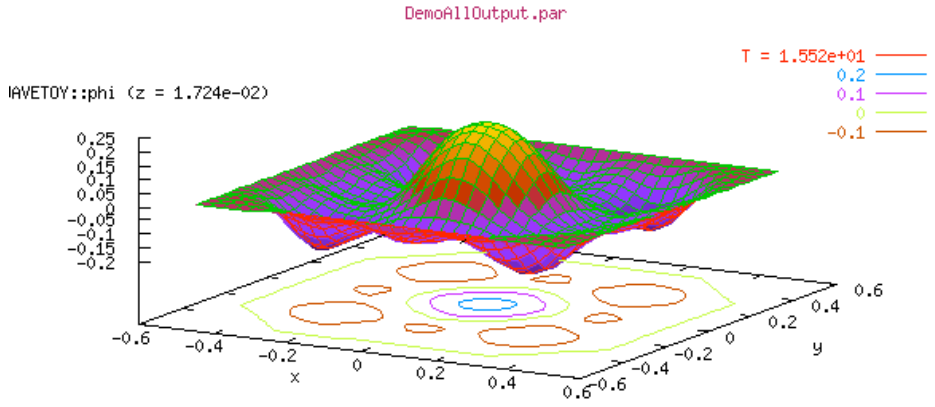
```
double factorial(int n);

int main()

{

   int  i;   double fac;

   for(i=0; i <10; i++)

        fac = factorial(j);

        cout<<fac<<endl;
```



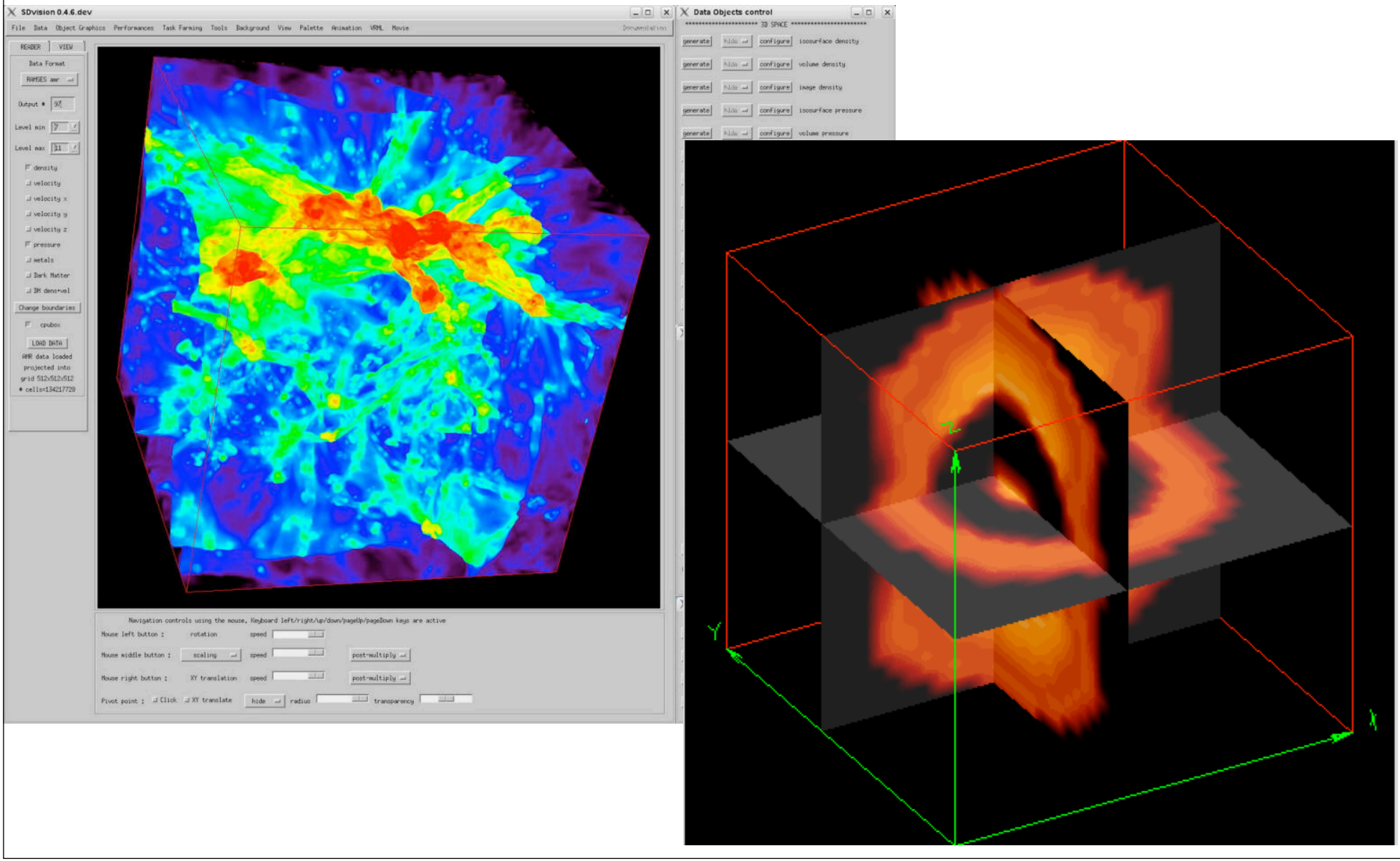
可視化の初歩

グラフツール GNUPLOT





もう少し本格的


SDvision 0.4.6.dev
File Data Object Graphics Performances Task Farming Tools Background View Palette Animation VRML Movie
READER VIEW
Data Format
Output #
Level min
Level max
density
velocity
velocity x
velocity y
velocity z
pressure
metals
Dark Matter
IM dens+vel
Change boundaries
cpubox
LOAD DATA
AMR data loaded
projected into
grid 512x512x512
cells=134217728
Navigation controls using the mouse. Keyboard left/right/up/down/pageUp/pageDown keys are active
Mouse left button : rotation speed
Mouse middle button : scaling speed post-multiply
Mouse right button : XY translation speed post-multiply
Pivot point : Click XY translate hide radius transparency
Data Objects control
3D SPACE
generate hide configure isosurface density
generate hide configure volume density
generate hide configure image density
generate hide configure isosurface pressure
generate hide configure volume pressure
Z
Y
X