大纲

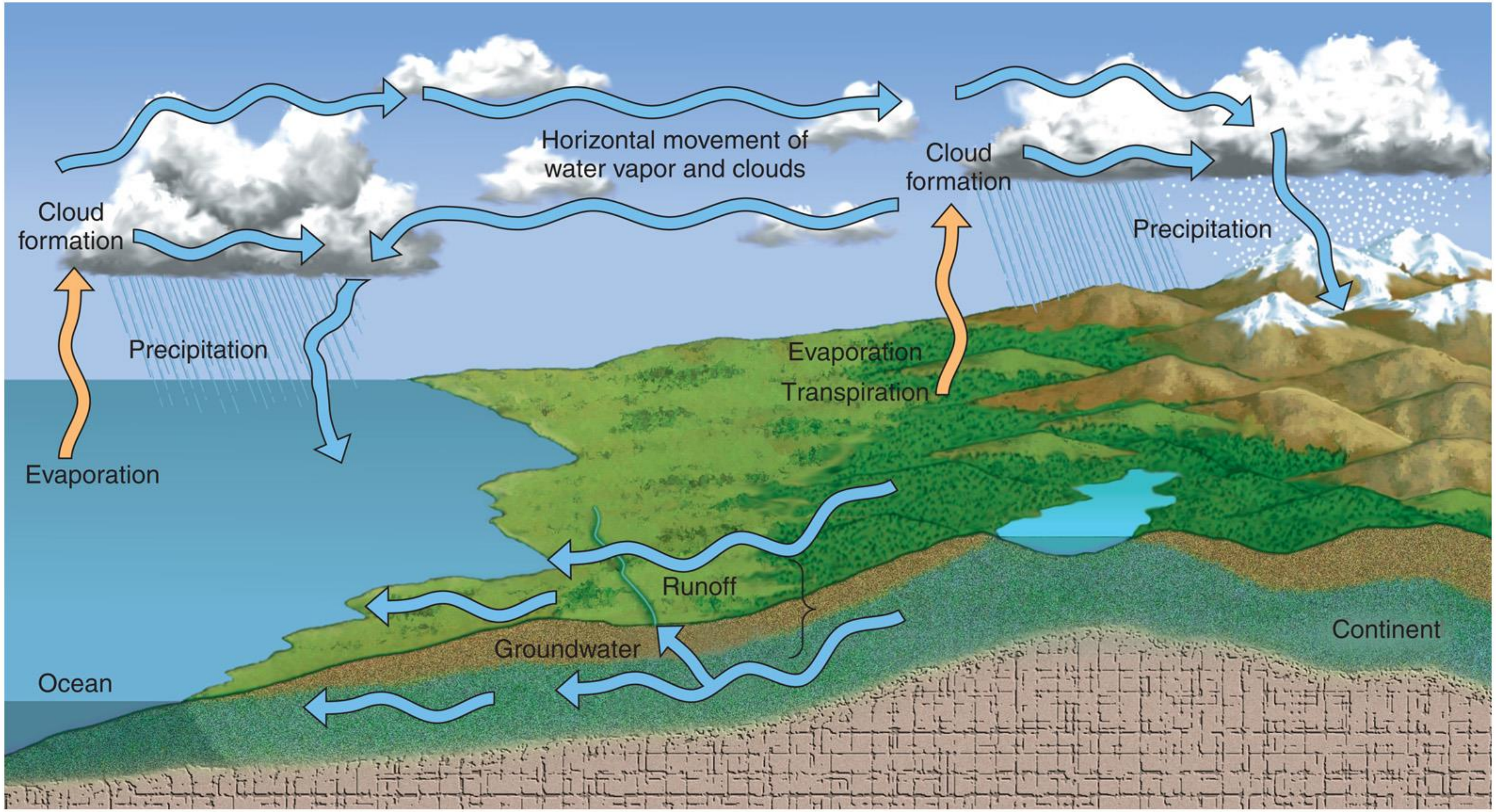
- **前言**
  - 为什么需要“大规模计算” [HPC, DL, Business platform system, Cloud已经合流]
    - 导入 – 科学计算(天气预报)，DL, 互联网平台(Google, Amazon, Alibaba, MeiTuan, …)
- **基础篇**
  - 并发程序的样子 – Divide & Conquer, Model & Challenges, PCAM, Data/Task, …
    - 天气预报的计算
  - 运行环境
    - 硬件 – 自己梳理的3个方案 – Shared/Unshared Memory, Hybrid
    - 系统软件 – 协议栈, Modern OS, Distributed Job Scheduler, GTM等
- **算法级篇**
  - OpenMP, MPI, CUDA (DL的实现), Big Data 中的MR/Spark等 (只涉及在Big Data SDK之上的编程；大数据本身的介绍放到后一部分)
- **系统级篇 – 互联网平台的实现**
  - “秒杀”的技术架构
  - 计算广告
  - 系统架构 (HTAP等)
    - Flink, ClickHouse, MaxCompute, ELK …

Chapter 2: HPC with an example

□ Faster for larger data

- Many problems/applications need HPComputers
  - **Weather / Climate**， Cryptography，Nuclear Weapons Design，Scientific Simulation，Petroleum Exploration，Aerospace Design，Automotive Design，Pharmaceutical Design，Data Mining，Data Assimilation
- Heat dynamics as an example to understand Numeric Computing/Scientific Computing
  - Like Finite Element Analysis (有限元分析)
- Other examples

The Hydrologic Cycle



© 2010 Pearson Education, Inc.

□ The Father of Modern Meteorology

NORGE
45
VILHELM
BJERKNES
1862
1962

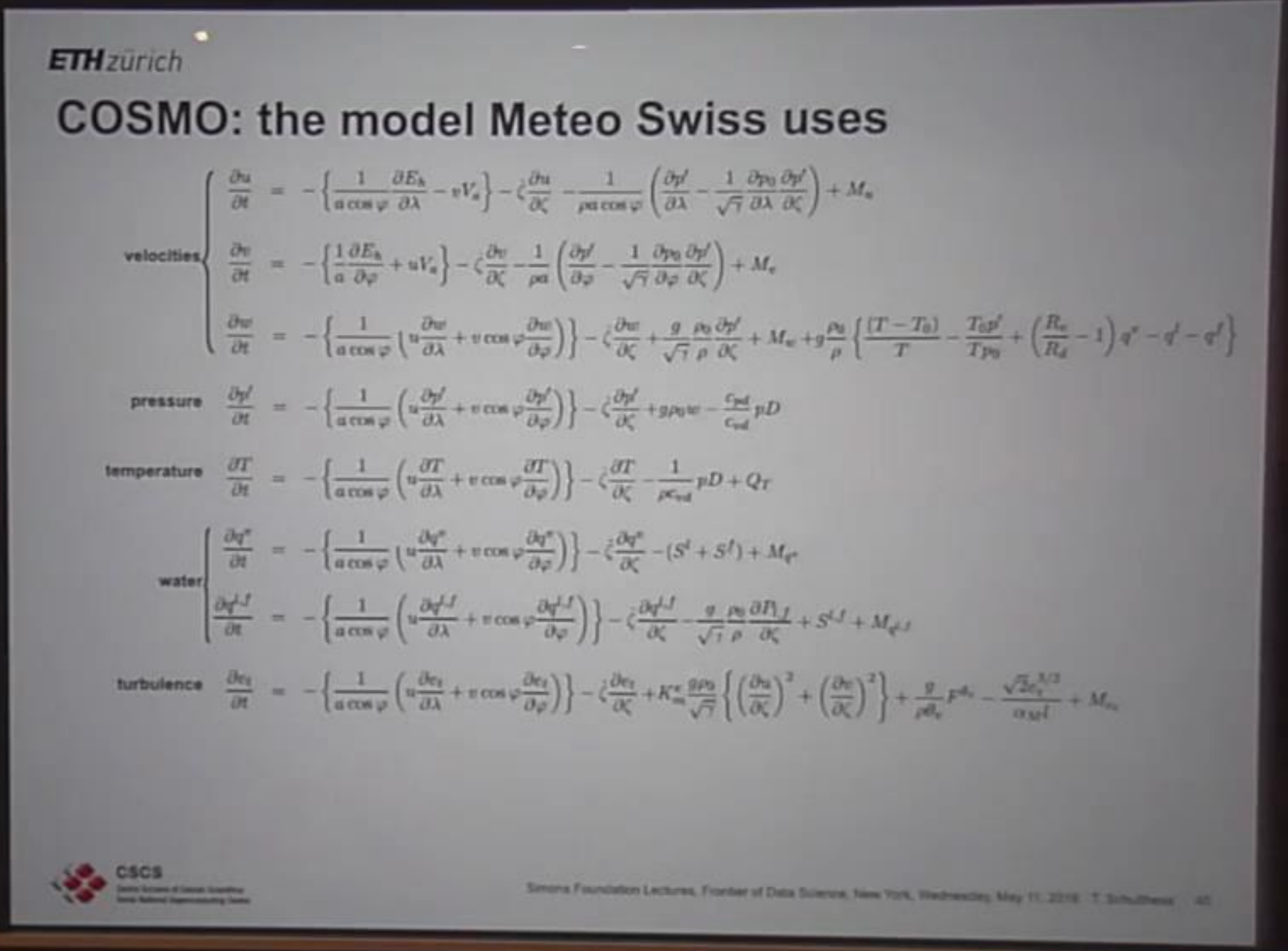
- **Vilhelm Bjerknes' Vision**
  - 1901 – Wanted to incorporate physics into weather forecasting
    - Start with complete set of initial conditions (3-D)
    - Solve equations using graphical methods
    - Initial state not sufficient for good forecasts
    - Did not use continuity equation to derive the initial vertical wind component (no direct measurements available)

Source: Historical Essays on Meteorology 1919-1995, AMS

No calculus solution!

- **Atmosphere dynamics with considering many parameters:**
  - 3-D: Temperature, Humidity, Wind speed and Wind direction, Atmospheric Pressure,
  - Later even Dew Point, Relative Humidity …

$$
\begin{cases}
\dfrac{du}{dt} - \dfrac{uv\tan\varphi}{r} + \dfrac{uw}{r} = -\dfrac{1}{\rho}\dfrac{\partial p}{r\cos\varphi\partial\lambda} + fv - \tilde{f}w + F_\lambda \\
\dfrac{dv}{dt} + \dfrac{u^2\tan\varphi}{r} + \dfrac{vw}{r} = -\dfrac{1}{\rho}\dfrac{\partial p}{r\partial\varphi} - fu + F_\varphi \\
\dfrac{dw}{dt} - \dfrac{u^2+v^2}{r} = -\dfrac{1}{\rho}\dfrac{\partial p}{\partial r} - g + \tilde{f}u + F_r \\
\dfrac{d\rho}{dt} + \rho\left[\dfrac{1}{r\cos\varphi}\dfrac{\partial u}{\partial\lambda} + \dfrac{1}{r\cos\varphi}\dfrac{\partial(v\cos\varphi)}{\partial\varphi} + \dfrac{\partial(wr^2)}{r^2\partial r}\right] = 0 \\
\dfrac{dT}{dt} - \dfrac{RT}{C_p p}\dfrac{dp}{dt} = \dfrac{\dot{Q}}{C_p} \\
p = \rho RT
\end{cases}
$$



COSMO: the model Meteo Swiss uses

velocities

$$\frac{\partial u}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\frac{\partial E_h}{\partial\lambda} - vV_a\right\} - \dot{\zeta}\frac{\partial u}{\partial\zeta} - \frac{1}{\rho a\cos\varphi}\left(\frac{\partial p'}{\partial\lambda} - \frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial\lambda}\frac{\partial p'}{\partial\zeta}\right) + M_u$$

$$\frac{\partial v}{\partial t} = -\left\{\frac{1}{a}\frac{\partial E_h}{\partial\varphi} + uV_a\right\} - \dot{\zeta}\frac{\partial v}{\partial\zeta} - \frac{1}{\rho a}\left(\frac{\partial p'}{\partial\varphi} - \frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial\varphi}\frac{\partial p'}{\partial\zeta}\right) + M_v$$

$$\frac{\partial w}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial w}{\partial\lambda} + v\cos\varphi\frac{\partial w}{\partial\varphi}\right)\right\} - \dot{\zeta}\frac{\partial w}{\partial\zeta} + \frac{g}{\sqrt{\gamma}}\frac{\rho_0}{\rho}\frac{\partial p'}{\partial\zeta} + M_w + g\frac{\rho_0}{\rho}\left\{\frac{(T-T_0)}{T} - \frac{T_0 p'}{T p_0} + \left(\frac{R_v}{R_d} - 1\right)q^v - q^l - q^f\right\}$$

pressure

$$\frac{\partial p'}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial p'}{\partial\lambda} + v\cos\varphi\frac{\partial p'}{\partial\varphi}\right)\right\} - \dot{\zeta}\frac{\partial p'}{\partial\zeta} + g\rho_0 w - \frac{c_{pd}}{c_{vd}}pD$$

temperature

$$\frac{\partial T}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial T}{\partial\lambda} + v\cos\varphi\frac{\partial T}{\partial\varphi}\right)\right\} - \dot{\zeta}\frac{\partial T}{\partial\zeta} - \frac{1}{\rho c_{vd}}pD + Q_T$$

water

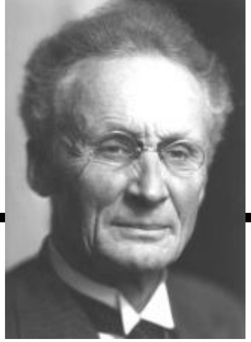
$$\frac{\partial q^v}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial q^v}{\partial\lambda} + v\cos\varphi\frac{\partial q^v}{\partial\varphi}\right)\right\} - \dot{\zeta}\frac{\partial q^v}{\partial\zeta} - (S^l + S^f) + M_{q^v}$$

$$\frac{\partial q^{l,f}}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial q^{l,f}}{\partial\lambda} + v\cos\varphi\frac{\partial q^{l,f}}{\partial\varphi}\right)\right\} - \dot{\zeta}\frac{\partial q^{l,f}}{\partial\zeta} - \frac{g}{\sqrt{\gamma}}\frac{\rho_0}{\rho}\frac{\partial P_{l,f}}{\partial\zeta} + S^{l,f} + M_{q^{l,f}}$$

turbulence

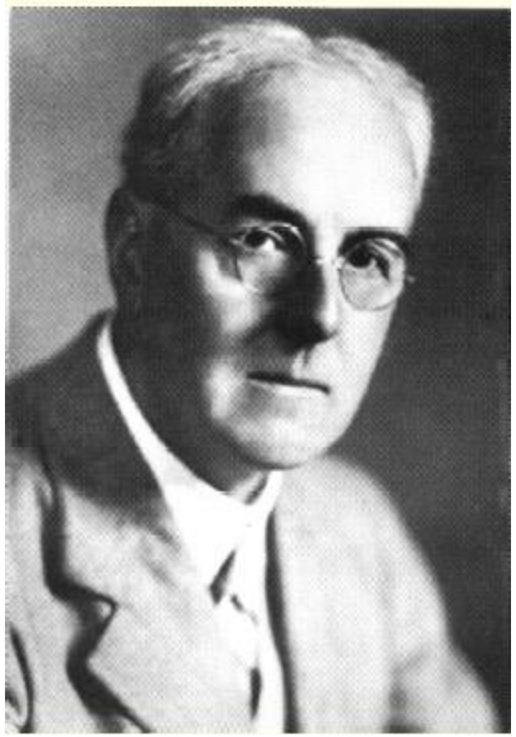
$$\frac{\partial e_t}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial e_t}{\partial\lambda} + v\cos\varphi\frac{\partial e_t}{\partial\varphi}\right)\right\} - \dot{\zeta}\frac{\partial e_t}{\partial\zeta} + K_m^v\frac{g\rho_0}{\sqrt{\gamma}}\left\{\left(\frac{\partial u}{\partial\zeta}\right)^2 + \left(\frac{\partial v}{\partial\zeta}\right)^2\right\} + \frac{g}{\rho\theta_v}F^{\theta_v} - \frac{\sqrt{2}e_t^{3/2}}{\alpha_M l} + M_{e_t}$$

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre



Vilhelm Bjerknes

- **Lewis F. Richardson**
  - About same time as Bjerknes; **WWI** (World War I) ambulance driver
  - Used continuity equation to obtain initial vertical velocities, as well as the other “primitive equations”
  - Failed due to insufficient initial data
    - Solved equations by hand!
    - **Time steps** were too large – would have resulted in computational instability



Source: Historical Essays on Meteorology 1919-1995, AMS

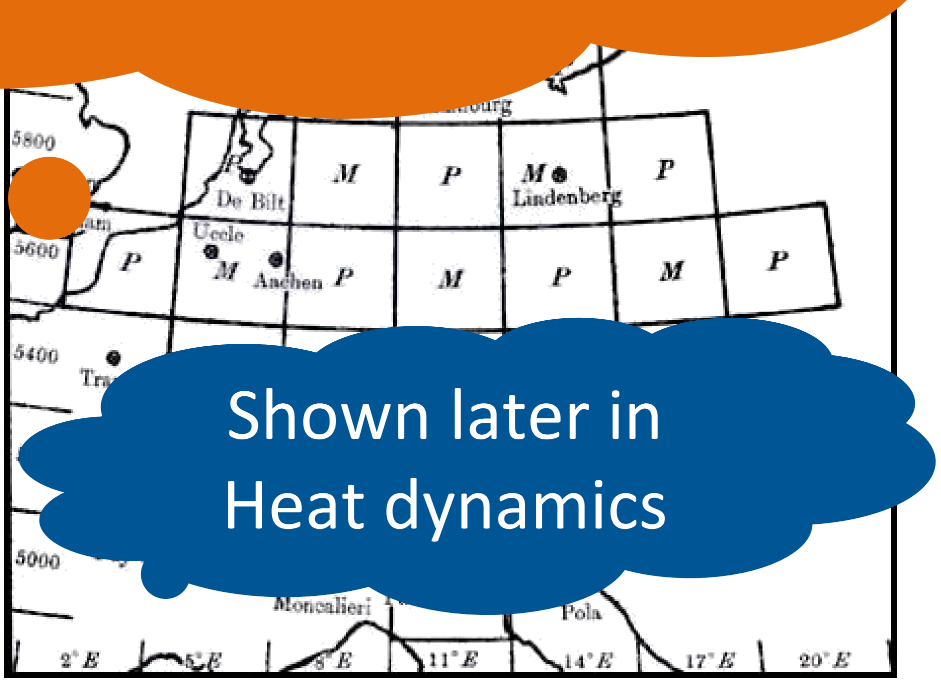
First numerical forecas
1922 by Lewis Fry Richa

Took several months, calculati
hand, to produce a 6-hour forecast.

It failed…badly!

But, it demonstrated the means of producing quantitative forecasts. Its failure has since been shown to be due to the limited understanding of some atmospheric processes at the time.

Yes, we need numeric weather prediction – by using computers/supercomputers/HPC

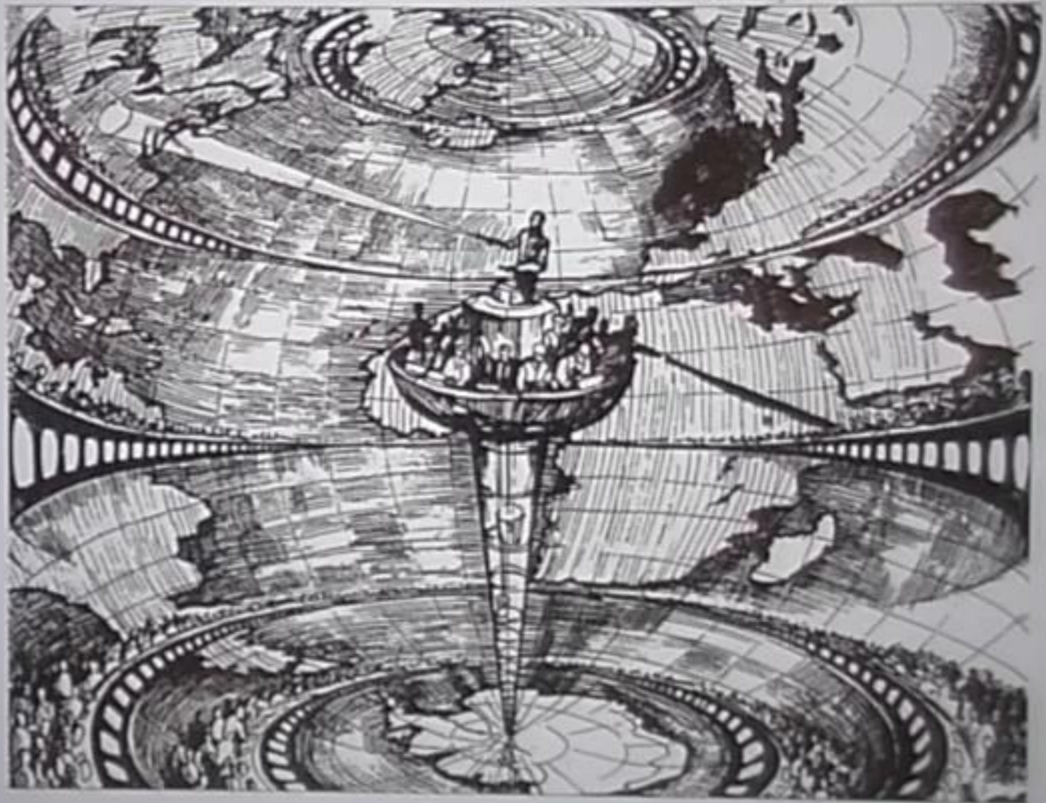


Shown later in Heat dynamics

L. F. Richardson’s computational grid: Pressure is determined in squares marked ‘P’, momentum in those marked ‘M’.

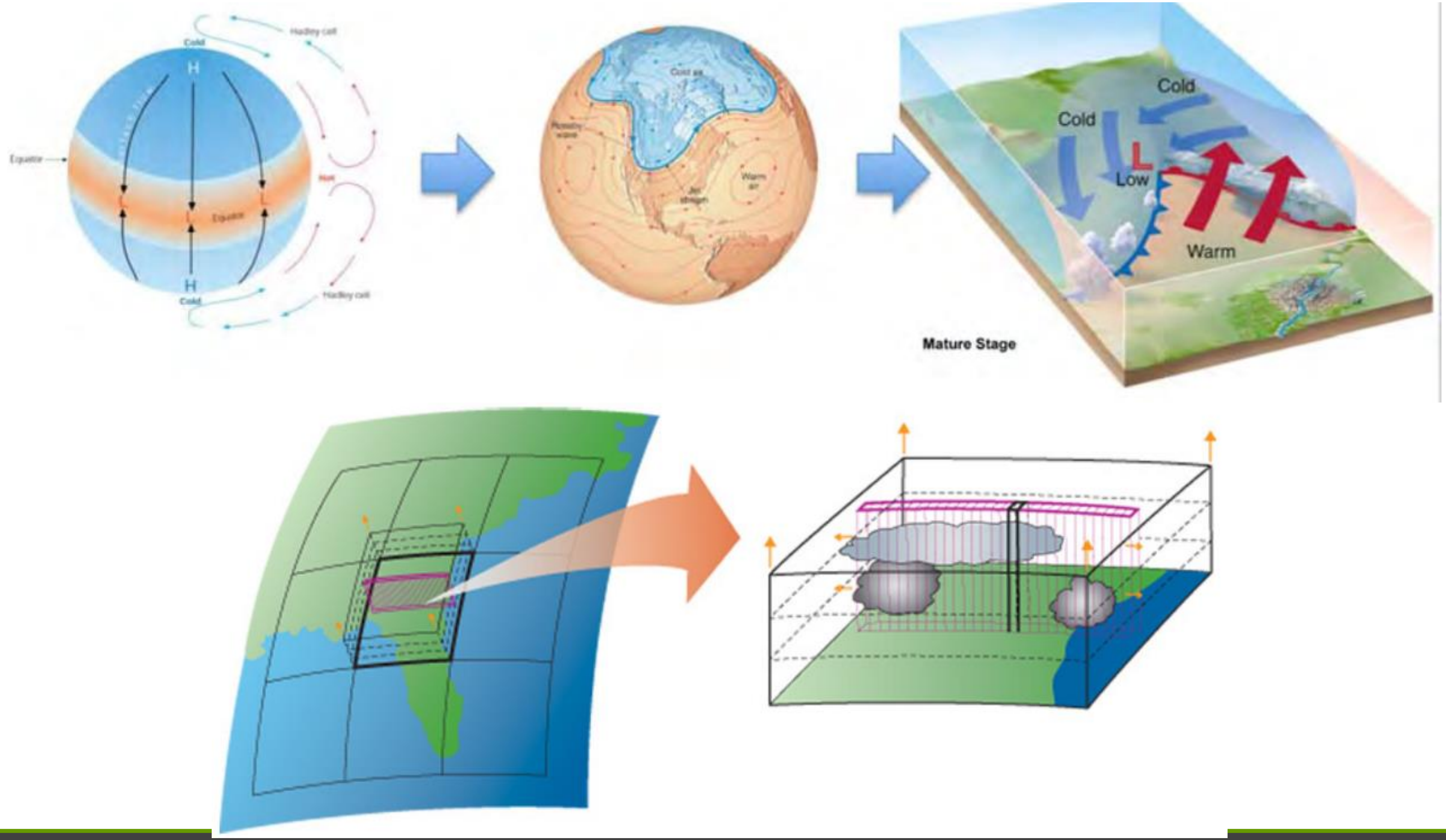
Richardson's forecast factory (1922)

Lewis Fry Richardson: Weather Prediction by Numerical Process

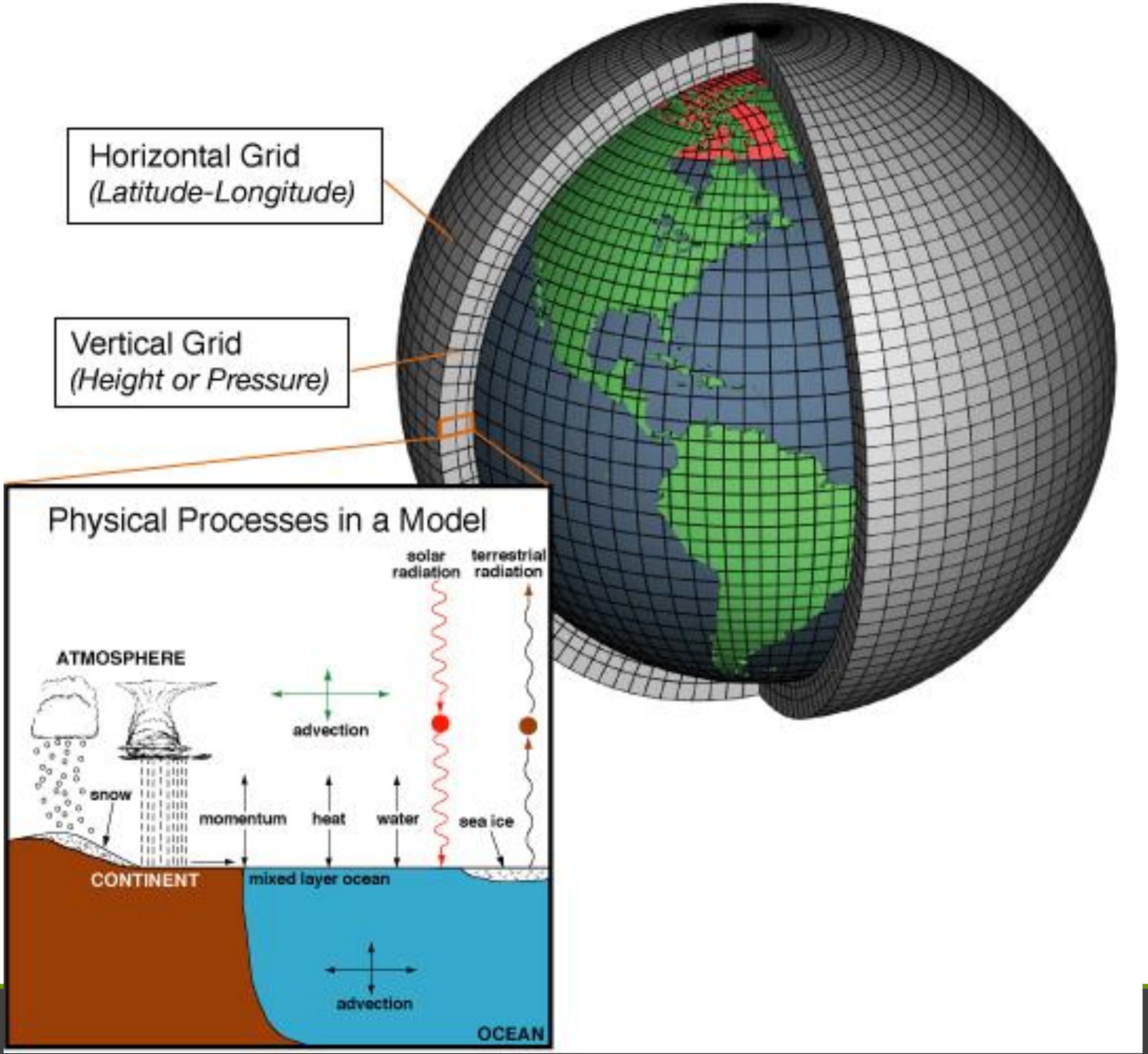


L. Bengtsson & NOAA

So called Numeric Computing



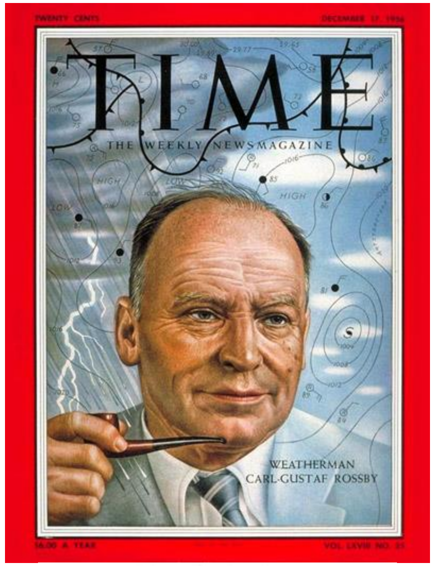
https://en.wikipedia.org/wiki/Numerical_weather_prediction



vorticity 美 [vɔ'tɪsɪti]
n. 旋涡状态; 涡量; 涡旋; 旋度;

geostrophic 美 [ˌdʒiə'strɒfɪk]
adj. 因地球自转而引起的，地转风的;
hydrostatic 美 [ˌhaɪdrə'stætɪk]
adj. 静水力学的，流体静力学的;

Rossby in 1933



□ Further Developments

- Carl-Gustaf Rossby (1939)
  - ➢Showed that atmospheric longwave motion could be explained by **vorticity** distribution
  - ➢Wave movement function of wavelength and speed of large-scale zonal flow (Rossby Waves)
- Jule Charney (1949)
  - ➢Developed first **barotropic** model
    - ✓ Large-scale motions approximately **geostrophic** and **hydrostatic**; no vertical motions; no vertical wind shear
  - ➢Numerical prediction now realizable as soon as computers become powerful enough to run the computations

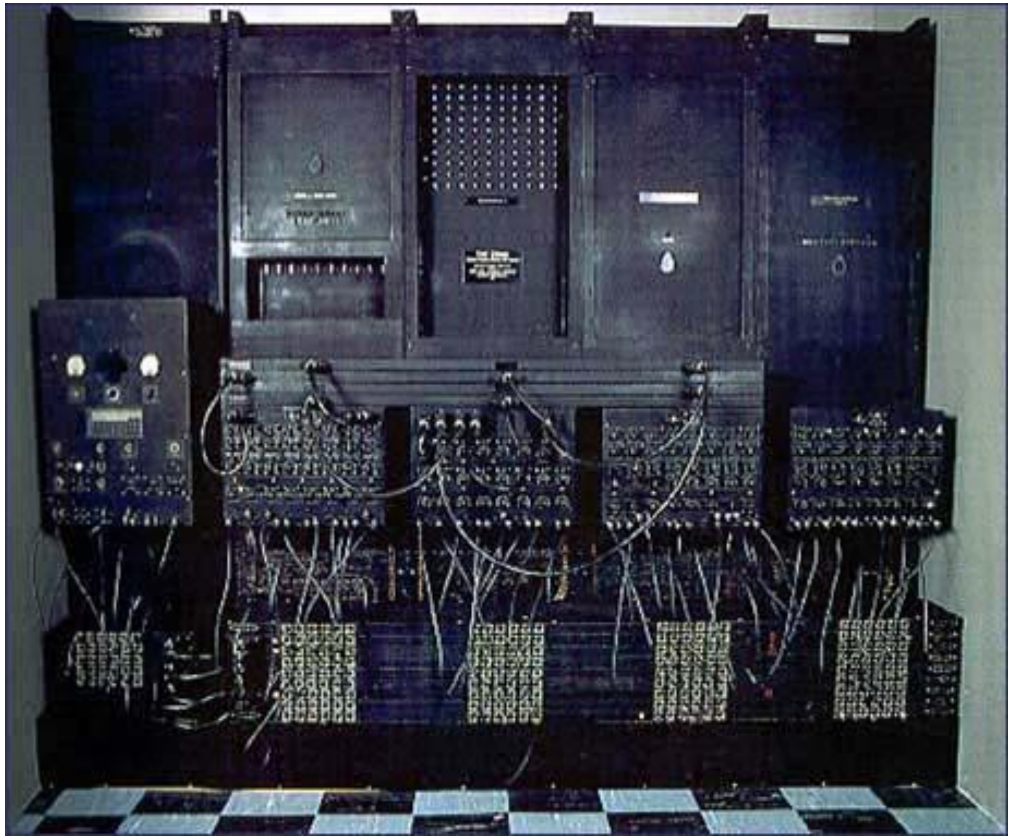
barotropic adj. 正压的;

Jule G. Charney

First Numerical Forecast

- Charney barotropic model run on **ENIAC** computer (1950)
  - Produced 500 mb (millibar) height forecast
  - Bad forecast but looked realistic

Jule Charney

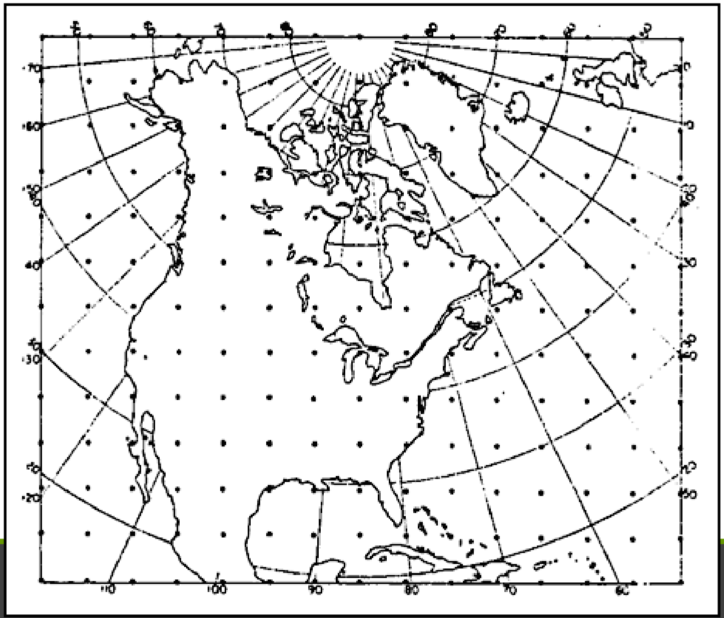


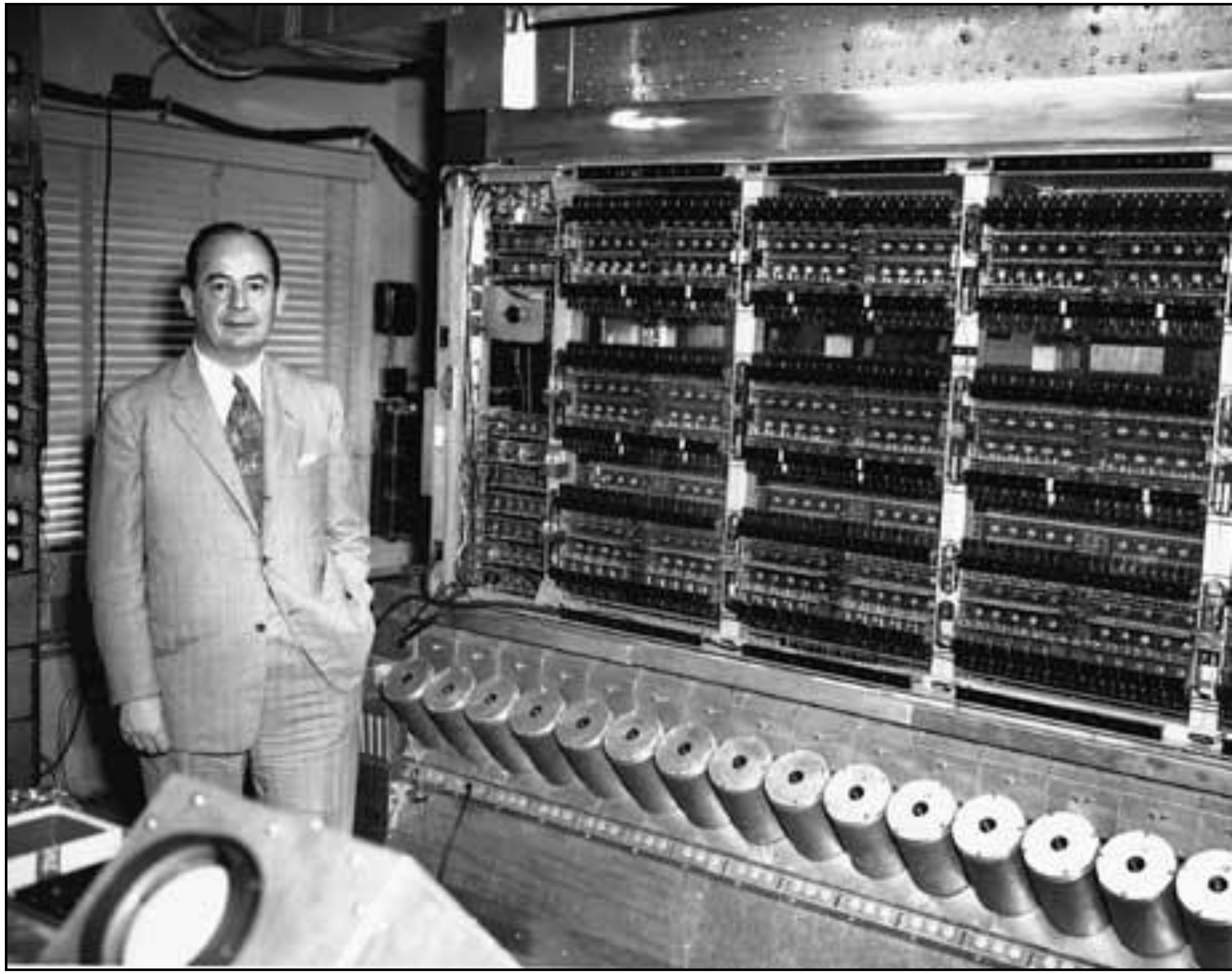
ENIAC Computer

Source: Historical Essays on Meteorology 1919-1995, AMS

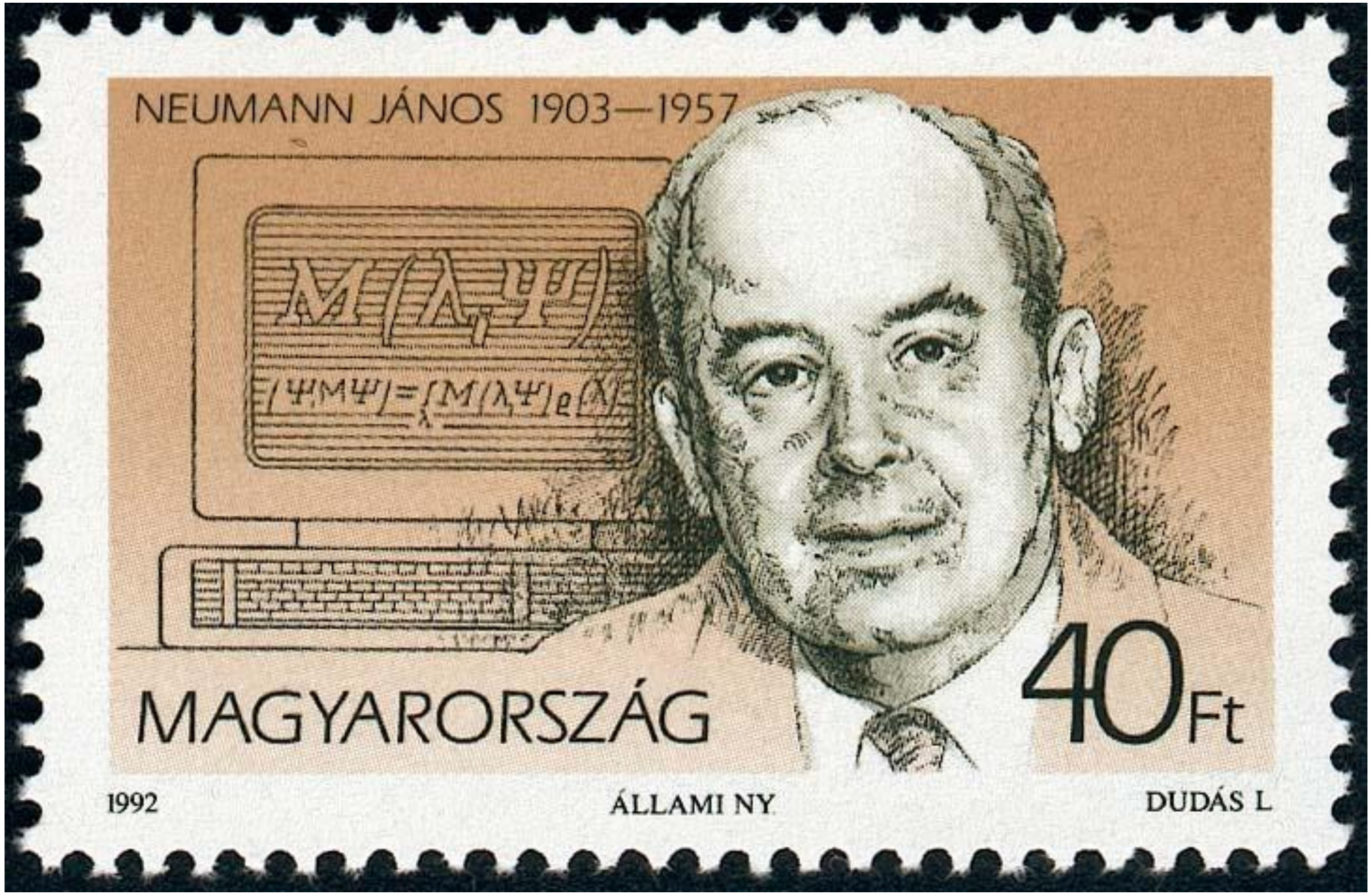
First successful forecast: 1950 by Jule Charney, Fjörtoft, and von Neumann, using ENIAC.

A 24-hour forecast took 33 days to produce, working day and night.





1950 ENIAC Meteorology Simulations



Jule Charney was the NWP (Numerical weather prediction) super hero

Operational Numerical Weather Prediction

- May 6, 1955 – First regular and continuing NWP forecasts issued for U.S.
  - Early results worse than lab experiments
  - Limitations of assumptions made in the models (Quasi-geostrophic approximation)
  - Many storms missed; public confidence wanes
  - Many early problems due to bad input data from global observation networks
  - **Barotropic model** (Charney) worked best for several years
    - ✓ Barotropic processes mostly controlled large-scale daily motions (e.g. long waves), while baroclinic controlled short-bursts of activity (e.g. mid-latitude cyclones)

barotropic adj. 正压的;

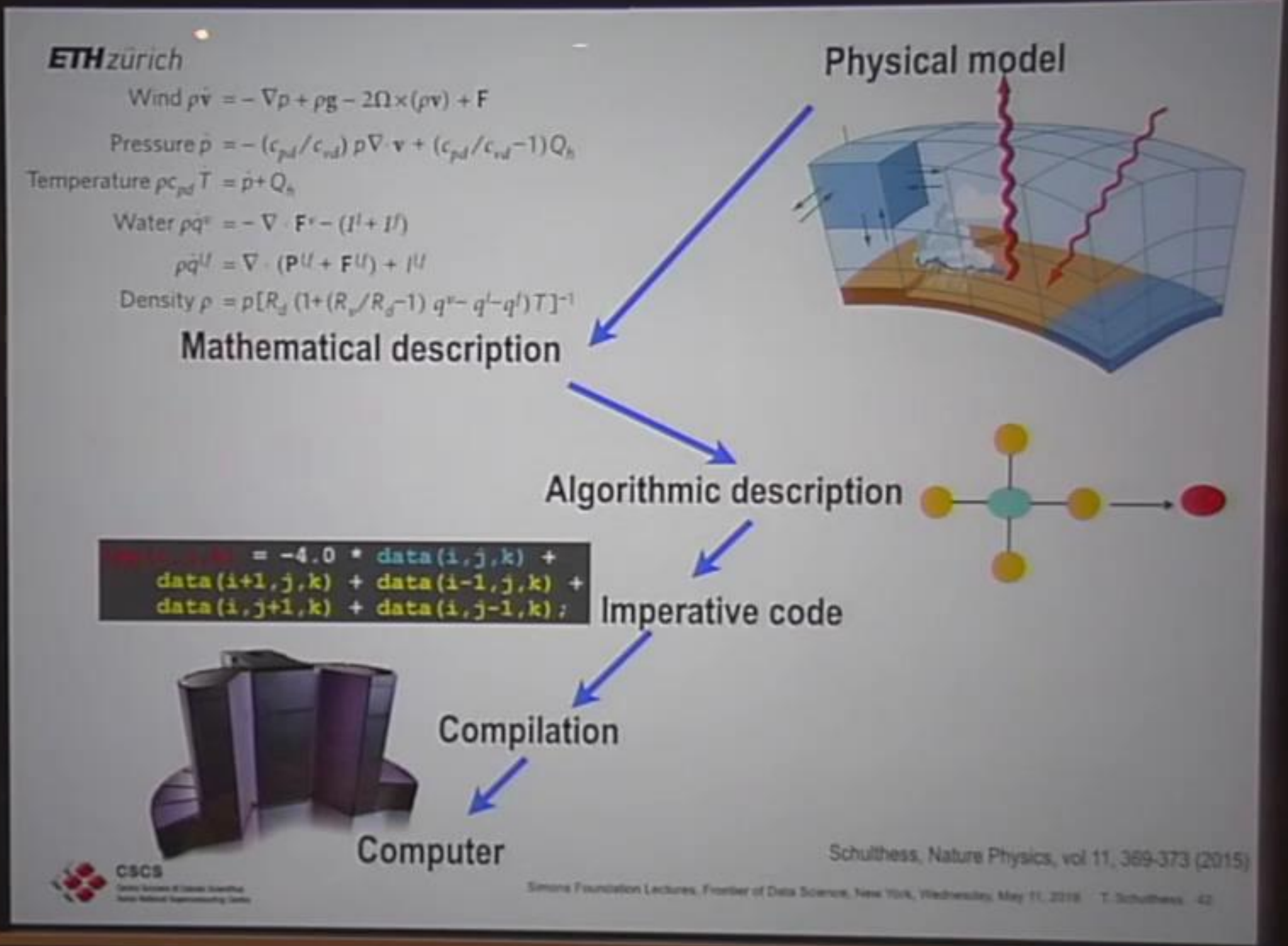
Barotropic Model

- Barotropic atmosphere (**constant density**/**temperature** on **pressure surface**, **no vertical motion**)
- Absolute vorticity conserved

$$\frac{d\left(f+\zeta\right)}{dt}=0$$

vorticity D.J.[vɔːˈtisəti] n. 旋涡状态

- Somewhat skillful at large-scale wave prediction



Physical model

$$
\begin{aligned}
\text{Wind } \rho\dot{\mathbf{v}} &= -\nabla p + \rho\mathbf{g} - 2\Omega\times(\rho\mathbf{v}) + \mathbf{F}\\
\text{Pressure } \dot{p} &= -(c_{pd}/c_{vd})\, p\nabla\cdot\mathbf{v} + (c_{pd}/c_{vd}-1)Q_h\\
\text{Temperature } \rho c_{pd}\dot{T} &= \dot{p}+Q_h\\
\text{Water } \rho\dot{q}^v &= -\nabla\cdot\mathbf{F}^v-(I^l+I^f)\\
\rho\dot{q}^{l,f} &= \nabla\cdot(\mathbf{P}^{l,f}+\mathbf{F}^{l,f})+I^{l,f}\\
\text{Density } \rho &= p[R_d\,(1+(R_v/R_d-1)\,q^v-q^l-q^f)T]^{-1}
\end{aligned}
$$

Mathematical description

Algorithmic description

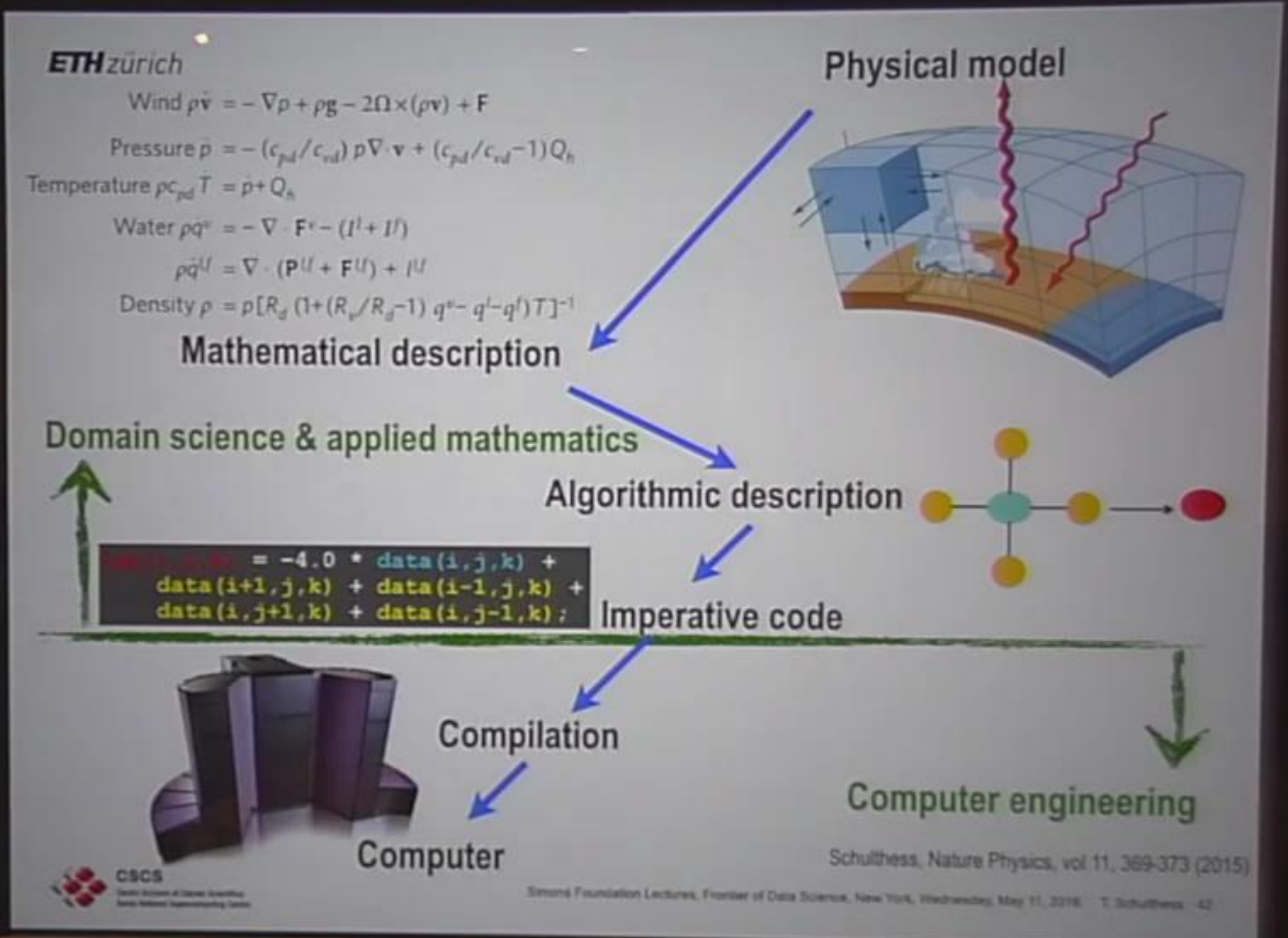
```
[illegible] = -4.0 * data(i,j,k) +
    data(i+1,j,k) + data(i-1,j,k) +
    data(i,j+1,k) + data(i,j-1,k);
```

Imperative code

Compilation

Computer

Schulthess, Nature Physics, vol 11, 369-373 (2015)



Physical model

Wind $\rho\dot{\mathbf{v}} = -\nabla p + \rho\mathbf{g} - 2\Omega\times(\rho\mathbf{v}) + \mathbf{F}$

Pressure $\dot{p} = -(c_{pd}/c_{vd})\,p\nabla\cdot\mathbf{v} + (c_{pd}/c_{vd}-1)Q_h$

Temperature $\rho c_{pd}\dot{T} = \dot{p} + Q_h$

Water $\rho\dot{q}^v = -\nabla\cdot\mathbf{F}^v - (I^l + I^f)$

$\rho\dot{q}^{l,f} = \nabla\cdot(\mathbf{P}^{l,f} + \mathbf{F}^{l,f}) + I^{l,f}$

Density $\rho = p[R_d\,(1+(R_v/R_d-1)\,q^v - q^l - q^f)\,T]^{-1}$

Mathematical description

Domain science & applied mathematics

Algorithmic description

```
[illegible] = -4.0 * data(i,j,k) +
      data(i+1,j,k) + data(i-1,j,k) +
      data(i,j+1,k) + data(i,j-1,k);
```

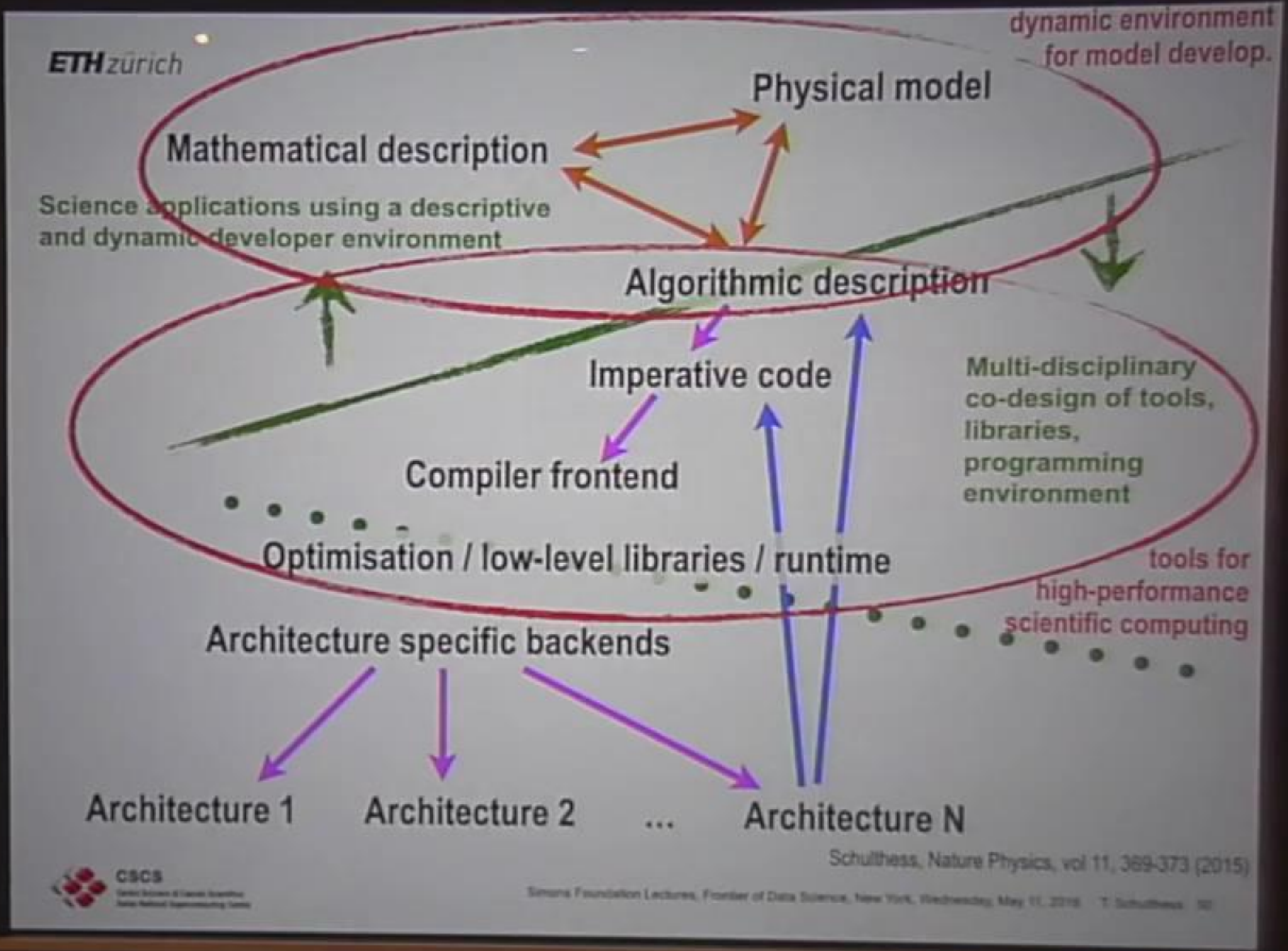
Imperative code

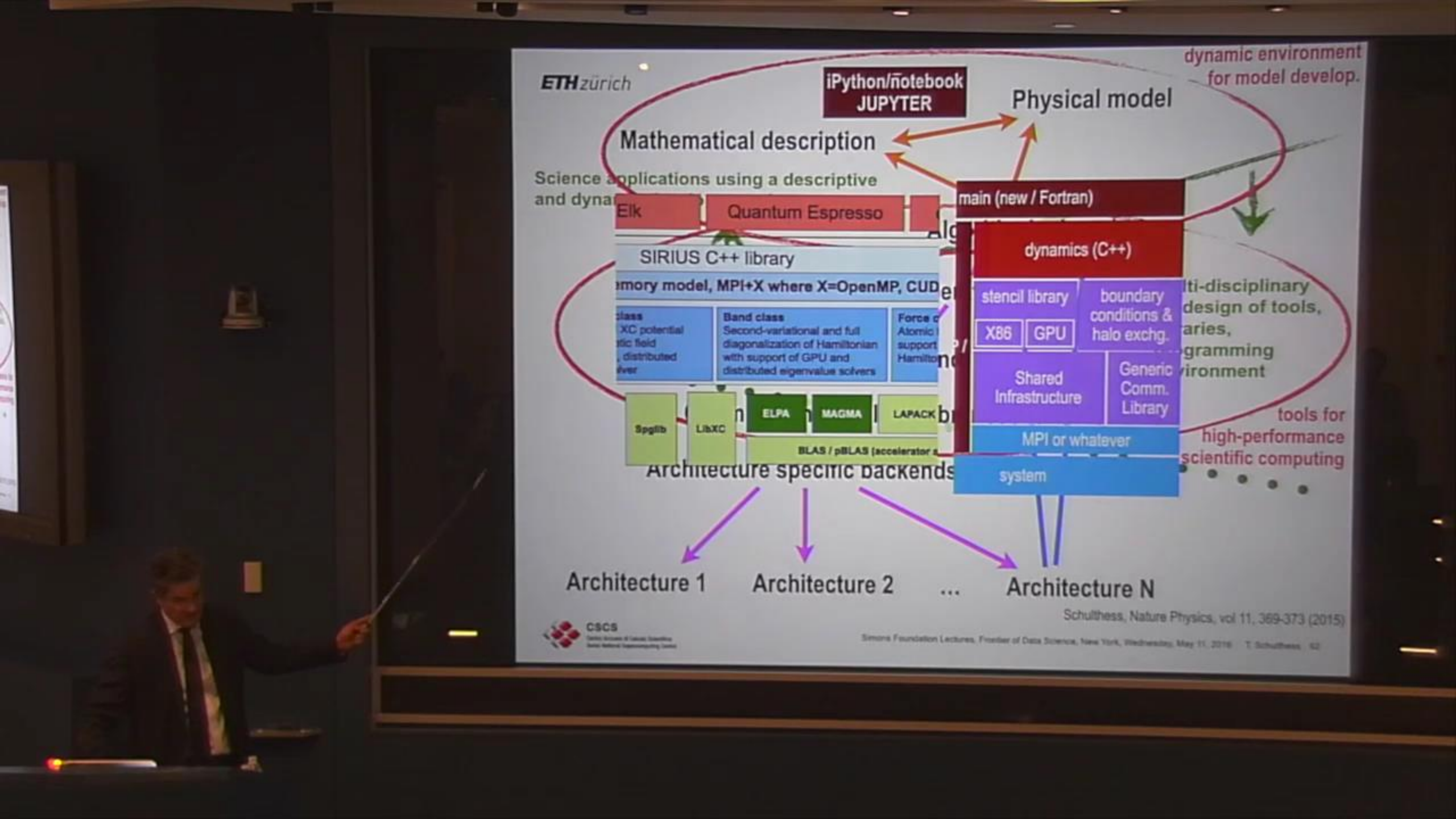
Compilation

Computer

Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)





ETH zürich

dynamic environment
for model develop.

iPython/notebook
JUPYTER

Physical model

Mathematical description

Science applications using a descriptive
and dyna

Elk

Quantum Espresso

main (new / Fortran)

Alg

SIRIUS C++ library

emory model, MPI+X where X=OpenMP, CUD

class
XC potential
tic field
, distributed
lver

Band class
Second-variational and full diagonalization of Hamiltonian with support of GPU and distributed eigenvalue solvers

Force c
Atomic
support
Hamilto

dynamics (C++)

stencil library

X86

GPU

boundary conditions & halo exchg.

Shared Infrastructure

Generic Comm. Library

MPI or whatever

system

lti-disciplinary
design of tools,
aries,
gramming
ironment

Spglib

LibXC

ELPA

MAGMA

LAPACK

BLAS / pBLAS (accelerator s

Architecture specific backends

tools for
high-performance
scientific computing

Architecture 1

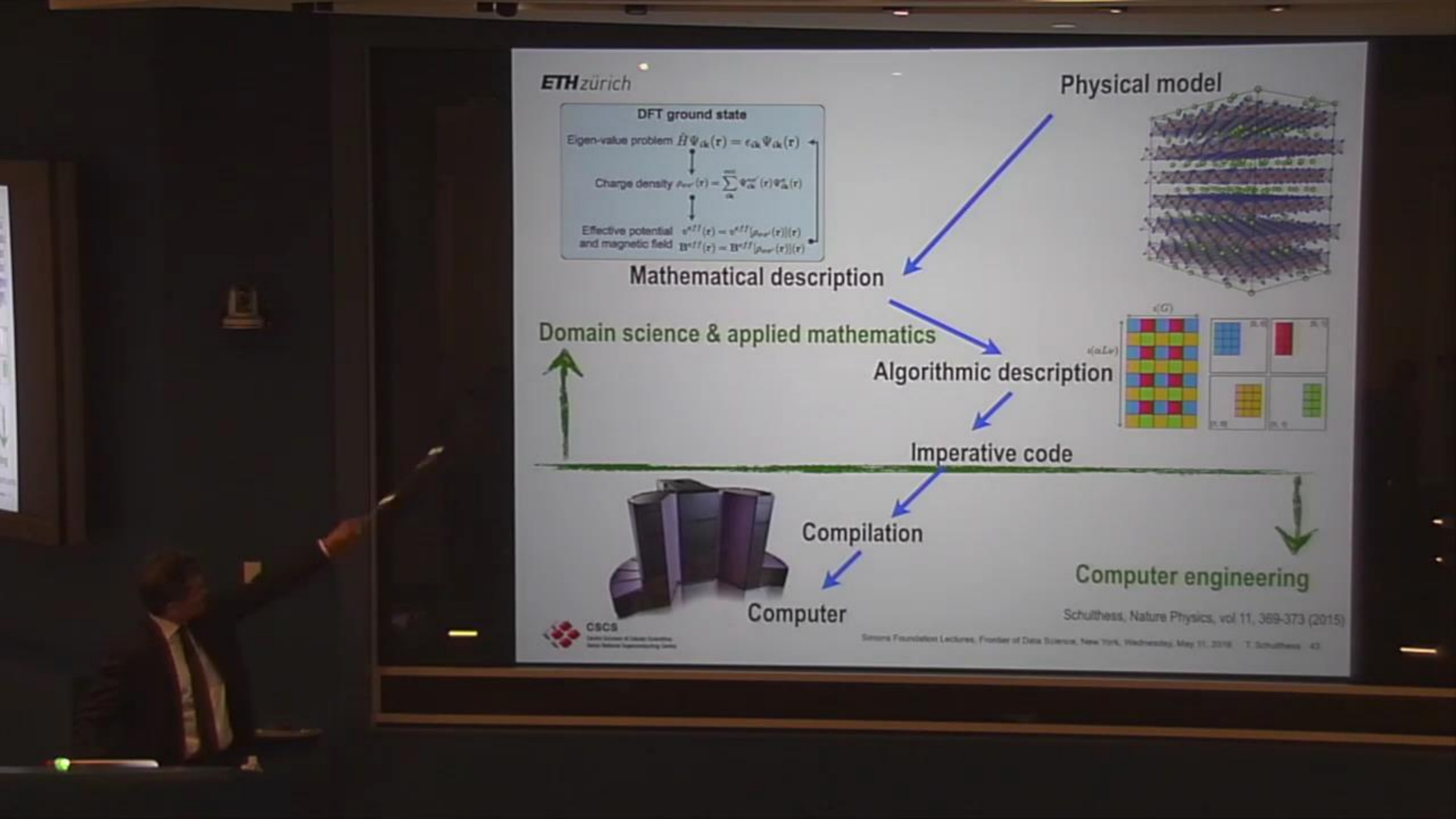
Architecture 2

...

Architecture N

Schulthess, Nature Physics, vol 11, 369-373 (2015)

CSCS



ETH zürich

DFT ground state

Eigen-value problem $\hat{H}\Psi_{i\mathbf{k}}(\mathbf{r}) = \epsilon_{i\mathbf{k}}\Psi_{i\mathbf{k}}(\mathbf{r})$

Charge density $\rho_{\sigma\sigma'}(\mathbf{r}) = \sum_{i\mathbf{k}}^{occ} \Psi^{\sigma}_{i\mathbf{k}}(\mathbf{r})\Psi^{\sigma'*}_{i\mathbf{k}}(\mathbf{r})$

Effective potential and magnetic field $v^{eff}(\mathbf{r}) = v^{eff}[\rho_{\sigma\sigma'}(\mathbf{r})](\mathbf{r})$ $\mathbf{B}^{eff}(\mathbf{r}) = \mathbf{B}^{eff}[\rho_{\sigma\sigma'}(\mathbf{r})](\mathbf{r})$

Physical model

Mathematical description

Domain science & applied mathematics

Algorithmic description

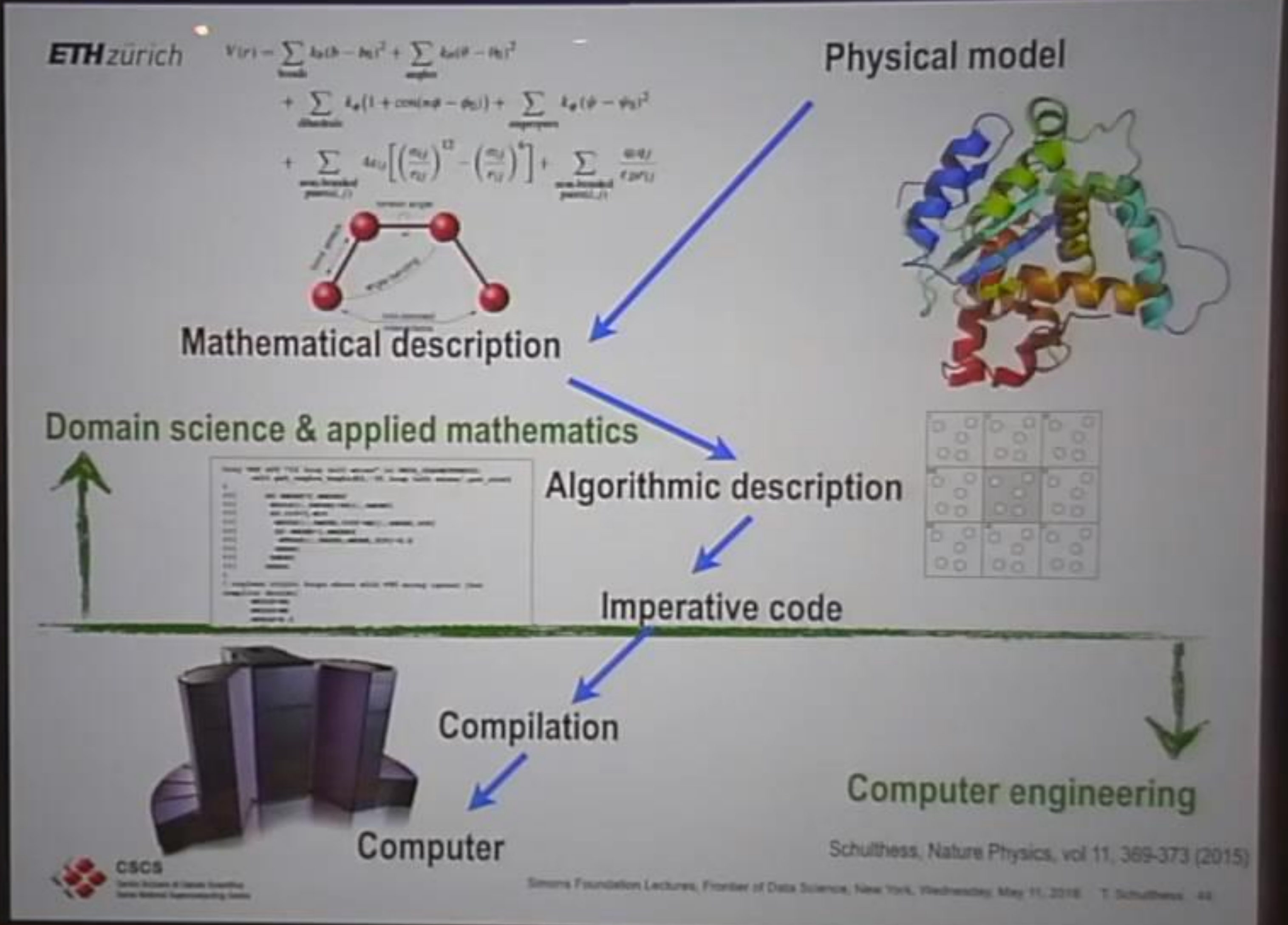
Imperative code

Compilation

Computer

Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)



ETH zürich

$$V(r) = \sum_{\text{bonds}} k_b(b - b_0)^2 + \sum_{\text{angles}} k_\theta(\theta - \theta_0)^2 + \sum_{\text{dihedrals}} k_\phi(1 + \cos(n\phi - \phi_0)) + \sum_{\text{impropers}} k_\psi(\psi - \psi_0)^2 + \sum_{\substack{\text{nonbonded}\\ \text{pairs}(i,j)}} 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \sum_{\substack{\text{nonbonded}\\ \text{pairs}(i,j)}} \frac{q_i q_j}{\epsilon r_{ij}}$$

Physical model

Mathematical description

Domain science & applied mathematics

Algorithmic description

Imperative code

Compilation

Computer

Computer engineering

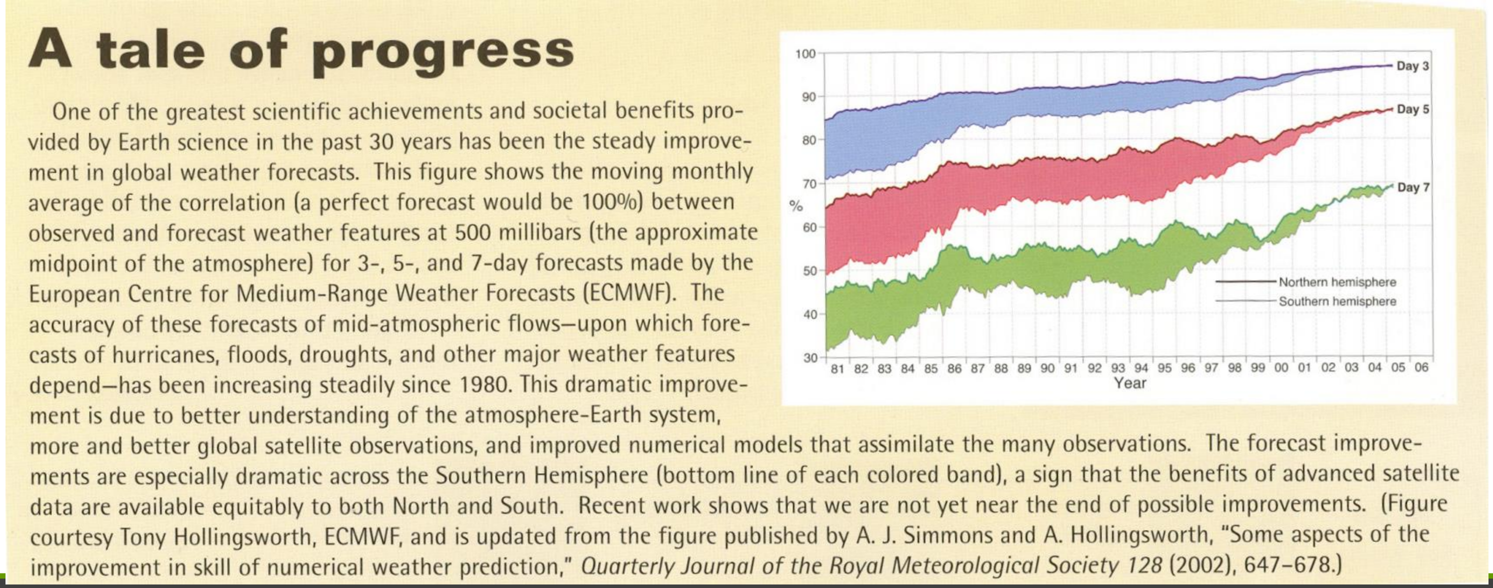
CSCS

Schulthess, Nature Physics, vol 11, 369-373 (2015)

Types of Numerical Models

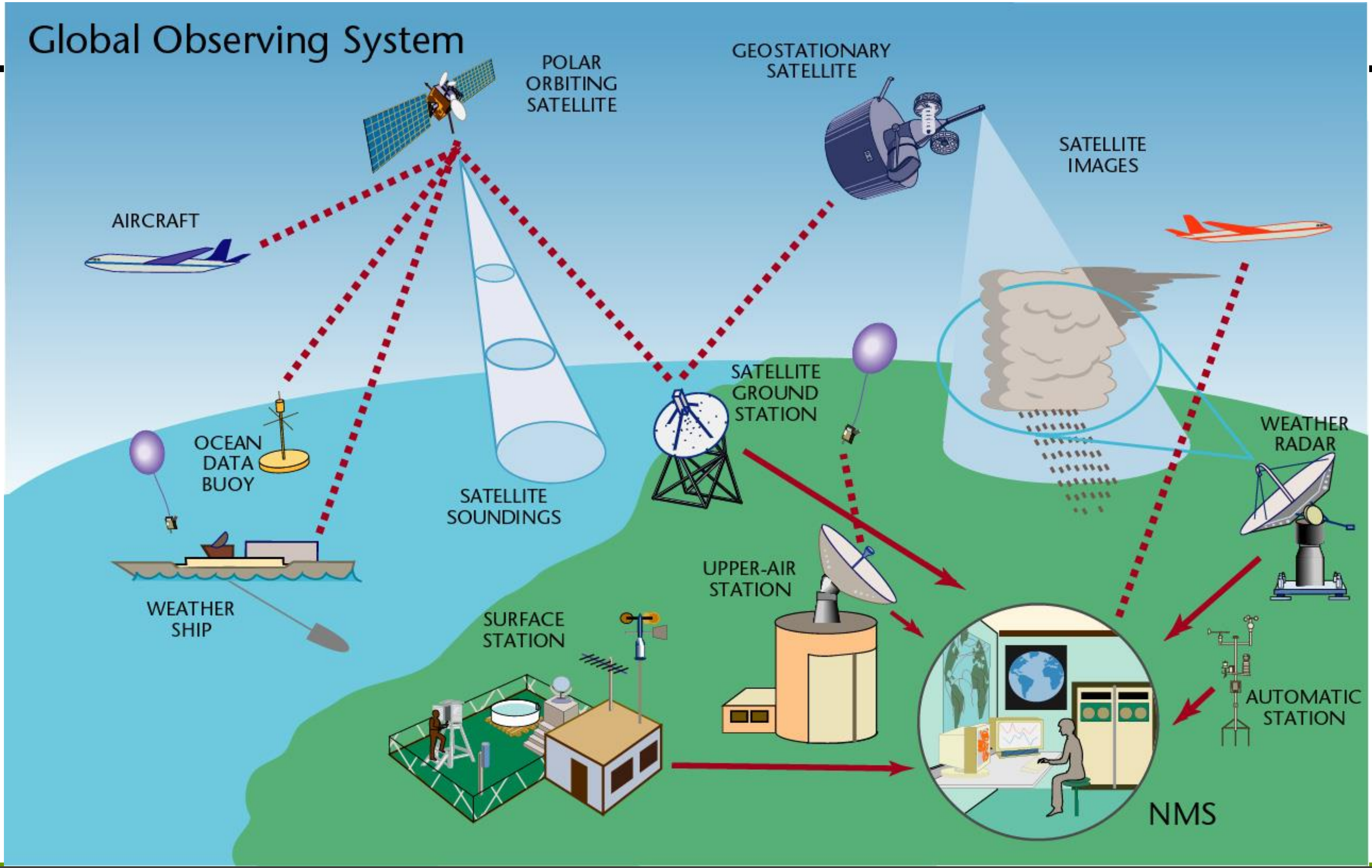
The Final Major Evolution

- Successful NWP using full suite of primitive equations occurred in 1966
  - Original Bjerknes/Richardson vision!
- Advances in computational power and improvements in input data let to acceptable forecasts
- Forecasts improved almost 50% in 10-years (1955-1965) from subjective forecasts



A tale of progress

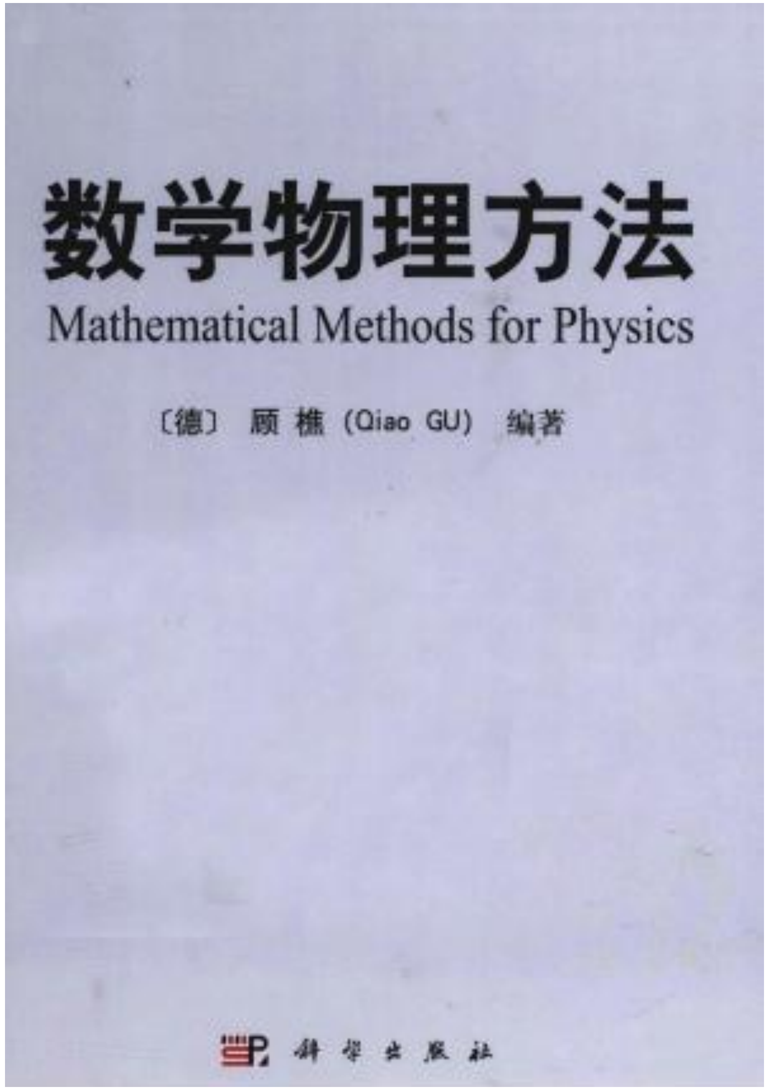
One of the greatest scientific achievements and societal benefits provided by Earth science in the past 30 years has been the steady improvement in global weather forecasts. This figure shows the moving monthly average of the correlation (a perfect forecast would be 100%) between observed and forecast weather features at 500 millibars (the approximate midpoint of the atmosphere) for 3-, 5-, and 7-day forecasts made by the European Centre for Medium-Range Weather Forecasts (ECMWF). The accuracy of these forecasts of mid-atmospheric flows—upon which forecasts of hurricanes, floods, droughts, and other major weather features depend—has been increasing steadily since 1980. This dramatic improvement is due to better understanding of the atmosphere-Earth system, more and better global satellite observations, and improved numerical models that assimilate the many observations. The forecast improvements are especially dramatic across the Southern Hemisphere (bottom line of each colored band), a sign that the benefits of advanced satellite data are available equitably to both North and South. Recent work shows that we are not yet near the end of possible improvements. (Figure courtesy Tony Hollingsworth, ECMWF, and is updated from the figure published by A. J. Simmons and A. Hollingsworth, "Some aspects of the improvement in skill of numerical weather prediction," *Quarterly Journal of the Royal Meteorological Society 128* (2002), 647–678.)


Global Observing System
POLAR ORBITING SATELLITE
GEOSTATIONARY SATELLITE
SATELLITE IMAGES
AIRCRAFT
SATELLITE GROUND STATION
OCEAN DATA BUOY
SATELLITE SOUNDINGS
WEATHER RADAR
UPPER-AIR STATION
WEATHER SHIP
SURFACE STATION
AUTOMATIC STATION
NMS

Chapter 2: Overview of HPC

□ Faster for larger data

- Many problems/applications need HPComputers
  - ➢ **Weather / Climate**， Cryptography，Nuclear Weapons Design，Scientific Simulation，Petroleum Exploration，Aerospace Design，Automotive Design，Pharmaceutical Design，Data Mining，Data Assimilation
- Heat dynamics as an example to understand Numeric Computing/Scientific Computing
  - ➢ Such as **F**inite **E**lement **M**odeling (FEM: 有限元分析)



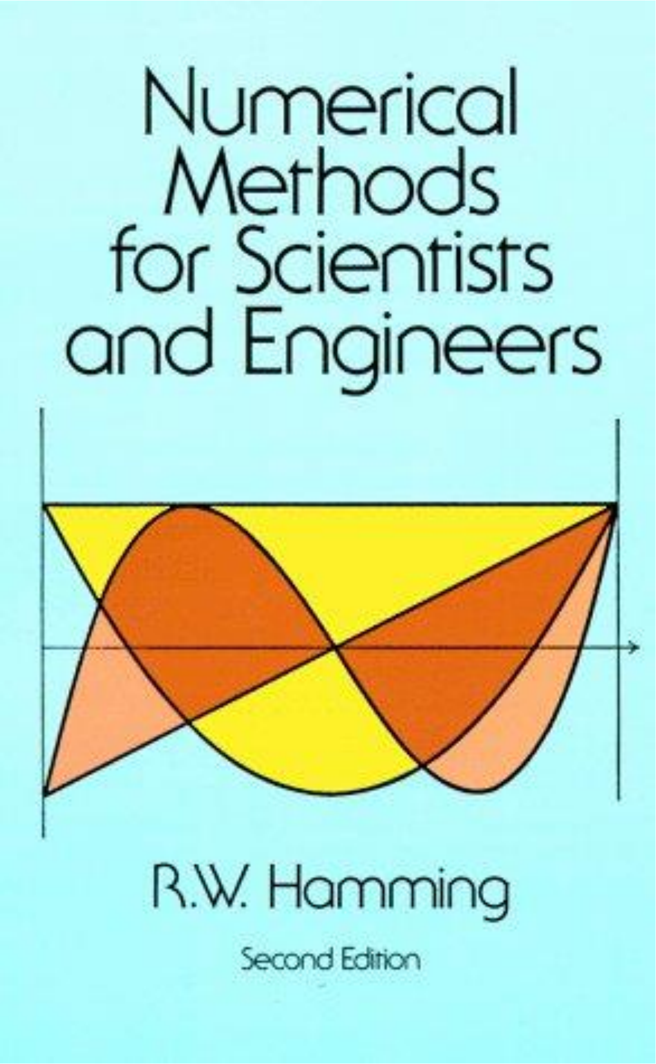
数学物理方法

Mathematical Methods for Physics

〔德〕 顾 樵 (Qiao GU) 编著

科 学 出 版 社

- □ 数学物理方法
- □ *顾樵*
  - ■ 由顾樵编著的《数学物理方法》根据作者顾樵20多年来在德国和中国开设数学物理方法讲座内容及相关的研究成果提炼而成。其主要内容包括傅里叶级数、傅里叶变换、拉普拉斯变换、数学物理方程的建立、分离变量法、本征函数法、施图姆—刘维尔理论、行波法、积分变换法、格林函数法、贝塞尔函数、勒让德多项式、量子力学薛定谔方程等。本书注重自身理论体系的科学性、严谨性、完整性与实用性，将中国传统教材讲授内容与国外先进教材相结合、教学实践与其他相关课程的需要相结合、抽象的数理概念与直观的物理实例相结合、经典的数理方法与新兴交叉学科的生长点相结合、基础的数理知识与科学前沿中的热点问题相结合。本书既可为教学所用，又可适应科研需要，同时，附有大



Numerical Methods for Scientists and Engineers

R.W. Hamming

Second Edition

- Numerical methods for scientists and engineers
- Richard Hamming

偏微分方程
数值解法（第2版）

陆金甫 关 治 编著

清华大学出版社

- □ 偏微分方程数值解法
- □ 第二版
- □ 陆金甫，关治

- □ 数值最优化算法与理论(第二版)
- □ 李董辉, 童小娇, 万中

Differential Equation – Basic

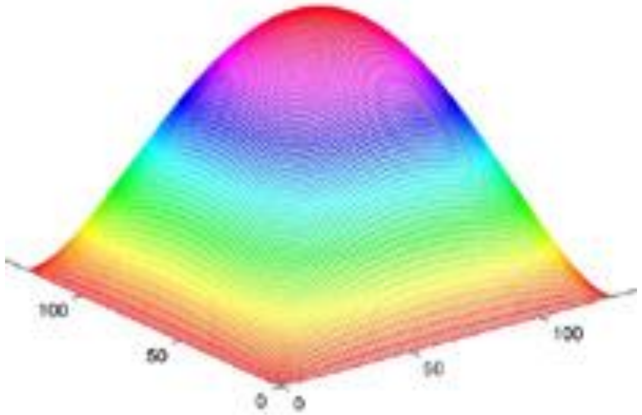
- dimension of unknown:
  - *ordinary differential equation* (**ODE**) – unknown is a function of one variable, e.g. $y(t)$
  - *partial differential equation* (**PDE**) – unknown is a function of multiple variables, e.g. $u(t,x,y)$

- number of equations:
  - *single* differential equation, e.g. $y' = y$
  - *system* of differential equations (*coupled*), e.g. $y_1' = y_2$, $y_2' = -g$
- *order*
  - *n*th order DE has *n*th derivative, and no higher, e.g. $y'' = -g$

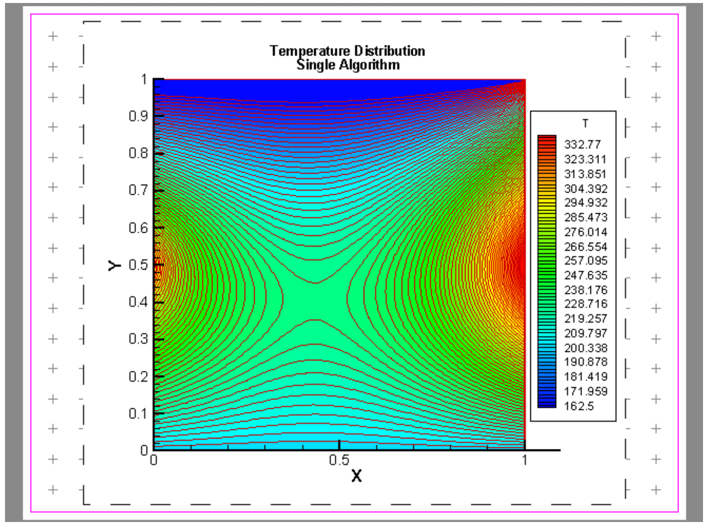
- linear & nonlinear:
  - *linear* differential equation: all terms linear in unknown and its derivatives
  - e.g.
    - $x''+ax'+bx+c=0$ – linear
    - $x'=t^2x$ – linear
    - $x''=1/x$ – nonlinear

PDE’s in Science & Engineering

- Laplace’s Equation: $\nabla^2 u = u_{xx} + u_{yy} + u_{zz} = 0$
  - unknown: $u(x,y,z)$
  - gravitational / electrostatic potential



- Heat Equation: $u_t = a^2\nabla^2 u$
  - unknown: $u(t,x,y,z)$
  - heat conduction



- Wave Equation: $u_{tt} = a^2\nabla^2 u$
  - unknown: $u(t,x,y,z)$
  - wave propagation

Laplace Equation

$$\frac{\partial^2 u(x,y,z)}{\partial x^2}+\frac{\partial^2 u(x,y,z)}{\partial y^2}+\frac{\partial^2 u(x,y,z)}{\partial z^2}=0$$

Used to describe the steady state distribution of heat in a body.

Also used to describe the steady state distribution of electrical charge in a body.

Heat Equation

$$\frac{\partial u(x,y,z,t)}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)$$

The function u(x,y,z,t) is used to represent the temperature at time *t* in a physical body at a point with coordinates (x,y,z)

α is the thermal diffusivity. It is sufficient to consider the case **α = 1**.

Wave Equation

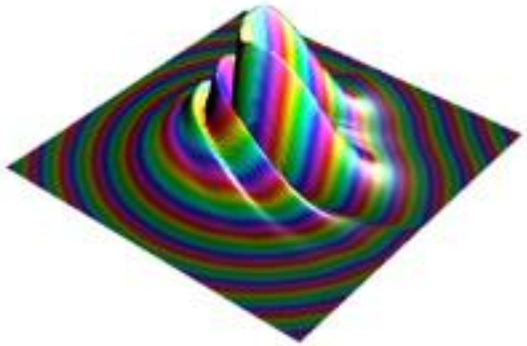
$$\frac{\partial^2 u(x,y,z,t)}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right)$$

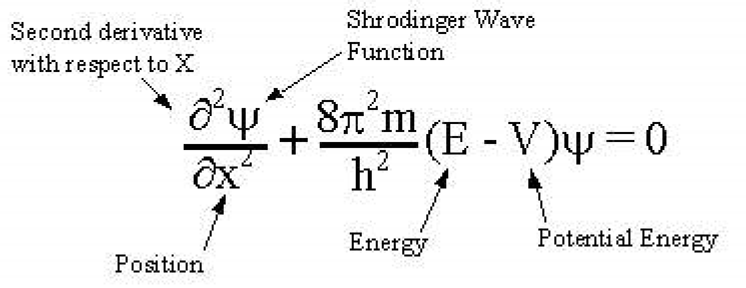
The function *u(x,y,z,t)* is used to represent the displacement at time t of a particle whose position at rest is *(x,y,z)* .

The constant *c* represents the propagation speed of the wave.

https://en.wikipedia.org/wiki/Schr%C3%B6dinger_equation https://www.physlink.com/education/askexperts/ae329.cfm

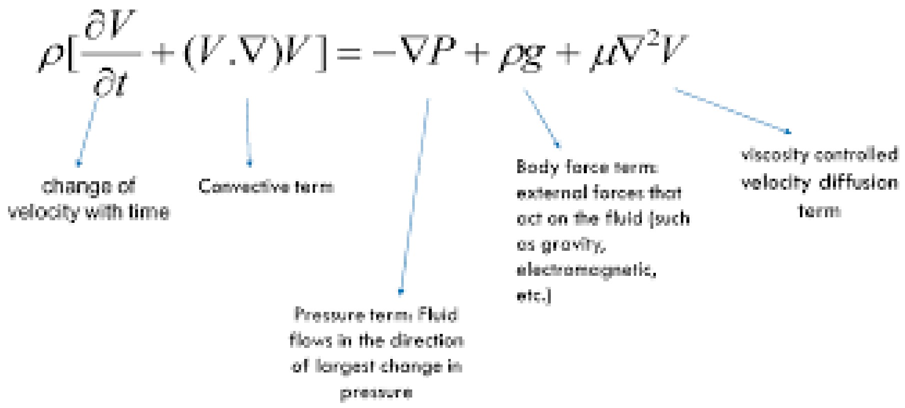
- Schrödinger Wave Equation
  - quantum mechanics
  - (electron probability densities)





$$\frac{\partial^2\psi}{\partial x^2} + \frac{8\pi^2 m}{h^2}(E - V)\psi = 0$$

- Navier-Stokes Equation
  - fluid flow (fluid velocity & pressure)



$$\rho[\frac{\partial V}{\partial t} + (V.\nabla)V] = -\nabla P + \rho g + \mu\nabla^2 V$$

change of velocity with time
Convective term
Pressure term: Fluid flows in the direction of largest change in pressure
Body force term: external forces that act on the fluid (such as gravity, electromagnetic, etc.)
viscosity controlled velocity diffusion term

Linear or Nonlinear

A PDE is linear if it is linear in the unknown function and its derivatives

Example of linear PDE :

$$2\,u_{xx} + 1\,u_{xt} + 3\,u_{tt} + 4\,u_x + \cos(2t) = 0$$

$$2\,u_{xx} - 3\,u_t + 4\,u_x = 0$$

Examples of Nonlinear PDE

$$2\,u_{xx} + \underline{(u_{xt})^2} + 3\,u_{tt} = 0$$

$$\underline{\sqrt{u_{xx}}} + 2\,u_{xt} + 3u_t = 0$$

$$2\,u_{xx} + \underline{2\,u_{xt}\,u_t} + 3u_t = 0$$

Convergence and Stability of the Solution

- □ **Convergence**

  The solutions converge means that the solution obtained using the finite difference method approaches the true solution as the steps approach zero.

  $$\Delta x \quad \text{and} \quad \Delta t$$

- □ **Stability:**

  An algorithm is stable if the errors at each stage of the computation are not magnified as the computation progresses.

Laplace Equation

Laplace equation appears in several engineering problems such as:

- Studying the steady state distribution of heat in a body.
- Studying the steady state distribution of electrical charge in a body.

$$\frac{\partial^2 T(x,y)}{\partial x^2} + \frac{\partial^2 T(x,y)}{\partial y^2} = f(x,y)$$

T : steady state temperature at point (x, y)

$f(x,y)$: heat source (or heat sink)

Solution Technique

- □ **A grid is used to divide the region of interest.**
  - ■ Since the PDE is satisfied at each point in the area, it must be satisfied at each point of the grid.
- □ **A finite difference approximation is obtained at each grid point.**

$$\frac{\partial^2\ T(x,y)}{\partial x^2} \approx \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta x)^2}, \quad \frac{\partial^2\ T(x,y)}{\partial y^2} \approx \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2}$$

Solution Technique

$$\frac{\partial^2\ T(x,y)}{\partial x^2} = \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta x)^2},$$

$$\frac{\partial^2\ T(x,y)}{\partial y^2} = \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2}$$

$$\Rightarrow \frac{\partial^2\ T(x,y)}{\partial x^2} + \frac{\partial^2\ T(x,y)}{\partial y^2} = 0$$

is approximated by:

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta x)^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2} = 0$$

Solution Technique

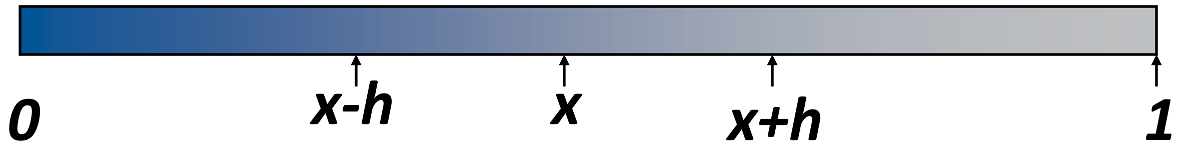
$$\frac{T_{i+1,j}-2T_{i,j}+T_{i-1,j}}{(\Delta x)^2}+\frac{T_{i,j+1}-2T_{i,j}+T_{i,j-1}}{(\Delta y)^2}=0$$

$$(Laplacian\ Difference\ Equation)$$

$$Assume:\Delta x=\Delta y=h$$

$$\Rightarrow T_{i+1,j}+T_{i-1,j}+T_{i,j+1}+T_{i,j-1}-4T_{i,j}=0$$

Example: Deriving the Heat Equation



Consider a simple problem

- A bar of uniform material, insulated except at ends
- Let *u(x,t)* be the temperature at position *x* at time *t*
- Heat travels from *x-h* to *x+h* at rate proportional to:

$$\frac{d\,u(x,t)}{dt} = C * \frac{(u(x-h,t)-u(x,t))/h - (u(x,t)- u(x+h,t))/h}{h}$$

- As *h* → *0*, we get the heat equation:

$$\frac{d\,u(x,t)}{dt} = C * \frac{d^2\,u(x,t)}{dx^2}$$

Details of the Explicit Method for Heat

$$\frac{d\,u(x,t)}{dt} = C * \frac{d^2\,u(x,t)}{dx^2}$$

- □ Discretize time and space using explicit approach (forward Euler) to approximate time derivative:

$$(u(x,t+\delta) - u(x,t))/\delta = C\ \ [\ (u(x-h,t)-u(x,t))/h - (u(x,t)- u(x+h,t))/h\ \ ]\ /\ h$$
$$= C\ [u(x-h,t) - 2*u(x,t) + u(x+h,t)]/h^2$$

Solve for $u(x,t+\delta)$ **:**

$$u(x,t+\delta) = u(x,t)+ C*\delta\ /h^2\ *(u(x-h,t) - 2*u(x,t) + u(x+h,t))$$

- □ Let z = $C*\delta\ /h^2$, simplify:

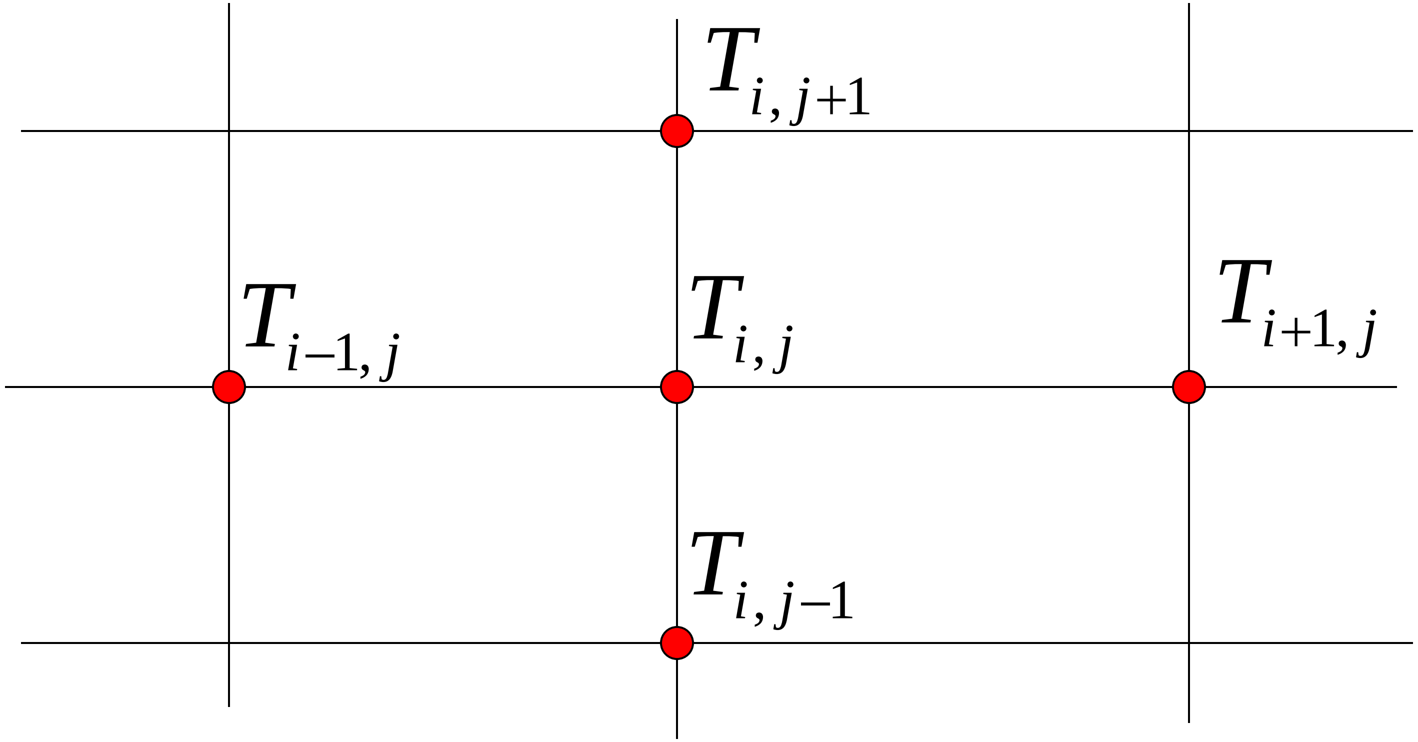
$$u(x,t+\delta) = z*\ u(x-h,t) + (1-2z)*u(x,t) + z*u(x+h,t)$$

- □ Change variable x to j*h, t to i*δ, and u(x,t) to u[j,i]

u[j,i+1]= z*u[j-1,i]+ (1-2*z)*u[j,i]+ z*u[j+1,i]

$$\frac{\partial^2 T(x,y)}{\partial x^2}+\frac{\partial^2 T(x,y)}{\partial y^2}=0$$

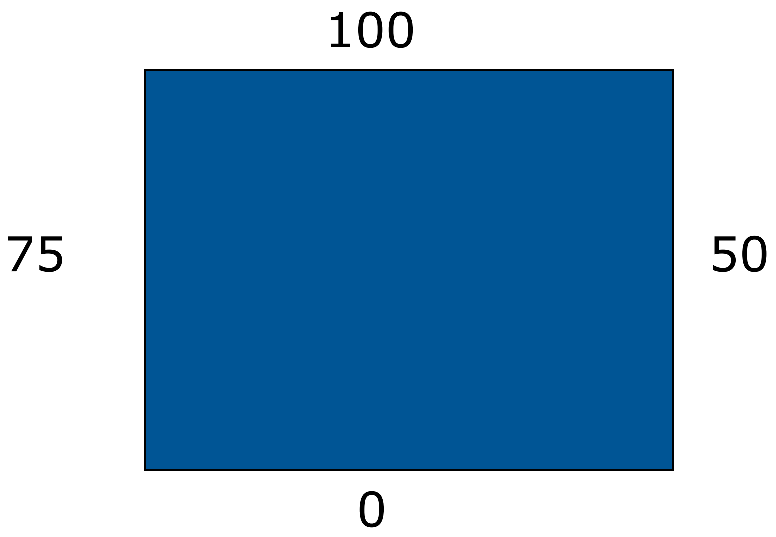
Solution Technique



$$\boxed{T_{i+1,j}+T_{i-1,j}+T_{i,j+1}+T_{i,j-1}-4T_{i,j}=0}$$

Example

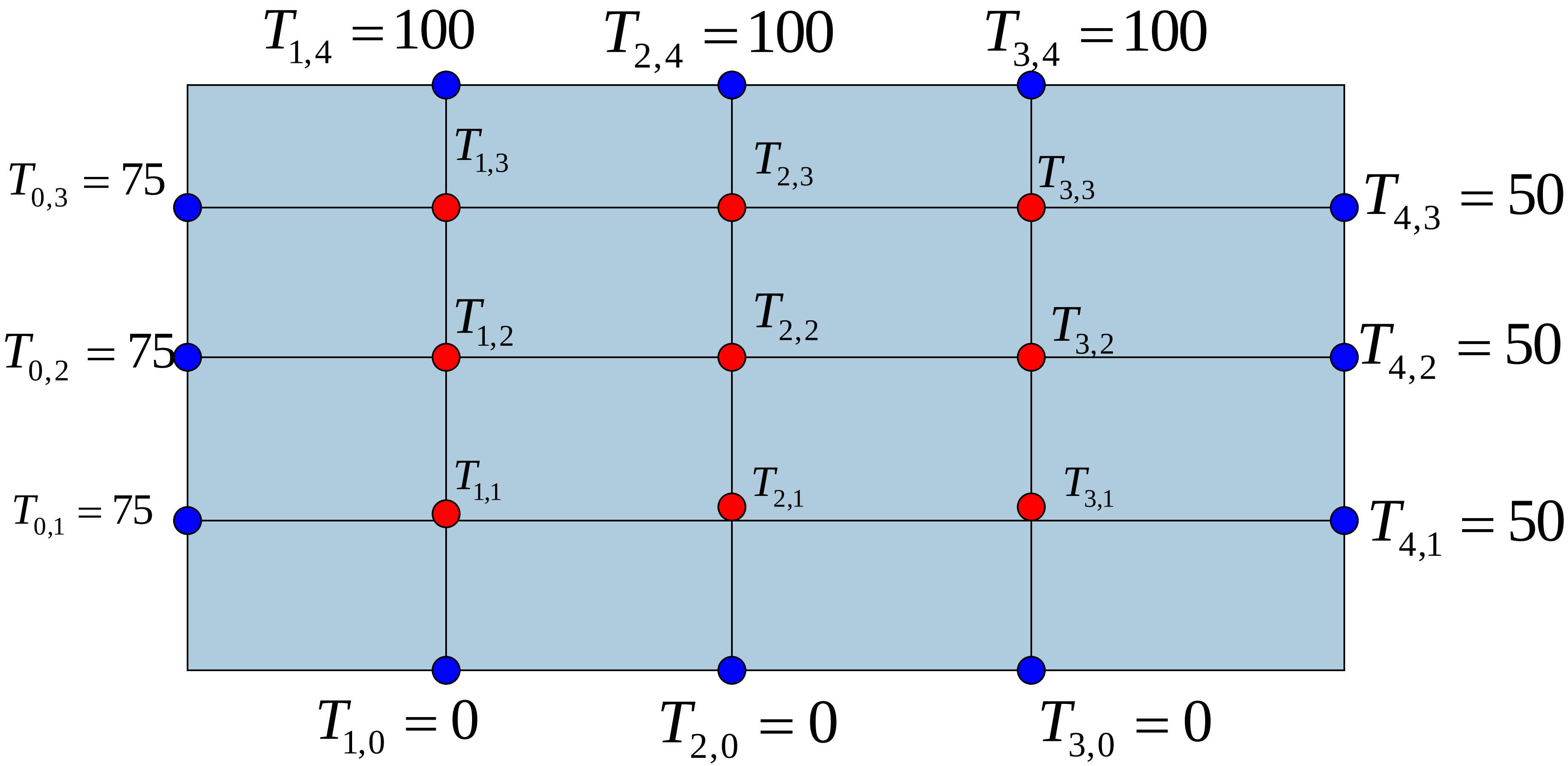
It is required to determine the steady state temperature at all points of a heated sheet of metal. The edges of the sheet are kept at a constant temperature: 100, 50, 0, and 75 degrees.



The sheet is divided to 5X5 grids.

$$\frac{\partial^2 T(x,y)}{\partial x^2}+\frac{\partial^2 T(x,y)}{\partial y^2}=0$$

Example



🔵 Known

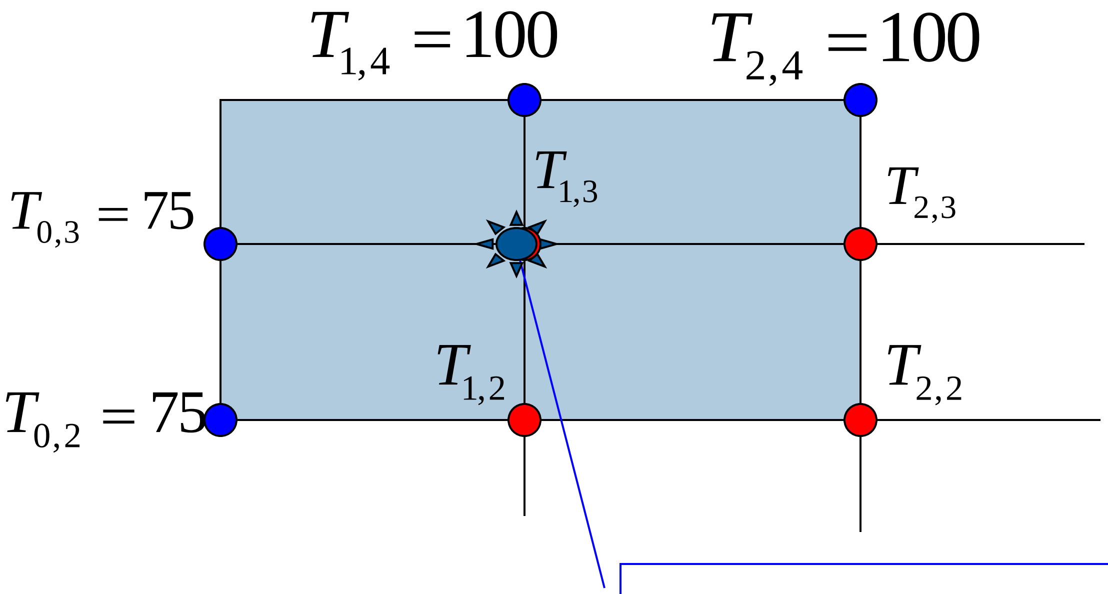
🔴 To be determined

First Equation

$$\frac{\partial^2 T(x,y)}{\partial x^2}+\frac{\partial^2 T(x,y)}{\partial y^2}=0$$

🔵 Known

🔴 To be determined



$$T_{0,3}+T_{1,4}+T_{1,2}+T_{2,3}-4T_{1,3}=0$$

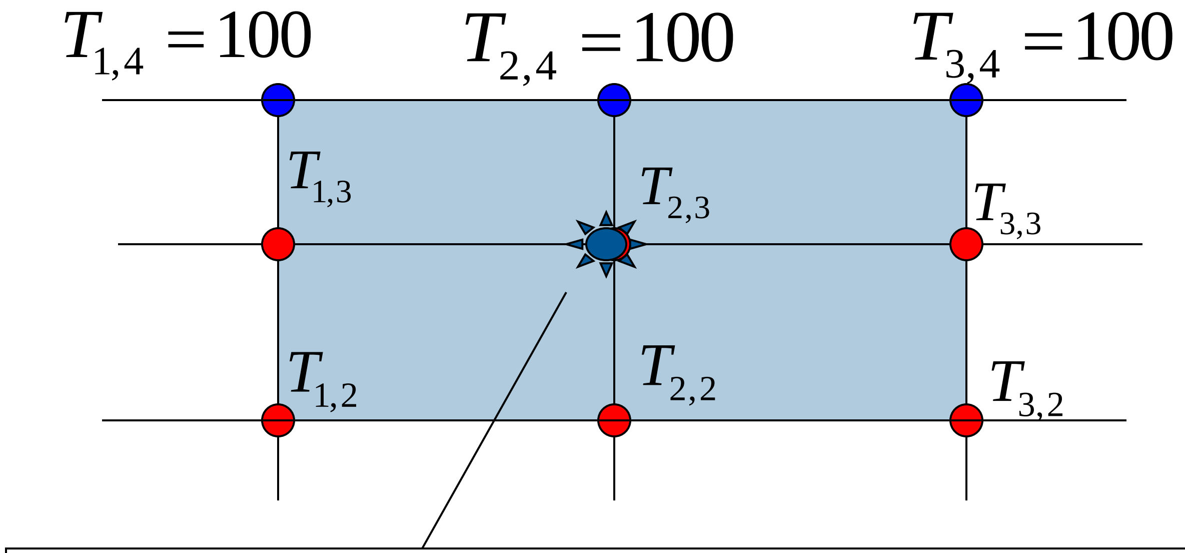
$$75+100+T_{1,2}+T_{2,3}-4T_{1,3}=0$$

Another Equation

$$\frac{\partial^2 T(x,y)}{\partial x^2}+\frac{\partial^2 T(x,y)}{\partial y^2}=0$$

● Known

● To be determined

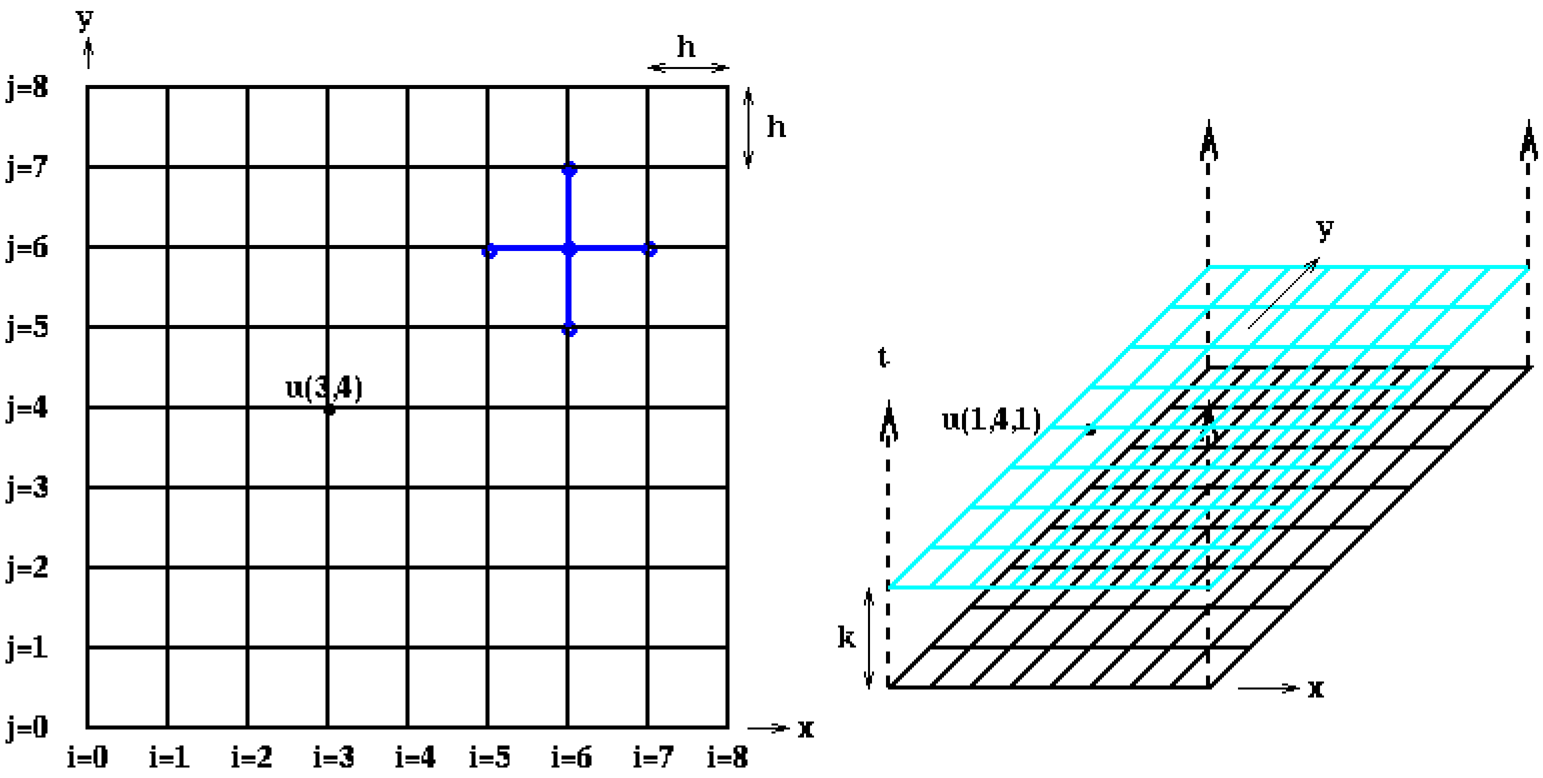


$$T_{1,3}+T_{2,4}+T_{3,3}+T_{2,2}-4T_{2,3}=0$$

$$T_{1,3}+100+T_{3,3}+T_{2,2}-4T_{2,3}=0$$

Another Mesh

Discretization of the 2D Heat Equation



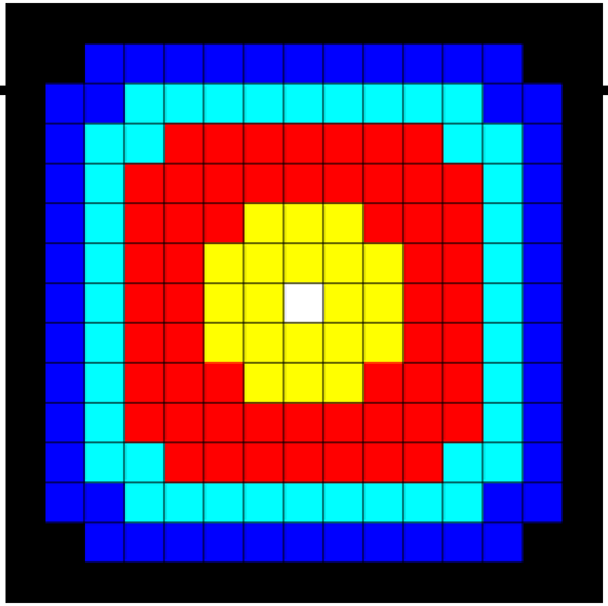
Solution

The Rest of the Equations

$$\frac{\partial^2 T(x,y)}{\partial x^2}+\frac{\partial^2 T(x,y)}{\partial y^2}=0$$

$$\begin{pmatrix} 4 & -1 & 0 & -1 & & & & & \\ -1 & 4 & -1 & 0 & -1 & & & & \\ 0 & -1 & 4 & 0 & 0 & -1 & & & \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & & \\ & -1 & 0 & -1 & 4 & -1 & 0 & -1 & \\ & & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ & & & -1 & 0 & 0 & 4 & -1 & 0 \\ & & & & -1 & 0 & -1 & 4 & -1 \\ & & & & & -1 & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} T_{1,1} \\ T_{2,1} \\ T_{3,1} \\ T_{1,2} \\ T_{2,2} \\ T_{3,2} \\ T_{1,3} \\ T_{2,3} \\ T_{3,3} \end{pmatrix} = \begin{pmatrix} 75 \\ 0 \\ 50 \\ 75 \\ 0 \\ 50 \\ 175 \\ 100 \\ 150 \end{pmatrix}$$

Another evolution equation



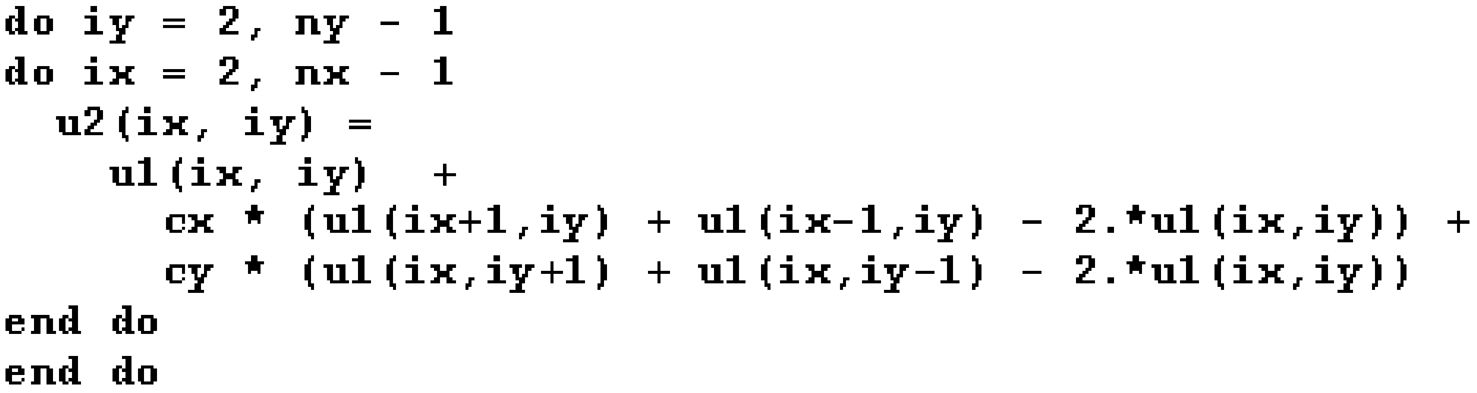
- The calculation of an element is dependent upon neighbour element values.

$$U_{x,y} = U_{x,y}$$

$$+ C_x * (U_{x+1,y} + U_{x-1,y} - 2 * U_{xy})$$

$$+ C_y * (U_{x,y+1} + U_{x,y-1} - 2 * U_{x,y})$$

- A serial progra like:



```
do iy = 2, ny - 1
do ix = 2, nx - 1
   u2(ix, iy) =
      u1(ix, iy)   +
         cx * (u1(ix+1,iy) + u1(ix-1,iy) - 2.*u1(ix,iy)) +
         cy * (u1(ix,iy+1) + u1(ix,iy-1) - 2.*u1(ix,iy))
end do
end do
```

Iterative Methods for Solving Ax = b

Ex:

$$\begin{array}{lrcl} (1) & 6x_1 - 2x_2 + x_3 & = & 11 \\ (2) & -2x_1 + 7x_2 + 2x_3 & = & 5 \\ (3) & x_1 + 2x_2 - 5x_3 & = & \text{-1} \end{array}$$

$\Longrightarrow$

$$\begin{aligned} x_1 &= \tfrac{11}{6} - \tfrac{1}{6}(-2x_2 + x_3) \\ x_2 &= \tfrac{5}{7} - \tfrac{1}{7}(-2x_1 + 2x_3) \\ x_3 &= \tfrac{1}{5} - \tfrac{1}{-5}(x_1 + 2x_2) \end{aligned}$$

$\Longrightarrow$

$$\begin{aligned} x_1^{(k+1)} &= \tfrac{1}{6}(11 - (-2x_2^{(k)} + x_3^{(k)})) \\ x_2^{(k+1)} &= \tfrac{1}{7}(5 - (-2x_1^{(k)} + 2x_3^{(k)})) \\ x_3^{(k+1)} &= \tfrac{1}{-5}(-1 - (x_1^{(k)} + 2x_2^{(k)})) \end{aligned}$$

Initial Approximation: $x_1 = 0, x_2 = 0, x_3 = 0$

Iter	0	1	2	3	4	$\cdots$	8
x_1	0	1.833	2.038	2.085	2.004	$\cdots$	2.000
x_2	0	0.714	1.181	1.053	1.001	$\cdots$	1.000
x_3	0	0.2	0.852	1.080	1.038	$\cdots$	1.000

Stop when $\| \vec{x}^{(k+1)} - \vec{x}^{(k)} \| < 10^{-4}$

Need to define **norm** $\| \vec{x}^{(k+1)} - \vec{x}^{(k)} \|$.

Jacobi Method for Ax = b

$$x_i^{k+1} = \frac{1}{a_{ii}}(b_i - \sum_{j\neq i} a_{ij}x_j^k) \quad i = 1, \cdots n$$

Example:

$$\begin{array}{llll} (1) & 6x_1 - 2x_2 + x_3 & = & 11 \\ (2) & -2x_1 + 7x_2 + 2x_3 & = & 5 \\ (3) & x_1 + 2x_2 - 5x_3 & = & \text{-1} \end{array}$$

$\Longrightarrow$

$$\begin{array}{lll} x_1 & = & \frac{11}{6} - \frac{1}{6}(-2x_2 + x_3) \\ x_2 & = & \frac{5}{7} - \frac{1}{7}(-2x_1 + 2x_3) \\ x_3 & = & \frac{1}{5} - \frac{1}{-5}(x_1 + 2x_2) \end{array}$$

Jacobi method in a matrix-vector form

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}^{k+1} = \begin{bmatrix} 0 & \frac{2}{6} & -\frac{1}{6} \\ \frac{2}{7} & 0 & -\frac{2}{7} \\ \frac{1}{5} & \frac{2}{5} & 0 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}^{k} + \begin{pmatrix} \frac{11}{6} \\ \frac{5}{7} \\ \frac{1}{5} \end{pmatrix}$$

Parallel

$x^{k+1} =$

or in general

x^{k+

Parallel solution:

- Distribute rows
- Perform comput owner-computes
- Perform all-gath after each iterat

SPMD Code for $y = H * x + d$

```
me=mynode();
for $i = 1$ to $n$ do
    if $proc\_map(i) == me$, then do $S_i$:
        $S_i$ :    $y[i] = 0$;
            $i1 = Local(i)$
            for $j = 1$ to $n$ do
                $y[i1] = y[i1] + a[i1][j] * x[j] + d[i1]$
            endfor
endfor
```

Optimized solution: write the Jacobi method as:

```
Repeat
    For $i$= 1 to $n$
        $y_i^{new} = 0.5(y_{i-1}^{old} + y_{i+1}^{old} - h^2)$
    Endfor
Until $\| \vec{y}^{new} - \vec{y}^{old} \| < \varepsilon$
```

The SOR method

SOR (Successive Over Relaxation).

The rate of convergence can be improved (accelerated) by the SOR method:

Step 1: Use the Gauss-Seidel Method.

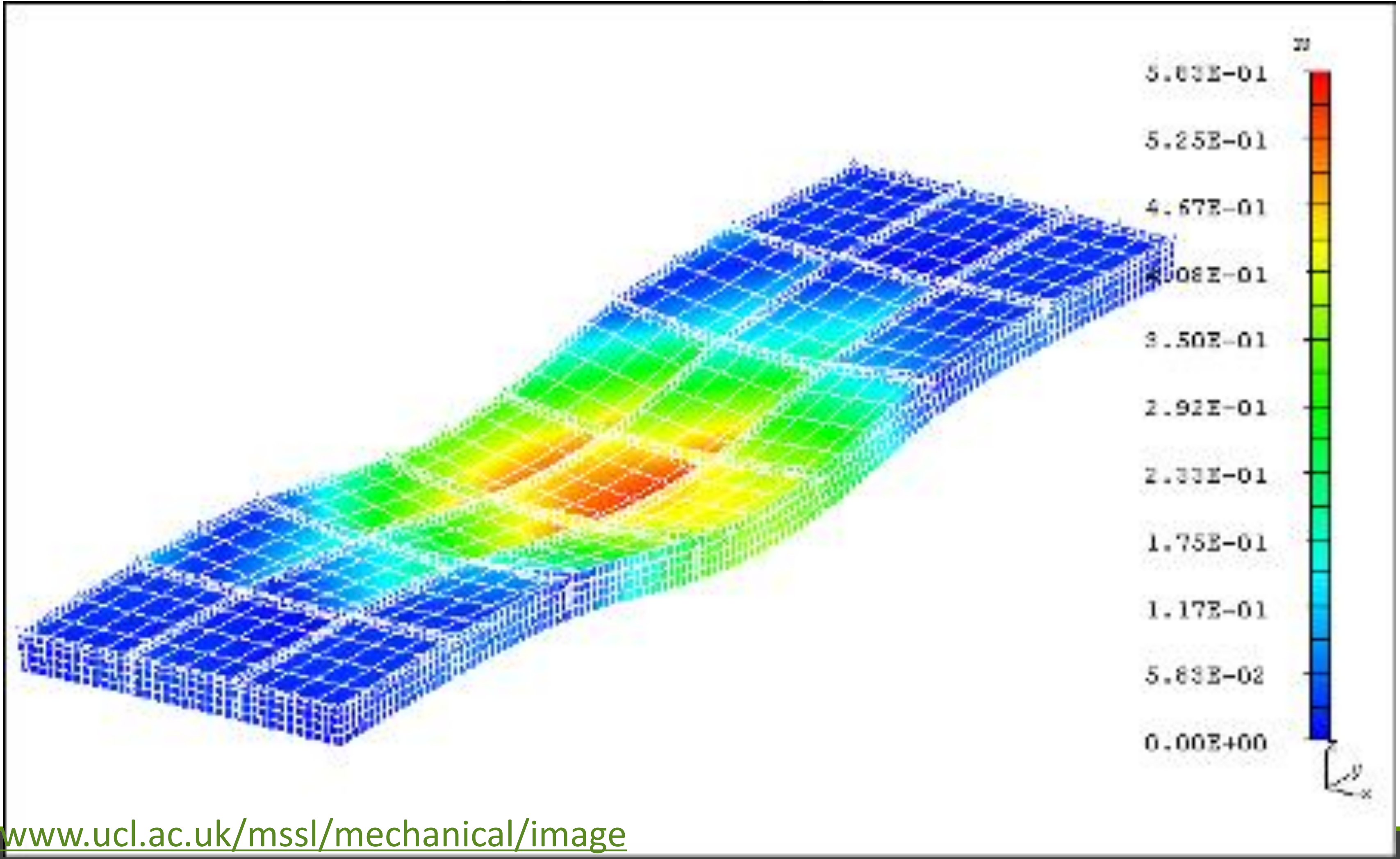
$$x^{k+1} = Hx^k + d$$

Step 2:

$$x^{k+1} = x^k + w(x^{k+1} - x^k)$$

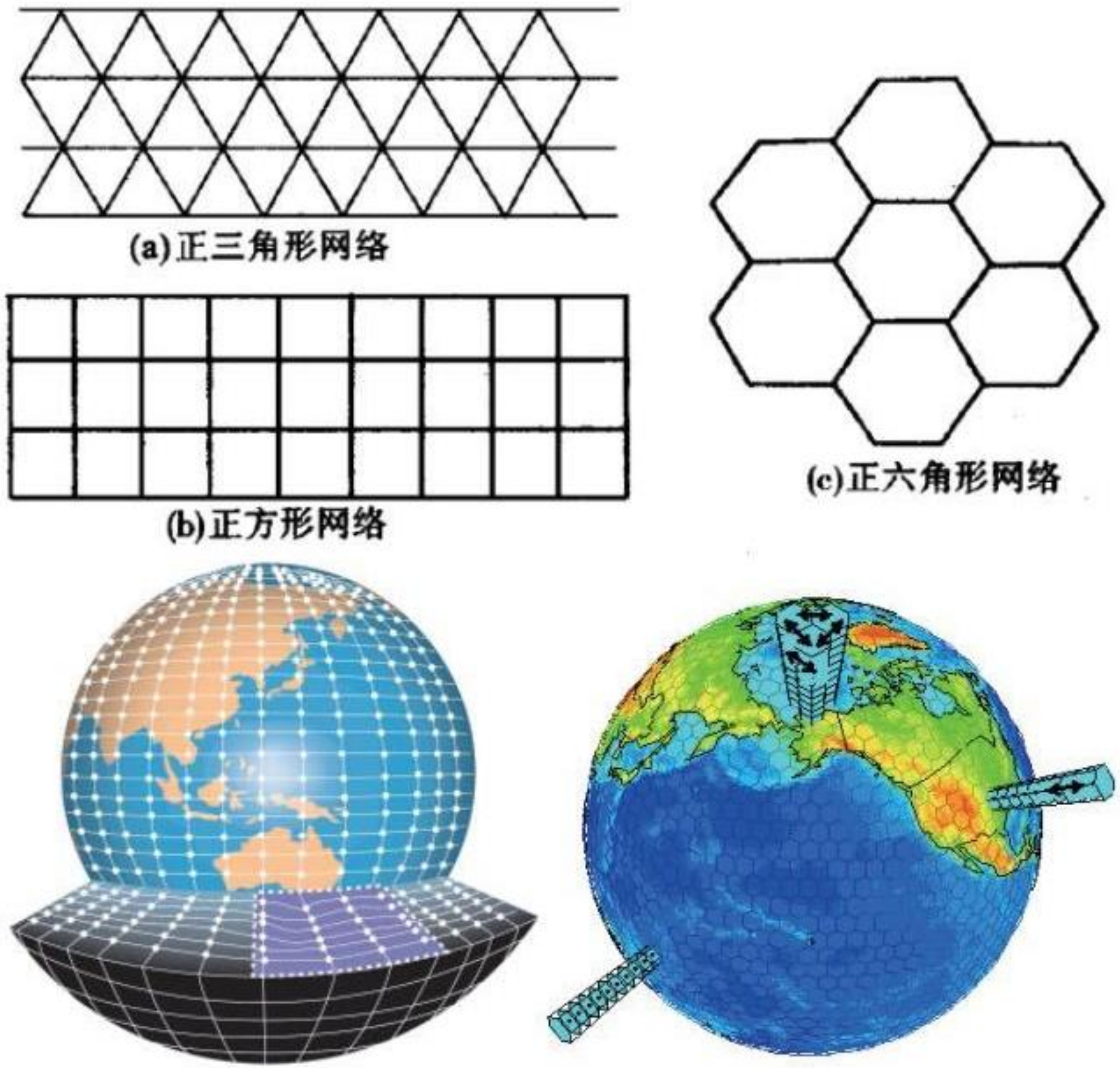
Weather forecasting? – FEM with similar idea

- □ **FEM: Finite Element Model [有限元分析]**



https://www.ucl.ac.uk/mssl/mechanical/images/analysis/Mode02A.png

We need Super Computer/HPC

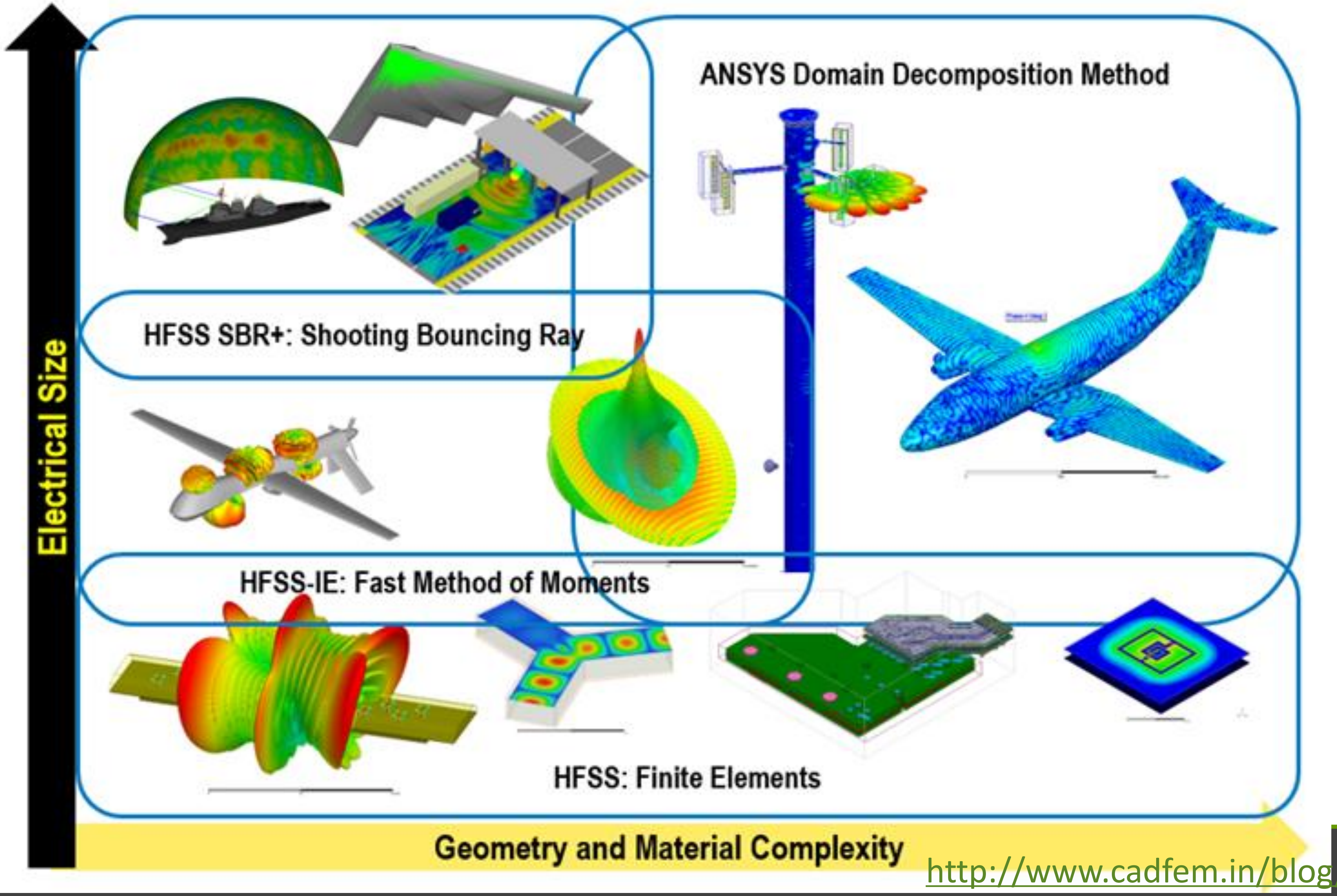


(a)正三角形网络

(b)正方形网络

(c)正六角形网络

How to model is the KEY


Electrical Size
ANSYS Domain Decomposition Method
HFSS SBR+: Shooting Bouncing Ray
HFSS-IE: Fast Method of Moments
HFSS: Finite Elements
Geometry and Material Complexity

http://www.cadfem.in/blog/

Global Climate Modeling Problem

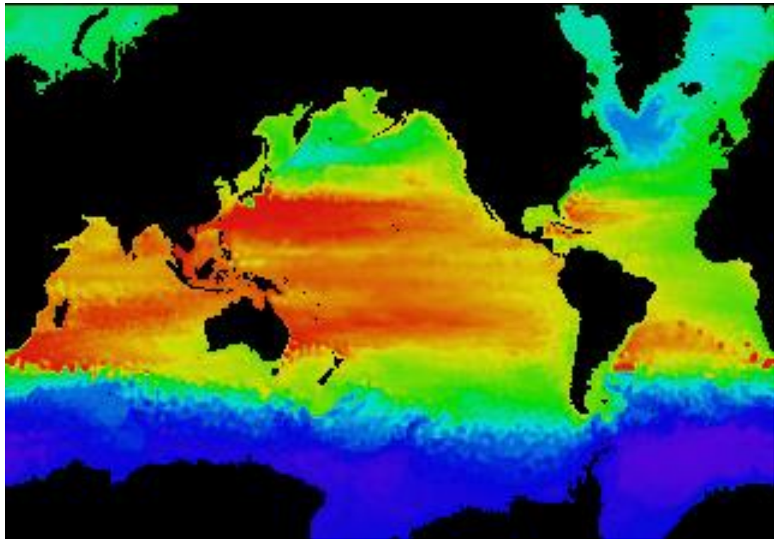
- Problem is to compute:

  f(latitude, longitude, elevation, time) → "weather" =

  (temperature, pressure, humidity, wind velocity)

- Approach:
  - *Discretize* the domain, e.g., a measurement point every 10 km
  - Devise an algorithm to predict weather at time t+δt given t

- Uses:
  - Predict major events, e.g., El Nino
  - Use in setting air emissions standards
  - Evaluate global warming scenarios



Source: http://www.epm.ornl.gov/chammp/chammp.html

Global Grid (0.25*0.25 → 1440X720 dots = 62208000)

- If we take resolution as 0.25*0.25, and 60 vertical layers, the matrix size will be 1440*720*60 = $6.22*10^7$.
- If 6 variables, the number is 6*$6.22*10^7$.
- If we want to predict the weather in 24 hours, and 5 minutes as span, we should compute values of
  - 288*6*$6.22*10^7$.≈$1.08*10^{11}$.
  - If you want to predict future 3 days, 5 days, 10 days?

- So HUGE! We need more powerful computers!
  - Super Computers, HPC etc.

https://www.esrl.noaa.gov/psd/

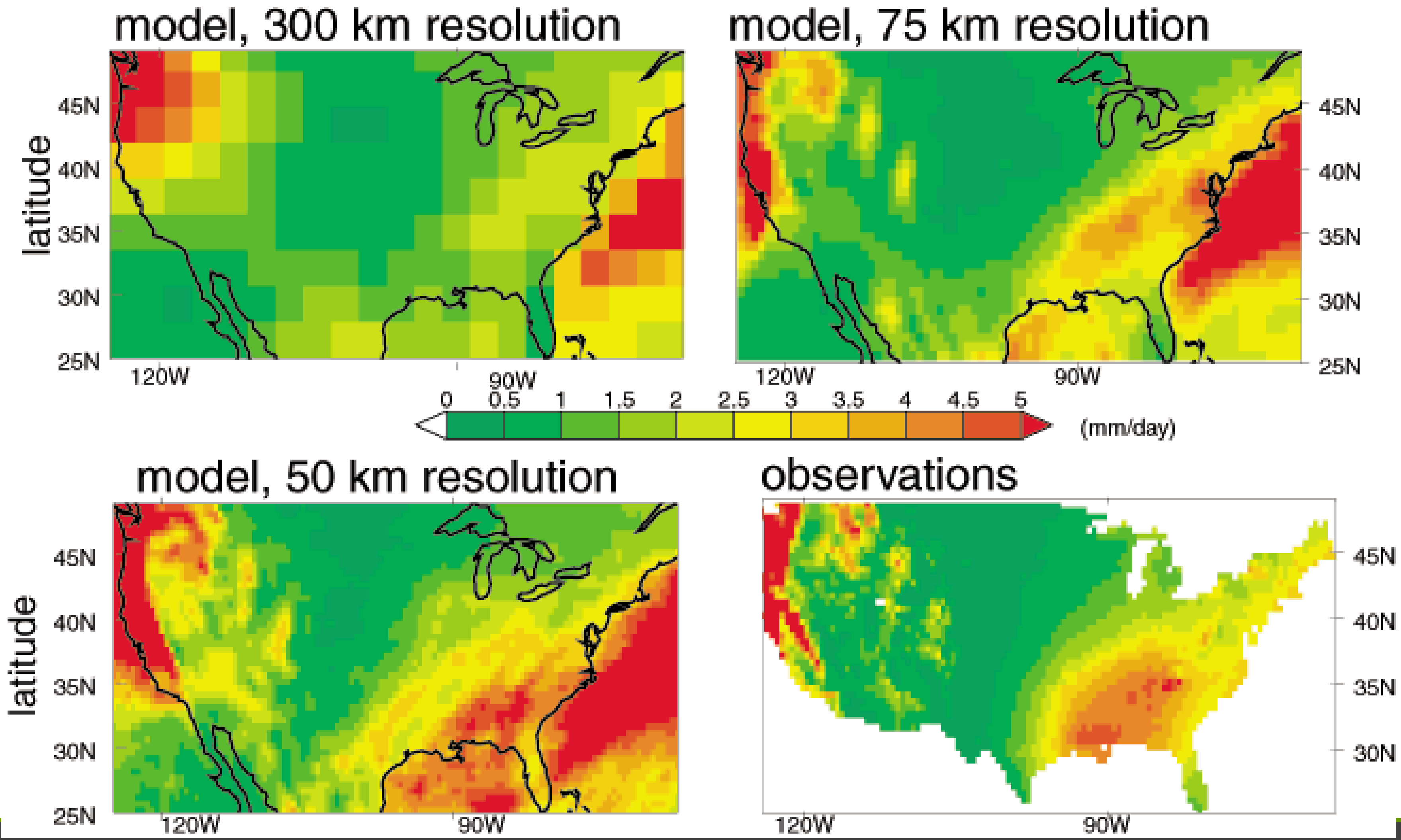
Global Climate Modeling Computation

- One piece is modeling the fluid flow in the atmosphere
  - Solve Navier-Stokes equations
    - Roughly 100 Flops per grid point with 1 minute timestep
- Computational requirements:
  - To match real-time, need 5 x 10^{11} flops in 60 seconds = 8 Gflop/s
  - Weather prediction (7 days in 24 hours) → 56 Gflop/s
  - Climate prediction (50 years in 30 days) → 4.8 Tflop/s
  - To use in policy negotiations (50 years in 12 hours) → 288 Tflop/s
- To double the grid resolution, computation is 8x to 16x
- State of the art models require integration of atmosphere, clouds, ocean, sea-ice, land models, plus possibly carbon cycle, geochemistry and more
- Current models are coarser than this

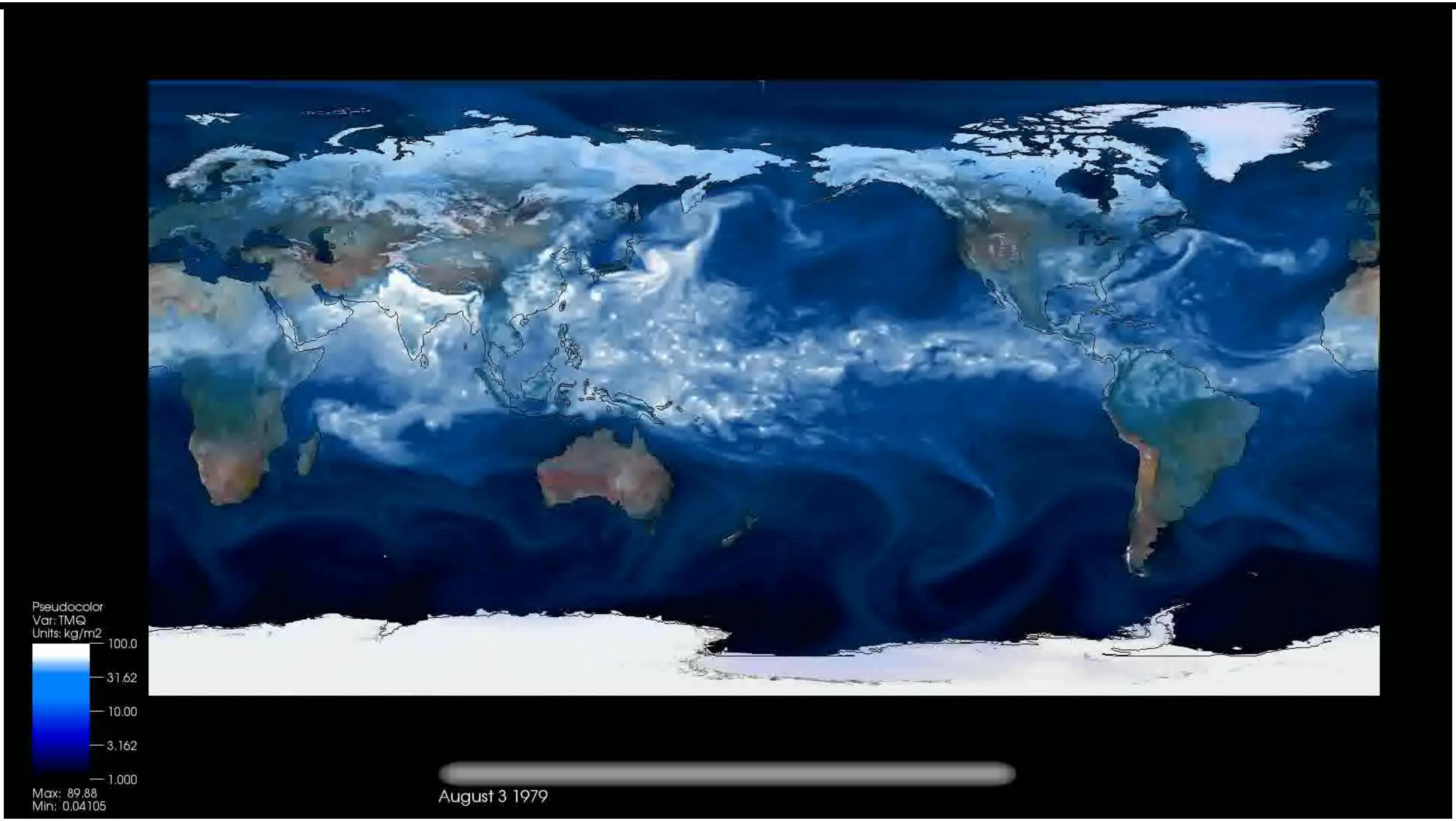
High Resolution Climate Modeling on NERSC-3 – P. Duffy, et al., LLNL

Wintertime Precipitation

As model resolution becomes finer, results converge towards observations

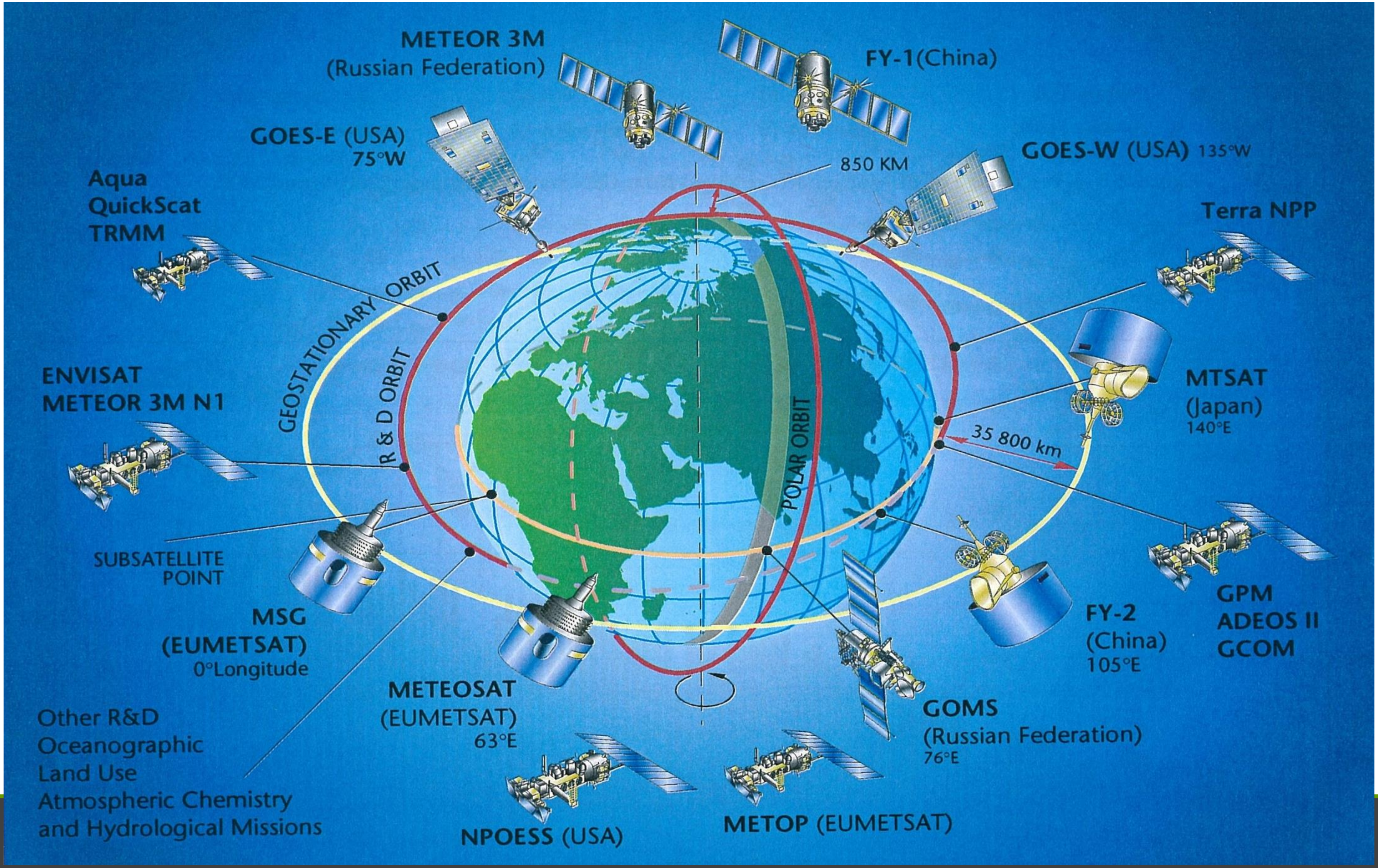


U.S.A. Hurricane

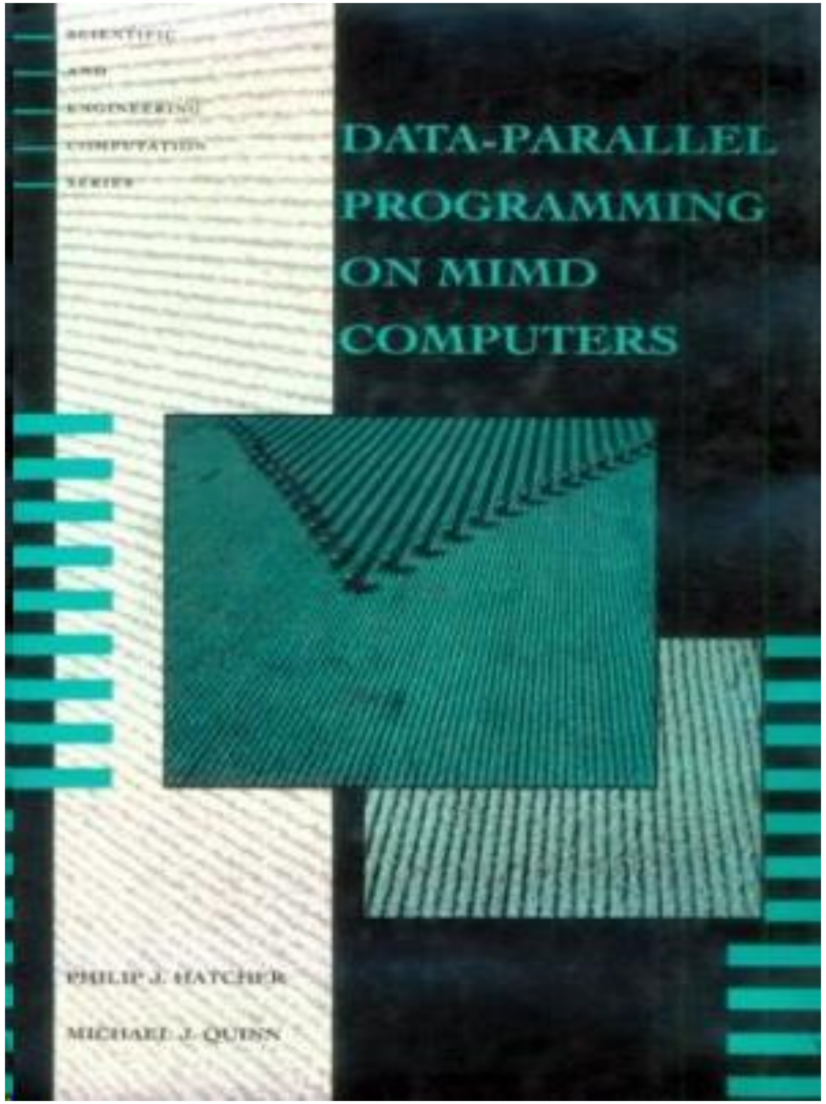


Source: Data from M.Wehner, visualization by Prabhat, LBNL

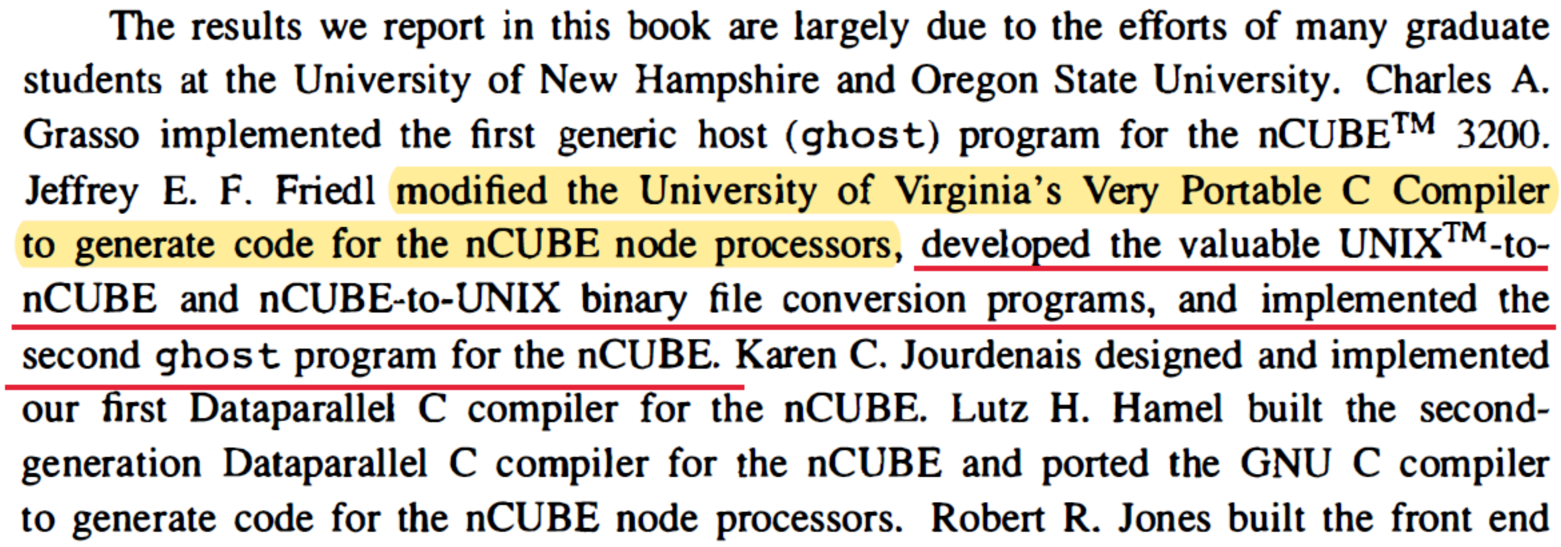
Meteorological Satellites



Michael J. Quinn



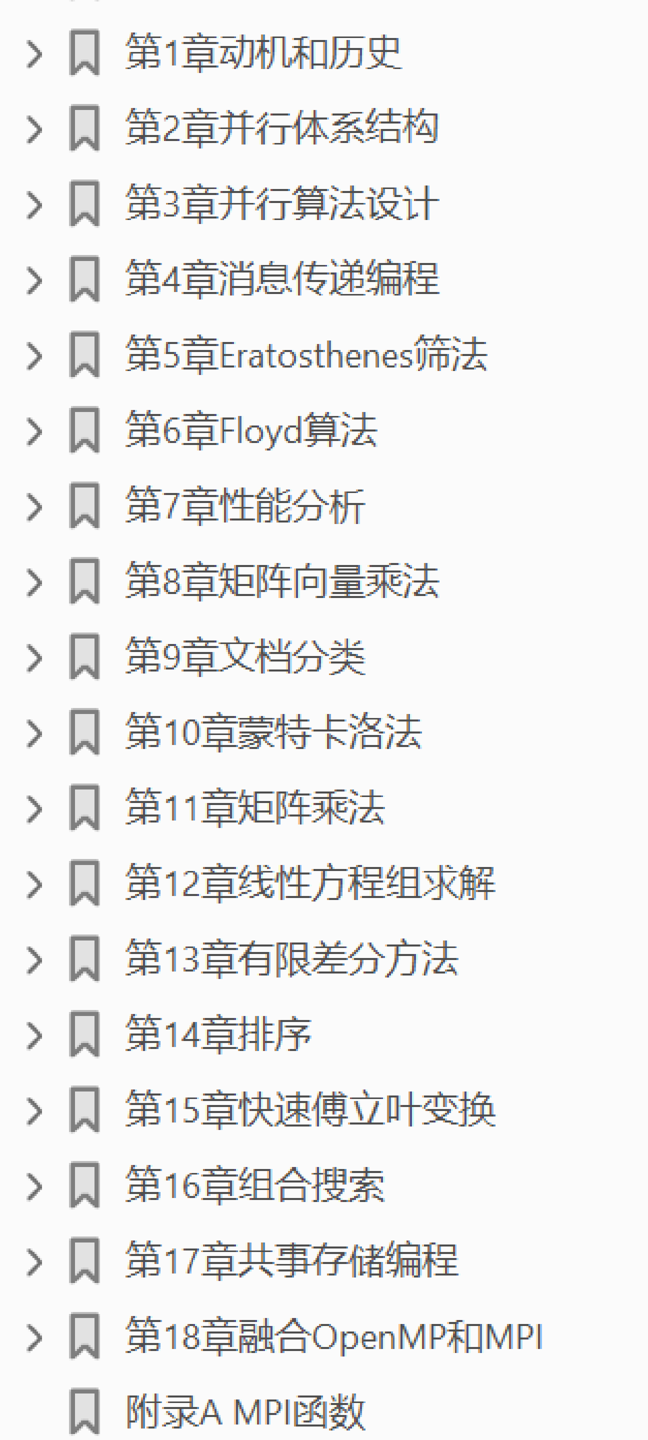
- Data-Parallel Programming on MIMD Computers
- Philip J. Hatcher, Michael J. Quinn
- **1991**



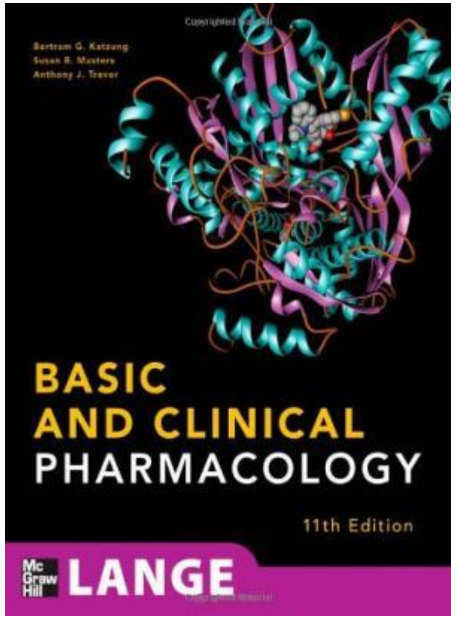
The results we report in this book are largely due to the efforts of many graduate students at the University of New Hampshire and Oregon State University. Charles A. Grasso implemented the first generic host (`ghost`) program for the nCUBE^{TM} 3200. Jeffrey E. F. Friedl modified the University of Virginia's Very Portable C Compiler to generate code for the nCUBE node processors, developed the valuable UNIX^{TM}-to-nCUBE and nCUBE-to-UNIX binary file conversion programs, and implemented the second `ghost` program for the nCUBE. Karen C. Jourdenais designed and implemented our first Dataparallel C compiler for the nCUBE. Lutz H. Hamel built the second-generation Dataparallel C compiler for the nCUBE and ported the GNU C compiler to generate code for the nCUBE node processors. Robert R. Jones built the front end

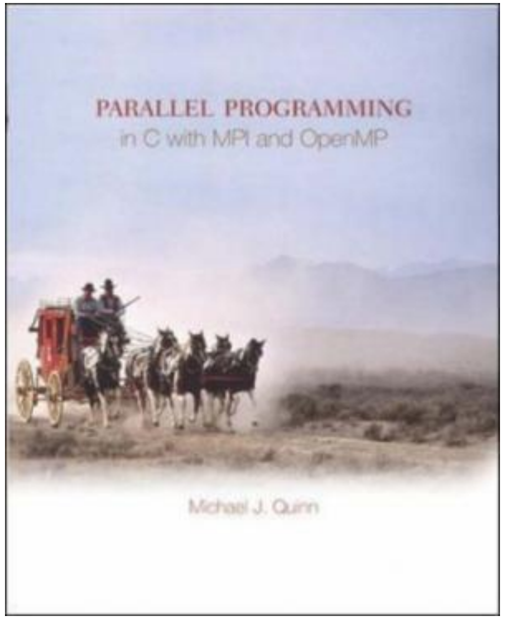
那时候的学生呀！现在，大数据开源，我们也应该引入高校，让学生改着玩！

Michael J. Quinn

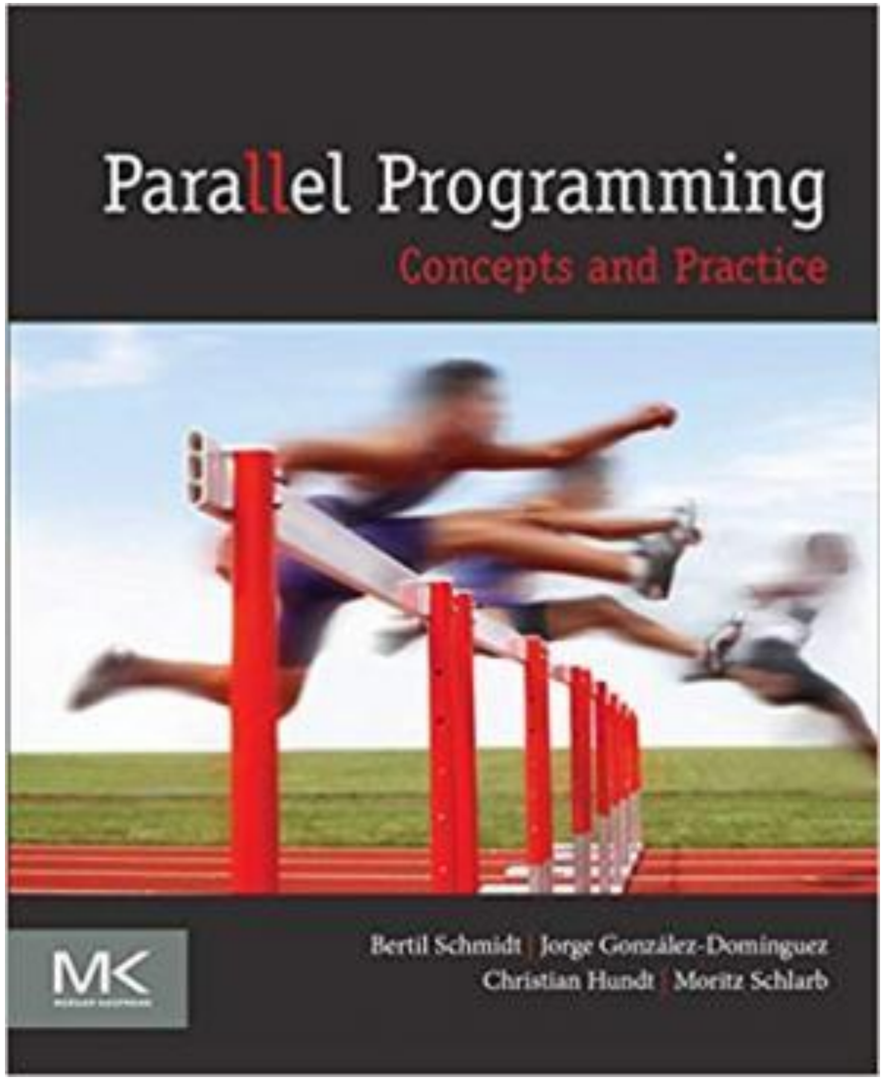


- Parallel programming in C with MPI and Open MP
- *Michael J Quinn*
- **2003**

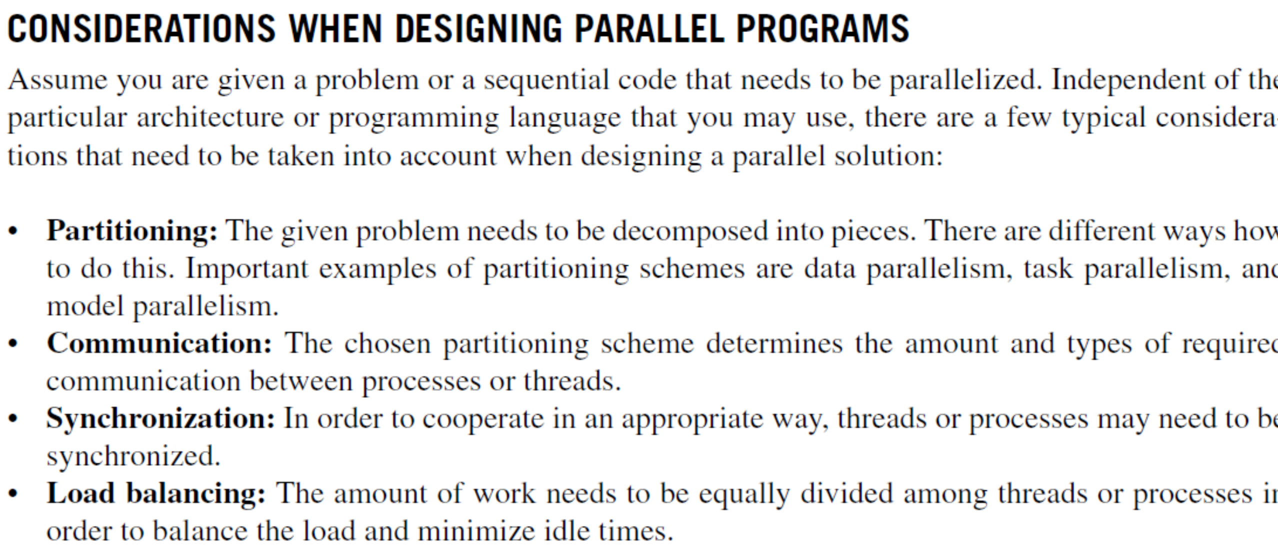




2021年5月27日20:31:47 这书也还不错


Parallel Programming
Concepts and Practice
MK
Bertil Schmidt | Jorge González-Domínguez
Christian Hundt | Moritz Schlarb

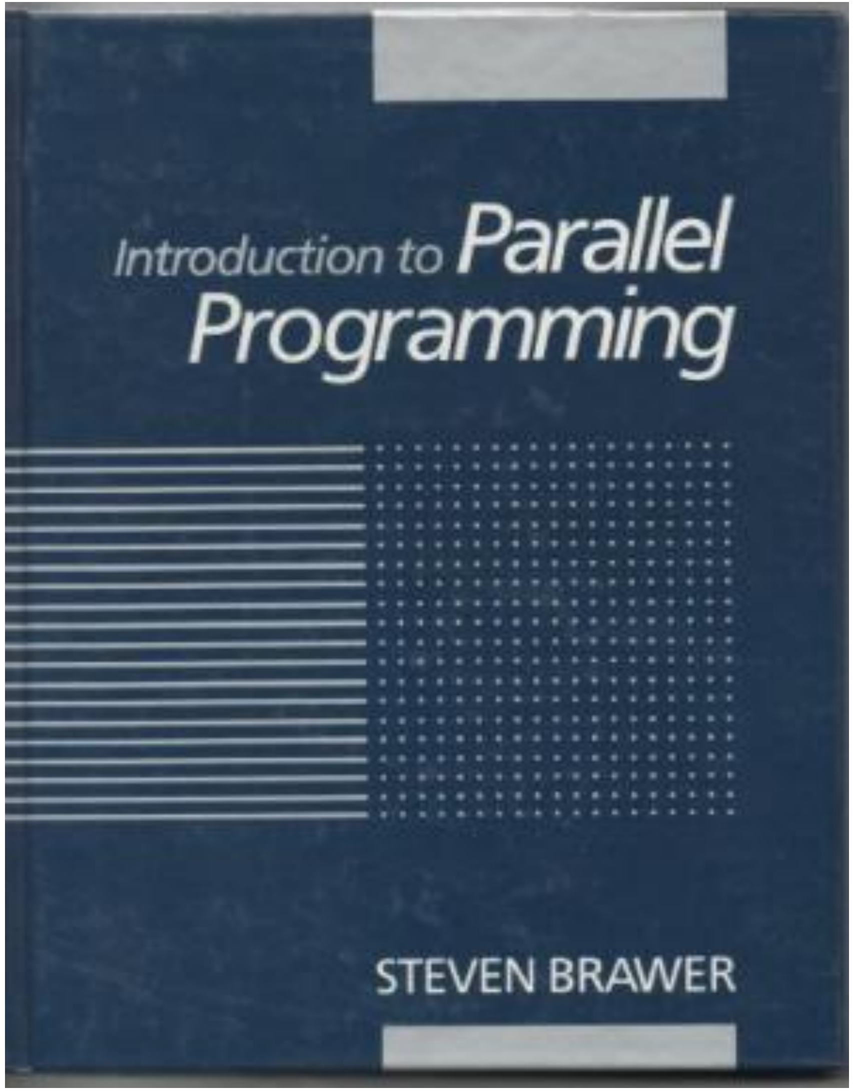
- Parallel Programming: Concepts and Practice
- *Bertil Schmidt*, *Jorge Gonzalez-Dominguez*, *Christian Hundt*, *Moritz Schlarb*
- **2017**



CONSIDERATIONS WHEN DESIGNING PARALLEL PROGRAMS

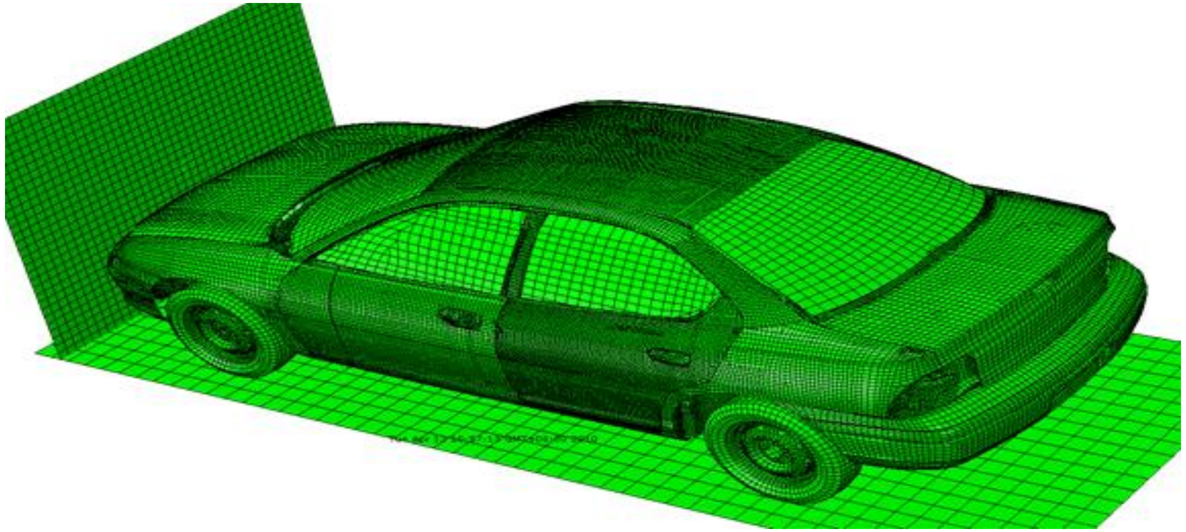
Assume you are given a problem or a sequential code that needs to be parallelized. Independent of the particular architecture or programming language that you may use, there are a few typical considerations that need to be taken into account when designing a parallel solution:

- **Partitioning:** The given problem needs to be decomposed into pieces. There are different ways how to do this. Important examples of partitioning schemes are data parallelism, task parallelism, and model parallelism.
- **Communication:** The chosen partitioning scheme determines the amount and types of required communication between processes or threads.
- **Synchronization:** In order to cooperate in an appropriate way, threads or processes may need to be synchronized.
- **Load balancing:** The amount of work needs to be equally divided among threads or processes in order to balance the load and minimize idle times.



- □ Introduction to Parallel Programming
- □ *Steven Brawer (Auth.)*
- □ **1989**

案例1：某汽车设计研究院整车碰撞分析



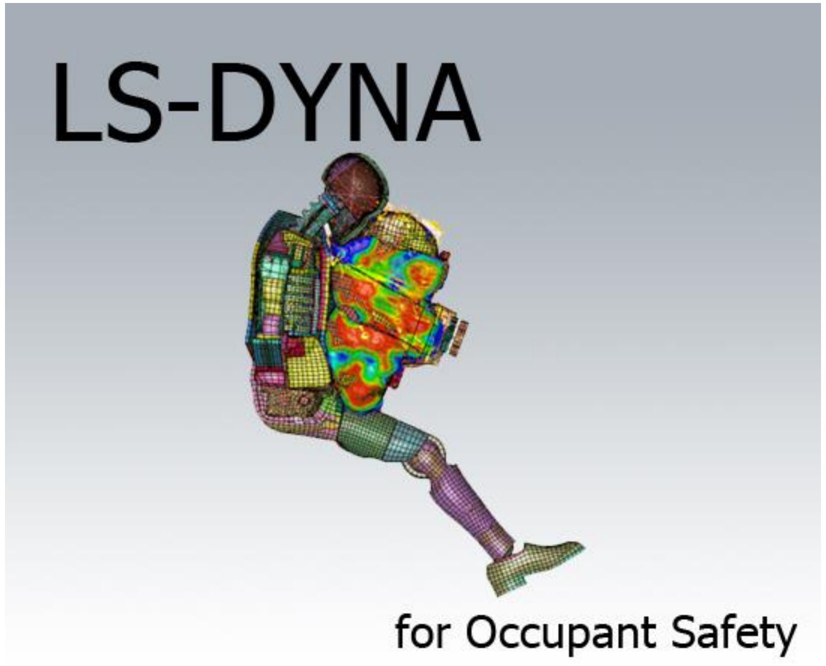
正碰仿真模型

- ■128核大规模并行、网格规模（千万级网格）；
- ■ Ls-dyna、Abaqus；
- ■仿真应用：汽车碰撞仿真，内容包括正碰、侧碰、偏置碰、多车追尾碰撞等；
- ■将原本在PC机上需要运行2-3天的单个计算模型的计算时间成功的缩短到了几个小时；

LS-DYNA

https://www.lstc.com/products/ls-dyna

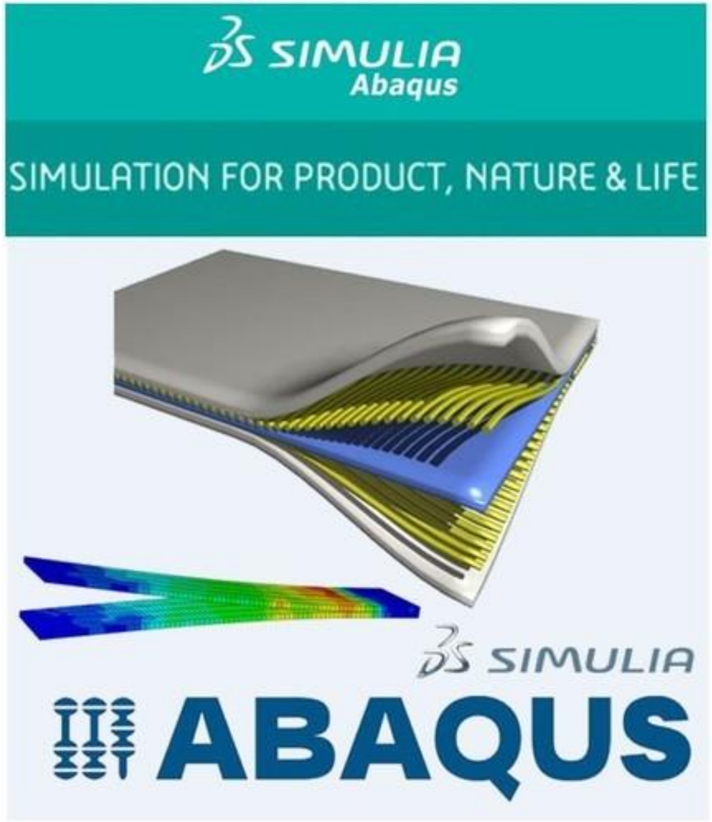
- LS-DYNA is a general-purpose finite element program capable of simulating complex real world problems. It is used by the automobile, aerospace, construction, military, manufacturing, and bioengineering industries. LS-DYNA is optimized for shared and distributed memory Unix, Linux, and Windows based, platforms, and it is fully QA'd by LSTC. The code's origins lie in highly nonlinear, transient dynamic finite element analysis using explicit time integration.



Abaqus

https://www.3ds.com/products-services/simulia/products/abaqus/

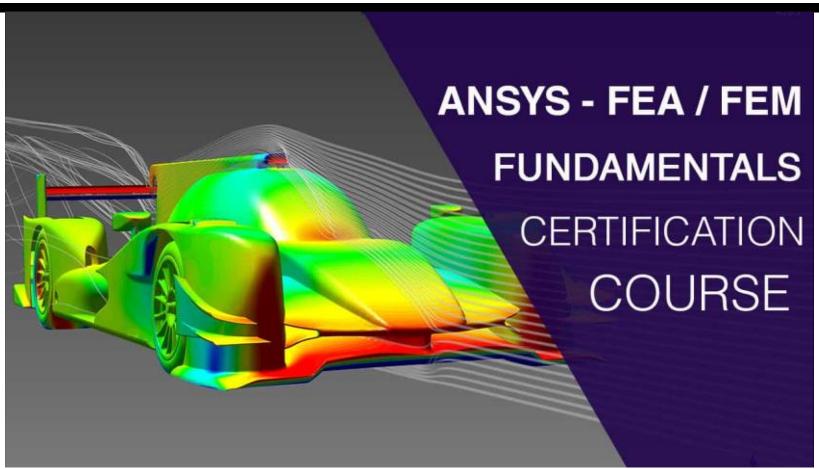
- The Abaqus Unified FEA product suite offers powerful and complete solutions for both routine and sophisticated engineering problems covering a vast spectrum of industrial applications. In the automotive industry engineering work groups are able to consider full vehicle loads, dynamic vibration, multibody systems, impact/crash, nonlinear static, thermal coupling, and acoustic-structural coupling using a common model data structure and integrated solver technology.

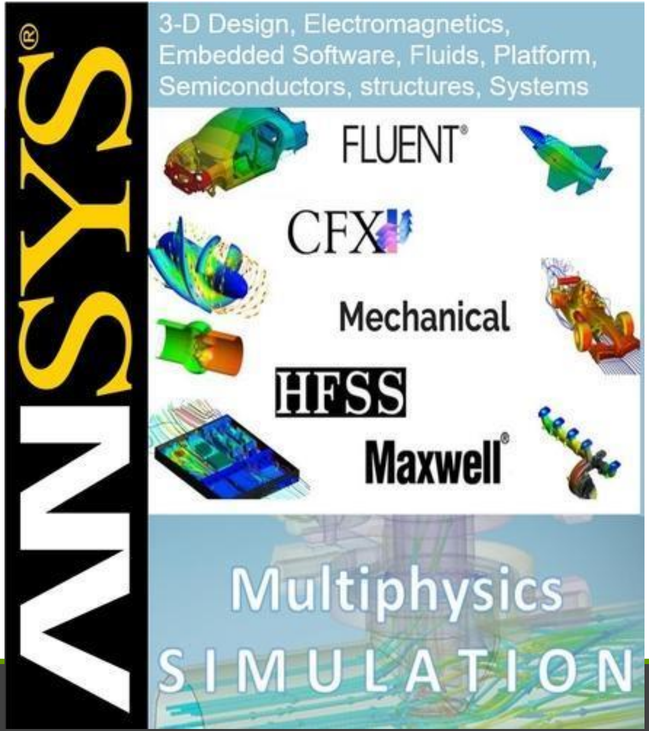


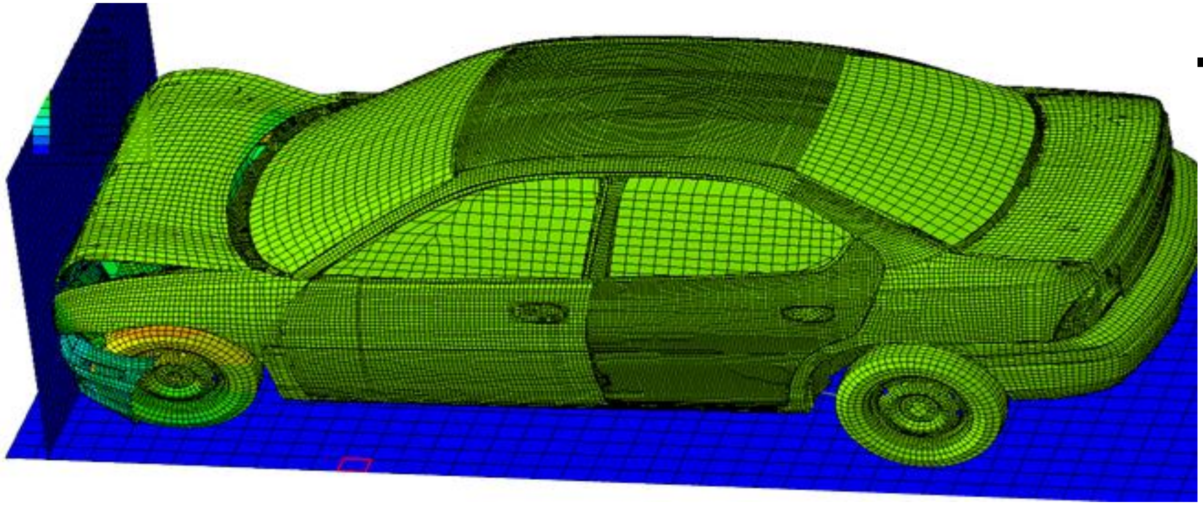
Ansys

https://www.ansys.com/about-ansys

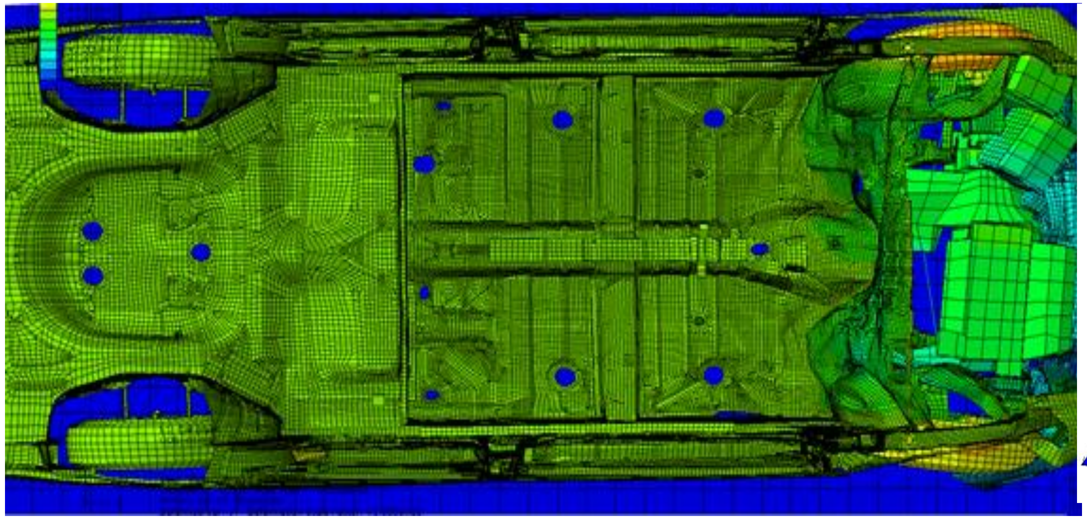
- If you’ve ever seen a rocket launch, flown on an airplane, driven a car, used a computer, touched a mobile device, crossed a bridge, or put on wearable technology, chances are you’ve used a product where ANSYS software played a critical role in its creation. ANSYS is the global leader in engineering simulation. We help the world’s most innovative companies deliver radically better products to their customers

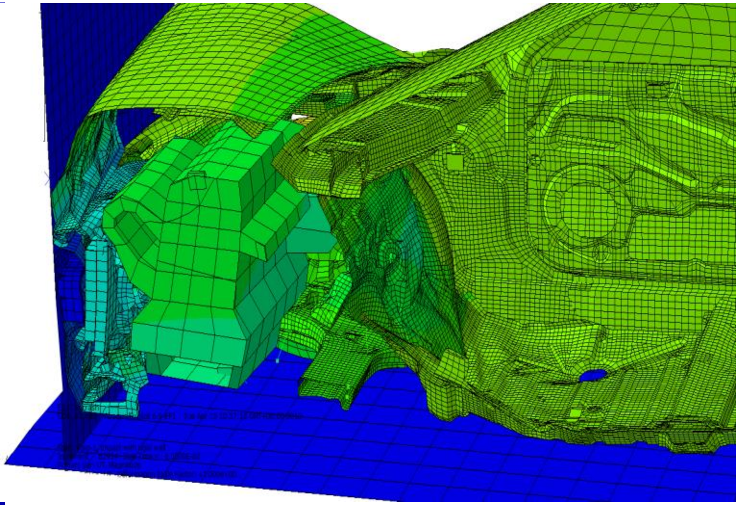



ANSYS®
3-D Design, Electromagnetics, Embedded Software, Fluids, Platform, Semiconductors, structures, Systems
FLUENT®
CFX®
Mechanical
HFSS
Maxwell®
Multiphysics
SIMULATION



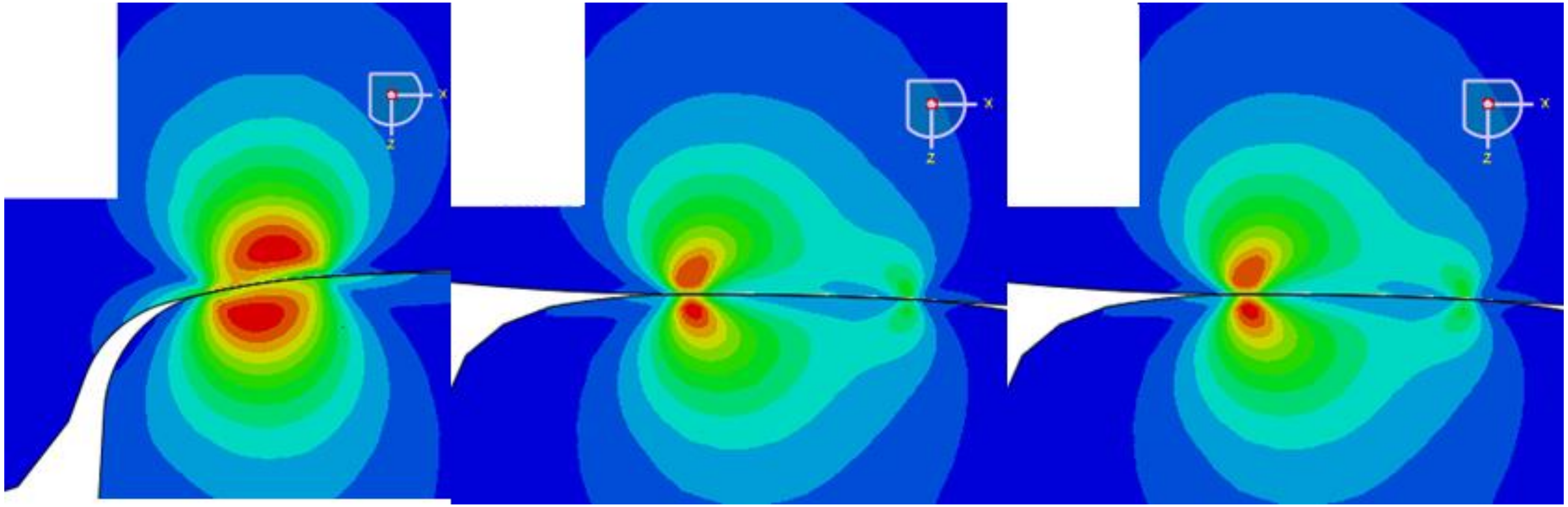
仿真结果：整车碰撞





仿真结果：车体内部零件损伤变形

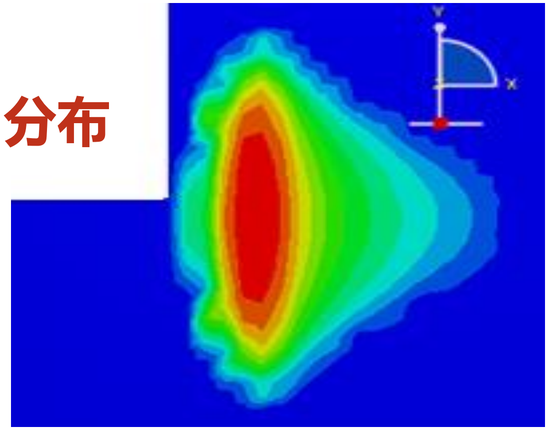
案例2：铁科院轮轨接触优化设计



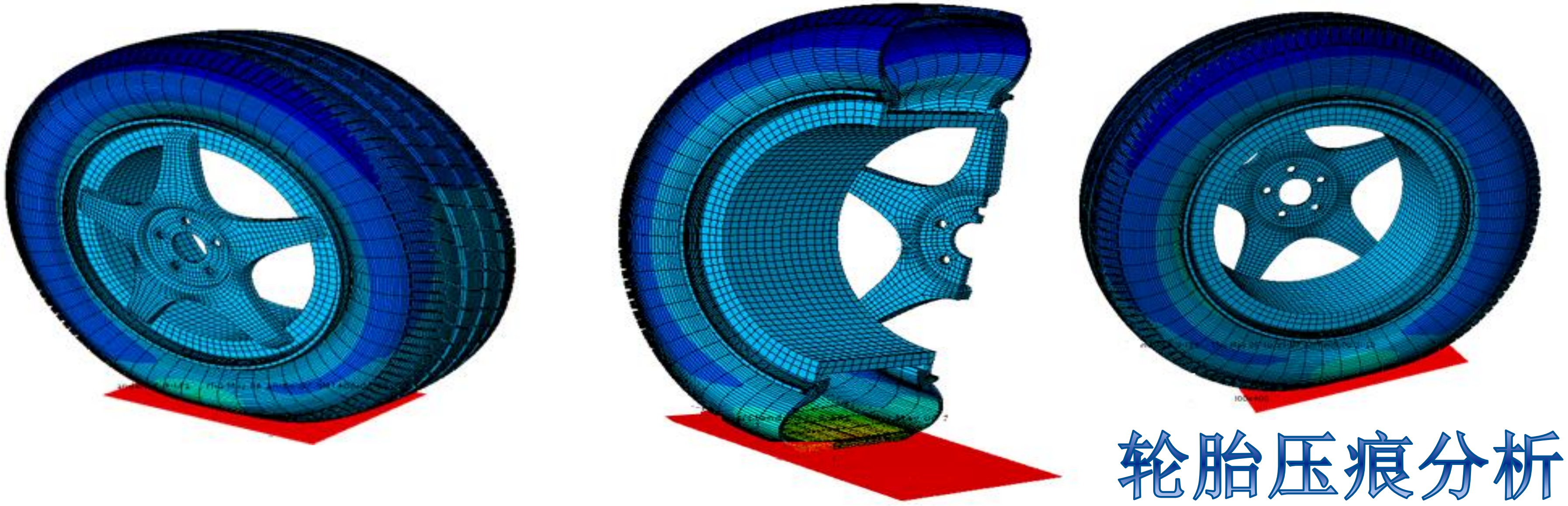
接触应力云图

■128核大规模并行、网格规模（千万级网格）；
■Abaqus；
■完成数十个仿真模型计算，作业规模从百万网格
到上亿网格不等；

接触压力分布

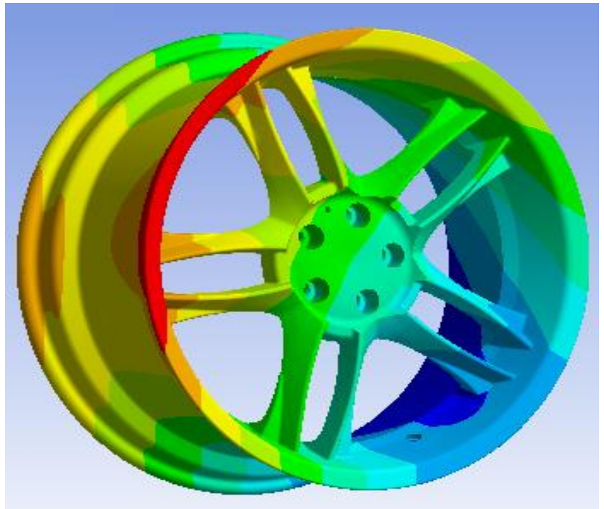


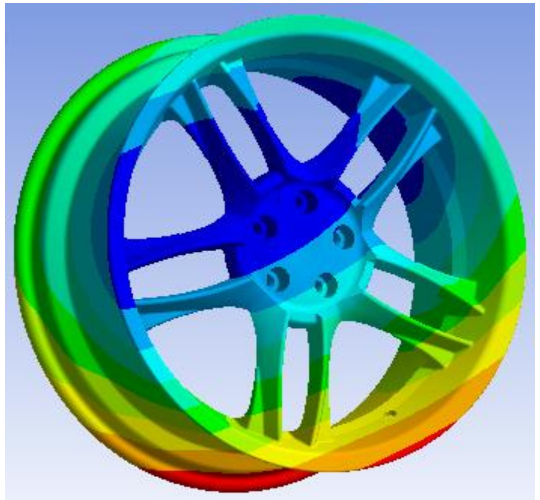
案例3：某汽车设计研究院汽车轮胎性能优化设计

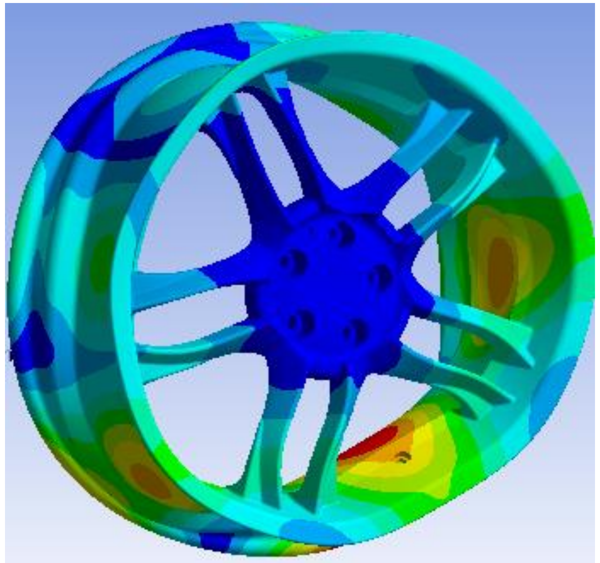


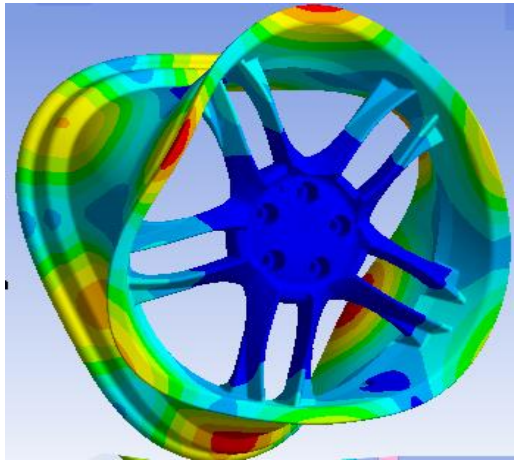
轮胎压痕分析

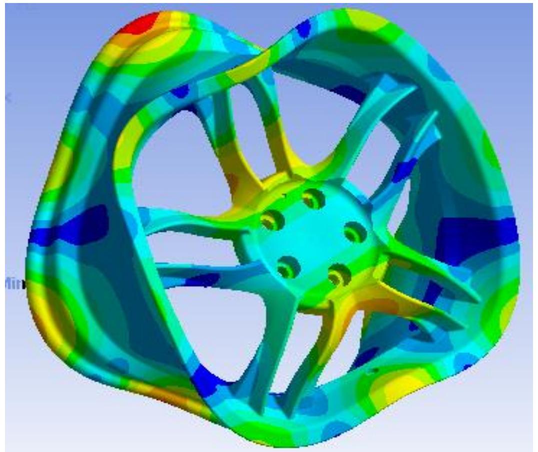
- 在工业云平台上完成汽车轮毂模态提取、汽车轮毂强度校核、轮胎压痕分析
- 128核大规模并行；
- Abaqus、Ansys；
- 整个仿真项目历时数月，先后提交23个计算模型，完成数十个仿真模型计算，作业规模从百万网格到千万级网格不等；









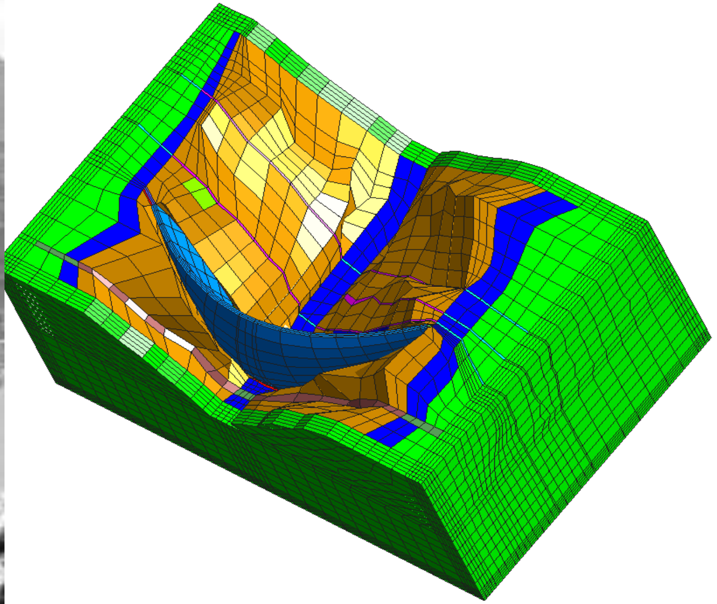


汽车轮毂模态提取

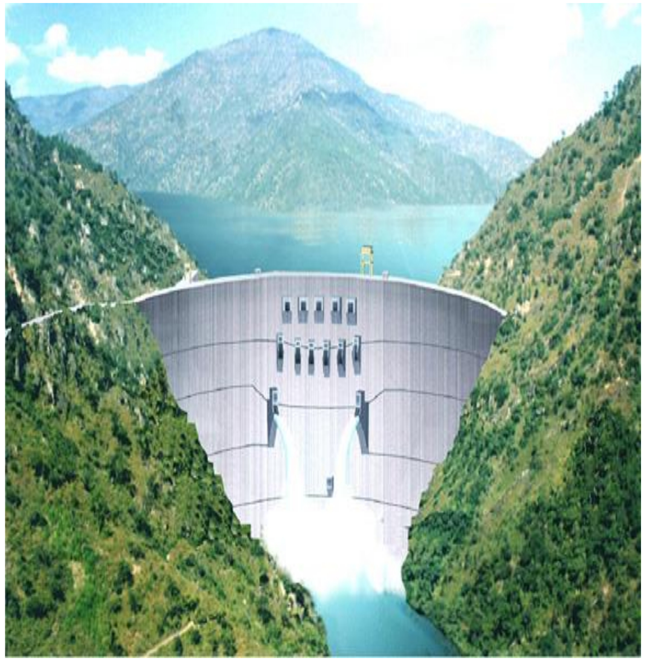
- Ansys
- 网格规模百万级网格

案例4：小湾拱坝地震响应分析

小湾拱坝实景图片

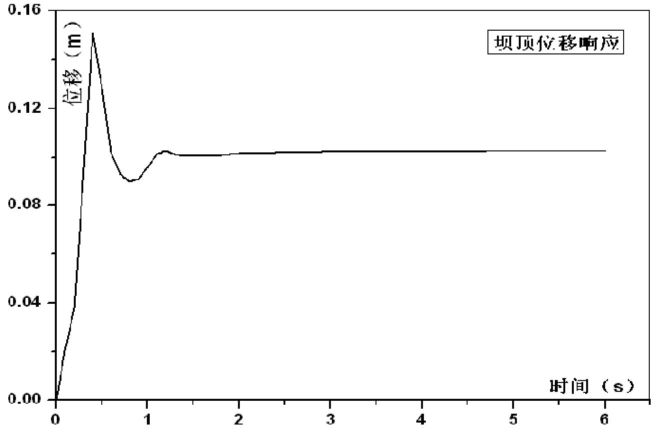


有限元造型

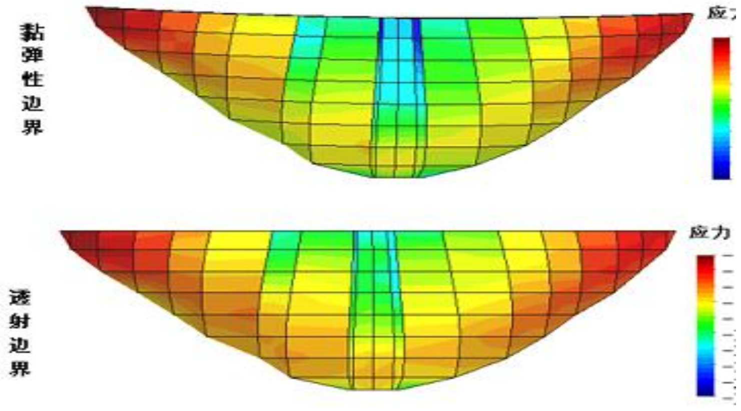


小湾拱坝实景图片

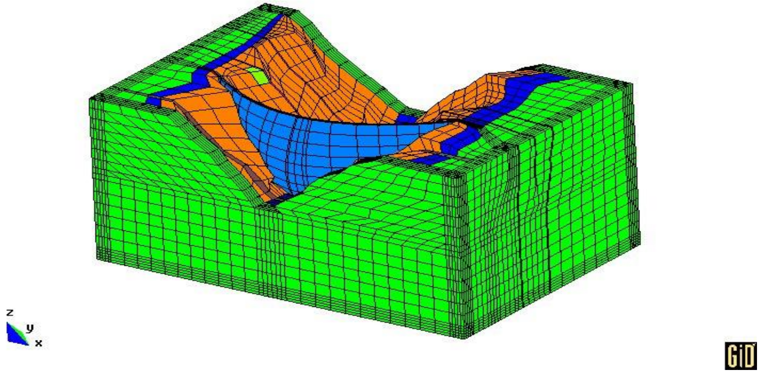
■共有28285个节点，1075个接触点对，总共23862个单元（其中23022个8节点六面体单元，736个六节点五面体棱柱单元，90个四边形边界单元，14个三角形边界单元），25种材料。



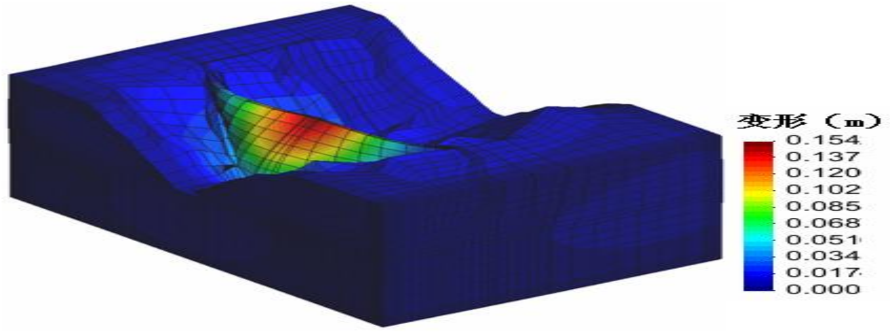
坝顶节点在坝体重力作用下的稳定过程



小湾坝体地震荷载作用下上游面压应力分布

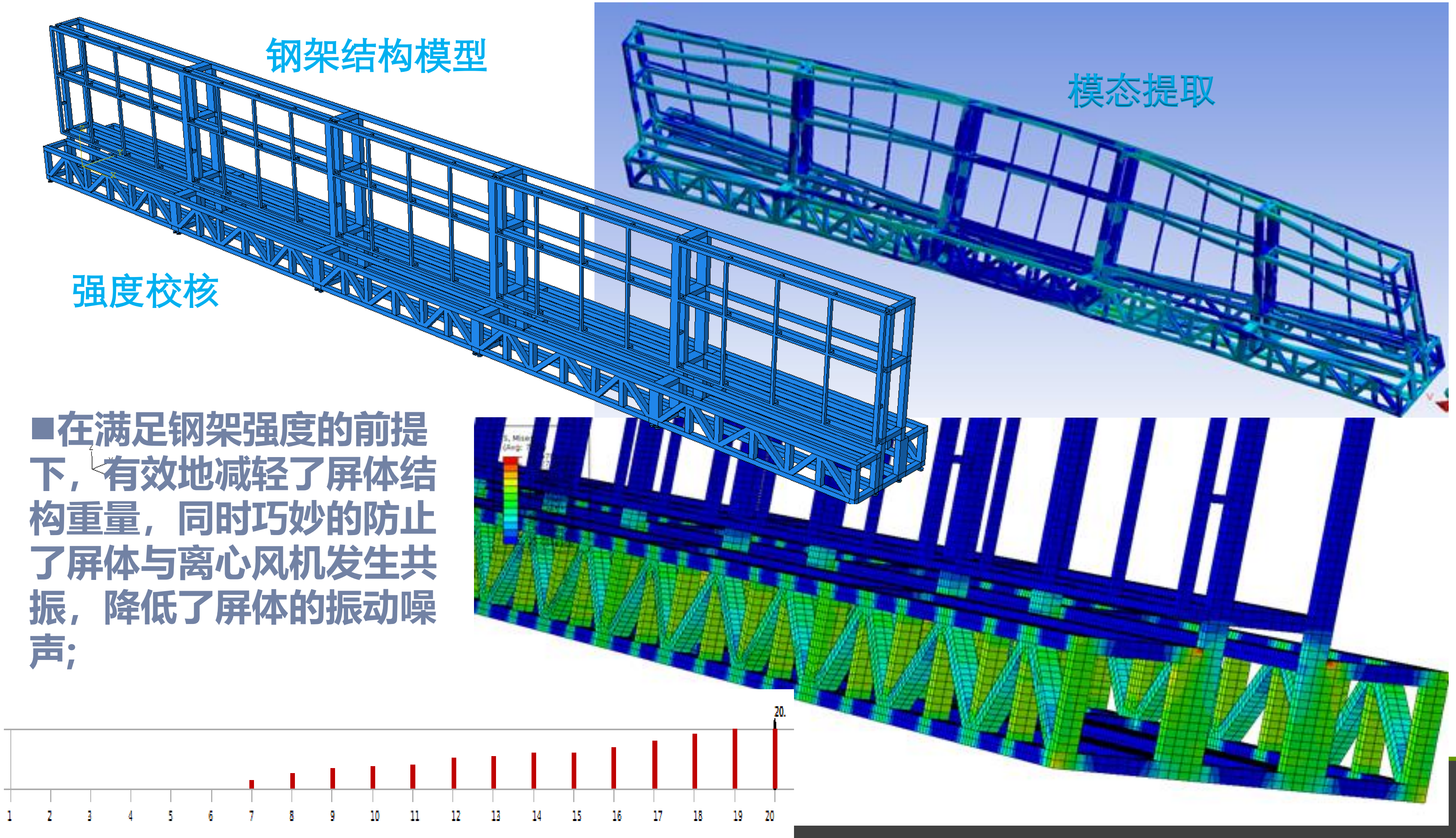


小湾拱坝计算模型



小湾拱坝造型整体静位移

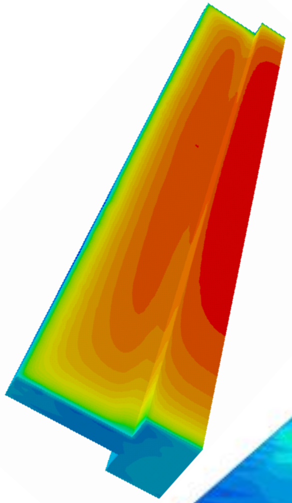
案例5：天安门广场LED屏钢架结构优化设计

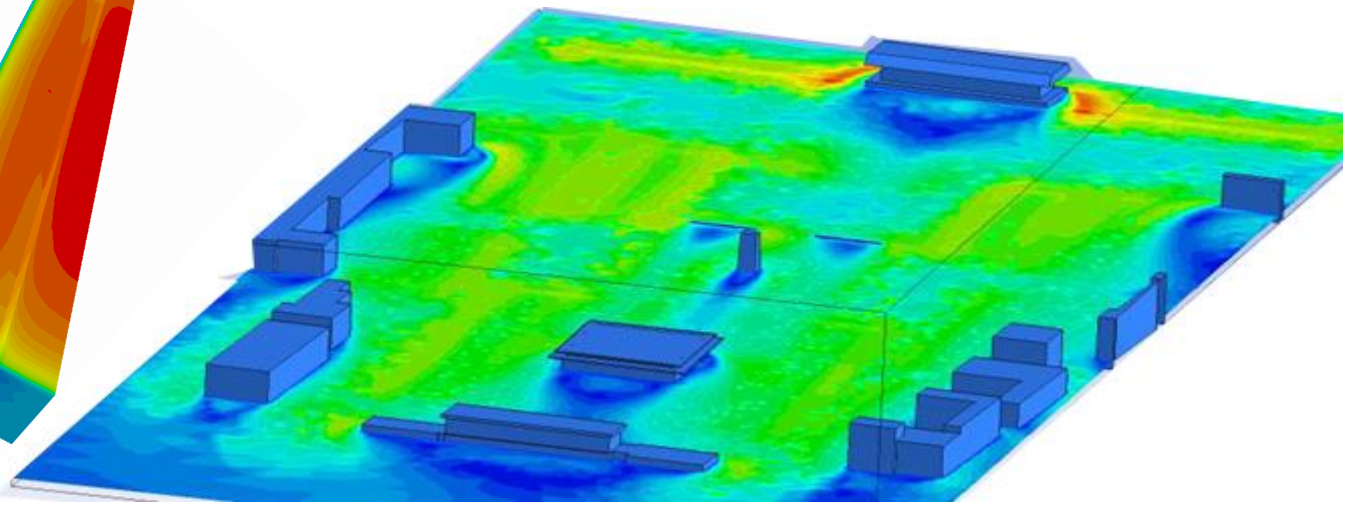


■在满足钢架强度的前提下，有效地减轻了屏体结构重量，同时巧妙的防止了屏体与离心风机发生共振，降低了屏体的振动噪声;

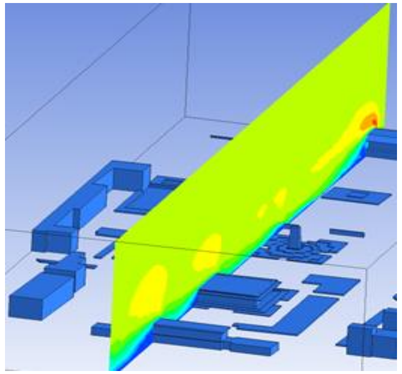
案例6：天安门广场LED屏抗风载性能设计

LED屏表面风压分布





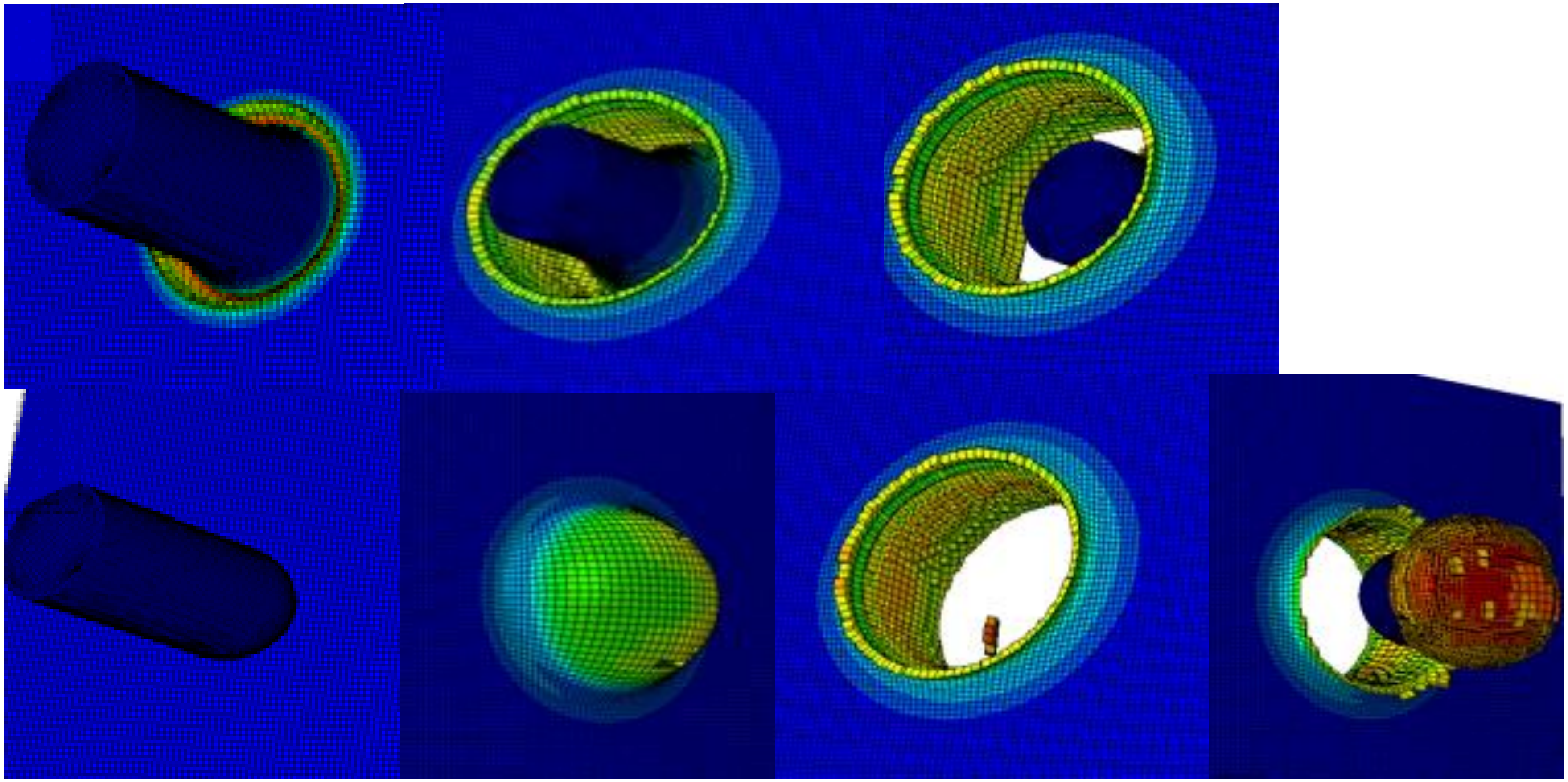
天安门广场风速场仿真



■采用Fluent软件模拟天安门广场风速场分布，计算出LED屏表面风压分布情况。将得到的风压结果作为边界条件附加到Abaqus计算模型上，进一步评测风载对屏体钢架结构的影响，确定出LED屏的最大抗风载能力;

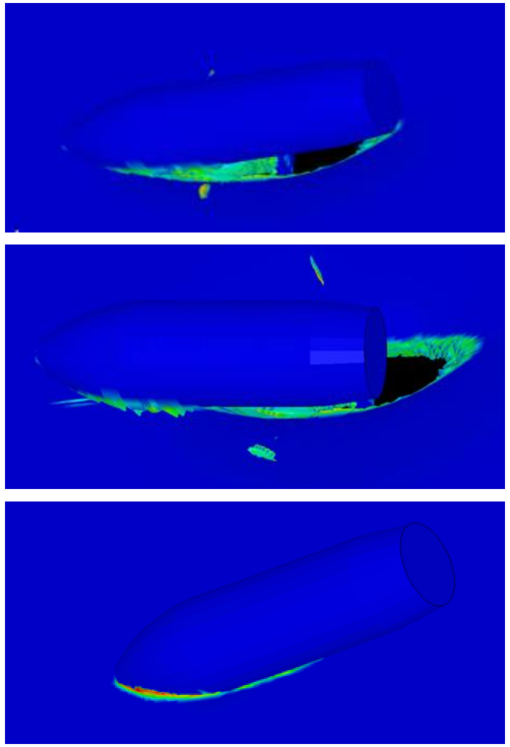
案例7:某兵工厂弹体侵彻分析

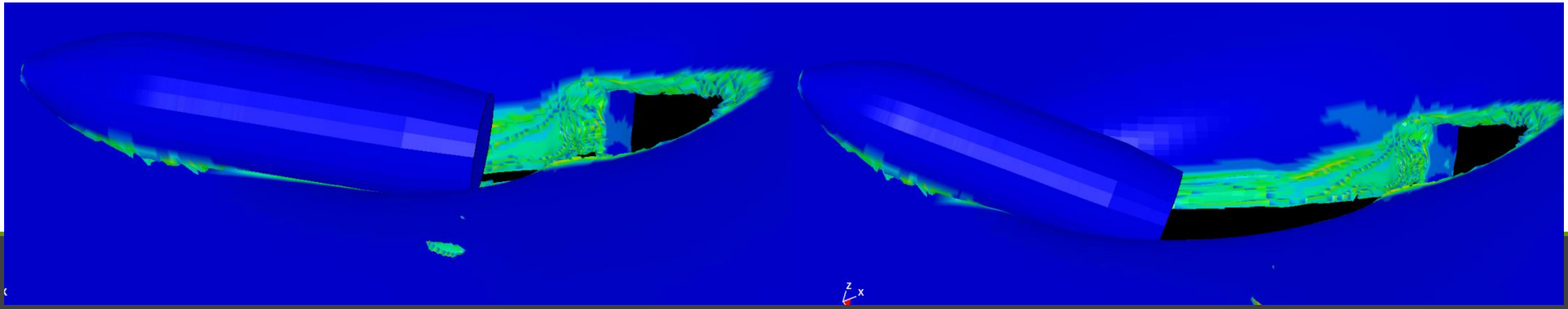
■在模拟弹体侵彻芳纶、钢板、陶瓷等材料的侵彻过程,有效减少打靶实验次数;
■采用软件Abaqus\Ls-dyna;
■网格规模(百万网格);

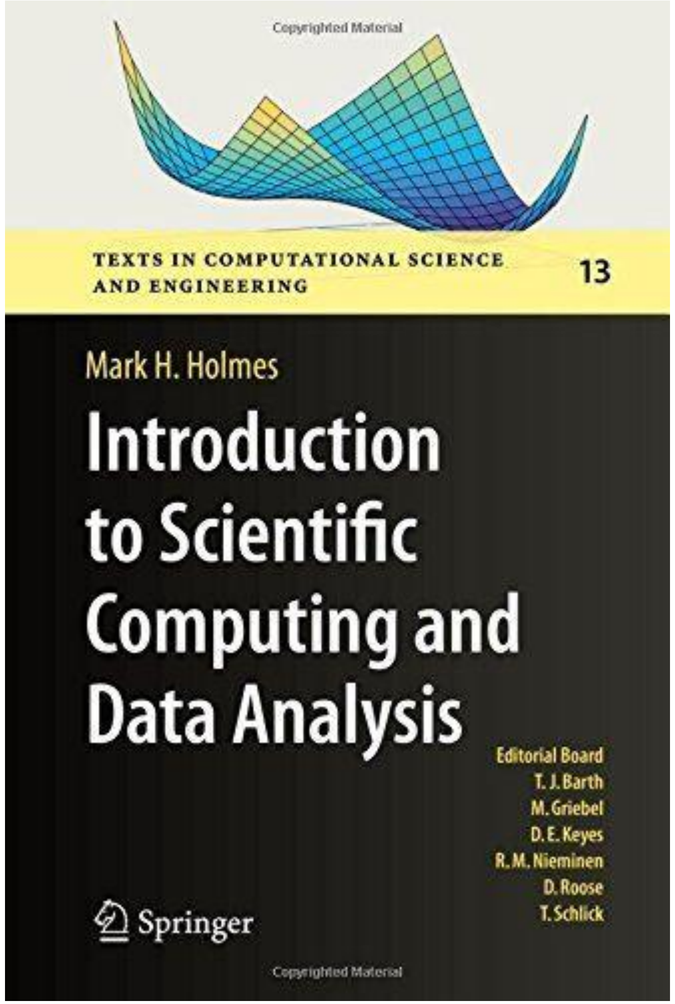


案例8:某兵工厂弹体破甲性能分析

■在主要用于某型坦克前甲板抗弹性能改良;
■采用软件Ls-dyna;
■网格规模(百万网格);
■通过破甲性能分析，得出甲板加强肋的最佳设计方案，
有效提高了某型坦克前甲板的抗弹性能;







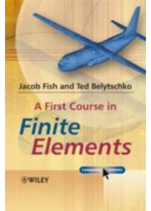
TEXTS IN COMPUTATIONAL SCIENCE AND ENGINEERING 13

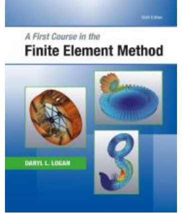
Mark H. Holmes

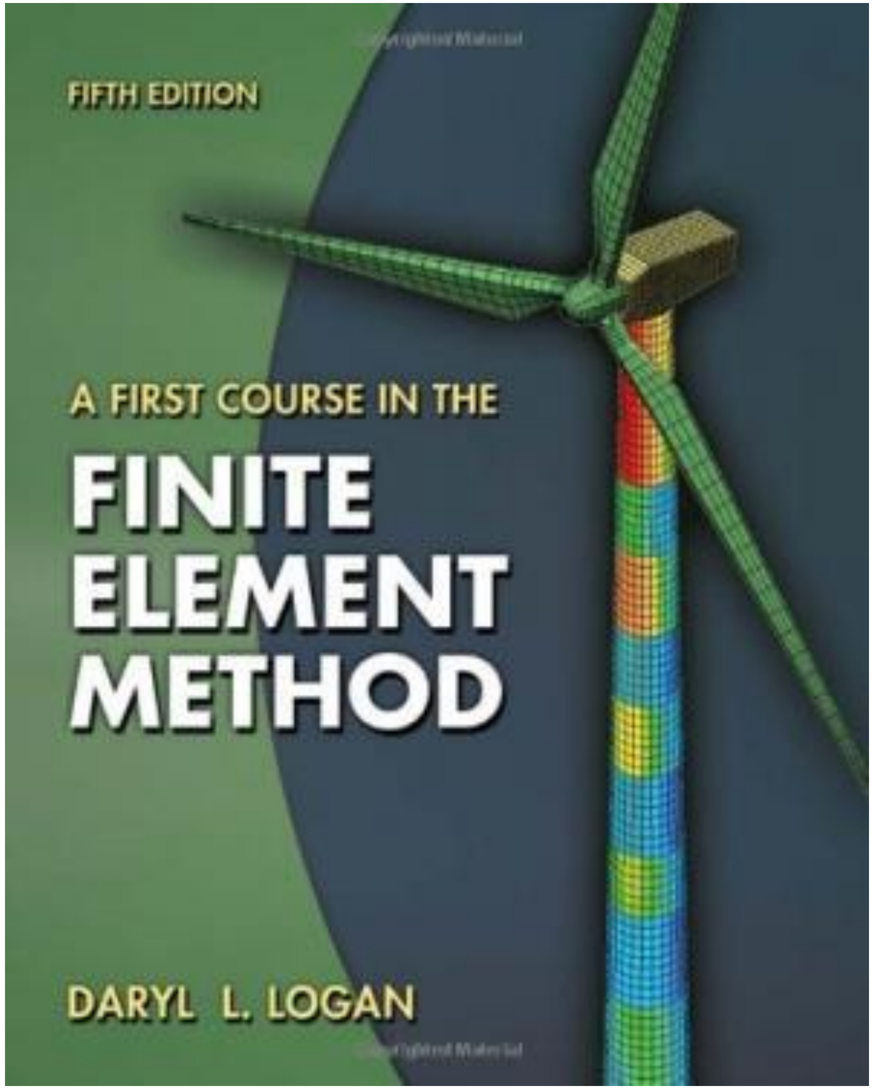
Introduction to Scientific Computing and Data Analysis

Springer

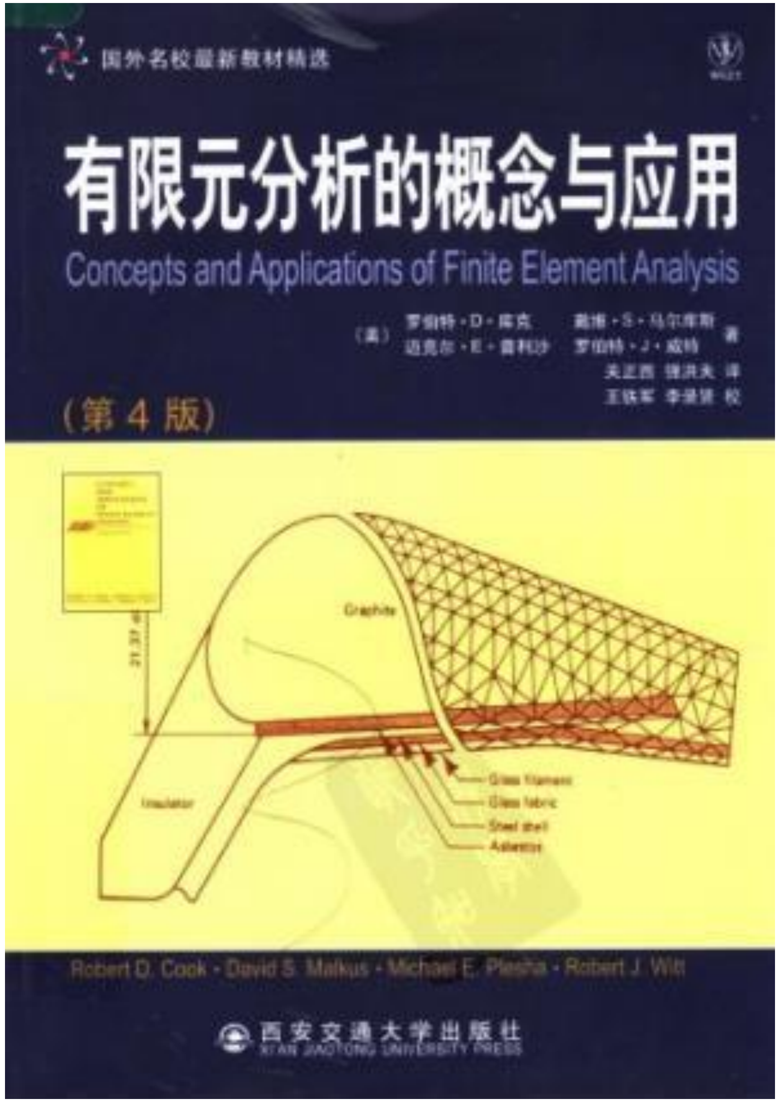
- □ Introduction to Scientific Computing and Data Analysis
- □ Authors Mark H. Holmes (auth.)
- □ Year 2016
- □ Pages 505
- □ Publisher Springer International Publishing
- □ Language en
- □ ISBN 9783319302546







- A First Course in the Finite Element Method
- *Daryl L. Logan*
  - A FIRST COURSE IN THE FINITE ELEMENT METHOD provides a simple, basic approach to the course material that can be understood by both undergraduate and graduate students without the usual prerequisites (i.e. structural analysis). The book is written primarily as a basic learning tool for the undergraduate student in civil and mechanical engineering whose main interest is in stress analysis and heat transfer. The text is geared toward those who want to apply the finite element method as a tool to solve practical physical problems.



国外名校最新教材精选

有限元分析的概念与应用

Concepts and Applications of Finite Element Analysis

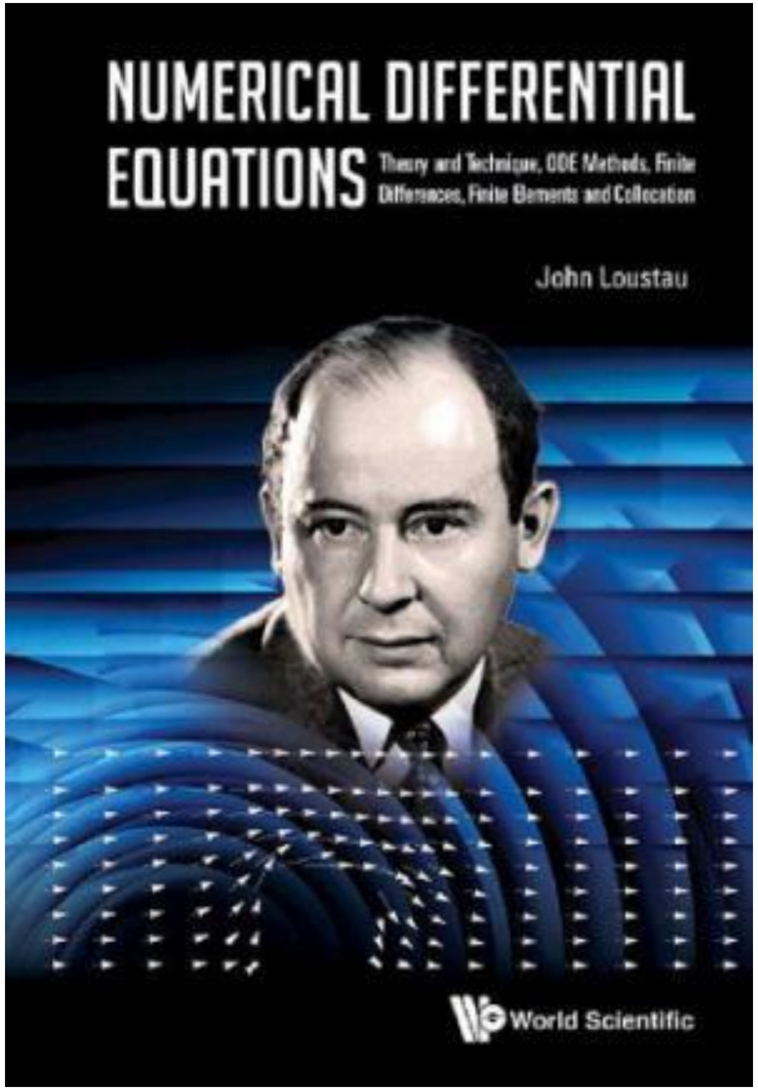
（美） 罗伯特·D·库克 戴维·S·马尔库斯 著
迈克尔·E·普利沙 罗伯特·J·威特
关正西 强洪夫 译
王铁军 李录贤 校

（第 4 版）

Robert D. Cook · David S. Malkus · Michael E. Plesha · Robert J. Witt

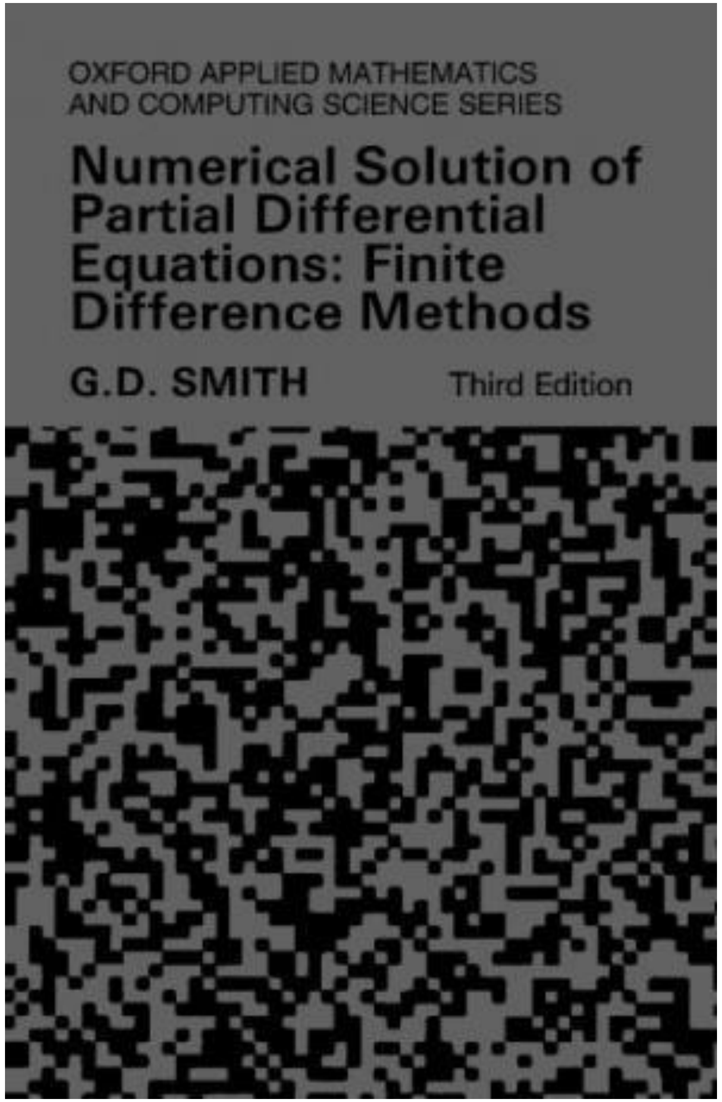
西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

- 有限元分析的概念与应用-第四版
- Robert D. Cook, David S. Malkus, Michael E. Plesha, Robert J. Witt
- 《有限元分析的概念与应用(第4版)》适合机械、力学、土木、动力、材料、水利和航空航天等专业高年级本科生和研究生作为有限元课程的教材及教学参考书，对相关专业的工程技术人员和科研工作者也有很好的参考价值。基础理论和工程应用并重的特色鲜明，是一本有限元课程的优秀教材。


NUMERICAL DIFFERENTIAL EQUATIONS
Theory and Technique, ODE Methods, Finite Differences, Finite Elements and Collocation
John Loustau
World Scientific

- Numerical Differential Equations: Theory and Technique, ODE Methods, Finite Differences, Finite Elements and Collocation
- *John Loustau*
  - This text presents numerical differential equations to graduate (doctoral) students. It includes the three standard approaches to numerical PDE, FDM, FEM and CM, and the two most common time stepping techniques, FDM and Runge-Kutta. We present both the numerical technique and the supporting theory.

    The applied techniques include those that arise in the present literature. The supporting mathematical theory includes the general convergence theory. This material should be readily accessible to



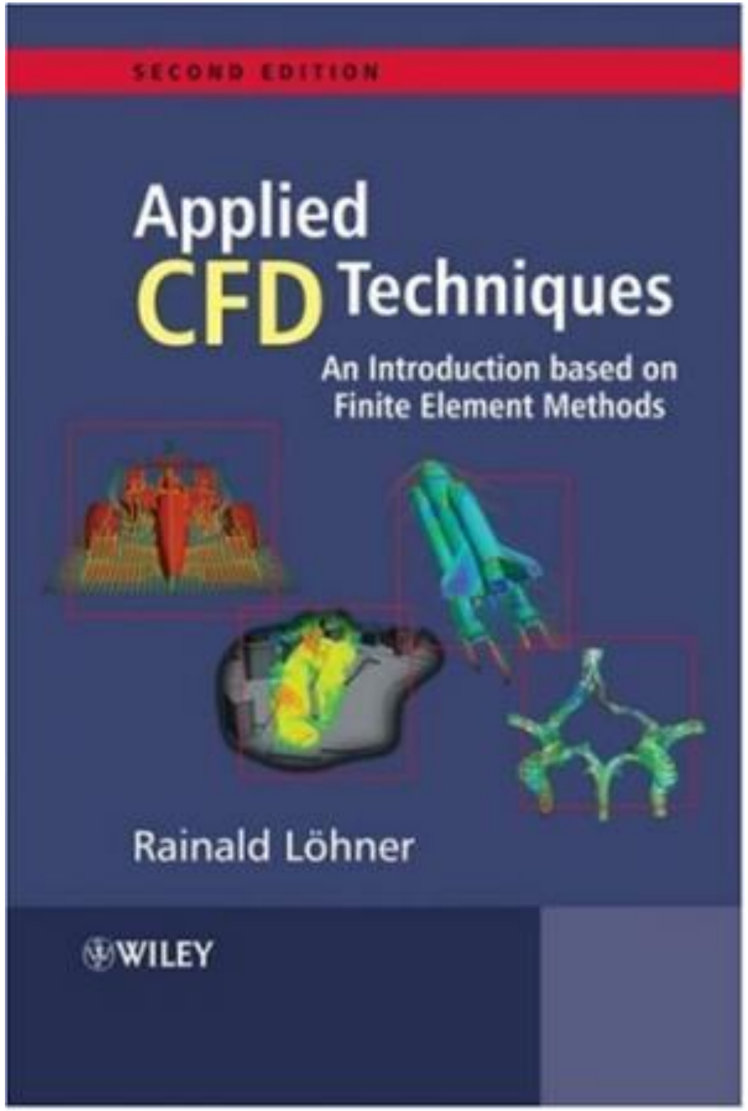
OXFORD APPLIED MATHEMATICS
AND COMPUTING SCIENCE SERIES

Numerical Solution of Partial Differential Equations: Finite Difference Methods

G.D. SMITH

Third Edition

- Numerical Solution of Partial Differential Equations: Finite Difference Methods
- *G. D. Smith*
    - Substantially revised, this authoritative study covers the standard finite difference methods of parabolic, hyperbolic, and elliptic equations, and includes the concomitant theoretical work on consistency, stability, and convergence. The new edition includes revised and greatly expanded sections on stability based on the Lax-Richtmeyer definition, the application of Pade approximants to systems of ordinary differential equations for parabolic and hyperbolic equations, and a considerably improved presentation of iterative methods. A fast-paced introduction to numerical methods, this will be a useful volume for students of mathematics and



SECOND EDITION

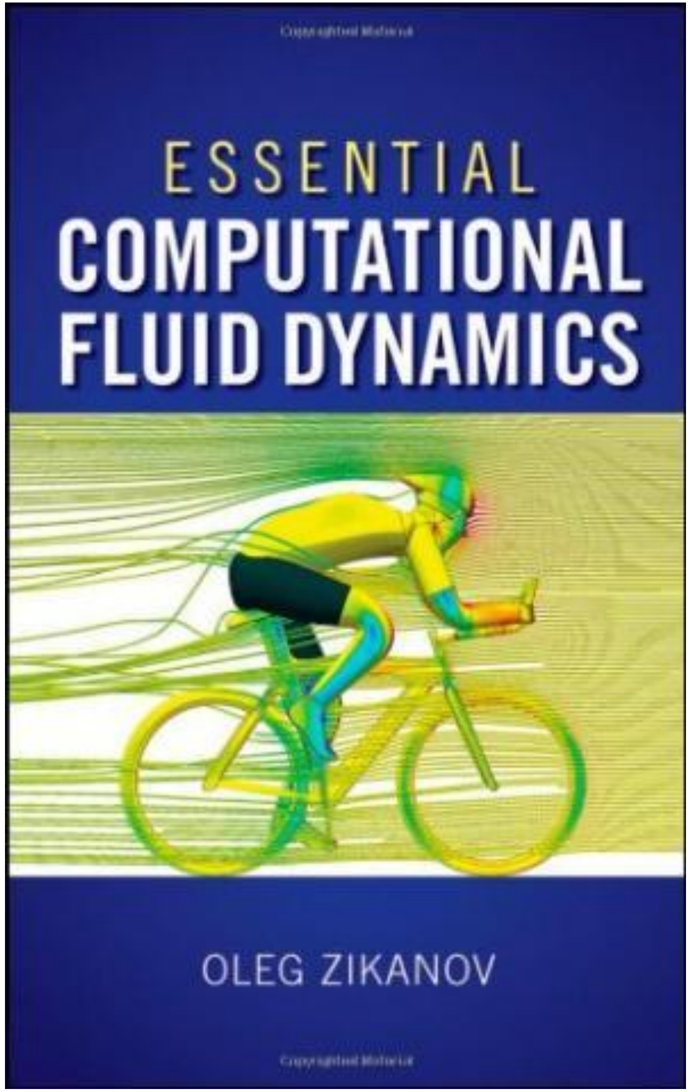
Applied CFD Techniques

An Introduction based on Finite Element Methods

Rainald Löhner

WILEY

- □ Applied computational fluid dynamics techniques: an introduction based on finite element methods
- □ *Prof Rainald Löhner*
  - Computational fluid dynamics (CFD) is concerned with the efficient numerical solution of the partial differential equations that describe fluid dynamics. CFD techniques are commonly used in the many areas of engineering where fluid behavior is an important factor. Traditional fields of application include aerospace and automotive design, and more recently, bioengineering and consumer and medical electronics. With Applied Computational Fluid Dynamics Techniques, 2nd edition, Rainald Löhner introduces the reader to the techniques required to achieve efficient CFD solvers, forming a bridge between basic theoretical and algorithmic



- Essential Computational Fluid Dynamics
- *Oleg Zikanov*
  - This book serves as a complete and self-contained introduction to the principles of Computational Fluid Dynamic (CFD) analysis. It is deliberately short (at approximately 300 pages) and can be used as a text for the first part of the course of applied CFD followed by a software tutorial. The main objectives of this non-traditional format are: 1) To introduce and explain, using simple examples where possible, the principles and methods of CFD analysis and to demystify the `black box' of a CFD software tool, and 2) To provide a basic understanding of how CFD problems are set and which factors affect the success and failure of the analysis. Included in the text are the mathematical and physical foundations of CFD, formulation of CFD

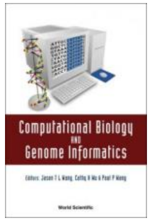
Chapter 2: HPC with an example

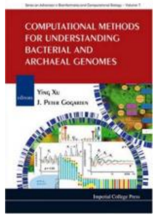
Faster for larger data

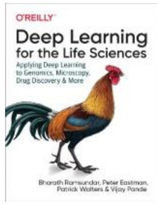
- Many problems/applications need HPComputers
  - **Weather / Climate**， Cryptography，Nuclear Weapons Design，Scientific Simulation，Petroleum Exploration，Aerospace Design，Automotive Design，Pharmaceutical Design，Data Mining，Data Assimilation
- Heat dynamics as an example to understand Numeric Computing/Scientific Computing
  - Like Finite Element Analysis (有限元分析)
- Other examples

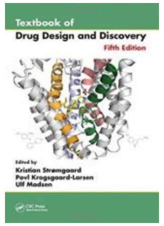
Many applications needs more computation power

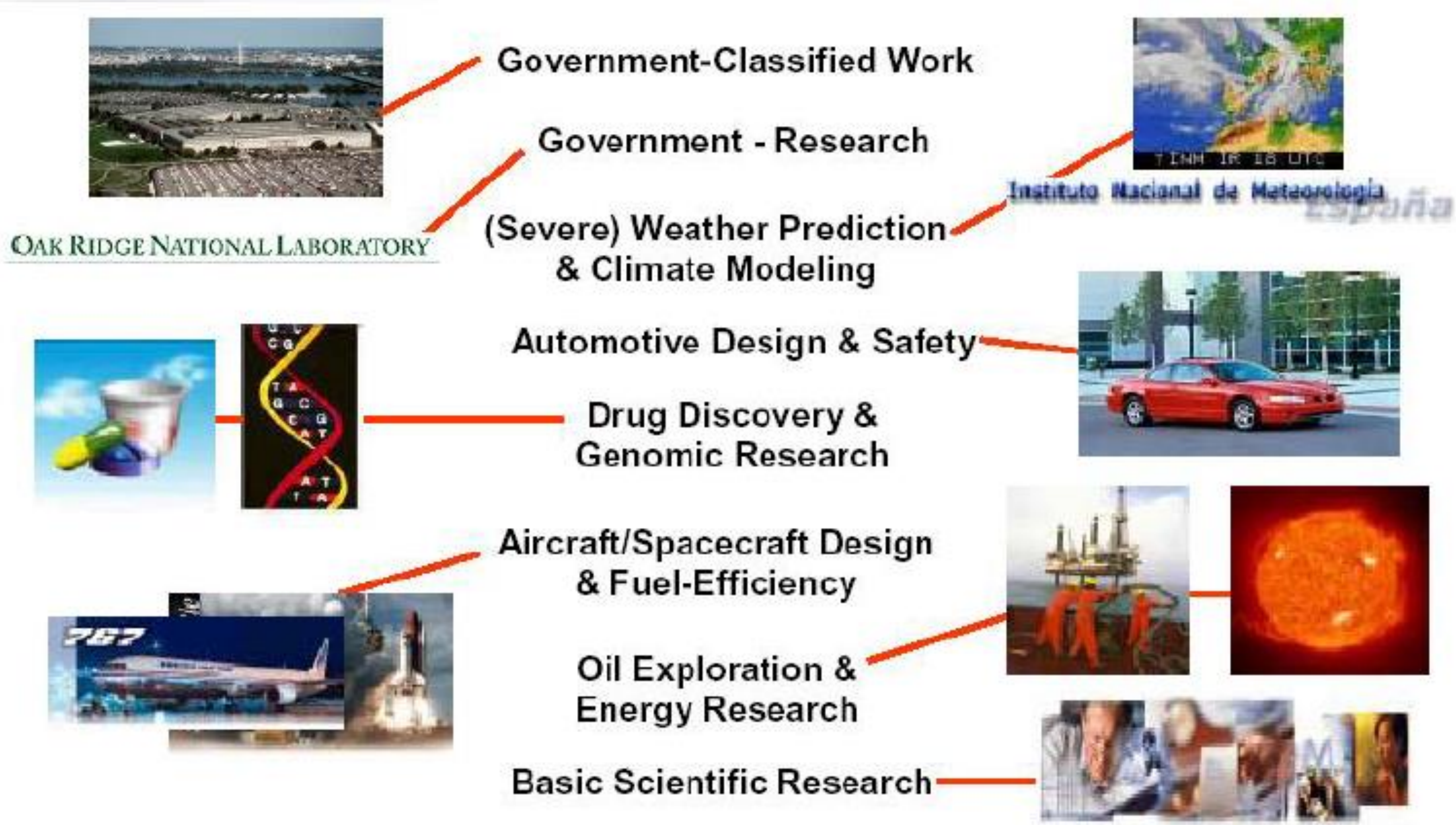
- **With HUGE data to be processed**
  - Scientific computing, business applications etc.

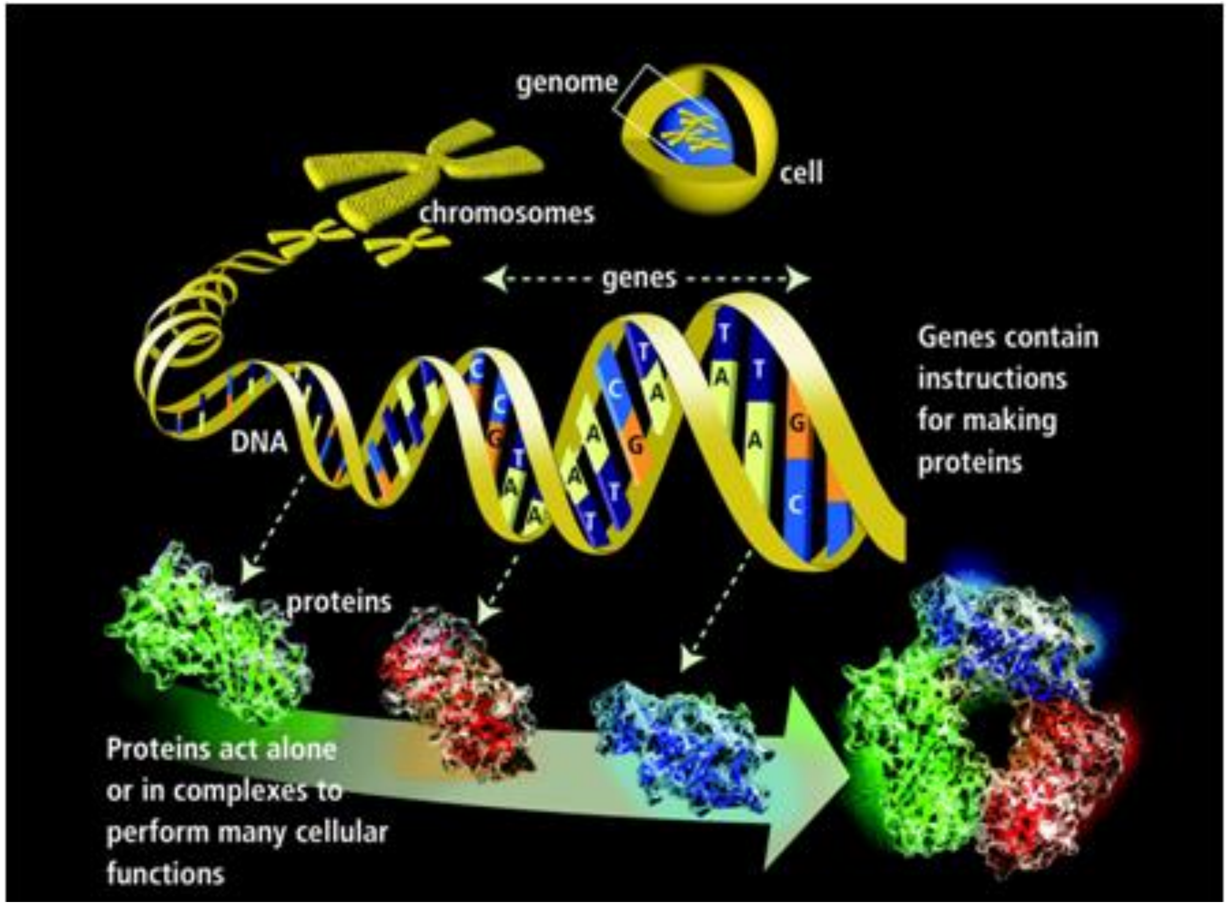



COMPUTATIONAL METHODS FOR UNDERSTANDING BACTERIAL AND ARCHAEAL GENOMES
Ying Xu
J. Peter Gogarten
Imperial College Press


O'REILLY
Deep Learning for the Life Sciences
Applying Deep Learning to Genomics, Microscopy, Drug Discovery & More
Bharath Ramsundar, Peter Eastman, Patrick Walters & Vijay Pande


Textbook of Drug Design and Discovery
Fifth Edition
Edited by Kristian Strømgaard, Povl Krogsgaard-Larsen, Ulf Madsen
CRC Press


Government-Classified Work
Government - Research
OAK RIDGE NATIONAL LABORATORY
(Severe) Weather Prediction & Climate Modeling
Instituto Nacional de Meteorologia
España
Automotive Design & Safety
Drug Discovery & Genomic Research
Aircraft/Spacecraft Design & Fuel-Efficiency
767
Oil Exploration & Energy Research
Basic Scientific Research


genome
cell
chromosomes
genes
DNA
Genes contain instructions for making proteins
proteins
Proteins act alone or in complexes to perform many cellular functions

DNA sequence

↓

Protein sequence

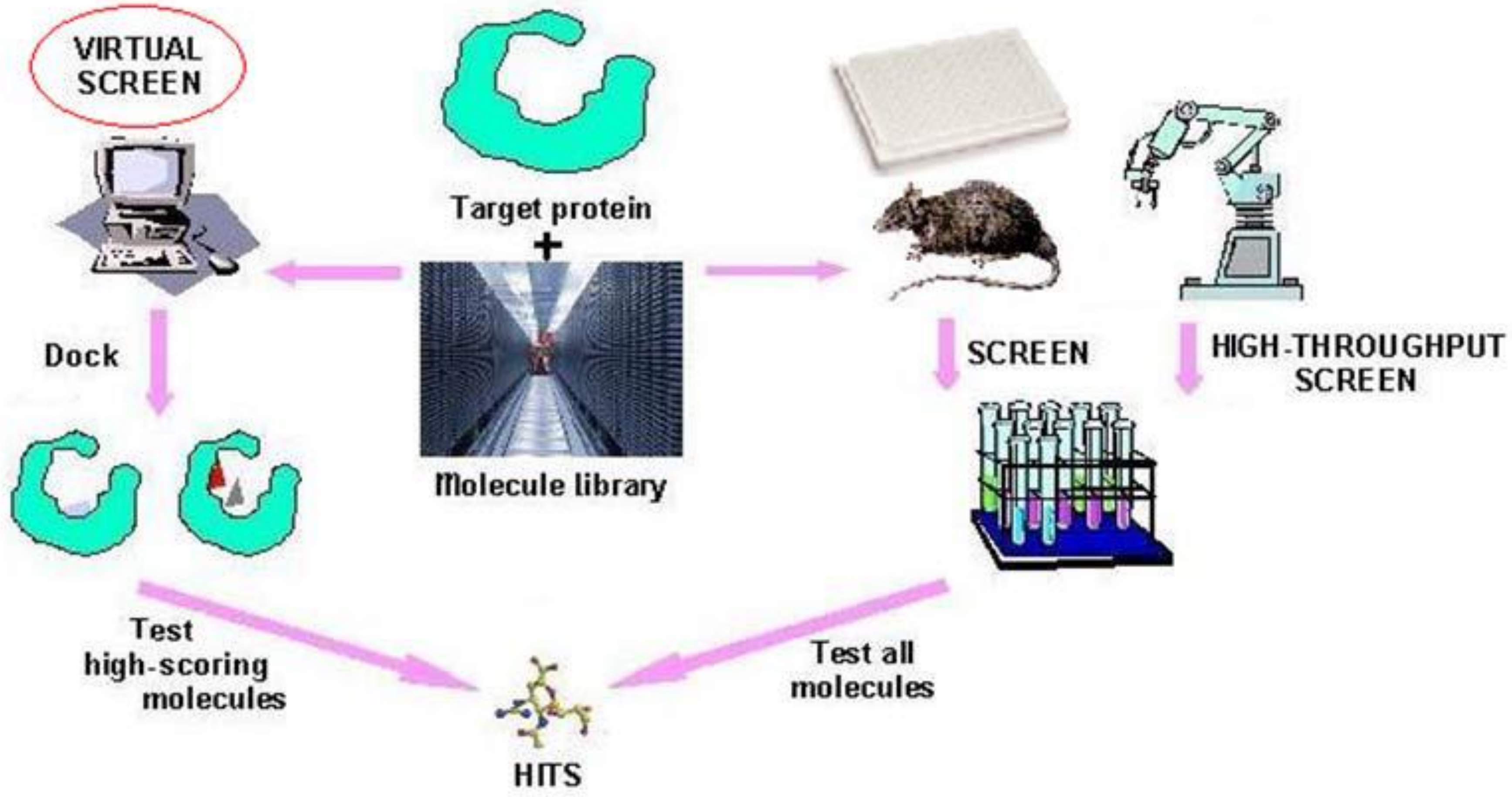
↓

Protein structure

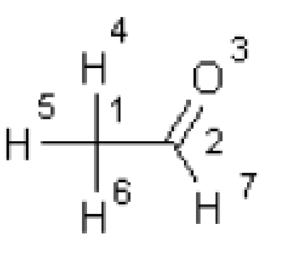
↓

Protein function

Virtual screening in drug discovery



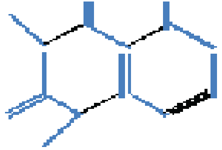
用矩阵和



原子列表	
1	C
2	C
3	O
4	H
5	H
6	H
7	H

$$\begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$$

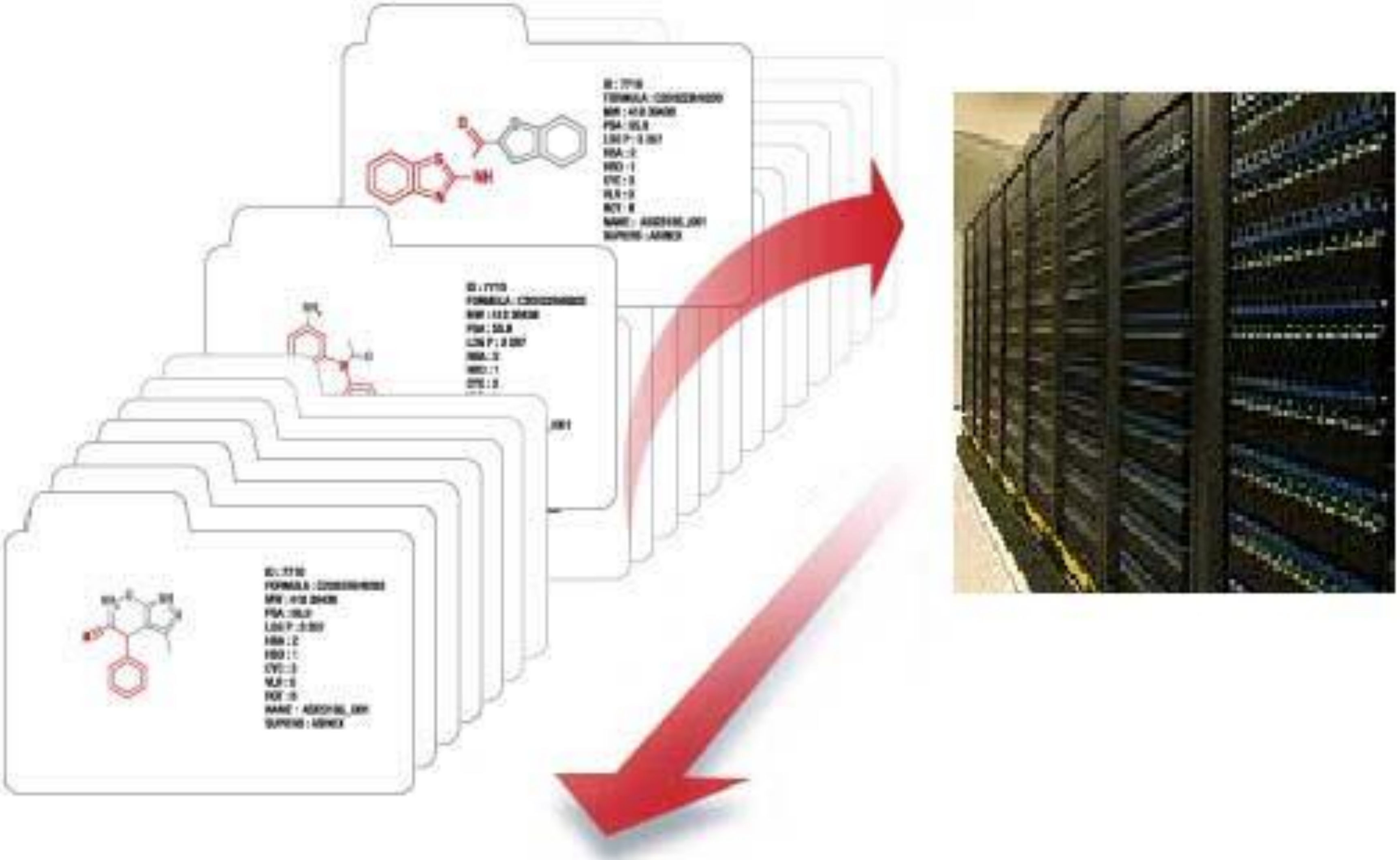
键列表		
第一个原子	第二个原子	键级
1	2	1
2	3	2
2	7	1
1	4	1
1	5	1
1	6	1

Representation	Name
Caffine	Common Name
trimethylxanthine coffeine, theine, mateine,	Synonyms
$C_8H_{10}N_4O_2$	Empirical formula
3,7-dihydro-1,3,7-trimethyl-1H-purine-2,6-dione	IUPAC Name
58-08-2	CAS Registry Number
T56 BN DN FNVNVJ B1 F1 H1	WLN Notation
CN1C=NC2=C1C(=O)N(C(=O)N2C)C	SMILES
1S/C8H10N4O2/c1-10-4-9-6-5(10)7(13)12(3)8(14)11(6)2/h4H,1-3H3	Inchl
	Markush Structure
	Connection Table

	1	2	3	4	5	6	7	8	9	10	11
1	0	1	0	0	0	0	0	0	0	2	0
2	1	0	2	0	0	0	0	0	0	0	0
3	0	2	0	1	0	0	0	1	0	0	0
4	0	0	1	0	1	0	0	0	0	0	0
5	0	0	0	1	0	2	1	0	0	0	0
6	0	0	0	0	2	0	0	0	0	0	0
7	0	0	0	0	1	0	0	0	0	0	0
8	0	0	1	0	0	0	0	0	2	0	0
9	0	0	0	0	0	0	0	2	0	1	0
10	2	0	0	0	0	0	0	0	1	0	1
11	0	0	0	0	0	0	0	0	0	1	0

Representation	Name
[OH]c1cccc1	Fragment Code
0000100110100111	Fingerprint
5244987098423150	Hash Code

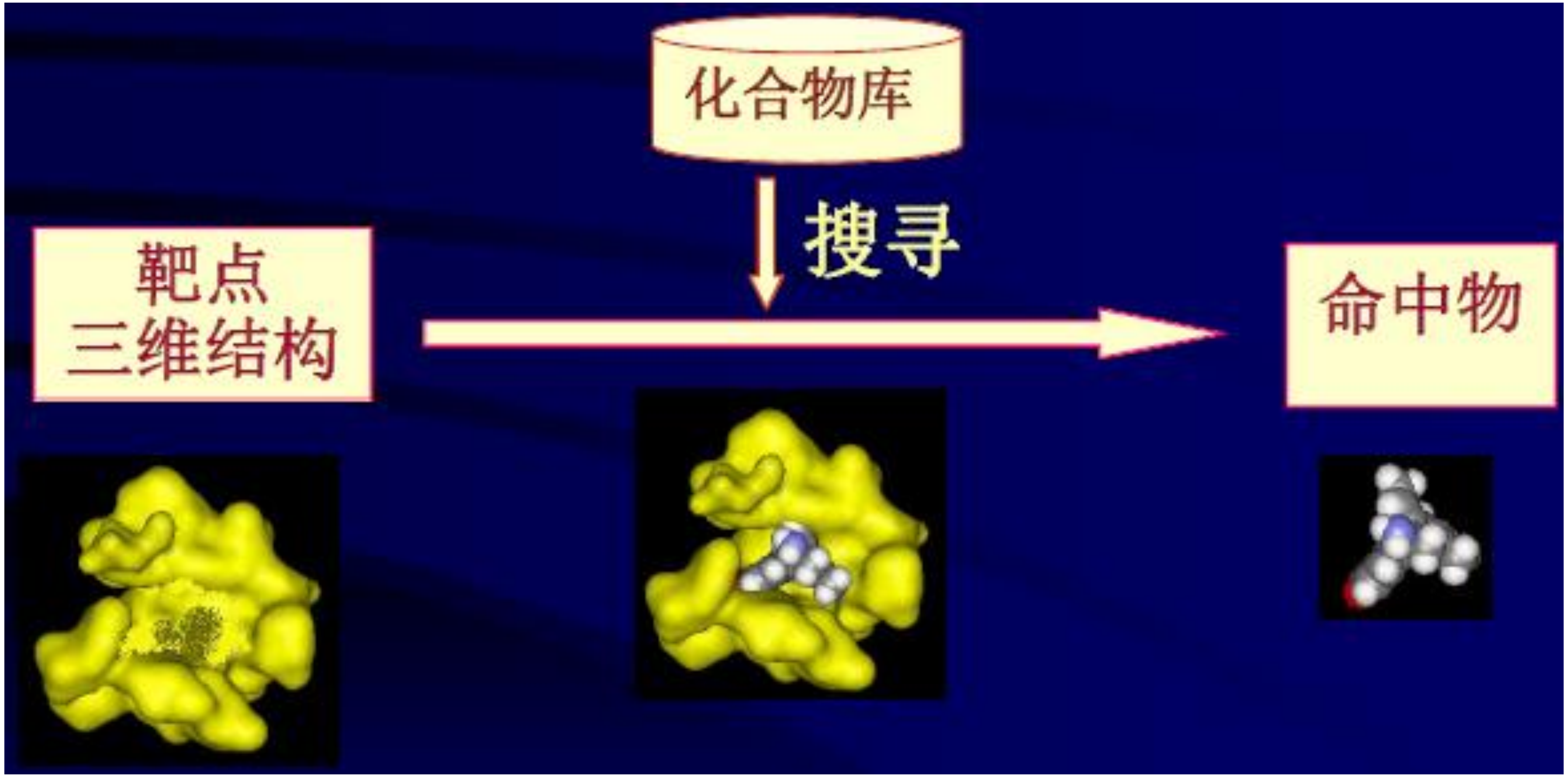
化学信息学资源



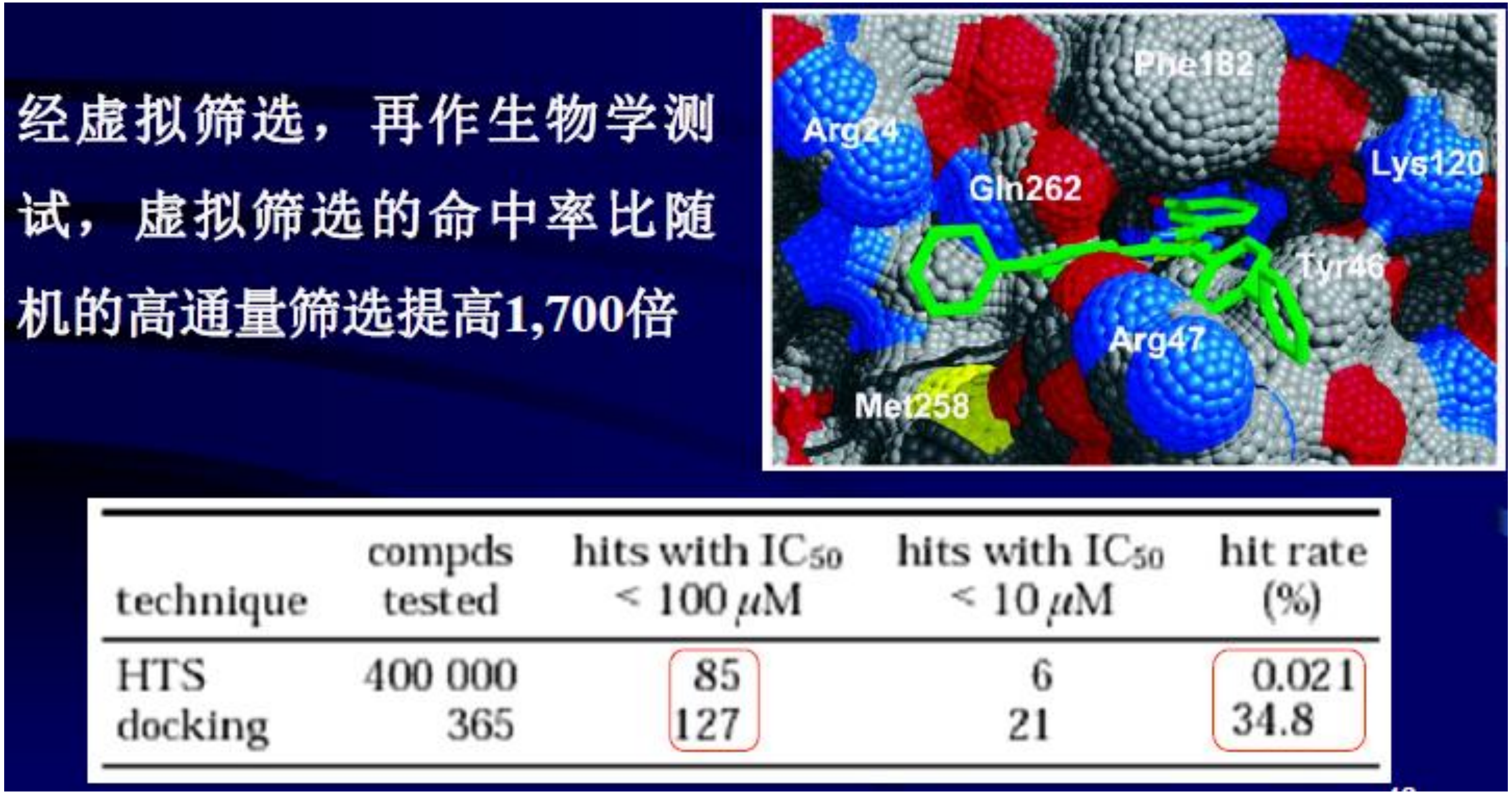
- □ FCD (Fine Chemicals Directory) —— MDL 维护。收载约90 000个化合物和20 000种化合物数据，包括化学系统名、俗称、分子式、分子量、供应商、价格、CAS登录号、纯度等。可通过结构式或其它任何数据检索
- □ ACD (Available Chemicals Directory) ——MDL维护。FCD数据库加上可大批量供货的化学品信息。目前有25万个化合物
- □ CSD（Cambridge Structure Database) —— 20多万个结晶的3D结构实验数据及相关数据

□ 基于靶点结构的虚拟筛选——对接

- target-based virtual screening —— docking

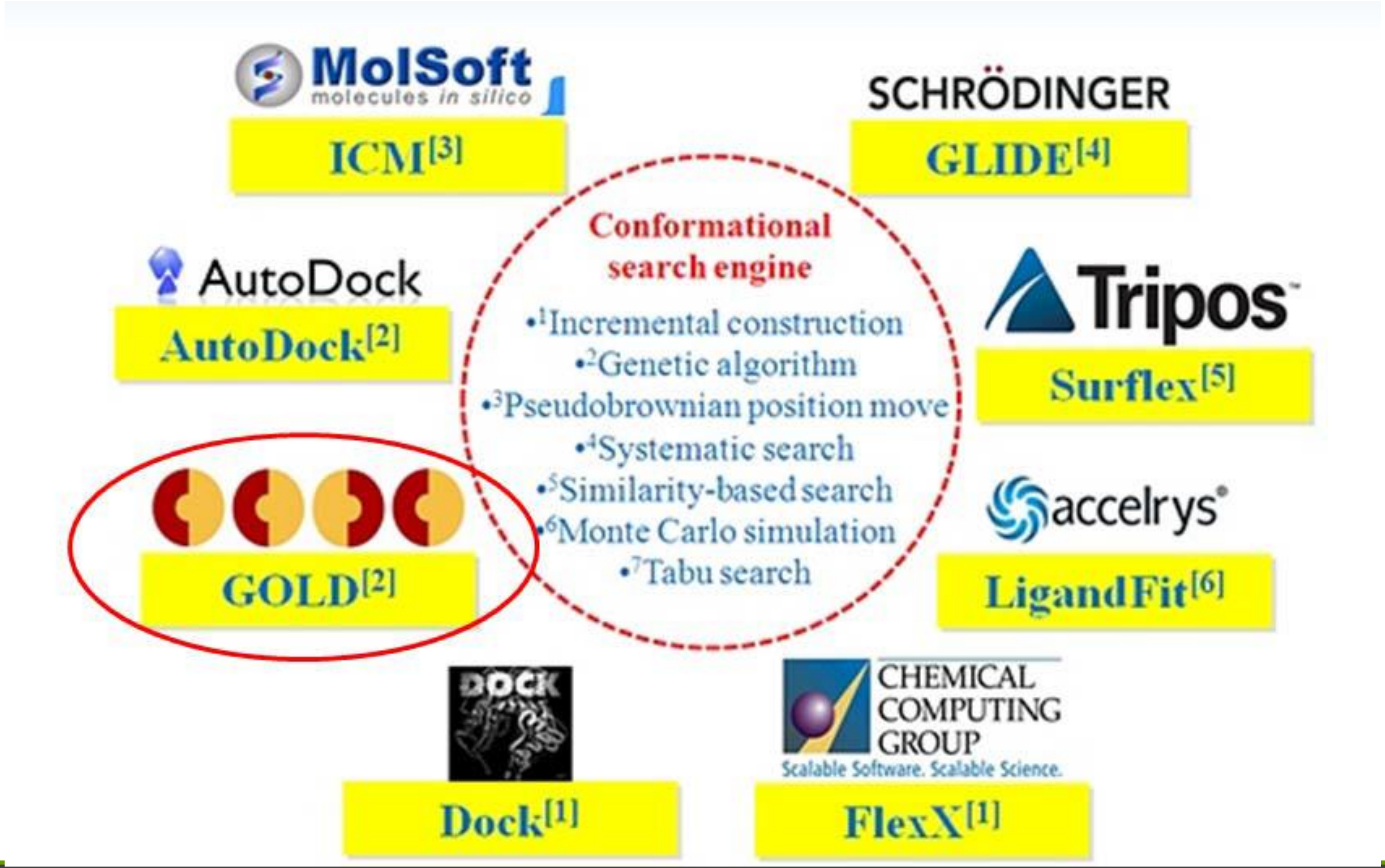


□ 例：蛋白酪氨酸磷酸酯酶1B（PTP1B）抑制剂的发现



经虚拟筛选，再作生物学测试，虚拟筛选的命中率比随机的高通量筛选提高1,700倍

technique	compds tested	hits with IC_{50} < 100 μM	hits with IC_{50} < 10 μM	hit rate (%)
HTS	400 000	85	6	0.021
docking	365	127	21	34.8



ICM[3]

GLIDE[4]

AutoDock[2]

Surflex[5]

GOLD[2]

LigandFit[6]

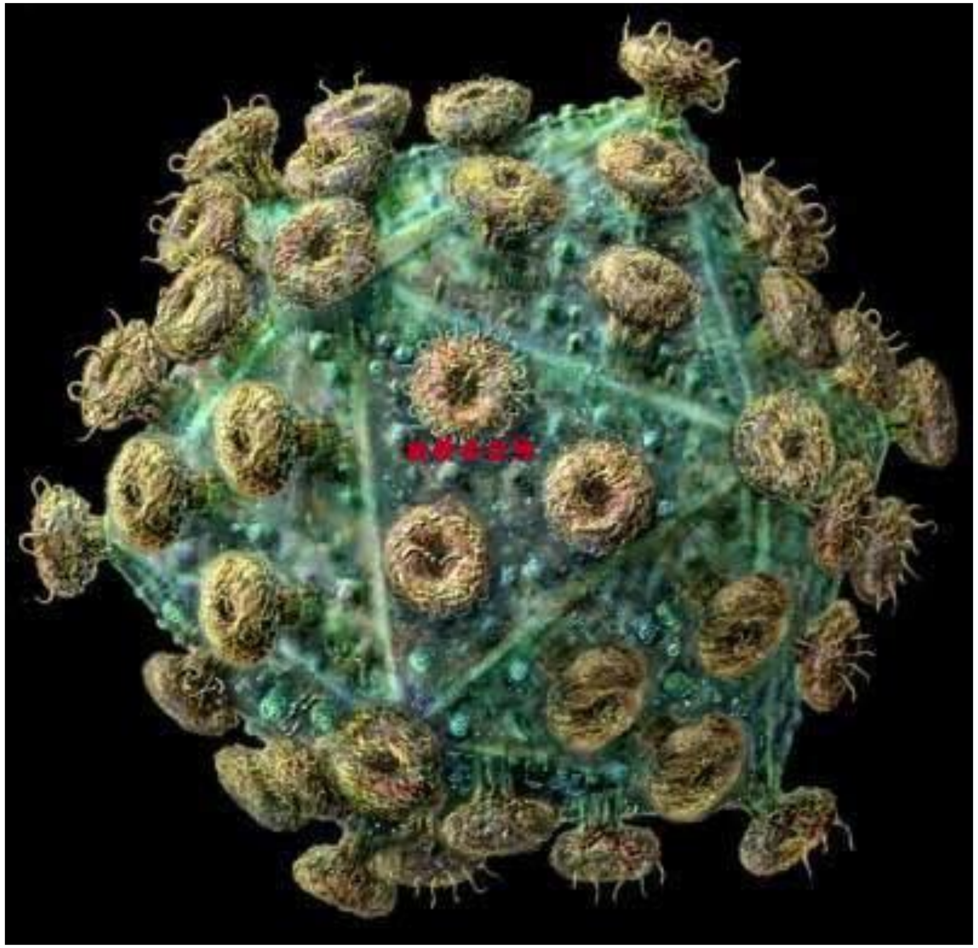
Dock[1]

FlexX[1]

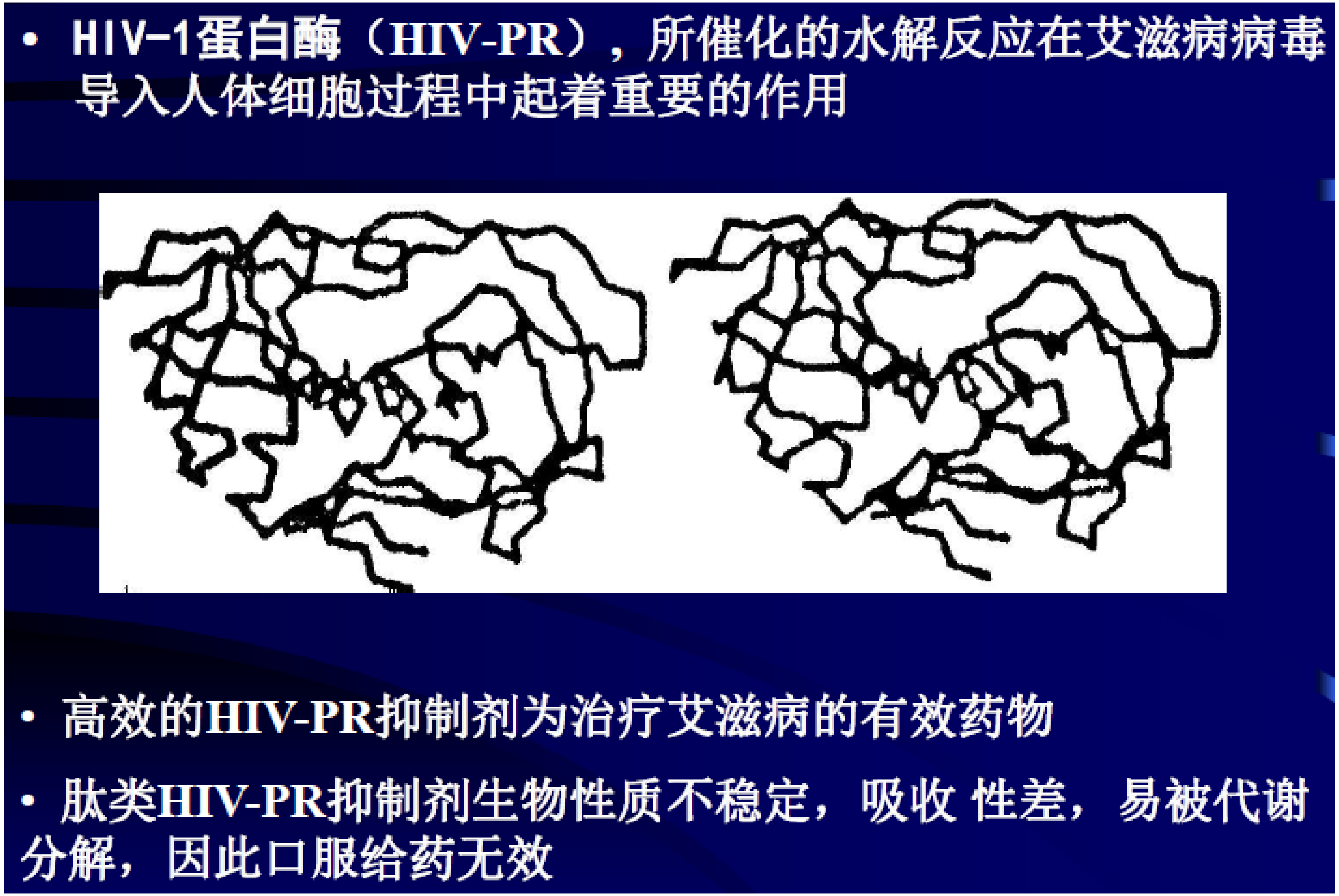
Conformational search engine

- [1]Incremental construction
- [2]Genetic algorithm
- [3]Pseudobrownian position move
- [4]Systematic search
- [5]Similarity-based search
- [6]Monte Carlo simulation
- [7]Tabu search

□ [例] 抗艾滋病药物的发现——虚拟筛选



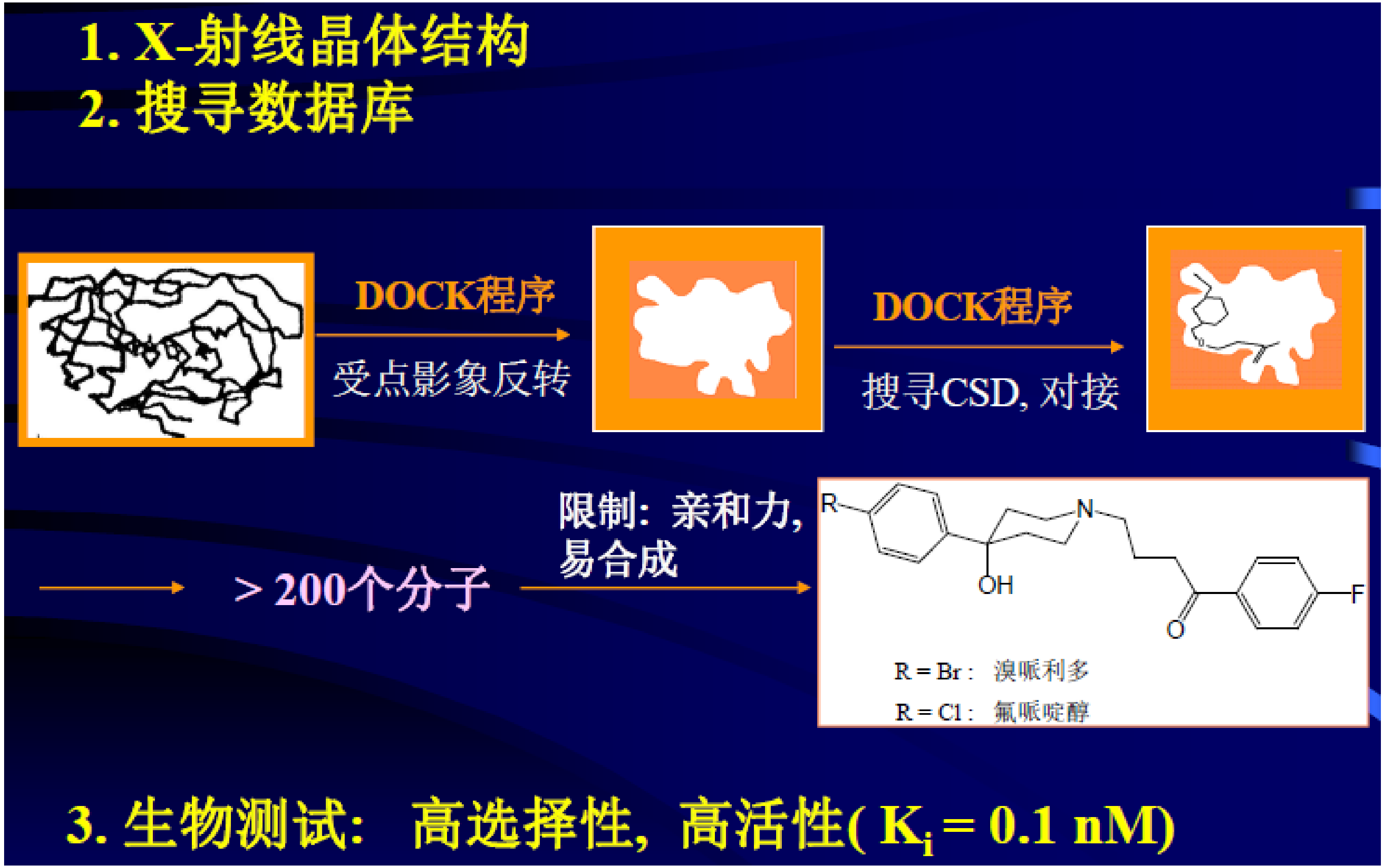
艾滋病病毒, 人类免疫缺陷病毒
human immunodeficiency virus, HIV



- HIV-1蛋白酶（HIV-PR），所催化的水解反应在艾滋病病毒导入人体细胞过程中起着重要的作用

- 高效的HIV-PR抑制剂为治疗艾滋病的有效药物

- 肽类HIV-PR抑制剂生物性质不稳定，吸收 性差，易被代谢分解，因此口服给药无效



1. X-射线晶体结构

2. 搜寻数据库

DOCK程序

受点影象反转

DOCK程序

搜寻CSD, 对接

> 200个分子

限制：亲和力，易合成

R = Br : 溴哌利多

R = Cl : 氟哌啶醇

3. 生物测试： 高选择性，高活性($K_i = 0.1$ nM)

- [例] 抗SARS冠状病毒药物的设计
- ——基于SARS-CoV 3CL蛋白酶的虚拟筛选

严重急性呼吸道综合征
Severe Acute Respiratory Syndrome
SARS

病原体——SARS冠状病毒
SARS-Cov

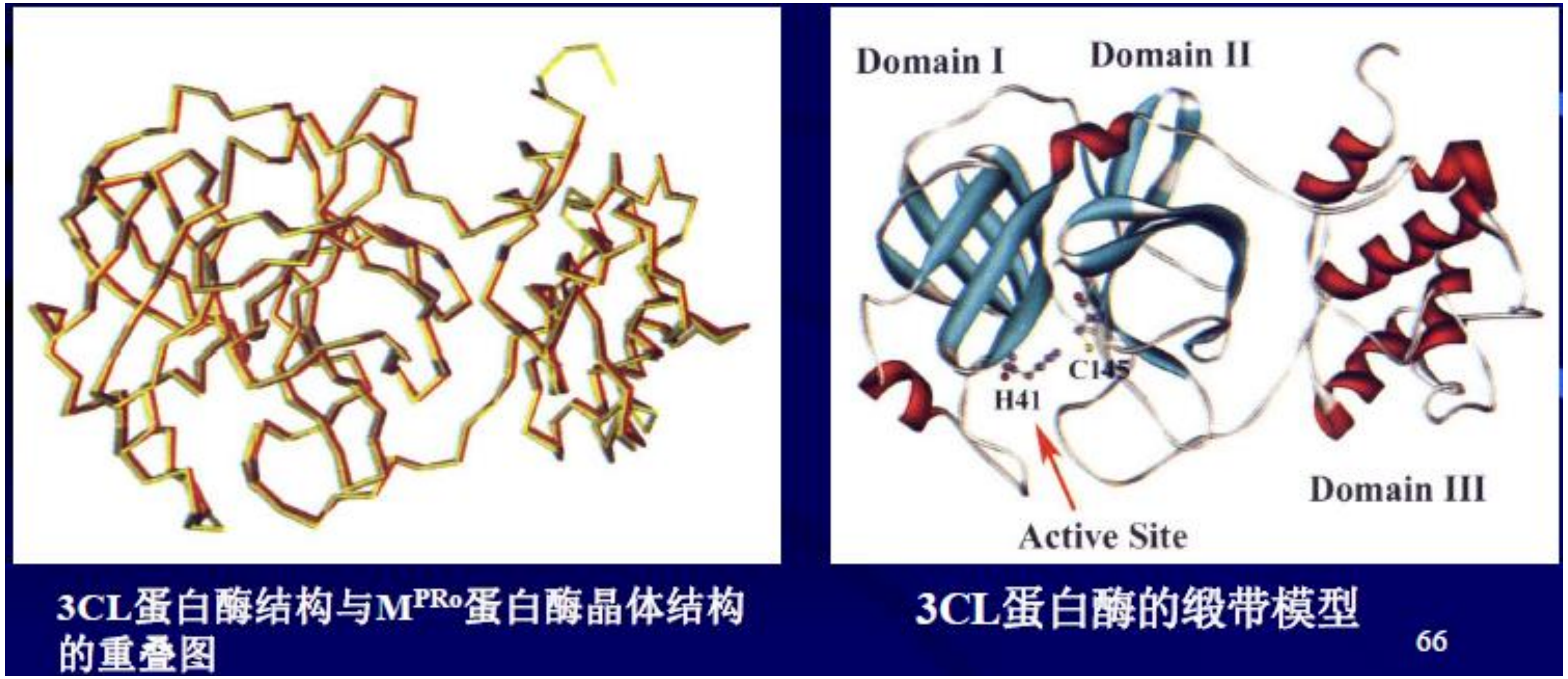
62

□ 步骤1. 同源模建

- (1) 3CL蛋白酶序列(GenBank)与各类冠状病毒蛋白酶序列（PDB）作序列分析和相似性分析（BLAST程序）
  - 人冠状病毒；鼠科肝炎病毒；猪传染性腹泻病毒；猫传染性腹膜炎病毒；禽传染性支气管炎病毒；猪冠状病毒；传染性胃肠炎病毒
- (2) 传染性胃肠炎病毒（TGEV）的MPRo与3CL蛋白酶有极高的相似性，特别在底物结合口袋（活性部位）
- (3) 以TGEV MPRo的X-射线晶体结构为模板，模建3CL蛋白酶三维结构（Sybyl 6.8 / SiteID程序）

□ 结果:

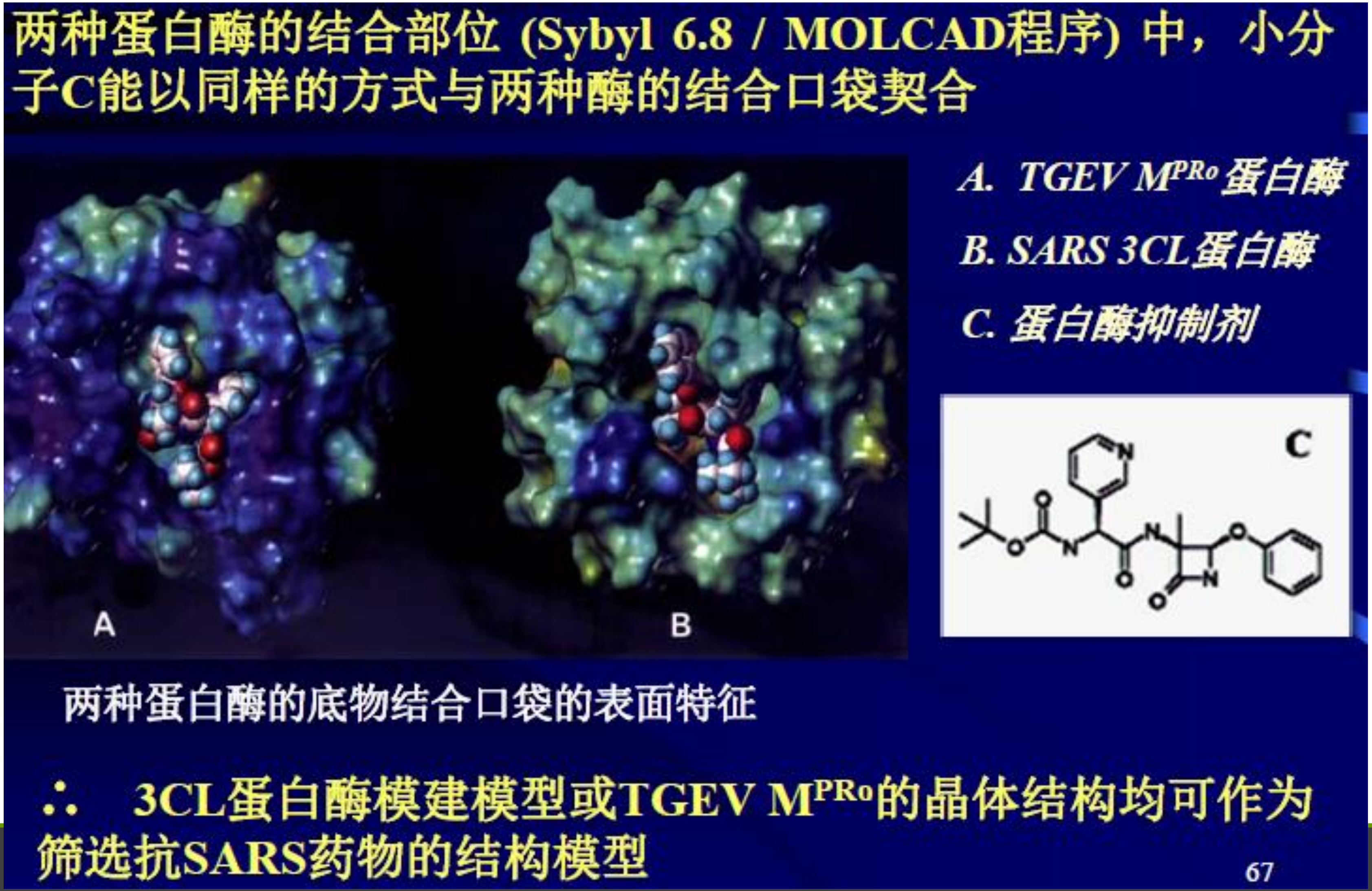
- （1）所建模型与TGEV MPRo晶体结构基本重叠
- （2）3CL蛋白酶的折叠方式与TGEV MPRo相同，结合口袋的结构以及空间特征几乎一样



3CL蛋白酶结构与M^{PRo}蛋白酶晶体结构的重叠图

3CL蛋白酶的缎带模型

□ 步骤2. 分析酶-配体作用模型



两种蛋白酶的结合部位 (Sybyl 6.8 / MOLCAD程序) 中，小分子C能以同样的方式与两种酶的结合口袋契合

A. TGEV M^{PRo}蛋白酶

B. SARS 3CL蛋白酶

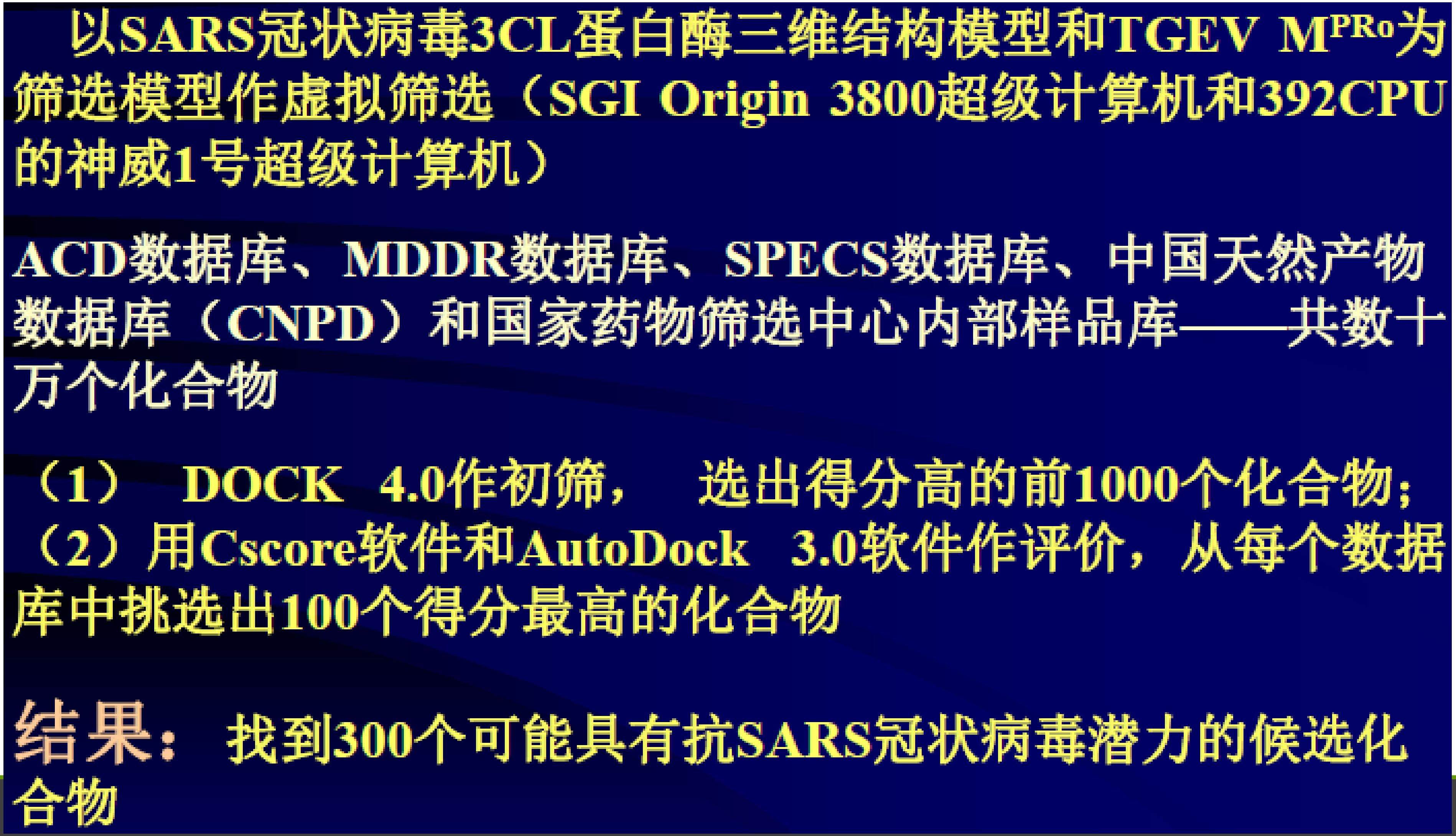
C. 蛋白酶抑制剂

两种蛋白酶的底物结合口袋的表面特征

∴ 3CL蛋白酶模建模型或TGEV M^{PRo}的晶体结构均可作为筛选抗SARS药物的结构模型

67

□ 步骤3. 虚拟筛选



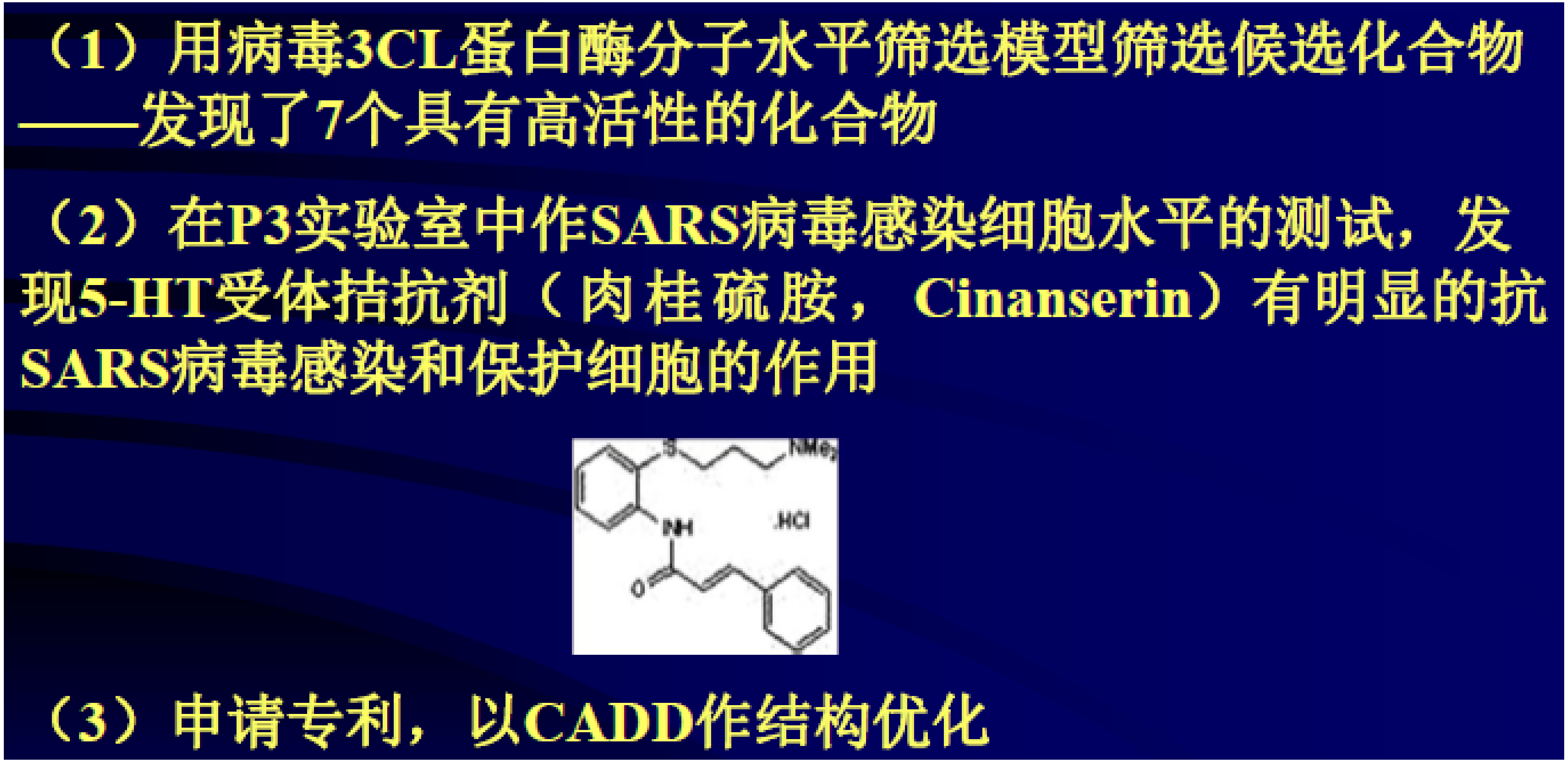
以SARS冠状病毒3CL蛋白酶三维结构模型和TGEV M^{PRo}为筛选模型作虚拟筛选（SGI Origin 3800超级计算机和392CPU的神威1号超级计算机）

ACD数据库、MDDR数据库、SPECS数据库、中国天然产物数据库（CNPD）和国家药物筛选中心内部样品库——共数十万个化合物

（1） DOCK 4.0作初筛， 选出得分高的前1000个化合物；
（2）用Cscore软件和AutoDock 3.0软件作评价，从每个数据库中挑选出100个得分最高的化合物

结果：找到300个可能具有抗SARS冠状病毒潜力的候选化合物

步骤4. 药理测试



（1）用病毒3CL蛋白酶分子水平筛选模型筛选候选化合物——发现了7个具有高活性的化合物

（2）在P3实验室中作SARS病毒感染细胞水平的测试，发现5-HT受体拮抗剂（肉桂硫胺，Cinanserin）有明显的抗SARS病毒感染和保护细胞的作用

S
NMe2
NH
.HCl
O

（3）申请专利，以CADD作结构优化