Theoretical Physics on Supercomputers

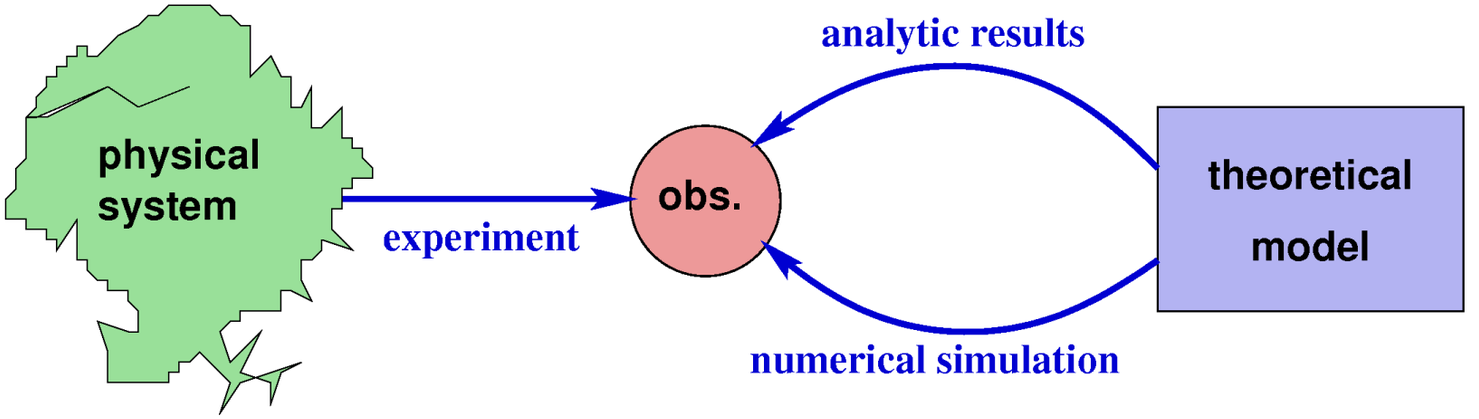
Hubert Simma

Università Milano Bicocca

Plan:

- ❑ Introduction
- ❑ Solution Steps
- ❑ Examples of Applications and Methods
- ❑ Lattice QCD
- ❑ Machines

Motivation



Three methodic pillars in natural sciences:

❒ experimental observation

. . . may be
- too expensive (M€)
- too dangerous (environment, humans)
- too difficult to measure (aerodynamics, astrophysics)
- not practical (cosmology, civil protection)
- controversal (biology)

❒ theoretical description (by first principles or phenomenological models)

. . . often impossible to solve by "analytic" methods (for realistic/relevant systems)

❒ numerical simulation . . .

Solution Process in Scientific Computing

Model:

representation of

- the system, e.g.
  - classical or quantum
  - degrees of freedom (particles or fields)
  - coordinates
  - constraints
- and its dynamics, e.g.
  - deterministic or stochastic
  - local balance (differential) or global conservation (integral)
  - boundary/initial values

Mathematical Problem:

- differential equations (ODE, PDE)
- extremal conditions
- integration (multi-dimensional, stochastic)
- algebraic relations

Solution Process in Scientific Computing

Discretisation:

approximation scheme to formulate mathematical problem in terms of a finite number of DOF and compute steps

- coordinates (time, space): "Finite Differences"
- partial solutions (plane waves, multipoles): "Finite Elements"
- sub-systems (particles, cells, ensembles)
- truncation of series expansions

Numerical Problem:

e.g. formulated as

- recurrence
- minimisation
- integration
- algebraic relations (implicit, explicit)
- linear systems

Solution Process in Scientific Computing

Algorithm:

method to solve numerical problem

- implicit or explicit
- direct or iterative
- adaptive
- stochastic

often determines

- resource requirements and balance
  - data storage
  - arithmetic operations
  - data transport
- data dependence (parallelisation)
- scaling with problem size
  (and machine parameters)

Computational Tasks:

sequence of solution steps

- with different levels of granularity,
  e.g.
  - evolve (sub-)system by one step
  - compute forces
  - solve $Ax = y$
  - compute eigenvalues of A
  - apply Ax (matrix $\times$ vector)
  - basic linear algebra, FFT,
    random numbers, . . .
- often described by pseudo-code
- without reference to a specific
  programming language
  or machine architecture

Solution Process in Scientific Computing

Implementation:

- high-level coding (human)
  - data representation
  - memory layout
  - selection and scheduling of (macro-)operations
  - management of communications
- low-level code generation (compiler)
  - management of memory accesses
  - register allocation
  - selection and scheduling of (micro-)instructions

Computer Code:

- sequence of machine instructions
- hardware dependent
- implementation dependent

Challenges

Fundamental research

* Astrophysics and Cosmology (gravitation, hydrodynamics)
* Climate and Meteorology (hydro- and thermodynamics)
* Chemistry, Biology (classical and quantum mechanics)
* Statistical Physics (critical phenomena, non-equilibrium dynamics)
* High-energy Physics (QCD)
* . . .
* Material Science
* Medicine
* Engineering (electro-, hydro-, thermodynamics)
* Computing (hardware and code optimisation)

Hamming: "The purpose of computing is not numbers but insight"

What makes simulations difficult?

- multiple scales, e.g. in
  - time $\longrightarrow$ multiple-time scale integration
  - space $\longrightarrow$ adaptive grids
  - energy

- coupled systems, e.g.
  - stars + gas (discrete + continuum)
  - molecules + solvent
  - different atoms or molecules
  - atoms + electrons (classical + quantum)
  - matter + fields (charges + radiation, masses + gravity, quarks + gluons)
  - fermions (Pauli principle!)

- problem size (degrees of freedom) e.g. imposed by
  - nature of the system (particles, fields)
  - numerical accuracy (grid points)

- parameter limits which are NOT directly accessible by simulations
  - often related to most interesting phenomena (e.g. phase transitions, singularities)

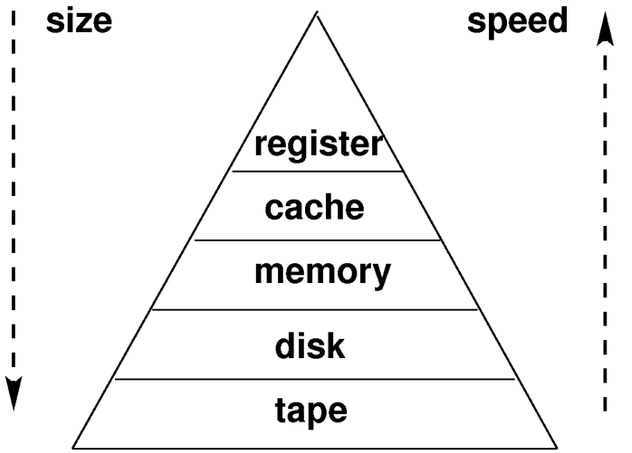
Limitations of Numerical Simulation

✘ costs
- infrastructure: 20. . . 50 M€ per Supercomputer (running 4 years)
- operation: $>$ 2000 €/day electricity bill (2.5 W/Gflops)
- environment: $>$ 10 t/day CO_2 emission (0.7 kg/kWh)

✘ limited processing speed

✘ limited storage resources

✘ numerical accuracy requires
- well-posed mathematical problem
- consistent discretisation
- stable numerical solution
- precise implementation



Error Sources

Computing the value of a function $f : \mathbb{R} \to \mathbb{R}$, we distinguish

- x = true value of input
- $\widetilde{x}$ = inexact input actually used
- $f(x)$ = desired result
- $\widehat{f}$ = approximate function exactly computed
- $\widetilde{f}$ = approximate function actually computed

Total error:

$$f(x) - \widetilde{f}(\widetilde{x}) = \underbrace{f(x) - f(\widetilde{x})}_{\text{propagated data error}} + \underbrace{f(\widetilde{x}) - \widetilde{f}(\widetilde{x})}_{\text{computational error}}$$

Propagated error: depends on conditioning of problem and stability of algorithm

Computational error: sum of

- Truncation error from approximation (e.g. discretisation) in actual algorithm (exact arithmetics)

$$f(\widetilde{x}) - \widehat{f}(\widetilde{x})$$

- Rounding error from inexact arithmetics in actual implementation

$$\widehat{f}(\widetilde{x}) - \widetilde{f}(\widetilde{x})$$

Example: Computational error in numerical differentiation

Compute

$$f(x) = F'(x)$$

by finite difference approximation

$$\widetilde{f}(x) = \frac{F(x+h) - F(x)}{h}$$

- truncation error is bound by

$$\frac{K}{2}h$$

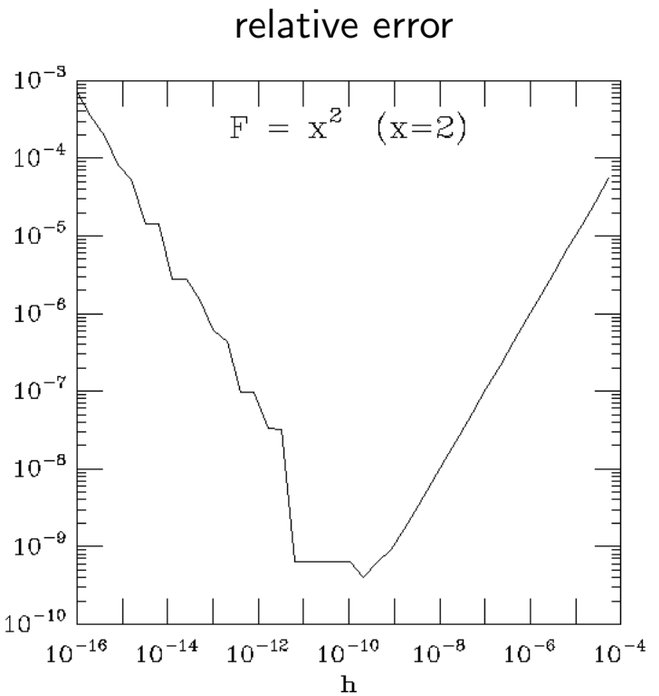
  if $|F''| \leq K$ near x

- rounding error is bound by

$$\frac{2}{h}\varepsilon_{mach}$$

$\Rightarrow$ total computational error is minimised when

$$h \approx 2\sqrt{\varepsilon_{mach}/K}$$



How much compute power is needed?

Example:

3-d lattice		classical MD	
number of sites	$1000^3 = 10^9$	number particles	10^5
variables per site	20	long-range forces	5×10^9
operations per variable	50	operations per force	20
iteration steps	100 000	time steps $fs \rightarrow ns$	10^6
total operations	10^{17}	total operations	10^{17}

Computer performance:

- 1 Mflops $= 10^6$ floating-point operations per second
- 1 Gflops $= 10^9$ floating-point operations per second
- 1 Tflops $= 10^{12}$ floating-point operations per second
- 1 Pflops $= 10^{15}$ floating-point operations per second

Computer resources:

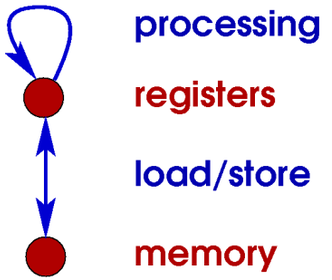
10^{17} FP operations = 1.2 Tflops $\times$ day = 30 Tflops $\times$ hour

How efficient is a Computer?

depends on

❒ **hardware characteristics**

- architectural structure
- storage devices: size σ
- processing/transport devices: bandwidth β



❒ **application signature**

- information exchange
- storage requirements
- parallelism

➜ Performance analysis and modeling

How much wall-clock time is (expected to be) required to solve a given task on a given architecture?

$$\mathbf{T} = \mathbf{f}\Big(\underset{\text{task}}{\underbrace{\text{algorithm}}} \;,\; \underset{\text{code}}{\underbrace{\text{implementation}}} \;,\; \text{machine} \Big)$$

Computer Hardware

Comparison: PC vs. Supercomputer

	PC	Supercomputer	factor
processors	2	1000 . . . 10000	$>$ 1'000
performance	30 Gflops	50 . . . 500 Tflops	$\geq$ 1'000
clock frequency	3 GHz	1. . . 3 GHz	$<$ 1
memory	10 GB	100 TB	$\geq$ 10'000
disk	500 GB	1000 TB	$\geq$ 2'000
storage	—	10 PB	(1'000'000 CDs)

Moore's Law [1965]

Processor complexity at minimal cost doubles every 18. . . 24 months

$$\#\text{gates} \;\propto\; 2^{\,t/\tau} \qquad (\tau = 18 \ldots 24 \text{ m})$$

Scaling of Chip Technology

- transistors/area $\sim 1/L^2$ (with feature size e.g. $L \approx 65\,nm$)
- clock $\sim V/L$ (with supply voltage e.g. $V \approx 1.8$ V)
- power/transistor $\sim V^2$

Examples of Supercomputers

See for instance (but not only): `http://www.top500.org`

#	site	machine	proc	net	cores [1000]	R_{max} [Tflops]	R_{peak} [Tflops]
1	LANL (US)	Roadrunner (IBM)	STI	InfB	122	1375	1028
2	LLNL (US)	BG/L (IBM)	IBM	3-D	212	478	560
3	Argonne (US)	BG/P (IBM)	IBM	3-D	164	450	557
4	U. Texas (US)	Blade (SUN)	AMD	InfB	63	326	503
5	Oak Ridge (US)	XT4 (Cray)	AMD	3-D	31	205	260
6	FZJ (Germany)	BG/P (IBM)	IBM	3-D	65	180	222
7	New Mexico (US)	Altix (SGI)	Intel	InfB	14	133	172
8	India	Blade (HP)	Intel	InfB	14	132	172
9	IDRIS (France)	BG/P (IBM)	IBM	3-D	40	112	139
10	Total (France)	Altix (SGI)	Intel	InfB	10	106	122

- Architecture: MPP / Blade Cluster
- Processor: multi-core (≥ 4)
- Network: Infiniband (switched) or proprieatary (3-d torus)

Usage of Top500 Supercomputers

Segments	Pflops	Systems
Industry	7.0	297
Research	6.0	91
Academic	3.7	98
Others	0.3	14

Areas	Pflops	Systems
Not Specified	7.0	134
Research	2.7	53
Finance	1.7	76
Geophysics	1.3	49
Service	.56	25
Semiconductor	.45	19
Hardware	.44	19
Info Service	.68	31
Info Processing	.30	9
Defense	.29	5
Benchmarking	.24	7
Energy	.24	8
Weather/Climate	.21	9
Media	.19	8
Telecomm	.16	5

Countries	Pflops	Systems
US	10	257
Germany	1.4	46
UK	1.4	53
France	1.0	34
Japan	0.7	22
Sweden	0.4	9
India	0.3	6
China	0.3	12
Russia	0.2	9
Spain	0.2	7
Italy	0.2	6

Application Examples

system	area	mathematical	numerical
mechanic	astro, molecular	ODE (Newton N-body) PDE (MHD, SPH)	Integrators Ewald Fast Multipole
electromagnetic	engineering	PDE (Maxwell) ODE (time-domain) EVP (frequency-domain)	FD, FE, FI, LinAlg Integrators FFT
hydrodynamic	turbolence, climate	PDE (Navier-Stokes, SPH) kinetic (Lattice Boltzmann)	FD, FE, FI, LinAlg Automata
statistical	phase transitions non-equilibrium	stochastic integration (MC)	Markov Process
quantum mech. (non-relativistic)	nuclei, atoms materials, chemistry	PDE (Schroedinger) Minimisation (DFT) stochastic integration (QMC)	CPMD FFT
quantum field (relativistic)	HEP, LQCD	path integrals	MC, MD, LinAlg

Example: Classical Molecular Dynamics

Mathematical Problem:

Equations of motion for point-like masses (atoms, molecules, stars, . . .) in force field

$$m_i \ddot{x}_i = F_i \qquad \text{or} \qquad \begin{cases} \dot{q} &= +\dfrac{\partial H}{\partial p} \\ \dot{p} &= -\dfrac{\partial H}{\partial q} \end{cases}$$

Numerical Methods:

- time integration
  - multi-time steps: separate scales of time evolution
  - adaptive time steps
- long-range forces (Coulomb, Gravitation)
  - Particle Mesh Ewald
    (split short and long distance ranges, summation in x or Fourier space)
  - Fast Multipole
- short-range forces
  - particle decomposition
  - space decomposition

N-body Simulations in Astrophysics

Computational Challenges:

- large number of particles: $N = 10^{8\cdots9}$ (but small for thermodynamic ensemble)
- high accuracy for long-range gravitational forces
  (no screening, infinite binding energy)
- different time scales (e.g. 2-body correlations of close binaries)

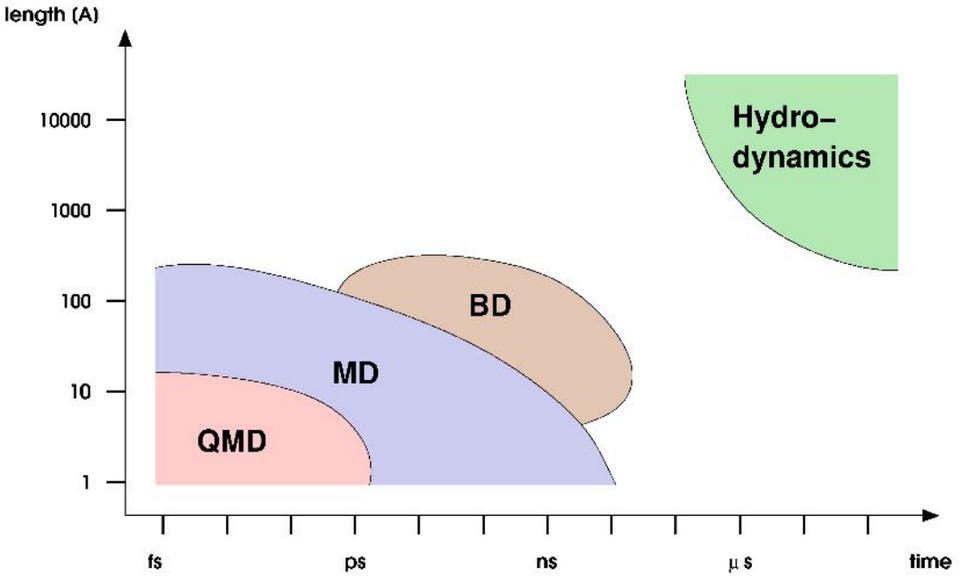
Computational Methods and Scaling:

- Exact direct N-body integration (PP): $O(N^2)$ $[+O(NN^*)]$
  - high-order integration: $\mathbf{r_i}, \mathbf{v_i}, \mathbf{a_i}, \dot{\mathbf{a}}_\mathbf{i}$
  - individual time-steps: $\Delta t = 2^{-n} t \approx 0.01 \frac{|\mathbf{a_i}|}{|\dot{\mathbf{a}}_\mathbf{i}|}$
  - neighbour lists
- Without direct gravitational forces: $O(N) + O(n_{mesh}, nlm)$
  - Particle Mesh (PM): particles $\rightarrow$ density $\rightarrow$ Poisson (FFT)
  - Fast Multipole
- Mixed codes: $O(N \log N)$
  - distance grouping (tree, $\mathrm{P^3M}$)
  - gas component with Smoothed Particle Hydrodynamics (SPH)
    $f(\mathbf{r}) \rightarrow \int dV' \, f(\mathbf{r}') \cdot W(|\mathbf{r} - \mathbf{r}'|, h)$

Simulation of Microscopic Systems

Physics Areas:

- material science (solid states, liquids, plasmas)
- quantum chemistry (catalizers, polymers)
- biophysics (macro-molecules with solvent)



Theoretical Descriptions:

- Hydrodynamics:
  dynamics of effective parameters
- Brownian Dynamics:
  particles in effective media (solvent)
- Classical Molecular Dynamics (MD): $N = 10^{3\ldots9}$, $T =$ ns . . . μs
  particles with effective potentials (ab initio or from experiment)
- Quantum Molecular Dynamics: $N = O(100)$, $T = 10$ ps
  solve (approximate) Schrödinger equation

Method: Integration of ODE

Consider n-dimensional system of explicit ODE

$$y' = F(t, y(t)) \ , \qquad y(0) = y_0 \qquad (y \in \mathbb{R}^n)$$

❒ Higher order ODE can be reduced by introducing further variables

$$y'' = F(y, y') \quad \Leftrightarrow \quad \begin{pmatrix} y_0' \\ y_1' \end{pmatrix} = \begin{pmatrix} y_1 \\ F(y_0, y_1) \end{pmatrix}$$

❒ Let λ_i denote eigenvalues of the Jacobian $\partial F_i / \partial y_j$, then the (exact) solution is

- stable if $Re(\lambda_i) \leq 0$ for all λ_i
- unstable if $Re(\lambda_i) > 0$ for any λ_i

❒ Numerical Solution Schemes:

For discrete time steps $t_k = k \cdot h$, iterate an integration scheme, which can be

- explicit

$$\widehat{y}_{k+1} = g(t_{k+1}, \widehat{y}_k, \widehat{y}_{k-1}, \ldots)$$

- implicit

$$g(t_{k+1}, \widehat{y}_{k+1}, \widehat{y}_k, \widehat{y}_{k-1}, \ldots) = 0$$

Implicit schemes are more stable, but need solution of algebraic equation

Truncation Errors from Discretisation

- The exact solution $y(t) \equiv \phi(t, y_0)$ defines a flow

$$\phi : \mathbb{R} \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$$

  and a time-evolution map $\phi_t(y_0) \equiv \phi(t, y_0)$

- **Local Truncation Error:**

$$\tau_k^{loc} \equiv \widehat{y}_k - \phi(t_k, \widehat{y}_{k-1})$$

- **Global Truncation Error:**

$$\tau_k^{glob} \equiv \widehat{y}_k - \phi(t_k, \widehat{y}_0)$$

If $\tau_k^{loc} = O(h^{p+1})$, then p is called the order of the integrator

Examples of Integration Methods

❒ **Explicit Euler Method:** $p = 1$

$$\widehat{y}_{k+1} = \widehat{y}_k + h \cdot F(\widehat{y}_k)$$

❒ **Implicit Euler Method (Backward Euler):** $p = 1$, stable for any $h > 0$

$$\widehat{y}_{k+1} = \widehat{y}_k + h \cdot F(\widehat{y}_{k+1})$$

Example:

$$\boxed{y' = -\lambda y \quad \text{with} \quad y(0) = y_0 \quad \text{and} \quad \lambda > 0} \quad \Rightarrow \quad \phi(t, y_0) = y_0 \cdot e^{-\lambda t}$$

- Explicit Euler:

$$\widehat{y}_n = (1 - \lambda h) \cdot \widehat{y}_{n-1} \quad = (1 - \lambda h)^n \cdot y_0 \rightarrow \begin{cases} +0 & \text{if } 0 < h \leq 1/\lambda \quad \text{(undershooting)} \\ \pm 0 & \text{if } 1/\lambda < h \leq 2/\lambda \quad \text{(oscillating)} \\ \pm\infty & \text{if } 2/\lambda < h \quad \text{(divergent)} \end{cases}$$

➜ numerical solution $\widehat{y}$ is only stable for $|1 - \lambda h| < 1$

- Implicit Euler:

$$\widehat{y}_n = \widehat{y}_{n-1} - \lambda h \cdot \widehat{y}_n \quad = y_0 / (1 + \lambda h)^n$$

Properties of Computational Methods

Definitions:

- Well-posed problem: existence of unique solution with continuous dependence on problem data (e.g. Lifschitz bound, or well-conditioned)

- Consistent method: discretisation becomes exact

$$\max_n |\widehat{y}_n - \phi(\widehat{y}_{n-1})| \longrightarrow 0 \quad (h \to 0)$$

  i.e. local truncation error vanishes (original vs. discrete problem)

- Convergent method: solution becomes exact

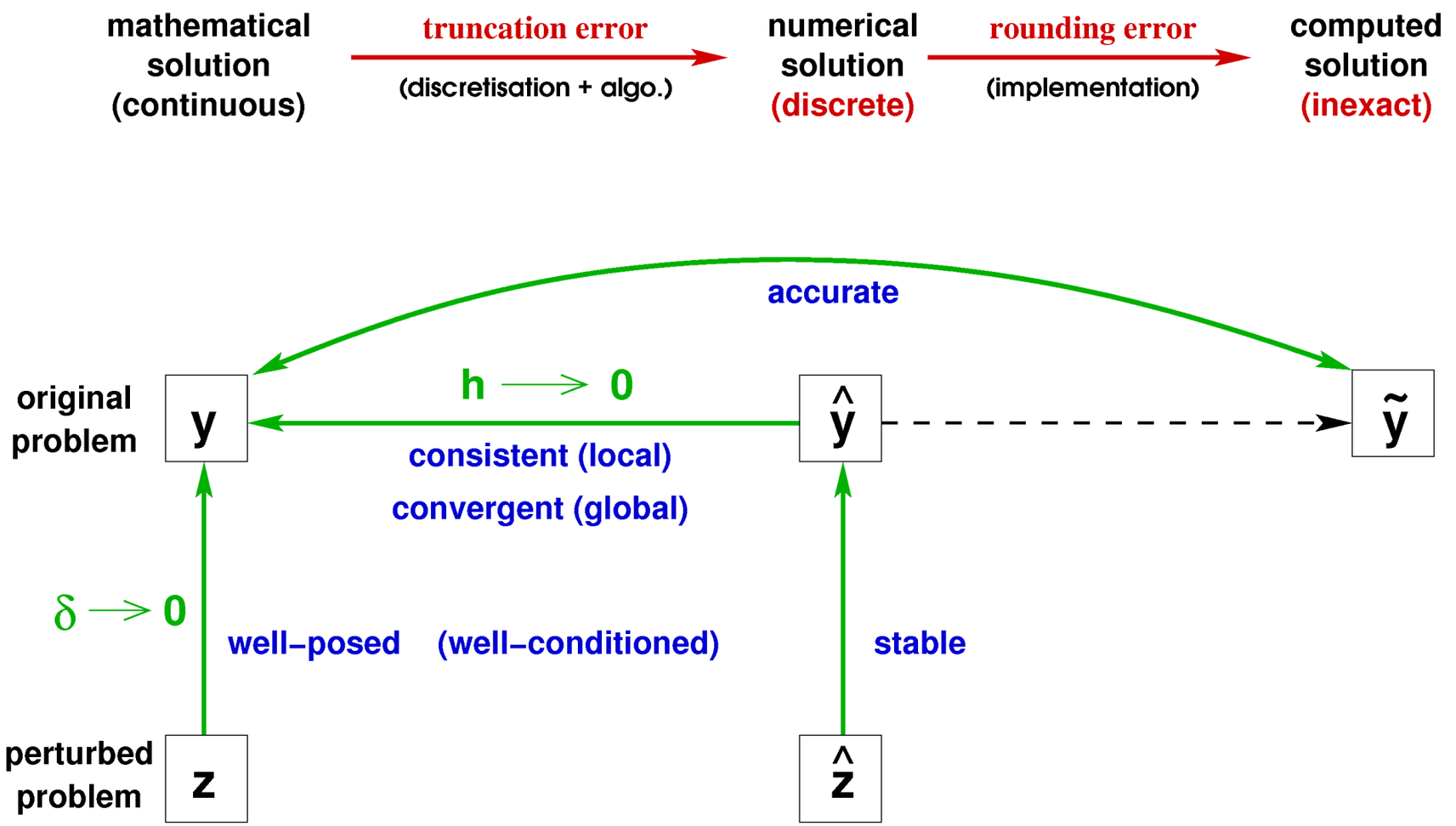
$$\max_n |\widehat{y}_n - \phi(y_0)| \longrightarrow 0 \quad (h \to 0)$$

  i.e. global truncation error vanishes (original vs. discrete problem)

- Stable method/algorithm: (analogous to well-conditioned problem)

$$\forall \epsilon \; \exists K : \max_n |\widehat{z}_n - \widehat{y}_n| < K$$

  for the solution $\widehat{z}$ of perturbed problem with small perturbation $\delta < \epsilon$



Relations: (for finite difference methods)

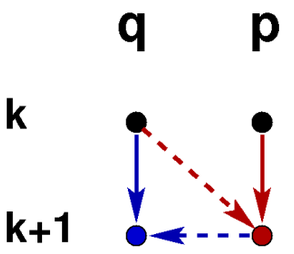
- For well-posed problem and consistent method: convergent $\Leftrightarrow$ stable
- well-conditioned problem + stable algorithm $\Rightarrow$ accurate

Simple Symplectic Integrators

Consider

$$y = \begin{pmatrix} q \\ p \end{pmatrix} \quad F(p,q) = \begin{pmatrix} f(p) \\ g(q) \end{pmatrix}$$

❒ **Symplectic Euler (Euler-Cromer):** $p = 1$



$$\widehat{y}_{k+1} \equiv \begin{pmatrix} q_{k+1} \\ p_{k+1} \end{pmatrix} = \begin{pmatrix} q_k + h \cdot f(p_{k+1}) \\ p_k + h \cdot g(q_k) \end{pmatrix}$$

is area preserving (geometric integrator)
i.e. Jacobian $\mathbf{Y} \equiv \partial y_{k+1}/\partial y_k$ of the map $y_k \mapsto y_{k+1}$ satisfies

$$det(\mathbf{Y}) = 1$$

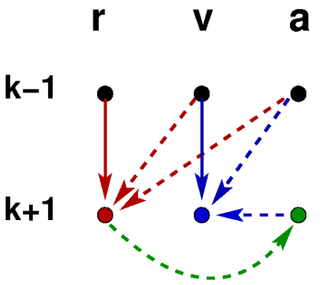
and symplectic

$$\mathbf{Y}^T \cdot \mathbf{J} \cdot \mathbf{Y} = \mathbf{J} \quad \text{with} \quad \mathbf{J} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

Simple Symplectic Integrators (cont.)

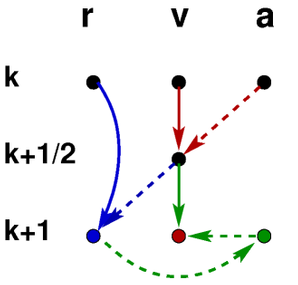
❒ **Velocity Verlet:** $p = 2$, and $\tau_k^{glob} = O(h^2)$

$$\begin{aligned} \mathbf{r}_{k+1} &= \mathbf{r}_k + \mathbf{v}_k \cdot h + \frac{h^2}{2} \cdot \mathbf{a}_k \\ \mathbf{v}_{k+1} &= \mathbf{v}_k + \frac{1}{2}(\mathbf{a}_k + \mathbf{a}_{k+1}) \cdot h \end{aligned}$$



❒ **Equivalent: Leapfrog Implementation**

$$\begin{aligned} \mathbf{v}_{k+1/2} &= \mathbf{v}_k + \frac{h}{2}\mathbf{a}_k \\ \mathbf{r}_{k+1} &= \mathbf{r}_k + \mathbf{v}_{k+1/2} \cdot h \\ \mathbf{v}_{k+1} &= \mathbf{v}_{k+1/2} + \frac{h}{2}\mathbf{a}_{k+1} \end{aligned}$$



Hamiltonian Systems

For a Hamiltonian system

$$\dot{y} = \{y, H(y)\} \quad \text{with} \quad y \equiv \begin{pmatrix} q \\ p \end{pmatrix} \quad \text{and} \quad \{f, g\} \equiv \frac{\partial f}{\partial p}\frac{\partial g}{\partial q} - \frac{\partial g}{\partial p}\frac{\partial f}{\partial q}$$

the time evolution has the exact formal solution

$$y(t+h) \equiv \mathbb{E}_H(h) \cdot y(t) \;=\; \exp\big(h \cdot \mathbb{L}_H\big) \cdot y(t)$$

with the linear Liouville operator $\mathbb{L}_H f \equiv \{f, H\}$

In particular, for a Hamiltonian of the form

$$H(q,p) = \frac{1}{2}p^2 + V(q)$$

we have

$$\mathbb{L}_H = \mathbb{Q} + \mathbb{P} \quad \text{with} \quad \mathbb{Q} = p \cdot \frac{\partial}{\partial q} \quad \text{and} \quad \mathbb{P} = -V' \frac{\partial}{\partial p}$$

and the associated time evolution operators

$$\begin{aligned} \mathbb{E}_Q(h) \equiv \exp(h\mathbb{Q}): \quad f(q,p) \quad &\mapsto \quad f(q{+}h \cdot p \;,\; p) \\ \mathbb{E}_P(h) \equiv \exp(h\mathbb{P}): \quad f(q,p) \quad &\mapsto \quad f(q \;,\; p{-}h \cdot V') \end{aligned}$$

are area preserving and symplectic

Building Symplectic Integrators

Idea: Repeatedly perform integration steps of size h using a product

$$\mathbb{S}(h) \equiv \mathbb{E}_P(c_1 h) \cdot \mathbb{E}_Q(d_1 h) \cdot \mathbb{E}_P(c_2 h) \cdots \mathbb{E}_P(c_n h) \cdot \mathbb{E}_Q(d_n h)$$

to approximate the exact time evolution

$$\exp\left(h\mathbb{L}_H\right) = \mathbb{S}(h) + O(h^{p+1})$$

- The coefficients can be determined by Baker-Campbell-Hausdorff formula

$$e^A e^B = exp(A + B + \frac{1}{2}[A, B] + \cdots)$$

e.g.

$$\begin{array}{lll} p = 1: & c_1 = d_1 = 1 & \text{(symplectic Euler)} \\ p = 2: & c_1 = c_2 = 1/2,\ d_1 = 1 & \text{(Leapfrog PQP)} \end{array}$$

- Leapfrog integrator $\mathbb{E}_P(h/2) \cdot \mathbb{E}_Q(h) \cdot \mathbb{E}_P(h/2)$ is also time-reversible, i.e. satisfies

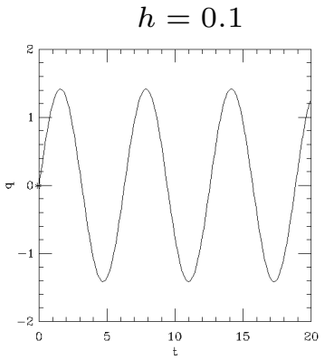
$$\mathbb{S}(-h) = \mathbb{S}^{-1}(h)$$

- All symplectic integrators have an exactly conserved Hamiltonian, i.e.

$$\exists K_h(p,q): \quad \mathbb{S}(h) = \exp\left(h\mathbb{L}_{K_h}\right) \quad \text{and} \quad K_h = H + O(h^p)$$

Example: Harmonic Oscillator

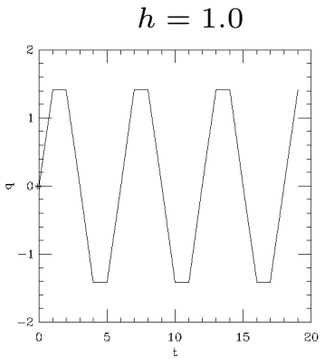
$$H(q,p) = \frac{1}{2}(p^2 + q^2)$$



Exact solution

$$\mathbb{E}_H(h) = \begin{pmatrix} \cos h & \sin h \\ -\sin h & \cos h \end{pmatrix}$$

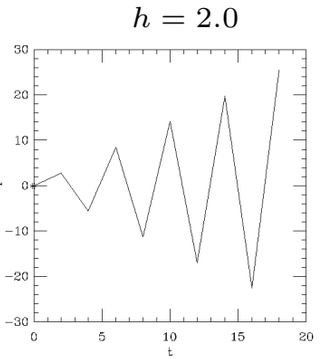
Generators of individual time-evolution steps



$$\mathbb{Q}(h) = \begin{pmatrix} 0 & h \\ 0 & 0 \end{pmatrix} \quad \Rightarrow \quad e^{h\mathbb{Q}} = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}$$

$$\mathbb{P}(h) = \begin{pmatrix} 0 & 0 \\ -\frac{V'}{q}h & 0 \end{pmatrix} \quad \Rightarrow \quad e^{h\mathbb{P}} = \begin{pmatrix} 1 & 0 \\ -h & 1 \end{pmatrix}$$

Symplectic Euler Integrator:



$$e^{h\mathbb{Q}} \cdot e^{h\mathbb{P}} = \begin{pmatrix} 1 - h^2 & -h \\ h & 1 \end{pmatrix} = \mathbb{E}_H(h) + O(h^2)$$

Leapfrog Integrator:

$$e^{\frac{h}{2}\mathbb{P}} \cdot e^{h\mathbb{Q}} \cdot e^{\frac{h}{2}\mathbb{P}} = \begin{pmatrix} 1 - h^2/2 & h \\ -h + h^3/4 & 1 - h^2/2 \end{pmatrix} = \mathbb{E}_H(h) + O(h^3)$$

N.B.: For $h > 2$, LF integrator has positive real eigenvalue $\Rightarrow$ can become instable

Multiple Timescales

If Hamiltonian has the form

$$H(p,q) = \frac{p^2}{2} + V_1(q) + V_2(q)$$

with

- forces from V_1 large (but cheap to compute)
- forces from V_2 small (and expensive to compute)

split $\mathbb{P} = \mathbb{P}_1 + \mathbb{P}_2$ accordingly

$$\mathbb{P}_i = -V_i' \frac{\partial}{\partial p}$$

and consider symplectic integrator of the form

$$\mathbb{S}(h,n) = \left[\mathbb{E}_{P_1}\left(\frac{h}{2n}\right) \cdot \mathbb{E}_Q\left(\frac{h}{2n}\right)\right]^n \cdot \mathbb{E}_{P_2}(h) \cdot \left[\mathbb{E}_Q\left(\frac{h}{2n}\right) \cdot \mathbb{E}_{P_1}\left(\frac{h}{2n}\right)\right]^n$$

Example: Classical MD with effective potentials

- Intra-molecular forces:
  - stretching

$$V_s = \sum_{bonds} \frac{K_b}{2}(r_b - r_{0b})^2$$

  - bending

$$V_b = \sum_{angles} \frac{K_a}{2}(\theta_a - \theta_{0a})^2$$

  - torsion and non-bond

- Inter-molecular forces: Coulomb or Lennard-Jones

$$V_{LJ} = \sum_{pairs} \sum_{i=1}^{N} \sum_{j=1}^{N'} 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right]$$

- Multi-time step integration: fast evolution only with $V_s + V_b$

Computational Cost:

task	scaling	operations	weight$^{*)}$
update x or v	MNS	$O(100)$	2 %
forces (intra, fast)	MNS	$O(1000)$	9 %
forces (intra, slow)	MN^2	$O(1000)$	2 %
forces (inter, slow)	M^2N^2P	$O(2000)$	85 %
communication	MNP	$O(10)$	1 %

$*)$ weight in total execution time for $M = 64$, $N = 64$, $P = 8$, $S = 8$

with

- $P =$ number of processors
- $M =$ number of molecules / P
- $N =$ number of atoms per molecule
- $S =$ number of fast time steps per slow step

Parallelisation Strategies:

❒ Particle Decomposition: $PID = f(i)$ ($\rightarrow$ all-to-all communications)

❒ Space Decomposition: $PID = f(x_i)$ ($\rightarrow$ nearest-neighbour communications)

Parallel Computation of long-range Forces

Systolic Algorithm:

- Initialize auxiliary arrays $\widehat{\mathbf{x}} \leftarrow \mathbf{x}$ and $\mathbf{y} \leftarrow f(\mathbf{x}, \mathbf{x})$
- For $k = 1, \ldots, P-1$
  - move $\widehat{\mathbf{x}}$ to next processor in ring (in parallel on each processor)
  - compute forces $f(\widehat{\mathbf{x}}, \mathbf{x})$ and accumulate them in $\mathbf{y}$ (in parallel on each processor)

Improvements:

- Avoid redundant force computations

$$f(x_i, x_j) = -f(x_j, x_i)$$

  by moving auxiliary force array
  (Half-Orrery Algorithm)

- Avoid redundant communication steps by storing data which passed
  (Hypersystolic Algoritm)
- Exploit all concurrent communication links of multi-dimensional torus network

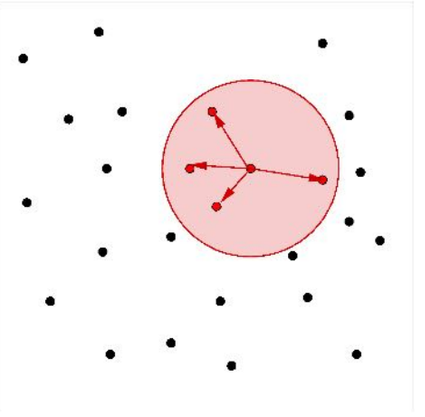
k	Processor / Data (P=8)							
0	**0**	1	2	3	4	5	6	7
1	7	**0**	1	2	3	4	5	6
2	6	7	**0**	1	2	3	4	5
3	5	6	7	**0**	1	2	3	4
4	4	5	6	7	**0**	1	2	3
5	3	4	5	6	7	**0**	1	2
7	2	3	4	5	6	7	**0**	1
6	1	2	3	4	5	6	7	**0**

Neighbour Lists for Computation of short-range Forces

In multi-particle simulations, like MD or Astrophysics, one may need lists to keep track of the particles to be taken into account in force computation

❒ Verlet List:

$$\forall \text{ particles } i\text{: } L_i = \{(j, x_j) : |x_j - x_i| < R + \delta\}$$

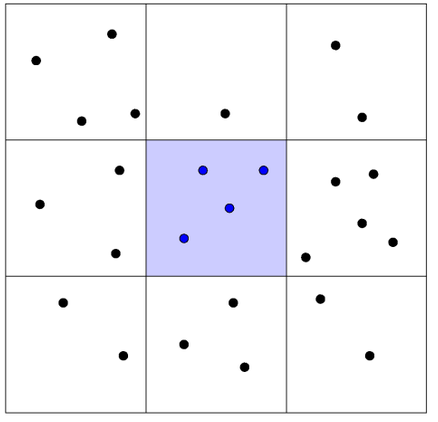


\+ force computation cost $N(N^* - 1)$
– update cost $O(N^2)$ e.g. each time when $\exists \Delta x > \delta/2$
– storage $O(3N \cdot N^*)$

❒ Linked Cell List:

Decompose space domain into cells with linear extent $R + \delta$

$$\forall \text{ cells } c\text{: } L_c = \{j : x_j \in \text{cell } c\}$$



\+ force computation only for particles in neighbour cells
\+ update $O(N)$ e.g. at each force computation
\+ storage $O(N)$
– inefficient if less than 4 cells in each direction ($L < 4R$)

Ab initio MD of microscopic systems

Quantum Mechanical Dynamics:

$$i\hbar\frac{\partial}{\partial t}\Phi(R_I, r_i, t) = H\Phi(R_I, r_i, t)$$

with

$$\begin{aligned} H &= \sum_I T_I + \sum_i T_i + V_{n-e}(R_I, r_i) \\ &= \sum_I T_I + H_e(R_I, r_i) \end{aligned}$$

Product Ansatz: separation of electronic and nuclear wave functions

$$\Phi(R_I, r_i, t) \approx \psi(r_i, t) \cdot \chi(R_I, t) \cdot \text{phase}$$

$\Rightarrow$ time-dependent self-consistent field (TDSCF) method: mean field

$$\left\{ \begin{array}{lll} \chi(R_I, t): & V \sim \langle\psi|V_{n-e}|\psi\rangle & \rightarrow \text{MD} \quad (\text{atoms}) \\ \psi(r_i, t): & V \sim \langle\chi|H_e|\chi\rangle & \rightarrow \text{QM, DFT} \quad (\text{electrons}) \end{array} \right.$$

Strategies for solution of coupled Ion-Electron system

- $\forall R_I$ solve electronic structure : Born-Oppenheimer MD
  classical MD on global potential energy surface (from time-independent el. structure)
  Problem: dimensional bottleneck $\sim 10^{3N-6}$
  $\rightarrow$ replace by parametrised effective potential

- $\forall t$ solve both equations on-the-fly: Ehrenfest MD
  coupled classical MD + time-dependent electronic structure
  Problem: different time scales (10^{12} steps of 1 fs for 1 ms)

- map both on classical problem: Car-Parrinello MD

$$\mathcal{L}[R,\psi] \sim \underbrace{\frac{\mu}{2}\langle\dot{\psi}|\dot{\psi}\rangle}_{T_e} + \underbrace{\frac{M}{2}\dot{R}_I^2}_{T_I} - \underbrace{\langle\psi|H_e|\psi\rangle}_{V[R_I,\psi]} + \text{constraints}$$

  - conserved energy: $T_e + T_I + V$
  - physical energy: $T_I + V$
  - (fictitious) electronic temperature = thickness of BO surface $\propto T_e \rightarrow 0$

Quantum-Mechanical Electronic Structure

Problem: anti-symmetry of many-electron wave functions (Pauli Principle)

Map exact many-body problem with Hamiltonian

$$H = T_e + V_{ee} + V_{ext}$$

on single-electron problem with suitable approximations to treat anti-symmetry

❒ **Hartree-Fock (HF)**

Single Slater determinant (not linear combination)

$$\Psi(\mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_N}) = det\left[\psi_1(\mathbf{r_1}) \cdots \psi_N(\mathbf{r_N})\right]$$

where each factor (orbital) satisfies single-electron Schrödinger equations with

$$H_s = -\frac{1}{2}\nabla^2 + H_{ee} + V_{ext}$$

and non-local two-electron terms H_{ee} (Coulomb interactions and exchange)

❒ **Configuration Interactions (CI)**

Include also correlation effects from linear combinations of Slater determinants

Density Functional Theory

Basic Idea: Hohenberg-Kohn Theorems

- $\exists$ one-to-one mapping between electron density and wavefunction for GS of many-particle system
- GS density minimizes total electronic energy of system

i.e. electron density that minimizes true energy functional (unknown) describes all we can know about electronic structure of GS

$$\begin{array}{ccc} V_{ext} & \longrightarrow & |\Psi_{GS}\rangle \\ & \nwarrow\!\!\searrow & \downarrow \\ & & n_{GS}(r) = |\Psi_{GS}|^2 \end{array}$$

Intuitive Argument: "cusp condition" for spherical average $\langle n(r)\rangle$

$$Z_{nucl} = -\frac{1}{2\langle n(0)\rangle}\left[\frac{\partial\langle n(r_i)\rangle}{\partial r_i}\right]_{r_i=0}$$

Car Parrinello MD

- Consider density functional of fictitious single-electron system

$$E_s[n] = \langle \psi_s[n] | (T_s + V_s) | \psi_s[n] \rangle$$

with

$$V_s = V_{ions} + V_{coulomb}[n] + E_{xc}[n]$$

and a local approximation for exchange energy E_{xc}

- Minimisation by self-consistent solution of Kohn-Sham equations for orthogonal orbitals (possibly expanded in convenient base)

$$\left[-\frac{1}{2}\nabla^2 + V_s(r) \right] \psi_i = \epsilon_i \cdot \psi_i \quad \text{with} \quad n \equiv \sum |\psi_i|^2$$

- Car-Parrinello evolution:

$$\begin{aligned} M_I \, \ddot{R}_I(t) &= -\frac{\partial}{\partial R_I}(E + \text{constraints}) \\ \mu \, \ddot{\psi}_i(t) &= -\frac{\partial}{\partial \psi_i^*}(E + \text{constraints}) \end{aligned}$$

Theoretical Physics on Supercomputers

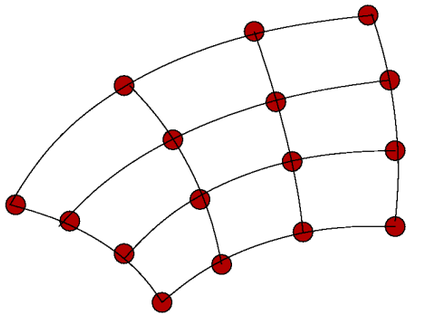
Hubert Simma

Università Milano Bicocca

Plan:

- ❑ Introduction
- ❑ Solution Steps
- ❑ Examples of Applications and Methods
- ❑ Lattice QCD
- ❑ Machines

PDE: Discretisation Techniques

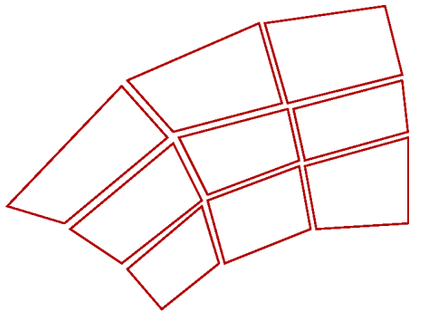


❒ Finite differences (FD)

DOF = values of solution on mesh points

➜ solve equations from FD approximation of derivatives

- simple to derive and effective
- limited to (block-)structured meshes
- moderate accuracy

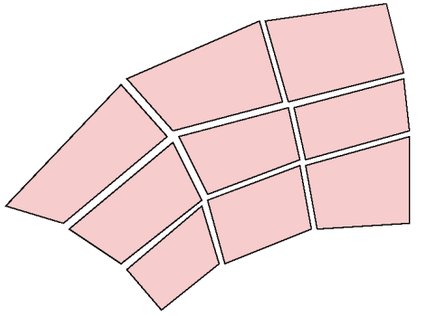


❒ Finite volumes/integrals (FI)

DOF = integrals of solution over mesh links/surfaces

➜ solve equations from integral of PDE on mesh cells (local)

- conservation laws by construction
- arbitrary meshes



❒ Finite elements (FE)

DOF = basis functions on mesh cells (elements)

➜ solve equations from minimisation of a functional (global)

- very flexible and general
- arbitrary meshes

FD Method

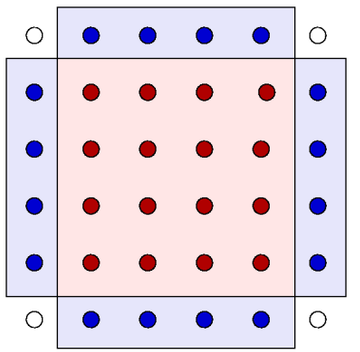
FD approximations:

$$\frac{\partial u}{\partial x} \longrightarrow \frac{u(x+h)-u(x)}{h} + O(h)$$

$$\frac{\partial^2 u}{(\partial x)^2} \longrightarrow \frac{u(x-h)-2u(x)+u(x+h)}{h^2} + O(h^2)$$

Example: Laplace operator with Dirichlet BC

$$\begin{aligned} \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \cdots &= 0 \quad (\underline{x} \in \mathcal{D}) \\ u(x) &= f(x) \quad (\underline{x} \in \partial\mathcal{D}) \end{aligned}$$

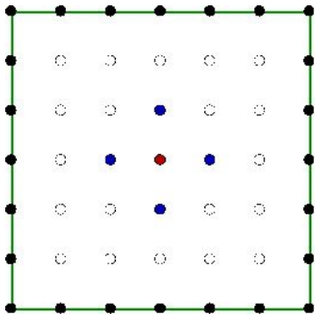


yields linear equation system with symmetric matrix:

$$\left(\begin{array}{ccc|ccc|ccc} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{array}\right) \left(\begin{array}{c} u_{11} \\ u_{12} \\ u_{13} \\ \hline u_{21} \\ u_{22} \\ u_{23} \\ \hline u_{31} \\ u_{32} \\ u_{33} \end{array}\right) = \left(\begin{array}{c} f_{01}+f_{10} \\ f_{02} \\ f_{14}+f_{03} \\ \hline f_{20} \\ 0 \\ f_{24} \\ \hline f_{30}+f_{41} \\ f_{42} \\ f_{43}+f_{34} \end{array}\right)$$

FD Methods for 2nd order quasi-linear PDE

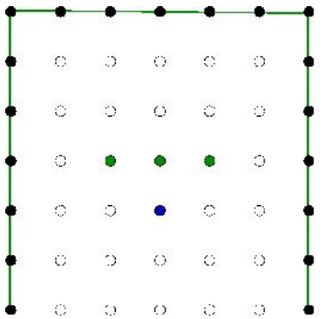
$$a\,\partial_x\partial_x\,u + b\,\partial_x\partial_y\,u + c\,\partial_y\partial_y\,u + = f(u, \partial_x\,u, \partial_y\,u)$$



❒ **Elliptic**: $b^2 - 4ac < 0$ (e.g. Laplace Equation)
- BV on all boundaries (steady states)
- implicit five-point difference formula

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = 0$$

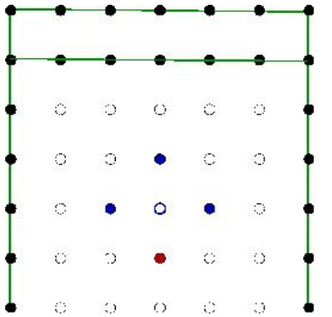
yields linear equation system with symmetric matrix



❒ **Parabolic**: $b^2 - 4ac = 0$ (e.g. Heat Equation)
- IV + BV
- explicit solution of recursion

$$u_{i,j+1} = (1 - 2r)u_{i,j} + r(u_{i+1,j} + u_{i-1,j})$$

stable for $r \equiv h_t/h_x^2 \leq 1/2$
- implicit methods (Crank-Nichelson)



❒ **Hyperbolic**: $b^2 - 4ac > 0$ (e.g. Wave Equation)
- IV for u and $\partial_t u$ + BV
- explicit solution of recursion

$$u_{i,j+1} = (2 - 2r^2)u_{i,j} + r^2(u_{i+1,j} + u_{i-1,j}) - u_{i,j-1}$$

stable for $r \equiv h_t/h_x \leq 1$

Krylov Subspace Methods for Iterative Solvers

General Projection Method to solve

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

- Find approximation $\mathbf{x}_n$ in subspace $\mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)$ with Krylov-subspace

$$\mathcal{K}_n(\mathbf{A}, \mathbf{v}) \equiv span\{\mathbf{v}, \mathbf{A}\mathbf{v}, \mathbf{A}^2\mathbf{v}, \ldots, \mathbf{A}^{m-1}\mathbf{v}\}$$

  such that residual $\mathbf{r}_n \equiv \mathbf{b} - \mathbf{A}\mathbf{x}_n$ satisfies (Petrov-Galerkin condition)

$$\mathbf{r}_n \perp \mathcal{L}_n$$

  for some other subspace $\mathcal{L}_n$ of dimension n, e.g. $\mathcal{L}_n = \mathcal{K}_n$, or $\mathbf{A}\mathcal{K}_n$

- If A symmetric positive definite use $\mathcal{L} = \mathcal{K}_n$:

  orthogonal projection $\Leftrightarrow$ x minizes $\|x - x_*\|_{\mathbf{A}}$ with $x_* = A^{-1}b$

- Perform iteration

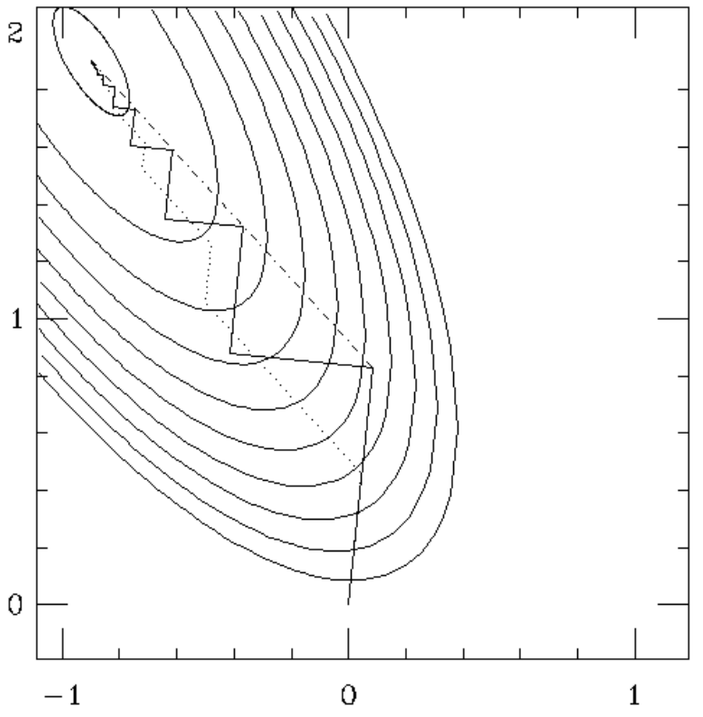
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \cdot \mathbf{s}_k$$

  with α_k deterimed by minimisation of $f(x) = \frac{1}{2}(x, \mathbf{A}x) - (x, b)$ along search directions $\mathbf{s}_k$

- $\mathbf{x}_k$ and $\mathbf{s}_k$ can be built up such that $\mathbf{x}_{k+1}$ minimises $f(x)$ also over whole space of previous directions $\{\mathbf{s}_1, \ldots, \mathbf{s}_k\}$

Illustration in 2 dimensions:

$$\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \quad , \quad \mathbf{b} = \begin{pmatrix} 0.1 \\ 1 \end{pmatrix} \quad , \quad \mathbf{x_0} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad , \quad \mathbf{x}_* \equiv \mathbf{A}^{-1}\mathbf{b} = \begin{pmatrix} -0.9 \\ 1.9 \end{pmatrix}$$



Algorithm	$\mathcal{K}$	$\mathcal{L}$	Minimisation	Iterations	$\|r\|$
Steepest Descend	$\mathcal{K}_1(r_n)$	$\mathcal{K}_1(r_n)$	$\|x - x_*\|_{\mathbf{A}}$	43	2.3×10^{-6}
Minimal Residual	$\mathcal{K}_1(r_n)$	$\mathcal{K}_1(\mathbf{A}r_n)$	$\|b - \mathbf{A}x\|_2$	33	2.3×10^{-6}
Conjugate Gradient	$\mathcal{K}_n(r_0)$	$\mathcal{K}_n(r_0)$	$\|x - x_*\|_{\mathbf{A}}$	2	2.1×10^{-16}

Example: FI Method in Electrodynamics

Mathematical Problem: Maxwell Equations

$$\partial_\mu F^{\mu\nu} = j^\nu \qquad \text{or} \qquad \begin{cases} rot\, E &= -\dot{B} \\ rot\, H &= +\dot{D} + J_e \\ div\, B &= 0 \\ div\, D &= \rho_e \end{cases}$$

Computational Method: Finite Integration Technique (FI)

- Discretisation on dual grids

		G		$\widetilde{G}$
volumes	$G = \cup V_i$	V_i	$\leftrightarrow$	$\widetilde{P}_i$
areas	$A_i = \cap V_i$	A_i	$\leftrightarrow$	$\widetilde{L}_i$
links	$L_i = \cap A_i$	L_i	$\leftrightarrow$	$\widetilde{A}_i$
points	$P_i = \cap L_i$	P_i	$\leftrightarrow$	$\widetilde{V}_i$

- Basic variables:

$$e_i \equiv \int_{L_i} E \cdot ds \qquad\qquad b_i \equiv \int_{A_i} B \cdot dA$$

electric voltage along links L_i magnetic flux through areas A_i

- matrices of difference operators

$$\begin{aligned} S &\leftrightarrow div \\ C &\leftrightarrow rot \end{aligned}$$

exactly preserve algebraic relations, like

$$\begin{aligned} SC = \widetilde{S}\widetilde{C} &= 0 \quad (div\ rot = 0) \\ C\widetilde{S}^T = \widetilde{C}S^T &= 0 \quad (rot\ grad = 0) \end{aligned}$$

- Maxwell Grid Equations

$$\begin{aligned} \int E \cdot ds &= -\int \dot{B} \cdot dA & \qquad Ce &= -\dot{b} \\ \int H \cdot ds &= \int (\dot{D} + j_e) \cdot dA & \qquad \widetilde{C}h &= \dot{d} + i_e \\ \int B \cdot dA &= 0 & \qquad Sb &= 0 \\ \int D \cdot dA &= \int \rho dV & \qquad \widetilde{S}d &= q_e \end{aligned}$$

- Can be coupled to dynamics of charges (PIC)
- Time-dependence can be treated in time or frequency domain
- Cost: $O(10^{13})$ FP operations for 10^9 grid points

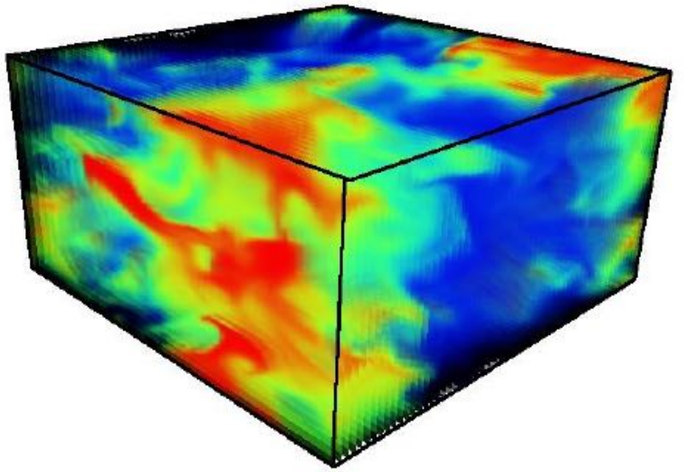
Example: Hydrodynamics

Navier-Stokes Equations

Governing (differential) equations (NSE):

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) &= 0 \\ \frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) &= -\nabla p + \nabla \tau + \rho \mathbf{f} \\ \frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho e \mathbf{v}) &= \nabla \cdot (\kappa \nabla T) + \rho q - p \nabla \cdot \mathbf{v} + \nabla \mathbf{v} : \tau \end{aligned}$$

- variables: ρ, $\mathbf{v}$, $e = E - |\mathbf{v}|^2/2$, p, τ, T



Constitutive (algebraic) relations:

- Newtonian stress tensor: $\tau = \tau(\mathbf{v}, \mu)$
- Thermodynamic: $p = \rho R T$ and $E = c_v T \Rightarrow p, T$

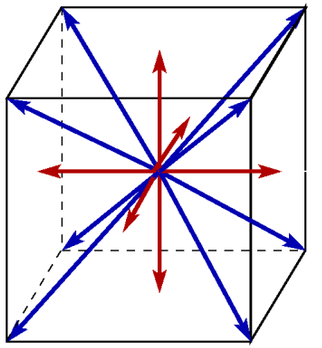
Simplification for incompressible flows: ($\rho = const$, $\mu = const$)

$$\begin{aligned} \nabla \cdot \mathbf{v} &= 0 \\ \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} &= -\nabla \widetilde{p} + \nu \Delta \mathbf{v} + \mathbf{f} \end{aligned}$$

with kinematic viscosity $\nu = \mu/\rho$

Example: Hydrodynamics from LBE

- simplest microscopic (particle, kinetic) approach to model macroscopic dynamics
- can be derived from Boltzmann equation
- yields Navier-Stokes equations (long wavelength and low frequencies)
- discrete coordinates and velocities



$$f_i(\mathbf{x} + \mathbf{e}_i,\, t + \Delta t) = f_i(\mathbf{x}, t) + \Omega_i[f(\mathbf{x}, t)]$$

with particle velocity distribution f_i along directions $i = 1, \ldots, q$

Bhatnagar-Gross-Krook (BGK) collision operator

$$f_i(\mathbf{x} + \mathbf{e}_i, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{f_i - f_i^{(eq)}}{\tau}$$

Macroscopic quantities:

$$\rho = \sum_i f_i \ , \quad \rho \mathbf{v} = \sum_i f_i \mathbf{e}_i \ ,$$

Implementation: two-step update

(1) collision step:

$$\tilde{f}_i(\mathbf{x},\, t) = (1-\omega)\cdot f_i(\mathbf{x},\, t) + \omega \cdot f_i^{(eq)}$$

(2) streaming (displace, diffuse) step:

$$f_i(\mathbf{x}+\mathbf{e}_i,\, t+\Delta t) = \tilde{f}_i(\mathbf{x},\, t)$$

Properties:

- O(200. . . 400) Flop in step (1)
  ➜ 10^{14} Flop for 10000 steps on 256^3 mesh
- intrinsically parallel
- may need temporal and spatial averaging (to eliminate noise)
- needs comparable or finer mesh resolution than Navier-Stokes solvers
- can easily incorporate complex boundary conditions (e.g. porous materials)

Stochastic Methods

Stochastic methods are required for physical or numerical problems if

- basic numerical problem is too difficult for direct methods
  e.g. complicated phase-space integration

- deterministic physical system is very complex and only stochastic description is practical
  e.g. partition function in statistical physics

$$Z(\beta, p) \sim \sum_{states} e^{-\beta H(s,p)}$$

- theory of physical system has an intrinsical stochastic nature
  e.g. path-integral in (eucidean) quantum field theory

$$Z(J) \sim \int_{fields} e^{-S(\phi,J)} D\phi$$

Stochastic Integration

Basic computational task is an integral

$$I = \int_{\mathcal{D}} f(x)\, d^n x$$

which may have

- high dimension of integration domain, e.g. $n = O(10^8)$
- irregular integration domain $\mathcal{D}$
- irregular behaviour of integrand $f(x)$

Sochastic integration exploits interpretation of I as a measurement on a stochastic system

$$I \;\sim\; \frac{1}{N}\sum_{i=1}^{N} f(x_i) \;+\; \begin{cases} O\big(1/N^{m/d}\big) & x_i \in \text{regular grid} \\ O\big(1/\sqrt{N}\big) & x_i \in \text{stochastic choice} \end{cases}$$

➜ numerical simulation of stochastic system
(stochastic simulation, Monte Carlo simulation)

➜ stochastic analysis of measured data from (numerical) experiment

Probability Theory

Given a probability density function (PDF)

$$\rho : \Omega \longrightarrow \mathbb{R}^+ \quad \text{with} \quad \int \rho(x)\, dx = 1$$

one can compute exact (analytic) expectation values

$$E[f] \equiv \langle f \rangle := \int_\Omega f(x)\, \rho(x)\, dx$$

of arbitrary functions

$$f : \Omega \longrightarrow \mathbb{V}$$

Examples:

- mean value of the PDF: $E[x] =: \mu$ $\quad (f(x) = x)$
- variance of the PDF: $E[(x-\mu)^2] = E[x^2] - \mu^2 =: \sigma^2$ $\quad (f(x) = (x-\mu)^2)$
- probability of a specific value: $E[\delta(x-\xi)] = \rho(\xi)$ $\quad (f(x) = \delta(x-\xi))$
- cumulants of f

$$\begin{aligned} C_1 &= E[f] \\ C_2 &= E[(f - C_1)^2] \\ &\cdots \end{aligned}$$

Stochastic Samples

Independent measurements of a stochastic system (described by the PDF ρ) provide a sample of uncorrelated values

$$\begin{array}{|c|} \hline \rho \\ \hline \end{array} \longrightarrow \xi_1$$
$$\begin{array}{|c|} \hline \rho \\ \hline \end{array} \longrightarrow \xi_2$$
$$\cdots$$
$$\begin{array}{|c|} \hline \rho \\ \hline \end{array} \longrightarrow \xi_N$$

Sample is described by the PDF

$$\rho_N\big(\xi_1, \xi_2, \ldots, \xi_N\big) = \rho(\xi_1) \cdot \rho(\xi_2) \cdots \rho(\xi_N)$$

- Expectation values are predictions for primary observables of measurements
- Sample averages

$$\overline{f} \equiv \frac{1}{N} \sum_{i=1}^{N} f(\xi_i)$$

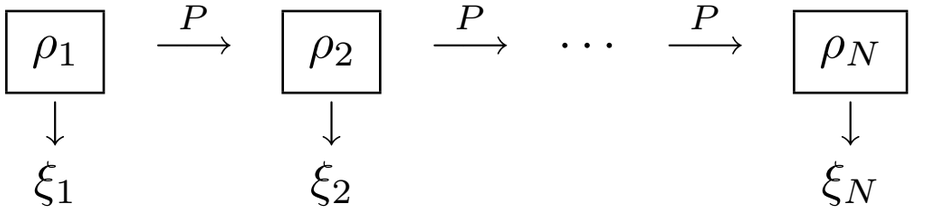
  are estimators for the expectation values of primary observables

- Functions of the expectation values are called secondary observables

Correlated Samples

In practice, it is often too expensive to generate an uncorrelated sample!

Instead, one uses a stochastic process to generates a correlated sample



according to a joint PDF

$$\rho_N\big(\xi_1, \xi_2, \ldots, \xi_N\big)$$

with determines the autocorrelation function

$$\Gamma_{ij} \equiv E\big[(x_i - E[x_i])(x_j - E[x_j])\big]$$

For a Markov Process, the PDF for successive elements of the ensemble (Markov Chain) is defined by a stochastic mapping between PDFs on Ω

$$P : \rho_n \mapsto \rho_{n+1}$$

N.B.: By restarting the process several times, one may generate a (small) uncorrelated sample of ensambles (but with the elements within each ensemble being correlated)

Markov Processes

The stochastic mapping of a Markov chain can be written as

$$\rho_{n+1}(y) = \int_\Omega P(y \leftarrow x) \cdot \rho_n(x)\, dx$$

with

$$\begin{aligned} P(x \leftarrow y) &\geq 0 \qquad (\forall x, y) \\ \int P(y \leftarrow x)\, dy &= 1 \qquad (\forall x) \end{aligned}$$

To use the Markov Process for Monte Carlo simulation of a system with PDF ρ^* we want:

- ρ^* to be the unique fixed point of the process

$$P\rho^* = \rho^*$$

- convergence from an arbitrary start distribution ρ_1 to the fixed point distribution

$$\lim_{n\to\infty} P^n \rho_1 = \rho^*$$

Sufficient, but not necessary condition:

- strong ergodicity

$$P(y \leftarrow x) > 0 \qquad (\forall x, y)$$

- detailed balance

$$P(y \leftarrow x) \cdot \rho^*(x) = P(x \leftarrow y) \cdot \rho^*(y) \quad (\forall x, y)$$

Examples

If Ω is finite (not only finite dimensional):

- $\rho_i \equiv \rho(i)$ is a stochastic vector, i.e. $\rho_i \geq 0$ and $\sum_i \rho_i = 1$
- $P_{fi} \equiv P(f \leftarrow i)$ is a stochastic matrix, i.e. $P_{fi} \geq 0$ and $\sum_f P_{fi} = 1$

  N.B.: Every stochastic matrix has at least one fixed point

Stochastic matrices with detailed balance:

- Non-ergodic process: $P^{2n} = 1 \neq P^{2n+1} = P \ (\forall n)$

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \rho^* = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1/2 \\ 1/2 \end{pmatrix}$$

- Heatbath-like process: $P_{fi} = \rho^*_f \ (\forall i) \quad \Rightarrow \quad P^2 = P$ and $P^n \rho_1 = \rho^* \ (\forall n \geq 1, \forall \rho_1)$

$$P = \begin{pmatrix} \alpha & \alpha & \alpha \\ 1-\alpha & 1-\alpha & 1-\alpha \\ 0 & 0 & 0 \end{pmatrix} \quad \rho^* = \begin{pmatrix} \alpha \\ 1-\alpha \\ 0 \end{pmatrix}$$

  but strong ergodicity condition is not satisfied if $\exists f : \rho^*_f = 0$

- Metropolis-like process:

$$P_{fi} = \begin{cases} \min\left(1, \frac{\rho^*_f}{\rho^*_i}\right) & (f \neq i) \\ 1 - \sum_{j \neq f} P_{ji} & (f = i) \end{cases}$$

Implementation

Generate sequence of states $x \in \Omega$ (rather than probability distributions ρ on Ω) by a stochastic update process

$$U : x_n \longrightarrow x_{n+1}$$

with transition probabilities $P(x_{n+1} \leftarrow x_n)$ of the Markov chain

Example: Metropolis Update

(1) Proposal Step: Generate a new (trial-) state with probability

$$P_S(f \leftarrow i)$$

(2) Metropolis Step: Accept new state with probability

$$P_A(f \leftarrow i) = min\Big(1, \frac{\rho_f^* \cdot P_S(i \leftarrow f)}{\rho_i^* \cdot P_S(f \leftarrow i)}\Big)$$

e.g. by comparing P_A with a (pseudo-)random number $r \in [0, 1]$

Remarks:

- Transition probability of process is $P_S(f \leftarrow i) \cdot P_A(f \leftarrow i)$
- Process satisfies detailed balance
- Metropolis-step simplifies for symmetric (e.g. flat) proposal, i.e. $P_S(f \leftarrow i) = P_S(i \leftarrow f)$
- In case of rejected update, x_i is kept in sample more than once

Integration (cont.)

- Integration with a normalised (stochastic) measure $\rho(x)$ is an expextation value and can be estimated by the mean over a sample S with distribution ρ (e.g. flat)

$$I \equiv \int f(x)\ \rho(x)\ dx = E[f] \approx \frac{1}{N} \sum_{x_i \in S} f(x_i)$$

- Can choose different distributions to improve measurement on S: importance sampling

$$I = \int f(x)\, dx = Z[1] \cdot \int f(x) \cdot \frac{dx}{Z[1]} = Z[g] \cdot \int \frac{f(x)}{g(x)} \cdot \frac{g(x)}{Z[g]}\, dx$$

  with normalisation factors

$$Z[g] \equiv \int_{\mathcal{D}} g(x)\, dx$$

- Ratios of integrals do not depend on pre-factor Z
- Sample from Markov chain does not depend on normalisation factor $1/Z$
- However, Z is needed to obtain absolute value of the integral and must be determined by other method (e.g. analytical or hit-and-miss)

Example: Integration with Importance Sampling

Consider a strongly peaked integrand

$$f(x) = (1 + \beta \cdot x) \cdot e^{-\alpha \cdot x}$$

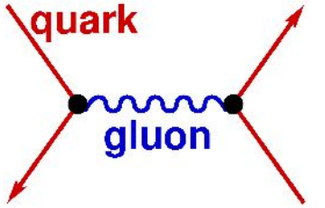
- Flat sample: $\rho \sim 1$
  - bad sampling (noisy, many small contributions to mean)
  - cheap generation of sample
- Importance sampling: $\rho \approx f$
  - good importance sampling (small variance of mean)
  - expensive generation of sample (small acceptance in Metropolis or slow Heatbath)
- Not a clever choice: $\rho = f$
  - exact mean: $\frac{1}{N} \sum 1 = 1$
  - expensive generation of sample if needed for integral of other functions
  - computation of Z is original integral

Lattice QCD

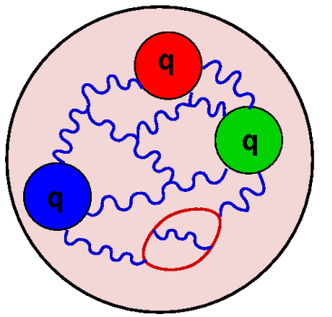
QCD = Relativistic Quantum Field Theory of Strong Interactions

↓ 4-d space-time ↓ non-perturbative

$$\mathcal{L}_{\rm QCD}(g_0, m_0^{(f)}) = \frac{1}{4} G_{\mu\nu} G_{\mu\nu} + \sum_{f=1}^{N_f} \overline{\psi}(i \not{D} - m_0^{(f)})\psi$$



Parameters: 1 gauge coupling + N_f quark masses



$$\overbrace{\begin{pmatrix} F_\pi \\ m_\pi \\ m_{\rm K} \\ m_{\rm D} \\ m_{\rm B} \end{pmatrix}}^{\text{Experiment}} \quad \begin{pmatrix} \mathcal{L}_{\rm QCD}(g_0, m_0) \\ \Longrightarrow \end{pmatrix} \quad \overbrace{\begin{pmatrix} \Lambda_{\rm QCD} \\ \hat{M} = (M_{\rm u} + M_{\rm d})/2 \\ M_{\rm s} \\ M_{\rm c} \\ M_{\rm b} \end{pmatrix}}^{parameters\ (RGI)} + \overbrace{\begin{pmatrix} \xi \\ F_B \\ B_B \\ \vdots \end{pmatrix}}^{\textbf{Predictions}}$$

Physics Challenges

- Verify or falsify QCD at low energies
- Determine fundamental parameters
- Compute basic hadron properties
  (spectroscopy, hadronic matrix elements)
- Make contact with Chiral Perturbation Theory (ChPT)
- Study exotic forms of matter
  (quark-gluon plasma)
- . . .

Euclidean n-point functions:

$$\langle \phi(x) \dots \phi(0) \rangle := \langle 0 | \phi(x^M) \dots \phi(0) | 0 \rangle \Big|_{x_0 = -i x_0^M} \qquad (x_0^M > 0)$$

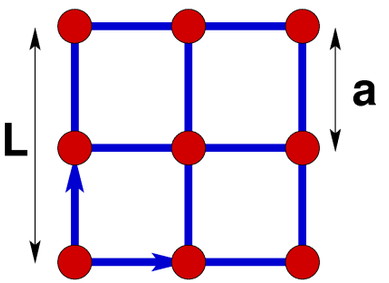
Path Integral Quantisation:

$$\Big\langle \phi_1(x_1) \dots \phi_n(x_n) \Big\rangle \sim \int \phi_1(x_1) \dots \phi_n(x_n) \cdot e^{-S[U,\overline{\psi},\psi]} \; D[\overline{\psi}] D[\psi] \; D[U]$$

well-defined (only) if space-time is discrete and finite

➜ 4-dimensional lattice: $V = L^3 \times T = O(10^6 \cdots 10^7)$

➜ integration space has finite number of dimensions: $d \sim V$



stochastic integration		here	
i	$\longleftrightarrow$	$\mathbf{x}$	sites
x_i	$\longleftrightarrow$	U, ψ	fields
ρ	$\longleftrightarrow$	e^{-S}	action
f	$\longleftrightarrow$	$\mathcal{O}$	observables

Lattice Action

$$\begin{array}{ccccccc}
\mathcal{L}_{\rm QCD}(g_0, m_0^{(u)}, \ldots) & = & \frac{1}{4} G_{\mu\nu} G_{\mu\nu} & + & \sum_{f=1}^{N_f} \overline{\psi}(i \not{D} - m_0^{(f)})\psi & & \\
\downarrow & & \downarrow & & \downarrow & & \\
S_{lat} & = & \underbrace{S_g(U)}_{\text{plaquettes}} & + & \underbrace{S_f(U,\psi)}_{\overline{\psi}\mathbf{M}(U)\psi} & + & O(a)
\end{array}$$

$$\beta \equiv 6/g_0^2 \qquad \kappa \equiv 2/(m_0 + 8)$$

Dynamical Fermions ($N_f = 2$)

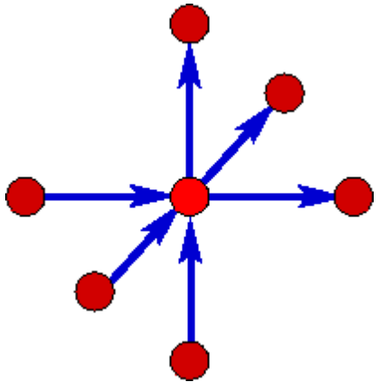
$$\int e^{-S(U,\psi)}\, D[\psi] \quad \sim \quad e^{-S_g(U)} \cdot det M(U) \quad \sim \quad e^{-S_g(U)} \cdot \int D[\phi]\, e^{-\bar{\phi}(M^\dagger M)^{-1}\phi}$$

quenched $\downarrow$ *const*

fermionic ψ bosonic ϕ

Wilson-Dirac Operator:

$$[\mathbf{M}\phi]_x \quad \sim \quad \phi_x - \kappa \sum_{\mu=\pm 1}^{\pm 4} \mathbf{U}_{\mu,\mathbf{x}} \otimes (1-\gamma_\mu) \cdot \phi_{x+\hat{\mu}} \; + \; \mathbf{O(a)}$$



➜ 1320 floating-point operations per lattice site

Alternative Fermion Actions:

Action	Properties	Truncation Erros		CPU Cost
		action	operators	
Staggered	1/4 non locality	$O(a^2)$	"taste" mixing	low
Wilson	χ recoverable	$O(a)$	$O(a)$	moderate
Clover	χ recoverable	$O(a^2)$	$O(a)$	moderate
Twisted	χ recoverable	$O(a)$	$O(a^2)$	moderate
Domain-Wall	χ for $L_5 \to \infty$	$O(a)$	$O(a)$	high
Overlap	χ exact	$O(a^2)$	$O(a^2)$	very high

$O(1)$

$\downarrow$

$O(10)$

$\downarrow$

$O(1000)$

Monte Carlo Simulation

- Generate sample of gauge configurations U from ensemble with distribution

$$\rho(U) \sim e^{-S_g(U)-S_f(U)}$$

- Estimate observables from ensemble average

$$\langle \mathcal{O} \rangle \;\rightarrow\; \frac{1}{N}\sum_{U} \mathcal{O}[U] \;+\; O(N^{-1/2})$$

Algorithms for Gauge Updates (quenched):

- overrelaxation ➜ fast decorrelation
- heatbath ➜ ergodicity

Measurement of observables is dominant computational cost (inversion of M)

Fermion Algorithms

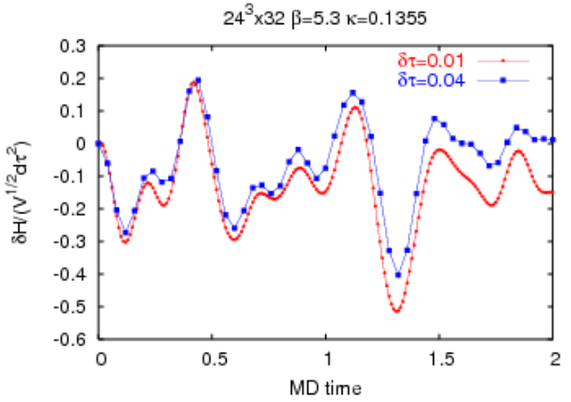
Hybrid Monte-Carlo:

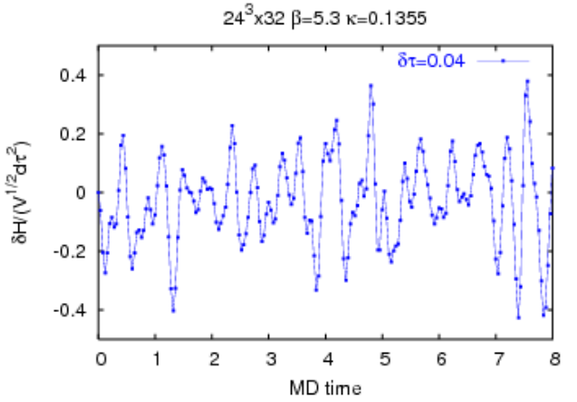
Consider (classical) Hamiltonian

$$H \;=\; \frac{1}{2}\pi^2 \;+\; S_g(U) \;+\; \overline{\phi}\cdot\Big[M^\dagger(U)M(U)\Big]^{-1}\cdot\phi$$

- Generate gaussian η and set $\phi = M^{-1}\eta$
- Generate gaussian conjugate momenta π of U fields
- Molecular-Dynamics evolution of (U,π) — reversibility
  - possibly useing multiple step-size (in Monte Carlo time)
  - each forces computation requires inversion(s) of M
- Metropolis accept/reject with probability — acceptance

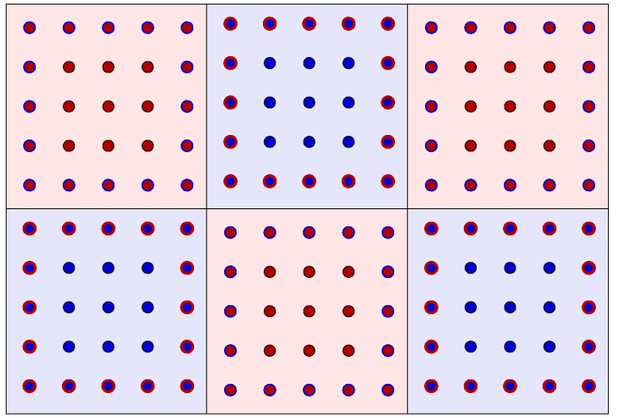
$$\max(1, e^{-\Delta H})$$





Domain Decomposition / Schwarz Alternating Procedure

Decompose lattice into regular grid of equal blocks Ω_i



To solve

$$\mathbf{M}\phi = \eta$$

repeatedly update fields on each block Ω_i by solving

$$\begin{aligned} \mathbf{M}\phi^{(k+1)} &= \eta(x) \quad (x \in \Omega_i) \\ \phi^{(k+1)} &= \phi^k \quad (x \notin \Omega_i) \end{aligned}$$

- update can be performed in parallel on all blocks with same checkerboard colour
- procedure is a Neumann series, which converges (slowly) to full solution $\mathbf{M}^{-1}\eta$
- use as preconditioner for suitable Krylov-space solver

Simulation Cost: (for 100 statistically independent configurations with DD-HMC)

$$cost \sim 0.05\ Tflops \times year \left(\frac{L}{3\,fm}\right)^5 \left(\frac{0.1\,fm}{a}\right)^6 \left(\frac{\overline{m}_q}{20\ MeV}\right)^1$$

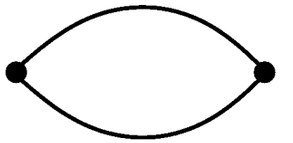
Measurement of Observables

- Consider local products of operators with suitable quantum numbers
  e.g. pion

$$P(x) = \overline{\phi}(x)\gamma_5\phi(x)$$

- Interpretation of correlation functions by Transfer Matrix formalism

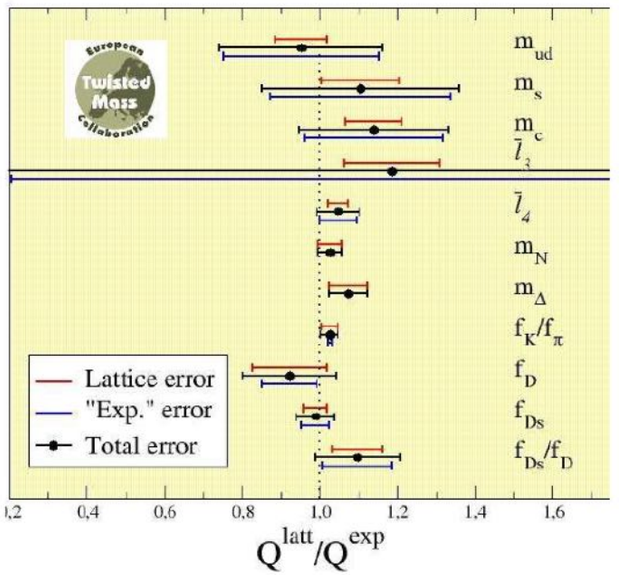
$$\langle P(x)P(y)\rangle \;\sim\; e^{-(x_0-y_0)\cdot m_\pi} + a_{\pi'} \cdot e^{-(x_0-y_0)\cdot m_{\pi'}}$$

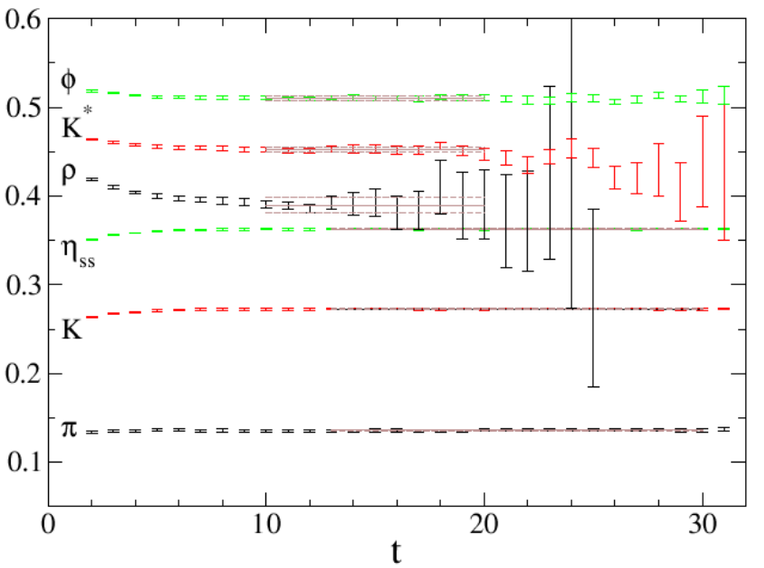


(no rotation back to Minkowski space required)

- Suppression of excited states requires large time-extent $(x_0 - y_0 \to \infty)$

$N_f = 2$
[ETMC]





$N_f = 2 + 1$
$m_\pi = 296$ MeV
[Kuramashi, PACS-CS]

Extrapolation to Physical Limits

- ✘ Continuum Limit: $a \longrightarrow 0$
  discretisation effects

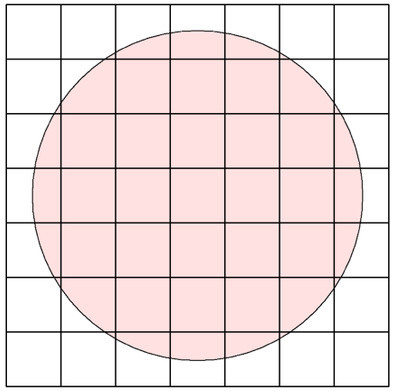
- ✘ Thermodynamic Limit: $V \longrightarrow \infty$
  finite size effects and excited states

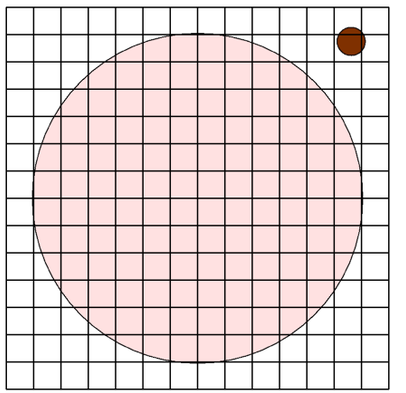
- ✘ Chiral Limit: $m_q \longrightarrow 0$
  chiral symmetry breaking: $\langle \overline{\psi}\psi \rangle \neq 0$

  stability:

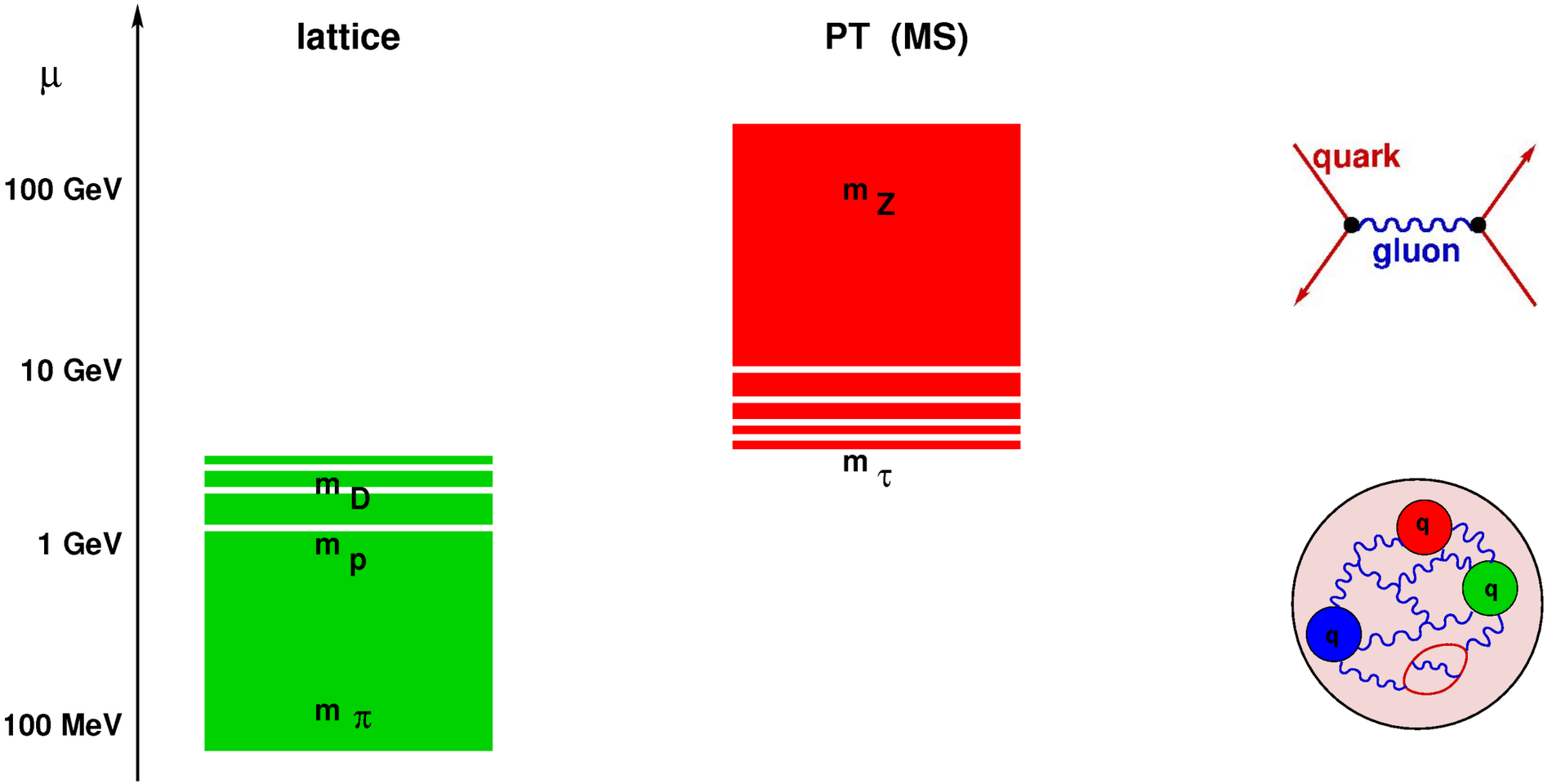
$$m_q > \frac{3a}{Z\sqrt{V}}$$

- ✘ Heavy Quarks: $a \lesssim m_B$





Energy Scales



$$a^{-1} \gg \mu_{PT} \gg m_H \gg L^{-1}$$

$$(0.08\ fm)^{-1} \quad O(10)\ GeV \quad 4.5 \ldots 0.14\ GeV \quad (2\ fm)^{-1}$$

Renormalisation

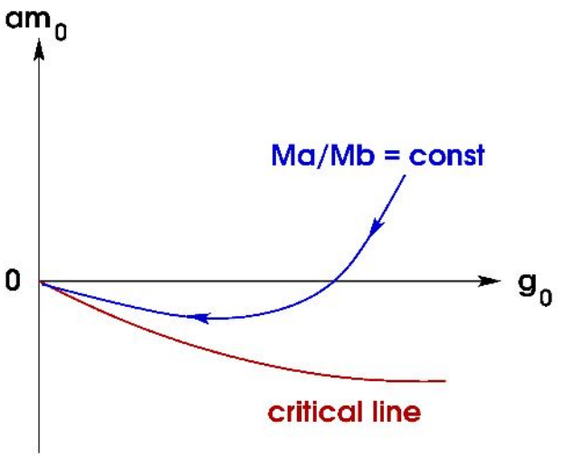
❒ **At high energies:** PT and $\overline{MS}$

$$\Phi(q,r) = C_0(q,r) + C_1(q,r,\mu)\cdot\alpha_{\overline{MS}}(\mu) + C_2(q,r,\mu)\cdot\alpha^2_{\overline{MS}}(\mu) + \cdots$$

- complicated but automatised computation (algebraic and numerical)
- $\alpha_{\overline{MS}}(\mu) \equiv g^2_{\overline{MS}}/4\pi$ depends on Φ, choice of $\mu \approx q$, and order of PT
- $\overline{m}_{\overline{MS}}(\mu)$ may require additional assumptions (e.g. QCD sum rules)
- unique procedure (also for composite operators)

❒ **At low energies:** LQCD with Hadronic Scheme

$$m_H^{exp} = \lim_{a\to 0} \frac{(am_H)}{a(g_0)}$$



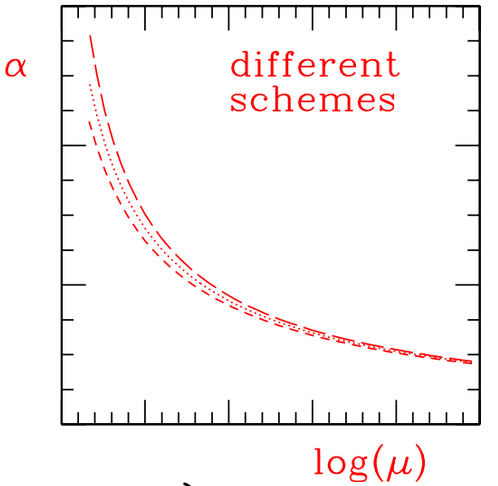
- depends on choice of m_H
- need to compute N_f ratios $m_{H'}/m_H$ and kept them fixed at physical values for $a \longrightarrow 0$
- different choices of non-perturbative renormalisation conditions with ambiguities from discretisation effects

Renormalization Group and Λ-Parameter

RGE for mass-independent scheme: $\overline{g} \equiv \overline{g}(\mu)$

$$\mu \frac{\partial \overline{g}}{\partial \mu} = \beta(\overline{g})$$

$$\overset{\overline{g} \to 0}{\sim} -\overline{g}^3 \left\{ b_0 + b_1 \overline{g}^2 + b_2 \overline{g}^4 + \ldots \right\}$$



- exact equation for "integration constant" Λ

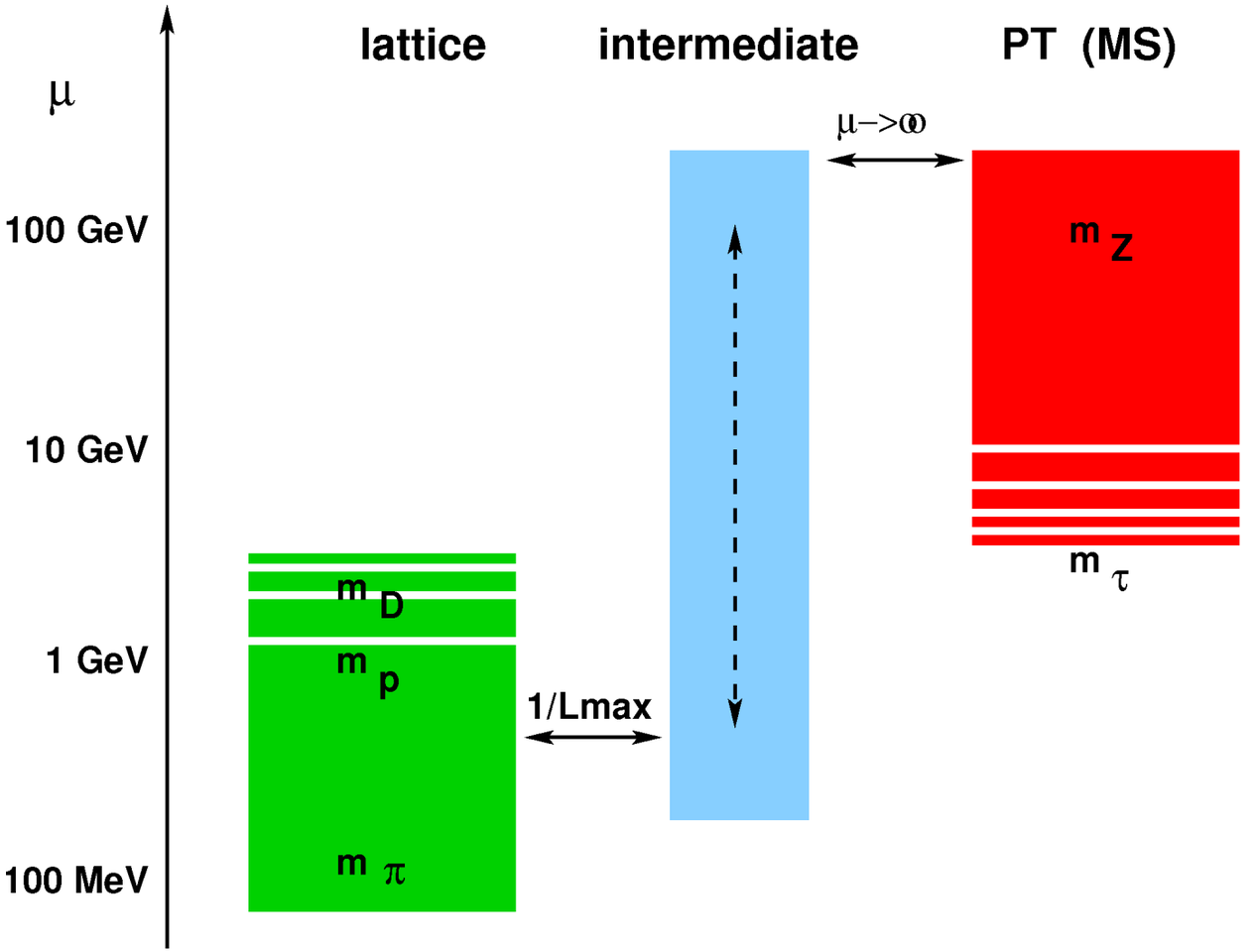
$$\Lambda = \mu \, (b_0 \overline{g}^2)^{-b_1/2b_0^2} \mathrm{e}^{-1/2b_0\overline{g}^2} \, \exp\left\{ -\int_0^{\overline{g}} \mathrm{d}g \left[\frac{1}{\beta(g)} + \frac{1}{b_0 g^3} - \frac{b_1}{b_0^2 g} \right] \right\}$$

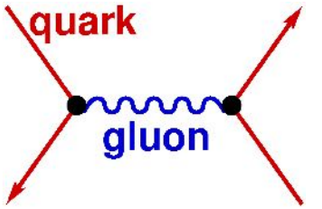
- trivial scheme dependence

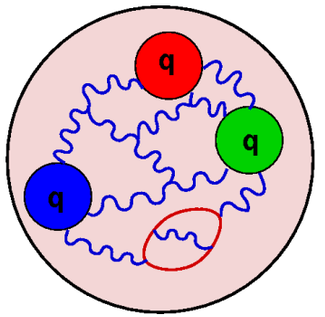
$$\alpha_a = \alpha_b + c_{ab} \, \alpha_b^2 + O(\alpha_b^3) \quad \Rightarrow \quad \Lambda_a / \Lambda_b = \mathrm{e}^{c_{ab}/(4\pi b_0)}$$

- use a suitable physical coupling (scheme) and non-perturbative $\beta(\overline{g})$

Connecting Hadronic and High-Energy Physics







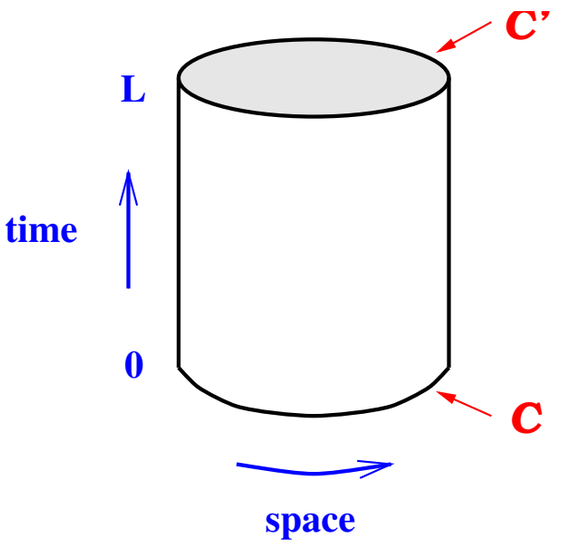
e.g. using Schrödinger functional to

- define and compute NP renormalisation and running
- implement and test Symanzik improvement
- perform reliable continuum limit
- verify that systematic errors are under control

Definition of Schrödinger Functional

- finite physical volume L^4, $T = L$
- Dirichlet b.c. $C(\eta)$, $C'(\eta)$ at $x_0 = 0, T$
- periodic b.c. in space (up to phase θ)

$$Z_{SF}(C, C') = e^{-\Gamma(\eta)} = \int_{\text{fields}} e^{-S(\eta)}$$



(LxLxL box with periodic b.c.)

- renormalised coupling

$$\left.\frac{\partial\Gamma(\eta)}{\partial\eta}\right|_{\eta=0} \equiv \frac{k}{\overline{g}^2_{SF}(L)}$$

- mass-independent scheme

$$m_{PCAC} = 0$$

- renormalisation scale

$$\mu = 1/L$$

Step Scaling Function (SSF)

- "discrete" β-function

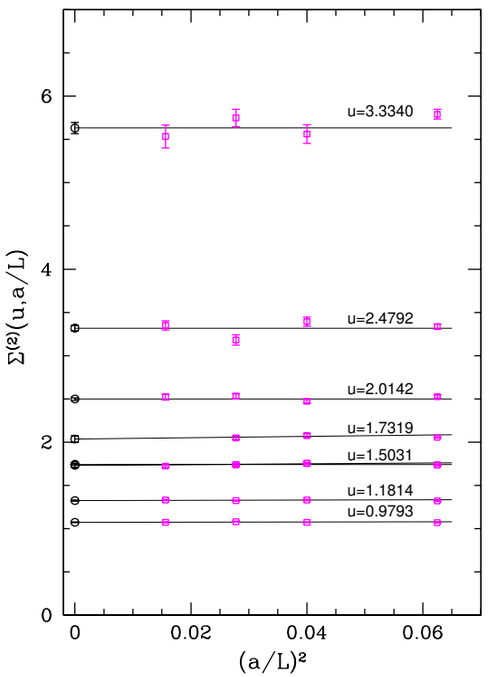
$$\sigma(\overline{g}^2(L)) \equiv \overline{g}^2(2L)$$

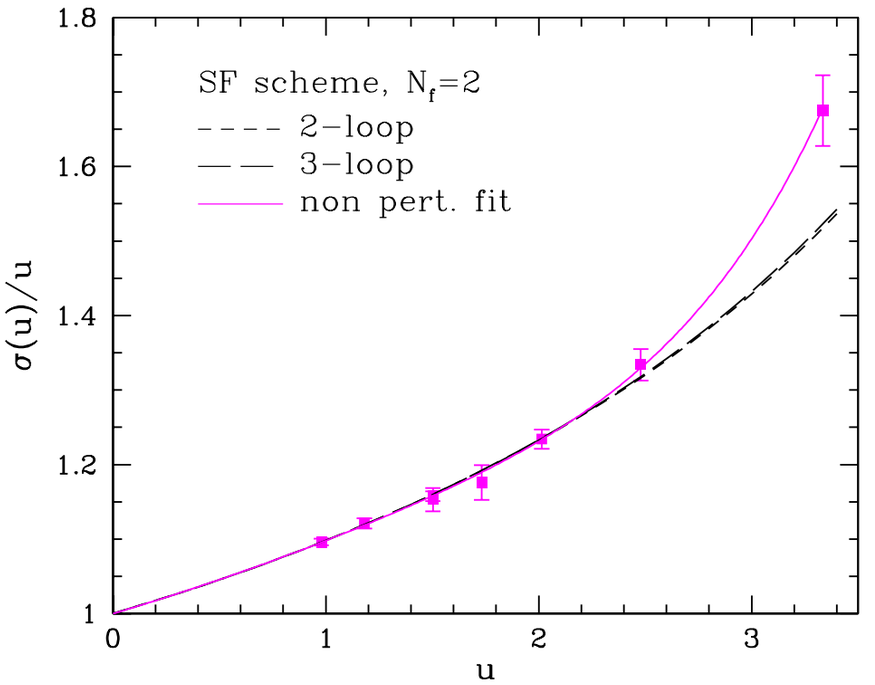
- determines NP running

$$u_k = \overline{g}^2\left(L_{max}/2^k\right)$$
$$\updownarrow$$
$$u_0 = \overline{g}^2\left(L_{max}\right)$$

- computation on the lattice

$$\Sigma(u, a/L) = \sigma(u) + O(a/L)$$



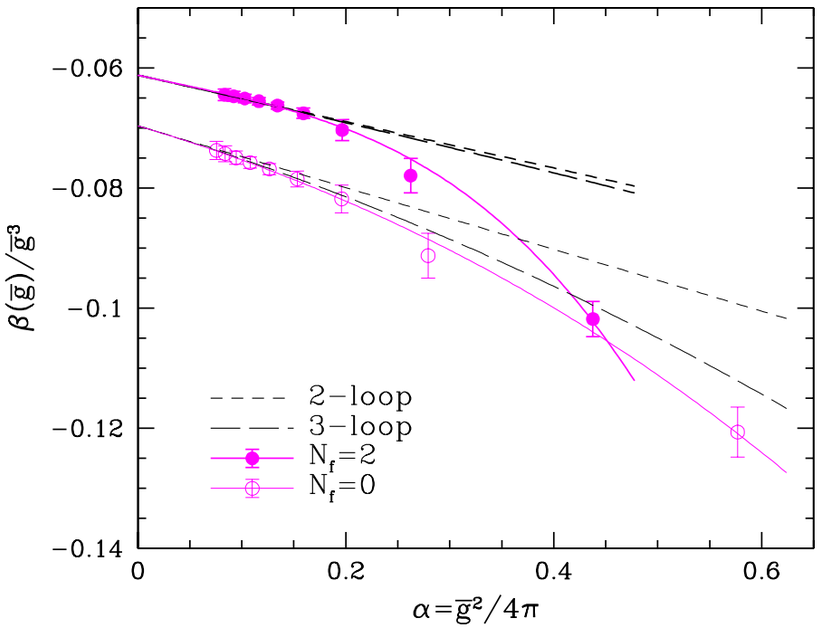


ALPHA Collaboration

Conversion of SSF to Beta Function

by solving (with parametrized SSF)

$$-2ln2 = \int_u^{\sigma(u)} \frac{\mathrm{d}x}{\sqrt{x}\beta(\sqrt{x})}$$

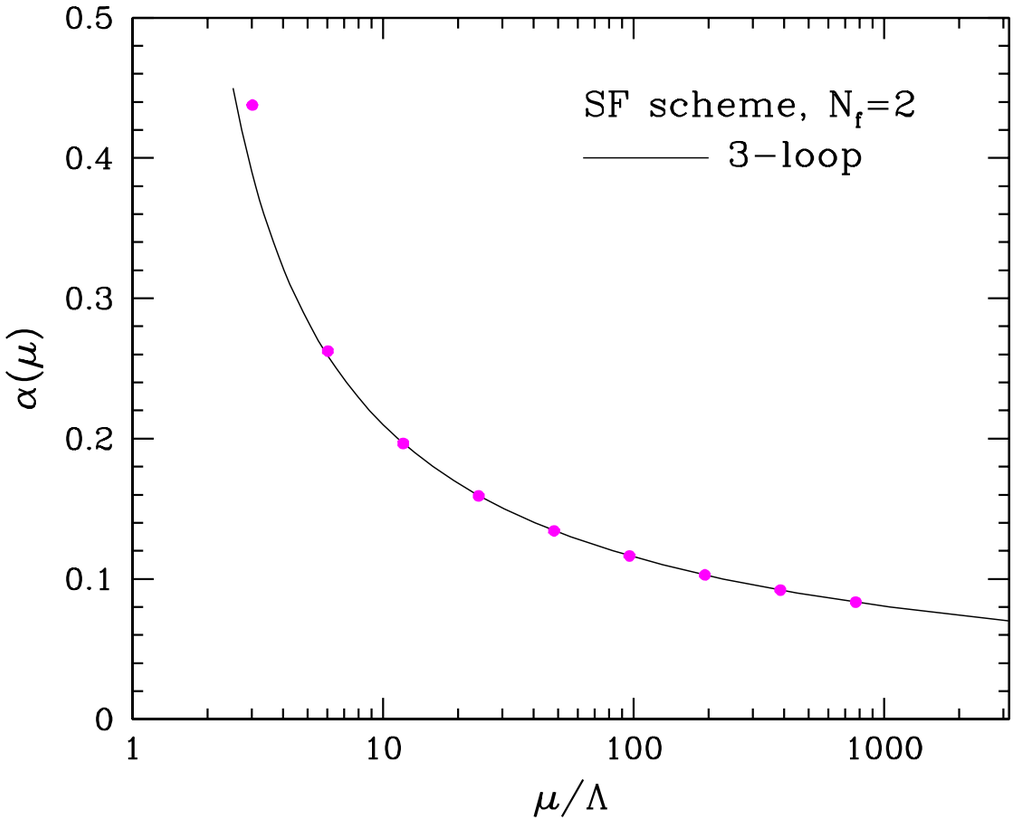


- clear effect of N_f
- strong deviation from 3-loop PT for $\alpha_{SF} \geq 0.25$
- without indication from within PT

Running of α_s

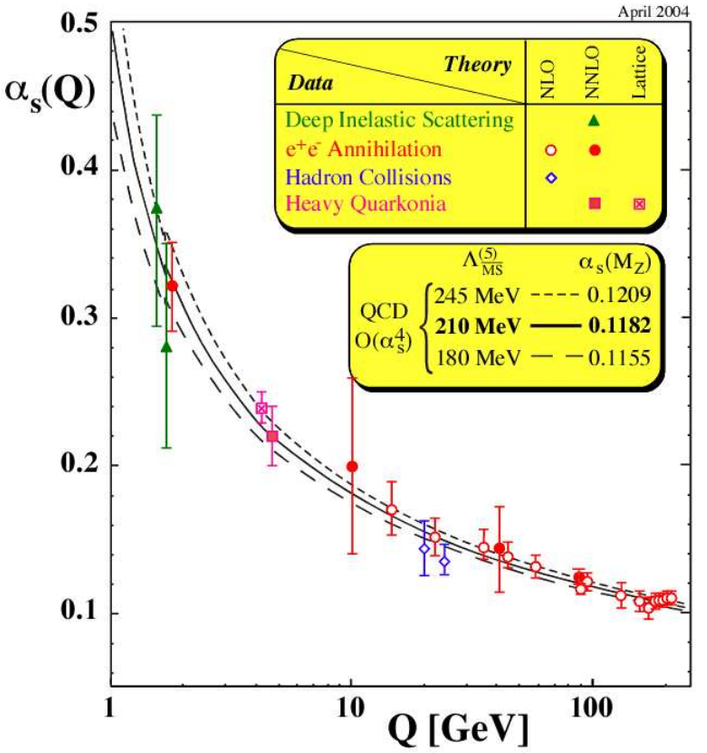
$N_f = 2$, NP + PT, SF scheme

error bars smaller than symbol size

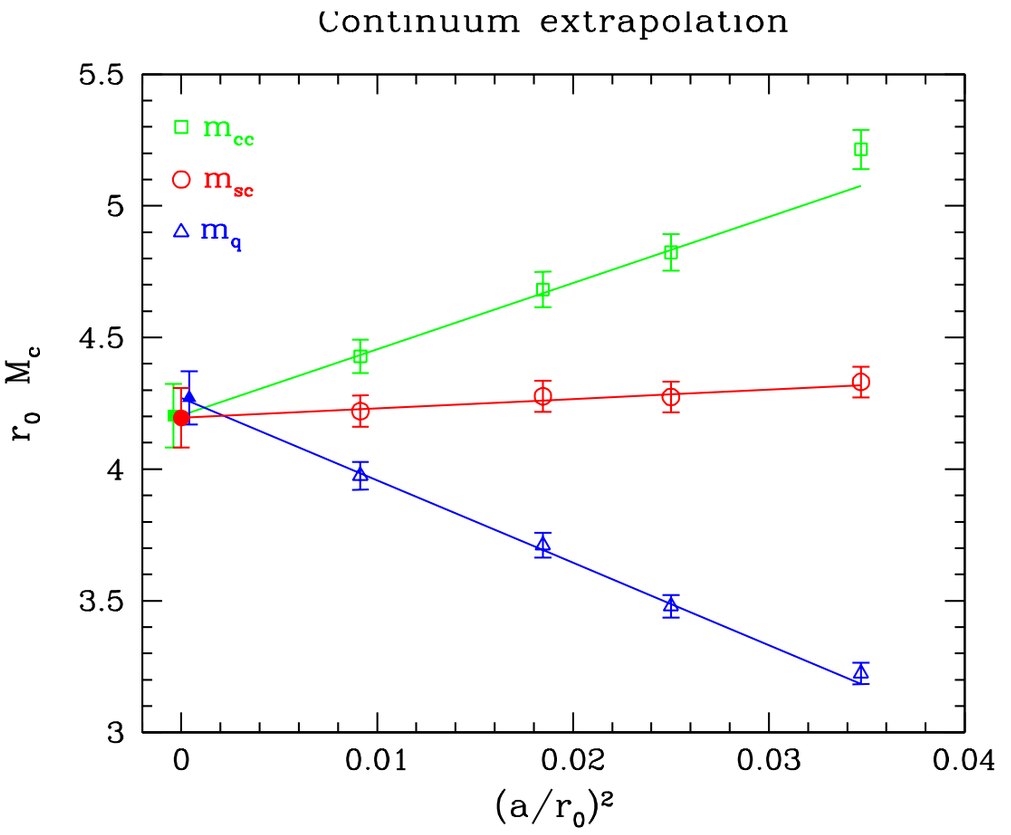


Experiment + PT, $\overline{MS}$ scheme

Bethke 2000



Example: Charm Quark Mass ($N_f = 0$)



- large mass renders $O(a)$ improvement essential
- different definitions of $m_0^{(c)}$ differ by $O(a^2 m_c^2)$ errors
- difficult continuum extrapolation

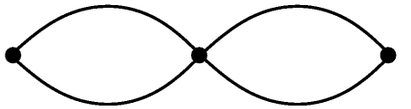
Example: Hadronic Matrix Elements (semi-leptonic)

E.g.

$$\frac{d\Gamma(B \to \pi \ell \bar{\nu})}{dq^2} \sim |V_{ub}|^2 \cdot |\mathcal{M}(q^2)|^2$$

with $q \equiv p_K - p_\pi$

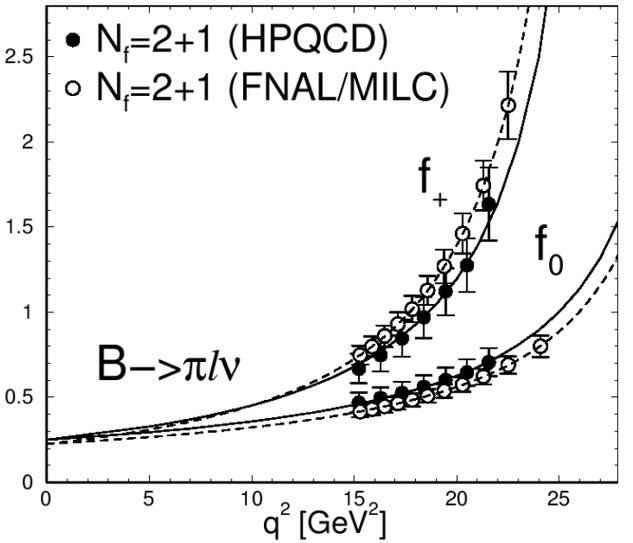
$$0 \leq q^2/m_K^2 = 1 - 2\frac{E_\pi}{m_K} \leq 1$$



requires HME

$$\mathcal{M}(q^2) = \langle \pi(p_\pi)|V_\mu|K(p_K)\rangle = f_+(q^2) \cdot (p_K + p_\pi)_\mu + \cdots$$

Kinematical range with best experimental sensitivity at low q^2 (phase space) is not accessible by LQCD calculations



Example: Hadronic Matrix Elements $K \to \pi\pi$

Aim:

* $\Delta I = 1/2$ rule
* $\varepsilon'/\varepsilon \leftrightarrow$ direct CP violation

Problems:

✘ Complicated mixings in renormalisation of $\Delta S = 2$ four-fermi operators

✘ HME with 2-hadron final states has FSI (strong phases)
but only average of in and out states can be determined in Euclidian space

✘ Contribution of intermediate $\bar{c}c$ states?

Strategies:

❒ Compute HME for $K \to 0$, $K \to \pi$ ($K \to \pi\pi$) at unphysical kinematics

$\downarrow$

determine low energy constants of ChPT at LO (NLO)

❒ Compute in finite volume HME for $K \to \pi\pi$

$\updownarrow$

related to infinite volume HME by (elastic) $\pi\pi$ scattering phase shifts in finite volume

Theoretical Physics on Supercomputers

Hubert Simma

Università Milano Bicocca

Plan:

- ❑ Introduction
- ❑ Solution Steps
- ❑ Examples of Applications and Methods
- ❑ Lattice QCD
- ❑ Machines

Computational Tasks of LQCD

Run-time Profile:

useful (if not yet known from theoretical analysis) to determine

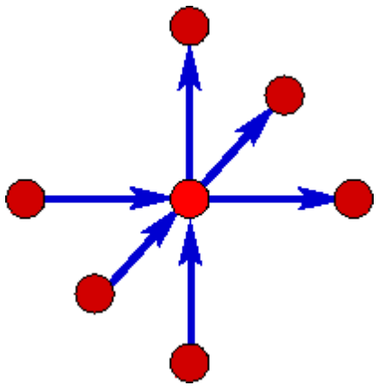
- algorithmic cost (how often is each computational tasks needed)
- CPU cost (how much time is spent for each computational tasks)

1 HMC trajectory, $24^3 \times 32$ lattice

routine	calls	time
Dirac operator (3 variants)	80844	58 %
Linear algebra (3 routines)	60736	26 %
Gauge forces + update	320	8 %
Global sum ($4 \times 8 \times 8$ nodes, 128 bit)	83554	0.4 %
Others ($\approx$ 70 routines)		7 %

➜ Dominant Task: Wilson-Dirac Operator

$$\phi' \equiv [D\phi]_x = \sum_{\mu=1}^{4} \{U(x,\mu)(1-\gamma_\mu)\phi(x+\hat{\mu}) + \cdots\}$$



LQCD on Parallel Computers

Data Storage:

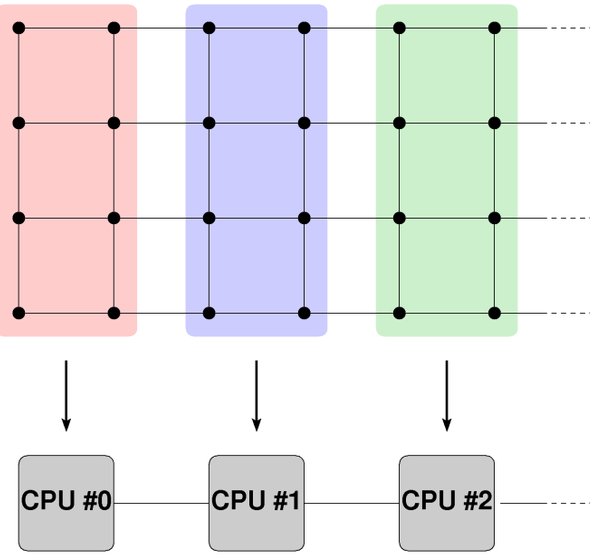
$\psi(x)$: 12 complex/site

$U(x, \mu)$: 9 complex/link

Lattice size $V = L^3 \times T$

e.g. $64^3 \times 128 \Rightarrow 3 \cdot 10^7$ sites (10 GB for gauge field U)

Processor grid: $P_0 \times P_1 \times P_2 \times P_3 = P$



➜ Trivial parallelisation by data distribution (uniform, static)

Communications:

- mainly nearest neighbour
- bandwidth requirements depends on implementation, algorithm, and physics choices

Number of remote neighbour sites (i.e. on different processor):

$$A^+ = \frac{V}{P} \sum_i \frac{1}{L_i} \qquad (P_i > 1)$$

Analysis of Computational Tasks

Total Computing Cost

$$N_{ops} \equiv \#\ \text{FP operations} \qquad = 1320$$

Computing vs. Memory Access

$$R_{ops} \equiv \frac{\#\ \text{arithmetic operations}}{\#\ \text{memory accessess}} \qquad \approx 7$$

Communication Requirement

$$R_{rem} \equiv \frac{\#\ \text{remote accesses}}{\#\ \text{memory accessess}} \qquad \sim A^{+}/V$$

Hardware Characteristics

Memory System

$$\rho_{mem} \equiv \frac{\text{flops}}{\text{bandwidth}} \quad [flop/byte]$$

Communication Network

$$\rho_{net} \equiv \frac{\text{network bandwidth}}{\text{memory bandwidth}}$$

Balance: Application vs. Hardware

$$R_{ops} \approx \rho_{mem} \qquad \text{and} \qquad R_{rem} \approx \rho_{net}$$

N.B.: Depending on (and to be taken into account in) various steps of the solution process

Example: APE Machines

History:

- mid 80's: APE1 idea for "Array Processor Experiment"
  by theoretical physicists at INFN
  $\longrightarrow$ 1 GFlops APE1 prototypes

- 1989–1994: APE100 full-custom development by INFN
  $\longrightarrow$ $O(200)$ GFlops installations (Quadrics)

- 1995–2000: APEmille development by INFN (+DESY)
  $\longrightarrow$ O(2) Tflops installations (Eurotech)

  CP-PACS/Hitachi

  QCDSP/Columbia

- 2001–2006: apeNEXT collaboration by INFN, DESY, Orsay
  $\longrightarrow$ O(15) Tflops installations (Eurotech)

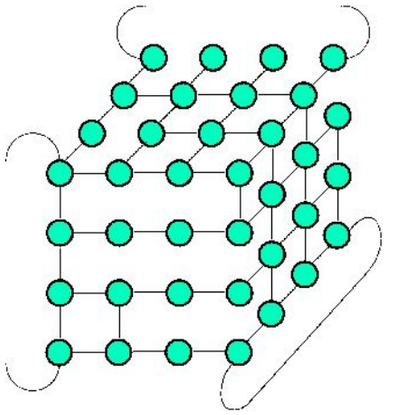
  QCDOC/Columbia

  BlueGene/IBM

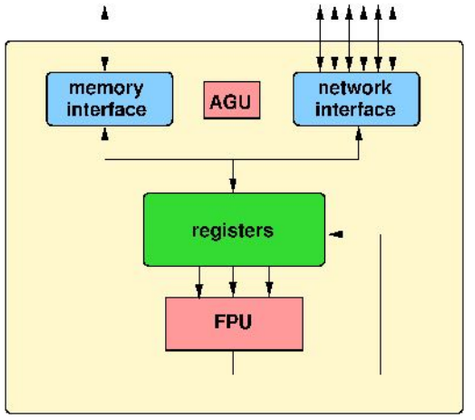
APE has provided major computing resources for LQCD in Europe

Optimization of APE Architectures for LQCD

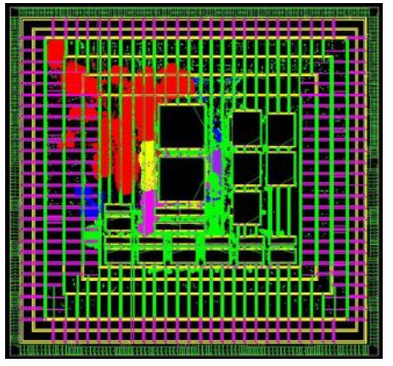
- 3D torus network

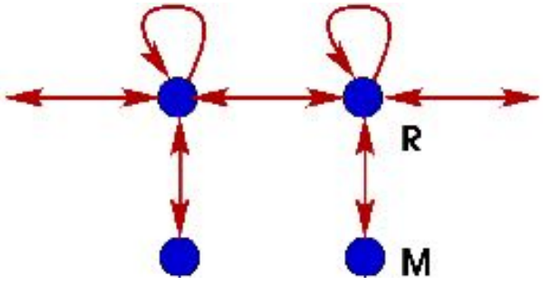


- slow clock, but many FP operations (complex) per cycle
  ⇒ low power consumption
- integrated memory and communication interface
  ⇒ compact design
- Very Long Instruction Word (VLIW) architecture
  ⇒ optimized scheduling at compile-time

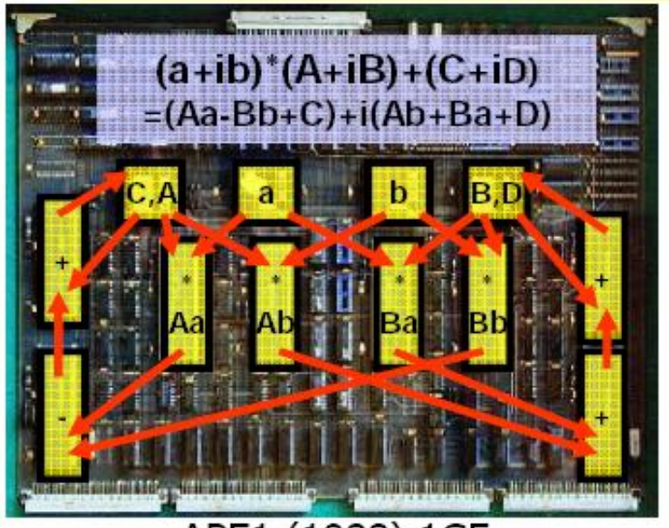


- large register file instead of cache
  ⇒ predictable and synchronous execution
- global syncronisation (mechanisms)
- RAS (ECC, status registers, . . .)
  ⇒ duration of single program execution $O(days)$
- SIMD programming model + communications by address mapping





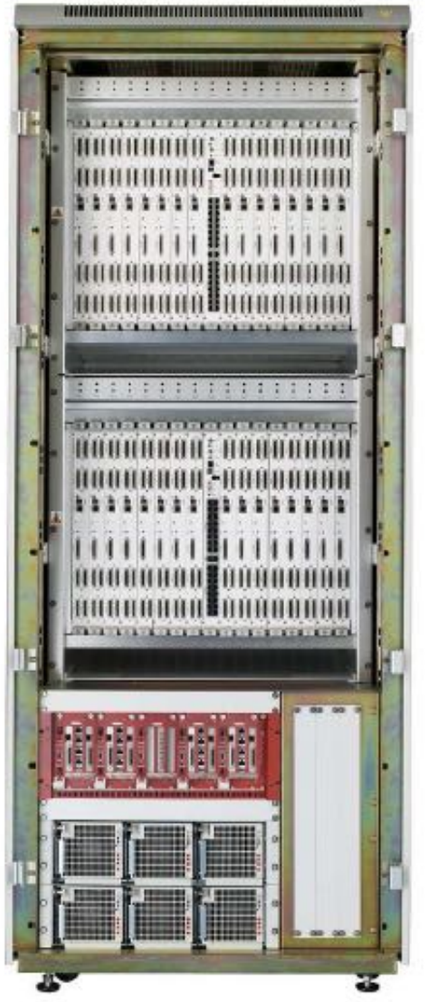
APE1 — APE100 — APEmille — apeNEXT



APE1 (1988) 1GF

APE100 (1992) 25GF, SP, REAL

APEmille (1999) 128GF, SP, Complex



apeNEXT (2004) 800GF, DP, Complex

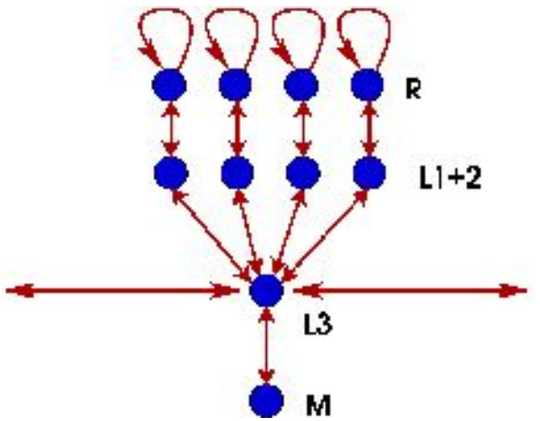
Example: BlueGene/L and P

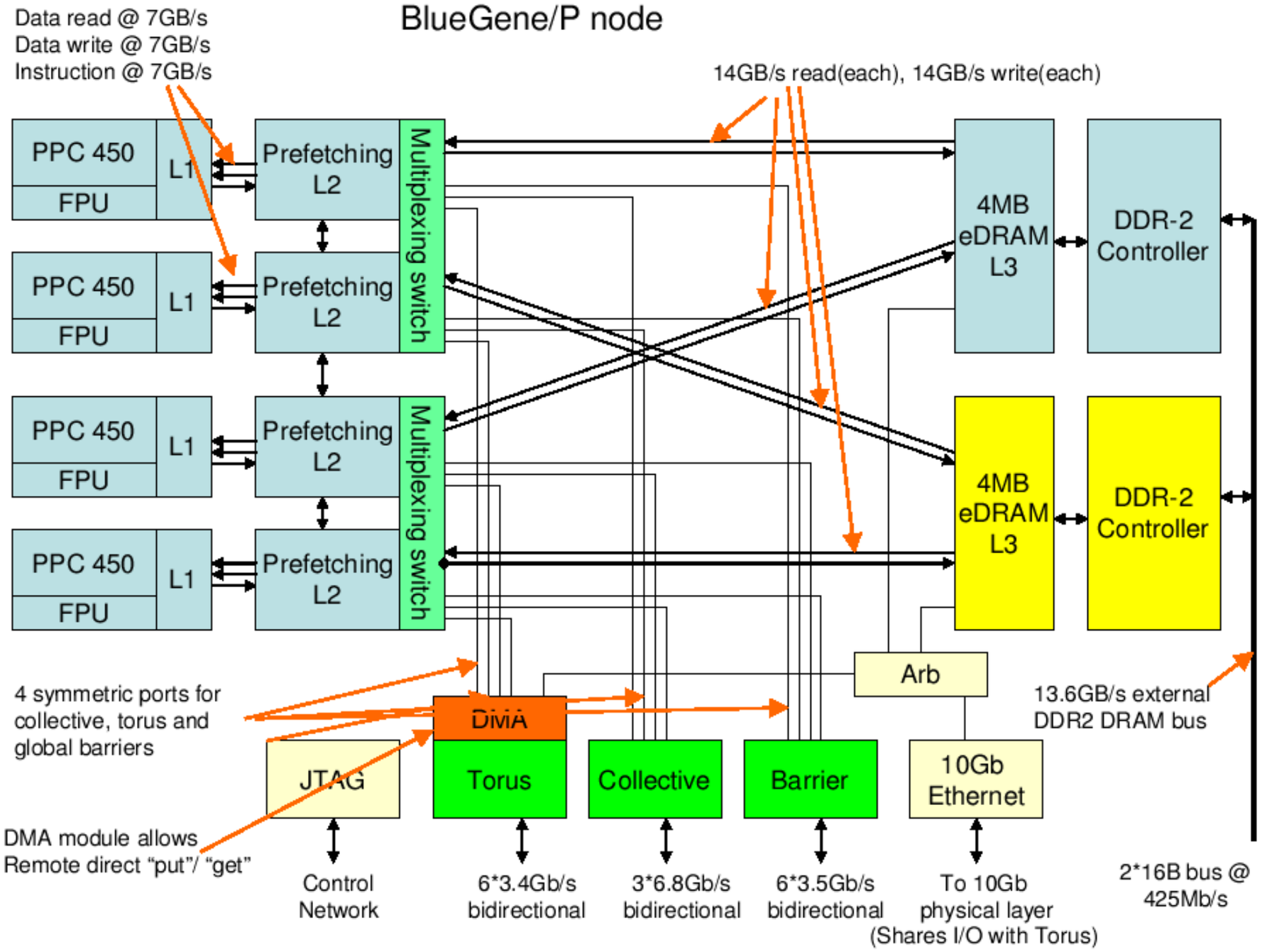
Characteristics: [QCDOC]

- PowerPC 440 core @ 850 MHz [500 MHz]
- MIMD distributed memory
- on-chip L1, L2, L3 caches (8 MB) [4 MB]

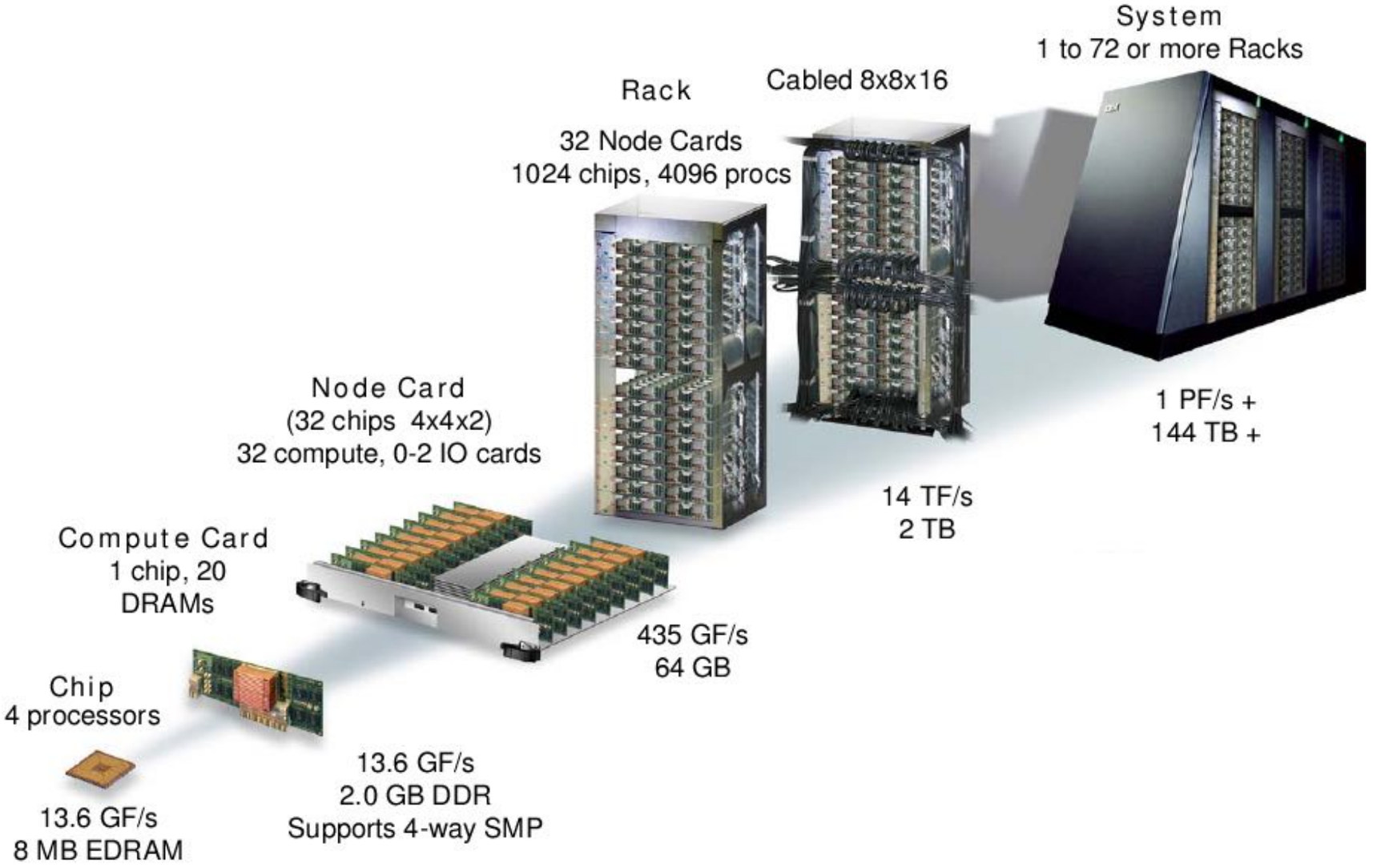
Node:

- 4 cores [1 core]
- 2 GB DDR
- 3-d torus network (6 links at 425 MB/s = 0.5 B/clock) [6-d at 65 MB/s]
- tree network (850 MB/s)




BlueGene/P node
Data read @ 7GB/s
Data write @ 7GB/s
Instruction @ 7GB/s
14GB/s read(each), 14GB/s write(each)
PPC 450
FPU
L1
Prefetching L2
Multiplexing switch
4MB eDRAM L3
DDR-2 Controller
Arb
DMA
JTAG
Torus
Collective
Barrier
10Gb Ethernet
4 symmetric ports for collective, torus and global barriers
DMA module allows Remote direct "put"/ "get"
13.6GB/s external DDR2 DRAM bus
Control Network
6*3.4Gb/s bidirectional
3*6.8Gb/s bidirectional
6*3.5Gb/s bidirectional
To 10Gb physical layer (Shares I/O with Torus)
2*16B bus @ 425Mb/s

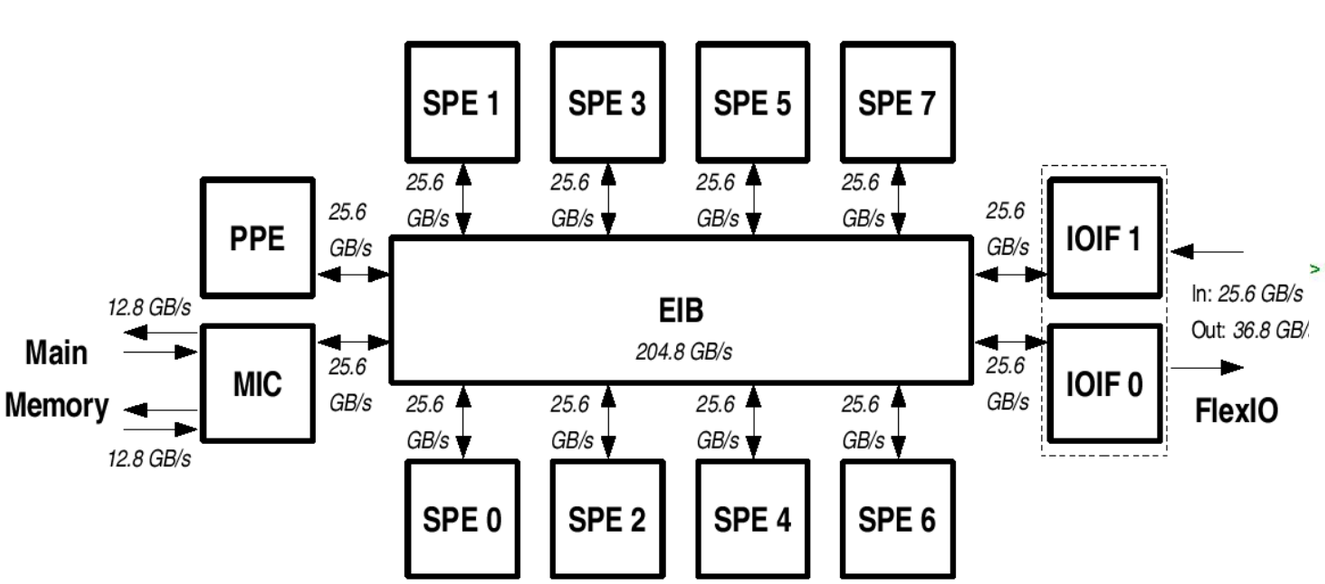
System

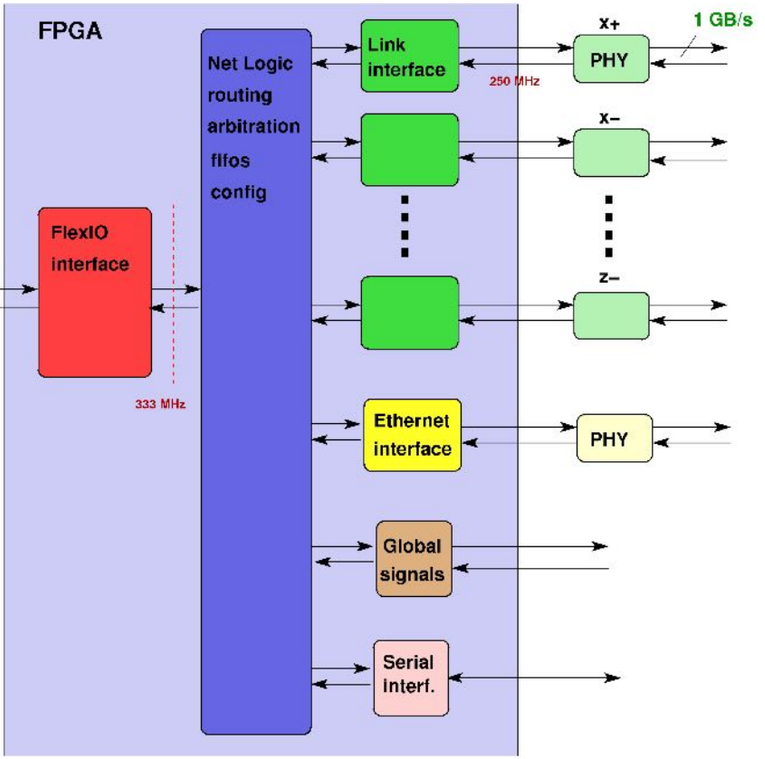

System
1 to 72 or more Racks
Rack
Cabled 8x8x16
32 Node Cards
1024 chips, 4096 procs
Node Card
(32 chips 4x4x2)
32 compute, 0-2 IO cards
Compute Card
1 chip, 20
DRAMs
Chip
4 processors
13.6 GF/s
8 MB EDRAM
13.6 GF/s
2.0 GB DDR
Supports 4-way SMP
435 GF/s
64 GB
14 TF/s
2 TB
1 PF/s +
144 TB +

Example: Cell / QPACE

- Idea: Combine enhanced Cell BE (PowerXCell 8i) with APE-like torus network
  - 200 GFlop/s single precision (peak)
  - 100 GFlop/s double precision (peak)
  - DDR2 memory controller
  - $\sim 25\%$ sustained performance (performance model)
- Collaboration between academic partners (Uni Regensburg, Jülich, DESY, Milano, Ferrara) and IBM Research Lab Böblingen
- Funded as part of SFB/TR-55 by Deutsche Forschungsgemeinschaft (DFG)
- Design work started $\approx$ Jan 2008
- Running Board Jul 2008
- Production of 2048-node machine early 2009

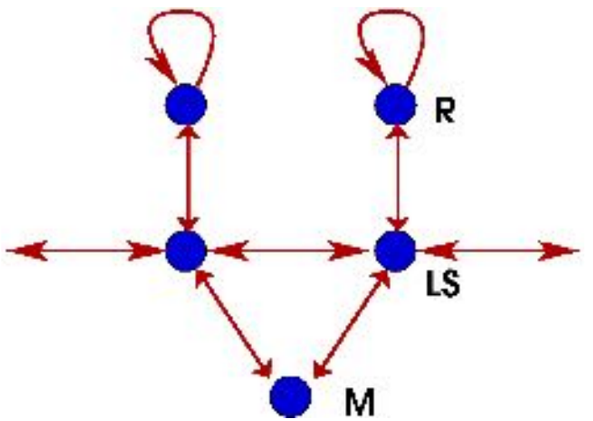
Cell/QPACE



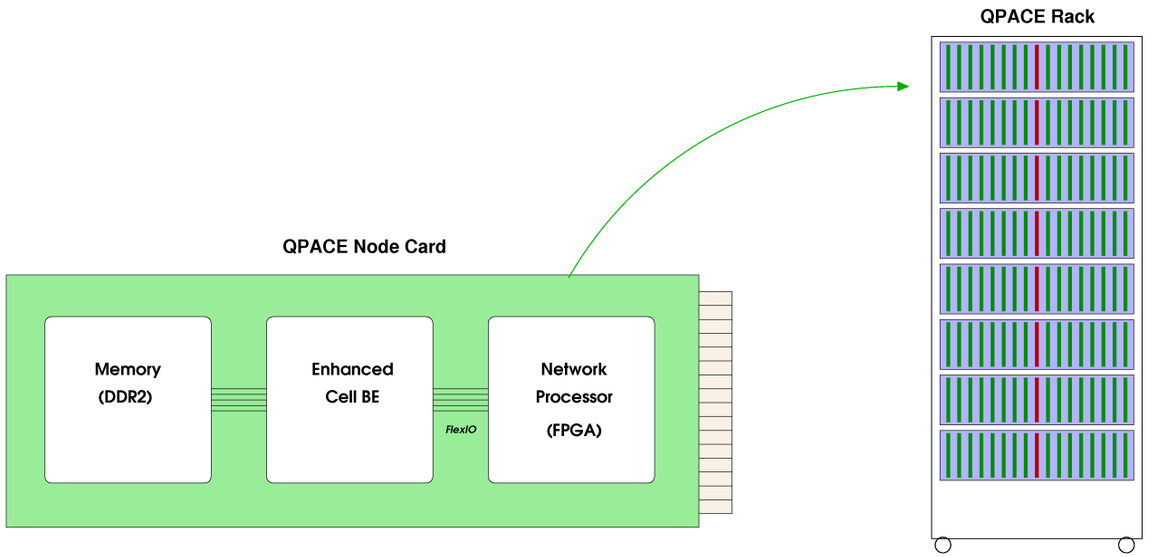


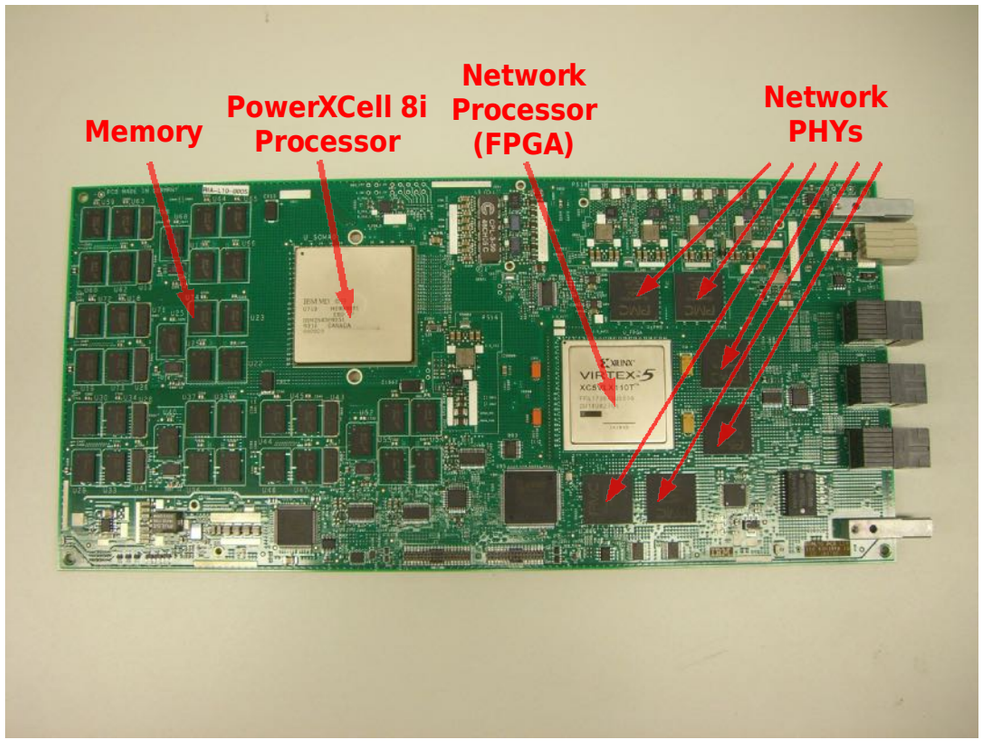
SPE:

- In-order execution
- 4 MulAdd/clk = 25.6 Gflops (SP)
- Local Store (LS) 256 k
- Data transport memory ↔ LS can (and must) be controlled by SW



Cell/QPACE

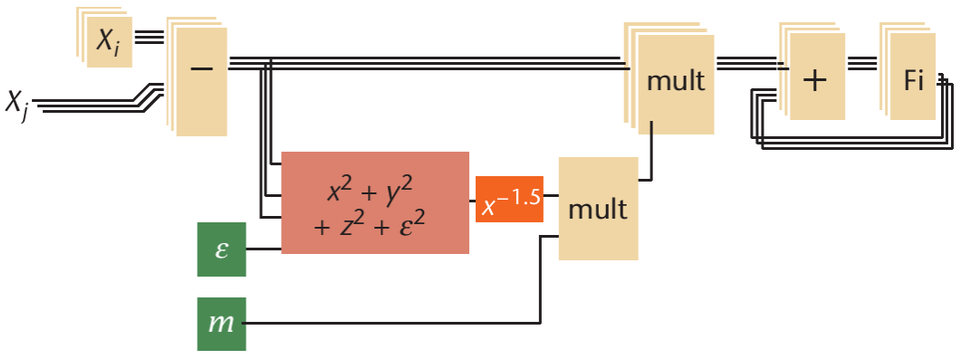

QPACE Rack
QPACE Node Card
Memory
(DDR2)
Enhanced
Cell BE
FlexIO
Network
Processor
(FPGA)


Memory
PowerXCell 8i
Processor
Network
Processor
(FPGA)
Network
PHYs

Example: GRAPE

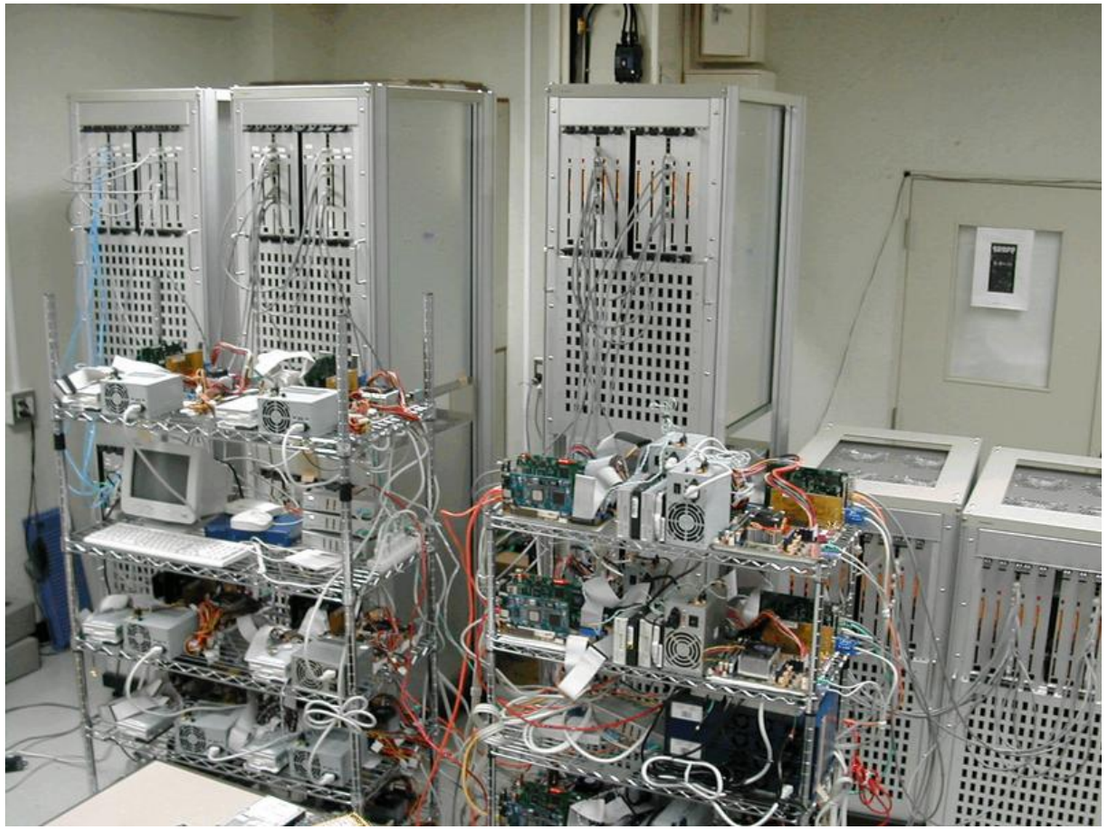
History:

1989	Idea by physicists at Tokyo University to implement N-body computation in HW
	GRAPE-1 prototype
1991	GRAPE-1A, GRAPE-2
	discrete commercial chips, 40 MFlops
1993	GRAPE-3
	first custom LSI chip
1996	MD-GRAPE
1997	GRAPE-4
	19 flop/3 $\times$ 32 MHz = 203 Mflops
2000	GRAPE-5
2001	Gordon Bell Prize for GRAPE-6 prototype
2003	GRAPE-6
	6 pipelines $\times$ 57 flop $\times$ 90 MHz = 30 Gflops
2005	MD-GRAPE-3
	20 pipelines $\times$ 36 flop $\times$ 250 MHz = 180 Gflops



N.B.: Odd series have low accuracy, even series have high accuracy

GRAPE-6



- Pipeline = 60 arithmetic units
- FPGA = 6 pipelines
- PCI module (123 Gflops) = 4 chips + FPGA + SRAM
- Mother board (1 Tflops) = 8 modules
- Full system (64 Tflops) = 64 mother boards

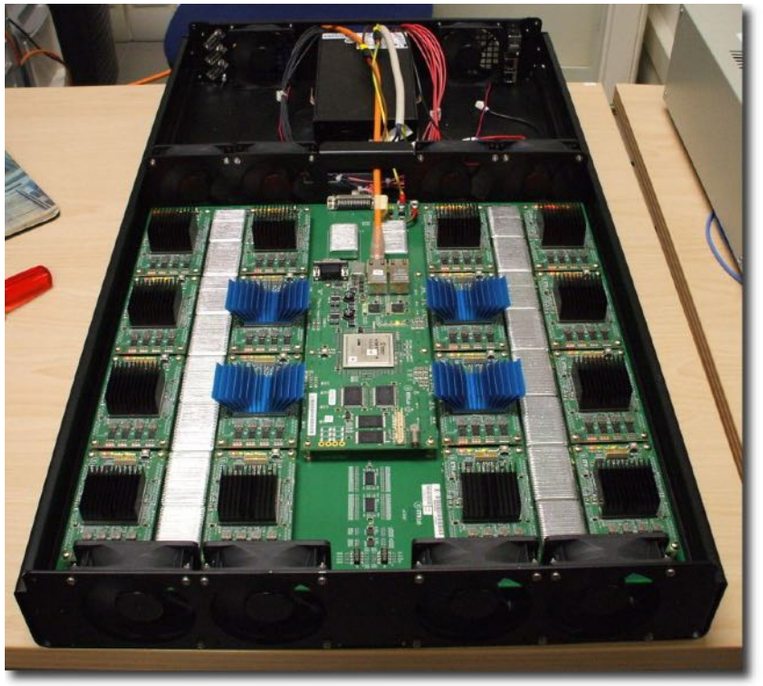
Example: IANUS

History:

- 1991: RTN with transputers (Zaragoza)
- 2000: SUE "Spin Update Engine" with FPGA (Zaragoza)
  $\longrightarrow$ 217 ps/update on full machine
- 2006: IANUS "Spin Update Engine" with FPGA (Zaragoza + Ferrara)
  $\longrightarrow$ 1 ps/update on one PB

Architecture:

- FPGA (62.5 MHz)
  - 512 update engines (RNG+LUT)
  - on-chip memory
  - links for 2-d network
- PB = 16 FPGA (4×4)



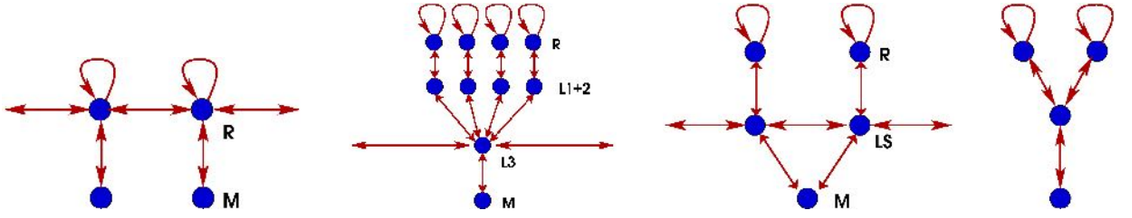
Simulations of Edward-Anderson Spin Glas

$$H = -\sum_{\langle i,j \rangle} J_{ij} \cdot x_i \cdot x_j \quad (J_{ij} = \pm 1 \text{ random})$$

allows 10^{11} MC steps on $L^3 = 80^3$ with $\Delta t = 10^{-12}$ sec $\rightarrow$ 0.1 sec

Comparison of Machines used for LQCD

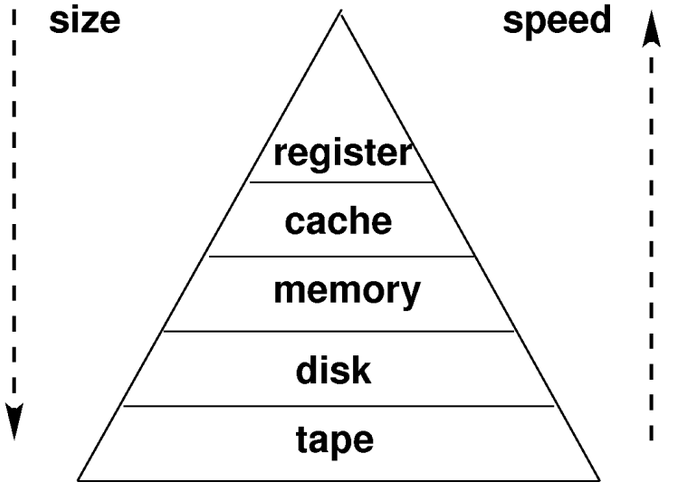
	unit	apeNEXT 2006	BG/P 2008	Cell 2009	PC 2009
Arithmetics					
f_{clk}	[GHz]	0.13	0.85	3.2	2.8
FP word	[bits]	64	64	64 (32)	64 (32)
FP/core	[flop/clk]	8	4	4	2×2
core/chip		1	4	8	4
FP/chip	[Gflops]	1	13.6	100	50
power (overall)	[W/Gflops]	9	3	1.5	2.5
Cache		—	L3	LS	L3
size	[word]	—	1 M	8×32 K	1 M
Memory					
bandwidth	[word/flop]	1/4	1/8	1/32	1/12.5
latency	[clk]	≈ 20	≥ 30	≥ 200	≥ 100
Network		3d	3d + tree	3d	—
bandwidth	[word/flop]	1/24	1/64	$< 1/16$	—
latency	[clk]	≈ 40	≈ 700	≈ 3000	—



Simplified Hardware Model

Distinction between devices/units for:

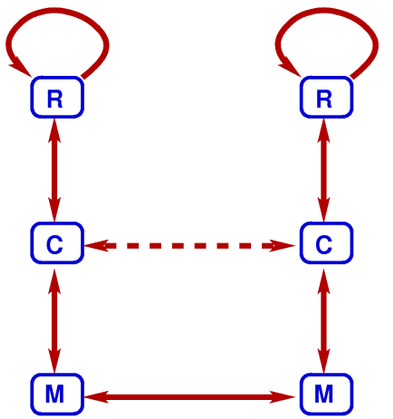
- control (of data and program flow)



- storage of data (and code)
  - memory
  - cache(s)
  - registers
  - internal buffers, fifos, flip-flops, ...

  σ_x = storage size

- processing/transport of data
  - arithmetic pipelines
  - storage access (hopefully pipelined)
  - combinatorical logics
  - buses?



β_{xy} = bandwidth (data throughput/time)
λ_{xy} = latency (delay between input and first output)
ISA = instruction set architecture

Implementation of Computational Tasks

Three inter-related **problems:**

(1) translation of the computational task into hardware operations

(2) allocation of the hardware resources for data storage and transport

(3) scheduling of the operations

N.B.: (2) and (3) are NP-hard

. . . need to be tackled at various abstraction **levels:**

- development or selection of algorithm
- development of a high-level code
- code generation by the compiler
  (code selection, register allocation, and instruction scheduling)
- out-of-order execution by hardware (micro-instructions)

DAG of computational task:		Model of hardware architecture:
• vertices: operations	$\leftrightarrow$	• edges: transport devices
• edges: variables	$\leftrightarrow$	• vertices: storage devices