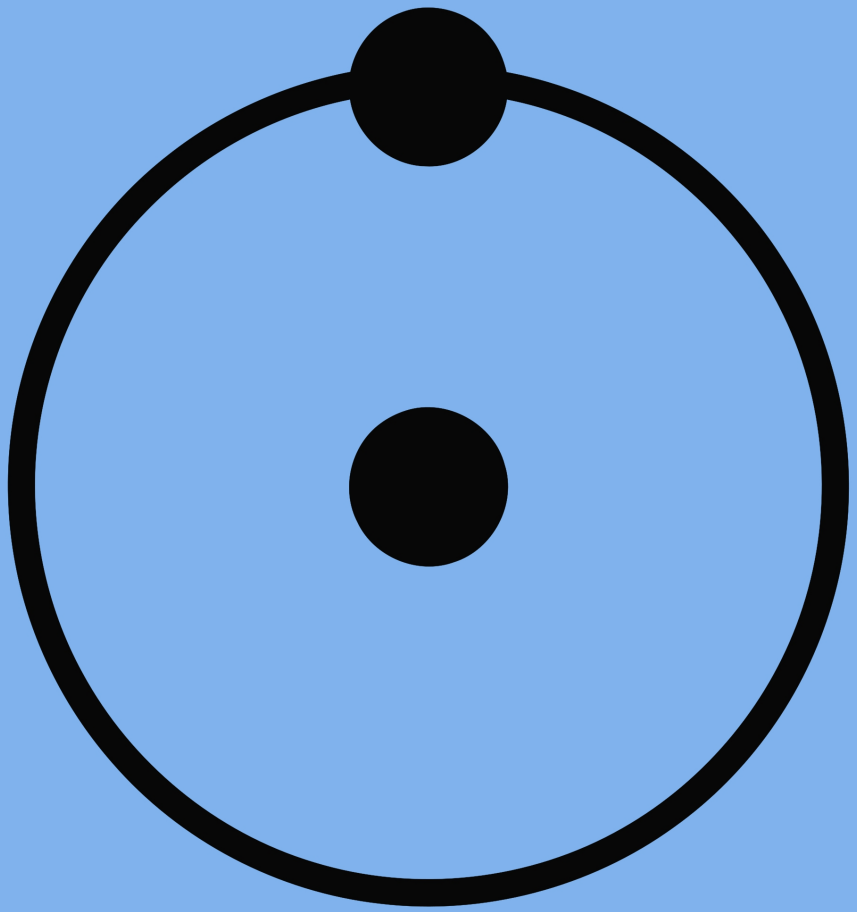


Molecular dynamics simulations and drug discovery

Jacob D. Durrant, J. Andrew McCammon

BMC Biology 2011 9:71 | DOI: 10.1186/1741-7007-9-71

"With constant improvements in both computer power and algorithm design, the future of computer-aided drug design is promising; molecular dynamics simulations are likely to play an increasingly important role."

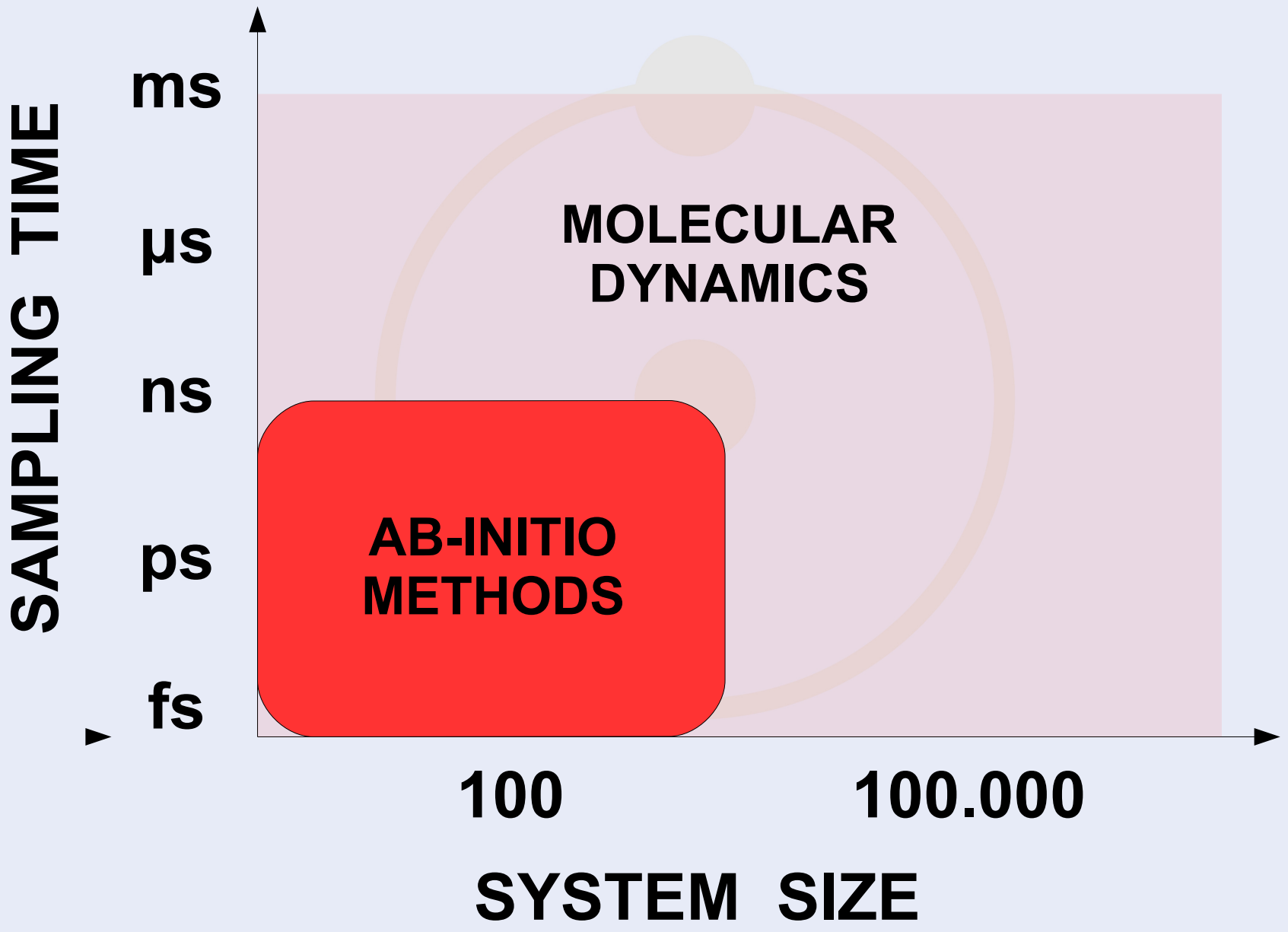
First principles computational materials design for energy storage materials in lithium ion batteries

Ying Shirley Meng, M. Elena Arroyo-de Dompablo

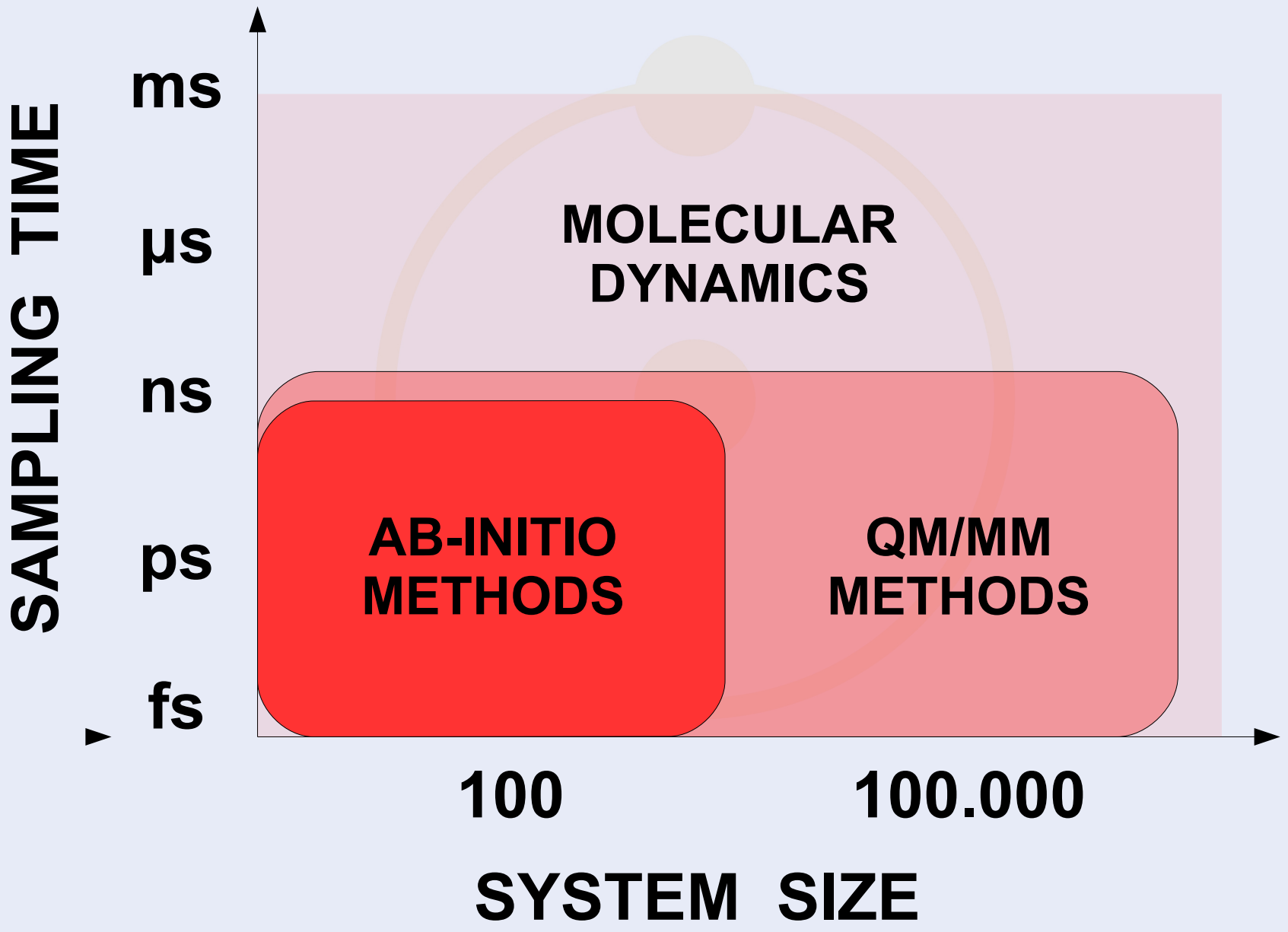
Energy Environ. Sci. 2009, 2, 589–609 | DOI: 10.1039/b901825e

"Specifically, we show how each relevant property can be related to the structural component in the material and can be computed from first principles."

Map for Atomistic Simulations



Map for Atomistic Simulations

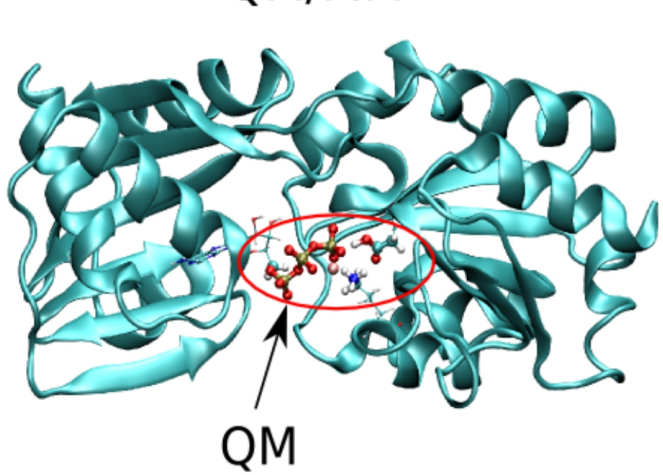


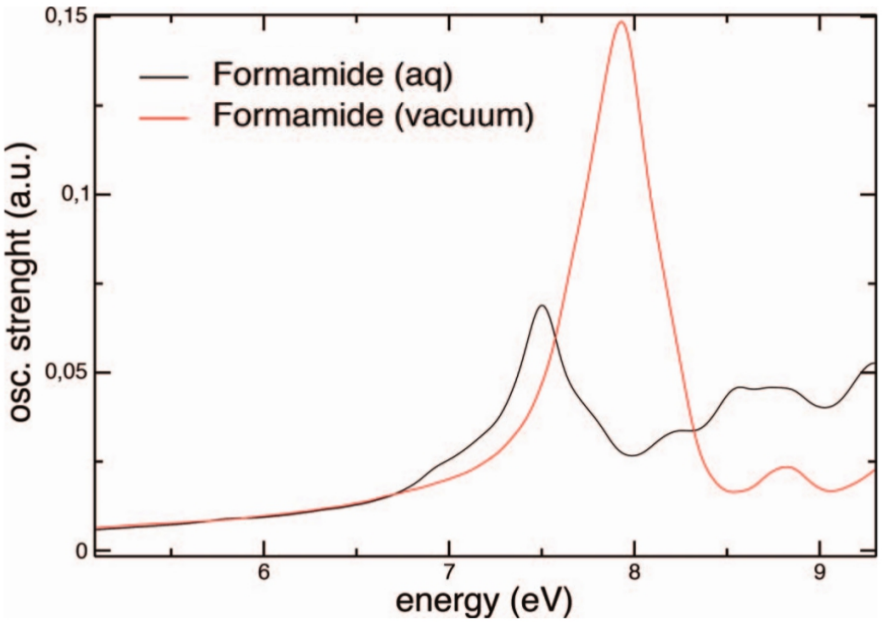
Nobel Prize 2013 in Chemistry: Development of Multiscale Models

Martin Karplus
Harvard University

Michael Levitt
Stanford Universiry

Arieh Warshel
Southern California University





Further details:

Electron dynamics in complex environments with real-time time dependent density functional theory in a QM-MM framework

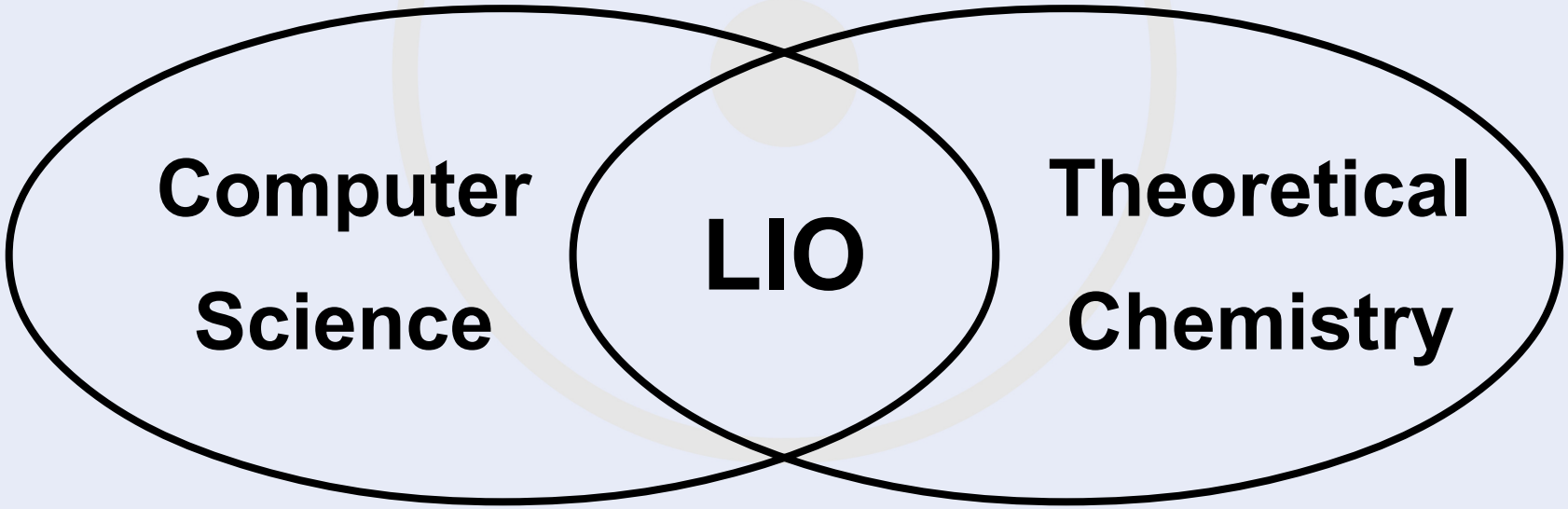
U.N. Morzan et al.

The Journal of Chemical Physics 140, 164105 (2014).

DOI: 10.1063/1.4871688

THE LIO PROJECT

"Implementing novel methods of atomistic ***quantum simulation*** *that* ***efficiently use*** *the current state-of-the-art hardware tools"*



github.com/nanolebrero/lio

THE LIO PROJECT

Quantum Calculations (QM/MM with Amber)

DFT + Gaussian Basis

Single Point Calculation

Born-Oppenheimer Dynamics

Electronic Propagation

TD-DFT Electron Dynamics

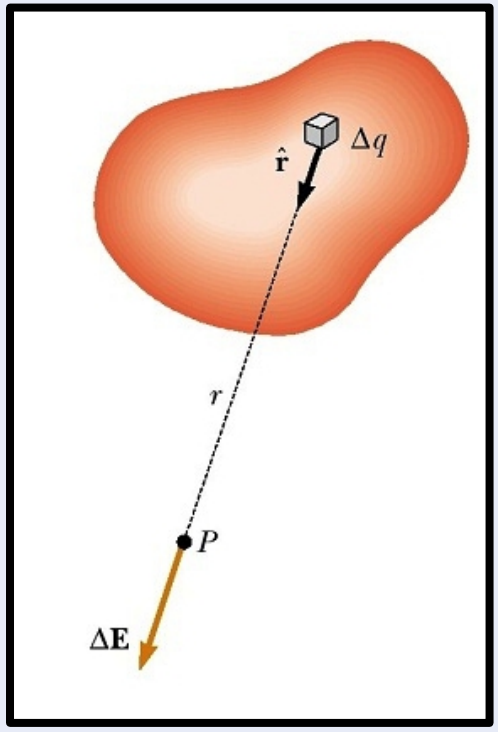
Ehrenfest Dynamics

github.com/nanolebrero/lio

DFT: The way of the density

Hohenberg - Khon Theorem

$$\rho(x, y, z) \longleftrightarrow \Psi(\vec{r}_1, \ldots, \vec{r}_n)$$



"Motions are governed not by deterministic laws, but **by probability functions**; chemical bonds are formed not mechanically, but by **shifting clouds of electrons** that are simultaneously waves and particles."

Molecular dynamics simulations and drug discovery

DFT: The way of the density

Hohenberg - Khon Theorem

$$\rho(x, y, z) \longleftrightarrow \Psi(\vec{r}_1, \ldots, \vec{r}_n)$$

$$\rho(x, y, z) = \sum_{i=1}^{M} \sum_{j=1}^{M} P_{ij} \cdot \chi_i(x, y, z) \cdot \chi_j(x, y, z)$$

$$E[\rho] = T[\rho] + V_{ne}[\rho] + V_{ee}[\rho] + E_{XC}[\rho]$$

$$E[\rho] = T[\rho] + V_{ne}[\rho] + V_{ee}[\rho] + E_{XC}[\rho]$$

Analytical Resolution:

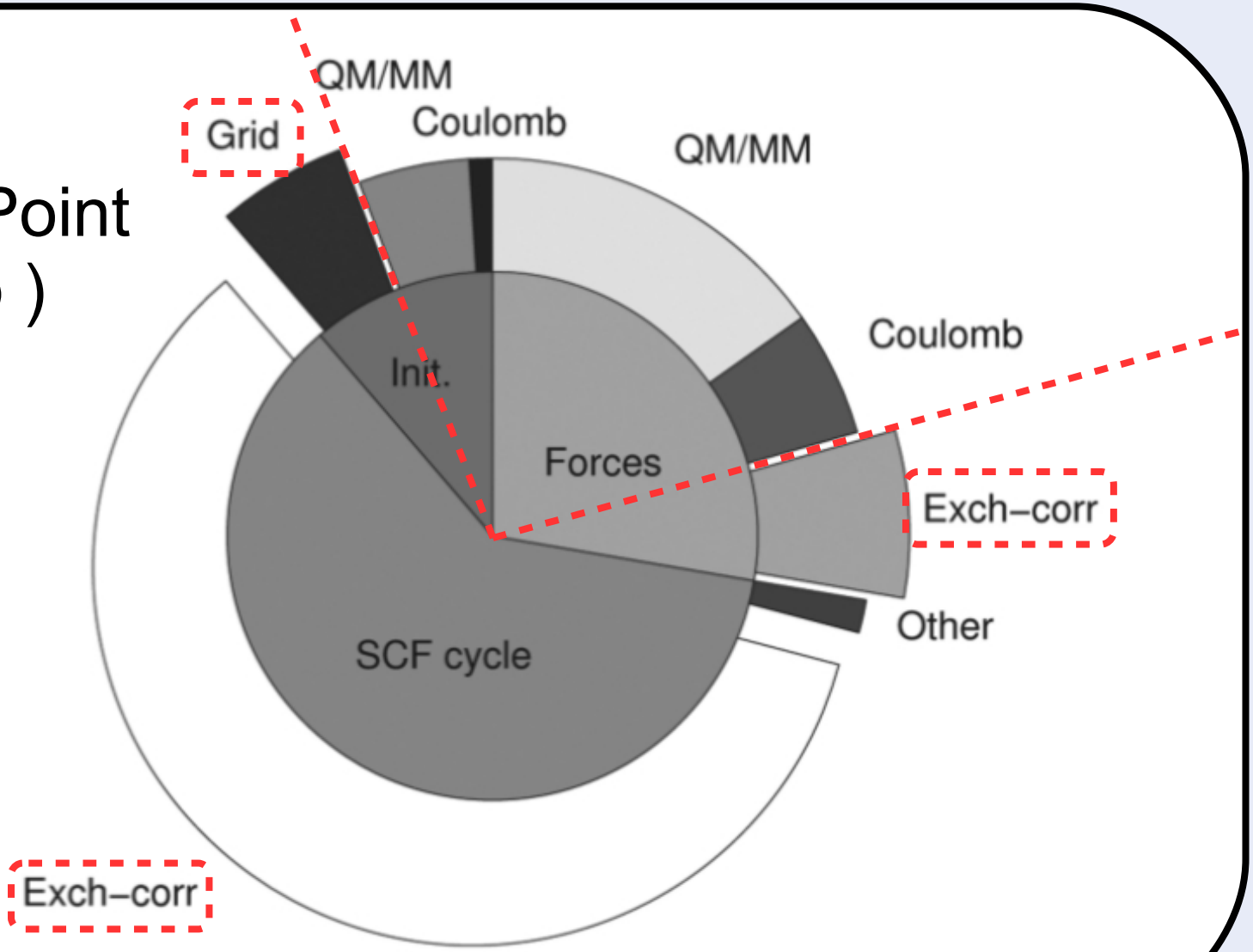
$$\rho(r) = \sum_{i=1}^{M} \sum_{j=1}^{M} P_{ij} \cdot \chi_i(r) \cdot \chi_j(r)$$

gets distributed

$$= T[P] + V_{ne}[P] + V_{ee}[P]$$

$$E[\rho] = T[\rho] + V_{ne}[\rho] + V_{ee}[\rho] + E_{XC}[\rho]$$

Single Point
(1 Step)



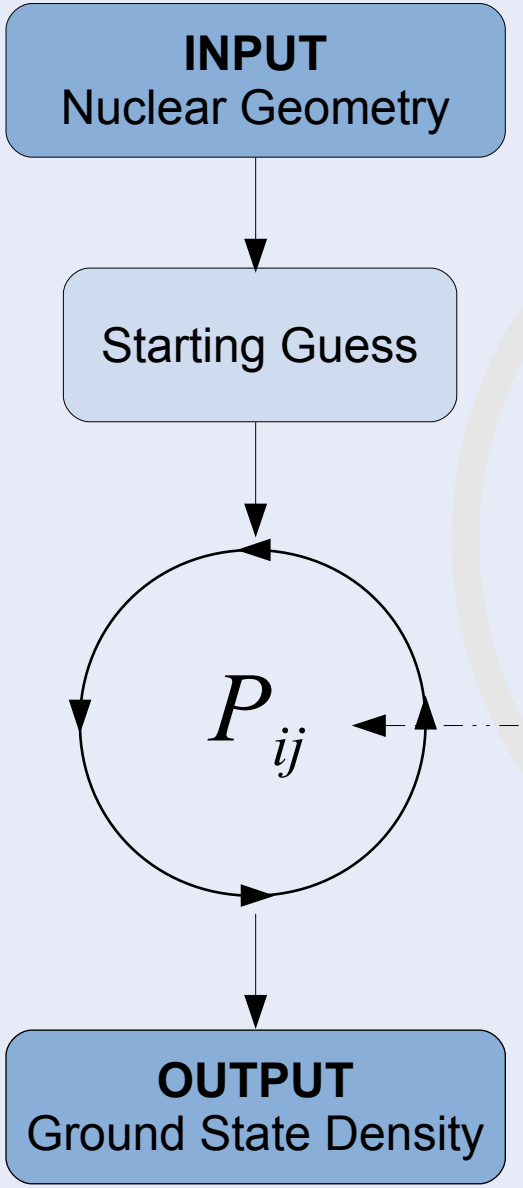
$$E_{XC}[\,\rho\,] \simeq \iiint \epsilon_{XC}[\,\rho(x,y,z), \nabla\rho(x,y,z)\,]\,dV$$

$$E_{XC}[\,\rho\,] \simeq \sum_{k} \epsilon_{XC}[\,\rho(r_k), \nabla\rho(r_k)\,] \cdot V_k$$

$$\rho(r_k) = \sum_{i=1}^{M}\sum_{j=1}^{M} P_{ij}\,\chi_i(r_k)\,\chi_j(r_k)$$

$$\frac{\partial\rho}{\partial x} = \sum_{i=1}^{M} \frac{\partial\chi_i}{\partial x} \sum_{j=1}^{M} P_{ij}\,\chi_j + \sum_{i=1}^{M} \chi_i \sum_{j=1}^{M} P_{ij} \frac{\partial\chi_j}{\partial x}$$

SINGLE POINT CALCULATION



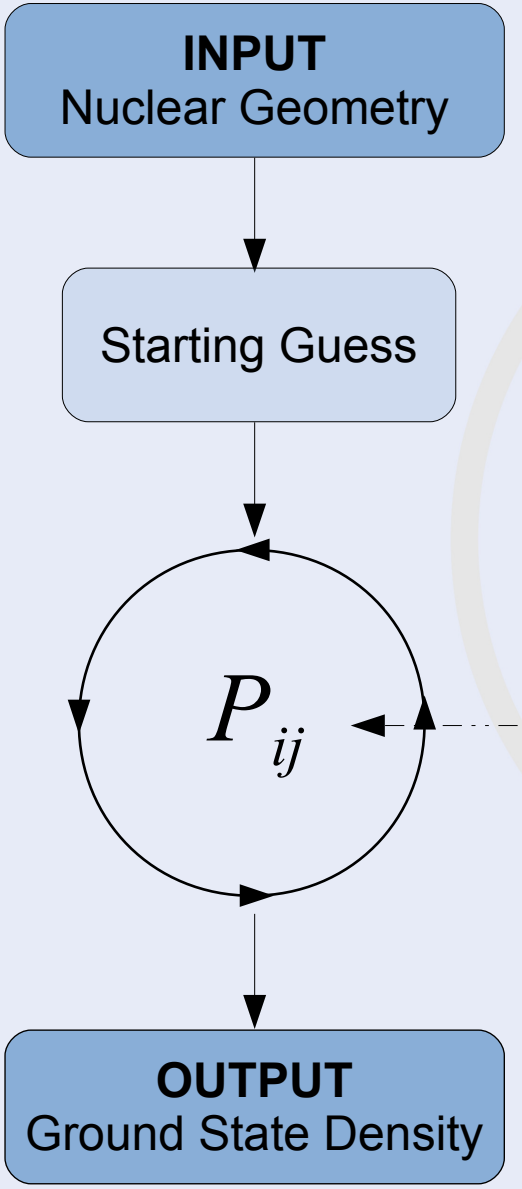
$E_{XC}[\,\rho\,]$

(1) Compute Basis Values.

(2) Compute Density Values.

(3) Numeric Integral

SINGLE POINT CALCULATION



$$\begin{matrix} \chi_1(r_1) & \cdots & \chi_1(r_K) \\ \cdots & \cdots & \cdots \\ \chi_M(r_1) & \cdots & \chi_M(r_K) \end{matrix}$$

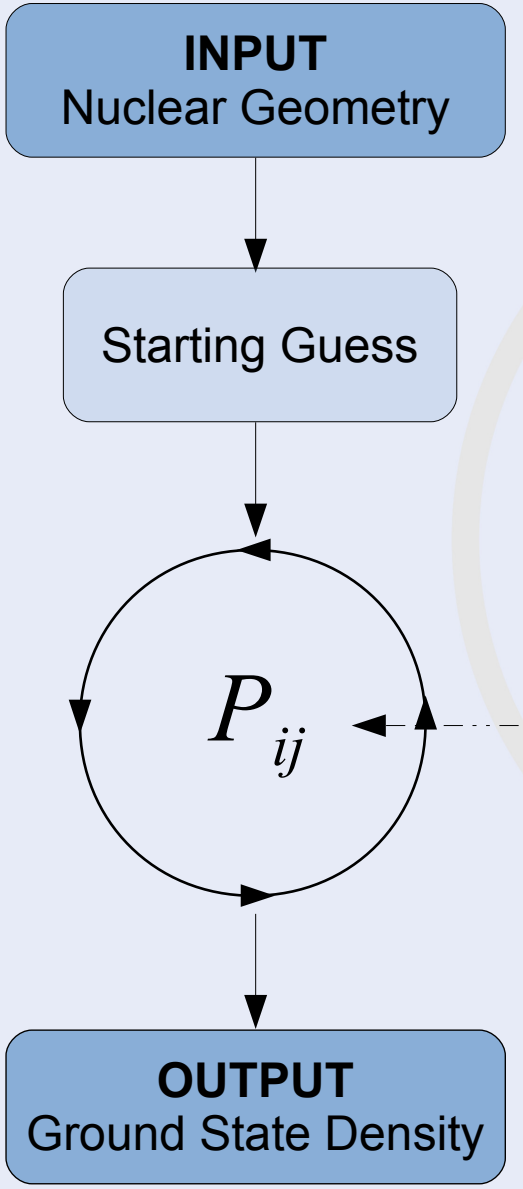
$E_{XC}[\,\rho\,]$

(1) Compute Basis Values.

(2) Compute Density Values.

(3) Numeric Integral

SINGLE POINT CALCULATION



$$\rho(r_k) = \sum_{i=1}^{M} \sum_{j=1}^{M} P_{ij}\, \chi_i(r_k)\, \chi_j(r_k)$$

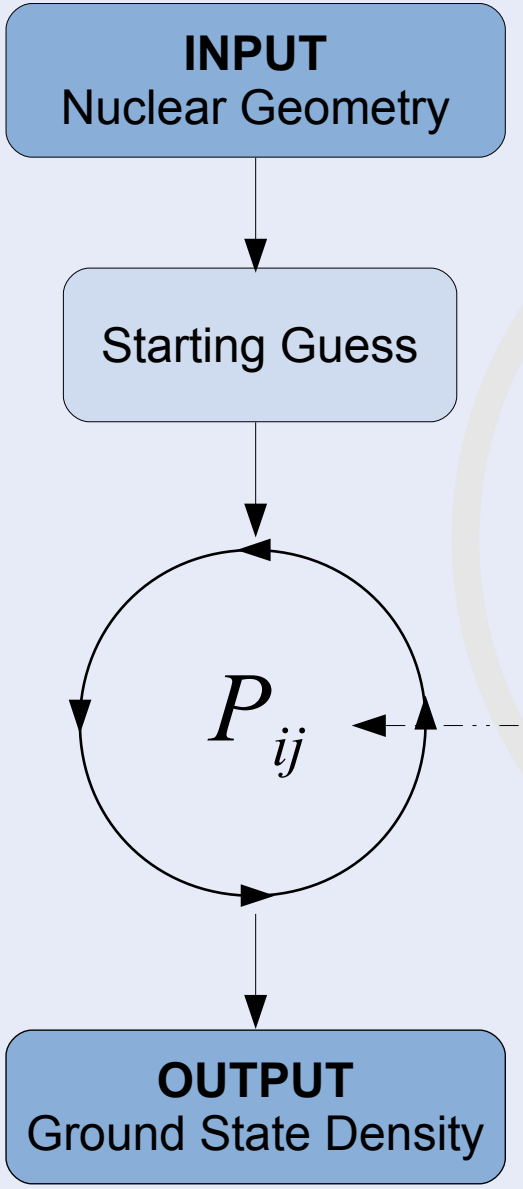
$E_{XC}[\,\rho\,]$

(1) Compute Basis Values.

(2) Compute Density Values.

(3) Numeric Integral

SINGLE POINT CALCULATION



$$\sum_k \epsilon_{XC}[\,\rho(r_k), \nabla\rho(r_k)\,] \cdot V_k$$

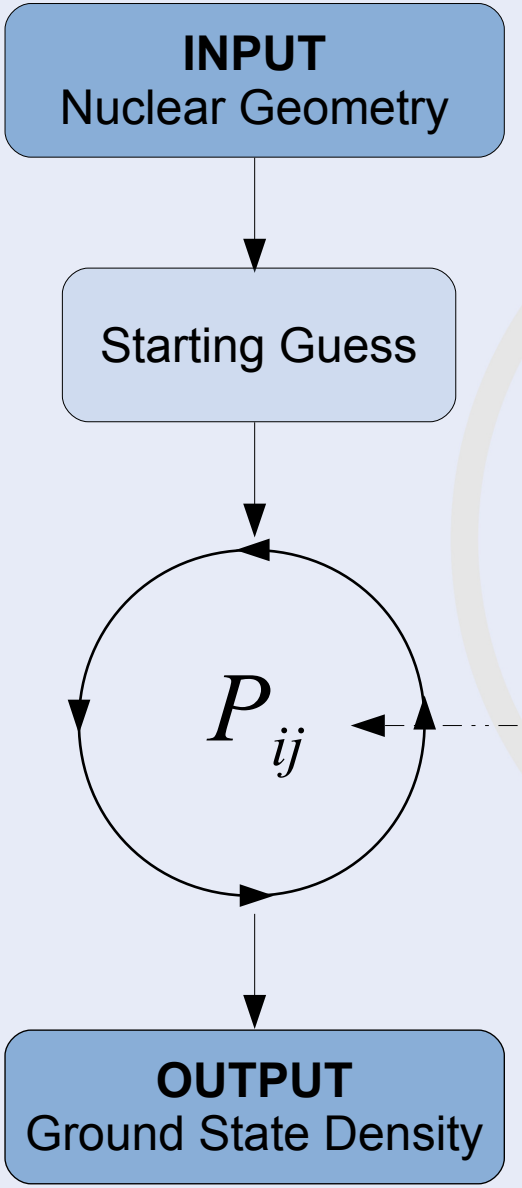
$$E_{XC}[\,\rho\,]$$

(1) Compute Basis Values.

(2) Compute Density Values.

(3) Numeric Integral

SINGLE POINT CALCULATION



$$\rho(r_k) = \sum_{i=1}^{M} \sum_{j=1}^{M} P_{ij}\, \chi_i(r_k)\, \chi_j(r_k)$$

$E_{XC}[\,\rho\,]$

(1) Compute Basis Values.

(2) Compute Density Values.

(3) Numeric Integral

$$\rho(r_k) = \sum_{i=1}^{M} \sum_{j=1}^{M} P_{ij}\, \chi_i(r_k)\, \chi_j(r_k)$$

1 Thread = 1 Point + Double Sum

$$\rho(r_k) = \sum_{i=1}^{M} \sum_{j=1}^{M} P_{ij}\, \chi_i(r_k)\, \chi_j(r_k)$$

$$\rho(r_k) = \sum_{i=1}^{M} \chi_i(r_k) \sum_{j=1}^{M} P_{ij}\, \chi_j(r_k)$$

1 Thread = 1 Point + Only j-Sum

(+ other subroutine to i-accumulate)

$$\rho(r_k) = \sum_{i=1}^{M} \sum_{j=1}^{M} P_{ij}\, \chi_i(r_k)\, \chi_j(r_k)$$

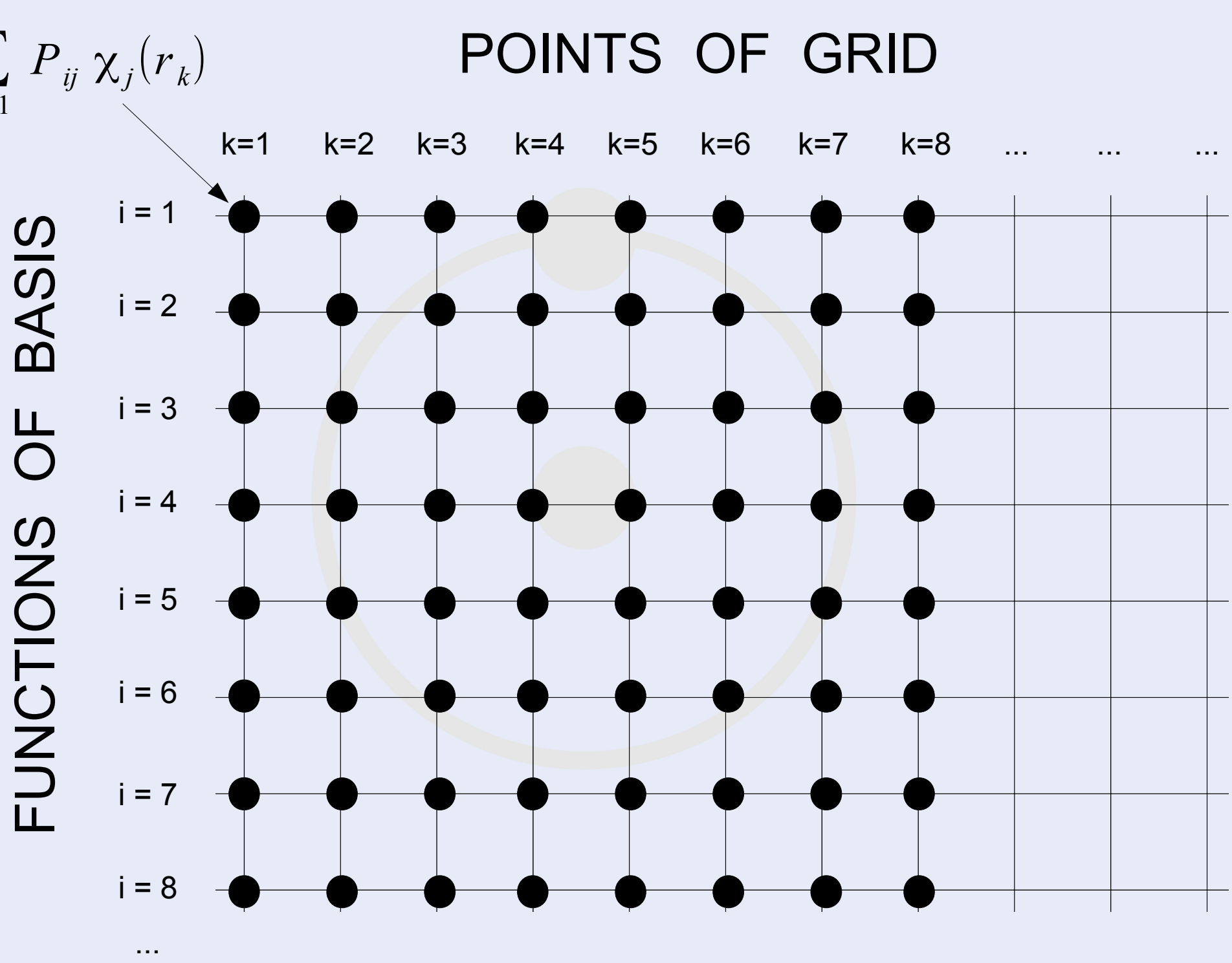
$$\rho(r_k) = \sum_{i=1}^{M} \chi_i(r_k) \sum_{j=1}^{M} P_{ij}\, \chi_j(r_k)$$

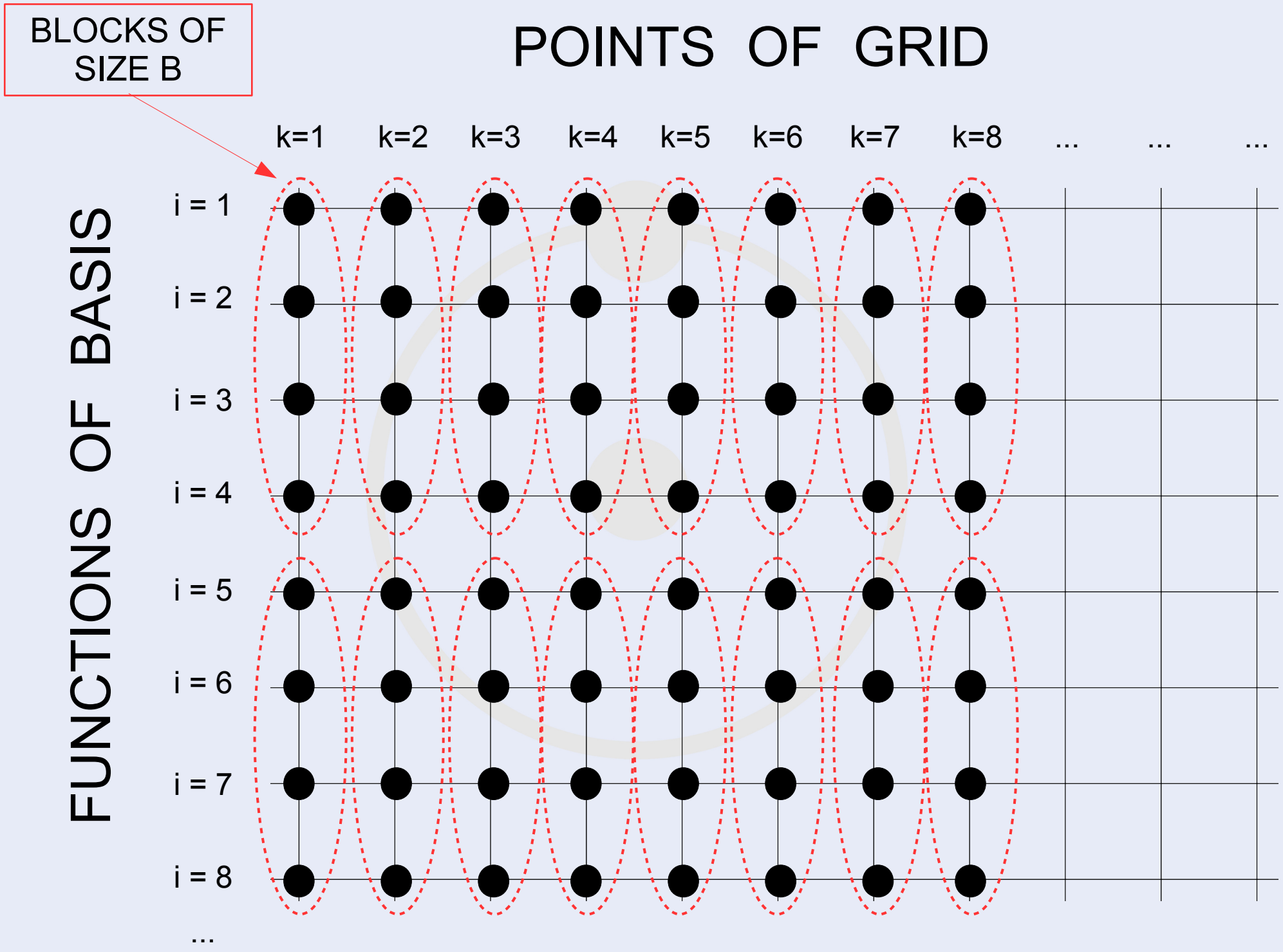
SAME NUMBER

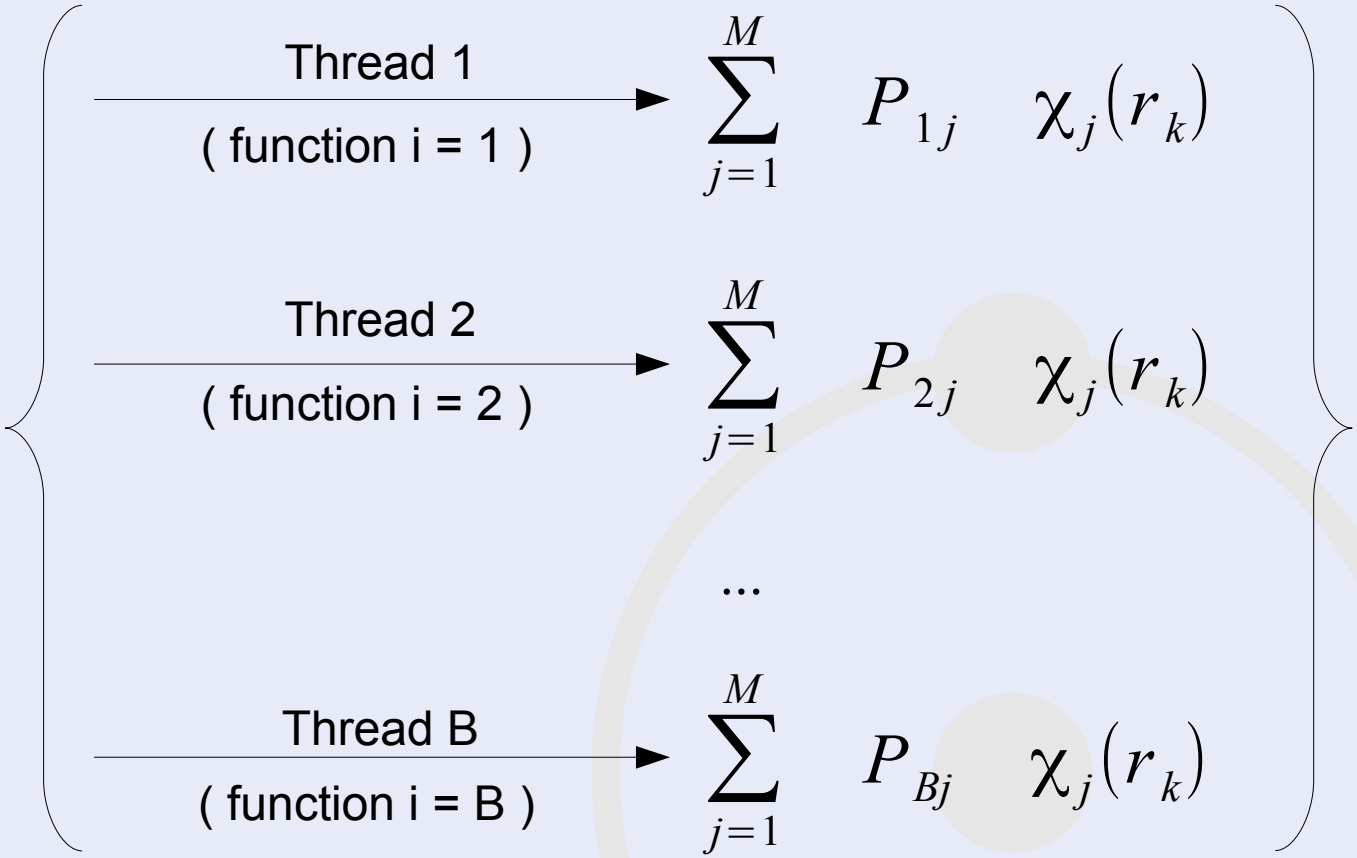
SAME PATTERN

$$\frac{\partial \rho}{\partial x} = \sum_{i=1}^{M} \frac{\partial \chi_i}{\partial x} \sum_{j=1}^{M} P_{ij}\, \chi_j + \sum_{i=1}^{M} \chi_i \sum_{j=1}^{M} P_{ij} \frac{\partial \chi_j}{\partial x}$$

$$\sum_{j=1}^{M} P_{ij}\,\chi_j(r_k)$$

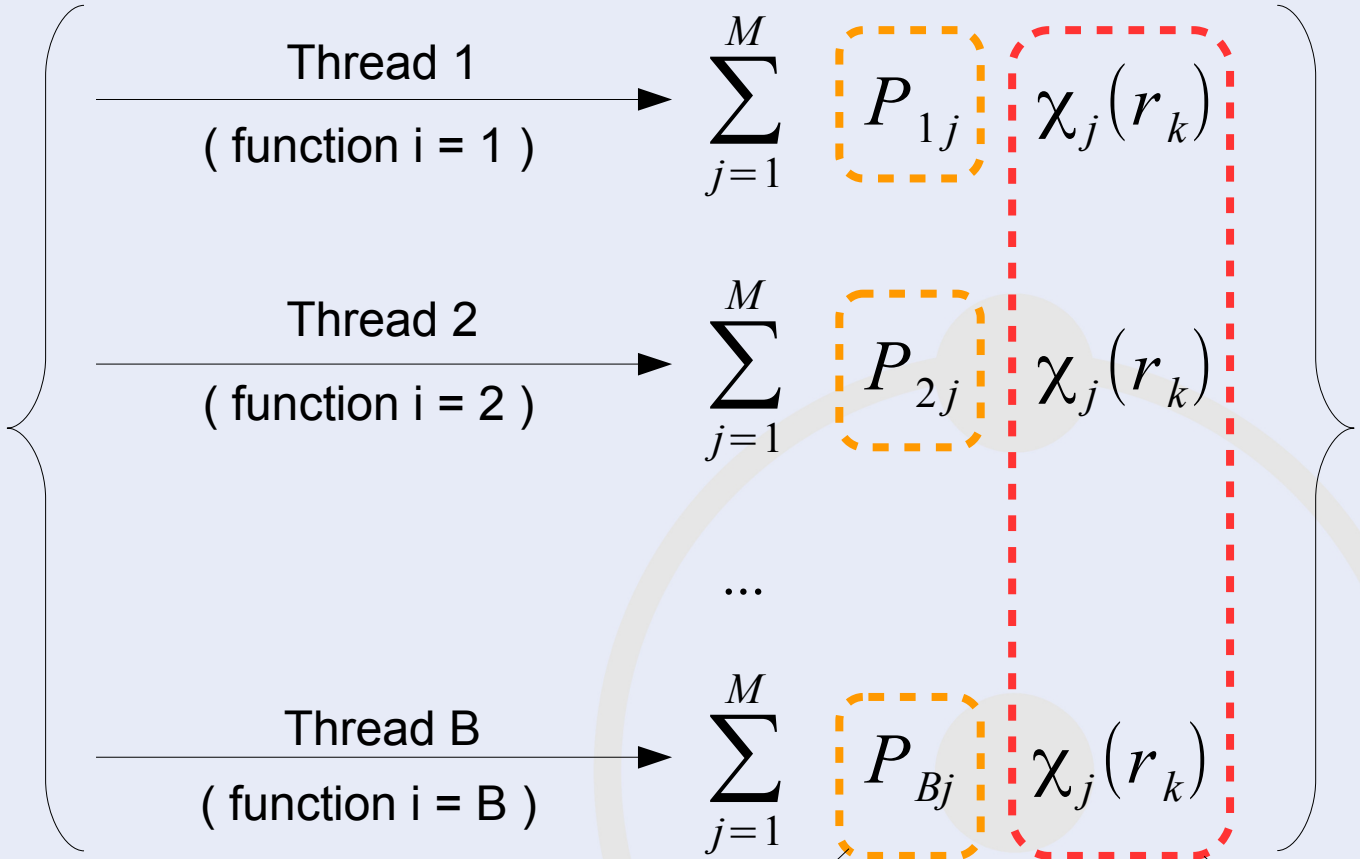



BLOCKS OF SIZE B
POINTS OF GRID
FUNCTIONS OF BASIS
k=1
k=2
k=3
k=4
k=5
k=6
k=7
k=8
...
...
...
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
...



BLOCKS OF SIZE B

Point k fixed



BLOCKS OF SIZE B

Point k fixed

Different sets of j = 1 to M for all threads

The **same** set of j = 1 to M for all threads

$P_{i1}, P_{i2}, \ldots, P_{iM}$

$\chi_1(r_k), \chi_2(r_k), \ldots, \chi_M(r_k)$

$$
\begin{cases}
\rightarrow \sum_{j=1}^{B} P_{1j}\,\chi_j(r_k) + \sum_{j=B+1}^{2B} P_{1j}\,\chi_j(r_k) + \ldots + \sum_{j=M-B+1}^{M} P_{1j}\,\chi_j(r_k) \\
\rightarrow \sum_{j=1}^{B} P_{2j}\,\chi_j(r_k) + \sum_{j=B+1}^{2B} P_{2j}\,\chi_j(r_k) + \ldots + \sum_{j=M-B+1}^{M} P_{2j}\,\chi_j(r_k) \\
\ldots \\
\rightarrow \sum_{j=1}^{B} P_{Bj}\,\chi_j(r_k) + \sum_{j=B+1}^{2B} P_{Bj}\,\chi_j(r_k) + \ldots + \sum_{j=M-B+1}^{M} P_{Bj}\,\chi_j(r_k)
\end{cases}
$$

$$\rightarrow \sum_{j=1}^{B} P_{1j}\,\chi_j(r_k) + \sum_{j=B+1}^{2B} P_{1j}\,\chi_j(r_k) + \ldots + \sum_{j=M-B+1}^{M} P_{1j}\,\chi_j(r_k)$$

$$\rightarrow \sum_{j=1}^{B} P_{2j}\,\chi_j(r_k) + \sum_{j=B+1}^{2B} P_{2j}\,\chi_j(r_k) + \ldots + \sum_{j=M-B+1}^{M} P_{2j}\,\chi_j(r_k)$$

$$\ldots$$

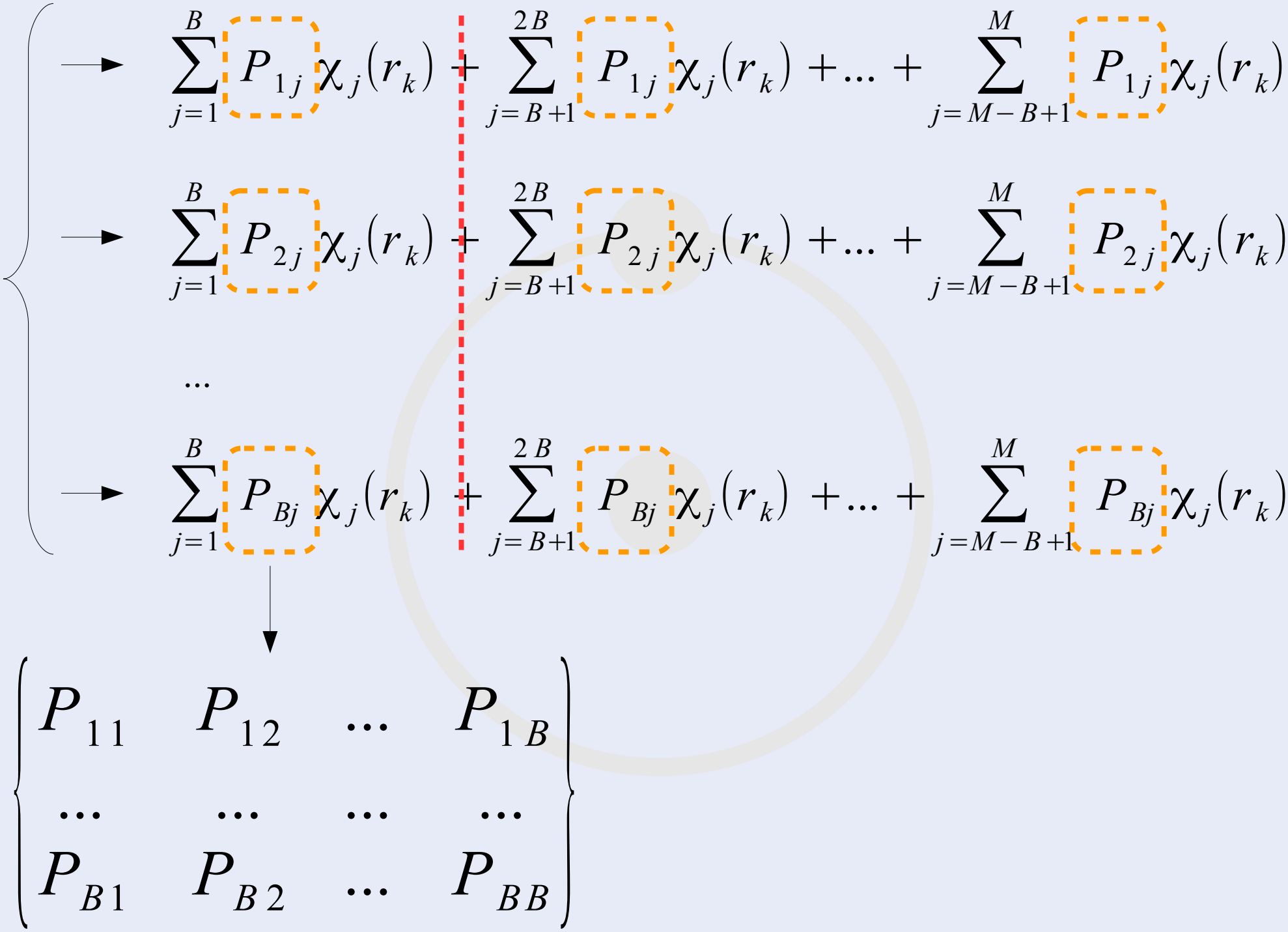
$$\rightarrow \sum_{j=1}^{B} P_{Bj}\,\chi_j(r_k) + \sum_{j=B+1}^{2B} P_{Bj}\,\chi_j(r_k) + \ldots + \sum_{j=M-B+1}^{M} P_{Bj}\,\chi_j(r_k)$$

(1) Load functions “j” = 1 to B (one each thread) to **shared mem**.

(2) __syncthreads()

(3) Each thread performs sums from 1 to B.

Repeat (1) to (3) for “j” = B+1 to 2B, 2B+1 to 3B, etc.


$\sum_{j=1}^{B} P_{1j}\chi_j(r_k) + \sum_{j=B+1}^{2B} P_{1j}\chi_j(r_k) + \ldots + \sum_{j=M-B+1}^{M} P_{1j}\chi_j(r_k)$
$\sum_{j=1}^{B} P_{2j}\chi_j(r_k) + \sum_{j=B+1}^{2B} P_{2j}\chi_j(r_k) + \ldots + \sum_{j=M-B+1}^{M} P_{2j}\chi_j(r_k)$
...
$\sum_{j=1}^{B} P_{Bj}\chi_j(r_k) + \sum_{j=B+1}^{2B} P_{Bj}\chi_j(r_k) + \ldots + \sum_{j=M-B+1}^{M} P_{Bj}\chi_j(r_k)$
$\begin{Bmatrix} P_{11} & P_{12} & \cdots & P_{1B} \\ \cdots & \cdots & \cdots & \cdots \\ P_{B1} & P_{B2} & \cdots & P_{BB} \end{Bmatrix}$

$$\sum_{j=1}^{B} P_{1j}\,\chi_j(r_k) + \sum_{j=B+1}^{2B} P_{1j}\,\chi_j(r_k) + \ldots + \sum_{j=M-B+1}^{M} P_{1j}\,\chi_j(r_k)$$

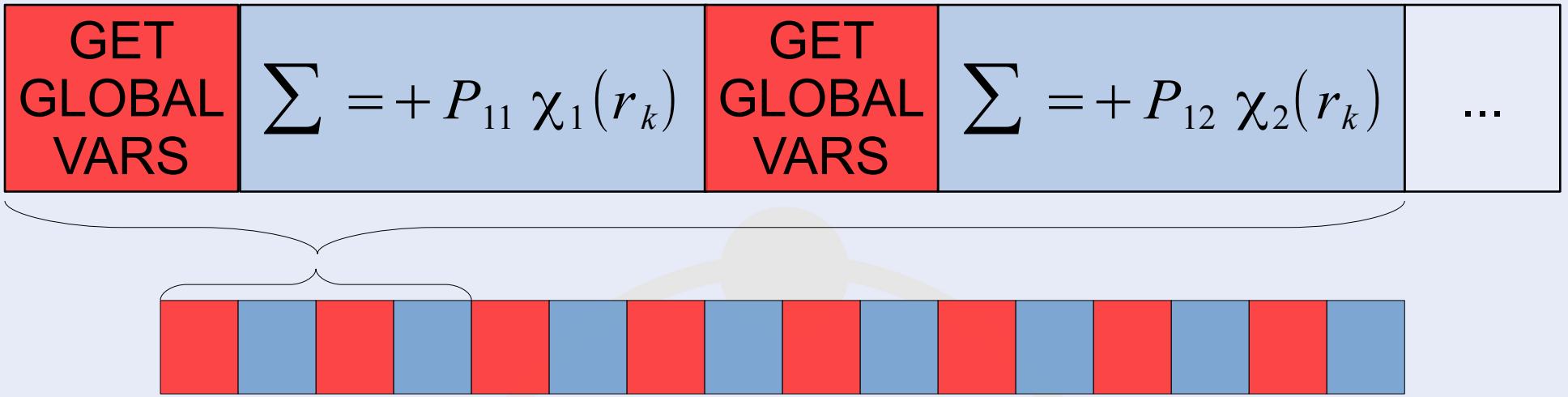
$$\sum_{j=1}^{B} P_{2j}\,\chi_j(r_k) + \sum_{j=B+1}^{2B} P_{2j}\,\chi_j(r_k) + \ldots + \sum_{j=M-B+1}^{M} P_{2j}\,\chi_j(r_k)$$

$$\ldots$$

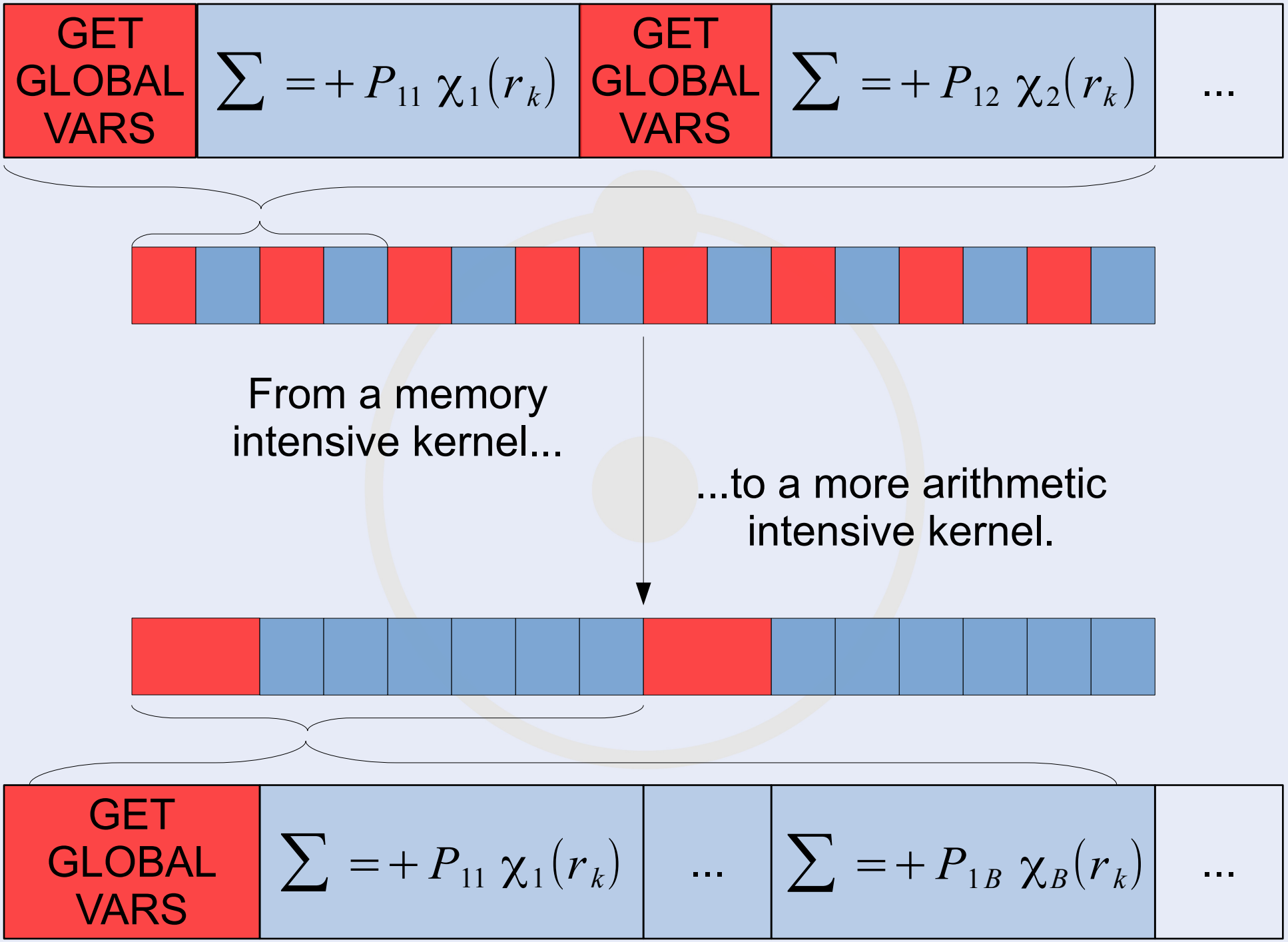
$$\sum_{j=1}^{B} P_{Bj}\,\chi_j(r_k) + \sum_{j=B+1}^{2B} P_{Bj}\,\chi_j(r_k) + \ldots + \sum_{j=M-B+1}^{M} P_{Bj}\,\chi_j(r_k)$$

$$\begin{Bmatrix} P_{11} & P_{12} & \cdots & P_{1B} \\ \cdots & \cdots & \cdots & \cdots \\ P_{B1} & P_{B2} & \cdots & P_{BB} \end{Bmatrix}$$

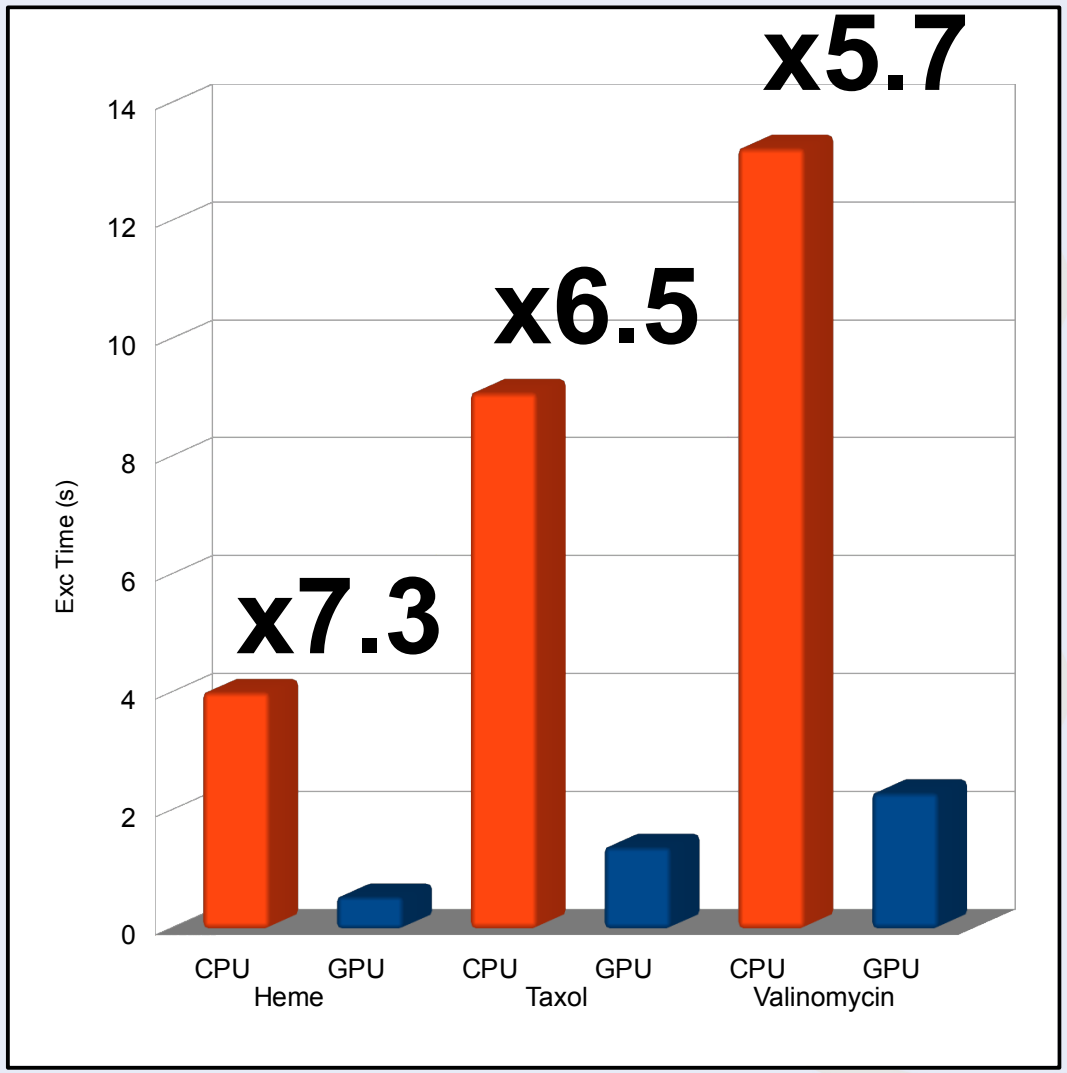
TEXTURE MEMORY
• 2D access • Local cache



From a memory intensive kernel...


GET GLOBAL VARS
$\sum = + P_{11}\ \chi_1(r_k)$
GET GLOBAL VARS
$\sum = + P_{12}\ \chi_2(r_k)$
...
From a memory intensive kernel...
...to a more arithmetic intensive kernel.
GET GLOBAL VARS
$\sum = + P_{11}\ \chi_1(r_k)$
...
$\sum = + P_{1B}\ \chi_B(r_k)$
...

(Estimated time for the 4 core CPU)



HARDWARE

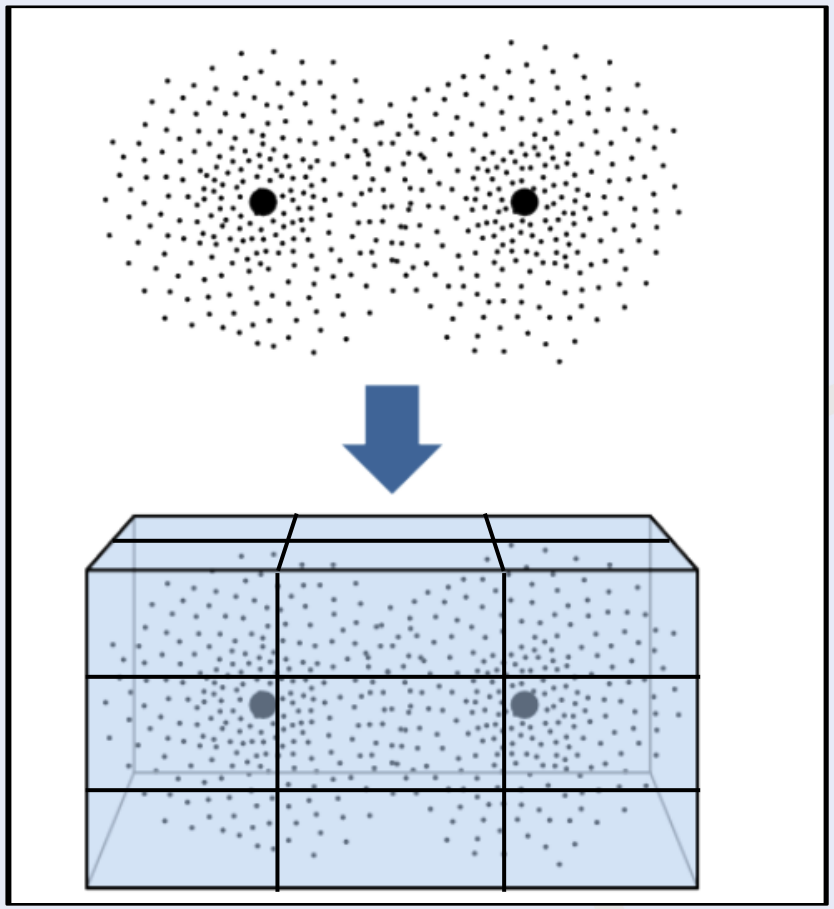
GPU: GeForce GTX 780

CPU: Intel i5-3330

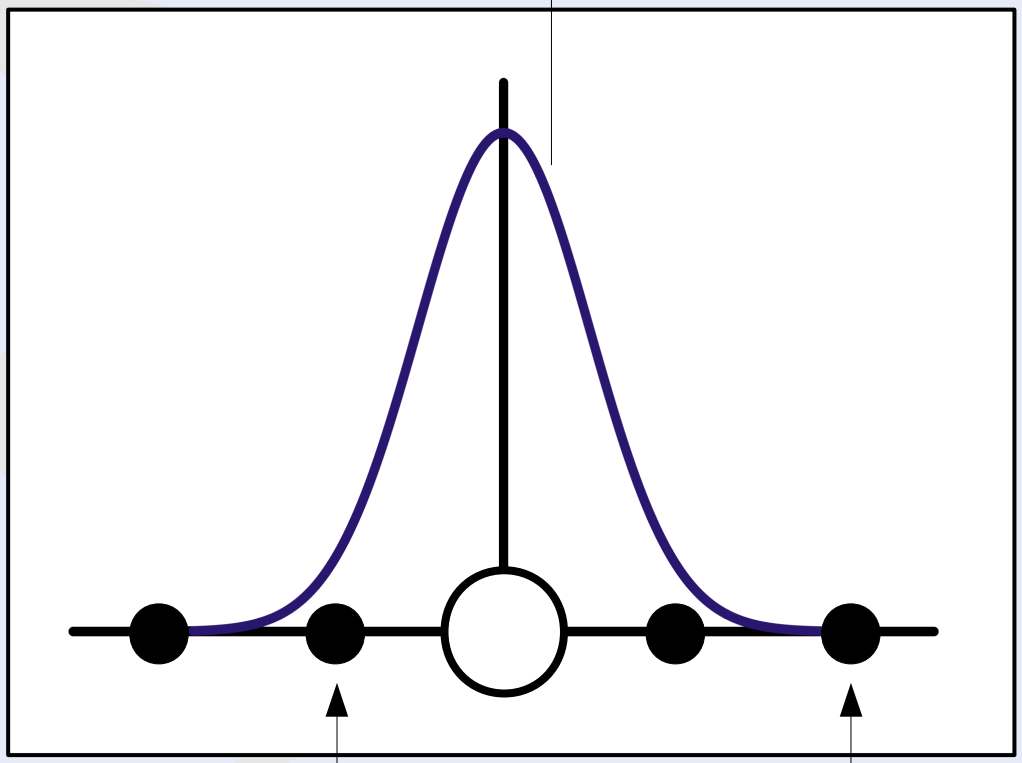
Test Systems	Speedup (1 core)
Heme	**x 29**
Taxol	**x 26**
Valinomycin	**x 23**

(no openMP)

Details: M. A. Nitsche, M. Ferreria, E. E. Mocskos, M. C. Gonzalez Lebrero, J. Chem Theory and Comput. 2014, 10, 959–967
DOI: 10.1021/ct400308n

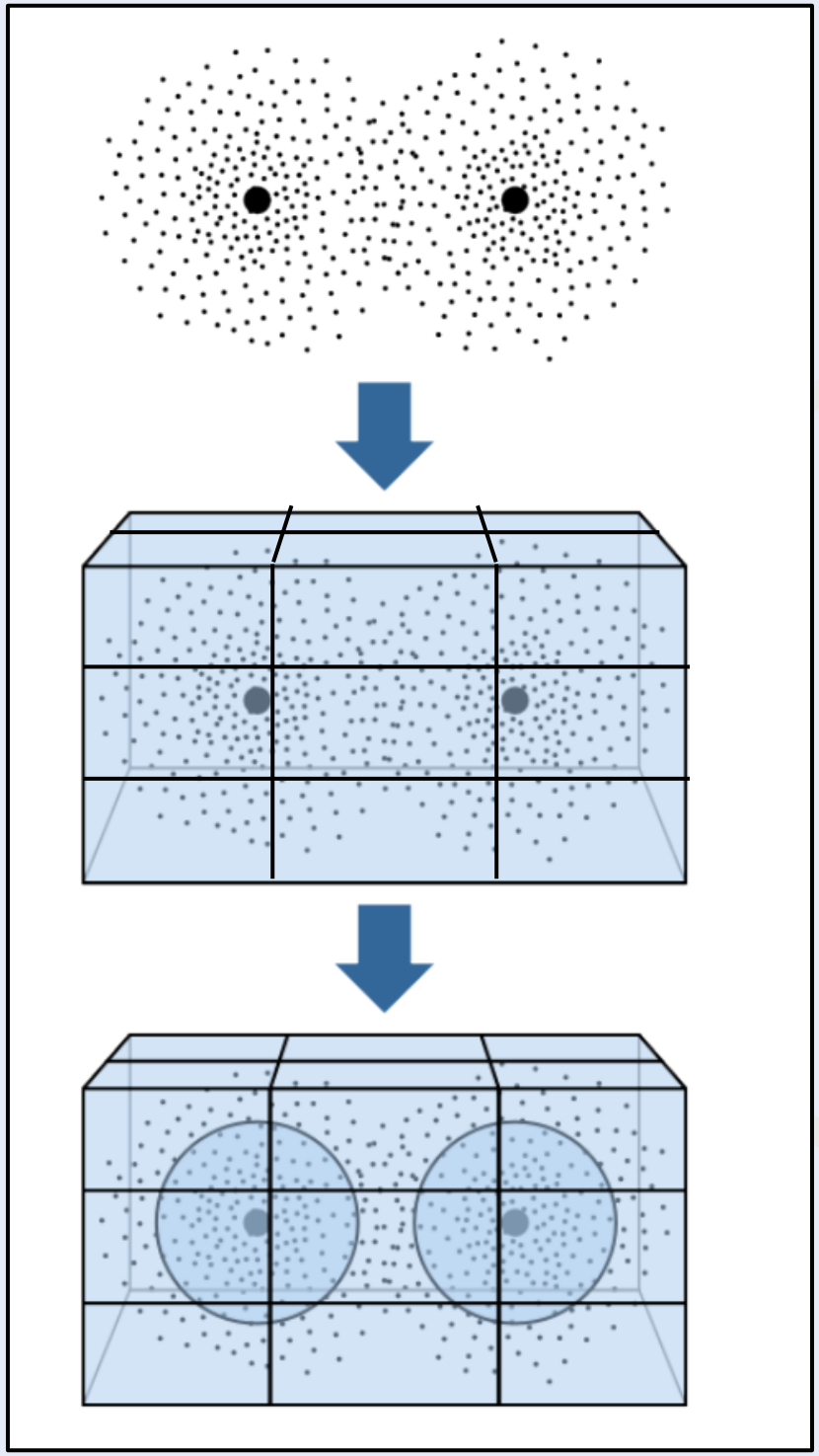


Gaussian function
centered in atom



Function value
is **relevant** for
this point

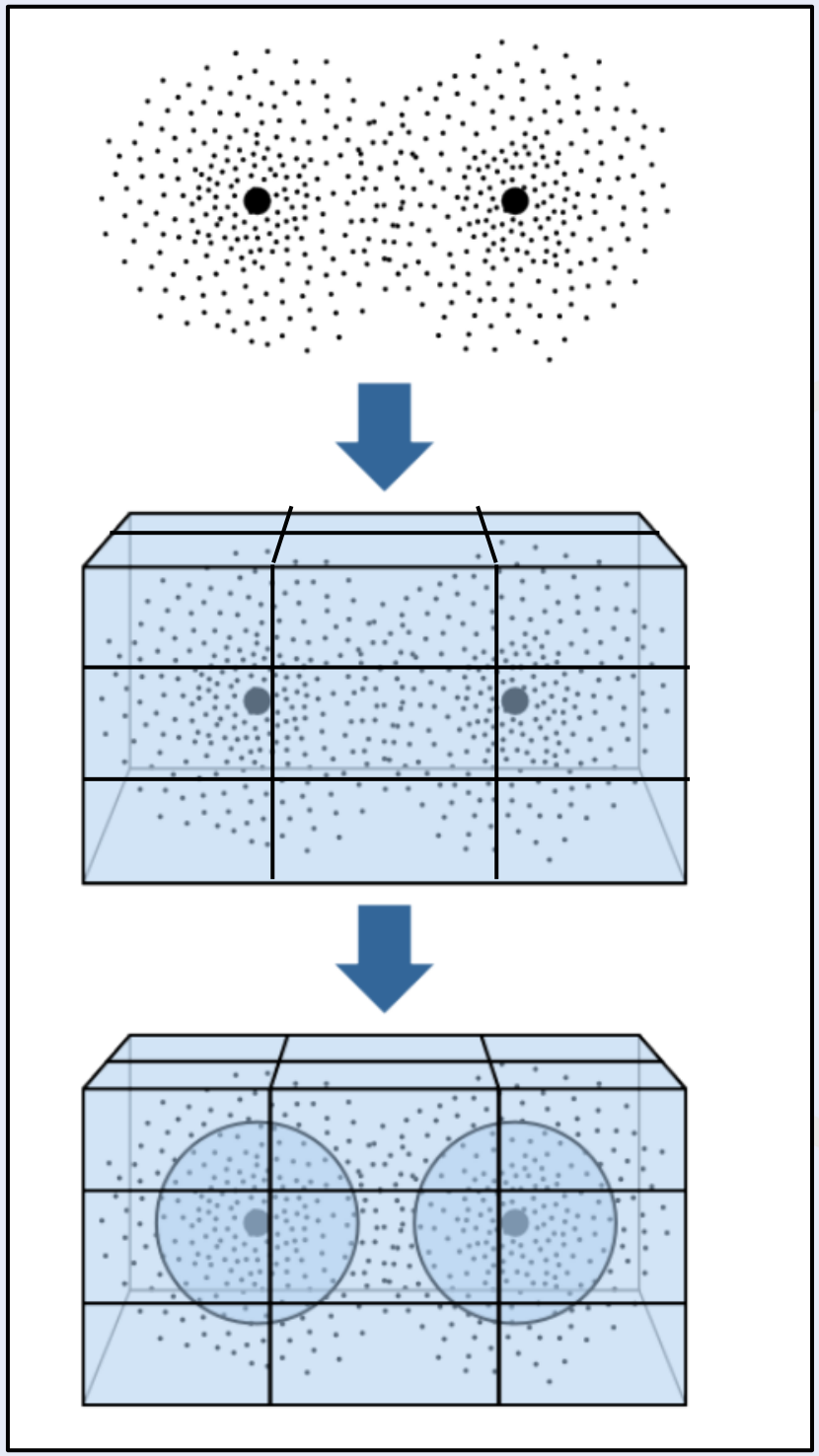
Function value
is **not relevant**
for this point



GROUPS OF POINTS

- **Groups solved separately:** reduced version of problem.
- **Inhomogeneous workload:** few groups with a lot of points/functions VS a lot of groups with very few.
- “Small” Group: bad for GPU.

Hybrid Partitioning: Small spheres (lots of points) and big cubes (fewer points)



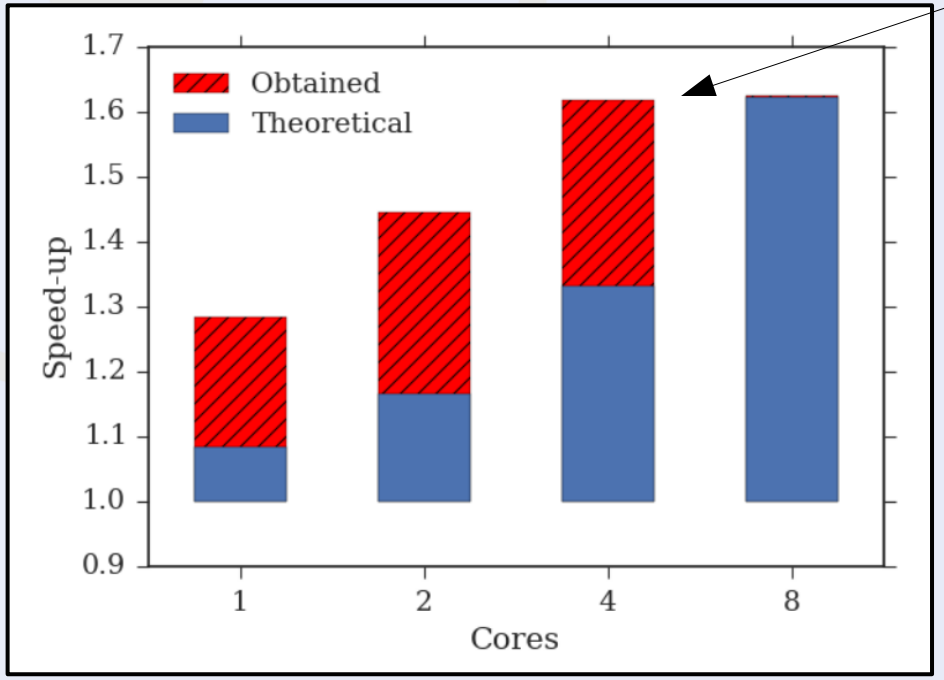
HYBRID SCHEME

Compute the “smallest” groups in the CPU.

Hardware:

- Tesla K80 (1 device)
- N x Intel Xeon E5-2698 v3

synergistic behavior



RESULTS FOR THE HYBRID

XC TERM KERNEL

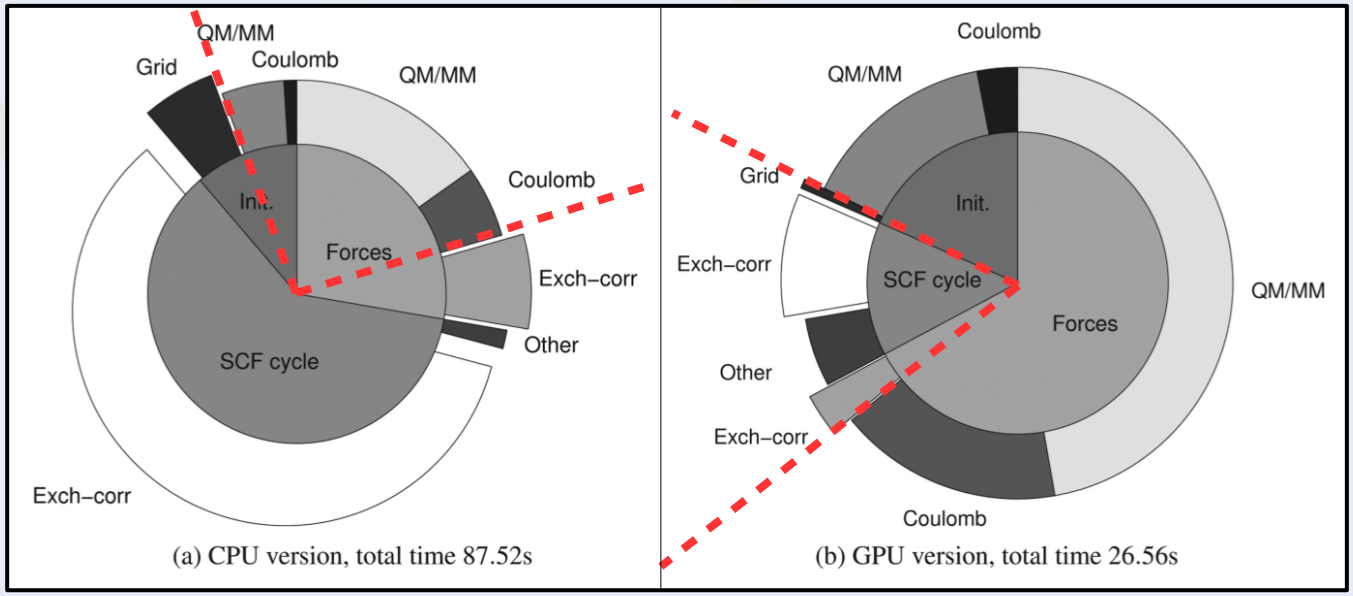
HYBRID COMPUTING SCHEME

Hardware	Speedup
GeForce GTX 980 **vs** Intel i5-3450	**x 8.0**
Tesla K80 **vs** Xeon E5-2698	**x 1.9**
Hybrid (K80+8c) **vs** Xeon E5-2698	**x 2.8**

Limitations

Coulomb

QM/MM

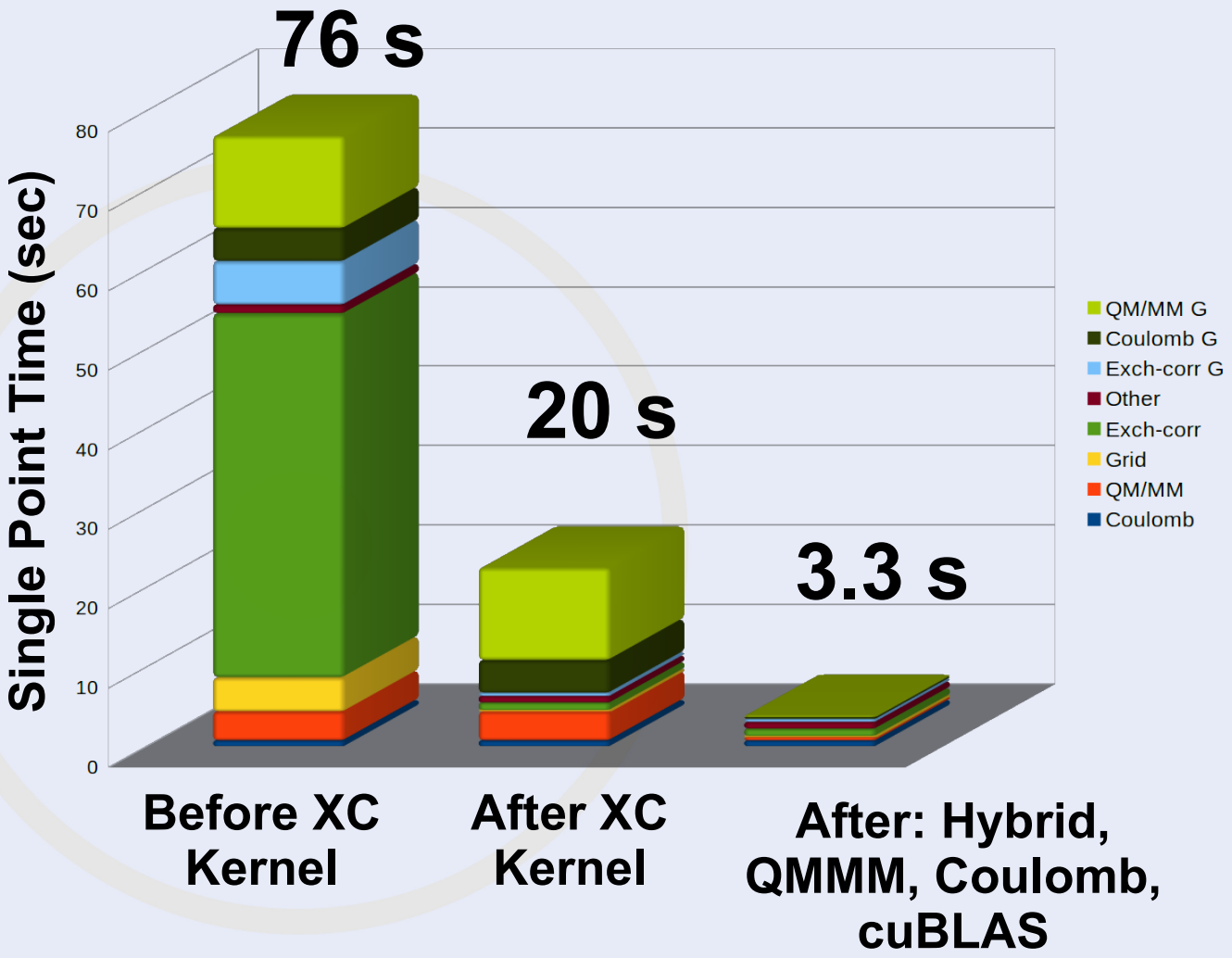


CURRENT STATE

XC TERM KERNEL

HYBRID COMPUTING SCHEME

QM/MM Coulomb cuBLAS



Current Limitations

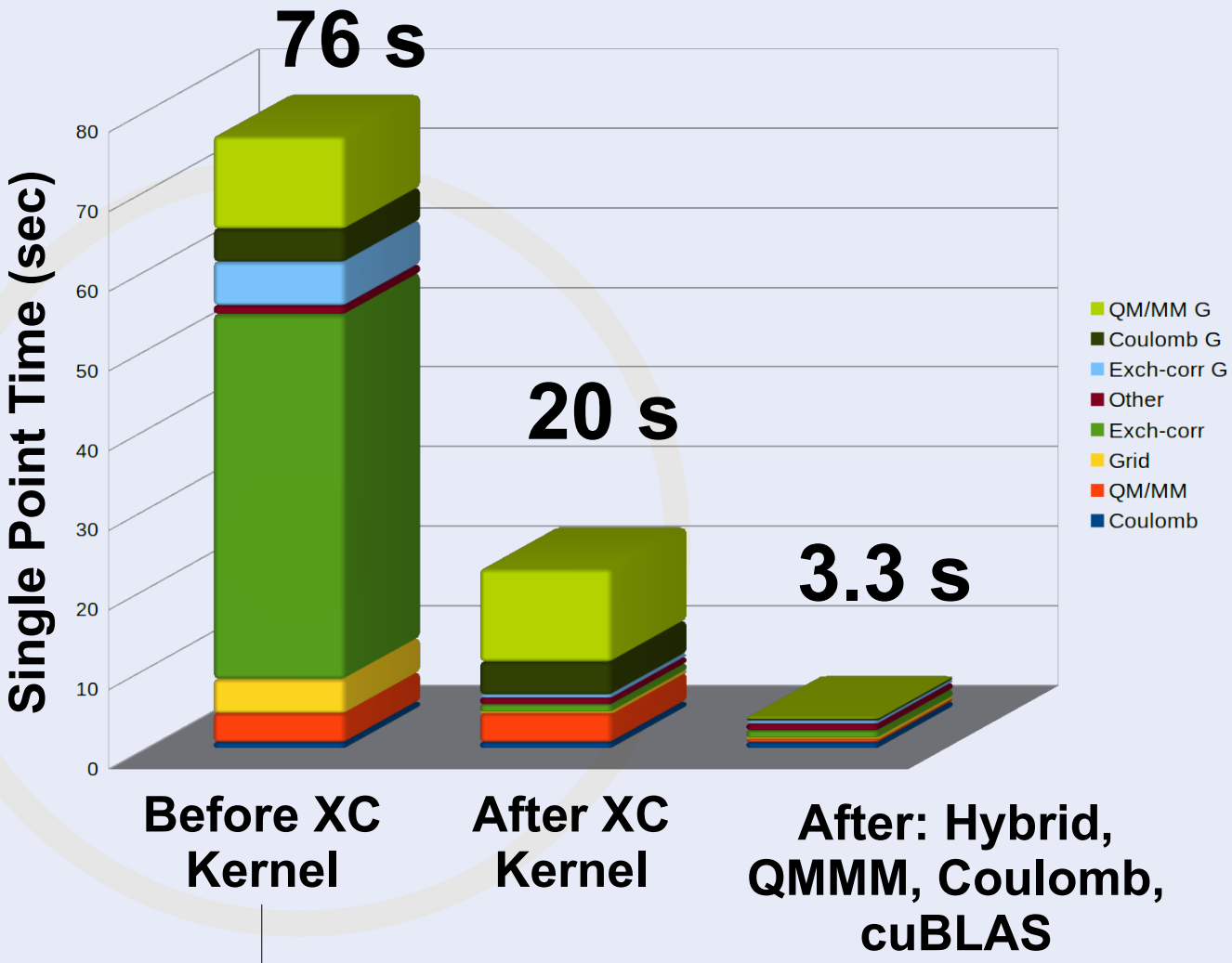
- Exc (again!)
- Matrix Diag/Dcmp?

CURRENT STATE

XC TERM KERNEL

HYBRID COMPUTING SCHEME

QM/MM Coulomb cuBLAS



Free Energy Calculation

800 – 200 days

30 days

Current Limitations

- Exc (again!)
- Matrix Diag/Dcmp?

PROBLEM

- Math expression specific to the physical model.
- Very inhomogeneous workload.
- Memory intensive calculations.

Direct manipulation of registers and memory influences algorithmic structure.

Multiple blocks of threads can run in the same execution unit

Re-organizing work to make it more homogeneous.

Distribution of work that takes into account the different capabilities of CPUs / GPUs

Customized solution with performance comparable to those of problems more naturally suited for GPU.

Thank you for your attention!

And thanks to...

Everyone at the group of computer simulation of the DQIAyQF, FCEN, UBA.

Collaborators:

- Jorge Kohanoff
- Cristián G. Sánchez
- Ivan Girotto

Institutional Support:

- CONICET
- UBA
- ICTP

THE LIO GROUP

- **Mariano González Lebrero**
- Dario Estrin
- Damian Scherlis
- Esteban Mosckos
- Uriel Morzan
- William Agudelo
- Nicolás Foglia
- Fernando Boubetta
- Federico Pedron
- Manuel Ferreria
- Matías Nitsche
- Juan Pablo Darago
- Chad Hopkins

visit our code! **github.com/nanolebrero/lio**