

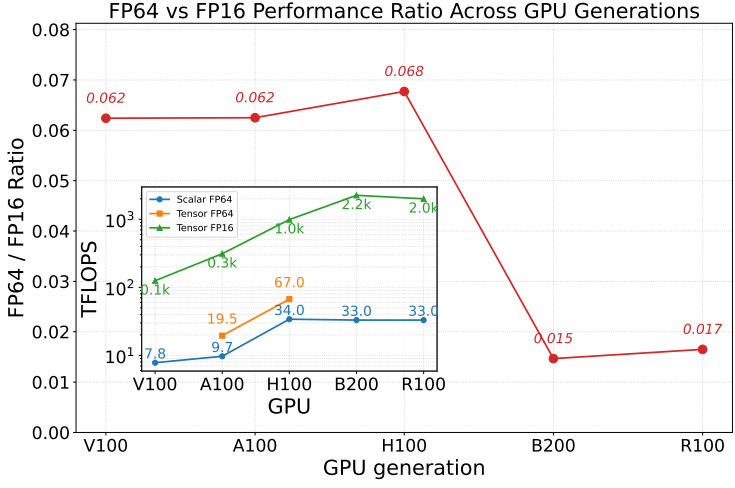
SPEEDING UP N-BODY SIMULATION
Using Low-Precision Data Types

1 September 2026 | Rodrigo Bartolomeu; René Halver; Jan Meinke and Godehard Sutmann | JSC

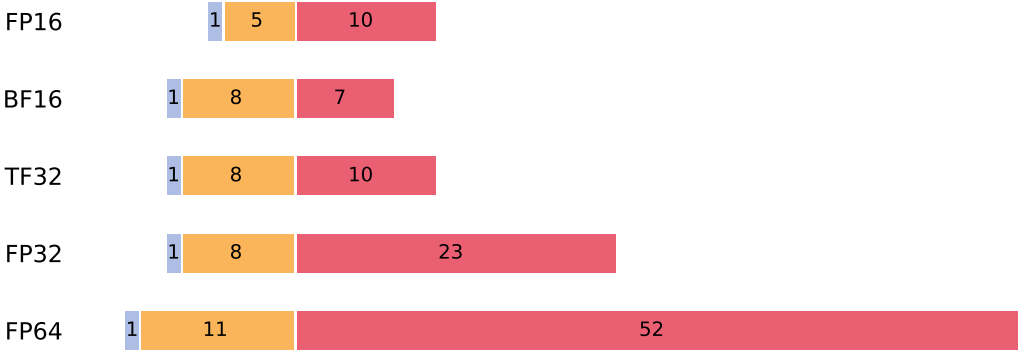


Part I: Motivation

Shift to Low Precision



Data types and bits

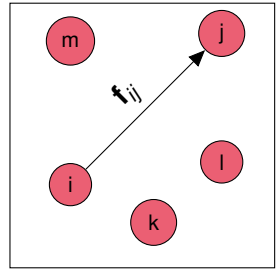


The N-Body Problem

System of particles in which particles interact pair-wise with each other with a given potential, e.g. electro-statics or gravity

$$\mathbf{f}_{ij} = -\nabla U(\mathbf{r}_{ij}) = -G\frac{m_i m_j}{|\mathbf{r}_{ij}|^3}\mathbf{r}_{ij}.$$

- All force interactions need to be computed
- Computation has $\mathcal{O}(N^2)$, w.r.t. number of particles in system
- # of interactions and arithmetic operations are constant
- With masking no branching in inner loops required





Part II: Distance Calculations With Matrix-Multiply Accumulate

Overview

Used techniques:

coordinate scaling [MLH+24]

use a cell structure to scale the coordinates to a smaller span of values

distance computation expansion [GG22]

use binomial formula to rephrase the distance computation

Overview

Used techniques:

coordinate scaling [MLH+24]

use a cell structure to scale the coordinates to a smaller span of values
$\rightarrow$allows the use of smaller data types

distance computation expansion [GG22]

use binomial formula to rephrase the distance computation

Overview

Used techniques:

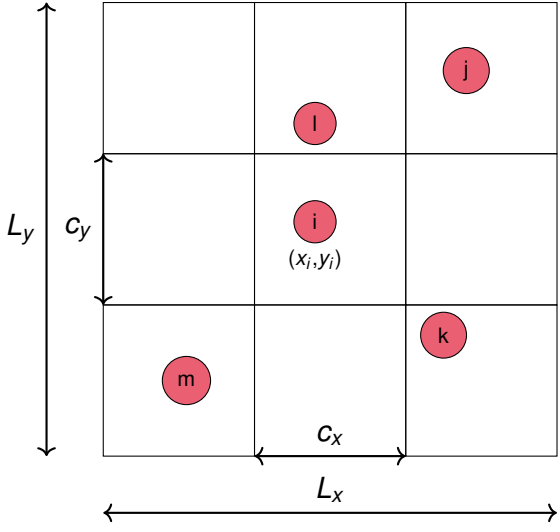
coordinate scaling [MLH+24]

use a cell structure to scale the coordinates to a smaller span of values
$\rightarrow$allows the use of smaller data types

distance computation expansion [GG22]

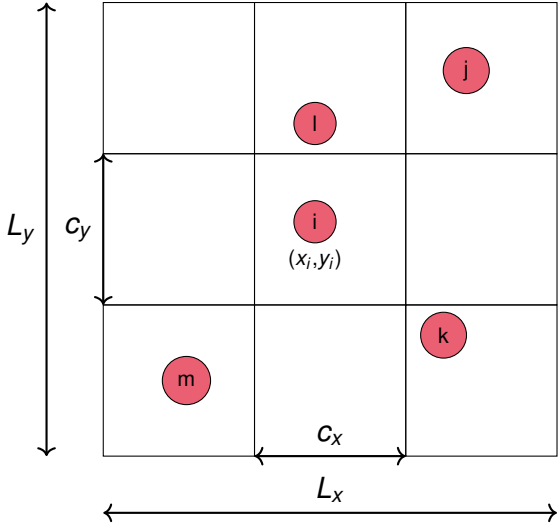
use binomial formula to rephrase the distance computation
$\rightarrow$reformulation of distance computation into matrix-matrix computation

Scaling



Let L_x and L_y be the system dimensions in x-/y- direction, $\vec{r_i} = (x_i,y_i)$, $\forall(x_i, y_i) : 0 \leq x_i \leq L_x, 0 \leq y_i \leq L_y$ be the coordinates of particle i and N_x and N_y be the cell number in that direction.

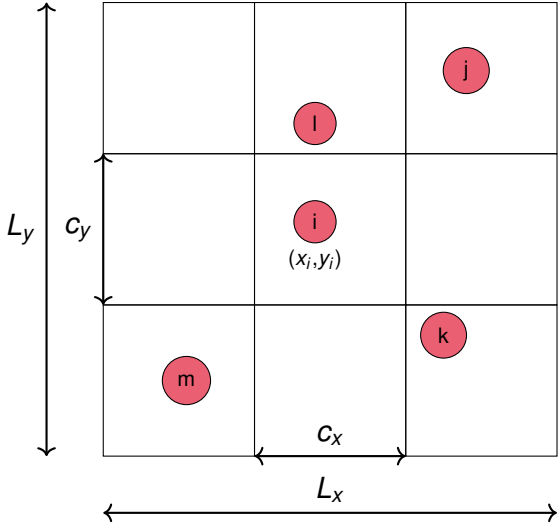
Scaling



Let L_x and L_y be the system dimensions in x-/y- direction, $\vec{r_i} = (x_i,y_i)$, $\forall(x_i, y_i) : 0 \leq x_i \leq L_x, 0 \leq y_i \leq L_y$ be the coordinates of particle i and N_x and N_y be the cell number in that direction.

Then let $c_* = \frac{L_*}{N_*}$ be the cell lengths in the corresponding directions.

Scaling

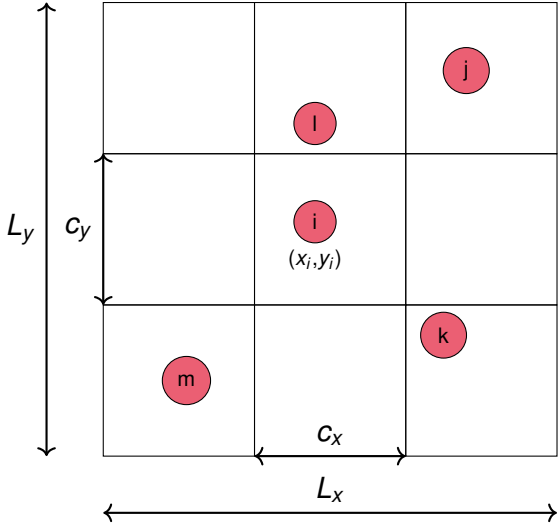


Let L_x and L_y be the system dimensions in x-/y- direction, $\vec{r_i} = (x_i,y_i)$, $\forall(x_i, y_i) : 0 \leq x_i \leq L_x, 0 \leq y_i \leq L_y$ be the coordinates of particle i and N_x and N_y be the cell number in that direction.

Then let $c_* = \frac{L_*}{N_*}$ be the cell lengths in the corresponding directions.

Using these cell lengths, one can compute scaled coordinates $\hat{\vec{r}}_i = (\hat{x}_i, \hat{y}_i) = (\frac{x_i}{c_x}, \frac{y_i}{c_y})$, so that $\forall(\hat{x}_i, \hat{y}_i) : 0 \leq \hat{x}_i \leq N_x, 0 \leq \hat{y}_i \leq N_y$

Scaling



Let L_x and L_y be the system dimensions in x-/y- direction, $\vec{r_i} = (x_i,y_i)$, $\forall (x_i, y_i) : 0 \leq x_i \leq L_x, 0 \leq y_i \leq L_y$ be the coordinates of particle i and N_x and N_y be the cell number in that direction.

Then let $c_* = \frac{L_*}{N_*}$ be the cell lengths in the corresponding directions.

Using these cell lengths, one can compute scaled coordinates $\hat{\vec{r}}_i = (\hat{x}_i, \hat{y}_i) = (\frac{x_i}{c_x}, \frac{y_i}{c_y})$, so that $\forall (\hat{x}_i, \hat{y}_i) : 0 \leq \hat{x}_i \leq N_x, 0 \leq \hat{y}_i \leq N_y$

N_* is a parameter to set the range of the scaled coordinates in case of resolution problems

Expansion

$$
\begin{aligned}
d_{ij}^2 &= (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \\
&= |\mathbf{r}_i|^2 + |\mathbf{r}_j|^2 - 2\,(\mathbf{r}_i \mathbf{r}_j) \\
&= G_{ii} + G_{jj} - 2\,G_{ij}
\end{aligned}
$$

with $G_{ij} = (\mathbf{r_i}\,\mathbf{r_j})$ as scalar product of position vectors of particles.

Expansion

$$
\begin{aligned}
d_{ij}^2 &= (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \\
&= |\mathbf{r}_i|^2 + |\mathbf{r}_j|^2 - 2\,(\mathbf{r}_i \mathbf{r}_j) \\
&= G_{ii} + G_{jj} - 2\,G_{ij}
\end{aligned}
$$

with $G_{ij} = (\mathbf{r_i}\,\mathbf{r_j})$ as scalar product of position vectors of particles.
Note that $|\mathbf{r}_i|^2$ can be calculated once in $\mathcal{O}(N)$ in time and storage and used for all distance pairs i, j.

Matrix Formulation

Let Q be the coordinate matrix of the reference points

$$Q = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ & \dots & \\ x_n & y_n & z_n \end{pmatrix}$$

Matrix Formulation

Let Q be the coordinate matrix of the reference points and P the coordinate matrix of some probe points

$$Q = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ & \dots & \\ x_n & y_n & z_n \end{pmatrix} \text{ and } P = \begin{pmatrix} x_A & y_A & z_A \\ x_B & y_B & z_B \\ & \dots & \\ x_m & y_m & z_m \end{pmatrix},$$

Matrix Formulation

Let Q be the coordinate matrix of the reference points and P the coordinate matrix of some probe points

$$Q = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ & \dots & \\ x_n & y_n & z_n \end{pmatrix} \text{ and } P = \begin{pmatrix} x_A & y_A & z_A \\ x_B & y_B & z_B \\ & \dots & \\ x_m & y_m & z_m \end{pmatrix},$$

then each element of

$$R = QP^{\mathsf{T}}$$

contains the scalar product G_{ij} of a reference point i and a probe point j.

Matrix Formulation

Let Q be the coordinate matrix of the reference points and P the coordinate matrix of some probe points

$$Q = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ & \dots & \\ x_n & y_n & z_n \end{pmatrix} \text{ and } P = \begin{pmatrix} x_A & y_A & z_A \\ x_B & y_B & z_B \\ & \dots & \\ x_m & y_m & z_m \end{pmatrix},$$

then each element of

$$R = QP^{\mathsf{T}}$$

contains the scalar product G_{ij} of a reference point i and a probe point j.
For example,

$$G_{1A} = x_1 x_A + y_1 y_A + z_1 z_A$$

Matrix Formulation

Let Q be the coordinate matrix of the reference points and P the coordinate matrix of some probe points

$$Q = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ & \cdots & \\ x_n & y_n & z_n \end{pmatrix} \text{ and } P = \begin{pmatrix} x_A & y_A & z_A \\ x_B & y_B & z_B \\ & \cdots & \\ x_m & y_m & z_m \end{pmatrix},$$

then each element of

$$R = QP^{\mathsf{T}}$$

contains the scalar product G_{ij} of a reference point i and a probe point j.
For example,

$$G_{1A} = x_1 x_A + y_1 y_A + z_1 z_A$$

and

$$d_{1A}^2 = G_{11} + G_{AA} - 2G_{1A}$$

Combining strategies

$$
\begin{aligned}
r_{ij} &= \sqrt{\sum_d \left(\tilde{x}_{i,d} - \tilde{x}_{j,d}\right)^2} \\
&= \sqrt{\sum_d \left[\left(S^*_{i,d} - S^*_{j,d}\right) h_c\right]^2} \\
&= \sqrt{\sum_d \left(S^*_{i,d} - S^*_{j,d}\right)^2} h_c \\
&= \hat{d}_{ij}\, h_c
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{f_{ij}} &= -\frac{G m_i m_j}{r_{ij}^3}\, \mathbf{r}_{ij} \\
&= -\frac{G m_i m_j}{\hat{d}_{ij}^3\, h_c^3}\, \hat{\mathbf{d}}_{\mathbf{ij}} h_c \\
&= -\frac{G}{h_c^2} \frac{m_i m_j}{\hat{d}_{ij}^3}\, \hat{\mathbf{d}}_{\mathbf{ij}} \\
&= -\frac{G}{h_c^2}\, \hat{\mathbf{f}}_{ij},
\end{aligned}
$$

JÜLICH Forschungszentrum



Part III: Verification and Benchmarks

Implementations

Uniform precision (**U**)

- Only worked for Double Precision

Separate Accumulator (**SA**)

- Uses more registers
- Smaller memory footprint
- Although promising, only Double Precision succeeded

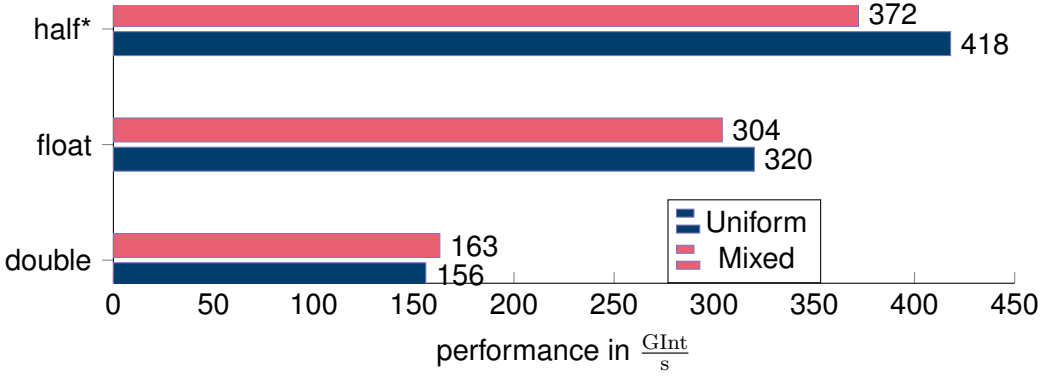
Rescaling Distances (**RD**)

- Allows for accumulation in Floating Precision

Table: Overview for implementations distance calculations on each architecture. “−” indicates that the datatype combination is not supported by the implementation using the data type combinations.

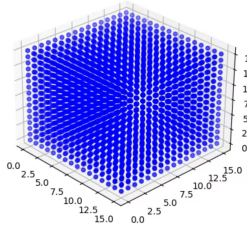
Datatype	DPDP	SPDP	HPDP	SPSP	HPSP	HPHP
U	✓	-	-	✗	-	✗
SA	✓	✗	-	-	✗	-
RD	✓	✓	✓	✓	✓	-

Performance



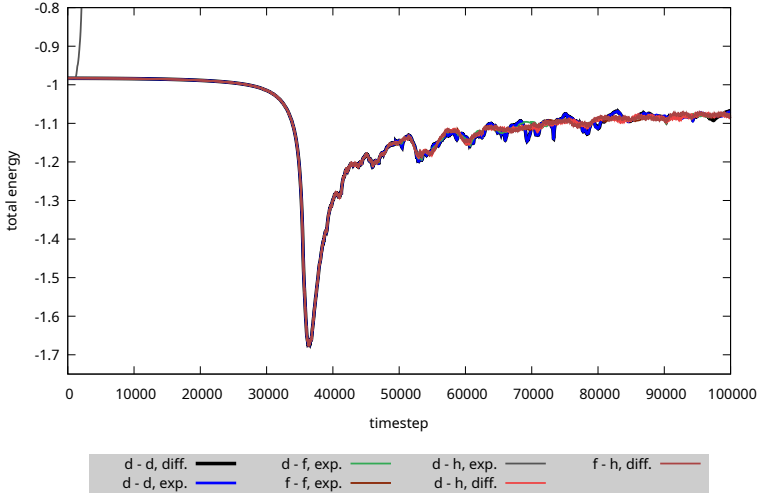
* indicates that results were physically incorrect.

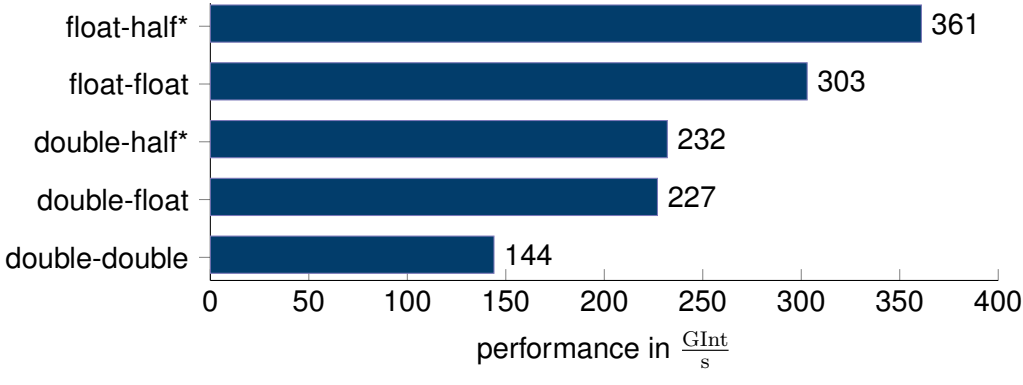
Energy conservation



- Total energy should remain constant
- Collapse introduces numerical instability
- Double precision simulation is used as reference
- Fluctuations arise from post-collapse pulsing behaviour

Distance calculation using square expansion





* indicates that results were physically incorrect.



Part IV: Conclusions and outlook

Summary of Main Findings

- **Performance:**
  Mixed arithmetic yields significant speed-ups for the N-Body problem (up to 50% on modern GPUs).
- **Precision management:**
  Accurate results demand careful handling of value ranges (e.g. scaling).
- **Mixed-precision limits:**
  For large systems, even the usual (FP64/FP32) mix can break down in long-range force calculations, jeopardising integration stability.

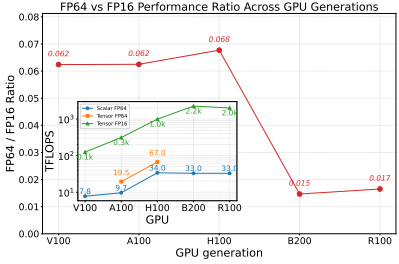
Algorithmic Contributions

Contribution	What Was Done	Impact
Coordinate scaling	Introduced a global scaling factor for particle positions before distance evaluation	Enables stable simulations across a wide distance.
Distance-computation approach	Compared direct vs. expansion-based distance calculations in FP16	Scaling improves expansion-method stability.
“Onion-style” scaling (proposed)	Concept of region-specific scaling (inner/outer shells)	Potential to raise accuracy while retaining performance.

4. Outlook & Future Work

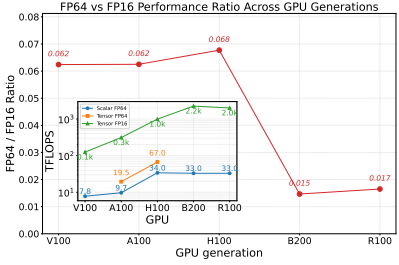
1. **Multi-level scaling**
   - Design several scaling shells.
   - Quantify trade-off: administrative cost vs. accuracy gain.
2. **Detailed numerical diagnostics**
   - Error-propagation analysis for FP16 expansion formulas.
   - Test mitigation techniques (compensated summation, Kahan, mixed-precision kernels).
3. **Hardware-accelerated kernels**
   - Port the stable expansion method to tensor-cores.
   - Benchmark speed-up versus accuracy for large-scale systems.

Take home



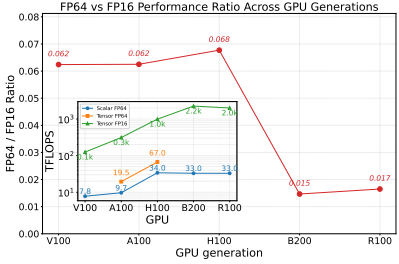
- Performance has shifted to low-precision

Take home



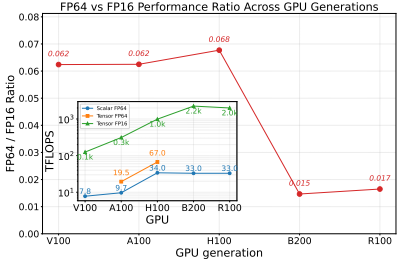
- Performance has shifted to low-precision
- New performance portability paradigm

Take home



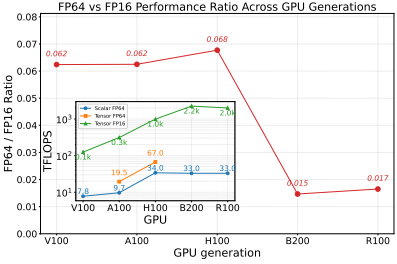
- Performance has shifted to low-precision
- New performance portability paradigm
- Time to solution $\times$ Energy efficiency

Take home



- Performance has shifted to low-precision
- New performance portability paradigm
- Time to solution $\times$ Energy efficiency
- Error correction schemes

Take home



- Performance has shifted to low-precision
- New performance portability paradigm
- Time to solution $\times$ Energy efficiency
- Error correction schemes

Bundesministerium für Forschung, Technologie und Raumfahrt

Multiscale

EuroHPC Joint Undertaking

Bibliography



Benoit Gallet and Michael Gowanlock.
Leveraging GPU Tensor Cores for Double Precision Euclidean Distance Calculations .

In *2022 IEEE 29th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, pages 135–144, Los Alamitos, CA, USA, December 2022. IEEE Computer Society.



Zirui Mao, Xinyi Li, Shenyang Hu, Ganesh Gopalakrishnan, and Ang Li.
A gpu accelerated mixed-precision smoothed particle hydrodynamics framework with cell-based relative coordinates.
Engineering Analysis with Boundary Elements, 161:113–125, 2024.