APPLICATION OF CUDA TECHNOLOGY FOR CALCULATION OF GROUND STATES OF FEW-BODY NUCLEI BY FEYNMAN'S CONTINUAL INTEGRALS METHOD

M.A. Naumenko, V.V. Samarin
Joint Institute for Nuclear Research, Dubna, Russia

The work is supported by grant 15-07-07673-a of the Russian Foundation for Basic Research (RFBR)

Arkhangelsk, 28.03 – 01.04.2016
Motivation

- High interest in structure and reactions with few-body nuclei from both theoreticians and experimentalists (e.g. \(^2\text{H},\ 3\text{H},\ 3\text{He},\ 6\text{He}\ etc.)

- The Feynman's continual integrals method [1] provides a mathematically more simple possibility for calculating the energy and probability density for the ground states of \(N\)-particle systems compared to other approaches (e.g. expansion on hyperspherical harmonics [2])

- The choice of parameters is very simple (only nucleon-nucleon, nucleon-cluster, or cluster-cluster interaction potentials)

- The method allows application of modern parallel computing solutions to speed up the calculations

Two approaches to quantum mechanics

1) Schrödinger equation \[ \hat{H} \Psi = E \Psi \]

2) Feynman continual (path) integral

\[
K(q,t;q_0,0) = \int Dq(t) \exp\left\{ \frac{i}{\hbar} S[q(t')] \right\} = \left\langle q \left| \exp\left( -\frac{i}{\hbar} \hat{H}t \right) \right| q_0 \right\rangle
\]

is a propagator – the amplitude of the probability of propagation of the particle of mass \( m \) from the point \( q_0 \) to the point \( q \) in time \( t \), \( S[q(t)] \) and \( \hat{H} \) are action and the Hamiltonian of the system, respectively, \( Dq(t) \) is integration measure.

Euclidean time \( t = -i\tau \)

For time-independent potential energy the transition to the imaginary (Euclidean) time \( t = -i\tau \) gives the propagator

\[
K_E(q,\tau;q_0,0) = \int D_E q(\tau) \exp\left\{ -\frac{1}{\hbar} S_E[q(\tau')] \right\}
\]

Calculation of energy and wave function

\[
K_E(q,\tau;q,0) = \sum_n |\Psi_n(q)|^2 \exp\left( -\frac{E_n \tau}{\hbar} \right) + \int_{E_{\text{cont}}}^{\infty} |\Psi_E(q)|^2 \exp\left( -\frac{E \tau}{\hbar} \right) g(E) dE.
\]

Here \( g(E) \) is the density of states in the continuous spectrum \( E \geq E_{\text{cont}} \). For a system with a discrete spectrum and finite motion of particles the square of the wave function of the ground state may also be found in the limit \( \tau \to \infty \) [10] together with the energy \( E_0 \)

\[
K_E(q,\tau;q,0) \to |\Psi_0(q)|^2 \exp\left( -\frac{E_0 \tau}{\hbar} \right), \tau \to \infty.
\]

Feynman’s continual integral may be represented as the limit of the multiple integral

\[ K_\Theta(q, \tau, q_0, 0) = \lim_{N \to \infty} \int \cdots \int \exp \left\{-\frac{1}{\hbar} \sum_{k=1}^{N} \left[ \frac{m(q_k - q_{k-1})^2}{2\Delta\tau} - V(q_k) \Delta\tau \right] \right\} C^{N} dq_1 dq_2 \cdots dq_{N-1}, \]

where

\[ q_k = q(\tau_k), \ \tau_k = k\Delta\tau, \ k = 0, N, \ q_N = q, \ C = \left( \frac{m}{2\pi\hbar\Delta\tau} \right)^{1/2}. \]

\[ K_\Theta(q, \tau, q_0, 0) \approx K^{(0)}_\Theta(q, \tau, q_0, 0) \left\{ \exp \left[ -\frac{a}{\hbar} \sum_{k=1}^{N} V(q_k) \right] \right\}_{0, N}, \]

\[ K^{(0)}_\Theta(q, \tau, q_0, 0) = \left( \frac{m}{2\pi\hbar\tau} \right)^{1/2} \exp \left[ -\frac{m(q - q_0)^2}{2\hbar\tau} \right]. \]

\((N - 1)\)-dimensional vectors \(\{q_1, \ldots, q_{N-1}\}\) have the distribution law

\[ W(q_0; q_1, \ldots, q_{N-1}; q_N) = C^{N} \exp \left[ -\frac{m}{\hbar} \sum_{k=1}^{N} \frac{(q_k - q_{k-1})^2}{2\alpha} \right]. \]


Jacobi coordinates for 2,3,4-body systems

System of 2 particles ($^2$H nucleus)

$$\vec{R} = \vec{r}_2 - \vec{r}_1,$$

$\vec{r}_1$ and $\vec{r}_2$ are the radius vectors of a proton and a neutron, respectively.

System of 3 particles two of which are identical (2 neutrons or 2 protons in $^3$H, $^3$He nuclei)

$$\vec{R} = \vec{r}_2 - \vec{r}_1, \quad \vec{r} = \vec{r}_3 - \frac{1}{2}(\vec{r}_1 + \vec{r}_2).$$

$^3$H nucleus: $\vec{r}_3$ is the radius vector of the proton, $\vec{r}_1$ and $\vec{r}_2$ are the radius vectors of neutrons.

$^3$He nucleus: $\vec{r}_3$ is the radius vector of the neutron, $\vec{r}_1$ and $\vec{r}_2$ are the radius vectors of protons.

$^6$He: $\vec{r}_3$ is the radius vector of the $\alpha$ particle, $\vec{r}_1$ and $\vec{r}_2$ are the radius vectors of neutrons.

System of 4 particles with two pairs of identical particles (2 protons and 2 neutrons in the $^4$He nucleus)

$$\vec{R}_1 = \vec{r}_2 - \vec{r}_1, \quad \vec{R}_2 = \vec{r}_4 - \vec{r}_3, \quad \vec{r} = \frac{1}{2}(\vec{r}_3 + \vec{r}_4) - \frac{1}{2}(\vec{r}_1 + \vec{r}_2),$$

$\vec{r}_1$ and $\vec{r}_2$ are the radius vectors of protons, $\vec{r}_3$ and $\vec{r}_4$ are the radius vectors of neutrons.
Nucleon-nucleon interaction potentials

\[ V_{n-n}(r) \equiv V_{p-p}(r) = \sum_{k=1}^{3} u_k \exp\left(-r^2/b_k^2\right) \]

\[ V_{n-p}(r) = \eta V_{n-n}(r) \]

Parameter values: \( u_1 = 500 \text{ MeV}, \ u_2 = -102 \text{ MeV}, \ u_3 = 2 \text{ MeV}, \ b_1 = 0.59 \text{ fm}, \ b_2 = 1.40 \text{ fm}, \ b_3 = 2.94 \text{ fm} \) and \( \eta = 1.2 \text{ MeV} \) were determined by the value of the binding energy of the deuteron, the absence of bound states of two identical nucleons, as well as the value of the binding energy of nuclei \(^3\text{He}, \(^4\text{He}.\)
Hardware and software used

- Tesla K40 installed at Heterogeneous Cluster of LIT, JINR ([http://hybrilit.jinr.ru/](http://hybrilit.jinr.ru/))
  Scientific Linux, CUDA version 7.5
  Code compiled for architecture 3.5, single precision
  *For calculation*

- GeForce 9800 GT
  Windows 7
  *For debugging and testing*

- Intel® Core™ i5-3470 Processor (6M Cache, up to 3.60 GHz)
  Windows 7
  *For comparison*

- C++ language
- cuRAND random number generator
### Characteristics of a HybriLIT cluster

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of compute nodes</td>
<td>7</td>
</tr>
<tr>
<td>The total amount of RAM</td>
<td>896 Gb</td>
</tr>
<tr>
<td>The total amount of disk space</td>
<td>57.6 Tb</td>
</tr>
</tbody>
</table>

### Number of cores available for computations

<table>
<thead>
<tr>
<th>Blade hostname</th>
<th>CPU</th>
<th>GPU</th>
<th>PHI</th>
</tr>
</thead>
<tbody>
<tr>
<td>blade01</td>
<td>24</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>blade02</td>
<td>24</td>
<td>2688</td>
<td>60</td>
</tr>
<tr>
<td>blade03</td>
<td>24</td>
<td>-</td>
<td>122</td>
</tr>
<tr>
<td>blade04</td>
<td>24</td>
<td>8640</td>
<td>-</td>
</tr>
<tr>
<td>blade05</td>
<td>24</td>
<td>8640</td>
<td>-</td>
</tr>
<tr>
<td>blade06</td>
<td>24</td>
<td>8640</td>
<td>-</td>
</tr>
<tr>
<td>blade07</td>
<td>24</td>
<td>8640</td>
<td>-</td>
</tr>
<tr>
<td>Total cores</td>
<td>168</td>
<td>37248</td>
<td>182</td>
</tr>
</tbody>
</table>

### Software

- **OS**: Scientific Linux 6.6, Kernel: 2.6.32-504
- **File system**: NFS4
- **Scheduler**: SLURM-14.11.6-3 [1]
- **Modules**: MODULES 3.2.10 [2]

### Technologies

- **MPI**: OpenMPI 1.6.5, 1.8.1; [3]
- **CUDA**: CUDA 5.5, 6.0, 7.0; [4]
- **OpenMP**: GNU 4.4.7 [5], Intel 14.0.2 [6], PGI 15.3 [7];
- **OpenCL**: Intel 14.0.2, CUDA 6.0, 7.0.

### Compilers

- **C/C++, Fortran**:
  - GNU: gcc, g++, gfortran;
  - PGI: pgcc, pgc++, pgf77, pgf90;
  - Intel: icc, icpc, ifort, mpiicc, mpiicpc, mpiifort;
  - CUDA: nvcc;
  - OpenMPI: mpicc, mpicxx, mpir77, mpi90;

### Hardware

- Each node 2 x Intel Xeon E5-2695 v2 (2.40 GHz, 12 cores)
- 1 node NVIDIA Tesla K20X
- 2 nodes 3 x NVIDIA Tesla K40
- 2 nodes 2 x NVIDIA Tesla K80
- 1 node Intel Xeon Phi 5110P
- 1 nodes 2 x Intel Xeon Phi 7120P
## Tesla K40 Module - PRODUCT SPECIFICATIONS

<table>
<thead>
<tr>
<th>Category</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUDA PARALLEL PROCESSING CORES</td>
<td>2880</td>
</tr>
<tr>
<td>FRAME BUFFER MEMORY</td>
<td>12 GB GDDR5</td>
</tr>
<tr>
<td>PEAK DOUBLE PRECISION FLOATING POINT PERFORMANCE</td>
<td>1.43 Tflops</td>
</tr>
<tr>
<td>PEAK SINGLE PRECISION FLOATING POINT PERFORMANCE</td>
<td>4.29 Tflops</td>
</tr>
<tr>
<td>INTERFACE</td>
<td>384-bit</td>
</tr>
<tr>
<td>MEMORY BANDWIDTH</td>
<td>288 GB/s</td>
</tr>
<tr>
<td>DISPLAY CONNECTORS</td>
<td>None</td>
</tr>
<tr>
<td>MAX POWER CONSUMPTION</td>
<td>235 W</td>
</tr>
<tr>
<td>PROCESSOR CORE CLOCK</td>
<td>745 MHz</td>
</tr>
<tr>
<td>POWER CONNECTORS</td>
<td>1 x 6-pin PCI Express power connectors</td>
</tr>
<tr>
<td>GRAPHICS BUS</td>
<td>PCI Express 3.0 x16</td>
</tr>
<tr>
<td>FORM FACTOR</td>
<td>110 mm (H) x 265 mm (L) - Dual Slot, Full-Height</td>
</tr>
<tr>
<td>THERMAL SOLUTION</td>
<td>Passive</td>
</tr>
</tbody>
</table>
Scheme of calculation of ground state energy for one-dimensional case
Scheme of calculation of ground state wave function for one-dimensional case

Benefits: 1) accumulation of data points; 2) low memory consumption (no grid)
Results of calculation of ground state energy

The angular coefficient of resulting straight lines equals the binding energy

\[
\frac{1}{b_0} \ln \tilde{K}_E(q, \tau; q, 0) \rightarrow \frac{1}{b_0} \ln |\Psi_0(q)|^2 - E_0 \tilde{\tau}, \quad \tilde{\tau} \rightarrow \infty
\]

<table>
<thead>
<tr>
<th>Statistics, n</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^5$</td>
<td>○</td>
</tr>
<tr>
<td>$10^6$</td>
<td>●</td>
</tr>
<tr>
<td>$5 \cdot 10^6$</td>
<td>●</td>
</tr>
<tr>
<td>$10^7$</td>
<td>●</td>
</tr>
</tbody>
</table>
Results of calculation the energy of ground state

The angular coefficient of resulting straight lines equals the binding energy

\[
\frac{1}{b_0} \ln \tilde{K}_E (q, \tau; q, 0) \rightarrow \frac{1}{b_0} \ln |\Psi_0(q)|^2 - E_0 \bar{\tau}, \quad \bar{\tau} \rightarrow \infty
\]
Results of calculation of ground state energy

The straight line parameters were obtained for the straight parts of curves and data points with statistics $10^6$, $5 \cdot 10^6$, $10^7$.
Comparison of the theoretical and experimental energy of ground state

<table>
<thead>
<tr>
<th>Atomic nucleus</th>
<th>Theoretical value, MeV</th>
<th>Experimental value, MeV</th>
</tr>
</thead>
<tbody>
<tr>
<td>$^2$H</td>
<td>1.17 1</td>
<td>2.225</td>
</tr>
<tr>
<td>$^3$H</td>
<td>9.29 1</td>
<td>8.482</td>
</tr>
<tr>
<td>$^3$He</td>
<td>6.86 1</td>
<td>7.718</td>
</tr>
<tr>
<td>$^4$He</td>
<td>26.95 1</td>
<td>28.296</td>
</tr>
</tbody>
</table>

Experimental data was taken from the NRV Knowledge Base http://nrv.jinr.ru/nrv/

Reasonable agreement with experimental data without any fitting
Correctness check

Probability density for $^2\text{H}$
- Shell model (CPU)
- Feynman's continual integrals method (CUDA)

Charge distribution for $^3\text{He}$
- Experiment
- Feynman's continual integrals method (CUDA)

![Graph a) Probability density for $^2\text{H}$](image-a)

![Graph b) Charge distribution for $^3\text{He}$](image-b)

- distance between proton and neutron
- distance from the center of mass

Good agreement
Propagator for $^{3}\text{He} (p + p + n)$

$$K_{E}(q, \tau, q, 0) = \sum_{n} |\Psi_{n}(q)|^{2} \exp\left(-\frac{E_{n, \tau}}{\hbar}\right) + \int_{E_{\text{cont}}}^{\infty} |\Psi_{E}(q)|^{2} \exp\left(-\frac{E_{\tau}}{\hbar}\right) g(E) dE$$

$$K_{E}(X, Y, \theta)$$

Logarithmic scale
Probability density $|\Psi|^2$ for ground state of $^3$He (p + p + n)

$$K_E(q, \tau, q, 0) \approx |\Psi_0(q)|^2 \exp\left(-\frac{E_0\tau}{\hbar}\right)$$

$$K_E(X, Y, \theta)$$

$^3$He

$V=\infty$

Logarithmic scale
Probability density for $^3$He integrated over angle

$$\int \left| \Psi(\vec{x}, \vec{y}) \right|^2 d\Omega_x d\Omega_y$$

spatial correlational density plot

$$P(x, y) = x^2 y^2 \int \left| \Psi(\vec{x}, \vec{y}) \right|^2 d\Omega_x d\Omega_y$$

$^3$He

Charge radius: nice agreement

<table>
<thead>
<tr>
<th>Theoretical value, fm</th>
<th>Experimental value, fm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.94</td>
<td>$1.9664 \pm 0.0023$</td>
</tr>
</tbody>
</table>

Experimental data from
E.G. Nadjakov et al., Atomic Data and Nuclear Data Tables, 1994, vol.56, p.133


Both methods agree qualitatively (different systems and potentials!)
 Probability density for $^4\text{He}$

\[ |\Psi_0 (\vec{R}_1; \vec{r}; \vec{R}_2) |^2 = |\Psi_0 (R_{1x}, 0, 0, 0, r_z; 0, R_{2y} = R_{1x}, 0) |^2 \]

for symmetric tetrahedral configuration of four nucleons:

$\vec{R}_1 \perp \vec{r} \perp \vec{R}_2$, $|\vec{R}_1| = |\vec{R}_2|$, $\vec{R}_1 = (R_{1x}, 0, 0)$, $\vec{r} = (0, 0, r_z)$, $\vec{R}_2 = (0, R_{2y} = R_{1x}, 0)$

Full calculation will take $\approx 60 \cdot 12 \cdot 12 \cdot 11 = 95040$ hours (>11 years) on single Tesla K40?
### Comparison of calculation time

#### $^3\text{He}$ ground state energy

<table>
<thead>
<tr>
<th>Statistics, $n$</th>
<th>Intel Core i5 3470 (1 thread), sec</th>
<th>Tesla K40s, sec</th>
<th>Performance gain, times</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^5$</td>
<td>$\approx 1854$</td>
<td>$\approx 8$</td>
<td>$\approx 241$</td>
</tr>
<tr>
<td>$10^6$</td>
<td>$\approx 18377$</td>
<td>$\approx 47$</td>
<td>$\approx 389$</td>
</tr>
<tr>
<td>$5 \cdot 10^6$</td>
<td>$-$</td>
<td>$\approx 221$</td>
<td>$-$</td>
</tr>
<tr>
<td>$10^7$</td>
<td>$-$</td>
<td>$\approx 439$</td>
<td>$-$</td>
</tr>
</tbody>
</table>

$^3\text{He}$ wave function in 60*60*12 points

<table>
<thead>
<tr>
<th>Statistics, $n$</th>
<th>Intel Core i5 3470 (1 thread), estimation</th>
<th>Tesla K40s</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^6$</td>
<td>$\sim 177$ days</td>
<td>$\approx 11$ hours</td>
</tr>
</tbody>
</table>

The method enables calculations impossible before before.
Conclusions

• The algorithm of calculation of ground states of few-body nuclei by Feynman's continual integrals method allowing us to perform calculations directly on GPU using NVIDIA CUDA technology was developed and implemented on C++ language;

• The energy and the square modulus of the wave function of the ground states of several few-body nuclei have been calculated; the method may also be applied to the calculation of cluster nuclei;

• Correctness of the calculations was justified by comparison with
  • the wave function obtained using the shell model
  • experimental binding energies
  • experimental charge radii and charge distributions

• The results show that the use of GPGPU significantly increases the speed of calculations. This allows to
  • increase the statistics and accuracy of calculations
  • reduce the space step in the calculation of the wave functions
  • simplifies the process of debugging and testing
  • enables calculations impossible before
Thank You