Algorithm for Free Gas Elastic Scattering without Rejection Sampling

Elliott Biondo[1], Vladimir Sobes[2], Andrew Holcomb[1], Steven Hamilton[1], Thomas Evans[1]

[1]Oak Ridge National Laboratory
[2]University of Tennessee

October 5, 2021

Introduction

GPU computing

- Leadership-class supercomputers rely on GPUs for the majority of their processing power
- GPUs, unlike CPUs, use a Single Instruction Multiple Data (SIMD) paradigm
  - A *kernel launch* involves deploying a large number of independent threads
  - A single “slow” thread can prevent a kernel launch from completing, creating a performance bottleneck
- Many Monte Carlo (MC) radiation transport algorithms must be reworked to optimize GPU execution

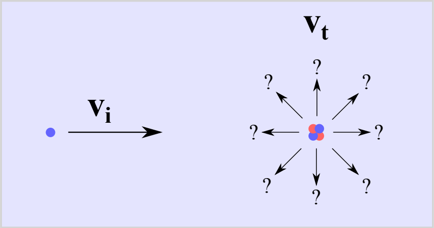
Summit (2018)
200 petaflops
95% from GPUs

↓

Frontier (2021)
$>$1500 petaflops
High % from GPUs

Free gas elastic scattering scenario

- During an MC random walk, a neutron with a velocity $\mathbf{v_i}$ undergoes a collision
- The collision type is sampled to be an elastic scatter
- How do we sample the target velocity ($\mathbf{v_t}$) of the nucleus?



- $\mathbf{v_t}$ follows a Maxwellian distribution according to the temperature of the medium
- Not all $\mathbf{v_t}$ are equally likely to cause a collision

Likely target velocities

- The relative energy (E_r) of a scattering event is given by

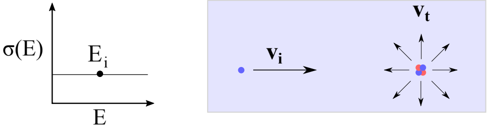
$$E_r = \frac{1}{2} m |\mathbf{v_i} - \mathbf{v_t}|^2$$

where m is the neutron mass

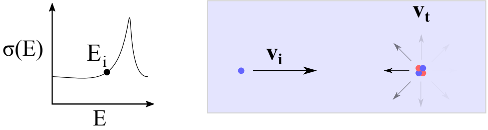
- $\mathbf{v_t}$ vectors that cause E_r to be closer to a resonance are more likely to cause scattering events

- This effect can significantly impact MC results [1]

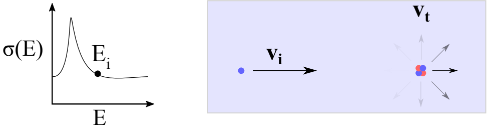
Case 1: flat cross section near E_i



Case 2: Resonance with an energy just above E_i

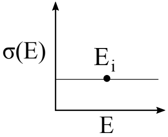


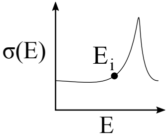
Case 3: Resonance with an energy just below E_i



Rejection sampling algorithms for free gas elastic scattering

- Doppler Broadening Rejection Correction (DBRC) [2] and Relative Velocity Sampling (RVS) [3]
- DBRC: Rejection sample a possible $\mathbf{v_t}$ from a Maxwellian, then accept/reject based on the likelihood that $\mathbf{v_t}$ causes a collision
- High sampling efficiency ($\sim$ 97% for DBRC, $\sim$ 22% for RVS) in regions in which the cross section is relatively flat [3]
- Low sampling efficiencies ($<$ 1%) near resonances [3]





- These disparate rejection sampling efficiencies degrade the GPU particle tracking rate
- **Shift MC code [1]: DBRC was found to reduce the GPU tracking rate by 4.9×, compared to only 5.3% on the CPU**

Summary

- A new method for sampling $\mathbf{v_t}$ without rejection sampling:
  1. Step 1: Sample E_r using windowed multipole data
  2. Step 2: Sample $\mathbf{v_t}$ based on E_r
- Validation
- Preliminary performance results
- Conclusion

Step 1: Sample E_r

Relative energy PDF and CDF

PDF of the E_r of the collision [4], in terms of $u = \sqrt{E}$ and $\xi \propto \sqrt{T}$:

$$f(u_r) = \left(e^{\frac{-(u_i - u_r)^2}{\xi^2}} - e^{\frac{-(u_i + u_r)^2}{\xi^2}} \right) u_r^2 \sigma(u_r)$$

The CDF is then:

$$F(u_r) = \frac{\int_0^{u_r} f(u_r')\,\mathrm{d}u_r'}{\int_0^{\infty} f(u_r)\,\mathrm{d}u_r} = \frac{\int_0^{u_r} \left(e^{\frac{-(u_i - u_r')^2}{\xi^2}} - e^{\frac{-(u_i + u_r')^2}{\xi^2}} \right) u_r'^2 \sigma(u_r')\,\mathrm{d}u_r'}{\xi \sqrt{\pi} u_i^2 \sigma_D(u_i, \xi)}$$

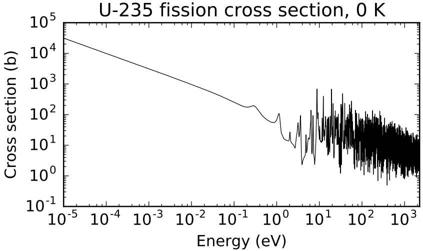
Goal: obtain an expression for the 0 K cross section (σ) that is:

1. closed form
2. allows the CDF equation to be integrated

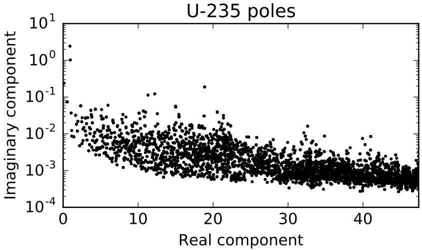
Multipole data representation

Provides an expression for the thermal/epithermal σ in terms of:

- poles (p): singularities in the complex plane
- corresponding residues (r), proportional to the path integrals around poles



→



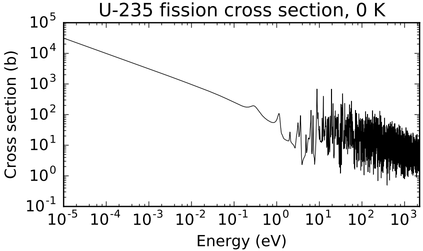
At $T = 0$ K, the cross section is given by [5]:

$$\sigma(u) = \frac{1}{u^2} \sum_k \mathbf{Re}\left(\frac{2r_k}{p_k - u}\right)$$

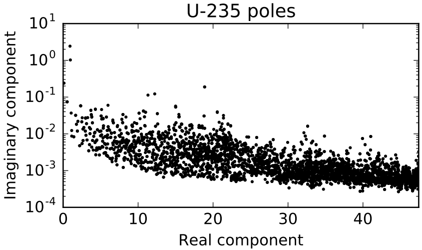
Multipole data representation

Provides an expression for the thermal/epithermal σ in terms of:

- poles (p): singularities in the complex plane
- corresponding residues (r), proportional to the path integrals around poles



→



At $T = 0$ K, the cross section is given by [5]:

$$\sigma(u) = \frac{1}{u^2} \sum_k \mathbf{Re}\left(\frac{2r_k}{p_k - u}\right)$$

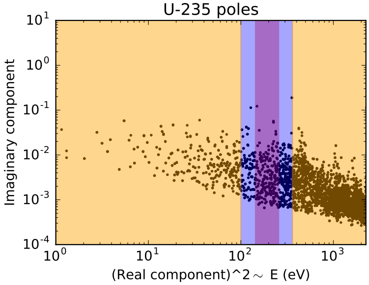
Computationally expensive

Windowed Multipole Method

Approximate "far away" poles with a polynomial [6]:

$$\sigma(u) = \underbrace{\frac{1}{u^2} \sum_{k \in \text{window}} \textbf{Re}\left(\frac{2r_k}{p_k - u}\right)}_{\text{poles}} + \underbrace{\frac{1}{u^2} \sum_{n=0}^{N-1} a_n \left(c_1 (u - c_0)\right)^n}_{\text{polynomial}}$$

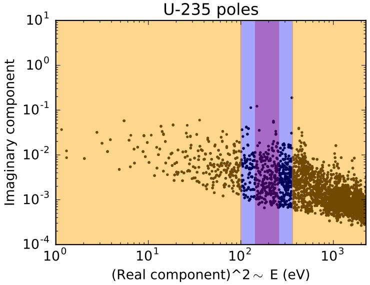
This does not provide an integrable PDF



Windowed Multipole Method

Approximate “far away” poles with a polynomial [6]:

$$\sigma(u) = \underbrace{\frac{1}{u^2}\sum_{k\in\text{window}} \textbf{Re}\left(\frac{2r_k}{p_k - u}\right)}_{\text{poles}} + \underbrace{\frac{1}{u^2}\sum_{n=0}^{N-1} a_n\left(c_1(u-c_0)\right)^n}_{\text{polynomial}}$$



This does not provide an integrable PDF

For free gas elastic scattering, use a Gaussian approximation:

$$\sigma(u) = \frac{1}{u^2}\sum_k\sum_j\left[\underbrace{h_{s,k,j}\,e^{\frac{-(u-u_k)^2}{w_{s,k,j}^2}}}_{\textit{symmetric}} + \underbrace{h_{a,k,j}(u-u_k)\,e^{\frac{-(u-u_k)^2}{w_{a,k,j}^2}}}_{\textit{antisymmetric}}\right] + \frac{1}{u^2}\sum_{n=0}^{N-1} a_n\left(c_1(u-c_0)\right)^n$$

Solved CDF

$$F(u_r) = \frac{1}{\xi\sqrt{\pi}u_i^2\sigma_D(u_i,\xi)}\left[\sum_k\sum_j h_{s,k,j}I_s(u_r) + \sum_k\sum_j h_{a,k,j}I_a(u_r) + \sum_n a_n I_{p,n}(u_r)\right]$$

Closed-form expression for $I_s(u_r)$, $I_a(u_r)$, and $I_{p,n}(u_r)$ are found in the paper.

To sample this CDF:

1. Select a random variate ε
2. Solve for u_r via root-finding:

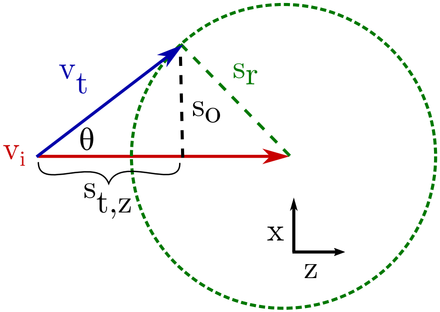
$$F(u_r) - \varepsilon = 0 \tag{1}$$

3. Final answer:

$$E_r = u_r^2 \tag{2}$$

Step 2: Sample v_t

PDF for z component of $\mathbf{v_t}$



$$f(s_{t,z}) = \underbrace{e^{\frac{-Ms_{t,z}^2}{2k_B T}}}_{1} \underbrace{s_o e^{\frac{-Ms_o^2}{2k_B T}}}_{2} \underbrace{\sqrt{1 + \left(\frac{s_{i,z} - s_{t,z}}{s_o}\right)^2}}_{3}$$

1. Maxwellian PDF of target speed in the z-direction
2. Maxwellian PDF of target speed in the orthogonal direction in cyl. coordinates
3. Ratio of the rate of change in the z-direction relative to the rate of change along the arc length of the sphere

CDF for z component of $\mathbf{v_t}$

Convert the PDF to a CDF:

$$F(s_{t,z}) = \frac{\int_{s_{i,z}-s_r}^{s_{t,z}} f(s'_{t,z})\,\mathrm{d}s'_{t,z}}{\int_{s_{i,z}-s_r}^{s_{i,z}+s_r} f(s_{t,z})\,\mathrm{d}s_{t,z}} = \frac{e^{\frac{-Ms_{i,z}s_{t,z}}{k_B T}} - e^{\frac{-Ms_{i,z}(s_{i,z}-s_r)}{k_B T}}}{e^{\frac{-Ms_{i,z}(s_{i,z}+s_r)}{k_B T}} - e^{\frac{-Ms_{i,z}(s_{i,z}-s_r)}{k_B T}}}$$

This CDF is invertible:

$$s_{t,z} = F^{-1}(\varepsilon) = \frac{-k_B T}{Ms_{i,z}} \log\left(\varepsilon e^{\frac{-Ms_{i,z}(s_{i,z}+s_r)}{k_B T}} + (1-\varepsilon) e^{\frac{-Ms_{i,z}(s_{i,z}-s_r)}{k_B T}}\right)$$

Once $s_{t,z}$ is sampled, $cos(\theta) = \frac{s_{t,z}}{s_t}$, and the azimuthal angle is sampled uniformly

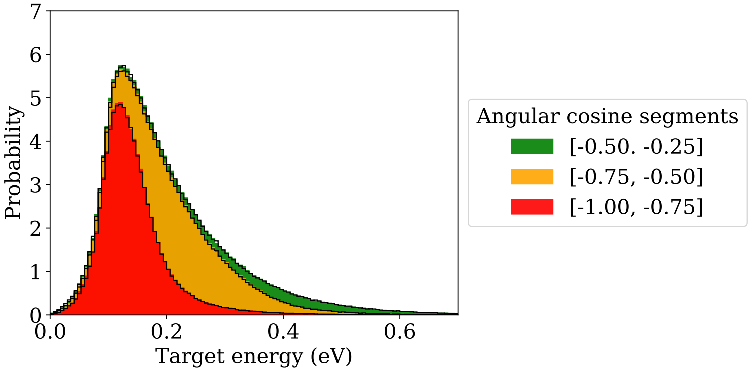
Validation

Prototype implementation

- Implemented in Python:
  - “Pole” method for free gas elastic scattering
  - Standard DBRC method
- CPU execution only

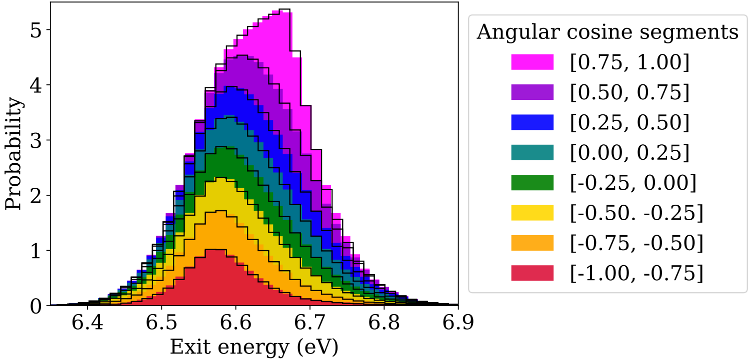
Target energy PDF comparison

- $E = 6.57$ eV, below the 6.67 eV resonance of ^{238}U, $T = 1200$ K
- Black lines: expected results (DBRC)



Neutron exit energy PDF comparison

- $E = 6.67$ eV, coincident with the 6.67 eV resonance of ^{238}U, $T = 1200$ K
- Black lines: expected results (DBRC)
- Results match those in the literature [7]

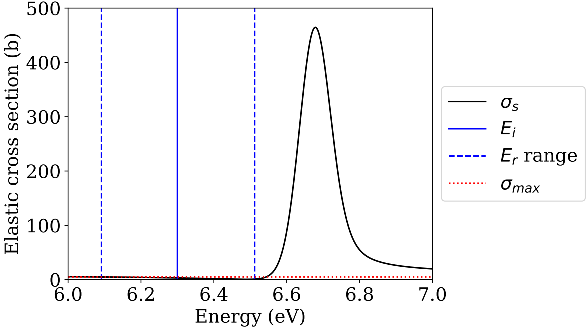


Preliminary timing results

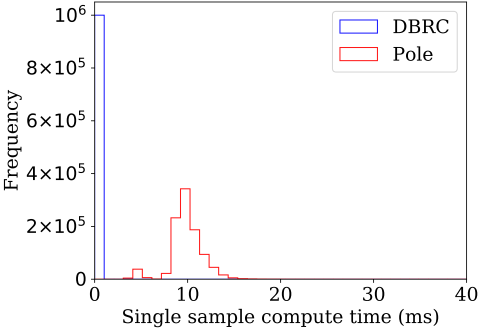
CPU timing results: far from resonance

$E_i = 6.3$ eV, far from the 6.67 eV ^{238}U resonance, $T = 300$ K

Problem setup



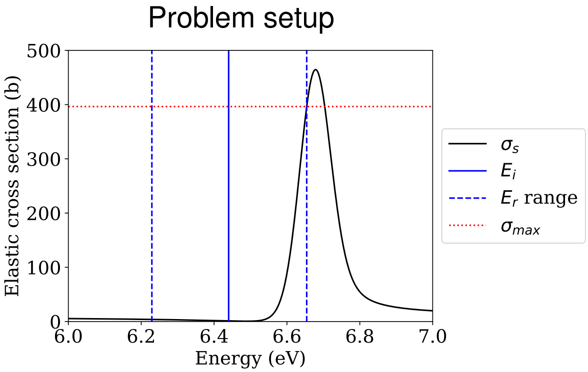
Timing results, 10^6 samples

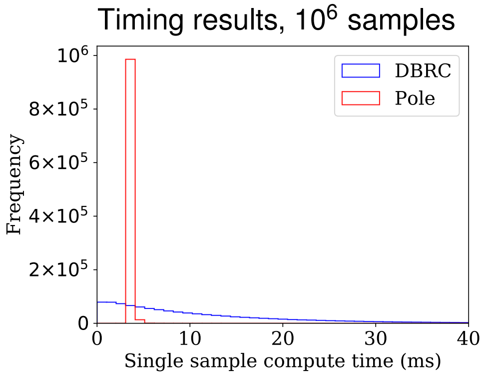


DBRC is $>$ 10$\times$ faster then pole for a flat σ

CPU timing results: near resonance

$E_i = 6.44$ eV, on the edge of the 6.67 eV ^{238}U resonance, $T = 300$ K



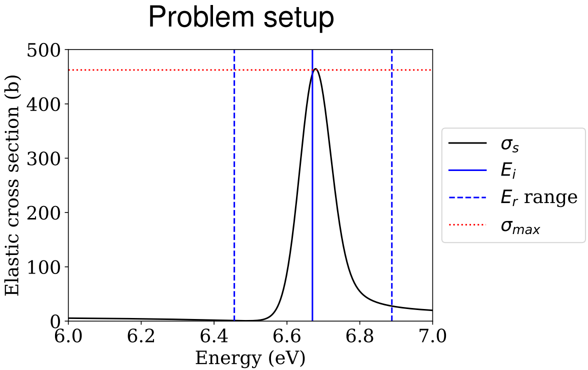


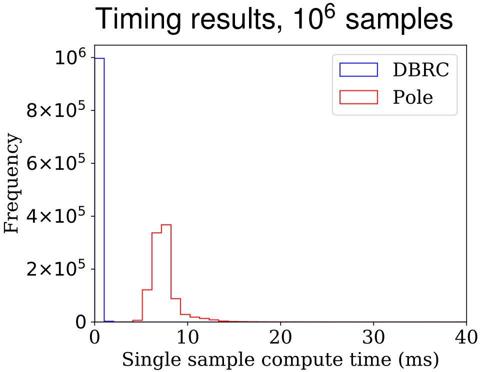
Pole is consistently low. DBRC has a long tail → poor GPU performance?

OAK RIDGE National Laboratory

CPU timing results: on resonance

$E_i = 6.67$ eV, coincident with the 6.67 eV ^{238}U resonance, $T = 300$





Same outcome as the "far from resonance" case

OAK RIDGE National Laboratory

Summary of results

- Pole: worst case single sample compute time is $\sim$ 15 ms on the CPU
- DBRC: worst case single sample compute time is $\sim$ 40 ms on the CPU
- GPU performance may be related to worst case CPU timing results

Conclusion

Conclusion

- A new free gas elastic scattering method without rejection sampling is proposed
  - Enabled by the Windowed Multipole method
  - Involves one root-finding step and one direct sampling step
- Preliminary results indicate this method may outperform existing methods on the GPU
- Future work:
  - Explore further approximations to eliminate the root-finding step
  - Implement and test on the GPU

Acknowledgments

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation's exascale computing imperative.

Questions?

References

[1] Tara M. Pandya, Seth R. Johnson, Thomas M. Evans, Gregory G. Davidson, Steven P. Hamilton, and Andrew T. Godfrey. Implementation, capabilities, and benchmarking of Shift, a massively parallel Monte Carlo radiation transport code. *Journal of Computational Physics*, 308:239–272, March 2016.

[2] B. Becker, R. Dagan, and G. Lohnert. Proof and implementation of the stochastic formula for ideal gas, energy dependent scattering kernel. *Annals of Nuclear Energy*, 36(4):470–474, 2009.

[3] Paul K. Romano and Jonathan A. Walsh. An improved target velocity sampling algorithm for free gas elastic scattering. *Annals of Nuclear Energy*, 114:318 – 324, 2018.

[4] F. H. Fröhner. Applied neutron resonance theory. Technical Report KfK-2669, Kernforschungszentrum Karlsruhe, 1978.

[5] R. N. Hwang. A rigorous pole representation of multilevel cross sections and its practical applications. *Nuclear Science and Engineering*, 96(3):192–209, 1987.

[6] C. Josey, P. Ducru, B. Forget, and K. Smith. Windowed multipole for cross section Doppler broadening. *Journal of Computational Physics*, 307:715–727, 2016.

[7] Bjorn Becker. *On the influence of the resonance scattering treatment in Monte Carlo codes on high temperature reactor characteristics*. PhD thesis, University of Stuttgart, 2010.

Backup Slides

Solution to $I_s(u_r)$ and $I_a(u_r)$

$$I_s(u_r) = \frac{e^{\lambda_3^+}\sqrt{\pi}}{2\sqrt{\lambda_1}}\left(\mathrm{erf}(\sqrt{\lambda_1}(\lambda_2^+ + u_r)) - \mathrm{erf}(\sqrt{\lambda_1}\lambda_2^+)\right)$$
$$+ \frac{e^{\lambda_3^-}\sqrt{\pi}}{2\sqrt{\lambda_1}}\left(\mathrm{erf}(\sqrt{\lambda_1}(\lambda_2^- + u_r)) - \mathrm{erf}(\sqrt{\lambda_1}\lambda_2^-)\right)$$

$$I_a(u_r) = \frac{-e^{\lambda_3^-}}{2\lambda_1}\left(e^{-\lambda_1(u_r+\lambda_2^-)^2} - e^{-\lambda_1(\lambda_2^-)^2}\right) + \frac{e^{\lambda_3^+}}{2\lambda_1}\left(e^{-\lambda_1(u_r+\lambda_2^+)^2} - e^{-\lambda_1(\lambda_2^+)^2}\right)$$
$$- \frac{e^{\lambda_3^-}\sqrt{\pi}(\lambda_2^- + u_k)}{2\sqrt{\lambda_1}}\left(\mathrm{erf}(\sqrt{\lambda_1}(\lambda_2^- + u_r)) - \mathrm{erf}(\sqrt{\lambda_1}\lambda_2^-)\right)$$
$$+ \frac{e^{\lambda_3^+}\sqrt{\pi}(\lambda_2^+ + u_k)}{2\sqrt{\lambda_1}}\left(\mathrm{erf}(\sqrt{\lambda_1}(\lambda_2^+ + u_r)) - \mathrm{erf}(\sqrt{\lambda_1}\lambda_2^+)\right)$$

λ values for $I_s(u_r)$ and $I_a(u_r)$

$$\lambda_1 = \left(\frac{1}{\xi^2} + \frac{1}{w_k^2}\right)$$

$$\lambda_2 = \frac{1}{\lambda_1}\left(\frac{\delta u_i}{\xi^2} - \frac{u_k}{w_k^2}\right)$$

$$\lambda_3 = -\frac{u_k^2}{w_k^2} - \frac{u_i^2}{\xi^2} + \frac{1}{\lambda_1}\left(\frac{\delta u_i}{\xi^2} - \frac{u_k}{w_k^2}\right)^2$$

where $\delta = \pm 1$, and λ^+ and λ^- indicate the value of δ

Solution to $I_{p,n}(u_r)$

$$I_{p,n}(u_r) = J_n(u_r, -1) - J_n(u_r, 1),$$

where J_n values are given by:

$$J_n(u_r,\delta) = \frac{c_1^2\xi^2(n-1)}{2} J_{n-2}(u_r,\delta) - c_1(c_0 + \delta u_i) J_{n-1}(u_r,\delta)$$
$$- \frac{c_1^2\xi^2}{2c_1}\left((c_1(u_r - c_0))^{n-1} e^{\frac{-(u_r+\delta u_i))^2}{\xi^2}} - (-c_0 c_1)^{n-1} e^{\frac{-(\delta u_i)^2}{\xi^2}} \right),$$

$$J_0(u_r,\delta) = \frac{\sqrt{\pi}\xi}{2}\left[\mathrm{erf}\left(\frac{\delta u_i + u_r}{\xi}\right) - \mathrm{erf}\left(\frac{\delta u_i}{\xi}\right)\right],$$

$$J_1(u_r,\delta) = \frac{-c_1\xi^2}{2}\left(e^{\frac{-(u_r+\delta u_i)^2}{\xi^2}} - e^{\frac{-(\delta u_i)^2}{\xi^2}} \right) - c_1(c_0 + \delta u_i) J_0(u_r,\delta).$$