Fast Evaluation of Helmholtz Potential on Graphics Processing Units (GPUs)

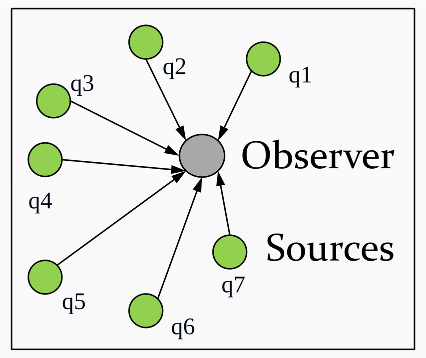
Shaojing Li*, Boris Livshitz* and Vitaliy Lomakin*

*Department of Electrical and Computer Engineering, University of California, San Diego

Problem Formulation

Interactions between N elements in Electromagnetics

$$u_j = \underbrace{\sum_{\substack{i=1 \\ i\neq j}}^{N} \frac{e^{-jk|\mathbf{r}_i-\mathbf{r}_j|}}{|\mathbf{r}_i-\mathbf{r}_j|}}_{\text{Helmholtz potential}} q_i, \quad j=1,\ldots,N \qquad \begin{array}{l} k\neq 0 - \text{dynamics} \\ k=0 - \text{statics} \end{array}$$



- ❑ High computational cost of $O(N^2)$
- ❑ Needs to be accelerated!

Fast Methods:

- ❑ Fast Fourier Transform (FFT): $O(N \log N)$ complexity
- ❑ Fast Multiple Method (FMM)[1,2]: $O(N)$ or $O(N \log N)$ complexity

Advanced Hardware and Software Techniques:

- ❑ Parallelization of fast methods is vital
- ❑ Multi-core /multi-CPU clusters have constraints
- ❑ Emerging systems: Graphics Processing Unit systems

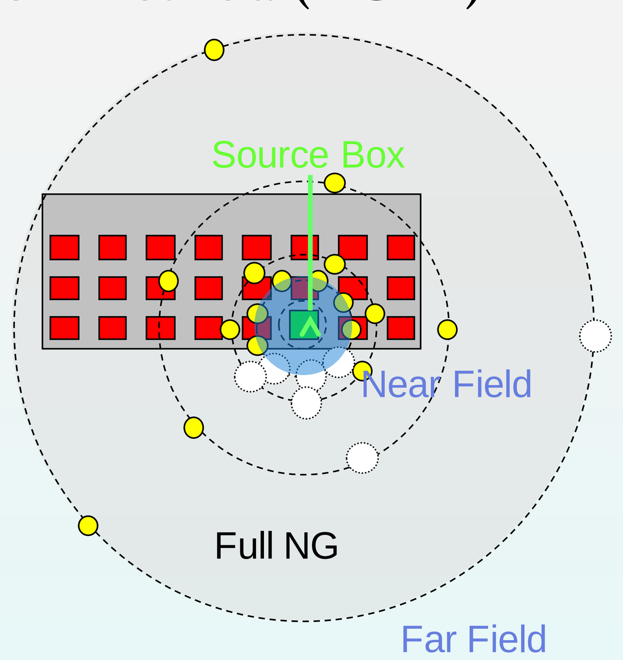
Significance

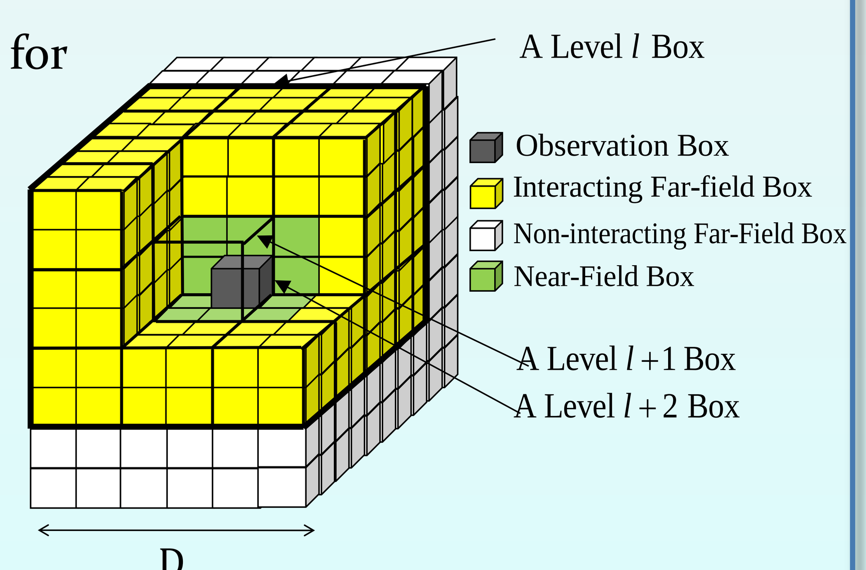
- ❑ No GPU implementations of fast Electromagnetic integral-type have been reported (implementations of static FFT and FMM exist)
- ❑ Applications in Magnetics, Microwave, Astrophysics, Acoustics, Optics, Bioengineering, Chemistry

Non-uniform Grid Interpolation Method (NGIM)[3,4,5]

Approach:

- ❑ Divide the computational domain into boxes on various levels
- ❑ Separate near and far fields calculations
- ❑ Use direct method for near-fields
- ❑ Field outside a source domain is smooth → Use sparse Non-uniform Grids (NG) to compute it
- ❑ $O(N)$ complexity is achieved for static and low-frequency fields
- ❑ Multilevel scheme is used





Advantages:

- ❑ Automatically adaptive to non-uniform source distributions
- ❑ Can handle low- and high- frequency problems → wide applicability
- ❑ No special functions required → Simplified code structure and faster
- ❑ Very fast, with GPU-CPU acceleration gains of 70-200
- ❑ Can be extended to different types of interactions (general kernels)

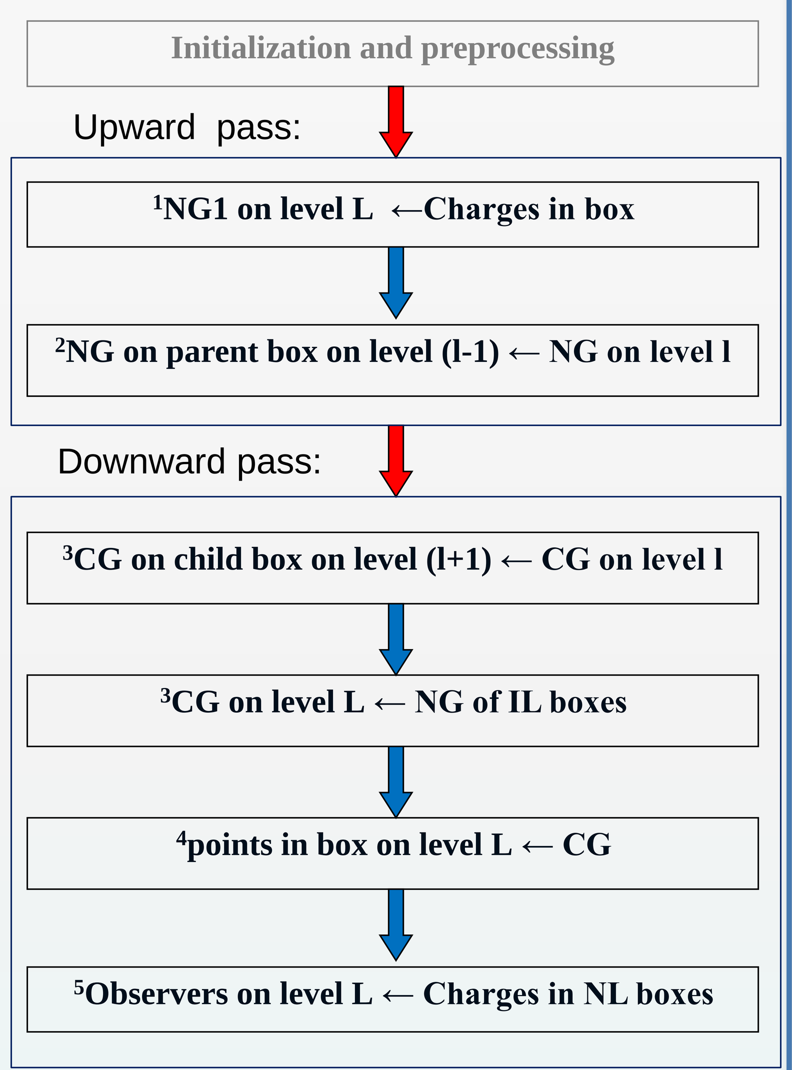
Graphic Processing Units

Implementation of the Algorithm[6,7,8]

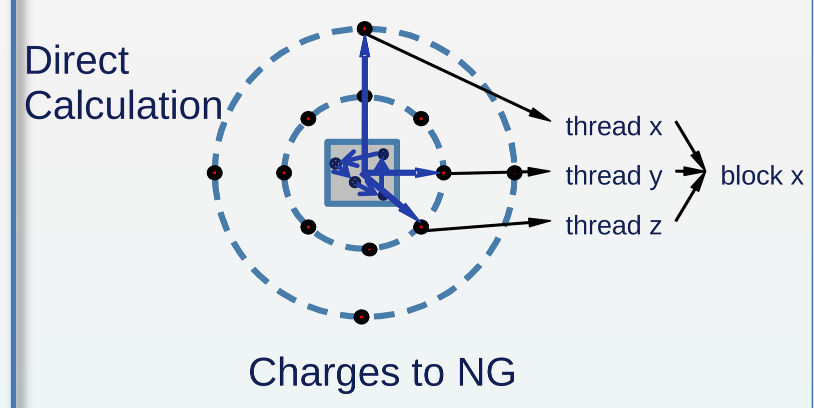
Approaches of GPU NGIM:

- ❑ One block handles one box, one thread handles one observer
- ❑ Source-observer information for a box is loaded only once
- ❑ Coalescent reading/writing

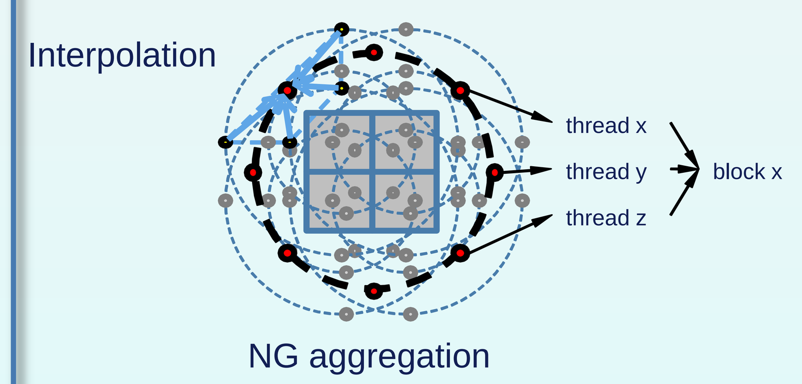
The flow chart and illustration:



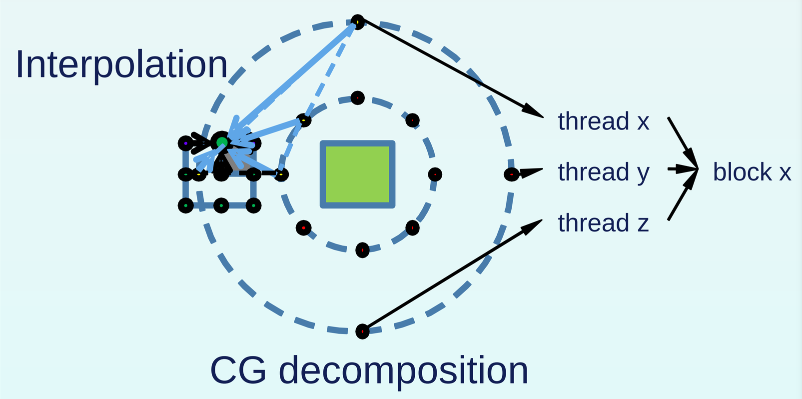
1 The Charges to NG



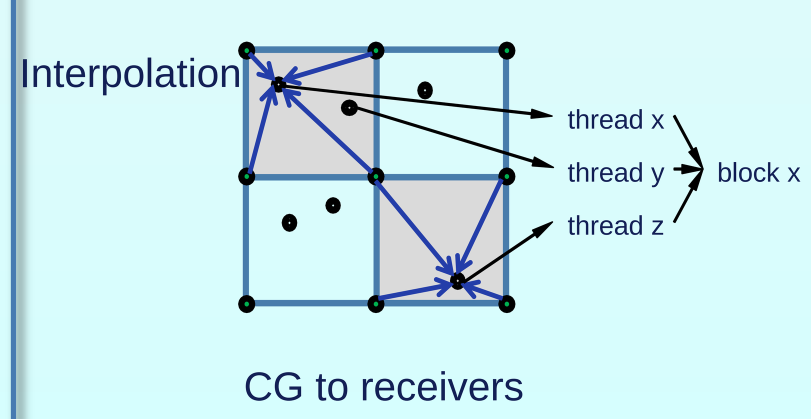
2 NG aggregation



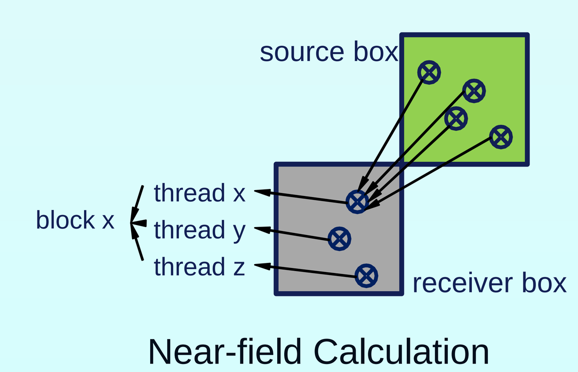
3 NG to CG transition and CG decomposition



4 CG to Receivers



5 Near Field Calculation



Results[9]

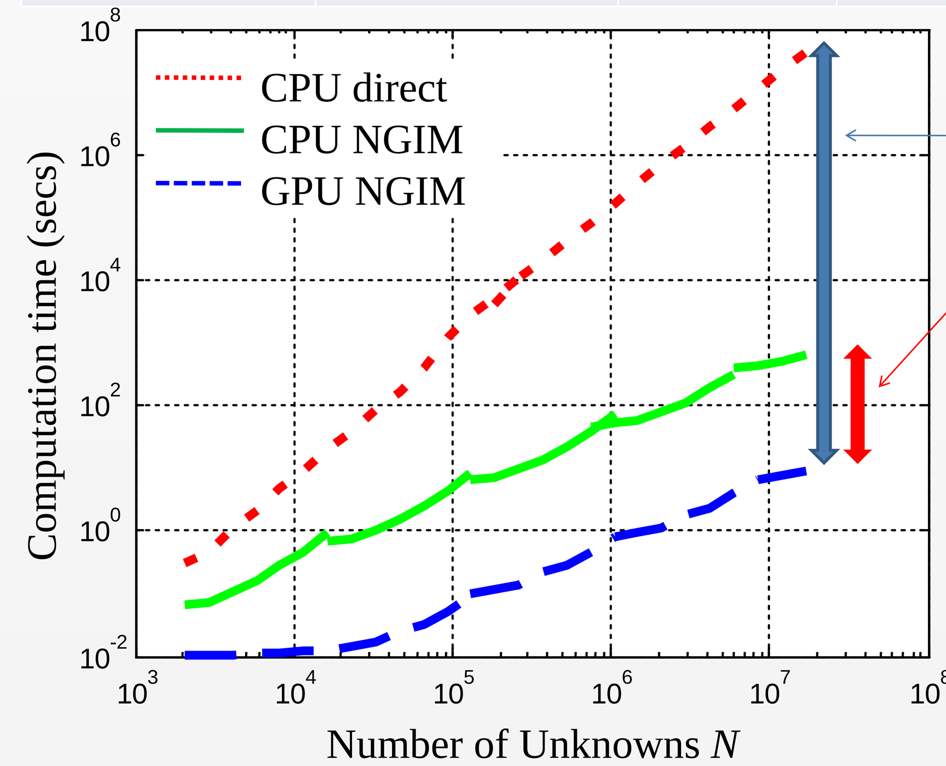
Device for comparison:
CPU: **Intel Xeon X5482@3.2GHz**
(Intel Fortran, -O3 optimization)
GPU: **NVIDIA GTX 280**
(CUDA Toolkit v2.2)

Computational Time

Low frequency. λ =100D. L2 error=1e-3

Problem Size	GPU NGIM Time (secs)	CPU NGIM Time (secs)	CPU NGIM Speedup	CPU Direct Time (secs)	CPU Direct Speedup
65,536	0.0326	2.54	77.9	394	1.21e4
1,048,576	0.815	52.9	64.9	1.69e5*	2.07e5
16,777,072	8.90	666	74.8	4.38e7*	4.9e6

*estimated numbers



4900000x speed-up!

75x – 200x speed-up! (depending on the required accuracy)

Almost the same computational time for any source distribution!

Medium frequency. λ=D. L2 error=1e-2

Problem Size	GPU NGIM Time (secs)	CPU NGIM Time (secs)	CPU NGIM Speedup	CPU Direct Time (secs)	CPU Direct Speedup
65,536	0.079	15.3	193	394	5.00e3
1,048,576	1.25	219	175	1.69e5*	1.35e5
8,388,608	10.5	1882	179	4.38e7*	4.17e6

Memory Usage Medium frequency. λ=D. L2 error=1e-2

Problem Size	GPU NGIM	CPU NGIM	Ratio
1,048,576	49.7M	3.6G	72.9
8,388,608	397M	26.2G	66.0

Future Work

- ❑ GPU implementation for the high-frequency regime (Partially done)
- ❑ Kernel independent "universal" algorithm, applicable to general kernel types (e.g. Helmholtz , Coulomb, London, Lennard-Jones, H-bonds)
- ❑ Multiple GPUs parallelization

Reference

[1] L. Greengard and V. Rokhlin, *J. Comput. Phys.*, vol. **73**, pp. 325-348, 1987.
[2] N. A. Gumerov and R. Duraiswami, *J. Comput. Phys.*, vol. **227**, pp. 8290-8313, 2008.
[3] A. Boag and E. Michielssen, A. Brandt, *IEEE Antennas and Wireless Propagation Letters,* vol. **1**, pp.142-145, 2002
[4] A. Boag and B. Livshitz, *IEEE Trans. on MTT*, vol. **54**, pp. 3365-3570, 2006.
[5] A. Boag, V. Lomakin, and E. Michielssen, *IEEE Trans. on Antennas and Propagation*, vol. **54**, no. 7, pp. 1943-1951, 2006.
[6] B. Livshitz, A. Boag, H. N. Bertram, and V. Lomakin, *J. Appl. Phys.*, vol. **105**, pp. 07D541, 2009.
[7] S. Li, B. Livshitz, and V. Lomakin, *2009 IEEE International Symposium on APS and URSI Meeting*, **IF313.1**, Charleston, SC.
[8] NVIDIA, "Compute Unified Device Architecture Programming Guide," 2008.
[9] S. Li, B. Livshitz, and V. Lomakin, "Fast evaluation of Helmholtz potential on graphics processing units (GPUs)", submitted to , *J. Appl. Phys.*, http://cem01.ucsd.edu/papers/Li_JCP2009.pdf