NONLINEAR FORCE-FREE MODELING OF CORONAL MAGNETIC FIELDS

an optimized code for Direct Boundary Integral Equation Method

Sijie Yu, Yihua Yan

National Astronomical Observatories of China, Chinese Academy of Sciences

Max Planck Institute for Solar System Research

March 10, 2014 Bern

The Boundary Integral Equation Method (BIE)

- The BIE method is proposed Yan and Sakurai (2000). The magnetic field $\mathbf{B}$ in the volume is related to a surface S integral involving the values of $\mathbf{B}$ and $\frac{\partial \mathbf{B}}{\partial n}$ and a reference function $\mathbf{Y}$ over S.

$$c_i \mathbf{B}_i = \oint_s (\overline{\mathbf{Y}} \frac{\partial \mathbf{B}}{\partial n} - \frac{\partial \overline{\mathbf{Y}}}{\partial n} \mathbf{B}) \mathrm{d}s$$

$$\overline{\mathbf{Y}} = \mathrm{diag}\{\frac{\cos(\lambda_x r)}{4\pi r}, \frac{\cos(\lambda_y r)}{4\pi r}, \frac{\cos(\lambda_z r)}{4\pi r}\}$$

The Direct Boundary Integral Equation Method (DBIE)

- Yan and Li (2006) introduced a new reference function $\mathrm{Y} = \frac{\cos(\lambda r)}{4\pi r} - \frac{\cos(\lambda r')}{4\pi r'}$ and the simplex method to avoid the numerical expensive computation of the volume integral.

$$\mathrm{B}_p(x_i, y_i, z_i) = \int_s \frac{z_i[\lambda r\ \sin(\lambda r)+\cos(\lambda r)]\mathrm{B}_{0p}(x,y,0)}{2\pi[(x-x_i)^2+(y-y_i)^2+z_i^2]^{3/2}} dx\, dy$$

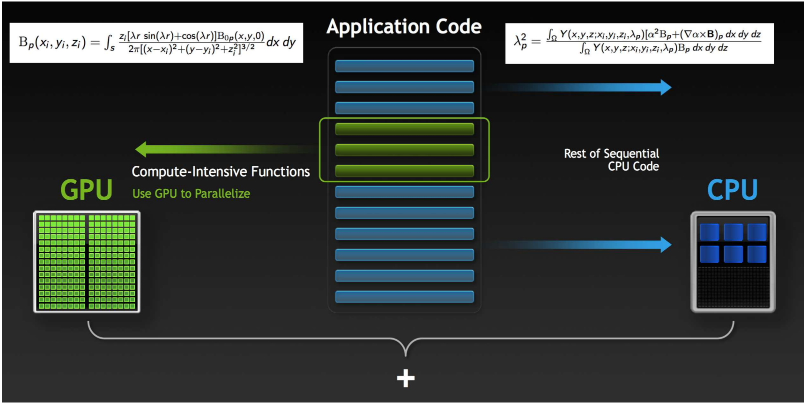
$$\lambda_p^2 = \frac{\int_\Omega Y(x,y,z;x_i,y_i,z_i,\lambda_p)[\alpha^2\mathrm{B}_p+(\nabla\alpha\times\mathbf{B})_p\, dx\, dy\, dz}{\int_\Omega Y(x,y,z;x_i,y_i,z_i,\lambda_p)\mathrm{B}_p\, dx\, dy\, dz}$$

Aim: To speed up the DBIE code.

Why we do this

- Numerical test demonstrated the DBIE a robust method (Yan and Li 2006).
- The DBIE allows to evaluate the NLFFF field at every arbitrary point within the domain from the boundary data, without the requirement to compute the field in an entire domain (Wiegelmann 2008).
- The original DBIE code is slow comparing to other methods if a magnetic field has to be evaluated in an entire 3D domain (Schrijver et al. 2006).

Speed-up by GPU acceleration



Performance

The boundary is a 121×121 pixel grid.

	Original DBIE code	New DBIE code
language	Fortran & IDL	cuda C
platform	Intel(R) Core(TM) i7-3820QM @2.70GHz	Intel(R) Xeon(R) X5650 @2.67GHz +448 cores GeForce GTX 480
time	30 s/one arbitrary point	0.005 s/one arbitrary point

- The new code speed up for 6,000 times.
- 22 min $\Rightarrow$ a 64^3-pixel box within the 121^2 boundary.
- 22 hr $\Rightarrow$ a 300^3-pixel box within the 300^2 boundary.

Performance

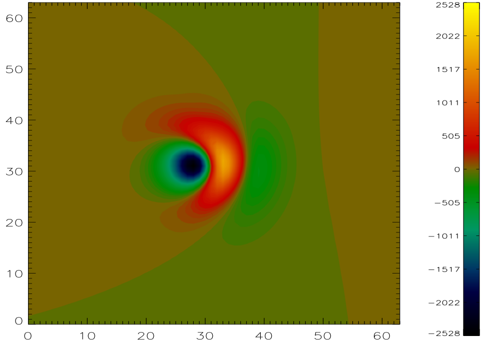
The boundary is a 121×121 pixel grid.

	Original DBIE code	New DBIE code
language	Fortran & IDL	cuda C
platform	Intel(R) Core(TM) i7-3820QM @2.70GHz	Intel(R) Xeon(R) X5650 @2.67GHz +448 cores GeForce GTX 480
time	30 s/one arbitrary point	0.005 s/one arbitrary point

- 4 GPU nodes
- 22 min/4 nodes = 5.5 min $\Rightarrow$ a 64^3-pixel box.
- 22 hr/4 nodes = 5.5 hr $\Rightarrow$ a 300^3-pixel box.

Low-Lou model

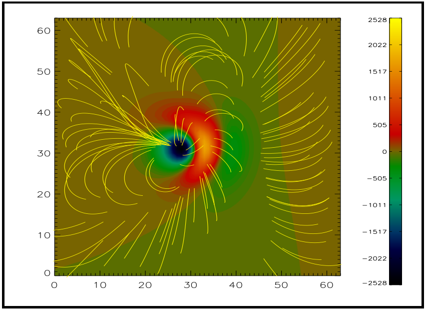
Case II (Schrijver et al. 2006)



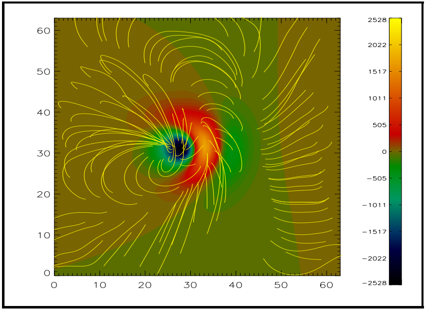
- $n_{\mathrm{LL}} = 3, m_{\mathrm{LL}} = 1$
- $l = 0.3, \Phi = 4\pi/5$
- $x,\ y \in [-3,\ +3]$
- 192×192 pixel grid
- test region: 64^3-pixel

Low-lou Results

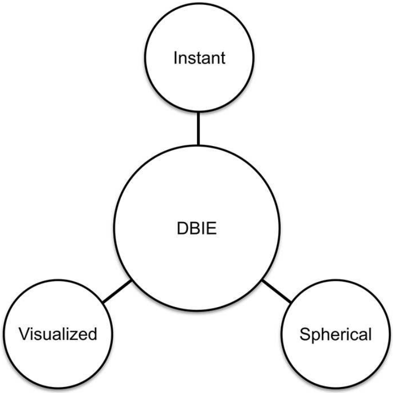
Exact Result



Numerical Result



What's next?



- A visualization tool
- Instant modeling
- Spherical implementation

References

[1] Yan, Y., & Sakurai, T. 2000, Sol. Phys., 192, 89
[2] Yan, Y., & Li, Z. 2006, ApJ, 638, 1162
[3] Wiegelmann, T. 2008, J. Geophys. Res., 113, A03S02
[4] SCHRIJVER, C. J. et.al 2006, Sol. Phys., 235, 161