VORTEX RING DYNAMICS AND INTERACTIONS IN THE NONLINEAR SCHRÖDINGER EQUATION

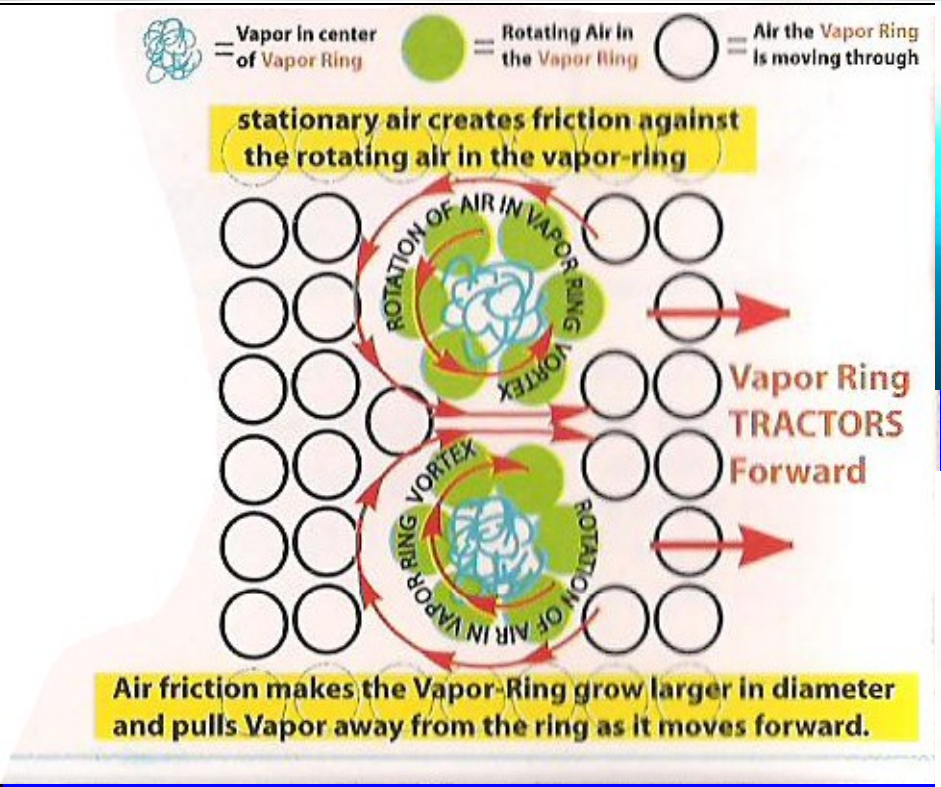
Ronald M. Caplan, Ricardo Carretero
Nonlinear Dynamical Systems Group
Computational Science Research Center
San Diego State University

IMACS 2011

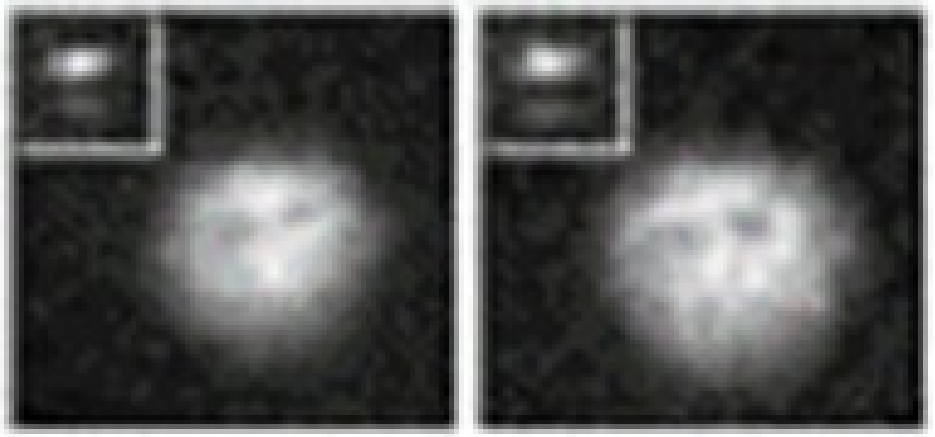
Outline

- Introduction to Vortex Rings
- Mathematical Models
- High-Order Compact Schemes
- GPU Code Implementation
- Simulation Results and Future Work

Vortex Rings



Side Front

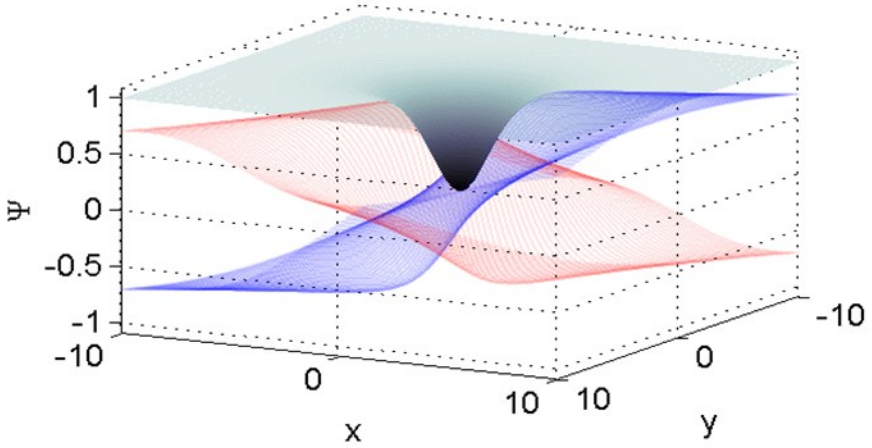


B.P. Anderson et al.
Phys. Rev. Lett. 86,
2926-2929 (2001).

NLSE and Two-Dimensional Vortex Review

Nonlinear Schrödinger Equation

$$i\frac{\partial \Psi}{\partial t} + a\,\nabla^2\Psi + s\,|\Psi|^2\,\Psi = 0$$



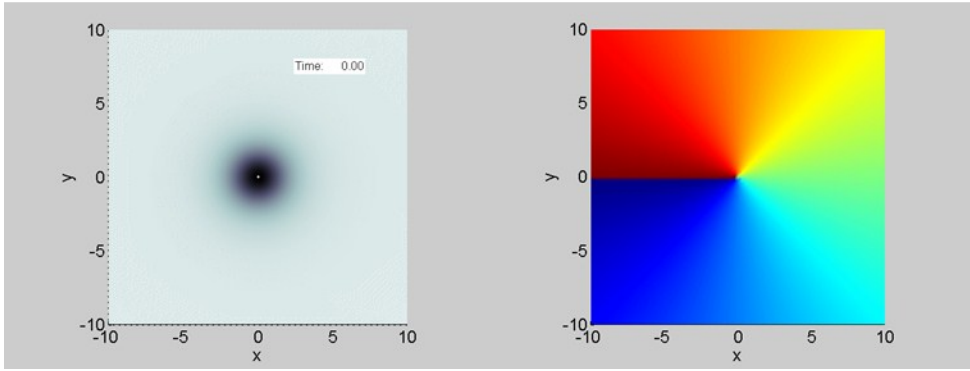
Structure of 2D vortices

$$\Psi(r,\theta,t) = f(r)\exp[i(m\theta + \Omega t)]$$

$$f(r) \in \Re \qquad m : \text{charge}$$

$$-\left(\Omega + \frac{am^2}{r^2}\right) f + \frac{a}{r} f' + a f'' + s\,|f|^2\,f = 0$$

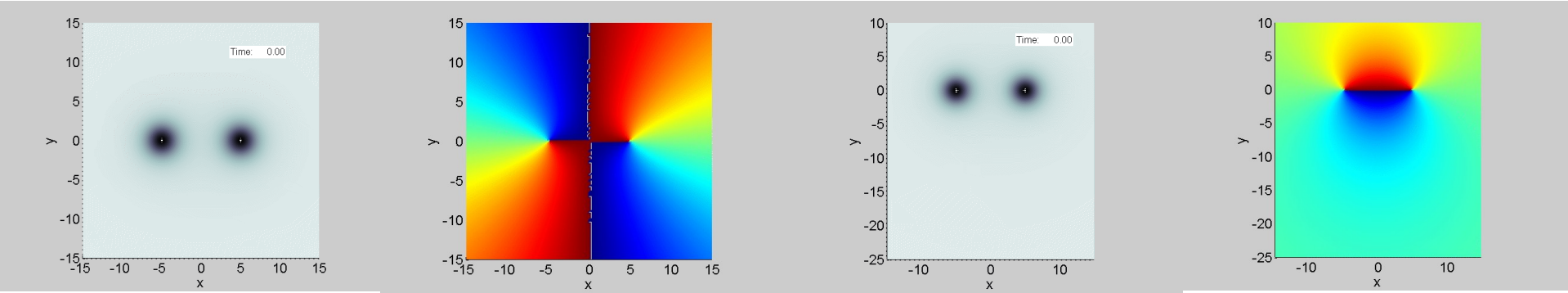
$$f(r) \approx \Re\left[\sqrt{\frac{\Omega}{s} + \frac{a\,m^2}{s\,r^2}}\right]$$



Interaction Dynamics

Equal unitary charge $v_a \approx a/(4d^2)$

Opposite unitary charge $v_p \approx a/(2d)$

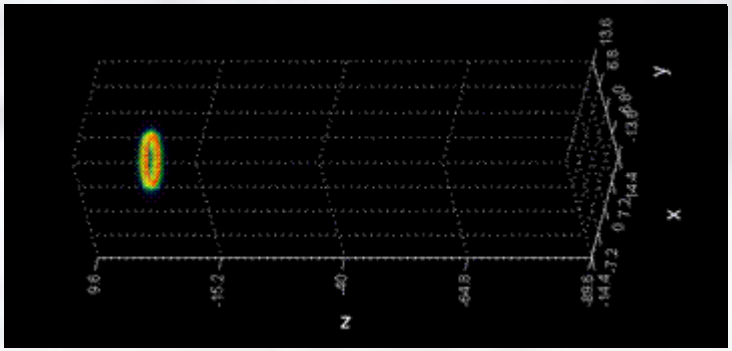


Vortex Rings

Due to vorticity, vortex rings have intrinsic velocity

[P.H. Roberts et. al (1971)]

$$c \approx -\frac{a\,m}{d}\left[\ln\left(8\,d\sqrt{\frac{|\Omega|}{a}}\right) - 0.615\right]$$



Co-moving Solutions to the NLSE (in z-direction)

$$\Psi(x,y,z,t) = \exp[ip(x,y,z,t)]\,U(x,y,Z) \qquad Z = z - c\,t$$

$$p(x,y,z,t) = \frac{c}{2a}\,z + \left[\Omega - \frac{c^2}{4a}\right]\,t + \Phi$$

Arbitrary phase shift

$U(x,y,Z)$ satisfies steady-state NLSE:

$$-\Omega\,U + a\nabla^2 U + s|U|^2 U = 0$$

Therefore, can find initial condition to a steady-state vortex ring solution as

$$\Psi(x,y,z,0) \rightarrow \Psi(x,y,z,0)\exp(-i\frac{c}{2a}z)$$

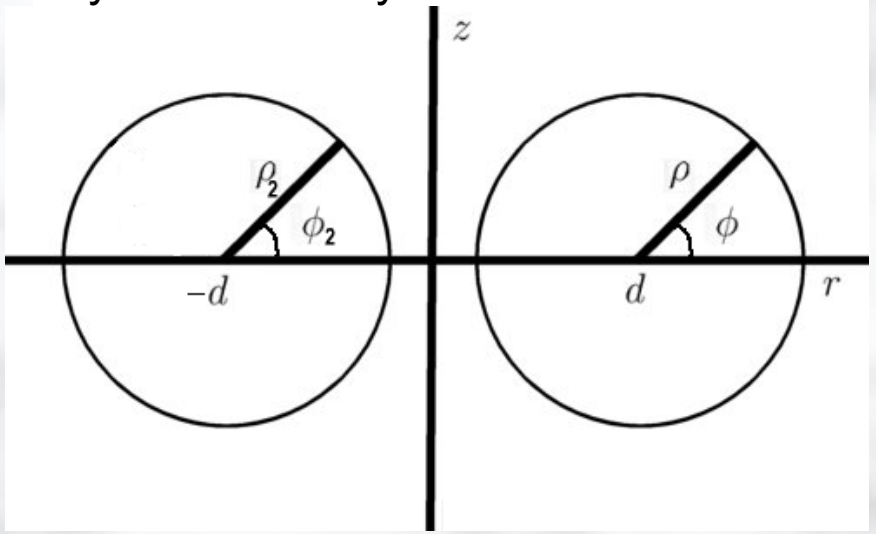
Vortex Ring Structure

Form of vortex ring in half-plane:

$$\Psi_0(r,\theta,z) = g(r,z)\exp[i\,m\,\phi(r,z;d)]$$

$$g(r,z) \in \mathcal{C}$$

Cylindrical Axisymmetric Coordinates



Formulate co-moving steady-state:

$$U_0 = \Psi_0 \exp(-i\frac{c}{2a}z)$$

Insert into co-moving steady-state NLSE:

$$-\left[\Omega + \frac{c^2}{4\,a} + \frac{a\,m^2}{\rho^2} - (r-d)\frac{m\,c}{\rho^2}\right] g + a\left[\frac{1}{r}\,g_r + g_{rr} + g_{zz}\right] + s\,|g|^2\,g +$$

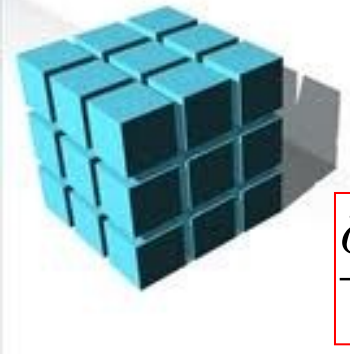
$$i\,\frac{a\,m}{\rho^2}\left[2\,(r-d)\;g_z - z\left(2g_r + \frac{1}{r}\,g\right)\right] - i\,c\,g_z = 0, \qquad \boxed{\rho = (r-d)^2 + z^2}$$

Better form for structure leads to new steady-state PDE

$$\Psi_0(r,\theta,z) = g(r,z)\exp[i\,m\,(\phi(r,z;d) - \phi_2(r,z;d))]$$

Two-Step High-Order Compact Scheme

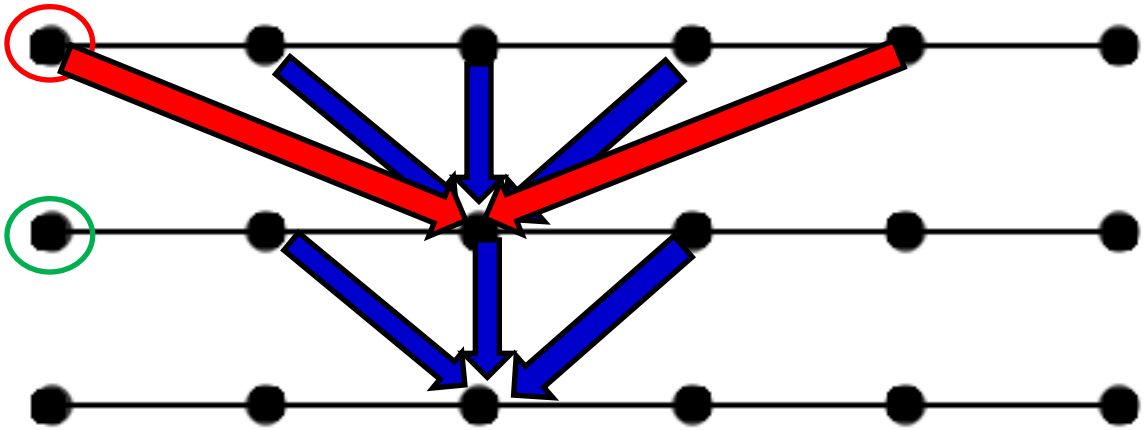
$$\nabla^2\Psi = \frac{\partial^2\Psi}{\partial x^2} + \frac{\partial^2\Psi}{\partial y^2} + \frac{\partial^2\Psi}{\partial z^2}$$



Ψ_i

$$\frac{\partial^2\Psi_i}{\partial x^2} + O(h^4)$$

$$\frac{\partial^4\Psi_i}{\partial x^4} + O(h^2)$$

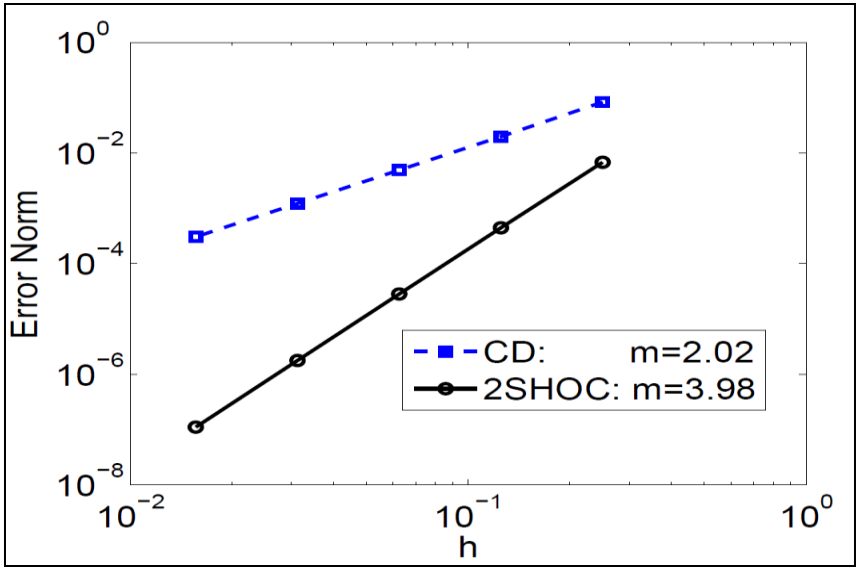


$$\frac{\partial^2\Psi_i}{\partial x^2} = \frac{\Psi_{i+1} - 2\Psi_i + \Psi_{i-1}}{h^2} - \frac{h^2}{12}\frac{\partial^4\Psi_i}{\partial x^4} + O(h^4)$$

2SHOC (1D)

1) $$D_i = \frac{\Psi_{i+1} - 2\Psi_i + \Psi_{i-1}}{h^2}$$

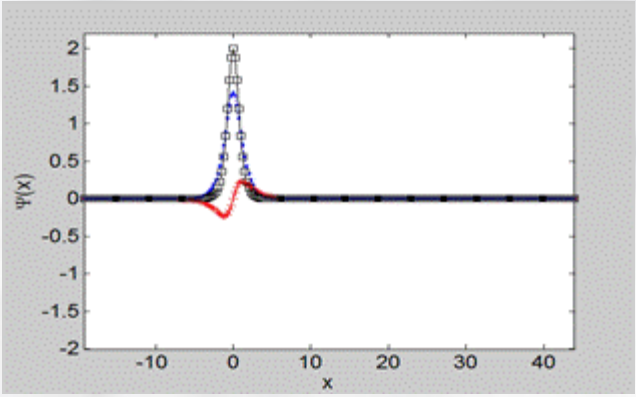
2) $$\frac{\partial^2\Psi_i}{\partial x^2} \approx \frac{7}{6}D_i - \frac{D_{i+1} + D_{i-1}}{12}$$



MSD Boundary Conditions

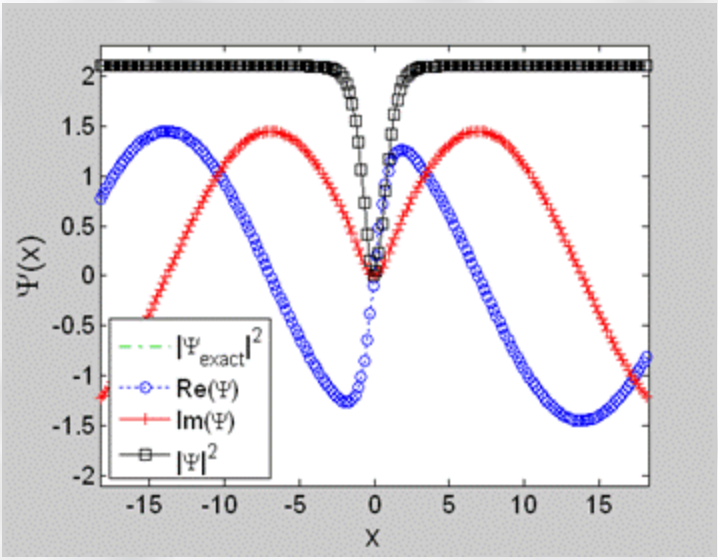
Want easy yet useful BCs

Dirichlet
Works great if "action" $\Psi_B = C$
Is far from the boundary.



For defocusing NLSE with no external potential, Dirichlet fails due to phase rotation.

Would like equivalent Dirichlet but for the modulus-squared: $|\Psi_B|^2 = C$



Modulus-Squared Dirichlet (MSD) boundary condition:

$$\frac{\partial \Psi_B}{\partial t} \approx \left(\frac{\Psi_B}{\Psi_{B-1}}\right) \frac{\partial \Psi_{B-1}}{\partial t}$$

Already computed!

Works for any time-dependent complex PDE and in any dimension

Applying to NLSE: $$\nabla^2 \Psi_B \approx \left[\frac{\nabla^2 \Psi_{B-1}}{\Psi_{B-1}} - \frac{V_{B-1} - V_B}{a} + \frac{s}{a}\left(|\Psi_{B-1}|^2 - |\Psi_B|^2\right)\right] \Psi_B$$

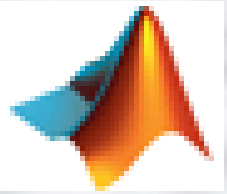
GPU Code Implementation

Graphical Processing Units (GPUs)

- **Cheap**: ~$0.75 per core for desktop card,
  ~$5 per core for compute-only workstation card.
- **Fast**: Up to **1024 cores** at ~1.5GHz per card,
  Top card boasts peak ~2.0 **teraflops**
- Medium-large problem capacity: RAM 1.5GB -> 6.0GB

Why doesn't everyone use it?

- **Not for all problems**
- **Learning Curve:** Need special C/FORTRAN code using CUDA API
- **Preexisting Codes:** Translation of standard parallel code into GPU "difficult".
- **Portability:** Vender-specific API. **Solution: OpenCL**



MathWorks® MATLAB

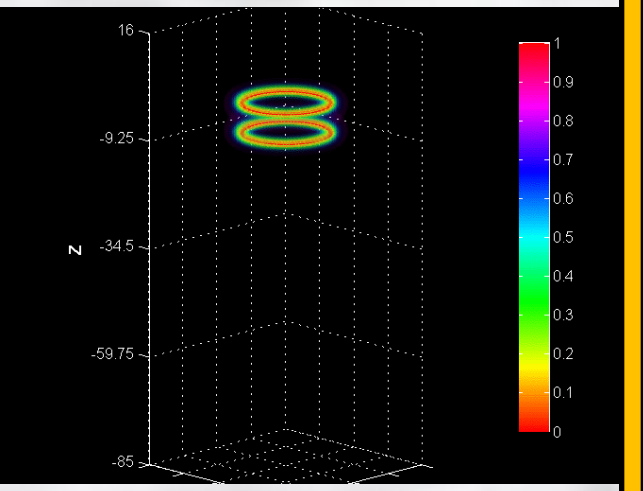
- **Advantages:** Easy integrated visualizations, analysis tools. built-in math functions, easy to code, etc
- **Problems:** Since codes are scripts (not compiled), "real" code use is SLOW
- **Solution:** Can write MEX files: C codes with a MATLAB interface
- **CUDA nvMEX:** Plugin allows CUDA C codes to be compiled into MEX

Development of CUDA MEX codes for integrating the NLSE in 1D/2D/3D have been completed

GPU Code Implementation

Timing Result

MATLAB Script



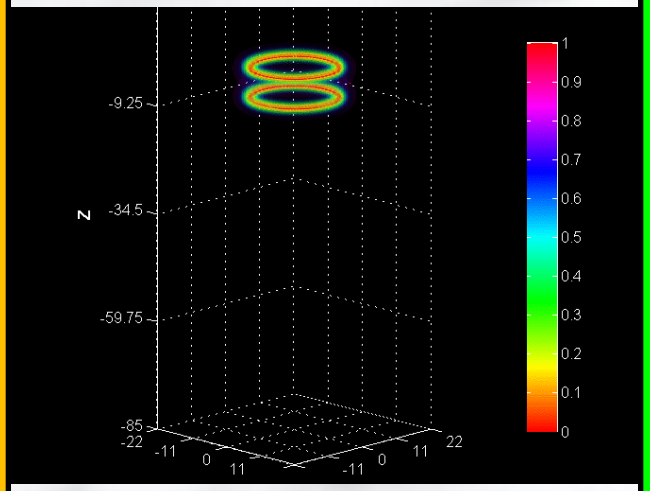
Simulation Details

Method: RK4 + 2SHOC
End time: 100.8
Time steps: 3360
k = 0.03 h = 0.5
Grid: 87x87x203 = 1,536,507
60 frames with 56 steps each
Plotting/movie time ~22.4 sec

Compute time

3:30:43

MATLAB MEX



System Details

Corei3 **3.07 GHz** Dual-thread
4GB RAM 64-bit Windows 7
MATLAB R2010b **Cost~$650**

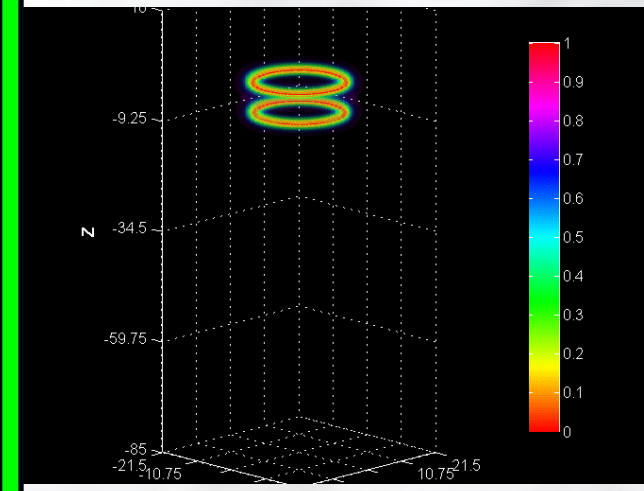
Compute time

0:37:47

Speedup

5.58X faster than script

MATLAB CUDA-MEX



GPU Details

NVIDIA GTX 580 **Cost ~$500**

Compute time

0:02:42

Speedup

13.95X faster than MEX

Speedup

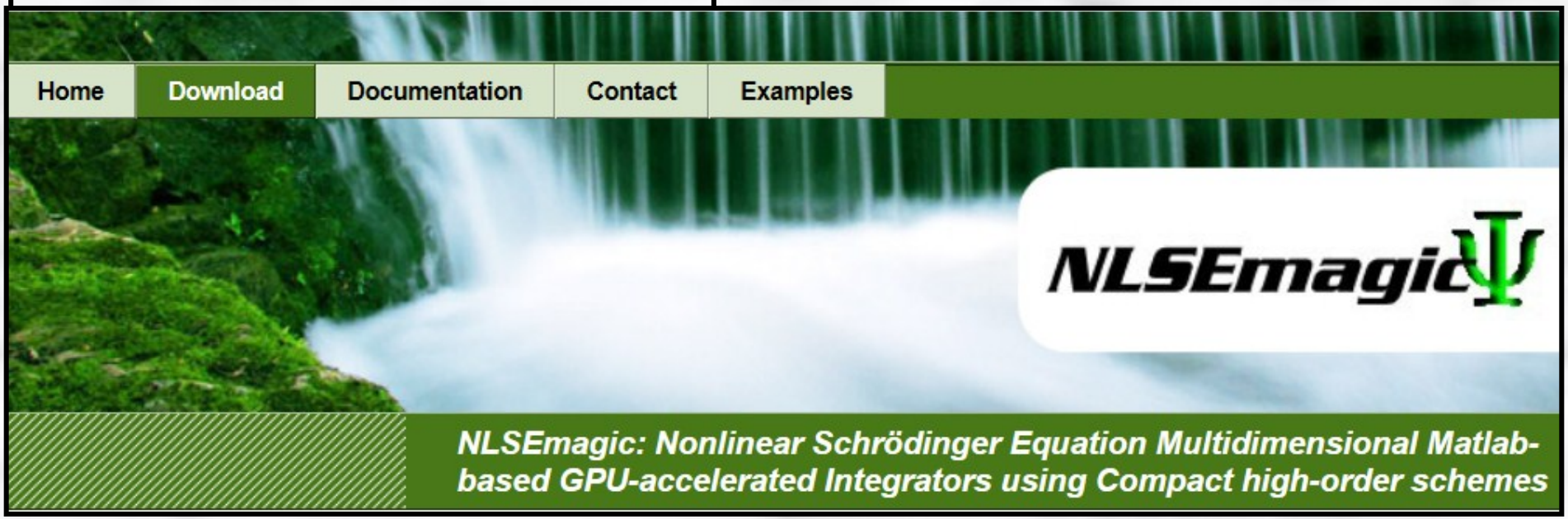
77.8X faster than script

Code Distribution: NLSEmagic

Distribution of codes

- Reproducibility of Computational Research
- Contribution to further studies involving the NLSE

http://www.nlsemagic.com



Full research scripts and documentation to come soon…

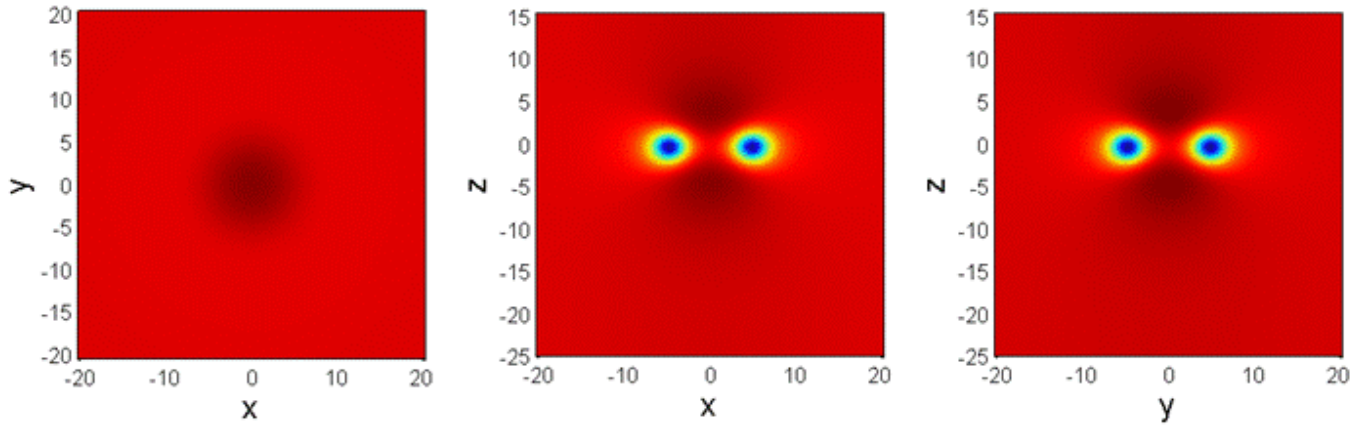
Simulation Results and Future Work

Single Vortex Ring

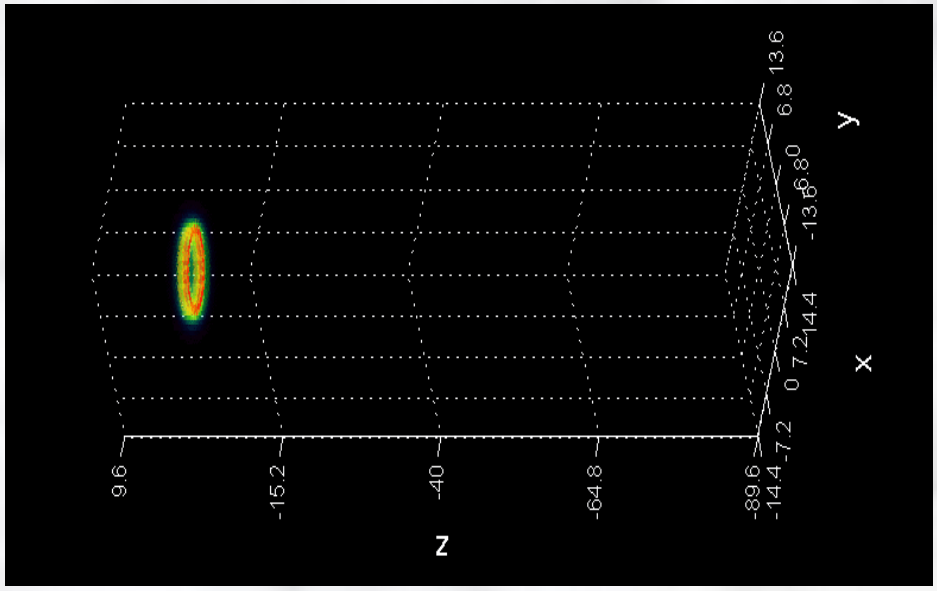
[C. T. Kelley]

Numerically "Exact" IC

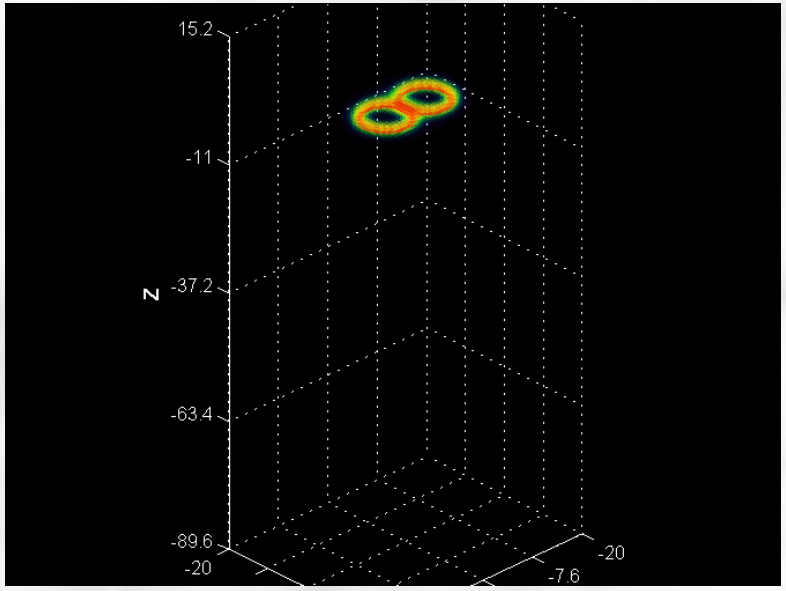
$$\Psi_0 \Psi_0(r, \theta, z) \rightarrow \boxed{\text{nsoli.m}} \rightarrow \Psi_0^*(r, \theta, z)$$



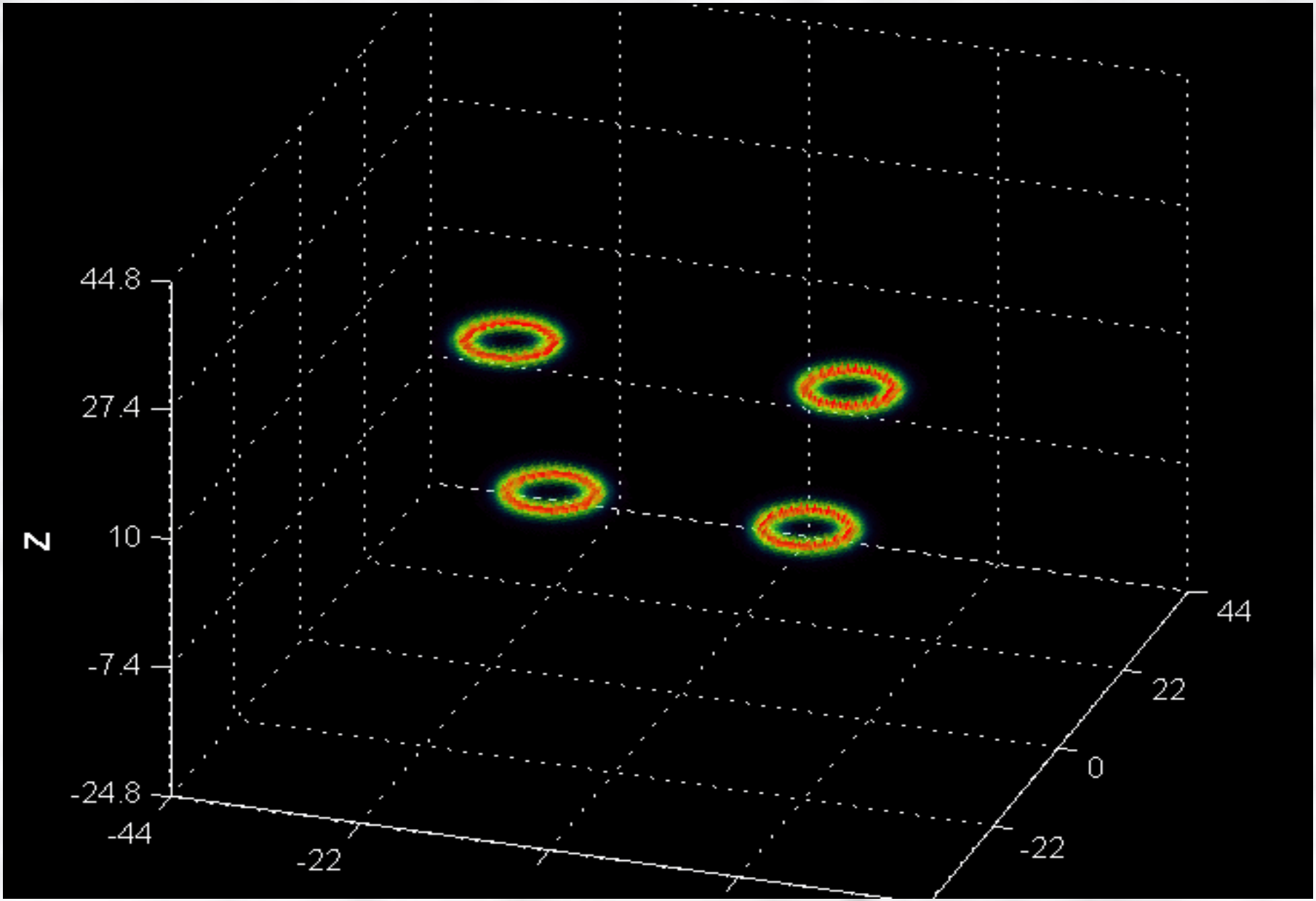
Co-moving VR Stable



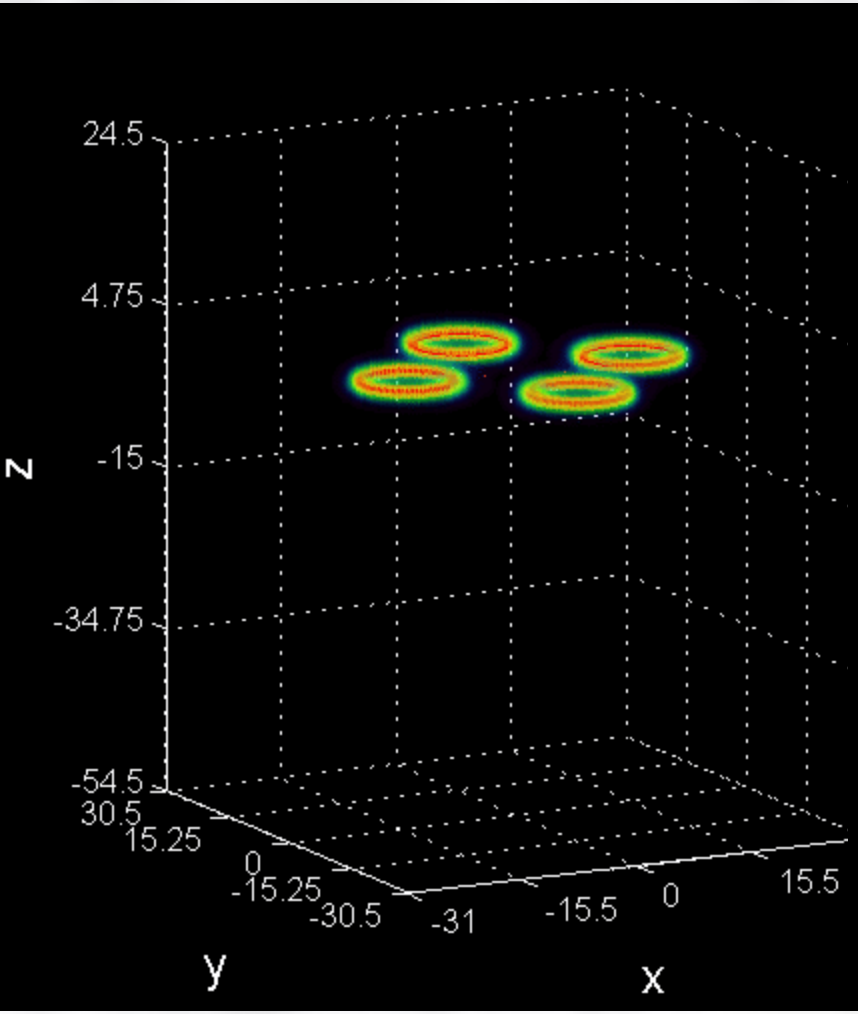
Quadrupole Oscillations

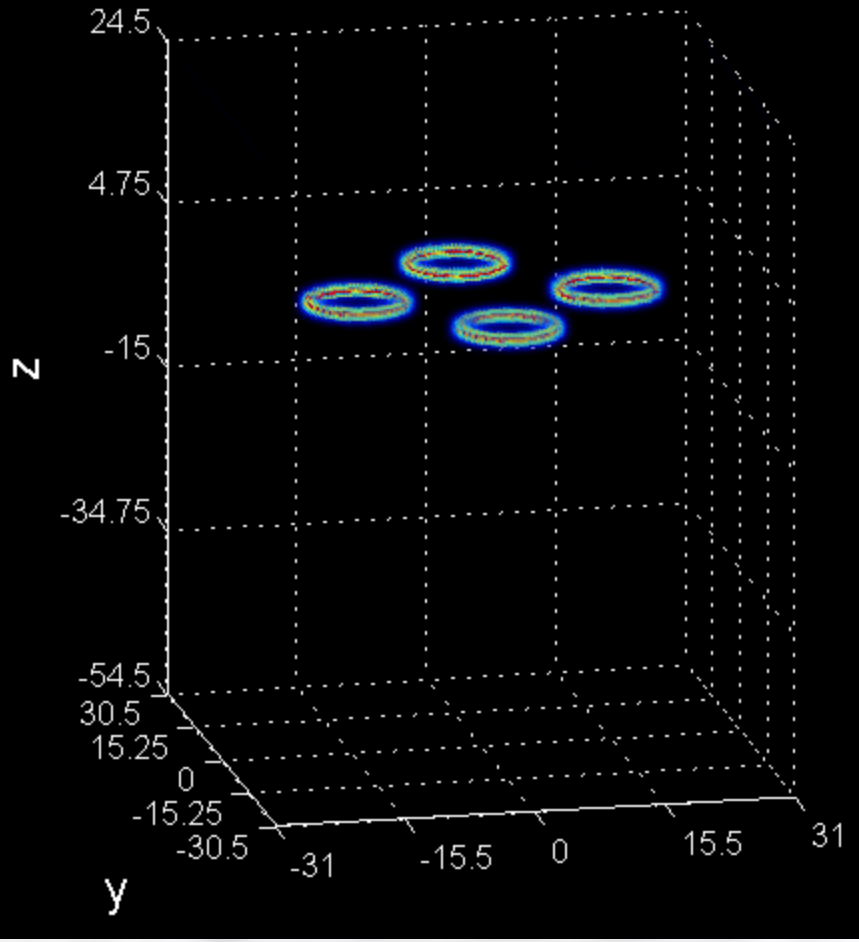


Multiple Vortex Ring Interactions



Multiple Vortex Ring Interactions





Conclusion

- Vortex rings in the NLSE
- Compact High-Order Schemes + MSD BC
- GPU Computing
- NLSEmagic.com
- Multiple interesting dynamic scenarios possible
- Future work

Questions?

2SHOC: Some Details

Second order differencing:

1) $$D_{i,j} = \frac{1}{h^2} \begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & -4 & 1 \\ \hline & 1 & \\ \hline \end{array} \Psi_{i,j}$$

Define:

Take second derivative of differencing

2) $$\nabla^2 \Psi_{i,j} \approx -\frac{1}{12} \begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & -12 & 1 \\ \hline & 1 & \\ \hline \end{array} D_{i,j} + \frac{1}{6h^2} \begin{array}{|c|c|c|} \hline 1 & & 1 \\ \hline & -4 & \\ \hline 1 & & 1 \\ \hline \end{array} \Psi_{i,j},$$ 2)

Fourth order approximation:

$$\frac{\partial^2 \Psi_j}{\partial x^2} \approx D_j - \frac{1}{12}(D_{j+1} - 2D_j + D_{j-1})$$

2SHOC (1D)

1) $$D_j = \frac{\Psi_{j+1} - 2\Psi_j + \Psi_{j-1}}{h^2}$$

2) $$\frac{\partial^2 \Psi_j}{\partial x^2} \approx \frac{7}{6} D_j - \frac{D_{j+1} + D_{j-1}}{12}$$

Fourth order Runga Kutta (RK4):

$$\Psi^{n+1} = \Psi^n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = F(\Psi^n) \qquad k_2 = F(\Psi^n + \frac{\Delta t}{2} k_1)$$

$$k_3 = F(\Psi^n + \frac{\Delta t}{2} k_2) \quad k_4 = F(\Psi^n + \Delta t k_3)$$

MSD BC: Some Details

Constant density at boundary:

$$|\Psi_0|^2 = B$$

Separate real and imaginary parts:

$$\frac{\partial}{\partial t}|\Psi_0|^2 = 0 \longrightarrow \Psi_R \frac{\partial \Psi_R}{\partial t} = -\Psi_I \frac{\partial \Psi_I}{\partial t}$$

Solution to ODEs, but need C:

$$\frac{\partial \Psi_R}{\partial t} = C\Psi_I \qquad \text{and} \qquad \frac{\partial \Psi_I}{\partial t} = -C\Psi_R$$

Rewrite Solution

$$\frac{\partial \Psi_0}{\partial t} = C\Psi_0^I - iC\Psi_0^R = -iC\Psi_0.$$

Take spatial derivative

$$\frac{\partial \Psi_0}{\partial x} = -\frac{1}{iC}\frac{\partial}{\partial x}\Psi_{t,0}.$$

Discretize

$$\frac{\Psi_1 - \Psi_0}{h} - \frac{h}{2}\Psi_{xx,0} + O(h^2) = \frac{1}{-iC}\left[\frac{\Psi_{t,1} - \Psi_{t,0}}{h} - \frac{h}{2}\Psi_{txx,0} + O(h^2).\right],$$

Use above to get C:

$$C = i\frac{\Psi_{t,1}}{\Psi_1}.$$

MSD:

$$\boxed{\frac{\partial \Psi_0}{\partial t} \approx \left(\frac{\Psi_0}{\Psi_1}\right)\frac{\partial \Psi_1}{\partial t}}$$

Applying to NLSE:

$$\nabla^2\Psi_0 \approx \left[\frac{\nabla^2\Psi_1}{\Psi_1} + \frac{V_1 - V_0}{a} + \frac{s}{a}(|\Psi_1|^2 - |\Psi_0|^2)\right]\Psi_0$$

GPU Code Implementation