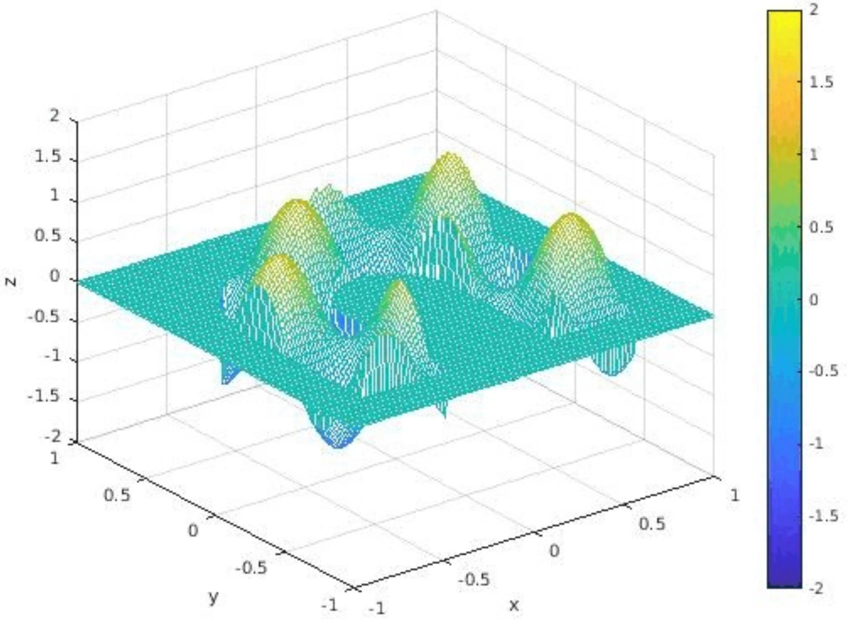
# Developing an FFT-Accelerated High-Order Solver for Heat Equations Over Irregular Domains





Eric Boerman

Brendan Coffey, Julia Zelinsky, Dr. Chuan Li


04/29/2022

---

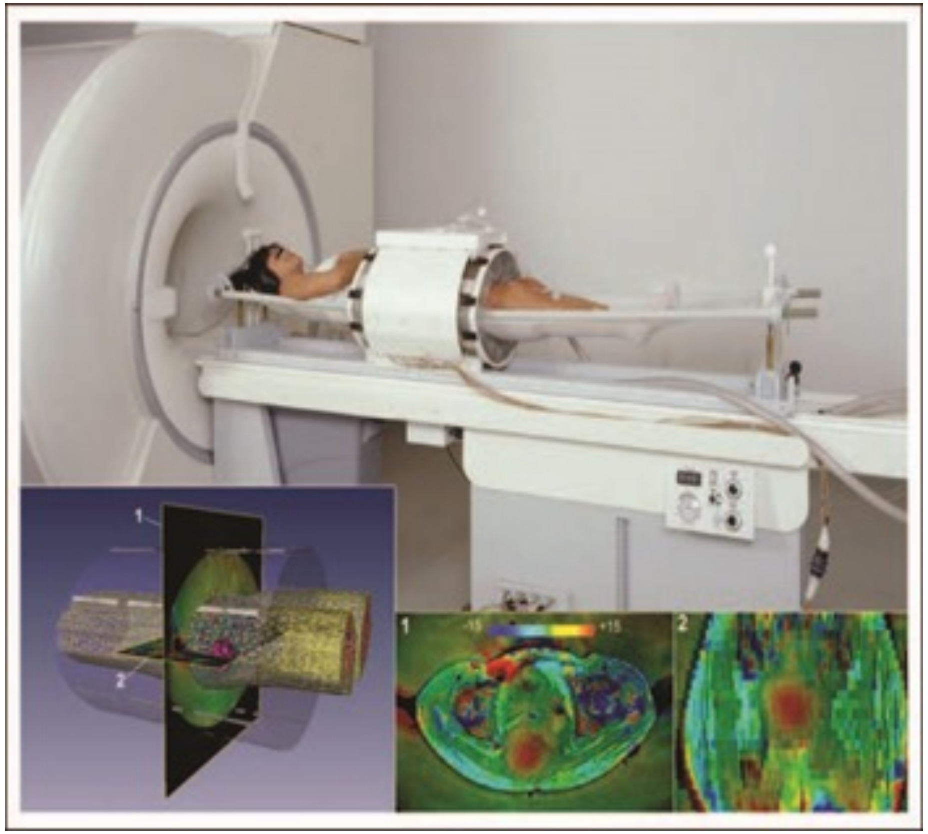
Developed with generous support from the Research in Mathematics and the Sciences grant from West Chester's College of Sciences and Mathematics

# The Goal

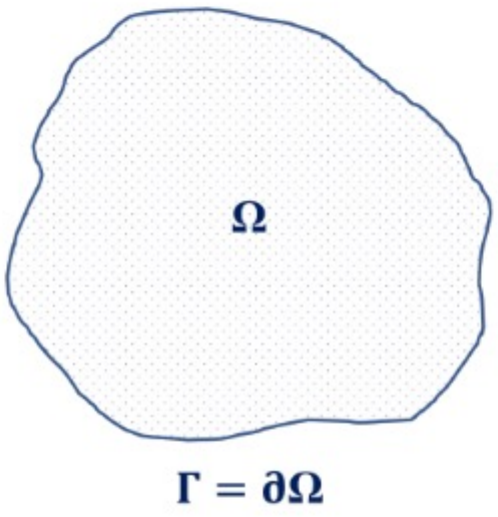
Magnetic Fluid Hyperthermia (MFH): Use magnetic nanoparticles in an alternating magnetic field to generate heat

Targeted application of heat can burn out tumorous growths in human tissue

But the big question – just how precisely can we target when heat spreads over time?



# Mathematical Formulation




(a) A Boundary Value Problem (BVP) imposed on a 2D complex-shaped domain Ω

The Heat Equation: $\frac{\partial u}{\partial t} = \beta \Delta U + f$

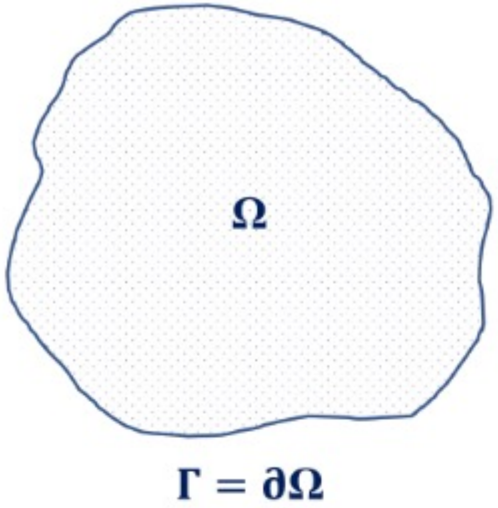
or the Poisson Equation: $\Delta U = f$ (where $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$)

Subject to the most general boundary conditions:

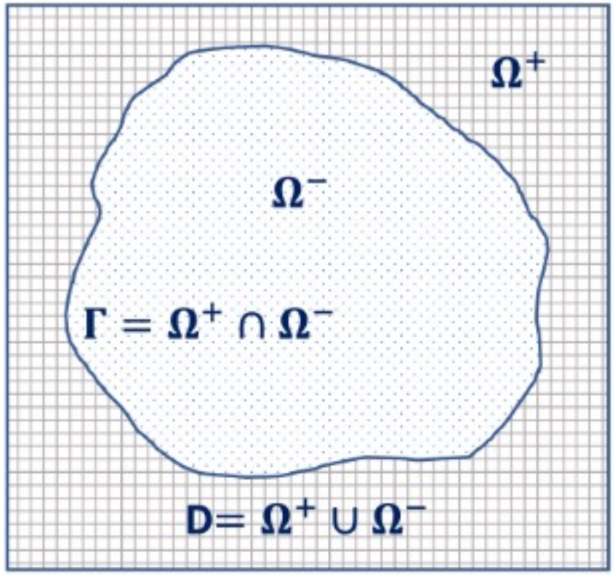
$$\alpha_\Gamma U + \beta_\Gamma \frac{\partial U}{\partial n} = \phi$$

On some domain Ω bounded by a closed interface Γ

# Converting to an Immersed Interface




(a) A Boundary Value Problem (BVP) imposed on a 2D complex-shaped domain $\Omega$




(b) An equivalent Immersed Boundary Problem (IBP)

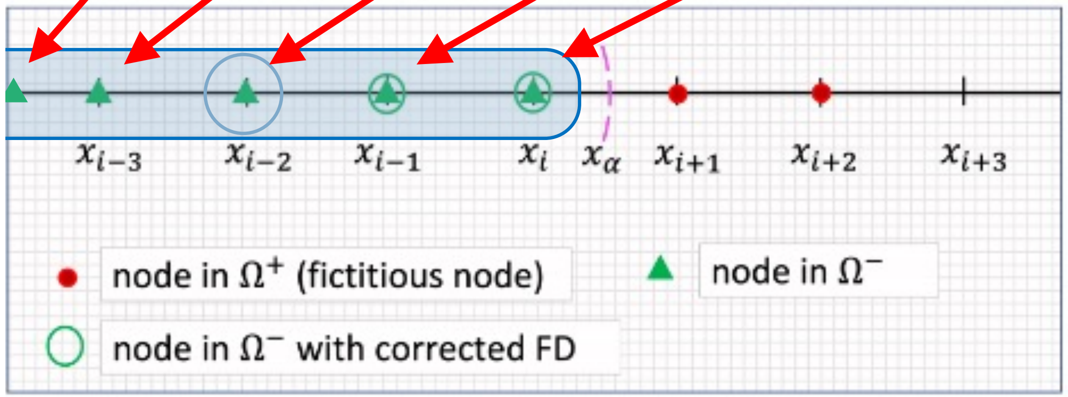
Exterior layers of constant value are added everywhere outside the original problem.

This enables fast solving later, as our method requires periodic boundary conditions.

# Fictitious Values and Jump Conditions

$$u_{xx}(x_i) \approx \frac{1}{h_x^2}\left[-\frac{1}{12}u(x_{i-4}) + \frac{4}{3}u(x_{i-3}) - \frac{5}{2}u(x_{i-2}) - \frac{4}{3}u(x_{i-1}) - \frac{1}{12}u(x_i)\right]$$

$$u(h^-) = \sum_{k=0}^{l} \frac{(-h)^k}{k!} u^{(k)}(h^+) - \sum_{k=0}^{l} \frac{(h^-)^k}{k!}\left[u^{(k)}\right] + O(h^{l+1})$$




For a fourth-order method, we need two layers of fictitious values outside our original domain.

These fictitious values are extrapolated from interior values and boundary conditions.

Do we know what they are? Not yet! But we know how they'd behave if they *were* real, which is enough.

# Augment System

We express all the equations we've been gathering as a blockwise system of matrices:

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{U} \\ \boldsymbol{Q} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F} \\ \boldsymbol{\phi} \end{pmatrix}$$

Our unknown, U, is a vector containing the actual value at every point in our domain – but only once we solve for it.

**Known**

***Matrices***

$\boldsymbol{A} = matrix\ for\ solving\ standard\ approximation$

$\boldsymbol{B} = correction\ term\ coefficients$

$\boldsymbol{C} = weights$

$\boldsymbol{I} = identity\ matrix$

***Vectors***

$\boldsymbol{F} = all\ f(x_i)$

$\boldsymbol{\phi} = known\ interface\ values$

**Unknown**

***Vectors***

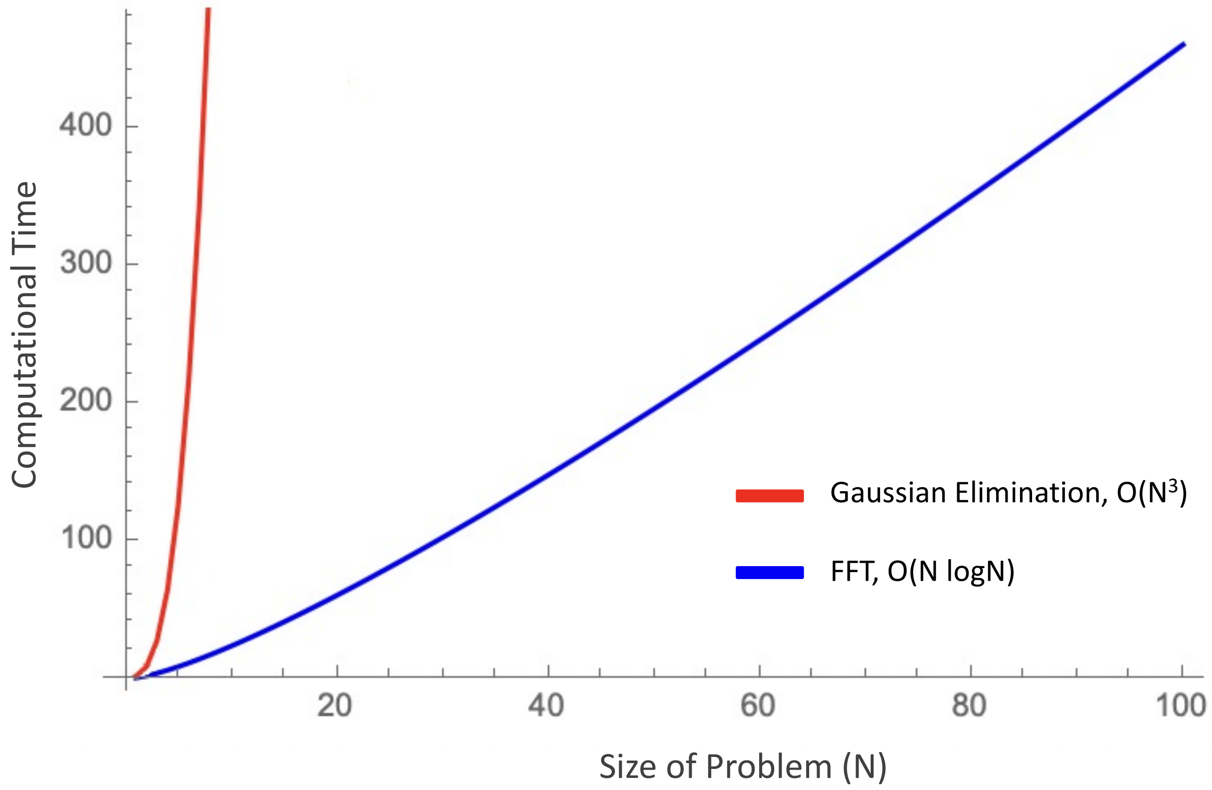
$\boldsymbol{U} = all\ u(x_i)$

$\boldsymbol{Q} = corrections\ at\ interfaces$

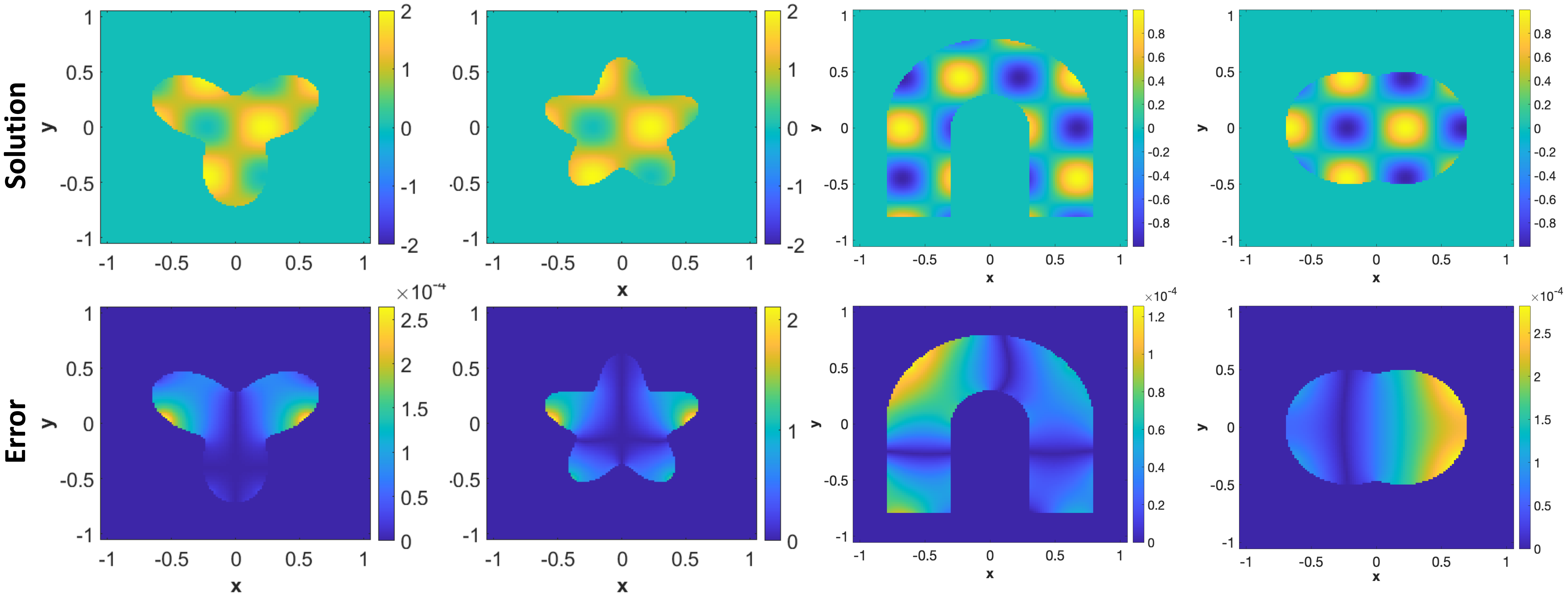
# The Fast Fourier Transform

After expressing all of our equations as matrices, we're ready to solve – but how to actually compute?

The Fast Fourier Transform makes a drawn-out process into one quick multiplication.



# Numerical Experiments - Interfaces



# Numerical Experiments - Convergence

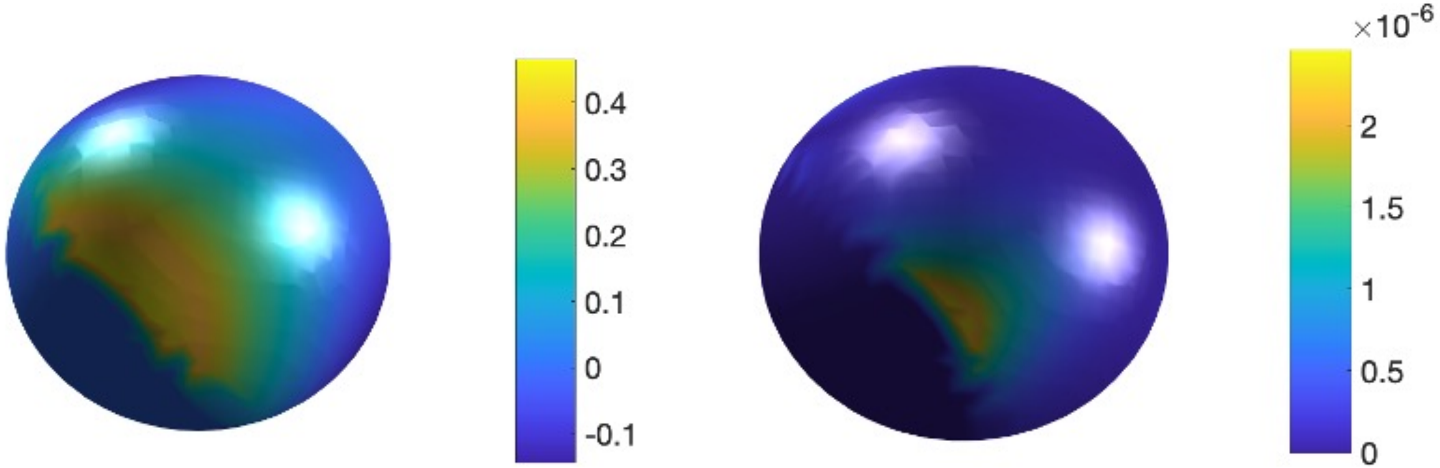
Do we get fourth-order accuracy in space?

If we do, doubling the number of gridlines should cut the error down to one sixteenth ($1/2^4$) of what it was before.

Is this borne out in our experiments?  Yes!

| Number of Points per Dimension | Maximum Error ($L^{Max}$) | Average Error ($L^2$) | $L^{Max}$ Order | $L^2$ Order |
|---|---|---|---|---|
| 65 | 6.71E-05 | 1.25E-05 | | |
| 129 | 3.89E-06 | 5.86E-07 | 4.11 | 4.41 |
| 257 | 1.19E-07 | 3.74E-08 | 5.03 | 3.97 |
| 513 | 6.53E-09 | 2.35E-09 | 4.19 | 3.99 |
| 1025 | 5.34E-10 | 1.66E-10 | 3.61 | 3.82 |

# Numerical Results – 3D




Spheres rotated to show most extreme features

We can solve problems on a variety of interface irregularities, including points, curves, and concavities.

The method is unconditionally stable, fourth-order accurate in space, and as fast as we'd theorized.

# Conclusions

We have succeeded in developing a fast, high-order method for numerically solving heat and Poisson equations on irregular domains.

Numerical experiments confirm we our method achieves a fourth-order accuracy in space, second-order accuracy in time, and computational complexity of $O(N_t N^D \log N)$.

We hope these methods expand the usefulness of mathematical simulation in understanding real-world application of heat flow in irregular situations.

# References

Li, C.; Long, G.; Li, Y.; Zhao, S. Alternating Direction Implicit (ADI) Methods for Solving Two-Dimensional Parabolic Interface Problems with Variable Coefficients. *Computation* 2021, *9*, 79. https://doi.org/10.3390/computation9070079

Wiegmann, A. & Bube, K. P. (1998). The Explicit-Jump Immersed Interface Method: Finite Difference Methods for PDEs with Piecewise Smooth Solutions. *Siam Journal of Numerical Analysis.* Vol. 37, No. 3, pp. 827-862.

Feng, H. & Zhao, S. (2020). FFT-Based High Order Central Difference Schemes for Three-Dimensional Poisson's Equation with Various Types of Boundary Conditions. *Journal of Computational Physics.* https://doi.org/10.1016/j.jcp.2020.109391

Ren, Y., Feng, H. & Zhao, S. (2021). A FFT Accelerated High Order Finite Difference Method for Elliptic Boundary Value Problems over Irregular Domains. *Journal of Computational Physics.* https://doi.org/10.1016/j.jcp.2021.110762

Fornberg, B. (1998). Calculation of Weights in Finite Difference Formulas. *Siam Review.* Vol. 40, No. 3, pp. 685-691. http://www.siam.org/journals/sirev/40-3/32250.html