Suspensions with small, spherical particles

SA104X Degree Project in Engineering Physics

Samuel Zackrisson[1]
Supervisor: Anna-Karin Tornberg[2]

[1,2]Department for Numerical Analysis, Royal Institute of Technology
[1]samuelz@kth.se
[2]akto@kth.se

April 17, 2015

Contents

1 Introduction

Analytic solutions for systems of many particles in fluid dynamics and electrostatics are few and far-between. Simulations are therefore essential to studying these systems. Studied in this project is the simulation of a group of particles sedimenting in a fluid. This is commonly done without directly calculating the fluid's velocity field - for example in [1] - as in a three dimensional setting with N gridpoints along each dimension gives an $O(N^3)$ scale to the problem. Finding fast methods of performing calculations on this large set can be simplified by making the (intensely debated) assumption of a periodic grid - assuming that the N^3-point cube is replicated much like the primitive cells of a crystal lattice. This introduces a periodicity which makes room for fourier methods and - for smooth functions - spectral (exponential) convergence in solutions and the use of the fast fourier transform, which in some places reduced the complexity to $O(NlogN)$. This is done for example in [2]. In this project a specific method of using the fourier transform is studied, applied to a ball of particles sedimenting in a fluid affected by gravity, and compared to similuar studies of the same system. The purpose is to evaluate this method in comparison with present work on the same simulation.

1.1 Sedimenting suspensions of small particles

ref "Falling clouds of particles in viscous fluids" (Metzger, Nicolas, Guazzelli)

1.2 Fourier methods

1.3 Regularizing delta functions

A method of spreading or smearing a delta function to an N-dimensional grid is explained here. The method is taken from [3] and [4].

1.4 Purpose of this study

1.5 Delimitations

2 Theory

2.1 Stoke's equation

2.2 Fourier series

The Fourier series coefficient $\mathcal{F}(f)_{\mathbf{k}}$ of a function f on the volume $V = [0, L)^3$ is taken to be

$$\mathcal{F}(f)_{\mathbf{k}} = \frac{1}{L^3} \int_V e^{-\frac{2\pi}{L} i(\mathbf{k} \cdot \mathbf{x})} f(\mathbf{x}) d\mathbf{x} \tag{1}$$

The coefficients for the gradient coefficients are

$$\mathcal{F}(\nabla f)_{\mathbf{k}} = \frac{2\pi}{L} i\mathbf{k} \mathcal{F}(f)_{\mathbf{k}} \tag{2}$$

and so Stoke's equation can be written in fourier space as
$(STUFF)$
...here (1) will be discretized to show how it relates to the DFT in Matlab [5].

2.2.1 Solving differential equations using the fft

(Also scaling the fourier transform, for use in **??**)

2.3 Regularized delta functions

2.4 Other ways of doing things

For example the Stokeslet is a popular way of doing things.

3 Setup and algorithms

3.1 The four-step simulation

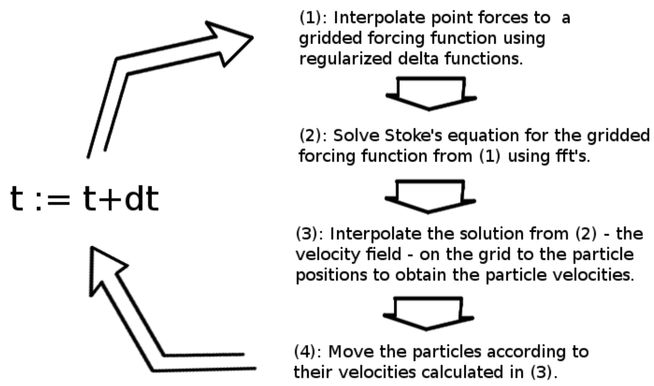

Figure 1: The simulation algorithm steps

Each timestep the simulation has four distinct moments, as illustrated in figure 1.

Step 1: Interpolating from particle positions to grid positions Gravitationskraften på partiklarna interpoleras till ett kubiskt 3D-grid på vilket vätskans hastighetsfält ska beräknas. Detta görs genom att en punktkraft – deltafunktion – representeras av en blobb – regulariserad deltafunktion, t ex gaussian eller bara en hatt – som evalueras I gridpunkterna. Detta interpolerar de M utspridda punktkrafterna till N^3 punktkrafter I griddet.

Steg 2: Lös ekvationen Nu när kraftfältet på griddet är uträknat så har vi ett högerled I Stokes ekvation, vilken vi löser med hjälp av snabba fouriertransformen. Det är här antagandet om periodiskt flöde kommer in.

Steg 3: Interpolera hastighetsfältet till partiklarnas positioner Detta steg har inte studerats ingående, utan genomförs med enkel polynomisk interpolation. Här används Matlabs inbyggda funktion gridinterp för att interpolera gridvärdena till partiklarnas positioner.

Steg 4: Tidsstega fram partiklarna Här används enkelt Eulers stegmetod eller liknande för att stega fram partiklarna med fluidens hastighet.

För att endast studera exempelvis konvergensen I lösningsmetoden för hastighetsfältet givet partiklarnas positioner så räcker steg 1-2. Stokes ekvationer beror enbart av forcingfunktionen I högerledet och därför beror lösningen vid tiden t enbart på partiklarnas position vid tiden t, ej deras tidigare positioner.

3.2 The particle cloud setup used here

4 Results

4.1 Convergence results

4.2 Fourier Convergence results

Fouriermetoder är effektiva vid låg upplösning tack vare spektral konvergens, eller exponentiell konvergens, för släta/glatta funktioner. Det går att visa [REF] att $c_k \approx |k|^{-}n$ för $f \in C^{n-1}$. Med denna enklaste deltafunktions-utsmetning så bör fourier koefficienterna avta som $|k|^{-1}$. Nedan ses vad som händer för fixerat epsilon. Jämför med senare konvergens hos hastighetsfältetslösningar Låter vi epsilon 1/N så avtar felnormen som $N^{-2.97}$, medan epsilon fixt ger felet $N^{-2.52}$. I det senare fallet fås tydligt inte den spektrala konvergens vi hade kunnat förvänta oss om de regulariserade deltafunktionerna hade varit glatta funktioner.

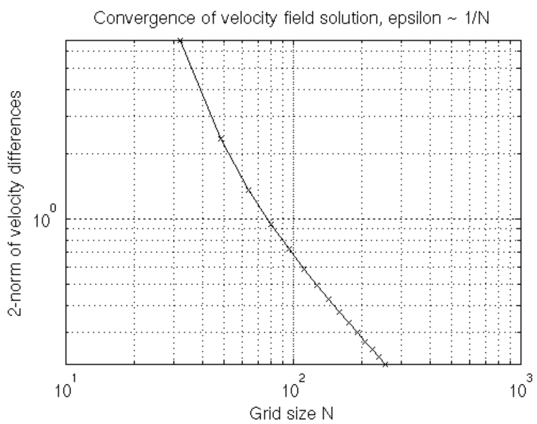

Figure 2: Normed successive differences in solutions as N increases and epsilon decreasing with it.

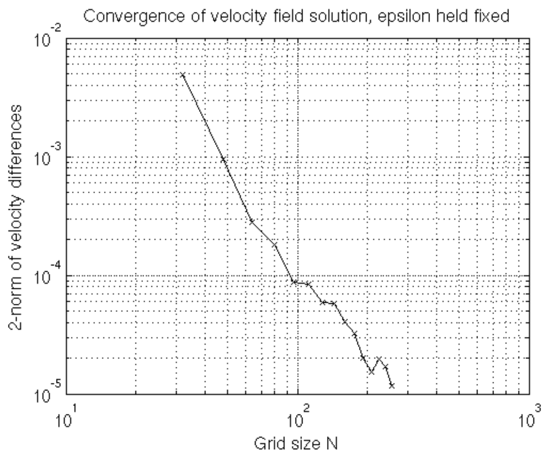

Figure 3: Normed successive differences in solutions as N increases, keeping epsilon fixed.

4.3 scaling the primitive cell

Unlike the system studied by Metzger et al [1], the system studied in this article is periodic. If the cloud of particles is too large compared to the period of the system it is not unlikely to have interactions between clouds of different primitive cells. This can be studied by scaling the length of the primitive cells and looking at the differences between successive solutions for the velocity field in a cube of fixed size containing the particle cloud. The error decreases as $L^{-0.87}$.

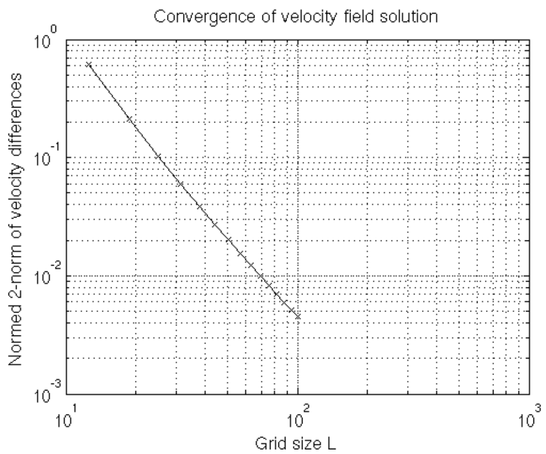

Figure 4: Normed successive differences in solutions as the primitive cell increases.

4.4 Complexity

4.5 Different regularization functions

At the moment the method with cardinal splines is not working, as can be seen in plot 5. The upper data points show the convergence for the regularization using cardinal splines, which should be considerably faster than the second data set from the delta "hats". (Author's note: it's actually working now, but I have not produced any plots and comparisons yet.)

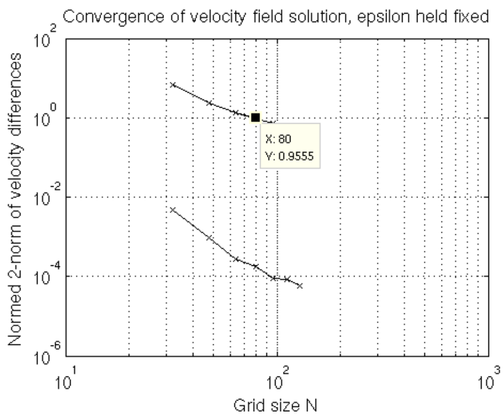

Figure 5: The upper segment is using splines, p=3 and 5 (overlapping). The lower segment is using epsilon hats, which clearly overtakes the splines when it comes to convergence, implying something is wrong with the implementation.

4.6 Phenomena

The evolution of the cloud shape in a periodic box appears to agree with that observed by Metzger et al [1] in a non-periodic setting. The sedimenting ball becomes oblate, forms a torus and eventually breaks into two smaller clouds. This can be confirmed in more detail by studying e.g. the particle density along the z-axis in the cloud center, the pressure gradient in and surrounding the cloud, the velocity field in and around the cloud.
For too few gridpoints the cloud appears to take a square shape, aligned with the grid cells. When this effect in practice disappears can be studied further, e.g. by plotting the particle density as a function of azimuthal angle.

5 Conclusions

References

[1] Bloen Metzger, Maxime Nicolas and Élizabeth Guazzelli, *Falling clouds of particles in viscous fluids*, J. Fluid Mech. (2007) vol. 580, pp. 283-301.

[2] David Saintillan, Eric S.G. Shaqfeh and Eric Darve, *The growth of concentration fluctuations in dilute dispersions of orientable and deformable particles under sedimentation*, J. Fluid Mech. (2006) vol 553, pp. 347-388.

[3] Dag Lindbo and Anna-Karin Tornberg, *Spectral accuracy in fast Ewald-based methods for particle simulations*, Journal of Computational Physics (2011) vol 230, pp. 8744-8761.

[4] Anna-Karin Tornberg and Björn Engquist, *Numerical approximations of singular source terms in differential equations*, Journal of Computational Physics (2004) vol 200, pp. 462–488.

[5] MathWorks (2015), *Discrete Fourier Transform (DFT)*, Matlab R2015a Documentation, retrieved from http://se.mathworks.com/help/matlab/math/discrete-fourier-transform-dft.html.