Exact Fractional Step Methods for Solving the Incompressible Navier-Stokes Equations

J. B. Perot & V. Subramanian[1]

[1]*Department of Mechanical and Industrial Engineering*
University of Massachusetts, Amherst, MA, 01003, USA

Email: *perot@ecs.umass.edu*

ABSTRACT

Fractional step (or projection) methods are a widely used technique for uncoupling the pressure solution in the incompressible Navier-Stokes equations while still satisfying the incompressibility constraint. Traditional fractional step methods have a time splitting error that can become quite significant if any of the terms in the momentum equation are treated implicitly. Higher order splittings with less error are possible but can be expensive and cumbersome. In this paper an approach is demonstrated that still eliminates the pressure efficiently yet results in no splitting error. It is shown that this approach is relatively general and can also be applied to different numerical methods and other types of constraints (such as immersed boundary constraints). In addition, it is shown that iterative inversion methods require fewer iterations (and therefore reduced cost) when the exact projection method is used compared to more traditional approaches.

1. INTRODUCTION

The basic idea behind all fractional step methods is to make some estimate of the velocity field at the next time level and then correct this estimate (which is not divergence-free) as little as possible so that the incompressibility constraint is satisfied. The alteration of the estimate to satisfy the divergence-free velocity constraint means that the momentum equation is not satisfied by the final velocity solution. The correction for the constraint introduces some error. A detailed analysis of this splitting error and traditional methods for reducing it are discussed in [1].

The incompressible (constant viscosity) Navier-Stokes equations in primitive variables are given by the equations,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\nabla p + \nu\nabla^2\mathbf{u} \qquad (1a)$$

$$\nabla\cdot\mathbf{u} = 0 \qquad (1b)$$

where u is the velocity vector, p is the kinematic pressure (divided by density), and ν is the kinematic viscosity.

When these equations are discretized using primitive variables they result in the following coupled matrix problem for the velocity and the pressure,

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix}\begin{bmatrix} u^{n+1} \\ p \end{bmatrix} = \begin{bmatrix} r^n \\ 0 \end{bmatrix} + \begin{bmatrix} bc_1 \\ 0 \end{bmatrix} \qquad (2)$$

where **G** is the discrete gradient operator and **D** is the discrete divergence operator. The matrix **A** contains the mass matrix (which is often diagonal in finite volume methods) as well as contributions from any implicit diffusion and convection. If convection is fully explicit the matrix **A** is symmetric and positive definite. The vector, r^n, contains the explicit diffusion, convection, and even sometimes explicit pressure, contributions. The final vector represents the influence of boundary conditions. There is some debate over the exact time level of the pressure variable. The pressure variable is actually a time average over the time interval, and as such has no particular time level associated with it.

The system given by Eqn. (2) could simply be inverted using a sparse matrix iterative solver. However, even when the matrix system is symmetric, it is not positive definite and the most attractive iterative solvers (like conjugate gradients) can not be applied. In addition, the system is fully coupled and therefore quite large. The work per iteration scales with the number of unknowns, and the number of iterations scales as the number of unknowns to the 1/2 (2D) or 1/3 (3D) power. Fractional step methods are a way to simplify Eqn (2) into a sequence of simpler, and numerically easier to solve, sub-problems.

2. CLASSICAL FRACTION STEP METHODS

Classical fractional step methods can be seen as an approximate block LU decomposition of Eqn (2) [1]. The exact block LU factorization of (2) is

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{D} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{G} \\ \mathbf{0} & -\mathbf{D}\mathbf{A}^{-1}\mathbf{G} \end{bmatrix} \qquad (3)$$

The negative sign in the lower right block is why the full system is indefinite. Fractional step methods are equivalent to the block LU decomposition,

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{D} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \tilde{\mathbf{A}}^{-1}\mathbf{G} \\ \mathbf{0} & -\mathbf{D}\tilde{\mathbf{A}}^{-1}\mathbf{G} \end{bmatrix} \qquad (4)$$

where $\tilde{\mathbf{A}}^{-1}$ is an approximate inverse. If all terms in the Navier-Stokes equations are updated explicitly except for the pressure, and a finite volume method is used then **A** is diagonal and the fractional step method can be implemented exactly with no splitting errors.

The LU decomposition splits the large indefinite system given by Eqn (2) into smaller problems involving the inversion of **A** and the inversion of the discrete Poison matrix $\mathbf{D}\tilde{\mathbf{A}}^{-1}\mathbf{G}$. The matrix **A** is an advection-diffusion operator that is often already coded and which sometimes can be further simplified. The discrete Poison equation is often symmetric and is always a positive definite matrix.

When the matrix **A** contains implicit advection and diffusion contributions or a mass matrix (such as for finite element methods), finding an accurate approximate inverse $\tilde{\mathbf{A}}^{-1}$ can be challenging. Even when formal order of accuracy requirements are met, the resulting approximate inverse can lead to practical errors that are large enough to be disconcerting.

In addition, a further more subtle point must be made about the classical fractional step method. Inexact solution of the discrete Poison equation for the pressure (and all iterative methods are inexact to some extent) leads to errors in the dilation (velocity divergence not equal to zero). So while fractional step methods make the solution close to incompressible, they are not exactly incompressible and the errors in the pressure solution (due to iterative solution – not the splitting error) result in velocity fields with local mass generation and destruction.

The exact fractional step methods have no splitting error, but in addition errors due to inexact solution of the matrices leads only to errors in the vorticity – not the dilatation. Incompressibility is satisfied to machine precision at all times.

3. EXACT FRACTIONAL STEP METHODS

Exact fractional step methods are not an approximate block LU decomposition. Instead they use properties of the discrete gradient, **G**, and divergence **D** operators to simplify the system.

The discrete divergence operator is always wider than it is tall. This must be case or else the number of incompressibility constraints would outnumber the number of velocity variables and the system would be over-constrained and unsolvable. There are therefore a number of vectors $\mathbf{c}_i$ (with a length equal to the number of velocity unknowns) which lie in the null space of the divergence operator, $\mathbf{D}\mathbf{c}_i = \mathbf{0}$. A solution for the velocity which is a linear combination of these $\mathbf{c}_i$ will always satisfy the incompressibility constraint to machine precision.

The key to exact fractional step methods is that the $\mathbf{c}_i$ can always be formed so that they are sparse vectors. For most methods, the full set of $\mathbf{c}_i$ vectors are easily formed explicitly. If we define the set of $\mathbf{c}_i$ grouped as column vectors as the matrix **C**, then we have $u^{n+1} = \mathbf{C}s^{n+1}$, and $\mathbf{DC} = \mathbf{0}$, so that the incompressibility constraint is always satisfied exactly (even when the new unknowns s^{n+1} have some error).

This is a discrete analog of the curl operation. So s^{n+1} is a discrete analog of the streamfunction vector. This does *not* make the method a streamfunction method. Most importantly, boundary conditions on the streamfunction are not required, and frequently only certain components (not the whole vector) of the real streamfunction are represented by s^{n+1}. This is purely a matrix transformation that reduces the number of discrete unknowns while exactly satisfying the constraints. The physical analogy does however allow one to see how to easily generate the sparse curl matrix, **C**. The matrix **C** is actually generated via its rows, not its columns. Each row corresponds to a line integral around a small closed loop. The smaller the loops – the sparser the matrix.

For the very small (2 cell) unstructured staggered mesh shown in Figure 1, the divergence operator is given by,

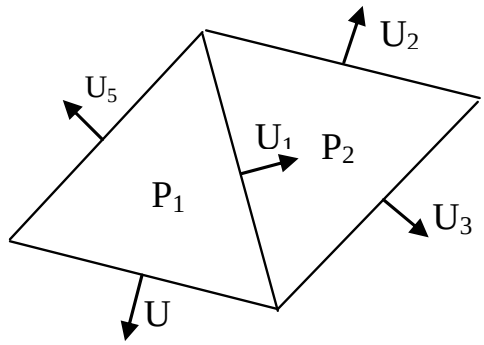

Figure 1. Location of normal velocity components and pressure unknowns in a two-dimensional unstructured staggered mesh.

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ -1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

and the curl matrix is given by

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

In 3D, each row of **C** is the loop formed by the edges surrounding each face. In 2D, such as this example, **C** is the difference between the two vertices connected by each edge. See [2,3] for more details.

When the discrete curl is used, Eqn (2) becomes

$$\mathbf{AC}s^{n+1} + \mathbf{G}p = r^n + bc_1 \qquad (5)$$

and the divergence constraint is automatically satisfied. But equation (5) has a different number of unknowns than equations and can be further simplified by eliminating the pressure.

In contrast to the divergence operator the gradient operator, **G** is taller than it is wide. This means that its columns form the null space of some other matrix, referred to as **R.** In matrix terms, $\mathbf{RG} = \mathbf{0}$. This is the discrete equivalent of the continuous identity $\nabla \times \nabla(\,) = 0$. The 'rotation' matrix **R** is therefore a different discrete curl operator. Many different matrices **R** exist. Some are very sparse and easy to construct. It is these sparse easily constructed matrices that make the exact fractional step method viable.

For finite element methods or staggered mesh discretizations the two discrete curl operations are closely related and are transpose operations, $\mathbf{R} = \mathbf{C}^T$. In 2D each column of **R** corresponds to a loop (sometimes unclosed on the boundaries) of all the edges around a vertex. In 3D the loops (rows of **R)** are typically around an edge (if the divergence is defined by a sum over cell faces) and pass through each face touching that edge.

Applying the 'rotation' matrix gives the exact fractional step method,

$$\mathbf{RAC}s^{n+1} = \mathbf{R}(r^n + bc_1) \qquad (6)$$

The matrix $\mathbf{RAC}$ is often symmetric (if **A** is), and is always positive definite. The system given by Eqn. (6) is both easier to solve numerically that Eqn. (2) and contains far fewer unknowns. For 3D unstructured staggered mesh methods the reduction in unknowns from Eqn (2) is roughly $\frac{6}{15} = 40\%$. In 2D it is $\frac{1}{5} = 20\%$. On Cartesian meshes the reduction is somewhat less (only 75% for 3D grids and 33% for 2D grids).

However, the real savings come from the reduced number of iterations necessary for Eqn (6) as compared to the original system (Eqn 2) or the classical fractional step method (essentially Eqn 4). Iterative solvers for Eqn (6) can be terminated when the iteration error is roughly equal to the truncation error of the discretization scheme, so very few iterations are required. No matter how large the error in the solution for s^{n+1}, the resulting velocity field given by $u^{n+1} = \mathbf{C}s^{n+1}$ is always divergence free to machine precision. In contrast, truncation of the iteration procedure in the Poisson equation for the pressure in the classical fractional step method (or iteration errors from an interative solution of the full matrix system, Eqn 2) lead to large dilatation errors (mass creation and destruction) that are very detrimental to the total solution accuracy.

While the matrix $\mathbf{RAC}$ could be formed explicitly this is not recommended for unstructured mesh solutions because the resulting stencil becomes larger than nearest neighbor. In many iterative solvers, explicit matrix construction is not required, only the matrix vector product needs to be computed.

4. RESULTS

The affect of the splitting error that results from the classical fractional step method is assessed in figure 2. In this test we study the temporal evolution of 2D lid driven cavity flow at a Reynolds number of 100 on an unstructured mesh containing 4024 triangles. The convection term is advanced using fully explicit 2nd order Adams-Bashforth, the diffusion term uses 2nd order trapezoidal time advancement. The splitting error of the classical fractional step method causes this nominally second order scheme to be first order accurate, whereas the exact fractional step method introduces no splitting error.

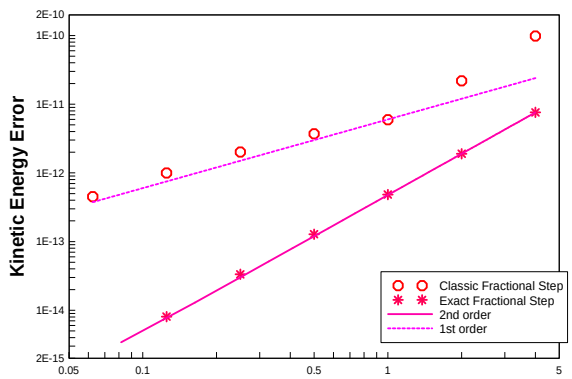

Figure 2. Effect of splitting error on the time accuracy of the classical fractional step approach.

For lower Re number simulations the first order splitting error is larger and the difference between the two solutions increasingly pronounced. When implicit convection is used, the splitting error is always large. The results in figure 2 required very stringent tolerances (many iterations) on the Poisson solver in order to remove the equally detrimental effects of dilatation errors (and erratic convergence). The tolerances on the exact fractional step method where able to be much looser (requiring far fewer iterations).

The effect of iteration errors are shown in figure 3. In this case the error is measured during the CG iteration process. The 'exact' solution is the solution after many iterations, so the reported error does not include the splitting error shown in figure 2, just the iteration error. Figure 3, shows that dilation errors (found in the classical fractional step method) are far more difficult for the CG solver to remove than the purely 'vortical' errors (found in the exact fractional step method). For a given accuracy level the exact method requires far fewer CG iterations (on the order of a ¼), and therefore less computational time.

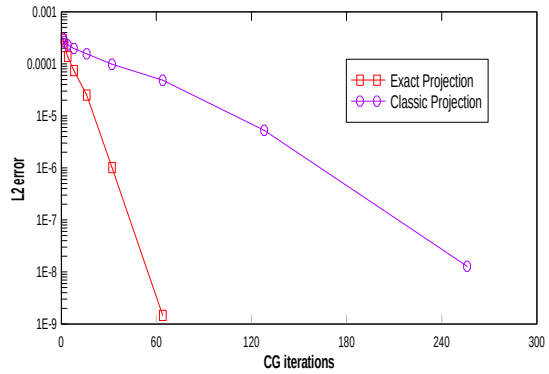

Figure 3. Effect of iteration error on the classical fractional step and exact fractional step methods.

The exact fractional step method is not more complex to implement that the classical fractional step method. Figure 4a-c shows a fairly complex 3D unsteady, two phase flow simulation of droplet collision using a moving unstructured mesh and the exact fractional step method.

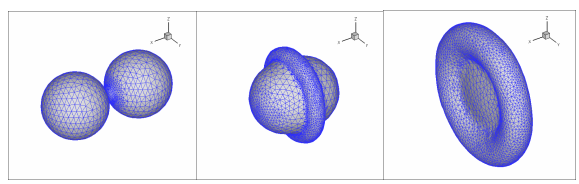

Figure 4. Simulations of droplet collision using the exact fractional step method.

ACKNOWLEDGEMENTS

Partial financial support for this work was provided by the Office of Naval Research (Grant N00014-01-1-0267), the Air Force Office of Scientific Research (Grant FA9550-04-1-0023) and the National Science Foundation (Grant CTS-0522089).

REFERENCES

[1] J. B. Perot, An analysis of the fractional step method, *J. Computational Physics*, **108**:51-58, 1993.

[2] J. B. Perot, Conservation properties of unstructured staggered mesh schemes, *J. Computational Physics*, **159**:58-89, 2000.

[3] V. Subramanian & J. B. Perot, Higher-Order Mimetic Methods for Unstructured Meshes, *Journal of Computational Physics*, **219** (1): 68-85, 2006.