Iterated Pressure-Correction Projection Methods for
the Unsteady Incompressible Navier-Stokes Equations

Jean Aoussou*, Jing Lin*, Pierre F. J. Lermusiaux**

Department of Mechanical Engineering, Massachusetts Institute of Technology
77 Massachusetts Avenue, Cambridge, MA 02139, U.S.

Abstract

Iterated pressure-correction projection schemes for the unsteady incompressible Navier-
Stokes equations are developed, analyzed and exemplified, in relation to preconditioned
iterative methods and the pressure-Schur complement equation. Typical pressure-
correction schemes perform only one iteration per stage or time step, and suffer from
splitting errors that result in spurious numerical boundary layers and a limited order
of convergence in time. We show that performing iterations not only reduces the ef-
facts of the splitting errors, but can also be more efficient computationally than merely
reducing the time step. We devise stopping criteria to recover the desired order of tem-
poral convergence, and to drive the splitting error below the time-integration error.
We also develop and implement the iterated pressure corrections with both multi-step
and multi-stage time integration schemes. Finally, to reduce further the computational
cost of the iterated approach, we combine it with an Aitken acceleration scheme. Our
theoretical results are validated and illustrated by numerical test cases for the Stokes
and Navier-Stokes equations, using implicit-explicit (IMEX) backwards differences and
Runge-Kutta time-integration solvers. The test cases comprise a now classical man-
ufactured solution in the projection method community and a modified version of a
more recently proposed manufactured solution. The different error types, stopping
criterion, recovered orders of convergence, and acceleration rates are illustrated, as
well as the effects of the rotational corrections and time-integration schemes. It is

*JA and JL are co-first authors.
**Corresponding author:
Email address: pierrel@mit.edu (Pierre F. J. Lermusiaux)
found that iterated pressure-correction schemes can retrieve the accuracy and temporal convergence order of fully-coupled schemes and are computationally more efficient than classic pressure-correction schemes.

**Keywords:** Incompressible Navier-Stokes, projection methods, pressure-correction schemes, preconditioned iterative methods, high-order time-marching, IMEX schemes

### 1. Introduction

Solving the unsteady fully coupled incompressible Navier-Stokes equations is computationally demanding [40, 59, 26, 62]. The projection method introduced by [14] and [56], which can decouple the momentum and continuity equations numerically by replacing the latter with a Poisson equation for the pressure, is often a preferred option. A non divergence-free velocity field is first obtained by omitting the pressure term in the momentum equation, and is then corrected using the pressure Poisson equation (PPE). This method was reinterpreted as a fractional-step scheme by [32] and a number of variations on the original scheme have been developed, a review of which can be found in [24]. In this study, only the incremental pressure correction schemes are considered. This time splitting decouples the computation of velocity and pressure in the sense that instead of solving a large linear system for all velocity components and pressure simultaneously, it only solves \((d+1)\) smaller linear systems for each velocity component and pressure (or pressure correction) separately [62], where \(d\) is the spatial dimension of the flow. Because the cost of solving a linear system typically scales worse than linear, such breakdown is beneficial. Moreover, the decoupled linear systems all have the same structure as a discrete Laplacian, which is symmetric positive definite (SPD) and can be solved efficiently with many mature techniques. In contrast, the coupled system yields a saddle-point problem (symmetric indefinite) and is worse-conditioned because it involves both the momentum equations and the continuity equation, which behave very differently. However, this gain for time splitting comes at the price