Power upgrades and plant life extension

Engineering design and analysis

Science-enabling high performance computing

Fundamental science

Plant operational data

L3:THM.CFD.P5.01
Mark Christon
LANL
Completed: 9/29/2012
Notes on Newton-Krylov based Incompressible Flow Projection Solver

R.R. Nourgaliev\textsuperscript{a}, M.Christon\textsuperscript{b}, J.Bakosi\textsuperscript{c}

\textsuperscript{a} Idaho National Laboratory, Nuclear Science and Technology Division, Thermal Science & Safety Analysis Department, 2525 North Fremont Ave. Idaho Falls, ID 83415-3860
Email: Robert.Nourgaliev@inl.gov

\textsuperscript{b} Los Alamos National Laboratory, Computer, Computational and Statistical Sciences Division, Computational Physics Group (CCS-2)) Los Alamos, NM 87545
Email: christon@lanl.gov

\textsuperscript{c} Los Alamos National Laboratory, Computer, Computational and Statistical Sciences Division, Computational Physics Group (CCS-2)) Los Alamos, NM 87545
Email: jbakosi@lanl.gov

Abstract

The purpose of the present document is to formulate Jacobian-free Newton-Krylov algorithm for approximate projection method used in Hydra-TH code. Hydra-TH is developed by Los Alamos National Laboratory (LANL) under the auspices of the Consortium for Advanced Simulation of Light-Water Reactors (CASL) for thermal-hydraulics applications ranging from grid-to-rod fretting (GTRF) to multiphase flow subcooled boiling. Currently, Hydra-TH is based on the semi-implicit projection method, which provides an excellent platform for simulation of transient single-phase thermal-hydraulics problems. This algorithm however is not efficient when applied for very slow or steady-state problems, as well as for highly non-linear multiphase problems relevant to nuclear reactor thermalhydraulics with boiling and condensation. These applications require fully-implicit tightly-coupling algorithms. The major technical contribution of the present report is the formulation of fully-implicit projection algorithm which will fulfill this purpose. This includes the definition of non-linear residuals used for GMRES-based linear iterations, as well as physics-based preconditioning techniques.

Key words: Multi-Physics, Multi-Scale problems, Finite Volume, Incompressible Flow, Approximate Projection, Jacobian-Free Newton Krylov (JFNK), Physics-Based Preconditioning

INL/EXT-12-27197 September 20, 2012
1. Introduction

The solution of the time-dependent incompressible single- and multi-phase flows poses several algorithmic problems due to the div-free constraint, and the concomitant spatial and temporal resolution required to perform time-accurate solutions particularly when complex geometry is involved. The initial deployment of Hydra-TH has focused on projection methods because of their computational efficiency and accuracy for transient flows. However, when applied to slow transients and steady-state problems, the currently existing projection methods are not cost-effective, due to stability restrictions imposed by material Courant limit. For these applications, fully-implicit algorithms are required. Here, we reformulate semi-implicit projection method to fit into the fully-implicit Jacobian-free Newton Krylov solution strategy.

We start with a short description of governing equations, defined in Section 3. Even though we limit our discussion here to single-phase flows, the basic ideas introduced are extendable to multi-fluid formulation [1].

A detailed review of projection methods is beyond the scope of this document, but a partial list of relevant work is provided for the interested reader. Projection methods, also commonly referred to as fractional-step, pressure correction methods, or Chorin’s method [2] have grown in popularity over the past 20 years due to the relative ease of implementation and computational performance. This is reflected by the volume of work published on the development of second-order accurate projection methods, see for example van Kan [3], Bell, et al. [4], Gresho, et al. [5, 6, 7, 8], Almgren, et al. [9, 10, 11], Rider [12, 13, 14, 15], Minion [16], Guermond and Quartapelle [17], Puckett, et al. [18], Sussman, et al. [19], and Knio, et al. [20]. The numerical performance of projection methods has been considered by Brown and Minion [21, 22], Wetton [23], Guermond [24, 25], Guermond and Quartapelle [26, 27], and Almgren et al. [11]. A short introduction to semi-implicit projection method is given in Section 4.

The main technical contribution of this report is described in sections 5 and 6, introducing fully-implicit projection and its physics-based preconditioning.

Concluding remarks are given in the final section 7.
2. HydraTH

Hydra-TH [28] refers to the specific physics module that provides the hybrid finite-volume/finite-element incompressible/low-Mach flow solver. This is built as one of the many physics modules using the Hydra multiphysics toolkit. The toolkit provides a rich suite of components that permits rapid application development, I/O interfaces to permit reading/writing multiple file formats for meshes, plot data, time-history and surface-based output. The toolkit also provides run-time parallel domain decomposition with data-migration for both static and dynamic load-balancing. Linear algebra is handled through an abstract interface that permits use of popular libraries such as PetSC and Trilinos. Hydra’s toolkit model for development provides lightweight, high-performance and reusable code components for agile development. Currently the toolkit supports finite-element based solvers for time-dependent heat conduction, time-dependent advection-diffusion, time-dependent incompressible flow, multiple Lagrangian hydrodynamics solvers, rigid-body dynamics, etc. In addition, unstructured-grid finite-volume solvers are available for solving time-dependent advection-diffusion, Burgers’ equation, the compressible Euler equations, and incompressible/low-Mach Navier-Stokes equations. There are also interfaces to the FronTier front-tracking software and to level-set methods.

3. Governing Equations

In the following discussion, we allow variable-density formulation. The mass conservation principle in divergence form is

\[ \frac{\partial \rho}{\partial t} + \frac{\partial (\rho v_j)}{\partial x_j} = 0. \]  

(1)

In the incompressible limit, the velocity field is solenoidal,

\[ \frac{\partial v_i}{\partial x_i} = 0 \]  

(2)

which implies a mass density transport equation,

\[ \frac{\partial \rho}{\partial t} + v_j \frac{\partial \rho}{\partial x_j} = 0. \]  

(3)

For constant density, Eq. (2) is neglected with Eq. (3) remaining as a constraint on the velocity field.
Momentum conservation. The conservation of linear momentum is
\[
\rho \frac{\partial v_i}{\partial t} + \rho v_j \frac{\partial v_i}{\partial x_j} = \frac{\partial \sigma_{ij}}{\partial x_j} + \rho f_i \tag{4}
\]
where \(v_i\) is the velocity, \(\sigma_{ij}\) is the stress tensor, \(\rho\) is the mass density, and \(f_i\) is the body force. The body force contribution \(\rho f_i\) typically accounts for buoyancy forces with \(f_i\) representing the acceleration due to gravity.

The stress may be written in terms of the fluid pressure and the deviatoric stress tensor as
\[
\sigma_{ij} = -p \delta_{ij} + \tau_{ij} \tag{5}
\]
where \(p\) is the pressure, \(\delta_{ij}\) is the Kronecker delta, and \(\tau_{ij}\) is the deviatoric stress tensor. A constitutive equation relates the deviatoric stress and the strain rate, e.g.,
\[
\tau_{ij} = 2\mu S_{ij} \tag{6}
\]
The strain-rate tensor is written in terms of the velocity gradients as
\[
S_{ij} = \frac{1}{2} \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right). \tag{7}
\]

Energy conservation. The energy equation may be expressed in terms of temperature, \(T\), as
\[
\frac{\partial \rho C_p T}{\partial t} + \frac{\partial}{\partial x_j} \left( \rho v_j C_p T \right) = -\frac{\partial q_i}{\partial x_j} + q'''' \tag{8}
\]
where \(C_p\) is the specific heat at constant pressure, \(q_i\) is the diffusional heat flux rate, and \(q''''\) represents volumetric heat sources and sinks, e.g., due to exothermic/endothermic chemical reactions. Fourier’s law relates the heat flux rate to the temperature gradient and thermal conductivity
\[
q_i = -\kappa \frac{\partial T}{\partial x_i} \tag{9}
\]
where \(\kappa\) is the thermal conductivity.

Alternatively, one can solve in terms of specific internal energy:
\[
\frac{\partial \rho u}{\partial t} + \frac{\partial}{\partial x_j} \left( \rho v_j u \right) = -\frac{\partial q_i}{\partial x_j} + q'''' \tag{10}
\]
with a given function
\[
u = \mathcal{F}(T)
\]
For example, 

\[ u(T) = u_0 + C_v (T - T_0) \]

where \( u_0 \) and \( T_0 \) are the values of specific internal energy and temperature at some reference point, while \( C_v \) is specific heat.

**Scalar transport.** In addition, we consider a coupled solution for transport of scalars:

\[
\frac{\partial \rho \phi_n}{\partial t} + \frac{\partial}{\partial x_j} \left( \rho v_j \phi_n \right) = - \frac{\partial}{\partial x_j} J_{nj} + J''_{nj} \tag{11}
\]

where by \( J_{nj} \) and \( J''_{nj} \) we denote diffusive flux and volumetric sources for a scalar \( \phi_n \). Note that \( \phi_n \) could represent turbulence transport quantities (e.g., turbulent kinetic energy \( k \)). In this case, momentum and heat diffusion coefficients are considered to be a function of \( \phi_n \). In the most general case, 

\[
\mu(T, \phi_n) \quad \text{and} \quad \kappa(T, \phi_n), \quad n = 0, ..., N - 1
\]

4. **Semi-Implicit Projection**

Following the well-established finite-volume procedure, we discretize momentum equation in space, integrate by parts, and apply the divergence theorem. Using a piecewise-constant weight functions yields

\[
\rho \frac{d}{dt} \int_{\Omega^e} \mathbf{v} \ dt + \oint_{\Gamma^e} \rho \mathbf{v} (\mathbf{v} \cdot \mathbf{n}) \ dt - \oint_{\Gamma^e} \mathbf{\tau} \cdot \mathbf{n} \ dt + \int_{\Omega^e} \nabla p \ dt - \int_{\Omega^e} f \ dt \tag{12}
\]

Using definition for the cell-average,

\[
\bar{u} = \frac{1}{\Omega^e} \int_{\Omega^e} u^h \tag{13}
\]

the spatially-discrete momentum equations become

\[
\rho \bar{\Omega}^e \frac{d \bar{\mathbf{v}}^e}{dt} + \oint_{\Gamma^e} \rho \mathbf{v} (\mathbf{v} \cdot \mathbf{n}) \ dt - \oint_{\Gamma^e} \mathbf{\tau} \cdot \mathbf{n} \ dt + \int_{\Omega^e} \nabla p \ dt - \int_{\Omega^e} f \ dt \tag{14}
\]

The projection algorithm can be derived a number of ways. Here, we choose to first develop the time-integrator, and identify the terms associated with the projection via a Helmholtz decomposition of the velocity.
proceeding we define the following mass, advective, viscous, gradient and body-force operators.

\[ M = \rho \Omega^e \]  
(15)

\[ A(\rho, v) = \oint_{\Gamma^e} \rho v (v \cdot n) \, d\Gamma^e \]  
(16)

\[ K = \oint_{\Gamma^e} \tau \cdot n \, d\Gamma^e \]  
(17)

\[ B \bar{p} = \int_{\Omega^e} \nabla p \, d\Omega^e \]  
(18)

\[ F = \int_{\Omega^e} f \, d\Omega^e \]  
(19)

We form the global operators, apply forward-Euler first, then backward-Euler with explicit advection in both cases, and take the sum of the fully-discrete systems results in the following

\[
M \frac{\nabla^{n+1} - \nabla^n}{\Delta t} = (1 - \theta) K \nabla^n + \theta K \nabla^{n+1} + (1 - \theta) F^n + \theta F^{n+1} - (1 - \theta) A(\rho, v) \nabla^n - \theta A(\rho, v) \nabla^{n+1} - B \bar{p}^n - \theta \bar{B} (\bar{p}^{n+1} - \bar{p}^n)
\]  
(20)

where \(0 \leq \theta \leq 1\), \(\theta = 0\) corresponds to a forward-Euler, \(\theta = 1/2\) a trapezoidal rule, and \(\theta = 1\) backward-Euler treatment of viscous and body-force terms.

Using the Helmholtz decomposition as

\[ \rho v^n = \rho v^{n+1} + \nabla \lambda \]  
(21)

we introduce the following definition

\[ \lambda = \theta \rho \Delta t (\bar{p}^{n+1} - \bar{p}^n) \]  
(22)

Plugging these into Eq. (20), the momentum equation can be formulated for the approximate ("predictor") velocity as

\[
[M - \theta \Delta t (K - A(\rho, v))] \nabla^* = [M + (1 - \theta) \Delta t (K - A(\rho, v))] \nabla^n + \Delta t ((1 - \theta) F^n + \theta F^{n+1} - B \bar{p}^n) + \theta \Delta t A(\rho, v) \left[ \frac{\nabla^0}{\rho} - \theta \bar{K} \frac{\nabla^0}{\rho} + \left[ M \frac{\nabla}{\rho} - \bar{B} \lambda \right]^0 \right]
\]  
(23)
Using the Helmholtz decomposition, and requiring $\nabla \nabla v^{n+1} = 0$, yields a pressure-Poisson equation (PPE) that can be solved for the Lagrange multiplier $\lambda$:

$$\nabla \cdot \frac{1}{\rho} \nabla \lambda = \nabla \cdot \mathbf{v}^*$$

(24)

Given a velocity and pressure at time-level $n$, the P2 algorithm proceeds as follows.

Algorithm 1. Basic P2 Algorithm

1. Solve for $\mathbf{v}^*$

$$[M - \theta \Delta t (K - A(\rho, \mathbf{v}))] \mathbf{v}^* = [M + (1 - \theta) \Delta t (K - A(\rho, \mathbf{v}))] \mathbf{v}^n + \Delta t \left((1 - \theta) \mathbf{F}^n + \theta \mathbf{F}^{n+1} - B\bar{p}^n\right)$$

(25)

2. Form the right-hand-side of the PPE, solve for $\lambda$,

$$K_p \lambda = D$$

(26)

3. Update the pressure

$$\bar{p}^{n+1} = \bar{p}^n + \frac{1}{\theta_p \Delta t} \lambda$$

(27)

Note that testing over the last 20 years or so has indicated that using $\theta_p = 1/2$ to update the pressure can lead to temporal oscillations in the pressure. For this reason, we use $\theta_p = 1$ in the implementation.

4. Project the cell-centered velocities

$$\mathbf{v}^{n+1} = \mathbf{v}^* - \frac{1}{\rho} \mathbf{B} \lambda$$

(28)

5. Compute face gradients and project the face-centered velocities

$$v_f = v_f^* - \frac{1}{\rho_f} ((B) \lambda)_f \cdot \mathbf{n}$$

(29)

6. Repeat steps 1 - 5 until the termination time is reached.
5. Fully-Implicit Projection

For the sake of simplicity, consider isothermal constant-density flow. Thus, the vector of unknowns is

\[ \mathbf{U} = \begin{bmatrix} \bar{p} \\ \bar{v} \end{bmatrix} \]

or, in terms of Lagrange multiplier:

\[ \mathbf{V} = \begin{bmatrix} \lambda \\ \bar{v} \end{bmatrix} \]

Let us search a new-time solution iteratively, defining new-iteration values \( \mathbf{U}^{n+} \) or \( \mathbf{V}^{n+} \) in the following incremental form:

\[ \bar{p}^{n+} = \bar{p}^{n} + p' \]  \hspace{1cm} (30)

\[ \lambda^{n+} = \lambda^{n} + \lambda' \]  \hspace{1cm} (31)

\[ \bar{v}^{n+} = \bar{v}^{n} + \bar{v}' \]  \hspace{1cm} (32)

and assume the following linearization of body force:

\[ \mathbf{F}^{n+} = \mathbf{F}^{n} + \mathbf{F}_{v} \left( \mathbf{F}^{n} \right) \bar{v}' + \mathbf{f}_{p} \left( \mathbf{F}^{n} \right) p' \]  \hspace{1cm} (33)

where specific forms of the linearization matrix \( \mathbf{F}_{v} \) and vector \( \mathbf{f}_{p} \) are problem-dependent.

Plug these into eqs.(21), (22) and (20):

\[ \bar{v}' = \frac{1}{\rho} \nabla \left( \lambda^{n} + \lambda' \right) \]  \hspace{1cm} (34)

\[ \lambda^{n} + \lambda' = \theta_{p} \Delta t \left( \bar{p}^{n} - \bar{p}^{n} \right) + \theta_{\rho} \Delta t p' \]  \hspace{1cm} (35)

\[ M \left( \bar{v}^{n} + \bar{v}' - \bar{v}^{n} \right) = \Delta t (1 - \theta) K \nabla \bar{n} + \Delta t \theta K \left( \bar{v}^{n} + \bar{v}' \right) - \Delta t (1 - \theta) A(\rho, \mathbf{v}) \nabla \bar{n} - \Delta t \theta A(\rho, \mathbf{v}) \left( \bar{v}^{n} + \bar{v}' \right) + \Delta t (1 - \theta) \mathbf{F}^{n} + \Delta t \theta \left( \mathbf{F}^{n} + \mathbf{F}_{v} \bar{v}' + \mathbf{f}_{p} p' \right) - \Delta t \theta p^{n} - \Delta t \theta B(\bar{p}^{n} + p' - \bar{p}^{n}) \]  \hspace{1cm} (36)
After re-grouping, the momentum equation becomes:

\[
\left[ \Delta t \theta M^{-1} \left( f_p - \frac{\theta}{\rho} B \right) \right] p' + \left[ 1 - \Delta t \theta M^{-1} \left( K - A(\rho, v) + F_v \right) \right] v' = -\text{res}_v
\]

(37)

where

\[
\text{res}_v = \nabla \theta - \nabla - \Delta t \theta \left( (1 - \theta)K \nabla^n + \theta K \nabla^\theta \right) + + \Delta t M^{-1} \left( (1 - \theta)A(\rho, v)\nabla^n + \theta A(\rho, v)\nabla^\theta \right) - - \Delta t M^{-1} \left( (1 - \theta)F^n + \theta F^\theta \right) + + \Delta t M^{-1} \left( B\nabla^n + \theta B(\nabla^\theta - \nabla^n) \right)
\]

(38)

From eq.(34),

\[
\nabla^\theta + v' = \nabla - \frac{\theta}{\rho} \Delta t \nabla \left( \nabla^\theta + p' - \nabla^n \right)
\]

(39)

which can be plugged into eq.(36) to get a counterpart of the momentum “predictor” equation (23):

\[
\begin{align*}
[M - \Delta t \theta (K - A(\rho, v) + F_v)] \nabla^v &= [M + \Delta t(1 - \theta) (K - A(\rho, v))] \nabla^n - \Box \frac{\Delta t \theta}{\rho} \nabla \left( \nabla^\theta + p' - \nabla^n \right) \\
&+ \Box \frac{\Delta t \theta}{\rho} A(\rho, v) \left( \nabla^\theta + p' - \nabla^n \right) \\
&+ \Delta t(1 - \theta)F^n + \Delta t \theta \left( F^\theta - F_v \left( \frac{\Delta t \theta}{\rho} \nabla \left( \nabla^\theta - \nabla^n \right) \right) - \Delta t \theta \left( \nabla - \frac{\Delta t \theta}{\rho} \nabla \nabla^n \right) \right) - \\
&- \Delta t B\nabla^n - \left( \frac{\Delta t \theta}{\rho} B(\nabla^\theta - \nabla^n) - \frac{\Delta t \theta}{\rho} \nabla \left( \nabla^\theta + p' - \nabla^n \right) \right)
\end{align*}
\]

(40)

Note that the terms in boxes are dropped in eq.(25). Next, we further re-group this equation as:

\[
\nabla^\theta v' = m^\theta - \Phi p'
\]

(41)
where
\[
\mathbf{m}^\circ = [M + \Delta t(1 - \theta)(K - A(\rho, \mathbf{v}))] \nabla \mathbf{v}^\circ + \\
\Delta t(1 - \theta)\mathbf{F}^\circ + \Delta t\theta \left( \frac{\mathbf{F}^\circ}{\rho} - \mathbf{F}_\rho \left( \frac{1}{\rho} \nabla \mathbf{v}^\circ \right) \right) - \Delta t\mathbf{B}\mathbf{p}^\circ - \\
- \Delta t\theta[K - A(\rho, \mathbf{v})] \left( \frac{1}{\rho} \nabla \mathbf{v}^\circ \right) - \mathbf{B}^\circ + \frac{1}{\rho} \nabla \mathbf{v}^\circ
\]
(42)

\[
\mathcal{G} \equiv \Delta t\theta \left[ \frac{\mathbf{F}_\rho}{\rho} \frac{\Delta t\theta}{\rho} \nabla \mathbf{f}_\rho + \frac{\theta}{\theta} \mathbf{B} - \frac{1}{\theta} \frac{\Delta t\theta}{\rho} \nabla \mathbf{f} + (K - A(\rho, \mathbf{v})) \frac{\Delta t\theta}{\rho} \nabla \mathbf{p} \right]
\]
(43)

\[
\mathcal{P} \equiv [M - \Delta t\theta(K - A(\rho, \mathbf{v}) + \mathbf{F}_\rho)]
\]
(44)

and we used equation (35) for definition of \( \lambda^\circ \). Thus, “predictor” velocity can be computed as
\[
\mathbf{v}' = \mathcal{P}^{-1} \left( \mathbf{m}^\circ - \mathcal{G} \mathbf{p}' \right)
\]
(45)

Finally, we can form **incremental PPE** by taking divergence of eq.(34) and using eq.(45):
\[
\nabla \cdot \frac{1}{\rho} \nabla \lambda' + \frac{1}{\theta^\prime \Delta t} \nabla \cdot \mathcal{G} \mathbf{p}' = -\text{res}_\lambda
\]
(46)

where
\[
\text{res}_\lambda = \nabla \cdot \frac{1}{\rho} \nabla \mathbf{v}^\circ - \nabla \cdot \left( \mathcal{P}^{-1} \mathbf{m}^\circ \right)
\]
(47)

With this, linear iterations of a Newton-based algorithm are defined by the following equation:

\[
\begin{bmatrix}
\nabla \cdot \frac{1}{\rho} \nabla + \frac{1}{\rho} \nabla \cdot \mathcal{G} & 0 \\
\frac{\theta}{\theta} M^{-1} \left( \mathbf{f}_\rho - \frac{\theta}{\theta} \mathbf{B} \right) & 1 - \Delta t\theta M^{-1}(K - A(\rho, \mathbf{v}) + \mathbf{F}_\rho)
\end{bmatrix}
\begin{bmatrix}
\mathbf{v}' \\
\mathbf{\lambda}'
\end{bmatrix}
= -\begin{bmatrix}
\text{res}_\lambda \\
\text{res}_\mathbf{v}
\end{bmatrix}
\]
(48)

Jacobian, \( J_{\mathbf{v}} \)

Non-linear residuals \( \text{res}_\lambda \) and \( \text{res}_\mathbf{v} \) are supplied to PETSC-SNES [29] for JFNK implementation.
6. Preconditioning

6.1. General strategy

Consider the following modification of eq.(48):

\[
\begin{align*}
\mathcal{J}_\nu \mathcal{P}^{-1} \mathcal{P} \mathcal{V}' = -r \partial \mathcal{V}' \\
\mathcal{J}_\nu \mathcal{P}^{-1} \mathcal{V}'' = \vec{r} \mathcal{P} \mathcal{V}'
\end{align*}
\] (49)

where \( \mathcal{P} \) symbolically represents the preconditioning matrix (or process), and \( \mathcal{P}^{-1} \) is its inverse. Thus, the solution procedure is splitted into two processes:

1. Solving for

\[
\mathcal{J}_p \mathcal{V}'' = \vec{b}
\] (50)

(this is what actually crunched by GMRES), and

2. Preconditioning:

\[
\mathcal{V}' = \mathcal{P}^{-1} \mathcal{V}''
\] (51)

While one refers to the matrix/process \( \mathcal{P} \), operationally the algorithm only requires the action of \( \mathcal{P}^{-1} \) on a vector. The main requirement is that \( \mathcal{P} \) designed properly, to enable clustering eigenvalues of the \( \mathcal{J}_p \), making the solution of eq.(50) to converge faster.

For effective preconditioning of the fully-implicit projection algorithm, we can use semi-implicit algorithm described in Section 4. The strategy with involving a legacy (e.g., operator-splitting) algorithm for preconditioning is commonly referred to as Physics-(Process)-based preconditioning (PBP) [30, 31, 32, 33, 34], to be contrasted to the Matrix-(Math)-based preconditioning (MBP) algorithms. The later include different flavors of SOR, SSOR, ILU, MILU, ILUT, ILUTP, ILUS, ILUC, etc. preconditioners, see [35] for review. In these cases, the preconditioning matrix \( \mathcal{P} \) is required, as a suitable approximation for \( \mathcal{J}_\nu \).

In the following section, we will describe details of our implementation of the semi-implicit projection as PBP, emphasizing all differences relative to the using this algorithm as a solver (in an operator-splitting OS mode).
6.2. Semi-Implicit Projection as Physics-Based Preconditioning

At the input of the preconditioning step, we have current Newton iteration values of \( \mathbf{v}^\circ, \bar{p}^\circ \) and \( \lambda^\circ \), and current update values \( \mathbf{v}'', \bar{p}'' \) and \( \lambda'' \). In the OS splitting mode, these are:

\[
\mathbf{v}^\circ = \mathbf{v}_n, \quad \bar{p}^\circ = \bar{p}_n, \quad \lambda^\circ = 0, \quad \mathbf{v}'' = 0, \quad \bar{p}'' = 0 \quad \text{and} \quad \lambda'' = 0
\]

The task of the preconditioning is to convert these into \( \mathbf{v}^\diamond, \bar{p}^\diamond, \lambda^\diamond, \mathbf{v}', \bar{p}' \) and \( \lambda' \), where

\[
\begin{align*}
\mathbf{v}^\diamond &= \mathbf{v}^\circ + \mathbf{v}' \\
\bar{p}^\diamond &= \bar{p}^\circ + \bar{p}' \\
\lambda^\diamond &= \lambda^\circ + \lambda'
\end{align*}
\]  (52)

In the OS mode, \( \Phi^\diamond = \Phi_{n+1} \), where \( \Phi = \mathbf{v}, \bar{p} \) and \( \lambda \).

We define Helmholtz decomposition as

\[
\mathbf{v}^\diamond,\star = \mathbf{v}^\diamond + \nabla \left( \frac{1}{\rho} \left( \lambda^\diamond + \lambda' \right) \right)
\]

Divergence-free part, \( \mathbf{v}^\diamond \)

Non-incremental Form.

1. The first step would be to solve for non-solenoidal ("predictor") velocity field, \( \mathbf{v}^\diamond,\star \), using one of the following options.

Option-A:

\[
[M - \Delta t \theta (K - A(\rho, \mathbf{v}) + \mathbf{F}_v)] \mathbf{v}^\diamond,\star = \\
= [M + \Delta t (1 - \theta) (K - A(\rho, \mathbf{v}))] \mathbf{v}_n - \frac{\Delta t \theta}{\rho} \nabla \left( \frac{\Delta t \theta p}{\rho} \left( \bar{p}^\circ + p'' - \bar{p}_n \right) \right) + \\
+ \frac{\Delta t \theta A(\rho, \mathbf{v})}{\rho} \left( \frac{\Delta t \theta p}{\rho} \left( \bar{p}^\circ + p'' - \bar{p}_n \right) \right) + \\
+ \Delta t (1 - \theta) \mathbf{F}_n + \Delta t \theta \left( \mathbf{F}_v - \mathbf{F}_v \left( \frac{\Delta t \theta p}{\rho} \left( \bar{p}^\circ - \bar{p}_n \right) + \mathbf{v}^\circ \right) \right) - \\
- \Delta t \theta \left[ \mathbf{F}_v \frac{\Delta t \theta p}{\rho} \nabla - \mathbf{F}_p \right] p'' - \\
- \Delta t \mathbf{B} \bar{p}'' - \left( \frac{\Delta t \theta p}{\rho} \mathbf{B} \left( \bar{p}^\circ + p'' - \bar{p}_n \right) - \frac{\Delta t \theta p}{\rho} \nabla \left( \bar{p}^\circ + p'' - \bar{p}_n \right) \right)
\]  (54)
In the OS mode, $\nabla^{\cap,*} = \nabla^*$ and eq.(54) reduces to eq.(25).

Option-B:

Equation (54) is of advection-diffusion type, which is not well amenable to multigrid algorithm, and solved in Hydra by ILU-based solver. Another viable option would be to convert it into the parabolic equation, by taking out advection operator on the left-hand-side (leaving it to GMRES to deal with). Thus, the parabolic equation would be:

$$[M - \Delta t \theta (K + F_v)] \nabla^{\cap,*} = [M + \Delta t(1 - \theta) (K - A(\rho, v))] \nabla^n -$$

$$-\Delta t A(\rho, v) \left( \nabla^{\cap} + \nabla'' \right) - \Delta t \theta K \left( \frac{\Delta t \theta}{\rho} \nabla \left( \bar{p}^{\cap} + p'' - \bar{p}^n \right) \right) +$$

$$+ \Delta t A(\rho, v) \left( \frac{\Delta t \theta}{\rho} \nabla \left( \bar{p}^{\cap} + p'' - \bar{p}^n \right) \right) +$$

$$+ \Delta t (1 - \theta) F^n + \Delta t \theta \left( F^{\cap} - F_v \left( \frac{\Delta t \theta}{\rho} \nabla \left( \bar{p}^{\cap} - \bar{p}'' \right) + \nabla^{\cap} \right) \right) -$$

$$- \Delta t \theta \left[ F_v \frac{\Delta t \theta}{\rho} \nabla - f_{p''} \right] p'' -$$

$$- \Delta t B \bar{p}^n - \left( \Delta t \theta \left[ B(\bar{p}^{\cap} + p'' - \bar{p}^n) - \frac{\Delta t \theta}{\rho} \nabla \left( \bar{p}^{\cap} + p'' - \bar{p}^n \right) \right) \right) \right]$$

(55)

2. The second step would be to form and solve PPE. Taking divergence of eq.(53) leads to the following PPE:

$$\nabla \cdot \nabla^{\cap,*} = \nabla \cdot \nabla^{\cap,*}$$

(56)

which reduces to eq.(26) in the OS mode.

3. Pressure is computed from the new Lagrange multiplier as:

$$\bar{p}^{\cap} = \bar{p}^n + \frac{1}{\theta \Delta t} \lambda^{\cap}$$

(57)

4. Next, we project the cell-centered velocities as

$$\nabla^{\cap} = \nabla^{\cap,*} - \frac{1}{\rho} B \lambda^{\cap}$$

(58)
5. Finally, we can compute

\[
\begin{aligned}
\mathbf{v}' &= \mathbf{v}^\circ - \mathbf{v}^\circ \\
\bar{p}' &= \bar{p}^\circ - \bar{p}^\circ \\
\lambda' &= \lambda^\circ - \lambda^\circ
\end{aligned}
\]  

and these are the values which are returned to PETSC-SNES. As mentioned above, in the OS mode, this step is absent, as \( \Phi^\circ = \Phi^{n+1} \).

**Incremental Form.** One can re-write eq.(53) as:

\[
\mathbf{v}^{\circ,v,*} = \mathbf{v}^\circ + \mathbf{v}^{*,v}
\]  

where

\[
\mathbf{v}^{*,v} = \mathbf{v}' + \frac{1}{\rho} \nabla \left( \lambda^\circ + \lambda' \right)
\]  

1. Solve for non-solenoidal velocity increment \( \mathbf{v}^{*,v} \).

**Option-A:**

\[
\begin{aligned}
&[M - \Delta t \theta (K - A(\rho, \mathbf{v}) + \mathbf{F}_v)] \mathbf{v}^{*,v} = -[M - \Delta t \theta (K - A(\rho, \mathbf{v}) + \mathbf{F}_v)] \mathbf{v}^\circ + \\
&+ [M + \Delta t (1 - \theta) (K - A(\rho, \mathbf{v}))] \mathbf{v}^n - \frac{\Delta t \theta}{\rho} \nabla \left( \bar{p}^\circ + p'' - \bar{p}^n \right) + \\
&+ \frac{\Delta t \theta}{\rho} \nabla \left( \bar{p}^\circ + p'' - \bar{p}^n \right) + \\
&+ \Delta t (1 - \theta) \mathbf{F}^n + \Delta t \theta \left( \mathbf{F}^\circ - \mathbf{F}_v \left( \frac{\Delta t \theta}{\rho} \nabla \left( \bar{p}^\circ - \bar{p}^n \right) + \mathbf{v}^\circ \right) \right) - \\
&- \Delta t \theta \left[ \mathbf{F}_v \frac{\Delta t \theta}{\rho} \nabla - f_p \right] p'' - \\
&- \Delta t \mathbf{B} \bar{p}'' - \left( \frac{\Delta t \theta}{\rho} \mathbf{B} (\bar{p}^\circ + p'' - \bar{p}^n) - \frac{\Delta t \theta}{\rho} \nabla \left( \bar{p}^\circ + p'' - \bar{p}^n \right) \right)
\end{aligned}
\]  

14
Option-B:

\[
\begin{align*}
[M - \Delta t \theta (K + F_v)] \mathbf{v}^{*'} &= - [M - \Delta t \theta (K + F_v)] \mathbf{v}^{\circ} + \\
&+ [M + \Delta t (1 - \theta) (K - A(\rho, \mathbf{v})]] \mathbf{v}^n - \\
- \Delta t \theta A(\rho, \mathbf{v}) \left( \mathbf{v}^{\circ} + \mathbf{v}^n \right) - \Delta t \theta K \left( \frac{\Delta t \theta}{\rho} \nabla \left( \bar{p}^{\circ} + p'' - \bar{p}^n \right) \right) + \\
&+ \Delta t \theta A(\rho, \mathbf{v}) \left( \frac{\Delta t \theta}{\rho} \nabla \left( \bar{p}^{\circ} + p'' - \bar{p}^n \right) \right) + \\
+ \Delta t (1 - \theta) \mathbf{F}^n &+ \Delta t \theta \left( \mathbf{F}^{\circ} - F_v \left( \frac{\Delta t \theta}{\rho} \nabla \left( \bar{p}^{\circ} - \bar{p}^n \right) + \mathbf{v}^{\circ} \right) \right) - \\
&- \Delta t \theta \left[ F_v \frac{\Delta t \theta}{\rho} \nabla - f_p \right] p'' - \\
- \Delta t \mathbf{B} p^n &= \left( \Delta t \theta \mathbf{B} (\bar{p}^{\circ} + p'' - \bar{p}^n) - \frac{\Delta t \theta}{\rho} \nabla \left( \bar{p}^{\circ} + p'' - \bar{p}^n \right) \right)
\end{align*}
\]

(63)

2. Solve incremental PPE:

\[
\nabla \cdot \frac{1}{\rho} \nabla \lambda' = \nabla \cdot \mathbf{v}^{*'} - \nabla \cdot \frac{1}{\rho} \nabla \lambda^{\circ}
\]

(64)

3. Convert to pressure correction as

\[
\bar{p}' = \frac{\lambda'}{\Delta t \theta_p}
\]

(65)

4. Return to PETSC-SNES a preconditioned solution as

\[
\begin{bmatrix}
\bar{p}' \ (or \ \lambda') \\
\mathbf{v}' = \mathbf{v}^{*'} - \frac{1}{\rho} \nabla \left( \lambda^{\circ} + \lambda' \right)
\end{bmatrix}
\]

(66)

7. Concluding Remarks

The main technical contribution of the present report is the formulation of the fully-implicit projection algorithm for implementation in Hydra-TH code. We discussed definition of non-linear residual vector, as well as the strategy for efficient preconditioning of linear (GMRES) solver, utilizing the variation of the currently-available in Hydra-TH semi-implicit projection algorithm. While focusing here on single-phase flow formulation, the basic ideas of the fully-implicit projection should be straightforwardly extendable to multi-fluid flows. These extensions will be presented in future.
Aknowledgement

This work has been authored by Battelle Energy Alliance, LLC under contract No. DE-AC07-05ID14517 (INL/EXT-12-27197) with the U.S. Department of Energy. The United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manual, or allow others to do so, for United States Government purposes.

References


