Open Channel Flow II

Having found that approaching open channel flow (or any coupled PDE system involving non-linearities within the kinetic flux vector!) from the primitive variable perspective resulted in a decoupled Newton algorithm that would converge linearly at best and diverge at worst. Let us now employ the non-divergence form of the problem statement and see where we can go.

\[
L(q) = \frac{\partial q}{\partial t} + A \frac{\partial q}{\partial x} = 0 \quad \text{on } \Omega
\]

(1a)

where

\[
q = \begin{bmatrix} h \\ m \end{bmatrix} \quad \text{and} \quad A = \begin{bmatrix} 0 & 1 \\ \frac{m^2}{h^2} + gh & \frac{m}{h} \end{bmatrix}
\]

(1b)

Substituting (1b) into (1a) and expanding

\[
L(h) = \frac{\partial h}{\partial t} + 0 \frac{\partial h}{\partial x} + 1 \frac{\partial m}{\partial x} = 0 \quad \text{on } \Omega
\]

\[
L(m) = \frac{\partial m}{\partial t} + \left( \frac{m^2}{h^2} + gh \right) \frac{\partial h}{\partial x} + \left( \frac{2 m}{h} \right) \frac{\partial m}{\partial x} = 0 \quad \text{on } \Omega
\]

(2)

Assuming a series approximation to the unknown state variables and associated grouped variables

\[
h(t,x) \approx h^N(t,x) = \Psi_a(x)H_a(t) \quad \text{for } 1 \leq \alpha \leq N
\]

\[
m(t,x) \approx m^N(t,x) = \Psi_a(x)M_a(t) \quad \text{for } 1 \leq \alpha \leq N
\]

\[
\frac{m^2}{h^2}(t,x) \approx \left( \frac{m^2}{h^2} \right)^N(t,x) = \Psi_a(x)MSH_a(t) \quad \text{for } 1 \leq \alpha \leq N
\]

\[
\frac{m}{h}(t,x) \approx \left( \frac{m}{h} \right)^N(t,x) = \Psi_a(x)MH_a(t) \quad \text{for } 1 \leq \alpha \leq N
\]

(3)

Forming the Galerkin Weak Statement

\[
GWS^N_h = \int_\Omega \Psi_h \left( \frac{\partial h^N}{\partial t} + \frac{\partial m^N}{\partial x} \right) dx = 0
\]

\[
GWS^N_m = \int_\Omega \Psi_m \left( \frac{\partial m^N}{\partial t} + \left( \frac{m^2}{h^2} \right)^N + gh^N \right) \frac{\partial h^N}{\partial x} + 2 \left( \frac{m}{h} \right)^N \frac{\partial m^N}{\partial x} dx = 0
\]

(4)

Substituting the series expansions and expanding
Upon assembly, the spatially discretized form (6) yields two coupled ordinary differential equations in time.

Identifying the usual master matrices

\[
\begin{bmatrix}
\text{MASS} & \text{CONVM1} \\
\text{CONVM2} & \text{MASS}
\end{bmatrix}
\begin{bmatrix}
\frac{d\{H\}}{dt} \\
\frac{d\{M\}}{dt}
\end{bmatrix}
+ \begin{bmatrix}
\text{CONVH1} & \text{CONVH2} & \text{CONVM2}
\end{bmatrix}
\begin{bmatrix}
\{H\} \\
\{M\}
\end{bmatrix}
= \{0\}
\] (7)

Unlike the previous formulation, we are able to combine the two equations in (7) into one large matrix statement

\[
\begin{bmatrix}
\text{MASS} & 0 & \text{CONVM1} \\
0 & \text{MASS} & \text{CONVM2}
\end{bmatrix}
\begin{bmatrix}
\frac{d\{H\}}{dt} \\
\frac{d\{M\}}{dt}
\end{bmatrix}
+ \begin{bmatrix}
\text{CONVH1} & \text{CONVH2} & \text{CONVM2}
\end{bmatrix}
\begin{bmatrix}
\{H\} \\
\{M\}
\end{bmatrix}
= \{0\}
\]

which is now of the form

\[
\begin{bmatrix}
\text{BIGMASS} & \text{BIGCONV}
\end{bmatrix}
\begin{bmatrix}
\frac{d\{Q\}}{dt}
\end{bmatrix}
= \{0\}
\] (8)

Employing the Theta Taylor Series to move the solution forward in time

\[
\Theta T S = \begin{bmatrix}
\text{BIGMASS}
\end{bmatrix}\Delta t \{\Theta \text{BIGRES} + (1-\Theta)\text{BIGRES}\}
\] (9)
To solve our big non-linear equation, we shall call upon Newton

\[ \{F_Q\}^p_{n,n+1} = \Theta T S^p_{n,n+1} = [\text{BIGMASS}]^p_{n,n+1} \Delta Q + \Delta t \left( \Theta [\text{BIGRES}]^p_{n+1} + (1 - \Theta) [\text{BIGRES}]_n \right) \]

The total residual was easy - now for the jacobian.

\[ [JAC]^p_{n+1} = \frac{\partial [F_Q]^p_{n,n+1}}{\partial [Q]_{n+1}} \]

The temporal term is readily evaluated as

\[ \frac{\partial}{\partial [Q]_{n+1}} ([\text{BIGMASS}] \Delta Q) = [\text{BIGMASS}] [\text{BIG}] \]

Evaluating the spatial jacobian

\[ \frac{\partial}{\partial [Q]_{n+1}} \left( \Delta t \left( \Theta [\text{BIGRES}]^p_{n+1} + (1 - \Theta) [\text{BIGRES}]_n \right) \right) = \Delta t \Theta \frac{\partial}{\partial [Q]_{n+1}} [\text{BIGRES}]^p_{n+1} \]

Taking our cue from the temporal development, the spatial jacobian will consist of four blocks

\[ \begin{bmatrix} JHH & JHM \\ JMH & JMM \end{bmatrix} \]