Pressure Solve with Finite Elements

Christian Hafner*
Vienna University of Technology

January 7, 2016

1 Preliminaries

After dropping the viscous term, the Navier-Stokes equations for incompressible fluids are given by

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho} \nabla p &= \mathbf{g}, \\ \nabla \cdot \mathbf{u} &= 0. \end{aligned} \tag{1}$$

This system of partial differential equations (PDEs) describes how an incompressible fluid behaves in space and time. In computer graphics, the spatial component is usually two- or three-dimensional, but for the purpose of this discussion, it will be viewed as two-dimensional. The vector quantity $\mathbf{u}$ is the velocity field of the fluid, and it is defined on the fluid domain $\Omega \subset \mathbb{R}^2$. This vector-valued functions assigns a velocity vector to every position, and can thus be written as $\mathbf{u} : \Omega \to \mathbb{R}^2$. The velocity field changes with time, and thus a more complete description would be

$$\begin{aligned} \mathbf{u} : \Omega \times \mathbb{R} &\to \mathbb{R}^2 \\ (\mathbf{x}, t) &\mapsto \mathbf{u}(\mathbf{x}, t), \end{aligned}$$

where t refers to time. However, we will write $\mathbf{u}(\mathbf{x})$ and $\mathbf{u}(t)$ in order to emphasize dependence on space or time alone.

The scalar quantity p is referred to as pressure. The Navier-Stokes equations ensure that the fluid flows from high-pressure regions to low-pressure regions. The pressure is defined on Ω, and it uses the same parameter conventions as $\mathbf{u}$.

The remaining quantities are the density ρ, which we view as a constant, and the acceleration due to body forces $\mathbf{g}$, such as acceleration due to gravity, which we also view as constant.

In solving the Navier-Stokes equations, we view time as a sequence of snapshots that are connected with each other using the forward Euler method. This means that equations including partial derivatives w.r.t. time, such as

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{f},$$

are approximated by forward finite differencing. This yields

$$\frac{\mathbf{u}(t + \Delta t) - \mathbf{u}(t)}{\Delta t} = \mathbf{f},$$

from which the state of the fluid at time $t + \Delta t$ can be found as

$$\mathbf{u}(t + \Delta t) = \mathbf{u}(t) + \Delta t \, \mathbf{f}.$$

from the state of the fluid at the previous time t.

The Navier-Stokes equations (1) also include spatial derivatives in the form of the gradient operator "∇" and the divergence operator "$\nabla\cdot$". Since it is difficult to tackle Eqs. (1) all at once, they are approximated using operator splitting. This allows us to solve a series of simple PDEs in sequence instead of solving a single difficult PDE. The splitting of Eqs. 1 yields three parts: advection, force application, and pressure projection. The advection problem

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = 0$$

*chafner@cg.tuwien.ac.at

is usually solved using Semi-Lagrangian advection or the modified MacCormack scheme in a real-time setting. Otherwise, methods with higher accuracy can be used. The force application problem

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{g}$$

becomes trivial after approximating the time derivative of $\mathbf{u}$ with forward finite differencing.

This leaves the pressure projection problem

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{\rho}\nabla p = 0, \tag{2}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{3}$$

during which the second part of the Navier-Stokes equations $\nabla \cdot \mathbf{u} = 0$ must also be fulfilled. The first step in solving Eq. (2) is to apply the forward Euler approximation to the time derivative. This yields

$$\frac{\mathbf{u} - \mathbf{w}}{\Delta t} + \frac{1}{\rho}\nabla p = 0,$$

where $\mathbf{u}$ refers to the "finished" velocity field at time $t + \Delta t$, and $\mathbf{w}$ refers to the "old" velocity field that results after advection and force application. We rewrite the equation as

$$\mathbf{u} = \mathbf{w} - \frac{\Delta t}{\rho}\nabla p, \tag{4}$$

where we know from Eq. (3) that $\mathbf{u}$ must have zero divergence.

Helmholtz's theorem, from the field of vector calculus, states that every vector field, such as $\mathbf{w}$, can be uniquely decomposed into a sum of a zero-divergence vector field, such as $\mathbf{u}$, and a gradient field, such as $\frac{\Delta t}{\rho}\nabla p$. Therefore, the quantities $\mathbf{u}$ and ∇p are uniquely defined via Eq. 4.

In order to solve for $\mathbf{u}$ in Eq. (4), we must compute the pressure field p from $\mathbf{w}$. We can find a PDE connecting p and $\mathbf{w}$ by applying the divergence operator to Eq. (4), giving

$$\nabla \cdot \mathbf{u} = \nabla \cdot \mathbf{w} - \frac{\Delta t}{\rho}\nabla \cdot \nabla p,$$

$$0 = \nabla \cdot \mathbf{w} - \frac{\Delta t}{\rho}\nabla \cdot \nabla p,$$

which can be written as

$$\frac{\Delta t}{\rho}\nabla^2 p = \nabla \cdot \mathbf{w}. \tag{5}$$

The operator "∇^2" is referred to as the Laplace operator, and it is equal to the divergence of the gradient, or "$\nabla \cdot \nabla$", of a scalar field. Solving Eq. (5) is the most delicate and computationally most expensive part of a fluid simulation. The reason is that Eq. (5) is a so-called "Poisson equation", which is a partial differential equation of elliptic type. It is characteristic of elliptic PDEs that their solutions are globally smooth, and that information present in one part of the domain influences the behavior globally. There is no sense of "information travelling", such as propagating waves, and therefore elliptic equations cannot be tackled using local methods such as the forward Euler method.

Instead, the Poisson Equation (5) is usually solved with the finite differencing method or the finite element method. The remainder of this text explains how it can be solved using the finite element method.

2 The Finite Element Method I

Given a vector field, in this case the velocity field $\mathbf{w}$, a solution to the Poisson problem

$$\frac{\Delta t}{\rho}\nabla^2 p = \nabla \cdot \mathbf{w}.$$

consists of a scalar function $p : \Omega \to \mathbb{R}$, where Ω is the fluid domain. In order to satisfy the equation exactly, p must be twice-differentiable w.r.t. the spatial coordinates; otherwise the pressure Laplacian $\nabla^2 p$ would not be defined. The pressure field p can be regarded as an infinite-dimensional variable, since it can have a different value at each point in Ω. Therefore the Poisson problem can not be solved analytically except for a few special cases.

The finite element method approximates the Poisson problem by substituting the infinite-dimensional pressure field p by a finite-dimensional pressure variable, which represents the values of p at a finite set of locations throughout Ω. These locations are referred to as "nodes" in a finite-elements context. If the value of p is needed at a location in Ω that does not coincide with a node, it is interpolated between nodal values. The same goes for any other quantity that is defined on Ω, such as the velocity field $\mathbf{w}$: The velocity vectors are defined at the node locations and interpolated in-between if needed.

The first step in a finite element aproximation is a partitioning of the domain Ω into a set of smaller subdomains, referred to as "elements". Elements come in all shapes and sizes, the most common being triangles and quads in two dimensions. The simplest elements, such as the 3-node triangle and the 4-node quad, have exactly one node at each corner. If higher accuracy is needed, additional nodes can be introduced, for example one additional node at the center of each edge. This leads to a 6-node triangle, or an 8-node quad. In scientific computing, it is common to work with elements that have numerous nodes on each edge, and also inside the element. For very high accuracy, one could for example construct a 25-node quad by arranging a 5-by-5 regular grid of nodes on a square. An element like this is referred to as a 5^{th}-order element.

Two neighboring elements in a finite element mesh are joined at an edge, and thus they share the nodes on that edge. E.g., a finite element mesh consisting of two 4-node quads can be formed by joining them at a single edge. Therefore the mesh consists of six nodes in total, two of which are shared by the two quads.

An element is first defined in a natural coordinate system (ξ, η). For a 4-node quad, it is common to place the corners at the locations $v_1 = (-1, -1)$, $v_2 = (-1, 1)$, $v_3 = (1, 1)$, $v_4 = (1, -1)$. As mentioned before, a quantity like pressure or velocity is defined at the nodal locations and interpolated in-between. In a 3-node triangle, the interpolation can be performed using barycentric coordinates, and in a 4-node quad, using bilinear interpolation.

Let $q_1, \ldots, q_4$ be the values of a quantity q at the four nodes $v_1, \ldots, v_4$ of a 4-node quad. The quantity q could be scalar, like pressure, or a vector-values quantity, like velocity. In order to find the value of q at (ξ, η), we first define

$$\underline{q}(\xi) = \frac{1}{2}\left((1-\xi)q_1 + (1+\xi)q_4\right),$$
$$\overline{q}(\xi) = \frac{1}{2}\left((1-\xi)q_2 + (1+\xi)q_3\right).$$

These two quantities $\underline{q}$ and $\overline{q}$ are the linear interpolations of q between nodes 1 and 4, and 2 and 3, respectively. The bilinearly interpolated value is then given by

$$q(\xi, \eta) = \frac{1}{2}\left((1-\eta)\underline{q}(\xi) + (1+\eta)\overline{q}(\xi)\right).$$

Next, we collect the coefficients of the nodal quantities $q_1, \ldots, q_4$ in $q(\xi, \eta)$. This gives

$$N_1(\xi, \eta) = \frac{1}{4}(1-\xi)(1-\eta),$$
$$N_2(\xi, \eta) = \frac{1}{4}(1-\xi)(1+\eta),$$
$$N_3(\xi, \eta) = \frac{1}{4}(1+\xi)(1+\eta),$$
$$N_4(\xi, \eta) = \frac{1}{4}(1+\xi)(1-\eta).$$

The interpolated quantity can be written as

$$q(\xi, \eta) = \sum_{i=1}^{4} N_i(\xi, \eta)\, q_i.$$

The functions $N_1, \ldots, N_4$ are referred to as shape functions of the 4-node quad element. They can be interpreted as the influence weights of the four corner nodes on the interior of the element.

It can be checked that this is a real interpolation scheme by noting that the shape functions sum to 1 at each point in Ω, and that $q(v_i) = q_i$ at the four corner nodes, i.e., the interpolated quantity matches the nodal quantity if evaluated at the node.

3 The Weak Form

Before the finite element approximation can be applied to the Poisson problem, it has to be cast into a different form: the weak form of the PDE. The weak form does not require that a PDE be fulfilled exactly at

each point on Ω, but that it be fulfilled "on average". This averaging is performed by multiplying the PDE with a "test" function ϕ and integrating. The weak form of the Poisson problem is

$$\frac{\Delta t}{\rho}\int_\Omega \nabla^2 p\ \phi\ \mathrm{d}A = \int_\Omega \nabla\cdot\mathbf{w}\ \phi\ \mathrm{d}A, \quad \phi \text{ is any test function.}$$

A solution to the original problem is always a solution to the weak form. The converse is not strictly true in general, but it is true for sufficiently smooth functions.

Next, we use the vector calculus identity

$$\nabla\cdot(\phi\nabla p) = \nabla^2 p\ \phi + \nabla\phi\cdot\nabla p$$

to transform the weak form into

$$\frac{\Delta t}{\rho}\int_\Omega \nabla p\cdot\nabla\phi\ \mathrm{d}A = \int_\Omega \nabla\cdot\mathbf{w}\ \phi\ \mathrm{d}A + \frac{\Delta t}{\rho}\int_\Omega \nabla\cdot(\phi\nabla p)\ \mathrm{d}A.$$

The last term can be transformed using the divergence theorem. This yields

$$\frac{\Delta t}{\rho}\int_\Omega \nabla p\cdot\nabla\phi\ \mathrm{d}A = \int_\Omega \nabla\cdot\mathbf{w}\ \phi\ \mathrm{d}A + \frac{\Delta t}{\rho}\oint_{\partial\Omega} \phi\ \nabla p\cdot\mathbf{n}\ \mathrm{d}s.$$

Here, $\partial\Omega$ denotes the boundary of the fluid domain, i.e. the closed path (or set of closed paths) along which the fluid comes into contact with air or with a solid. The vector $\mathbf{n}$ denotes the normal vector at the boundary.

The reason why the weak form is better suited to computation than the original, strong form is because no second derivatives appear in it: The Laplacian operator has been eliminated. Another advantage is that even solutions that are not differentiable, or not even continuous, are meaningful in the weak form because of the integration involved. Since it is indeed difficult to obtain completely smooth solutions, this is an important property of the weak form.

The next step is to deal with the path integral along the fluid boundary

$$\frac{\Delta t}{\rho}\oint_{\partial\Omega} \phi\ \nabla p\cdot\mathbf{n}\ \mathrm{d}s.$$

The boundary $\partial\Omega$ can be viewed as the union of two parts: the fluid-air interface $\partial\Omega_A$ and the fluid-solid interface $\partial\Omega_S$. It holds that $\partial\Omega_A\cap\partial\Omega_S = \emptyset$ and $\partial\Omega_A\cup\partial\Omega_S = \partial\Omega$.

We have to prescribe boundary conditions at the boundary for the PDE to have a unique solution. At the fluid-air interface $\partial\Omega_A$, we prescribe a zero-pressure boundary condition, i.e., $p = 0$. If $p = 0$ anyway, this means that we do not have to "test" this value using different values of the test functions ϕ. Instead we can set ϕ to zero at the fluid-air interface because we know beforehand that values of $p = 0$ will not contribute anything to the integral in the weak form.

At fluid-solid interface $\partial\Omega_S$, we cannot say anything about the value of p, but we postulate that the gradient in the direction of the boundary normal be equal to zero. Recall that the Navier-Stokes equations direct the fluid to flow from high-pressure regions to low-pressure regions. If there was a pressure gradient in the direction of the fluid-solid boundary normal, this would mean that fluid flows into the solid or out of the solid. To avoid this, the pressure gradient in the normal direction $\nabla p\cdot\mathbf{n}$ has to be zero.

We combine this information to cancel the path integral

$$\frac{\Delta t}{\rho}\oint_{\partial\Omega} \phi\ \nabla p\cdot\mathbf{n}\ \mathrm{d}s = \frac{\Delta t}{\rho}\oint_{\partial\Omega_A} \underbrace{\phi}_{0}\ \nabla p\cdot\mathbf{n}\ \mathrm{d}s + \frac{\Delta t}{\rho}\oint_{\partial\Omega_S} \phi\ \underbrace{\nabla p\cdot\mathbf{n}}_{0}\ \mathrm{d}s = 0.$$

This leaves us with a significantly simplified weak form

$$\frac{\Delta t}{\rho}\int_\Omega \nabla p\cdot\nabla\phi\ \mathrm{d}A = \int_\Omega \nabla\cdot\mathbf{w}\ \phi\ \mathrm{d}A, \quad \phi \text{ is any test function,} \tag{6}$$

which is finally ready for the application of the finite element method.

4 The Finite Element Method II

At this point, we need a finite element mesh that covers the fluid domain Ω. This partitioning $\Omega = \Omega^1\cup\cdots\cup\Omega^N$ could be a regular grid in the easiest case, or a complicated assembly of high-order elements. For this discussion, it is assumed that the assembly contains only 4-node quad elements. However, these elements are not required to be square or rectangles; all convex quadrangles are allowed. Note that everything discussion in

this section holds for different element types as well. One need only substitute the shape functions $N_1, \ldots, N_4$ with the shape functions $N_1, \ldots, N_n$ of the element that is being used.

It is also required, that a global numbering of all nodes $1, \ldots, N$ be available. Additionally, every individual element, i.e., every individual quad, has a local node numbering $1, \ldots, 4$, which needs to be compatible with the orientation of the nodes in the natural coordinate system as discussed in Section 2. Since the ordering of $v_1, \ldots, v_4$ was chosen to be clockwise, the local numbering of every element has to be clockwise as well.

We will now discuss the finite element version of the weak form Eq. (6) for a single element Ω^e. Once this is available, the weak form over the whole domain Ω can be recovered by adding the integrals like so

$$\int_\Omega (\cdot)\, \mathrm{d}A = \sum_{e=1}^{N} \int_{\Omega^e} (\cdot)\, \mathrm{d}A.$$

In building the finite-element version of Eq. (6), there are two challenges. Firstly, the continuous variables p, $\mathbf{w}$, and ϕ, need to be substituted with their finite-element counterparts, quantities that have individual values at the node locations and are interpolated in-between. Secondly, the domain of the integrals are the element domains Ω^e in a global coordinate system, which can be strongly distorted quadrangles. On the other hand, the shape functions $N_1, \ldots, N_4$ are defined in terms of natural coordinates ξ, η. In order to make these integrals manageable, we have to perform a coordinate change from global to natural coordinates.

To deal with this added complexity, we will employ matrix notation. Let

$$\mathbf{N} = \begin{pmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_1}{\partial \eta} \\ \frac{\partial N_2}{\partial \xi} & \frac{\partial N_2}{\partial \eta} \\ \frac{\partial N_3}{\partial \xi} & \frac{\partial N_3}{\partial \eta} \\ \frac{\partial N_4}{\partial \xi} & \frac{\partial N_4}{\partial \eta} \end{pmatrix}, \quad \tilde{\mathbf{X}} = \begin{pmatrix} x_1^e & x_2^e & x_3^e & x_4^e \\ y_1^e & y_2^e & y_3^e & y_4^e \end{pmatrix}.$$

Note that $\mathbf{N}$ and $\mathbf{G}$ are matrix-values functions that depend on ξ, η. The constant matrix $\tilde{\mathbf{X}}$ collects the global coordinates of the nodes of the element. Since derivatives of p, $\mathbf{w}$, and ϕ appear in the integrals, it will be necessary to evaluate $\frac{\partial N_i}{\partial x}$ and $\frac{\partial N_i}{\partial y}$. However, N_i are defined in terms of natural coordinates ξ, η, and thus the chain rule has to be used, e.g.,

$$\frac{\partial N_3}{\partial y} = \frac{\partial N_3}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial N_3}{\partial \eta}\frac{\partial \eta}{\partial y}.$$

The derivatives of the shape functions w.r.t. natural coordinates are available directly from the definitions of the shape functions. In order to evaluate expressions like $\frac{\partial \xi}{\partial y}$, we write the global coordinates x, y and their derivatives as functions of ξ, η by using the element interpolation scheme.

$$\begin{pmatrix} x \\ y \end{pmatrix} = \tilde{\mathbf{X}}\mathbf{N}, \quad \mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{pmatrix} = \tilde{\mathbf{X}}\mathbf{G}.$$

By inverting the Jacobi matrix $\mathbf{J}$, we find the derivatives we need like so

$$\mathbf{J}^{-1} = \begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_1}{\partial y} \\ \frac{\partial N_2}{\partial x} & \frac{\partial N_2}{\partial y} \\ \frac{\partial N_3}{\partial x} & \frac{\partial N_3}{\partial y} \\ \frac{\partial N_4}{\partial x} & \frac{\partial N_4}{\partial y} \end{pmatrix} = \mathbf{G}\mathbf{J}^{-1}.$$

By calculating $\mathbf{B}$ for a given pair of natural coordinates ξ, η, we can evaluate the derivatives of the shape functions w.r.t. to global coordinates.

We will now calculate the finite-element counterpart of $\nabla \cdot \mathbf{w} = \frac{\partial w_x}{\partial x} + \frac{\partial w_y}{\partial y}$ from the right-hand side of Eq. (6), where we have denoted $\mathbf{w} = (w_x, w_y)^T$. Let

$$\tilde{\mathbf{w}}_x^e = \begin{pmatrix} w_{1,x}^e \\ w_{2,x}^e \\ w_{3,x}^e \\ w_{4,x}^e \end{pmatrix}, \quad \tilde{\mathbf{w}}_y^e = \begin{pmatrix} w_{1,y}^e \\ w_{2,y}^e \\ w_{3,y}^e \\ w_{4,y}^e \end{pmatrix}$$

be the x- and y-coordinate vectors of the velocities at the nodal locations. Then the finite-element counterpart of the scalar quantity $\nabla \cdot \mathbf{w}$ is given by

$$\mathbf{B}_{|1|}^T \tilde{\mathbf{w}}_x^e + \mathbf{B}_{|2|}^T \tilde{\mathbf{w}}_y^e,$$

where $\mathbf{B}_{|1|}$ and $\mathbf{B}_{|2|}$ denote the first and second column of $\mathbf{B}$ respectively.

The test function ϕ is also replaced with its finite-element counterpart

$$\left(\tilde{\boldsymbol{\phi}}^e\right)^T \mathbf{N}, \quad \tilde{\boldsymbol{\phi}}^e = \begin{pmatrix} \phi_1^e \\ \phi_2^e \\ \phi_3^e \\ \phi_4^e \end{pmatrix}.$$

Putting everything together and performing the coordinate change from global to natural coordinates, we arrive at

$$\int_{\Omega^e} \nabla \cdot \mathbf{w}\ \phi\ \mathrm{d}A \approx \left(\tilde{\boldsymbol{\phi}}^e\right)^T \int_{\Omega^e} \left(\mathbf{B}_{|1|}^T \tilde{\mathbf{w}}_x^e + \mathbf{B}_{|2|}^T \tilde{\mathbf{w}}_y^e\right) \mathbf{N}\ \mathrm{d}A = \left(\tilde{\boldsymbol{\phi}}^e\right)^T \int_{\Omega_0} |\mathbf{J}| \left(\mathbf{B}_{|1|}^T \tilde{\mathbf{w}}_x^e + \mathbf{B}_{|2|}^T \tilde{\mathbf{w}}_y^e\right) \mathbf{N}\ \mathrm{d}A.$$

Note that the last integral evaluates to a 4-by-1 vector. Here, $|\mathbf{J}|$ denotes the determinant of the Jacobi matrix. The domain Ω_0 is the domain of the element in natural coordinates. For the 4-node quad element, this is the region $[-1,1] \times [-1,1]$. The quantities $\mathbf{J}$, $\mathbf{B}$, and $\mathbf{N}$ all depend on the natural coordinates ξ, η. On the other hand $\tilde{\mathbf{w}}_x$ and $\tilde{\mathbf{w}}_y$ are constants across the element, since they denote nodal quantities. The vector $\tilde{\boldsymbol{\phi}}^e$ determines the test function.

We will go through a similar derivation for the left-hand side of Eq. (6) in less detail. Let $\tilde{\mathbf{p}}^e = (p_1^e, p_2^e, p_3^e, p_4^e)^T$ denote the unknown pressure values at the nodes of the element. Then the finite-element counterpart of p is $\mathbf{N}^T \tilde{\mathbf{p}}^e$, and the counterpart of ∇p is $\mathbf{B}^T \tilde{\mathbf{p}}^e$. The integral on the left-hand side of Eq. (6) is transformed as follows

$$\begin{aligned} \int_{\Omega^e} \nabla p \cdot \nabla \phi\ \mathrm{d}A &= \int_{\Omega^e} (\nabla \phi)^T \nabla p\ \mathrm{d}A \\ &\approx \int_{\Omega^e} \left(\mathbf{B}^T \tilde{\boldsymbol{\phi}}^e\right)^T \mathbf{B}^T \tilde{\mathbf{p}}^e\ \mathrm{d}A = \left(\tilde{\boldsymbol{\phi}}^e\right)^T \left(\int_{\Omega^e} \mathbf{B}\mathbf{B}^T \mathrm{d}A\right) \tilde{\mathbf{p}}^e = \left(\tilde{\boldsymbol{\phi}}^e\right)^T \left(\int_{\Omega_0} |\mathbf{J}| \mathbf{B}\mathbf{B}^T \mathrm{d}A\right) \tilde{\mathbf{p}}^e. \end{aligned}$$

Note that the last integral evaluates to a 4-by-4 matrix.

Solving any of the two integrals for the left-hand side or the right-hand side exactly would be computationally expensive, and therefore they are solved numerically using Gauss quadrature. Which Gauss quadrature rule to use depends on the element, but for the 4-node quad element a 4-point Gauss rule suffices. The rule is defined with the sample locations and sample weights

$$\mathbf{s}_{1,\ldots,4} = \begin{pmatrix} \pm\sqrt{\frac{1}{3}} \\ \pm\sqrt{\frac{1}{3}} \end{pmatrix}, \quad w_1 = \ldots = w_4 = 1.$$

Integrals over the domain Ω_0 can be solved numerically like

$$\int_{\Omega_0} |\mathbf{J}| \mathbf{B}\mathbf{B}^T \mathrm{d}A \approx \sum_{i=1}^{4} w_i\, |\mathbf{J}(\mathbf{s}_i)|\, \mathbf{B}(\mathbf{s}_i)\, \mathbf{B}^T(\mathbf{s}_i).$$

Now that we can evaluate both sides of finite-element the Poisson equation for a single element, the last step is to combine results of all elements. We will start with the right-hand side

$$\mathbf{f}^e = \int_{\Omega_0} |\mathbf{J}| \left(\mathbf{B}_{|1|}^T \tilde{\mathbf{w}}_x^e + \mathbf{B}_{|2|}^T \tilde{\mathbf{w}}_y^e\right) \mathbf{N}\ \mathrm{d}A.$$

The sum over all element domains yields

$$\sum_{e=1}^{M} \left(\tilde{\boldsymbol{\phi}}^e\right)^T \mathbf{f}^e,$$

where M is the number of elements in the assembly. Note that each element contains exactly four nodes, whose test function variables $\phi_1^e, \ldots, \phi_4^e$ are contained in $\tilde{\boldsymbol{\phi}}^e$. Let $\tilde{\boldsymbol{\phi}} = (\phi_1, \phi_2, \ldots, \phi_N)^T$ be the N-by-1 vector listing all test function nodal variables in the assembly. Then we can rewrite the sum above as

$$\sum_{e=1}^{M} \left(\tilde{\boldsymbol{\phi}}^e\right)^T \mathbf{f}^e = \tilde{\boldsymbol{\phi}}^T \mathbf{f}.$$

The vector $\mathbf{f}$ is computed by pasting the individual $\mathbf{f}^e$ into the appropriate node locations in $\mathbf{f}$, where contributions to the same node from different elements are accumulated.

The same process is followed for the left-hand side. Let

$$\mathbf{K}^e = \int_{\Omega_0} |\mathbf{J}| \mathbf{B}\mathbf{B}^T \mathrm{d}A.$$

The sum over all left-hand side contributions can be rewritten as

$$\sum_{e=1}^{M} \left(\tilde{\boldsymbol{\phi}}^e\right)^T \mathbf{K}^e \tilde{\mathbf{p}}^e = \tilde{\boldsymbol{\phi}}^T \mathbf{K} \tilde{\mathbf{p}},$$

where $\mathbf{p} = (p_1, p_2, \ldots, p_N)^T$ is the N-by-1 vector listing all nodal pressure variables in the assembly. The matrix $\mathbf{K}$ is N-by-N, and the contributions of each $\mathbf{K}^e$ are pasted into the corresponding nodal locations in K. The finite-element version of Eq. (6) now reads

$$\tilde{\boldsymbol{\phi}}^T \left(\frac{\Delta t}{\rho} \mathbf{K} \tilde{\mathbf{p}}\right) = \tilde{\boldsymbol{\phi}}^T \mathbf{f}, \quad \text{for all } \tilde{\boldsymbol{\phi}}.$$

An equivalent statement is

$$\frac{\Delta t}{\rho} \mathbf{K} \tilde{\mathbf{p}} = \mathbf{f}, \tag{7}$$

which is an N-by-N system of linear equations in the nodal pressure values that has a unique solution.

These pressure values are needed to compute $\mathbf{u}$ in Eq. (4). The gradient ∇p can be found by central differencing in the interior of the fluid, and by differencing in a single direction at the boundary. Also note that the factors $\frac{\Delta t}{\rho}$ in Eqs. (4) and (7) cancel and can therefore be ignored consistently.