

# Explicit Jacobians for the Incompressible Euler Equations on Curvilinear Grids

Oskar Ålund, Fredrik Laurén, Jan Nordström  
*Department of Mathematics, Computational Mathematics,  
Linköping University,  
SE-581 83 Linköping, Sweden*

---

## Abstract

We derive the Jacobian for a finite difference discretization of the incompressible Euler equations in two dimensions. By using encapsulated difference operators it is straight forward to explicitly obtain the Jacobian also for curved grids. Different sets of boundary conditions are imposed and the corresponding parts of the Jacobian are derived. The numerical scheme is verified by a manufactured solution.

---

## 1. Introduction

Discretizing partial differential equations revolves around three major considerations:

1. Stability
2. Simplicity
3. Efficiency

These aspects continually compete for attention, and narrow focus on one of them is often detrimental to the other two. This paper is an attempt to strike a balance between the three in the context of the incompressible Euler equations. Point 1 (stability) is achieved by using well established techniques, such as weakly imposed boundary conditions and integration-by-parts mimicking discrete differential operators. Point 2 (simplicity) is achieved through a convenient notational structure that closely resembles the continuous notation both functionally and visually. Our ambition is to narrow the bridge between the continuous and discrete settings as much as possible so that understanding of the continuous problem transfers with little effort to the discrete problem. Point 3 (efficiency) is achieved using two key tools: encapsulation and explicit Jacobians. By encapsulation we mean that the discrete derivative operators can be thought of as abstract operators that act on discrete approximations of differentiable functions, while satisfying certain key properties (most notably accuracy and summation-by-parts). Such abstraction is useful, not only because it simplifies stability proofs, but also because it isolates concerns: Discrete differentiation

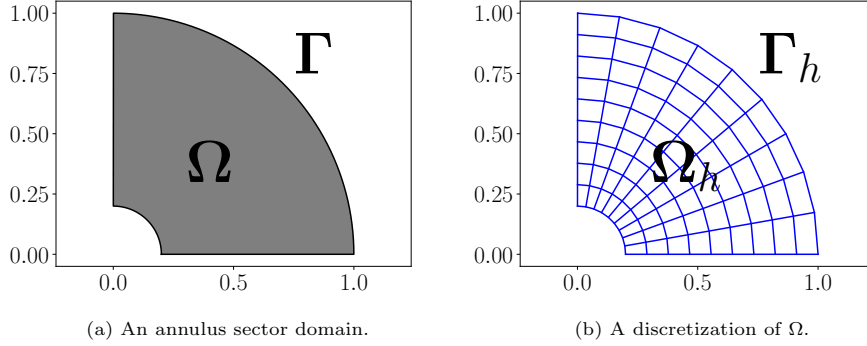


Figure 1: A curved domain  $\Omega$  and a named boundary  $\Gamma$ .

can be implemented, optimized, and maintained in separation from any particular solver. Encapsulation also allows us to write down, in very compact form, explicit Jacobians for the discretized incompressible Euler equations in terms of the derivative operators.

Explicit Jacobians are particularly powerful in the context of implicit timestepping since they are typically orders of magnitude faster to evaluate than for example a finite difference based approximation of the Jacobian. Although we in this paper consider the incompressible Euler equations in two dimensions, the framework can be extended to any type of hyperbolic or parabolic equations.

## 2. Notation

In the continuous setting we consider time dependent scalar- and vector-valued functions on a spatial domain  $\Omega$  (such as in Figure 1a). Vector-valued functions are column vectors and indicated with an arrow, for example  $\vec{w} = \begin{pmatrix} u & v \end{pmatrix}^\top$  denoting the velocity field. Let  $\vec{\phi}, \vec{\psi} : \Omega \rightarrow \mathbb{R}^n$ . We denote by  $\|\vec{\phi}\|_\Omega$  and  $(\vec{\phi}, \vec{\psi})_\Omega$  the  $L^2$  inner product and norm on  $\Omega$  respectively,

$$(\vec{\phi}, \vec{\psi})_\Omega := \int_\Omega \vec{\phi} \cdot \vec{\psi} dx dy, \quad \|\vec{\phi}\|_\Omega := \sqrt{\int_\Omega |\vec{\phi}|^2 dx dy}.$$

Similarly, the  $L^2$  inner product on  $\partial\Omega$  is denoted  $(\cdot, \cdot)_{\partial\Omega}$ ,

$$(\vec{\phi}, \vec{\psi})_{\partial\Omega} := \int_{\partial\Omega} \vec{\phi} \cdot \vec{\psi} ds.$$

Recall the definitions of the gradient  $(\nabla)$  and divergence  $(\nabla \cdot)$  operators,

$$\nabla u = (u_x, u_y), \quad \nabla \cdot \vec{w} = u_x + v_y,$$

and Green's identity,

$$\left(\vec{\phi}, \nabla \psi\right)_{\Omega} = \left(\vec{\phi} \cdot \vec{n}, \psi\right)_{\partial\Omega} - \left(\nabla \cdot \vec{\phi}, \psi\right)_{\Omega}. \quad (1)$$

where  $\vec{n}$  is the outward point unit normal of the boundary  $\partial\Omega$ . We allow the gradient, the divergence, and the dot product to operate row-wise, so that for example

$$\vec{w} \cdot \nabla \vec{w} = \begin{pmatrix} \vec{w} \cdot \nabla u \\ \vec{w} \cdot \nabla v \end{pmatrix} = \begin{pmatrix} uu_x + vv_y \\ uv_x + vv_y \end{pmatrix}$$

and

$$\nabla \cdot \vec{w} \vec{w}^{\top} = \nabla \cdot \begin{pmatrix} uu & uv \\ uv & vv \end{pmatrix} = \begin{pmatrix} (uu)_x + (uv)_y \\ (uv)_x + (vv)_y \end{pmatrix}.$$

The semi-discrete setting mimics the continuous setting to a large extent, with a few caveats (such as no product rule for differentiation). Notationally we will see that the continuous and discrete settings are virtually identical.

Consider an  $N_x$ -by- $N_y$  curvilinear grid  $\Omega_h$  as in Figure 1b. We think of our semi-discrete solutions as column vectors of function evaluations on this grid, denoted in bold. For example, if  $u : \Omega \rightarrow \mathbb{R}$ , then  $\mathbf{u} = (\mathbf{u}_1^{\top} \quad \mathbf{u}_2^{\top} \quad \cdots \quad \mathbf{u}_{N_x}^{\top})^{\top}$ , where

$$\mathbf{u}_i = (u_{i1} \quad u_{i2} \quad \cdots \quad u_{iN_y})^{\top} \approx (u(x_{i1}, y_{i1}) \quad u(x_{i2}, y_{i2}) \quad \cdots \quad u(x_{iN_y}, y_{iN_y}))^{\top}.$$

Note that  $\mathbf{u}$  is a special kind of vector in that its purpose is to approximate a continuous function. To make this distinction clear let us make a formal definition.

**Definition 1.** *Given an  $N_x$ -by- $N_y$  curvilinear grid  $\Omega_h$ , a vector  $\mathbf{u} \in \mathbb{R}^{N_x N_y}$  is called a grid function.*

We make extensive use of elementwise operations on grid functions. If  $\mathbf{u}$  and  $\mathbf{v}$  are grid functions, then the elementwise product is indicated with no symbol,

$$\mathbf{uv} = (u_{11}v_{11} \quad u_{12}v_{12} \quad \cdots \quad u_{21}v_{21} \quad u_{22}v_{22} \quad \cdots)^{\top}.$$

Similarly elementwise fractions are indicated by a division symbol,

$$\mathbf{u/v} = (u_{11}/v_{11} \quad u_{12}/v_{12} \quad \cdots \quad u_{21}/v_{21} \quad u_{22}/v_{22} \quad \cdots)^{\top}.$$

For a scalar function  $h : \mathbb{R} \rightarrow \mathbb{R}$  we will write  $h(\mathbf{u})$  to indicate elementwise application of  $h$  to  $\mathbf{u}$ ,

$$h(\mathbf{u}) = (h(u_{11}) \quad h(u_{12}) \quad \cdots \quad h(u_{21}) \quad h(u_{22}) \quad \cdots)^{\top}.$$

To further mimic the continuous setting we introduce matrices with grid function components. To get a sense for what this means and why it is useful,

let us consider an example. In the incompressible Euler equations, the outer product

$$\vec{w} \vec{w}^\top = \begin{pmatrix} uu & uv \\ uv & vv \end{pmatrix}$$

appears (see equation (3) in Section 3). To mimic this operation it is not enough to simply concatenate grid functions into a long vector. Consider what happens if we multiply the vectors  $(\mathbf{u}^\top \ \mathbf{v}^\top)^\top$  and  $(\mathbf{u}^\top \ \mathbf{v}^\top)$ . The result is an  $N_x N_y$ -by- $N_x N_y$  matrix that does not correspond to any sort of approximation of  $\vec{w} \vec{w}^\top$ . Instead, we need to be able to form matrices where the components of the matrices are not scalars, but grid functions. Such matrices will be denoted by square brackets, and their transposes by  $\dagger$ . For example, the discrete velocity field is

$$\vec{w} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}$$

and its transpose is  $\vec{w}^\dagger = [\mathbf{u} \ \mathbf{v}]$ . Note that transposition *does not* act on the individual grid functions of the matrix. This means that when we form the outer product

$$\vec{w} \vec{w}^\dagger = \begin{bmatrix} \mathbf{u}\mathbf{u} & \mathbf{u}\mathbf{v} \\ \mathbf{u}\mathbf{v} & \mathbf{v}\mathbf{v} \end{bmatrix}$$

we get a matrix that approximates  $\vec{w} \vec{w}^\top$ .

A column matrix of grid functions, like  $\vec{w}$ , is called a *vector valued* grid function. We define the dot product between two vector valued grid functions  $\vec{\phi} = [\phi_1 \ \phi_2 \ \cdots \ \phi_n]^\dagger$  and  $\vec{\psi} = [\psi_1 \ \psi_2 \ \cdots \ \psi_n]^\dagger$  as

$$\vec{\phi} \cdot \vec{\psi} = \vec{\phi}^\dagger \vec{\psi} = \sum_{i=1}^n \phi_i \psi_i.$$

Note that if  $\vec{\phi}$  and  $\vec{\psi}$  approximate the vector valued functions  $\vec{\phi}$  and  $\vec{\psi}$ , then  $\vec{\phi} \cdot \vec{\psi}$  is a grid function that approximates the scalar function  $\vec{\phi} \cdot \vec{\psi}$ .

For numerical differentiation we use the encapsulated SBP operators described in [1], together with some additional convenient notation. Let  $\mathbf{D}_x$  and  $\mathbf{D}_y$  be encapsulated SBP operators on  $\Omega_h$  in the  $x$ - and  $y$ -direction respectively. Associated with the operators  $\mathbf{D}_x$  and  $\mathbf{D}_y$  is a diagonal positive definite quadrature matrix  $\mathbf{P}$ . This matrix defines a discrete inner product and a norm:

$$(\mathbf{u}, \mathbf{v})_{\Omega_h} := \mathbf{u}^\top \mathbf{P} \mathbf{v} \approx (u, v)_\Omega, \quad \|\mathbf{u}\|_{\Omega_h} := \sqrt{(\mathbf{u}, \mathbf{u})_{\Omega_h}} \approx \|u\|_\Omega.$$

The above norm and inner product have natural extensions to vector valued grid functions:

$$(\vec{\phi}, \vec{\psi})_{\Omega_h} := \sum_{i=1}^n (\phi_i, \psi_i)_{\Omega_h}, \quad \|\vec{\phi}\|_{\Omega_h} := \sqrt{\sum_{i=1}^n \|\phi_i\|_{\Omega_h}^2}.$$

We refer to the operation  $(\cdot, \cdot)_{\Omega_h}$  as *discrete integration in space*.

Consider now  $\Gamma_d$ ,  $d \in \{e, w, s, n\}$ —the four boundaries of  $\Omega_h$ . Also associated with the SBP operators are the numerical outward-pointing normals of  $\Gamma_d$ :  $\vec{n} = [\mathbf{n}_x \ \mathbf{n}_y]^\dagger$ , as well as diagonal boundary quadrature matrices  $\mathbf{P}_d$ . The matrix  $\mathbf{P}_d$  defines a discrete inner product on the boundary  $\Gamma_d$ :

$$(\mathbf{u}, \mathbf{v})_{\Gamma_d} := \mathbf{u}^\top \mathbf{P}_d \mathbf{v} \approx \int_{\Gamma_d} u v ds.$$

We also define the discrete inner product on the full boundary  $\partial\Omega_h$ :

$$(\mathbf{u}, \mathbf{v})_{\partial\Omega_h} := \sum_d (\mathbf{u}, \mathbf{v})_{\Gamma_d} \approx \int_{\partial\Omega} u v ds.$$

Recall from [1] that the operators  $\mathbf{D}_x$  and  $\mathbf{D}_y$  satisfy the following summation-by-parts properties.

**Proposition 1.** *Assume  $\mathbf{D}_x$  and  $\mathbf{D}_y$  are SBP operators on a curvilinear grid  $\Omega_h$ , and  $\mathbf{u}$ ,  $\mathbf{v}$  are grid functions. Then*

$$\begin{aligned} (\mathbf{u}, \mathbf{D}_x \mathbf{v})_{\Omega_h} &= (\mathbf{u}, \mathbf{n}_x \mathbf{v})_{\partial\Omega_h} - (\mathbf{D}_x \mathbf{u}, \mathbf{v})_{\Omega_h}, \\ (\mathbf{u}, \mathbf{D}_y \mathbf{v})_{\Omega_h} &= (\mathbf{u}, \mathbf{n}_y \mathbf{v})_{\partial\Omega_h} - (\mathbf{D}_y \mathbf{u}, \mathbf{v})_{\Omega_h}. \end{aligned}$$

**Remark 1.** *Proposition 1 is a discrete version of the integration by parts formulas  $(u, v_x)_\Omega = (u, n_x v)_{\partial\Omega} - (u_x, v)_\Omega$  and  $(u, v_y)_\Omega = (u, n_y v)_{\partial\Omega} - (u_y, v)_\Omega$ .*

We define the discrete gradient  $\nabla$  and divergence  $\nabla \cdot$  by

$$\begin{aligned} \nabla \mathbf{u} &:= [\mathbf{D}_x \mathbf{u} \ \mathbf{D}_y \mathbf{u}] \\ \nabla \cdot \vec{w} &:= \mathbf{D}_x \mathbf{u} + \mathbf{D}_y \mathbf{v}. \end{aligned}$$

These operators may also act row-wise on matrices just like in the continuous setting, so that for example

$$\vec{w} \cdot \nabla \vec{w} = \begin{bmatrix} \vec{w} \cdot \nabla \mathbf{u} \\ \vec{w} \cdot \nabla \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{u} \mathbf{D}_x \mathbf{u} + \mathbf{v} \mathbf{D}_y \mathbf{u} \\ \mathbf{u} \mathbf{D}_x \mathbf{v} + \mathbf{v} \mathbf{D}_y \mathbf{v} \end{bmatrix}$$

and

$$\nabla \cdot \vec{w} \vec{w}^\dagger = \nabla \cdot \begin{bmatrix} \mathbf{u} \mathbf{u} & \mathbf{u} \mathbf{v} \\ \mathbf{u} \mathbf{v} & \mathbf{v} \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{D}_x(\mathbf{u} \mathbf{u}) + \mathbf{D}_y(\mathbf{u} \mathbf{v}) \\ \mathbf{D}_x(\mathbf{u} \mathbf{v}) + \mathbf{D}_y(\mathbf{v} \mathbf{v}) \end{bmatrix}.$$

A discrete version of Green's identity follows directly from Proposition 1.

**Proposition 2.** *Assume  $\mathbf{D}_x$  and  $\mathbf{D}_y$  are encapsulated SBP operators on a curvilinear grid  $\Omega_h$ ,  $\vec{\phi}$  is a vector-valued grid function with two components, and  $\psi$  is a grid function. Then*

$$(\vec{\phi}, \nabla \psi)_{\Omega_h} = (\vec{\phi} \cdot \vec{n}, \psi)_{\partial\Omega_h} - (\nabla \cdot \vec{\phi}, \psi)_{\Omega_h}.$$

*Proof.* By Proposition 1,

$$\begin{aligned}
\left( \vec{\phi}, \nabla \psi \right)_{\Omega_h} &= (\phi_1, \mathbf{D}_x \psi)_{\Omega_h} + (\phi_2, \mathbf{D}_y \psi)_{\Omega_h} \\
&= (\phi_1, \mathbf{n}_x \psi)_{\partial\Omega_h} - (\mathbf{D}_x \phi_1, \psi)_{\Omega_h} + (\phi_2, \mathbf{n}_y \psi)_{\partial\Omega_h} - (\mathbf{D}_y \phi_2, \psi)_{\Omega_h} \\
&= (\phi_1, \mathbf{n}_x \psi)_{\partial\Omega_h} + (\phi_2, \mathbf{n}_y \psi)_{\partial\Omega_h} - (\mathbf{D}_x \phi_1 + \mathbf{D}_y \phi_2, \psi)_{\Omega_h} \\
&= \left( \vec{\phi} \cdot \vec{n}, \psi \right)_{\partial\Omega_h} - \left( \nabla \cdot \vec{\phi}, \psi \right)_{\Omega_h} .
\end{aligned}$$

□

### 3. The incompressible Euler equations

Consider the incompressible Euler equations

$$\vec{w}_t + \vec{w} \cdot \nabla \vec{w} = -\nabla p \quad (2a)$$

$$\nabla \cdot \vec{w} = 0 \quad (2b)$$

where  $\vec{w} = [u \ v]^\top$  is the velocity field,  $p$  is the pressure. The form (2a) is not suitable for discretization because its energy analysis relies on the chain rule, which cannot be replicated discretely. Let us go through the energy analysis to see that it does indeed rely on the chain rule (this will also help us understand how to get around the problem).

The rate of change in the kinetic energy  $\frac{1}{2} \|\vec{w}\|_\Omega^2$  can be found by multiplying (2a) by  $2\vec{w}^\top$  and integrating in space, since

$$2(\vec{w}, \vec{w}_t)_\Omega = \frac{d}{dt} \|\vec{w}\|_\Omega^2 .$$

The convective term  $\vec{w} \cdot \nabla \vec{w}$  adds or dissipates energy through the boundaries via the integral  $2(\vec{w}, \vec{w} \cdot \nabla \vec{w})_\Omega$ . Let  $w_n = \vec{w} \cdot \vec{n}$  (the normal velocity). An expression for the energy rate of change due to convection can be derived as follows.

**Lemma 1.** *The incompressible Euler equations (2) imply that*

$$2(\vec{w}, \vec{w} \cdot \nabla \vec{w})_\Omega = (w_n, |\vec{w}|^2)_{\partial\Omega} .$$

*Proof.*

$$\begin{aligned}
2(\vec{w}, \vec{w} \cdot \nabla \vec{w})_\Omega &= 2(u, \vec{w} \cdot \nabla u)_\Omega + 2(v, \vec{w} \cdot \nabla v)_\Omega \\
&= (\vec{w}, \nabla u^2)_\Omega + (\vec{w}, \nabla v^2)_\Omega && \text{(Chain rule)} \\
&= (\vec{w}, \nabla |\vec{w}|^2)_\Omega \\
&= (w_n, |\vec{w}|^2)_{\partial\Omega} - (\nabla \cdot \vec{w}, |\vec{w}|^2)_\Omega && \text{(Green's identity)} \\
&= (w_n, |\vec{w}|^2)_{\partial\Omega} && \text{(Zero divergence)} .
\end{aligned}$$

□

Using Lemma 1 it is straightforward to write  $\frac{d}{dt} \|\vec{w}\|_{\Omega}^2$  in terms of boundary contributions.

**Proposition 3.** *The incompressible Euler equations (2) imply that*

$$\frac{d}{dt} \|\vec{w}\|_{\Omega}^2 = - (w_n, |\vec{w}|^2)_{\partial\Omega} - 2 (w_n, p)_{\partial\Omega} .$$

*Proof.*

$$\begin{aligned} \frac{d}{dt} \|\vec{w}\|_{\Omega}^2 &= -2 (\vec{w}, \vec{w} \cdot \nabla \vec{w})_{\Omega} - 2 (\vec{w}, \nabla p)_{\Omega} \\ &= - (w_n, |\vec{w}|^2)_{\partial\Omega} - 2 (\vec{w}, \nabla p)_{\Omega} && \text{(Lemma 1)} \\ &= - (w_n, |\vec{w}|^2)_{\partial\Omega} - 2 (w_n, p)_{\partial\Omega} + 2 (\nabla \cdot \vec{w}, p)_{\Omega} && \text{(Green's identity)} \\ &= - (w_n, |\vec{w}|^2)_{\partial\Omega} - 2 (w_n, p)_{\partial\Omega} && \text{(Zero divergence)} \end{aligned}$$

□

We want to remove the dependence on the chain rule in the energy analysis. To this end, note that the identity  $\nabla \cdot (\vec{w} \vec{w}^{\top}) = \vec{w} \nabla \cdot \vec{w} + \vec{w} \cdot \nabla \vec{w} = \vec{w} \cdot \nabla \vec{w}$  holds since  $\nabla \cdot \vec{w} = 0$ . Hence, the incompressible Euler equations can be rewritten in split form as

$$\vec{w}_t + \frac{1}{2} \vec{w} \cdot \nabla \vec{w} + \frac{1}{2} \nabla \cdot (\vec{w} \vec{w}^{\top}) = -\nabla p \quad (3a)$$

$$\nabla \cdot \vec{w} = 0. \quad (3b)$$

The energy analysis for (3) can then be performed without the chain rule by using the following lemma.

**Lemma 2.** *For any smooth  $\vec{w} : \Omega \rightarrow \mathbb{R}^2$ ,*

$$(\vec{w}, \vec{w} \cdot \nabla \vec{w})_{\Omega} = (w_n, |\vec{w}|^2)_{\partial\Omega} - \left( \vec{w}, \nabla \cdot (\vec{w} \vec{w}^{\top}) \right)_{\Omega} .$$

*Proof.*

$$\begin{aligned} (\vec{w}, \vec{w} \cdot \nabla \vec{w})_{\Omega} &= (u, \vec{w} \cdot \nabla u)_{\Omega} + (v, \vec{w} \cdot \nabla v)_{\Omega} \\ &= (u \vec{w}, \nabla u)_{\Omega} + (v \vec{w}, \nabla v)_{\Omega} \\ &= (u w_n, u)_{\partial\Omega} - (\nabla \cdot (u \vec{w}), u)_{\Omega} + (v w_n, v)_{\partial\Omega} - (\nabla \cdot (v \vec{w}), v)_{\Omega} \\ &= (w_n, u^2 + v^2)_{\partial\Omega} - (u, \nabla \cdot (u \vec{w}))_{\Omega} - (v, \nabla \cdot (v \vec{w}))_{\Omega} \\ &= (w_n, |\vec{w}|^2)_{\partial\Omega} - (u, \nabla \cdot (u \vec{w}))_{\Omega} - (v, \nabla \cdot (v \vec{w}))_{\Omega} \\ &= (w_n, |\vec{w}|^2)_{\partial\Omega} - \left( \vec{w}, \nabla \cdot (\vec{w} \vec{w}^{\top}) \right)_{\Omega} . \end{aligned}$$

In the above equalities we have only used Green's identity and the definitions of the inner products. □

Finally we restate Proposition 3 for the split form incompressible Euler equations (3), and prove it without reference to the chain rule.

**Proposition 4.** *The split form incompressible Euler equations (3) imply that*

$$\frac{d}{dt} \|\vec{w}\|_{\Omega}^2 = - (w_n, |\vec{w}|^2)_{\partial\Omega} - 2 (w_n, p)_{\partial\Omega} .$$

*Proof.*

$$\begin{aligned} \frac{d}{dt} \|\vec{w}\|_{\Omega}^2 &= - (\vec{w}, \vec{w} \cdot \nabla \vec{w})_{\Omega} - \left( \vec{w}, \nabla \cdot (\vec{w} \vec{w}^{\top}) \right)_{\Omega} - 2 (\vec{w}, \nabla p)_{\Omega} \\ &= - (w_n, |\vec{w}|^2)_{\partial\Omega} - 2 (\vec{w}, \nabla p)_{\Omega} && \text{(Lemma 2)} \\ &= - (w_n, |\vec{w}|^2)_{\partial\Omega} - 2 (w_n, p)_{\partial\Omega} + 2 (\nabla \cdot \vec{w}, p)_{\Omega} && \text{(Green's identity)} \\ &= - (w_n, |\vec{w}|^2)_{\partial\Omega} - 2 (w_n, p)_{\partial\Omega} && \text{(Zero divergence)} \end{aligned}$$

□

**Remark 2.** *The proof of Proposition 4 relies only on properties that can be replicated discretely, namely Green's identity and algebraic properties of the inner product. In fact, as we shall see in Section 5, the energy analysis of our discretization of the split form incompressible Euler equations (3) is identical to the proofs of Lemma 2 and Proposition 4.*

If the boundary terms in Proposition 4 can be bounded in terms of data, then a bound on the kinetic energy can be found at any given time in terms of initial and boundary data.

**Proposition 5.** *Assume that  $\vec{w}$  and  $p$  are smooth solutions to (3) and that  $\vec{w} = \vec{f}$  for  $t = 0$ , where  $\vec{f}$  is the initial data. Furthermore, assume that  $-(w_n, |\vec{w}|^2)_{\partial\Omega} - 2 (w_n, p)_{\partial\Omega} = g$ . Then for  $T > 0$ ,*

$$\|\vec{w}\|_{\Omega}^2(T) = \|\vec{f}\|_{\Omega}^2 + \int_0^T g(t) dt .$$

*Proof.* By Proposition 4,

$$\int_0^T \frac{d}{dt} \|\vec{w}\|_{\Omega}^2 dt = \|\vec{w}\|_{\Omega}^2(T) - \|\vec{w}\|_{\Omega}^2(0) = \int_0^T g(t) dt .$$

Hence,

$$\|\vec{w}\|_{\Omega}^2(T) = \|\vec{f}\|_{\Omega}^2 + \int_0^T g(t) dt .$$

□



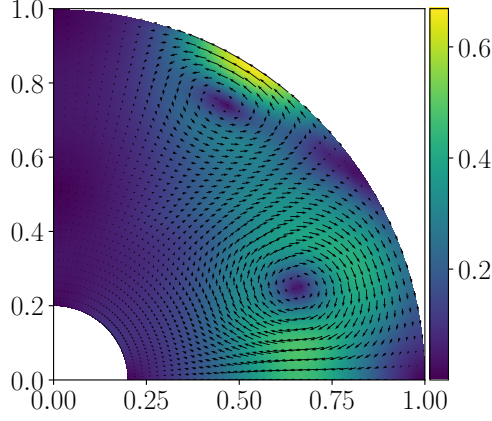


Figure 2: Flow inside an annulus sector with the boundary condition  $w_n = 0$  imposed on all four boundaries.

#### 4. Boundary conditions

A number of different boundary conditions can be used for various modeling goals. Below we list the ones that are used in this paper. We select, for each of the four boundaries  $\Gamma_d$  of  $\Omega$ , one of the conditions listed below. All conditions (except for the inflow condition in Section 4.4) are built to either dissipate or conserve kinetic energy. More precisely, we choose to impose boundary conditions at  $\Gamma_d$  which lead to

$$-(w_n, |\vec{w}|^2)_{\Gamma_d} - 2(w_n, p)_{\Gamma_d} \leq 0. \quad (4)$$

Note that

$$(w_n, |\vec{w}|^2)_{\partial\Omega} + 2(w_n, p)_{\partial\Omega} = \sum_d \left( (w_n, |\vec{w}|^2)_{\Gamma_d} + 2(w_n, p)_{\Gamma_d} \right).$$

Hence, if (4) is satisfied for all four boundaries, Proposition 5 implies that the kinetic energy does not grow.

As we shall see in Section 6, the continuous bounds are mimicked in the discrete setting by using encapsulated SBP operators and weakly imposed boundary conditions.

##### 4.1. Solid wall

To model a solid wall we may impose vanishing normal velocity,  $w_n = 0$ , so that the boundary terms in (4) cancel. Figure 2 shows a simulation of a cavity flow using this boundary condition on the entire boundary  $\partial\Omega$ .

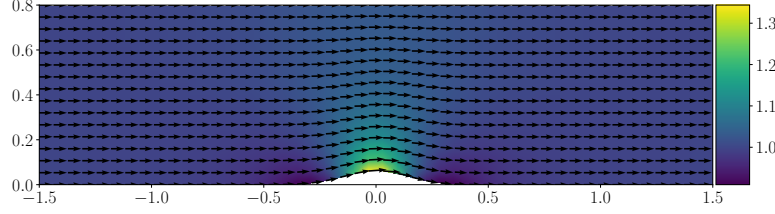


Figure 3: Flow inside a channel with a bump.

#### 4.2. Pressure outflow

If  $w_n > 0$  on  $\Gamma_d$  it suffices to specify the pressure  $p$ . Indeed, if  $w_n > 0$  and a nonnegative pressure  $p$  is imposed, then  $-(w_n, |\vec{w}|^2)_{\Gamma_d} - 2(w_n, p)_{\Gamma_d} \leq 0$ . This can be used to model a priori known outflow boundaries, for example a horizontal flow inside a channel (see Figure 3).

#### 4.3. Stabilized pressure outflow

Only prescribing the pressure as described in Section 4.2 will lead to instability if the boundary changes from an outflow to an inflow, since  $-(w_n, |\vec{w}|^2)_{\Gamma_d} - 2(w_n, p)_{\Gamma_d} > 0$  if  $w_n < 0$ . Consider instead the boundary condition

$$\begin{aligned} h(w_n)w_n^2 + 2p &= 0 \\ h(w_n)w_s^2 &= 0 \end{aligned} \quad \text{where} \quad h(w_n) = \begin{cases} 1 & \text{if } w_n < 0 \\ 0 & \text{if } w_n \geq 0 \end{cases}. \quad (5)$$

Here,  $w_s = -n_y u + n_x v$  is the tangential velocity. If  $w_n \geq 0$ , then (5) is reduced to the pressure outflow condition. If  $w_n < 0$ , then the condition  $w_n^2 + p = 0$  serves to attract the fluid since it decreases the pressure at the boundary, while the condition  $w_s = 0$  restricts tangential flow (see Figure 4). Furthermore, if  $w_n < 0$ , imposing (5) on  $\Gamma_d$  yields

$$-(w_n, |\vec{w}|^2)_{\Gamma_d} - 2(w_n, p)_{\Gamma_d} = -(w_n, w_n^2 + w_s^2 + 2p)_{\Gamma_d} = 0.$$

#### 4.4. Inflow

To model an inflow boundary we may impose the conditions  $w_n = g_n$  and  $w_s = g_s$ , specifying the normal and tangential velocity. Note that this implies that

$$-(w_n, |\vec{w}|^2)_{\Gamma_d} - 2(w_n, p)_{\Gamma_d} = -(g_n, g_n^2 + g_s^2)_{\Gamma_d} - 2(g_n, p)_{\Gamma_d},$$

which is only bounded in terms of data under the assumption that  $p$  is bounded. To the best of our knowledge, there is no proof that  $p$  is bounded. Our numerical experiments do not exhibit any issues related to uncontrolled growth in  $p$ , but the inflow boundary condition should still be used with care since there is no formal guarantee that the kinetic energy will not grow unexpectedly.

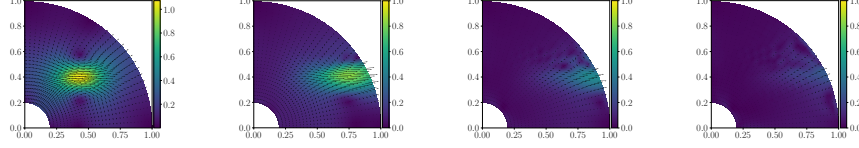


Figure 4: A simulation initialized with  $\mathbf{v} = \mathbf{p} = \mathbf{0}$ , and a Gaussian bell in  $\mathbf{u}$ , resulting in swirling fluid exiting the domain through a boundary using the stabilized pressure condition.

## 5. Discretization

Consider a spatial discretization of the incompressible Euler equations (3).

$$\vec{\mathbf{w}}_t + \frac{1}{2} \vec{\mathbf{w}} \cdot \nabla \vec{\mathbf{w}} + \frac{1}{2} \nabla \cdot (\vec{\mathbf{w}} \vec{\mathbf{w}}^\dagger) = -\nabla \mathbf{p} \quad (6a)$$

$$\nabla \cdot \vec{\mathbf{w}} = 0, \quad (6b)$$

where  $\vec{\mathbf{w}}$  and  $\mathbf{p}$  are time dependent grid functions. The form (6) is particularly convenient for discrete energy analysis since the continuous energy analysis can be replicated. Indeed, both Lemma 2 and Proposition 4 hold discretely (the proofs are identical, simply replace the normal letters by bold letters), resulting in the discrete energy:

**Proposition 6.** *The semidiscrete incompressible Euler equations (6) imply that*

$$\frac{d}{dt} \|\vec{\mathbf{w}}\|_{\Omega_h}^2 = -(\mathbf{w}_n, |\vec{\mathbf{w}}|^2)_{\partial\Omega_h} - 2(\mathbf{w}_n, \mathbf{p})_{\partial\Omega_h} + 2(\mathbf{p}, \nabla \cdot \vec{\mathbf{w}})_{\Omega_h}. \quad (7)$$

**Remark 3.** *We do not omit the term  $2(\mathbf{p}, \nabla \cdot \vec{\mathbf{w}})_{\Omega_h}$  in Proposition 6 even though it vanishes by (6b). This is because it is sometimes useful to make small perturbations to the discrete divergence (i.e. allow for local nonzero divergence at a boundary) in order to cancel the pressure term  $-2(\mathbf{w}_n, \mathbf{p})_{\Gamma_d}$  (see the discrete boundary treatment in Section 6).*

For implementation purposes it is better to write (6) in more explicit form as

$$\begin{bmatrix} \mathbf{u}_t \\ \mathbf{v}_t \\ \mathbf{0} \end{bmatrix} + \mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{p}) = \mathbf{0}, \quad (8)$$

where

$$\mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{p}) = [\mathcal{L}_1 \quad \mathcal{L}_2 \quad \mathcal{L}_3]^\dagger$$

and

$$\begin{aligned} \mathcal{L}_1 &= \frac{1}{2} u D_x u + \frac{1}{2} D_x u u + \frac{1}{2} v D_y u + \frac{1}{2} D_y u v + D_x p \\ \mathcal{L}_2 &= \frac{1}{2} v D_y v + \frac{1}{2} D_y v v + \frac{1}{2} u D_x v + \frac{1}{2} D_x u v + D_y p \\ \mathcal{L}_3 &= D_x u + D_y v. \end{aligned}$$

The discretization (8) defines a set of differential algebraic equations. These equations are not yet ready to be evolved in time since they do not take into account any boundary conditions. However, the Jacobian of the discrete spatial operator in (6) is a key component in any implicit timestepping scheme. The Jacobian can be derived using the following observations. Let  $\mathbf{f}, \mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  be vector-valued differentiable functions and define the diagonal matrices

$$\underline{\mathbf{f}} = \begin{pmatrix} f_1 & & & \\ & f_2 & & \\ & & \ddots & \\ & & & f_n \end{pmatrix} \text{ and } \underline{\mathbf{g}} = \begin{pmatrix} g_1 & & & \\ & g_2 & & \\ & & \ddots & \\ & & & g_n \end{pmatrix}.$$

It is readily seen that the Jacobian  $J_{\mathbf{f}\mathbf{g}}$  of the elementwise product  $\mathbf{f}\mathbf{g}$  satisfies the following product rule:

$$J_{\mathbf{f}\mathbf{g}} = \underline{\mathbf{f}}J_{\mathbf{g}} + \underline{\mathbf{g}}J_{\mathbf{f}}. \quad (9)$$

Furthermore, the Jacobian is linear in the sense that for any  $n$ -by- $n$  matrices  $A$  and  $B$  we have

$$J_{A\mathbf{f}+B\mathbf{g}} = AJ_{\mathbf{f}} + BJ_{\mathbf{g}}. \quad (10)$$

Applying these Jacobian properties to  $\mathcal{L}$  yields the following proposition.

**Proposition 7.** *The Jacobian  $J_{\mathcal{L}}$  of the discrete operator  $\mathcal{L}$  in (8) is*

$$J_{\mathcal{L}} = \begin{pmatrix} \nabla_{\mathbf{u}}\mathcal{L}_1 & \nabla_{\mathbf{v}}\mathcal{L}_1 & \nabla_{\mathbf{p}}\mathcal{L}_1 \\ \nabla_{\mathbf{u}}\mathcal{L}_2 & \nabla_{\mathbf{v}}\mathcal{L}_2 & \nabla_{\mathbf{p}}\mathcal{L}_2 \\ \nabla_{\mathbf{u}}\mathcal{L}_3 & \nabla_{\mathbf{v}}\mathcal{L}_3 & \nabla_{\mathbf{p}}\mathcal{L}_3 \end{pmatrix}$$

where

$$\begin{aligned} \nabla_{\mathbf{u}}\mathcal{L}_1 &= \frac{1}{2} (\underline{\mathbf{u}}D_{\mathbf{x}} + \underline{D_{\mathbf{x}}}\mathbf{u} + 2D_{\mathbf{x}}\underline{\mathbf{u}} + \underline{\mathbf{v}}D_{\mathbf{y}} + D_{\mathbf{y}}\underline{\mathbf{v}}) \\ \nabla_{\mathbf{v}}\mathcal{L}_1 &= \frac{1}{2} (\underline{D_{\mathbf{y}}}\mathbf{u} + D_{\mathbf{y}}\underline{\mathbf{u}}) \\ \nabla_{\mathbf{p}}\mathcal{L}_1 &= D_{\mathbf{x}} \\ \nabla_{\mathbf{u}}\mathcal{L}_2 &= \frac{1}{2} (\underline{D_{\mathbf{x}}}\mathbf{v} + D_{\mathbf{x}}\underline{\mathbf{v}}) \\ \nabla_{\mathbf{v}}\mathcal{L}_2 &= \frac{1}{2} (\underline{\mathbf{v}}D_{\mathbf{y}} + \underline{D_{\mathbf{y}}}\mathbf{v} + 2D_{\mathbf{y}}\underline{\mathbf{v}} + \underline{\mathbf{u}}D_{\mathbf{x}} + D_{\mathbf{x}}\underline{\mathbf{u}}) \\ \nabla_{\mathbf{p}}\mathcal{L}_2 &= D_{\mathbf{y}} \\ \nabla_{\mathbf{u}}\mathcal{L}_3 &= D_{\mathbf{x}} \\ \nabla_{\mathbf{v}}\mathcal{L}_3 &= D_{\mathbf{y}} \\ \nabla_{\mathbf{p}}\mathcal{L}_3 &= \mathbf{0}. \end{aligned}$$

*Proof.* Let us find the first couple of terms in  $\nabla_{\mathbf{u}}\mathcal{L}_1$ . The proofs for the other blocks are similar. Consider the first term  $\frac{1}{2}\underline{\mathbf{u}}D_{\mathbf{x}}\mathbf{u}$  of  $\mathcal{L}_1$ . Let  $\mathbf{f} = \mathbf{u}$  and

$\mathbf{g} = D_x \mathbf{u}$ . Then  $J_f = \mathbf{I}$  and  $J_g = D_x$ . Hence, by (9),

$$\nabla_{\mathbf{u}} \left( \frac{1}{2} \mathbf{u} D_x \mathbf{u} \right) = \frac{1}{2} (\underline{\mathbf{u}} D_x + D_x \underline{\mathbf{u}}).$$

Similarly, for the second term  $\frac{1}{2} D_x \mathbf{u} \mathbf{u}$  of  $\mathcal{L}_1$ , we set  $\mathbf{f} = \mathbf{u}$  and  $\mathbf{g} = \mathbf{u}$ . Then  $J_f = J_g = \mathbf{I}$  and by (9) and (10) we get

$$\nabla_{\mathbf{u}} \left( \frac{1}{2} D_x \mathbf{u} \mathbf{u} \right) = D_x \underline{\mathbf{u}}.$$

□

## 6. Discrete boundary conditions

In this section we describe how to implement the boundary conditions from Section 4 using penalty terms. The main idea is to associate to each boundary a penalty operator  $\mathcal{S}(\mathbf{u}, \mathbf{v}, \mathbf{p})$  which is added to the Euler spatial operator  $\mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{p})$ . Similarly, the Jacobian of  $\mathcal{S}$  is added to the Jacobian of  $\mathcal{L}$ , since  $J_{\mathcal{L}+\mathcal{S}} = J_{\mathcal{L}} + J_{\mathcal{S}}$ .

For a given boundary condition  $H(u, v, p) = g$ , a penalty operator  $\mathcal{S}$  will typically have the form  $\mathcal{S} = [\mathcal{S}_1 \quad \mathcal{S}_2 \quad \mathcal{S}_3]^\dagger$ , where  $\mathcal{S}_i = C_i(H(\mathbf{u}, \mathbf{v}, \mathbf{p}) - g)$ , and  $C_i$  is a penalty coefficient. When a boundary condition is not satisfied, its penalty operator  $\mathcal{S}$  will add appropriate contributions to  $\mathcal{L}$ , pushing the solution toward compliance with the boundary condition. A penalty operator is ideally built such that it cancels the growth terms in (7), or replaces them with data, so we know that the kinetic energy cannot grow uncontrollably.

**Definition 2.** A penalty operator  $\mathcal{S}^d = [\mathcal{S}_1^d \quad \mathcal{S}_2^d \quad \mathcal{S}_3^d]^\dagger$  for the boundary  $\Gamma_d$  is called stable if

$$PT + BT \geq g_d,$$

where

$$PT = 2 \left( \mathbf{u}, \mathcal{S}_1^d \right)_{\Omega_h} + 2 \left( \mathbf{v}, \mathcal{S}_2^d \right)_{\Omega_h} + \left( \mathbf{p}, \mathcal{S}_3^d \right)_{\Omega_h},$$

$$BT = (\mathbf{w}_n, |\vec{\mathbf{w}}|^2)_{\Gamma_d} + 2 (\mathbf{w}_n, \mathbf{p})_{\Gamma_d},$$

and  $g_d$  is a bounded time dependent function. Furthermore, if  $PT + BT = 0$ , then  $\mathcal{S}^d$  is called conservative. Finally, if  $PT + BT > 0$ , then  $\mathcal{S}^d$  is called dissipative.

**Remark 4.** Note that any conservative or dissipative penalty operator is also stable.

Definition 2 is motivated by the following proposition, ensuring that the discretization (8), combined with stable penalty operators, is stable with respect to the kinetic energy.

**Proposition 8.** Assume that for each boundary  $\Gamma_d$  we have a stable penalty operator  $\mathcal{S}^d$ . If  $\mathbf{u}, \mathbf{v}, \mathbf{p}$  solves the system

$$\begin{bmatrix} \mathbf{u}_t \\ \mathbf{v}_t \\ \mathbf{0} \end{bmatrix} + \mathcal{L} + \sum_d \mathcal{S}^d = \mathbf{0},$$

then

$$\frac{d}{dt} \|\vec{\mathbf{w}}\|_{\Omega_h}^2 \leq g,$$

where  $g$  is a bounded time dependent function.

*Proof.* By using Proposition 6, and adding the contribution from the penalty terms, we get that

$$\begin{aligned} \frac{d}{dt} \|\vec{\mathbf{w}}\|_{\Omega_h}^2 &= - \sum_d (\mathbf{w}_n, |\vec{\mathbf{w}}|^2 + 2p)_{\Gamma_d} \\ &\quad - 2 \sum_d \left( \left( \mathbf{u}, \mathcal{S}_1^d \right)_{\Omega_h} + \left( \mathbf{v}, \mathcal{S}_2^d \right)_{\Omega_h} + \left( \mathbf{p}, \mathcal{S}_3^d \right)_{\Omega_h} \right) \\ &= - \sum_d BT + PT \leq \sum_d g_d. \end{aligned} \quad (11)$$

In (11), the term  $\nabla \cdot \vec{\mathbf{w}}$  has been replaced by  $-\sum_d \mathcal{S}_3^d$ . The last inequality holds since the penalty operator is stable.  $\square$

Let us start by introducing the concept of a discrete lifting operator  $\mathbf{L}_d = \mathbf{P}^{-1} \mathbf{P}_d$  associated to the boundary  $\Gamma_d$ . The lifting operator  $\mathbf{L}_d$  is a sparse diagonal matrix with nonzero entries in the positions corresponding to the boundary  $\Gamma_d$ , which has the property that

$$\left( \mathbf{u}, \mathbf{L}_d \vec{\phi} \right)_{\Omega_h} = \left( \mathbf{u}, \vec{\phi} \right)_{\Gamma_d}. \quad (12)$$

Lifting operators are used in all our penalty terms because they allow us to convert integrals over  $\Omega_h$  to integrals over  $\Gamma_d$  in the energy analysis.

Below we list Penalty Operator/Jacobian pairs  $(\mathcal{S}, J_{\mathcal{S}})$  for the boundary conditions described in Section 4, on the form

$$\mathcal{S} = [\mathcal{S}_1 \quad \mathcal{S}_2 \quad \mathcal{S}_3]^\dagger, \quad J_{\mathcal{S}} = \begin{pmatrix} \nabla_{\mathbf{u}} \mathcal{S}_1 & \nabla_{\mathbf{v}} \mathcal{S}_1 & \nabla_{\mathbf{p}} \mathcal{S}_1 \\ \nabla_{\mathbf{u}} \mathcal{S}_2 & \nabla_{\mathbf{v}} \mathcal{S}_2 & \nabla_{\mathbf{p}} \mathcal{S}_2 \\ \nabla_{\mathbf{u}} \mathcal{S}_3 & \nabla_{\mathbf{v}} \mathcal{S}_3 & \nabla_{\mathbf{p}} \mathcal{S}_3 \end{pmatrix}.$$

NOTE: For the penalty operators considered in this paper, all blocks in  $J_{\mathcal{S}}$  are diagonal  $N_x N_y$ -by- $N_x N_y$  matrices. We shall therefore allow ourselves to omit the underline-notation and let it be understood that column vectors represent diagonal matrices in this context.

### 6.1. Solid wall

*Penalty operator*

$$\mathcal{S}_1 = -\frac{1}{2}\mathbf{L}_d\mathbf{u}\mathbf{w}_n, \quad \mathcal{S}_2 = -\frac{1}{2}\mathbf{L}_d\mathbf{v}\mathbf{w}_n, \quad \text{and} \quad \mathcal{S}_3 = -\mathbf{L}_d\mathbf{w}_n.$$

*Jacobian*

$$\nabla_p \mathcal{S}_1 = \nabla_p \mathcal{S}_2 = \nabla_p \mathcal{S}_3 = \mathbf{0} \quad \text{and}$$

$$\begin{aligned} \nabla_u \mathcal{S}_1 &= -\frac{1}{2}\mathbf{L}_d(\mathbf{u}\mathbf{n}_x + \mathbf{w}_n), & \nabla_u \mathcal{S}_2 &= -\frac{1}{2}\mathbf{L}_d\mathbf{v}\mathbf{n}_x, & \nabla_u \mathcal{S}_3 &= -\mathbf{L}_d\mathbf{n}_x, \\ \nabla_v \mathcal{S}_1 &= -\frac{1}{2}\mathbf{L}_d\mathbf{u}\mathbf{n}_y, & \nabla_v \mathcal{S}_2 &= -\frac{1}{2}\mathbf{L}_d(\mathbf{v}\mathbf{n}_y + \mathbf{w}_n), & \nabla_v \mathcal{S}_3 &= -\mathbf{L}_d\mathbf{n}_y. \end{aligned}$$

*Stability*

We have  $(2\mathbf{u}, \mathcal{S}_1)_{\Omega_h} = -(\mathbf{u}^2, \mathbf{w}_n)_{\Gamma_d}$ ,  $(2\mathbf{v}, \mathcal{S}_2)_{\Omega_h} = -(\mathbf{v}^2, \mathbf{w}_n)_{\Gamma_d}$ , and  $(\mathbf{p}, \mathcal{S}_3)_{\Omega_h} = -(\mathbf{p}, \mathbf{w}_n)_{\Gamma_d}$ . Hence  $\mathcal{S}$  is conservative by Definition 2.

### 6.2. Pressure outflow

*Penalty operator*

$$\mathcal{S}_1 = -\mathbf{L}_d\mathbf{n}_x\mathbf{p}, \quad \mathcal{S}_2 = -\mathbf{L}_d\mathbf{n}_y\mathbf{p}, \quad \text{and} \quad \mathcal{S}_3 = \mathbf{0}.$$

*Jacobian*

$$\nabla_p \mathcal{S}_1 = -\mathbf{L}_d\mathbf{n}_x, \quad \nabla_p \mathcal{S}_2 = -\mathbf{L}_d\mathbf{n}_y, \quad \text{and the rest are } \mathbf{0}.$$

*Stability*

We have  $(2\mathbf{u}, \mathcal{S}_1)_{\Omega_h} = -(2\mathbf{u}\mathbf{n}_x, \mathbf{p})_{\Gamma_d}$ ,  $(2\mathbf{v}, \mathcal{S}_2)_{\Omega_h} = -(2\mathbf{v}\mathbf{n}_y, \mathbf{p})_{\Gamma_d}$ , and  $(\mathbf{p}, \mathcal{S}_3)_{\Omega_h} = 0$ . Hence, if  $\mathbf{w}_n \geq \mathbf{0}$ , then  $\mathcal{S}$  is dissipative by Definition 2.

### 6.3. Stabilized pressure outflow

Note: For this boundary operator to be differentiable we approximate the Heaviside function  $h$  in Section 4.3 by a sigmoid function  $h(w_n) = 1/(e^{w_n} + 1)$ .

*Penalty operator*

$$\begin{aligned} \mathcal{S}_1 &= -\mathbf{L}_d\mathbf{n}_x (h(\mathbf{w}_n)|\vec{\mathbf{w}}|^2 + \mathbf{p}), \\ \mathcal{S}_2 &= -\mathbf{L}_d\mathbf{n}_y (h(\mathbf{w}_n)|\vec{\mathbf{w}}|^2 + \mathbf{p}), \\ \mathcal{S}_3 &= \mathbf{0}. \end{aligned}$$

*Jacobian*

$$\begin{aligned}
\nabla_{\mathbf{u}} \mathcal{S}_1 &= -L_d \mathbf{n}_x (\nabla_{\mathbf{u}} h(\mathbf{w}_n) |\vec{\mathbf{w}}|^2 + 2h(\mathbf{w}_n)(\mathbf{w}_n \mathbf{n}_x - \mathbf{w}_s \mathbf{n}_y)) \\
\nabla_{\mathbf{v}} \mathcal{S}_1 &= -L_d \mathbf{n}_x (\nabla_{\mathbf{v}} h(\mathbf{w}_n) |\vec{\mathbf{w}}|^2 + 2h(\mathbf{w}_n)(\mathbf{w}_n \mathbf{n}_y + \mathbf{w}_s \mathbf{n}_x)) \\
\nabla_{\mathbf{p}} \mathcal{S}_1 &= -L_d \mathbf{n}_x \\
\nabla_{\mathbf{u}} \mathcal{S}_2 &= -L_d \mathbf{n}_y (\nabla_{\mathbf{u}} h(\mathbf{w}_n) |\vec{\mathbf{w}}|^2 + 2h(\mathbf{w}_n)(\mathbf{w}_n \mathbf{n}_x - \mathbf{w}_s \mathbf{n}_y)) \\
\nabla_{\mathbf{v}} \mathcal{S}_2 &= -L_d \mathbf{n}_y (\nabla_{\mathbf{v}} h(\mathbf{w}_n) |\vec{\mathbf{w}}|^2 + 2h(\mathbf{w}_n)(\mathbf{w}_n \mathbf{n}_y + \mathbf{w}_s \mathbf{n}_x)) \\
\nabla_{\mathbf{p}} \mathcal{S}_2 &= -L_d \mathbf{n}_y
\end{aligned}$$

where

$$\begin{aligned}
\nabla_{\mathbf{u}} h(\mathbf{w}_n) &= -\frac{e^{\mathbf{w}_n}}{(e^{\mathbf{w}_n} + 1)^2} \mathbf{n}_x, \\
\nabla_{\mathbf{v}} h(\mathbf{w}_n) &= -\frac{e^{\mathbf{w}_n}}{(e^{\mathbf{w}_n} + 1)^2} \mathbf{n}_y.
\end{aligned}$$

Furthermore,

$$\nabla_{\mathbf{u}} \mathcal{S}_3 = \nabla_{\mathbf{v}} \mathcal{S}_3 = \nabla_{\mathbf{p}} \mathcal{S}_3 = \mathbf{0}.$$

*Stability*

We have

$$\begin{aligned}
(2\mathbf{u}, \mathcal{S}_1)_{\Omega_h} &= -(2\mathbf{u} \mathbf{n}_x, h(\mathbf{w}_n) |\vec{\mathbf{w}}|^2 + \mathbf{p})_{\Gamma_d} \\
(2\mathbf{v}, \mathcal{S}_2)_{\Omega_h} &= -(2\mathbf{v} \mathbf{n}_y, h(\mathbf{w}_n) |\vec{\mathbf{w}}|^2 + \mathbf{p})_{\Gamma_d} \\
(\mathbf{p}, \mathcal{S}_3)_{\Omega_h} &= 0.
\end{aligned}$$

Hence  $\mathcal{S}$  is conservative if  $\mathbf{w}_n \leq 0$  and dissipative otherwise.

*6.4. Inflow*

*Penalty operator*

$$\begin{aligned}
\mathcal{S}_1 &= -L_d \mathbf{n}_x \mathbf{w}_n (\mathbf{w}_n - \mathbf{g}_n) + L_d \mathbf{n}_y \mathbf{w}_n (\mathbf{w}_s - \mathbf{g}_s), \\
\mathcal{S}_2 &= -L_d \mathbf{n}_y \mathbf{w}_n (\mathbf{w}_n - \mathbf{g}_n) - L_d \mathbf{n}_x \mathbf{w}_n (\mathbf{w}_s - \mathbf{g}_s), \\
\mathcal{S}_3 &= -L_d (\mathbf{w}_n - \mathbf{g}_n).
\end{aligned}$$

*Jacobian*

$$\begin{aligned}
\nabla_{\mathbf{u}} \mathcal{S}_1 &= -L_d (\mathbf{n}_x^2 (2\mathbf{w}_n - \mathbf{g}_n) + \mathbf{n}_y^2 \mathbf{w}_n + \mathbf{n}_x \mathbf{n}_y (\mathbf{w}_s - \mathbf{g}_t)) , \\
\nabla_{\mathbf{v}} \mathcal{S}_1 &= L_d (\mathbf{n}_y^2 (\mathbf{w}_s - \mathbf{g}_t) - \mathbf{n}_x \mathbf{n}_y (\mathbf{w}_n - \mathbf{g}_n)) , \\
\nabla_{\mathbf{u}} \mathcal{S}_2 &= -L_d (\mathbf{n}_x^2 (\mathbf{w}_s - \mathbf{g}_t) + \mathbf{n}_x \mathbf{n}_y (\mathbf{w}_n - \mathbf{g}_n)) , \\
\nabla_{\mathbf{v}} \mathcal{S}_2 &= -L_d (\mathbf{n}_y^2 (2\mathbf{w}_n - \mathbf{g}_n) + \mathbf{n}_x^2 \mathbf{w}_n + \mathbf{n}_x \mathbf{n}_y (\mathbf{w}_s - \mathbf{g}_t)) , \\
\nabla_{\mathbf{u}} \mathcal{S}_3 &= -L_d \mathbf{n}_x , \\
\nabla_{\mathbf{v}} \mathcal{S}_3 &= -L_d \mathbf{n}_y ,
\end{aligned}$$

and the rest are  $\mathbf{0}$ .



### Stability

As seen in Section 6.4, imposing Dirichlet conditions at an inflow boundary do not result in a bound. We get

$$PT + BT = (\mathbf{w}_n, -\mathbf{w}_n^2 + 2\mathbf{w}_n \mathbf{g}_n)_{\Gamma_d} + (\mathbf{w}_n, -\mathbf{w}_s^2 + 2\mathbf{w}_s \mathbf{g}_n)_{\Gamma_d}$$

## 7. Numerical experiments

To evolve the system in time, we will use the implicit Backward Euler method. For an ordinary differential equation of the form

$$\mathcal{M}\chi_t + \mathcal{H}(\chi) = 0,$$

where  $\chi$  is a grid function and  $\mathcal{M}$  is a constant matrix, the Backward Euler schemes becomes

$$\frac{\mathcal{M}(\chi^{i+1} - \chi^i)}{\Delta t} + \mathcal{H}(\chi^{i+1}) = 0. \quad (13)$$

In (13),  $\Delta t$  is the size of the time step and the indices  $i$  and  $i+1$  are the solution at time level  $i$  and  $i+1$ , respectively.

In order to obtain  $\chi^{i+1}$ , the system of nonlinear equations in (13) needs to be solved. One strategy is to first form the function

$$\mathcal{F}(\chi^{i+1}) = \frac{\mathcal{M}(\chi^{i+1} - \chi^i)}{\Delta t} + \mathcal{H}(\chi^{i+1}). \quad (14)$$

If we find a vector  $\chi^*$  such that  $\mathcal{F}(\chi^*) = 0$ , then  $\chi^{i+1} = \chi^*$ . To solve (14), we can employ Newton's method [2], which is described in Algorithm 1. This allows us to solve a sequence of linear system of equations to obtain an approximation of  $\chi^*$ .

**Data:** Initial guess,  $\chi^0$ , and tolerance  $tol$

**Result:** An approximation of  $\chi^*$ , where  $\mathcal{F}(\chi^*) = 0$

```

for  $i = 0, 1, 2, \dots$  do
    solve  $J_{\mathcal{F}}(\chi^i) \mathbf{h}^i = -\mathcal{F}(\chi^i)$ 
    set  $\chi^{i+1} = \chi^i + \mathbf{h}^i$ 
    if  $\|\mathcal{F}(\chi^{i+1})\| < tol$  then
        | return  $\chi^{i+1}$ 
    end
end

```

**end**

**Algorithm 1:** Newton's method.

For the Euler equations,  $\mathcal{H}(\chi) = \mathcal{L}(\chi) + \mathcal{S}(\chi)$ , where

$$\chi = [\mathbf{u} \quad \mathbf{v} \quad \mathbf{p}]^\top \quad \text{and} \quad \mathcal{M} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

Hence, (13) becomes

$$\frac{1}{\Delta t} \left( \begin{bmatrix} \mathbf{u}^{i+1} \\ \mathbf{v}^{i+1} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{u}^i \\ \mathbf{v}^i \\ \mathbf{0} \end{bmatrix} \right) + \mathcal{L}(\chi^{i+1}) + \mathcal{S}(\chi^{i+1}) = 0 \quad (15)$$

and (14) turns into

$$\mathcal{F}(\chi^{i+1}) = \frac{1}{\Delta t} \left( \begin{bmatrix} \mathbf{u}^{i+1} \\ \mathbf{v}^{i+1} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{u}^i \\ \mathbf{v}^i \\ \mathbf{0} \end{bmatrix} \right) + \mathcal{L}(\chi^{i+1}) + \mathcal{S}(\chi^{i+1}). \quad (16)$$

Furthermore,  $J_{\mathcal{H}}(\chi) = J_{\mathcal{L}}(\chi) + J_{\mathcal{S}}(\chi)$ , which yields

$$J_{\mathcal{F}}(\chi) = \frac{1}{\Delta t} \mathcal{M} + J_{\mathcal{L}}(\chi) + J_{\mathcal{S}}(\chi),$$

to be used in the Newton iterations. The explicit forms of  $J_{\mathcal{L}}(\chi)$  and  $J_{\mathcal{S}}(\chi)$  are found in 5 and Section 6, respectively.

In all computations below, the initial guess is the solution from the previous time step and the tolerance is set to  $10^{-12}$ .

### 7.1. The order of accuracy

The method of manufactured solution is used to verify the implementation. For the SBP-operators SBP21 and SBP42, the expected orders of accuracy are 2 and 3, respectively [3]. The manufactured solution is

$$\begin{aligned} u &= 1 + 0.5 \sin(2\pi x - t) \sin(2\pi y - t) \\ v &= 1 + 0.5 \cos(2\pi x - t) \cos(2\pi y - t) \\ p &= 0.5 \sin(2\pi x - t) \cos(2\pi y - t) \end{aligned}$$

and the domain is illustrated in Figure 3. Since the west and south sides of the domain are inflow boundaries, both  $w_n$  and  $w_t$  are specified (see Section 4.4 and Section 6.4 for details). At the east and north sides, which are outflow boundaries, only the pressure is prescribed (see Section 4.2 and Section 6.2 for details).

The step size is chosen to be  $\Delta t = 10^{-4}$  and the computations are terminated at  $t = 1$ . Then, we compute the pointwise error,  $\mathbf{e}$  and its  $L_2$ -norm  $\|\mathbf{e}\|_{\Omega_h}$ . The convergence rate is given by  $r = \log(\|\mathbf{e}\|_i / \|\mathbf{e}\|_j) / \log((j-1)/(i-1))$ , where  $i$  and  $j$  refer to the number of grid points in both spatial dimensions. The results are presented in Table 1 and agree well with theory.

operator	SBP21		SBP42	
N	$\ e\ $	$r$	$\ e\ $	$r$
11	1.60e+00	–	5.19e-01	–
21	1.84e-01	3.12	8.93e-02	2.54
41	4.25e-02	2.12	1.32e-02	2.76
61	2.84e-02	2.05	3.92e-03	2.98
81	1.03e-02	2.02	1.67e-03	2.97
101	6.58e-03	2.01	8.61e-04	2.95
Theoretical		2		3

Table 1: Error and convergence rate.

## 8. Summary and conclusions

We derived an explicit expression for the Jacobian of the discretization of the incompressible Euler equations. Due to the use of encapsulated difference operators, the extension to a curved grid is straight forward. Different boundary conditions such as the solid wall, the Dirichlet inflow and pressure outflow conditions have been used and the Jacobian for all these sets of boundary conditions were also presented. Lastly, the numerical method has been illustrated for different cases and also verified by the method of manufactured solutions.

## References

- [1] Oskar Ålund and Jan Nordström. Encapsulated high order difference operators on curvilinear non-conforming grids. *Journal of Computational Physics*, pages 209–224, 2019.
- [2] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2010.
- [3] Magnus Svärd and Jan Nordström. Review of summation-by-parts schemes for initial–boundary-value problems. *Journal of Computational Physics*, 268:17–38, 2014.