

## A Revisiting the approach by Sukumar, et al. (2021)

We illustrate their approach first for a Dirichlet boundary condition

$$\forall \mathbf{x} \in \partial\Omega : u(\mathbf{x}) = g(\mathbf{x}), \quad (30)$$

where  $\Omega$  is the domain on which the corresponding differential equation is posed. Let  $\phi$  be the approximate distance function to  $\partial\Omega$ . Then, the solution is given by

$$u(\mathbf{x}) = g(\mathbf{x}) + \phi(\mathbf{x})\Psi(\mathbf{x}). \quad (31)$$

The function  $\Psi$  is learned by the neural network and, because  $\phi(\mathbf{x}) = 0$  on  $\partial\Omega$ ,  $u(\mathbf{x})$  satisfies the boundary condition (30) by construction.

For a Robin boundary condition

$$\forall \mathbf{x} \in \partial\Omega : \frac{\partial u(\mathbf{x})}{\partial \mathbf{n}} + c(\mathbf{x})u(\mathbf{x}) = h(\mathbf{x}), \quad (32)$$

the solution becomes

$$u(\mathbf{x}) = \underbrace{\Psi_1(\mathbf{x}) - \phi(\mathbf{x})\nabla\phi(\mathbf{x}) \cdot \nabla\Psi_1(\mathbf{x})}_{\text{boundary value}} + \underbrace{\phi(\mathbf{x}) (c(\mathbf{x})\Psi_1(\mathbf{x}) - h(\mathbf{x}))}_{\text{boundary derivative}} + \underbrace{\phi(\mathbf{x})^2\Psi_2(\mathbf{x})}_{\text{remainder term}} \quad (33)$$

It now depends on two functions  $\Psi_1$  and  $\Psi_2$  that the network needs to learn. The first term determines the value on the boundary but has a normal derivative of zero whereas the second term is zero on the boundary but determines the normal derivative on the boundary. If  $\phi$  is normalized with respect to  $\partial\Gamma$ , the Robin condition (32) is satisfied. Because of  $\phi^2$ , the remainder term influences neither the value nor the derivative on at the boundary.

Now consider a case where both Dirichlet and Robin boundary conditions are prescribed on different boundary segments  $\Gamma_1, \dots, \Gamma_M$ . We assume that each  $\Gamma_i$  is  $C^1$ , but the overall boundary  $\partial\Omega = \bigcup_{i=1}^M \Gamma_i$  need only be in  $C^0$ .<sup>1</sup> The boundary conditions are

$$\forall i \in I_D : \forall \mathbf{x} \in \Gamma_i : u(\mathbf{x}) = g_i(\mathbf{x}), \quad (34a)$$

$$\forall i \in I_R : \forall \mathbf{x} \in \Gamma_i : \frac{\partial u(\mathbf{x})}{\partial \mathbf{n}} + c_i(\mathbf{x})u(\mathbf{x}) = h_i(\mathbf{x}). \quad (34b)$$

We consider Neumann boundary conditions as a special case of Robin conditions with  $c_i \equiv 0$ .

In this case, constructed solution is of the form given in (6) and (7), generalizing (31) and (33) from one to  $M$  boundary segments. In Sukumar, et al. (2021)'s approach, the local solution structures  $u_i$  are chosen as

$$u_i(\mathbf{x}) = \begin{cases} g_i(\mathbf{x}), & i \in I_D \\ \Psi_i(\mathbf{x}) - \phi_i(\mathbf{x})\nabla\phi_i(\mathbf{x}) \cdot \nabla\Psi_i(\mathbf{x}) + \phi_i(\mathbf{x}) (c_i(\mathbf{x})\Psi_i(\mathbf{x}) - h_i(\mathbf{x})), & i \in I_R \end{cases}. \quad (35)$$

**Proposition A.1.** We have  $u(\mathbf{x}) = u_i(\mathbf{x})$  for any  $1 \leq i \leq M$  and any  $\mathbf{x} \in \Gamma_i$ .

Therefore, for  $i \in I_D$  the Dirichlet boundary condition (34a) is satisfied by construction.

**Definition A.2** (Distance Function with vanishing gradient). Let  $\phi$  be a distance function to  $\Gamma$ . We say that  $\phi$  has a *vanishing gradient* if  $\nabla\phi(\mathbf{x}) = 0$  for  $\mathbf{x} \in \Gamma$ .

**Proposition A.3.** If  $\phi_i^{\mu_i}$  has a vanishing gradient in the sense of Definition A.2, it holds that

$$\frac{\partial u(\mathbf{x})}{\partial \mathbf{n}} = \frac{\partial u_i(\mathbf{x})}{\partial \mathbf{n}}. \quad (36)$$

**Remark A.4.** For  $\mu_i = 2$  for  $i \in I_R$  the corresponding functions  $\phi_i^{\mu_i}$  has a vanishing gradient.

**Proposition A.5.** For  $i \in I_R$ ,  $u$  satisfies the Robin condition because  $u_i$  satisfies it.

<sup>1</sup>Note that this could be relaxed to  $\partial\Omega \supset \bigcup_{i=1}^M \Gamma_i$  which would, for example, allow to treat initial-boundary value problems where the now boundary condition cannot be prescribed at final time. However, this case is not considered here.

## B Illustration of the problem arising for intersecting boundary segments

We illustrate the problem arising in the original approach by Sukumar, et al. (2021) with the example of two boundary segments  $\Gamma_1$  and  $\Gamma_2$  that share one common point  $P$ , i.e.  $\Gamma_1 \cap \Gamma_2 = \{P\}$ . In that case, the solution structure (6) would read

$$u(\mathbf{x}) = \frac{\phi_2^{\mu_2}(\mathbf{x})}{\phi_1^{\mu_1}(\mathbf{x}) + \phi_2^{\mu_2}(\mathbf{x})} u_1(\mathbf{x}) + \frac{\phi_1^{\mu_1}(\mathbf{x})}{\phi_1^{\mu_1}(\mathbf{x}) + \phi_2^{\mu_2}(\mathbf{x})} u_2(\mathbf{x}) + \Psi(\mathbf{x}) \phi_1(\mathbf{x})^{\mu_1} \phi_2(\mathbf{x})^{\mu_2}. \quad (37)$$

Because  $\phi_1(P) = \phi_2(P) = 0$ , the first two terms divide by zero in  $P$ . As pointed out by Rvachev, et al. (2001), we have

$$\lim_{\mathbf{x} \in \Gamma_1, \mathbf{x} \rightarrow P} u(\mathbf{x}) = \lim_{\mathbf{x} \in \Gamma_1, \mathbf{x} \rightarrow P} \frac{\phi_2(\mathbf{x})^{\mu_2}}{\phi_2(\mathbf{x})^{\mu_2}} u_1(\mathbf{x}) = u_1(P), \quad (38)$$

$$\lim_{\mathbf{x} \in \Gamma_2, \mathbf{x} \rightarrow P} u(\mathbf{x}) = \lim_{\mathbf{x} \in \Gamma_2, \mathbf{x} \rightarrow P} \frac{\phi_1(\mathbf{x})^{\mu_1}}{\phi_1(\mathbf{x})^{\mu_1}} u_2(\mathbf{x}) = u_2(P), \quad (39)$$

so that the solution structure is only continuous if  $u_1(P) = u_2(P)$ . This holds if  $u_1$  and  $u_2$  both correspond to Dirichlet boundary conditions and the boundary conditions are well-posed, that is, they match continuously and do not prescribe a discontinuous solution. However, if, say,  $u_1$  corresponds to a Robin condition, then  $u_1(P) = \Psi_1(P)$  would hold. Because  $\Psi_1$  is learned,  $\Psi_1(P)$  will not be equal to  $u_2(P)$  and the solution structure will be discontinuous. To solve this problem, we do not consider  $\Psi_1$  as a learned functions directly but instead express it as a function depending on two other learned functions in a way that  $\Psi_1(P) = u_2(P)$  is guaranteed to hold. If  $\Gamma_2$  is a Dirichlet segment, then  $\Psi_1(P) = g_2(P)$  has to be ensured. However, if  $\Gamma_2$  corresponds to a Robin condition, a new unknown function  $\Psi_P$  has to be introduced and it must be ensured that  $\Psi_1(P) = \Psi_2(P) = \Psi_P(P)$  holds.

## C Algorithms

For scalar problems, the full algorithm to create the solution structure is shown in Algorithm 1. The algorithm to construct the solution structures for the system case is shown in Algorithm 2 and Algorithm 3.

## D Normalizer functions

In section 2.1.2, we use a generalized Taylor polynomial as ansatz for the local solution structure. Here, we briefly describe the normalizer functions used there. Note that we use a notation for the normalizers that is different from the one originally used in Rvachev, et al. (1995)'s work.

Consider a function  $f : \bar{\Omega} \rightarrow \mathbb{R}$  and a boundary segment  $\Gamma_i$  with function  $\bar{\phi}_i$  that is normalized with respect to  $\Gamma_i$ . We call the concatenation  $f \circ \mathcal{N}(\cdot; \bar{\phi}_i)$  the *normalizer* of  $f$ , where  $\mathcal{N}$  is defined by  $\mathcal{N}(\mathbf{x}; \bar{\phi}) := \mathbf{x} - \bar{\phi}(\mathbf{x}) \nabla \bar{\phi}(\mathbf{x})$ . Near the boundary, the normalizer behaves like a constant in the normal direction of  $\Gamma_i$  so that

$$\forall \mathbf{x} \in \Gamma_i : \quad f(\mathcal{N}(\mathbf{x}; \bar{\phi}_i)) = f(\mathbf{x}), \quad \frac{\partial f(\mathcal{N}(\mathbf{x}; \bar{\phi}_i))}{\partial \nu} = 0 \quad (40)$$

holds if  $\bar{\phi}_i$  is normalized with respect to  $\Gamma_i$ . This property assures that the generalized Taylor polynomial in (11) satisfies the prescribed boundary conditions.

**Orthogonal projections** In our orthogonal projections approach, we assume that each  $\Gamma_i$  lies in a hyperplane and that  $\bar{\phi}_i$  is the exact signed distance function to this hyperplane. Since the exact signed distance function to  $\Gamma_i$  is normalized with respect to  $\Gamma_i$ , property (40) holds. Thus,  $\mathcal{N}(\cdot; \bar{\phi}_i)$  is simply the orthogonal projection, mapping its argument onto the corresponding hyperplane, as visualized in Figure 5. As described in Section 2.1.2, this comes with the advantage that the same ansatz can be used for multiple unknown functions, which reduces the number of functions the PINO has to learn.

---

**Algorithm 1** Algorithm to determine local solution structure  $u_i$  for for GLSS

---

```

for  $i = 1, \dots, M$  do
  if  $\Gamma_i$  is a Dirichlet segment then
    if  $\phi_i$  has a vanishing gradient then
       $\tilde{\Psi}_i$  is considered as an unknown function
       $u_i \leftarrow (x \mapsto g_i(x) + \bar{\phi}_i(x)\tilde{\Psi}_i(x))$ 
    else
       $u_i \leftarrow (x \mapsto g_i(x))$ 
    end if
  else
    if  $\Gamma_i$  has no intersection with any other segment then
       $\Psi_i$  is considered as an unknown function
    else if  $\Gamma_i$  has neighboring boundary segments  $\Gamma_a$  and  $\Gamma_b$  with corresponding intersecting
    points  $A$  and  $B$  as in Fig. 2 then
      for  $(p, P) \in \{(a, A), (b, B)\}$  do
        if  $\Gamma_p$  is Dirichlet segment then
           $u_P \leftarrow (x \mapsto g_p(x))$ 
        else if  $\Gamma_p$  is Robin segment then
           $\Psi_P$  is considered as an unknown function
           $u_P \leftarrow (x \mapsto \Psi_P(x))$ 
        end if
      end for
       $\bar{\Psi}_i$  is considered as an unknown function
       $\Psi_i \leftarrow (x \mapsto \frac{\phi_B(x)}{\phi_A(x) + \phi_B(x)} u_A(x) + \frac{\phi_A(x)}{\phi_A(x) + \phi_B(x)} u_B(x) + \phi_A(x)\phi_B(x)\bar{\Psi}_i(x))$ 
    end if
     $u_i \leftarrow (x \mapsto \Psi_i(x) - \bar{\phi}_i(x)\nabla\bar{\phi}_i(x) \cdot \nabla\Psi_i(x) + \bar{\phi}_i(x)(c_i(x)\Psi_i(x) - h_i(x)))$ 
  end if
end for

```

---

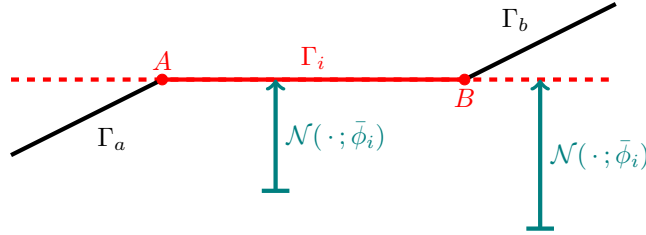


Figure 5: If  $\bar{\phi}_i$  is the exact signed distance function to the hyperplane that  $\Gamma_i$  lies on (indicated by the dotted line),  $\mathcal{N}(\cdot; \bar{\phi}_i)$  maps its argument onto the hyperplane, which is illustrated by the arrows.

## 758 E Proof of proposition 2.2

759 Let  $(i, j) \in I \times J$  and  $x \in \Gamma_i$ . Since  $\phi_i$  is a distance function with respect to  $\Gamma_i$ , we have  $\phi_i(x) = 0$ .  
 760 Therefore, the generic solution structure  $u$  given by (16) is equal to  $u_i$ . Using this and substituting  
 761  $u_i$  by (18) gets us

$$b_i^{(j)}(x) \cdot u(x) = b_i^{(j)}(x) \cdot u_i(x) = b_i^{(j)}(x) \cdot \sum_{k=1}^n b_i^{(k)}(x) u_i^{(k)}(x) = u_i^{(j)}(x), \quad (41)$$

762 where we abused the fact that  $b_i^{(1)}(x), \dots, b_i^{(n)}(x)$  is an orthonormal basis. This proves (19). To  
 763 prove (20), let  $(\tilde{i}, \tilde{j}) \in I \times J$  and let  $\phi_{\tilde{i}}^{\mu_{\tilde{i}}}$  have a vanishing gradient and  $x \in \Gamma_{\tilde{i}}$ . We substitute the

764 ansatz for  $u$  given by (16) into  $\frac{\partial(b_{\tilde{i}}^{(\tilde{j})}(x) \cdot u(x))}{\partial n}$  and calculate

---

**Algorithm 2** Algorithm to determine the local solution structures  $u_i^{(j)}$  for GLSS (Part 1)

---

```

1:  $IP \leftarrow \{\}$ 
2: for  $i = 1, \dots, M$  do
3:   if  $\Gamma_i$  has common points  $A$  and  $B$  with other segments and  $\exists j \in \{1, \dots, n\} : \mathbf{b}_i^{(j)}$  does not
   correspond to a Dirichlet condition then
4:      $IP \leftarrow IP \cup \{A, B\}$ 
5:   end if
6: end for ▷ Now,  $IP$  contains all intersection-points that need to be taken extra care of
7: for every  $P \in IP$  do
8:    $B \times G \leftarrow \{\} \times \{\}$ 
9:   for every  $\Gamma_i$  with  $P \in \Gamma_i$  do
10:    for every  $j \in \{1, \dots, n\}$  with  $\mathbf{b}_i^{(j)}$  belonging to a Dirichlet condition do
11:      if  $\dim(\text{span}(B \cup \{\mathbf{b}_i^{(j)}(P)\})) > \dim(\text{span}(B))$  then
12:         $B \times G \leftarrow B \times G \cup \{(\mathbf{b}_i^{(j)}(P), g_i^{(j)}(P))\}$ 
13:      end if
14:    end for
15:  end for
16:   $\mathbf{b}_P^{(1)}, \dots, \mathbf{b}_P^{(\dim(\text{span}(B)))} \leftarrow \text{basis of } \text{span}(B)$ 
17:   $A \leftarrow \text{zeros}(\dim(\text{span}(B)), \dim(\text{span}(B)))$ 
18:   $b \leftarrow \text{zeros}(\dim(\text{span}(B)), 1)$ 
19:  for  $k, (b, g) \in \text{enumerate}(B \times G)$  do
20:     $A[k, :] \leftarrow \mathbf{b}_P^T(\mathbf{b}_P^{(1)}, \dots, \mathbf{b}_P^{(\dim(\text{span}(B)))})$ 
21:     $b[k] \leftarrow g$ 
22:  end for
23:   $(g_P^{(1)}, \dots, g_P^{(\dim(\text{span}(B)))})^T \leftarrow \text{solve}(A, b)$ 
24:   $\mathbf{b}_P^{(\dim(\text{span}(B))+1)}, \dots, \mathbf{b}_P^{(n)} \leftarrow \text{basis of orthogonal complement of } \text{span}(B)$ 
25:   $\Psi_P^{(\dim(\text{span}(B))+1)}, \dots, \Psi_P^{(n)}$  are considered as unknown functions
26:   $\mathbf{u}_P \leftarrow \left( \mathbf{x} \mapsto \sum_{j=1}^{\dim(\text{span}(B))} \mathbf{b}_P^{(j)} g_P^{(j)} + \sum_{j=\dim(\text{span}(B))+1}^n \mathbf{b}_P^{(j)} \Psi_P^{(j)}(\mathbf{x}) \right)$ 
27: end for

```

---

$$\begin{aligned}
& \frac{\partial \left( \mathbf{b}_i^{(\tilde{j})}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) \right)}{\partial \mathbf{n}} \\
&= \frac{\partial \left( \mathbf{b}_i^{(\tilde{j})}(\mathbf{x}) \cdot \left( \sum_{i=1}^M w_i(\mathbf{x}) \mathbf{u}_i(\mathbf{x}) + [\Psi^{(1)}(\mathbf{x}), \dots, \Psi^{(n)}(\mathbf{x})]^T \prod_{i=1}^M \phi_i(\mathbf{x})^{\mu_i} \right) \right)}{\partial \mathbf{n}} \\
&= \frac{\partial \left( \sum_{i=1}^M w_i(\mathbf{x}) \mathbf{b}_i^{(\tilde{j})}(\mathbf{x}) \cdot \mathbf{u}_i(\mathbf{x}) \right)}{\partial \mathbf{n}} \\
&\quad + \frac{\partial \left( \mathbf{b}_i^{(\tilde{j})}(\mathbf{x}) \cdot [\Psi^{(1)}(\mathbf{x}), \dots, \Psi^{(n)}(\mathbf{x})]^T \prod_{i=1}^M \phi_i(\mathbf{x})^{\mu_i} \right)}{\partial \mathbf{n}}. \quad (42)
\end{aligned}$$

765 Using product rule, we find that the second term is zero because  $\phi_i^{\mu_i}(\mathbf{x}) = 0$  and  $\frac{\partial \phi_i^{\mu_i}(\mathbf{x})}{\partial \mathbf{n}} = 0$ . The  
766 remaining first term can be simplified as follows

---

**Algorithm 3** Algorithm to determine the local solution structures  $u_i^{(j)}$  for GLSS (part 2)

---

```

28: for  $i = 1, \dots, M$  do
29:   for  $j = 1, \dots, n$  do
30:     if  $b_i^{(j)}$  corresponds to a Dirichlet condition then
31:       if  $\phi_i^{\mu_i}$  has a non-zero gradient on  $\Gamma_i$  then
32:          $u_i^{(j)} \leftarrow (x \mapsto g_i^{(j)}(x))$ 
33:       else
34:          $\tilde{\Psi}_i^{(j)}$  is considered as unknown function
35:          $u_i^{(j)} \leftarrow (x \mapsto g_i^{(j)}(x) + \bar{\phi}_i(x) \tilde{\Psi}_i^{(j)}(x))$ 
36:       end if
37:     else
38:       if  $\Gamma_i$  has common points  $A$  and  $B$  with other segments then
39:          $\Psi_i^{(j)} \leftarrow x \mapsto b_i^{(j)}(x) \cdot \left( \frac{\phi_B(x)}{\phi_A(x) + \phi_B(x)} u_A(x) + \frac{\phi_A(x)}{\phi_A(x) + \phi_B(x)} u_B(x) \right)$ 
40:          $+ \phi_A(x) \phi_B(x) \tilde{\Psi}_i^{(j)}(x)$ 
41:       else
42:          $\Psi_i^{(j)}$  is considered as unknown function
43:       end if
44:       if  $b_i^{(j)}$  belongs to a Robin condition then
45:          $u_i^{(j)} \leftarrow x \mapsto \Psi_i^{(j)}(x) - \bar{\phi}_i(x) \nabla \bar{\phi}_i(x) \cdot \nabla \Psi_i^{(j)}(x)$ 
46:          $+ \bar{\phi}_i(x) \left( c_i^{(j)}(x) \cdot \sum_{k=1, (i,k) \notin IJ_D}^n b_i^{(k)}(x) \Psi_i^{(k)}(x) - h_i^{(j)}(x) \right)$ 
47:       else
48:          $u_i^{(j)} \leftarrow (x \mapsto \Psi_i^{(j)}(x))$ 
49:       end if
50:     end if
51:   end for
52: end for

```

---

$$\begin{aligned}
& \frac{\partial (b_i^{(\tilde{j})}(x) \cdot u(x))}{\partial \mathbf{n}} = \frac{\partial \left( \sum_{i=1}^M w_i(x) b_i^{(\tilde{j})}(x) \cdot u_i(x) \right)}{\partial \mathbf{n}} \\
& = \sum_{i=1}^M \left( \underbrace{\frac{\partial w_i(x)}{\partial \mathbf{n}}}_{=0} b_i^{(\tilde{j})}(x) \cdot u_i(x) + \underbrace{w_i(x)}_{= \begin{cases} 1, & \tilde{i} = i \\ 0, & \text{else} \end{cases}} \frac{\partial b_i^{(\tilde{j})}(x) \cdot u_i(x)}{\partial \mathbf{n}} \right) = \frac{\partial (b_i^{(\tilde{j})}(x) \cdot u_i(x))}{\partial \mathbf{n}}.
\end{aligned} \tag{43}$$

767 To show that  $\frac{\partial w_i(x)}{\partial \mathbf{n}} = 0$ , we first consider the case that  $i \neq \tilde{i}$  and calculate

$$\frac{\partial w_i(x)}{\partial \mathbf{n}} = \frac{\partial \frac{\prod_{j=1, j \neq i}^M \phi_j(x)^{\mu_j}}{\sum_{k=1}^M \prod_{j=1, j \neq k}^M \phi_j(x)^{\mu_j}}}{\partial \mathbf{n}} = \frac{\partial \phi_i^{\mu_i}(x)}{\partial \mathbf{n}} \frac{\prod_{j=1, j \neq i, j \neq \tilde{i}}^M \phi_j(x)^{\mu_j}}{\sum_{k=1}^M \prod_{j=1, j \neq k}^M \phi_j(x)^{\mu_j}} =: \frac{\partial \phi_i^{\mu_i}(x) f(x)}{\partial \mathbf{n}}, \tag{44}$$

768 where we introduce the function  $f$  for simplification. We use product rule and get

$$\frac{\partial \phi_i^{\mu_i}(x) f(x)}{\partial \mathbf{n}} = \underbrace{\frac{\partial \phi_i^{\mu_i}(x)}{\partial \mathbf{n}}}_{=0} f(x) + \underbrace{\phi_i^{\mu_i}(x)}_{=0} \frac{\partial f(x)}{\partial \mathbf{n}} = 0. \tag{45}$$

769 We now consider the case that  $i = \tilde{i}$  and calculate

$$\frac{\partial w_i(\mathbf{x})}{\partial \mathbf{n}} = \frac{\partial \frac{\prod_{j=1, j \neq i}^M \phi_j(\mathbf{x})^{\mu_j}}{\sum_{k=1}^M \prod_{j=1, j \neq k}^M \phi_j(\mathbf{x})^{\mu_j}}}{\partial \mathbf{n}} = \frac{\partial \frac{\prod_{j=1, j \neq i}^M \phi_j(\mathbf{x})^{\mu_j}}{\prod_{j=1, j \neq i}^M \phi_j(\mathbf{x})^{\mu_j} + \sum_{k=1, k \neq i}^M \prod_{j=1, j \neq k}^M \phi_j(\mathbf{x})^{\mu_j}}}{\partial \mathbf{n}} =: \frac{\partial \frac{u(\mathbf{x})}{u(\mathbf{x})+v(\mathbf{x})}}{\partial \mathbf{n}} \quad (46)$$

Next we use the quotient rule to find

$$\frac{\partial \frac{u(\mathbf{x})}{u(\mathbf{x})+v(\mathbf{x})}}{\partial \mathbf{n}} = \frac{\frac{\partial u(\mathbf{x})}{\partial \mathbf{n}}(u(\mathbf{x})+v(\mathbf{x})) - u(\mathbf{x}) \frac{\partial (u(\mathbf{x})+v(\mathbf{x}))}{\partial \mathbf{n}}}{(u(\mathbf{x})+v(\mathbf{x}))^2} = \frac{\frac{\partial u(\mathbf{x})}{\partial \mathbf{n}}v(\mathbf{x}) - u(\mathbf{x}) \frac{\partial v(\mathbf{x})}{\partial \mathbf{n}}}{(u(\mathbf{x})+v(\mathbf{x}))^2} = 0, \quad (47)$$

using  $v(\mathbf{x}) = \frac{\partial v(\mathbf{x})}{\partial \mathbf{n}} = 0$  in the last step. This is because we can factor out  $\phi_i(\mathbf{x})^{\mu_i}(\mathbf{x})$  as follows

$$v(\mathbf{x}) = \sum_{k=1, k \neq i}^M \prod_{j=1, j \neq k}^M \phi_j(\mathbf{x})^{\mu_j} = \phi_i(\mathbf{x})^{\mu_i} \sum_{k=1, k \neq i}^M \prod_{j=1, j \neq k, j \neq i}^M \phi_j(\mathbf{x})^{\mu_j}, \quad (48)$$

and that  $\phi_i(\mathbf{x})^{\mu_i} = \frac{\partial \phi_i(\mathbf{x})^{\mu_i}}{\partial \mathbf{n}} = 0$ . Now we can further simplify (43) by substituting the ansatz for  $\mathbf{u}_i$  given by (18), so that

$$\frac{\partial \left( \mathbf{b}_i^{(\tilde{j})}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) \right)}{\partial \mathbf{n}} = \frac{\partial \left( \mathbf{b}_i^{(\tilde{j})}(\mathbf{x}) \cdot \mathbf{u}_i(\mathbf{x}) \right)}{\partial \mathbf{n}} = \frac{\partial \left( \mathbf{b}_i^{(\tilde{j})}(\mathbf{x}) \cdot \sum_{j=1}^n \mathbf{b}_i^{(j)}(\mathbf{x}) u_i^{(j)}(\mathbf{x}) \right)}{\partial \mathbf{n}}. \quad (49)$$

For further simplification, we use that  $\mathbf{b}_i^{(\tilde{j})}(\mathbf{x}) \cdot \mathbf{b}_i^{(j)}(\mathbf{x}) = \delta_{\tilde{j}j}$  not only holds on  $\Gamma_i$  but in an open neighbourhood of  $\Gamma_i$ . Thus, we can say

$$\frac{\partial \left( \mathbf{b}_i^{(\tilde{j})}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) \right)}{\partial \mathbf{n}} = \frac{\partial \left( \sum_{j=1}^n \mathbf{b}_i^{(\tilde{j})}(\mathbf{x}) \cdot \mathbf{b}_i^{(j)}(\mathbf{x}) u_i^{(j)}(\mathbf{x}) \right)}{\partial \mathbf{n}} = \frac{\partial u_i^{(\tilde{j})}(\mathbf{x})}{\partial \mathbf{n}}, \quad (50)$$

which is what we wanted to show.

## F Proof that the solution structure satisfies the boundary conditions

We want to show that the solution structure (16) satisfies the boundary conditions with local solution structures being chosen either according to the GLSS or the OP approach. We begin by proving that the solution structure satisfies the Dirichlet boundary conditions. Let  $(i, j) \in IJ_D$  and  $\mathbf{x} \in \Gamma_i$ . Using Proposition 2.2 and substituting (21) for the local solution structure results in

$$\mathbf{b}_i^{(j)}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) = u_i^{(j)}(\mathbf{x}) = \begin{cases} g_i^{(j)}(\mathbf{x}) + \bar{\phi}_i(\mathbf{x}) \tilde{\Psi}_i^{(j)}(\mathbf{x}), & \phi_i^{\mu_i} \text{ has a vanishing gradient,} \\ g_i^{(j)}(\mathbf{x}), & \text{else.} \end{cases} \quad (51)$$

Because of  $\bar{\phi}_i(\mathbf{x}) = 0$  for  $\mathbf{x} \in \Gamma_i$ , it follows that

$$\mathbf{b}_i^{(j)}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) = g_i^{(j)}(\mathbf{x}), \quad (52)$$

which is what we wanted to show.

To show that the Robin conditions are satisfied, let  $(i, j) \in IJ_R$  and  $\mathbf{x} \in \Gamma_i$ . By Proposition 2.2 we have

$$\frac{\partial \left( \mathbf{b}_i^{(j)}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) \right)}{\partial \mathbf{n}} = \frac{\partial u_i^{(j)}(\mathbf{x})}{\partial \mathbf{n}}. \quad (53)$$

First we consider the GLSS approach. Using (22) for the local solution structure according yields

$$\frac{\partial \left( \mathbf{b}_i^{(j)}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) \right)}{\partial \mathbf{n}} = \frac{\partial \left( \Psi_i^{(j)}(\mathbf{x}) - \bar{\phi}_i(\mathbf{x}) \nabla \bar{\phi}_i(\mathbf{x}) \cdot \nabla \Psi_i^{(j)}(\mathbf{x}) \right)}{\partial \mathbf{n}} + \frac{\partial \left( \bar{\phi}_i(\mathbf{x}) f_i^{(j)}(\mathbf{x}) \right)}{\partial \mathbf{n}}. \quad (54)$$

The first term is zero because

$$\begin{aligned} \frac{\partial \left( \Psi_i^{(j)}(\mathbf{x}) - \bar{\phi}_i(\mathbf{x}) \nabla \bar{\phi}_i(\mathbf{x}) \cdot \nabla \Psi_i^{(j)}(\mathbf{x}) \right)}{\partial \mathbf{n}} &= \frac{\partial \Psi_i^{(j)}(\mathbf{x})}{\partial \mathbf{n}} - \underbrace{\frac{\partial \bar{\phi}_i(\mathbf{x})}{\partial \mathbf{n}}}_{=-1} \underbrace{\nabla \bar{\phi}_i(\mathbf{x}) \cdot \nabla \Psi_i^{(j)}(\mathbf{x})}_{=-\frac{\partial \Psi_i^{(j)}(\mathbf{x})}{\partial \mathbf{n}}} \\ &\quad - \underbrace{\bar{\phi}_i(\mathbf{x})}_{=0} \frac{\partial \nabla \bar{\phi}_i(\mathbf{x}) \cdot \nabla \Psi_i^{(j)}(\mathbf{x})}{\partial \mathbf{n}} = 0. \end{aligned} \quad (55)$$

Thus, we get

$$\frac{\partial \left( \mathbf{b}_i^{(j)}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) \right)}{\partial \mathbf{n}} = \frac{\partial \left( \bar{\phi}_i(\mathbf{x}) f_i^{(j)}(\mathbf{x}) \right)}{\partial \mathbf{n}} = \underbrace{\frac{\partial \bar{\phi}_i(\mathbf{x})}{\partial \mathbf{n}}}_{=-1} f_i^{(j)}(\mathbf{x}) + \underbrace{\bar{\phi}_i(\mathbf{x})}_{=0} \frac{\partial f_i^{(j)}(\mathbf{x})}{\partial \mathbf{n}} = -f_i^{(j)}(\mathbf{x}). \quad (56)$$

Substituting (23) for  $f_i^{(j)}$  yields

$$\begin{aligned} \frac{\partial \left( \mathbf{b}_i^{(j)}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) \right)}{\partial \mathbf{n}} &= -\mathbf{c}_i^{(j)}(\mathbf{x}) \cdot \sum_{k=1, (i,k) \notin IJ_D}^n \mathbf{b}_i^{(k)}(\mathbf{x}) \Psi_i^{(k)}(\mathbf{x}) + h_i^{(j)}(\mathbf{x}) \\ &= -\mathbf{c}_i^{(j)}(\mathbf{x}) \cdot \sum_{k=1, (i,k) \notin IJ_D}^n \mathbf{b}_i^{(k)}(\mathbf{x}) \Psi_i^{(k)}(\mathbf{x}) - \underbrace{\mathbf{c}_i^{(j)}(\mathbf{x}) \cdot \sum_{k=1, (i,k) \in IJ_D}^n \mathbf{b}_i^{(k)}(\mathbf{x}) u_i^{(k)}(\mathbf{x})}_{=0 \text{ because } \mathbf{c}_i^{(j)}(\mathbf{x}) \in \text{span}(\{\mathbf{b}_i^{(k)}(\mathbf{x}) | (i,k) \notin IJ_D\})} + h_i^{(j)}(\mathbf{x}) \\ &= -\mathbf{c}_i^{(j)}(\mathbf{x}) \cdot \underbrace{\sum_{k=1}^n \mathbf{b}_i^{(k)}(\mathbf{x}) u_i^{(k)}(\mathbf{x})}_{\mathbf{u}_i(\mathbf{x}) = \mathbf{u}(\mathbf{x}) \text{ on } \Gamma_i} + h_i^{(j)}(\mathbf{x}) = -\mathbf{c}_i^{(j)}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) + h_i^{(j)}(\mathbf{x}), \end{aligned} \quad (57)$$

which is what we wanted to show. Now, we want to show the same for the case that the local solution structure has been chosen according to OP approach. We use  $u_i^{(j)}$  from (53) in (27) to get

$$\frac{\partial \left( \mathbf{b}_i^{(j)}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) \right)}{\partial \mathbf{n}} = \frac{\partial \bar{\Psi}^{(j)}(\mathcal{N}(\mathbf{x}; \bar{\phi}_i))}{\partial \mathbf{n}} + \frac{\partial \bar{\phi}_i(\mathbf{x}) f_i^{(j)}(\mathcal{N}(\mathbf{x}; \bar{\phi}_i))}{\partial \mathbf{n}}. \quad (58)$$

That the first term is zero can be seen by applying the chain rule and using that

$$\frac{\partial \mathcal{N}(\mathbf{x}; \bar{\phi}_i)}{\partial \mathbf{n}} = \frac{\partial \mathbf{x}}{\partial \mathbf{n}} - \frac{\partial \bar{\phi}_i(\mathbf{x}) \nabla \bar{\phi}_i(\mathbf{x})}{\partial \mathbf{n}} = \mathbf{n} - \underbrace{\frac{\partial \bar{\phi}_i(\mathbf{x})}{\partial \mathbf{n}}}_{=-1} \underbrace{\nabla \bar{\phi}_i(\mathbf{x})}_{=-\mathbf{n}} - \underbrace{\bar{\phi}_i(\mathbf{x})}_{=0} \frac{\partial \nabla \bar{\phi}_i(\mathbf{x})}{\partial \mathbf{n}} = \mathbf{0}. \quad (59)$$

Similar as for GLSS, we conclude that

$$\frac{\partial \left( \mathbf{b}_i^{(j)}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) \right)}{\partial \mathbf{n}} = -f_i^{(j)}(\underbrace{\mathcal{N}(\mathbf{x}; \bar{\phi}_i)}_{=\mathbf{x}}), \quad (60)$$

In the same way wie did in (57), we can conclude that the solution structure satisfies the boundary condition. Note that because  $\mathcal{N}(\mathbf{x}; \bar{\phi}_i) = \mathbf{x}$  holds for  $\mathbf{x} \in \Gamma_i$ , the orthogonal projection  $\mathcal{N}(\mathbf{x}; \bar{\phi}_i)$  does not need to be applied to the function  $f_i^{(j)}$  but only to the function  $\bar{\Psi}^{(j)}$ .

## G Architecture and training of the network

All numerical experiments were conducted on a desktop computer with an Intel® Core™ i7-4790K CPU, 16 GB of RAM, and an NVIDIA GeForce GTX 960 GPU (4 GB).

We use the neural operator architecture from Li, et al. (2021), which has the form

$$\mathcal{G}_\theta = \mathcal{Q} \circ \sigma(\mathcal{W}_L + \mathcal{K}_L) \circ \dots \circ \sigma(\mathcal{W}_1 + \mathcal{K}_1) \circ \mathcal{L}, \quad (61)$$

Table 4: Parameters used to train the FNO.

Darcy flow			
	starting lr	epochs per milestone	total epochs
PINN-like training	0.0025	200	1000 (each parameter)
Operator training	0.005	100	1000
Finetuning	0.0025	200	1000 (each parameter)
Navier-Stokes equations			
	starting lr	epochs per milestone	total epochs
PINN-like training	0.005	500	4000

where  $\mathcal{L}$  and  $\mathcal{Q}$  are pointwise operators,  $\mathcal{L}$  lifts the input function to a higher dimensional space and  $\mathcal{Q}$  projects its input function back to lower dimensional space. In between, each layer  $\sigma(\mathcal{W}_l + \mathcal{K}_l)$  consists of a pointwise linear operator  $\mathcal{W}_l$  and a kernel integral operator  $\mathcal{K}_l$ . We use Fourier convolution operators for  $\mathcal{K}_l$ . Details are given by Li, et al. (2020b) and Li, et al. (2021). We use the FNO as implemented in FNN2d.<sup>2</sup> For all experiments, we use the architecture shown in (61) with four Fourier convolution layers and GeLU activation functions. In every Fourier layer we have 20 modes for each dimension. Each of the four layers  $\sigma(\mathcal{W}_l + \mathcal{K}_l)$  works in 64-dimensional space, which requires that the layer  $\mathcal{L}$  lifts the input to 64-dimensional space. Further,  $\text{fdim} = 128$  is the dimension that  $\mathcal{Q}$  lifts its inputs up to before projecting them back to the output dimension.

The computational domain for the FNO is required to be rectangular. In experiments where this is not the case, we use a bounding box of the domain as computational domain for the FNO. The rectangular domains are discretized on uniform grids. For the Darcy flow problem the resolution of the uniform grid is  $101 \times 101$  and  $441 \times 83$  for the Navier-Stokes problem. Derivatives in solution structures or losses are computed with finite differences on the uniform grids.

All four convolution layers together have 13, 123, 840 trainable parameters in all experiments. The lifting layer  $\mathcal{L}$  has 256 trainable parameters. The only change in size is in the output dimension  $I$  of  $\mathcal{Q}$  and thus the dimensionality of the mapping  $\mathcal{G}_\theta(\mathbf{a}) : \Omega \rightarrow \mathbb{R}^I$ . Dimension  $I$  depends on the used approach for the boundary conditions. For weakly enforced boundary conditions, where  $\mathcal{G}_\theta(\mathbf{a}) = \mathbf{u}$  is the solution,  $I$  is equal to the dimension of the solution, that is  $I = 1$  for the scalar Darcy flow and  $I = 3$  for the Navier-Stokes equations where  $\mathbf{u}$  contains two velocity components and the pressure. For GLSS, OP and semi-weak boundary conditions, which rely on a solution structure,  $I$  is equal to the number of functions  $\Psi_i$  that need to be trained. In our experiments, the  $\mathcal{Q}$  has  $8320 + 129I$  trainable parameters, see Table 1 for the overall number for the different boundary conditions.

Training uses the Adam optimizer Kingma, et al. (2017) provided as part of Li, et al. (2021)’s FNO implementation.<sup>3</sup> All models are trained for a fixed number of total epochs. In the training process, the learning rate is reduced by half at every milestone, see Table 4.

## H Darcy flow problem

The governing PDE reads

$$-\nabla \cdot (a(x, y) \nabla u(x, y)) = f(x, y) \quad (62)$$

Here,  $a(x, y)$  is the inhomogeneous and anisotropic diffusivity and  $f$  an inhomogeneous source. For a pair of parameters  $(\alpha, \beta) \in [1, 10]^2$  we set the diffusivity in (62) to

$$a(x, y) := \sin(\alpha x) \sin(\beta y) \quad (63)$$

and use the method of manufactured solutions to construct  $f$  such that

$$u(x, y) = \sin(\alpha x) \cos(\beta y) \quad (64)$$

<sup>2</sup>[https://github.com/neuraloperator/physics\\_informed/blob/Grad-ckpt/models/fourier2d.py](https://github.com/neuraloperator/physics_informed/blob/Grad-ckpt/models/fourier2d.py)

<sup>3</sup>[https://github.com/neuraloperator/physics\\_informed/blob/Grad-ckpt/train\\_utils/adam.py](https://github.com/neuraloperator/physics_informed/blob/Grad-ckpt/train_utils/adam.py)

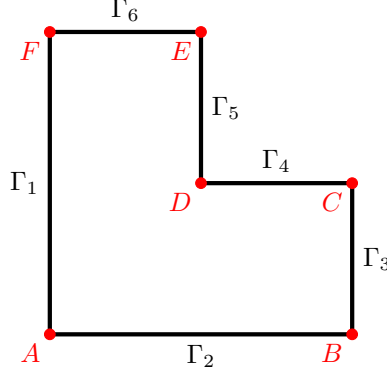


Figure 6: Computational domain for Darcy flow. Dirichlet data is prescribed on  $\Gamma_1, \Gamma_2$ , Neumann boundary conditions are set for  $\Gamma_3, \Gamma_4$ , and Robin conditions on  $\Gamma_5, \Gamma_6$ .

832 becomes the sought solution. By inserting  $a(x, y)$  from (63) into (62) we find

$$f(x, y) = -\frac{1}{2} \sin(2\beta y)(\alpha^2 \cos(2\alpha x) + \beta^2 \cos(2\alpha x) - \beta^2). \quad (65)$$

833 Training data is generated by generating 500 pairs  $(\alpha_i, \beta_i)$  from  $[1, 10]^2$  by uniform sampling. The  
 834 operator  $\mathcal{G}$  is trained on 400 of those pairs. For testing, it is evaluated on the remaining 100 pairs.  
 835 The produced solution is compared against the analytic solution (64) and the average  $l_2$ -error over  
 836 those 100 pairs is computed. Finetuning trains for the specific solution for each of the 100 remaining  
 837 parameter pairs independently, computes the  $l_2$  error against (64) and averages over all 100  $l_2$ -errors.  
 838 PINN-style training also trains the solutions for each of the 100 parameter pairs independently but  
 839 does not start from the operator that was trained on the first 400 pairs but from the untrained network  
 840 instead. The first 400 pairs remain unused in this case.

841 We consider the L-shaped domain shown in Figure 6 and derive the following boundary conditions  
 842 from the analytic solution (64)

$$\forall(x, y) \in \Gamma_1 : \quad u(x, y) = g_1(x, y) = 0 \quad (66)$$

$$\forall(x, y) \in \Gamma_2 : \quad u(x, y) = g_2(x, y) = \sin(\alpha x) \quad (67)$$

$$\forall(x, y) \in \Gamma_3 : \quad \frac{\partial u(x, y)}{\partial \mathbf{n}} = h_3(x, y) = \alpha \cos(\alpha) \cos(\beta y) \quad (68)$$

$$\forall(x, y) \in \Gamma_4 : \quad \frac{\partial u(x, y)}{\partial \mathbf{n}} = h_4(x, y) = -\beta \sin(\alpha x) \sin(0.5\beta) \quad (69)$$

$$\forall(x, y) \in \Gamma_5 : \quad \frac{\partial u(x, y)}{\partial \mathbf{n}} + u(x, y) = h_5(x, y) = (\alpha \cos(0.5\alpha) + \sin(0.5\alpha)) \cos(\beta y) \quad (70)$$

$$\forall(x, y) \in \Gamma_6 : \quad \frac{\partial u(x, y)}{\partial \mathbf{n}} + u(x, y) = h_6(x, y) = (\cos(\beta) - \beta \sin(\beta)) \sin(\alpha x). \quad (71)$$

843 The resulting solution structure is

$$u(x, y) = \sum_{i=1}^6 w_i(x, y) u_i(x, y) + \Psi(x, y) \phi_1(x, y) \phi_2(x, y) \phi_3(x, y)^2 \phi_4(x, y)^2 \phi_5(x, y)^2 \phi_6(x, y)^2. \quad (72)$$

844 The local solution structures  $u_i$  are chosen depending on whether GLSS or OP is used.

845 **GLSS.** Here, the local solution structures are

$$\begin{aligned} u_1(x, y) &= g_1(x, y), \\ u_2(x, y) &= g_2(x, y), \end{aligned}$$

846 for Dirichlet boundary conditions according to (8),

$$\begin{aligned} u_3(x, y) &= \Psi_3(x, y) - \bar{\phi}_3(x, y)(\nabla \bar{\phi}_3(x, y) \cdot \nabla \Psi_3(x, y) + h_3(x, y)), \\ u_4(x, y) &= \Psi_4(x, y) - \bar{\phi}_4(x, y)(\nabla \bar{\phi}_4(x, y) \cdot \nabla \Psi_4(x, y) + h_4(x, y)), \end{aligned}$$

847 for Neumann boundary conditions and

$$\begin{aligned} u_5(x, y) &= \Psi_5(x, y)(1 + \bar{\phi}_5(x, y)) - \bar{\phi}_5(x, y)(\nabla \bar{\phi}_5(x, y) \cdot \nabla \Psi_5(x, y) + h_5(x, y)), \\ u_6(x, y) &= \Psi_6(x, y)(1 + \bar{\phi}_6(x, y)) - \bar{\phi}_6(x, y)(\nabla \bar{\phi}_6(x, y) \cdot \nabla \Psi_6(x, y) + h_6(x, y)) \end{aligned}$$

848 for Robin boundary conditions according to (9). The functions  $\Psi_3, \Psi_4, \Psi_5$  and  $\Psi_6$  are not unknowns  
849 because their corresponding boundary segments have intersection points with other segments.  
850 That is why, we choose them as given in (10) and introduce new unknowns  $\Psi_C, \Psi_D$  and  $\Psi_E$  that  
851 correspond to the intersection points. With that, we set

$$\Psi_3(x, y) = \frac{\phi_C(x, y)}{\phi_B(x, y) + \phi_C(x, y)} g_2(x, y) + \frac{\phi_B(x, y)}{\phi_B(x, y) + \phi_C(x, y)} \Psi_C(x, y), \quad (73)$$

$$\Psi_4(x, y) = \frac{\phi_D(x, y)}{\phi_C(x, y) + \phi_D(x, y)} \Psi_C(x, y) + \frac{\phi_C(x, y)}{\phi_C(x, y) + \phi_D(x, y)} \Psi_D(x, y), \quad (74)$$

$$\Psi_5(x, y) = \frac{\phi_E(x, y)}{\phi_D(x, y) + \phi_E(x, y)} \Psi_D(x, y) + \frac{\phi_D(x, y)}{\phi_D(x, y) + \phi_E(x, y)} \Psi_E(x, y), \quad (75)$$

$$\Psi_6(x, y) = \frac{\phi_F(x, y)}{\phi_E(x, y) + \phi_F(x, y)} \Psi_E(x, y) + \frac{\phi_E(x, y)}{\phi_E(x, y) + \phi_F(x, y)} g_1(x, y). \quad (76)$$

852 The unknown functions  $\Psi, \Psi_C, \Psi_D$  and  $\Psi_E$  are approximated by the PINO.

853 **OP.** Here, the local solution structures are

$$\begin{aligned} u_1(x, y) &= g_1(x, y), \\ u_2(x, y) &= g_2(x, y), \end{aligned}$$

854 for Dirichlet boundary conditions,

$$\begin{aligned} u_3(x, y) &= \bar{\Psi}(\mathcal{N}(x, y; \bar{\phi}_3)) - \bar{\phi}_3(x, y) h_3(x, y), \\ u_4(x, y) &= \bar{\Psi}(\mathcal{N}(x, y; \bar{\phi}_4)) - \bar{\phi}_4(x, y) h_4(x, y), \end{aligned}$$

855 for Neumann boundary conditions and

$$\begin{aligned} u_5(x, y) &= \bar{\Psi}(\mathcal{N}(x, y; \bar{\phi}_5))(1 + \bar{\phi}_5(x, y)) - \bar{\phi}_5(x, y) h_5(x, y), \\ u_6(x, y) &= \bar{\Psi}(\mathcal{N}(x, y; \bar{\phi}_6))(1 + \bar{\phi}_6(x, y)) - \bar{\phi}_6(x, y) h_6(x, y) \end{aligned}$$

856 for Robin boundary conditions according to (11). We ensure that  $\bar{\Psi}$  satisfies all Dirichlet conditions  
857 by following (13) and setting

$$\bar{\Psi}(x, y) = \frac{\phi_2(x, y)}{\phi_1(x, y) + \phi_2(x, y)} g_1(x, y) + \frac{\phi_1(x, y)}{\phi_1(x, y) + \phi_2(x, y)} g_2(x, y) + \phi_1(x, y) \phi_2(x, y) \tilde{\Psi}(x, y), \quad (77)$$

858 where  $\Psi$  and  $\tilde{\Psi}$  need to be trained.

## 859 I Best- and worst-case solutions for the Darcy flow problem

860 In this section, we show the solutions produced by the different boundary conditions for the parameter  
861 pairs that produced the best- and worst-case errors in Table 2. In addition, the middle figures show  
862 the solution that provides the median error for the 100 evaluation pairs.

## 863 J Navier-Stokes equations

864 We consider the standard test case of a flow around a cylinder by Turek, et al. (1996). The domain is  
865 sketched in Figure 11. The benchmark solves the stationary, incompressible Navier-Stokes equations

$$-\nu \nabla^2 \mathbf{u}(x, y) + (\mathbf{u} \cdot \nabla) \mathbf{u}(x, y) + \nabla p(x, y) = 0, \quad (x, y) \in \Omega, \quad (78)$$

$$\nabla \cdot \mathbf{u}(x, y) = 0, \quad (x, y) \in \Omega \quad (79)$$

866 with inflow boundary condition  $\mathbf{u}(x, y) = \mathbf{u}_0(x, y)$  on  $\Gamma_{\text{in}} = \Gamma_1$ , no-slip boundary condition  
867  $\mathbf{u}(x, y) = 0$  on  $\Gamma_{\text{wall}} = \Gamma_2 \cup \Gamma_4 \cup \Gamma_5$  and outflow boundary condition  $\nu \frac{\partial \mathbf{u}(x, y)}{\partial \mathbf{n}} - \mathbf{n}(x, y) p(x, y) = 0$

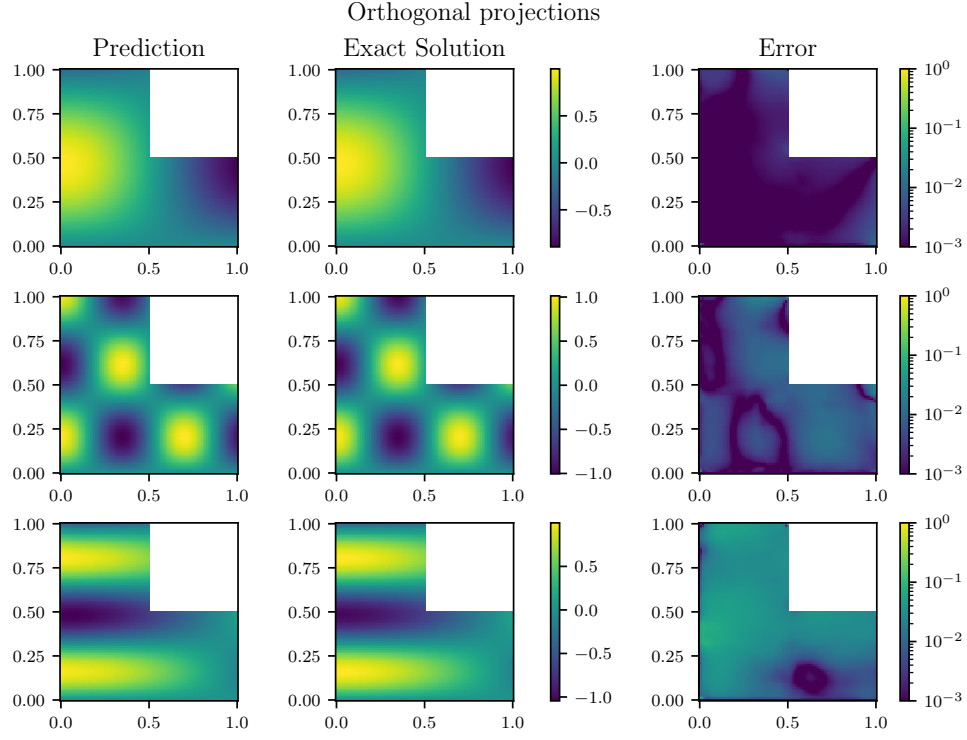


Figure 7: Best-case (upper), median (middle) and worst-case (lower) solution for the Darcy flow with OP boundary conditions.

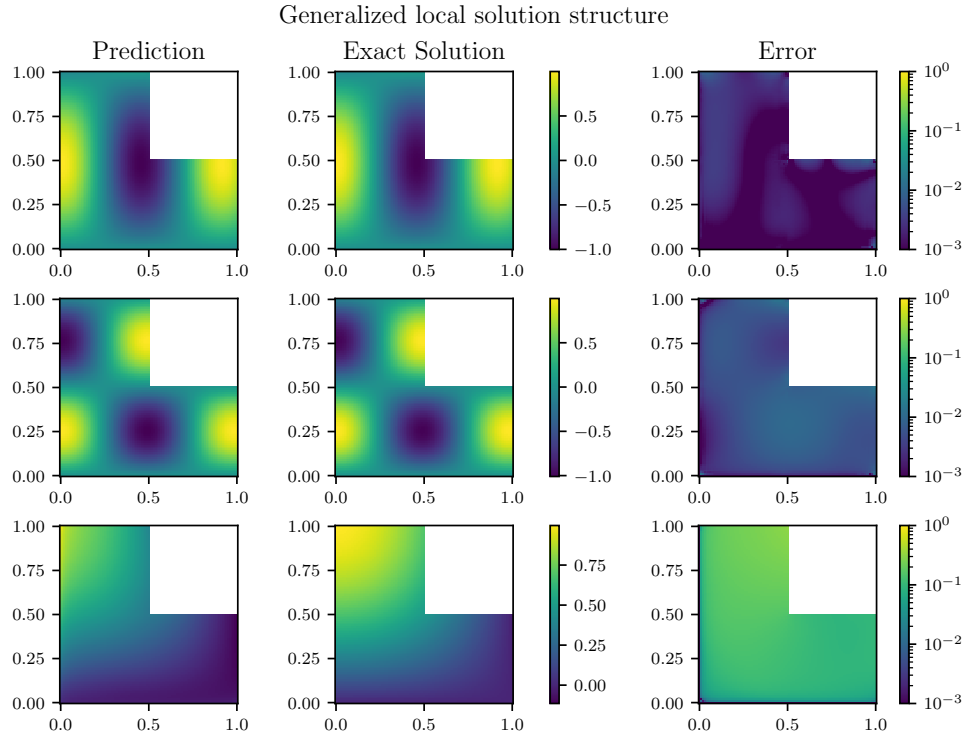


Figure 8: Best-case (upper), median (middle) and worst-case (lower) solution for the Darcy flow with GLSS boundary conditions.

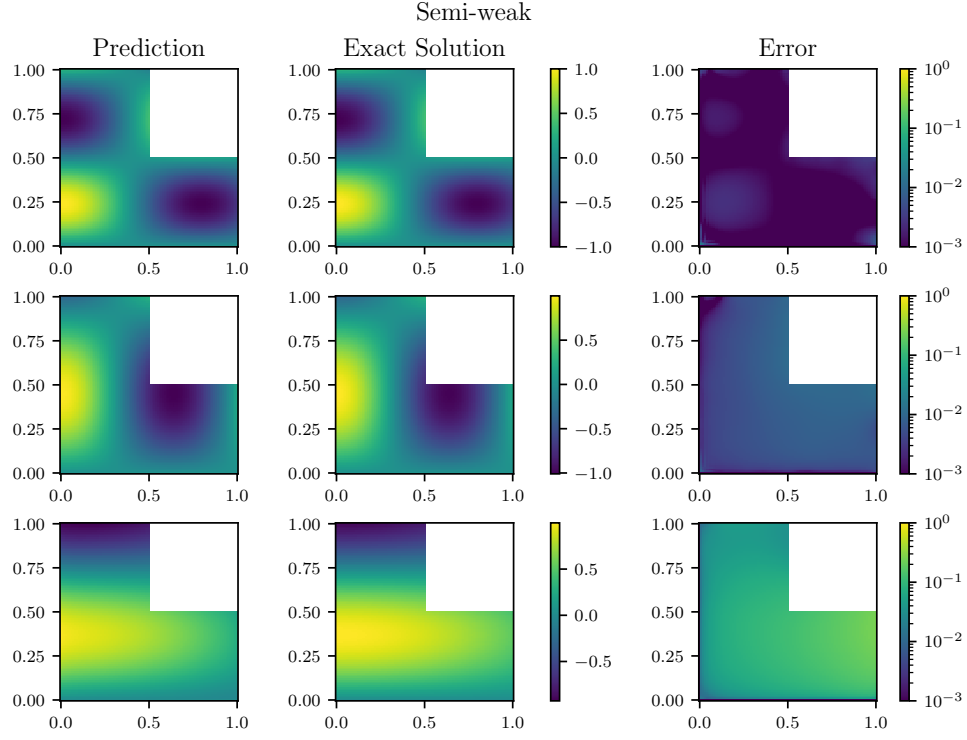


Figure 9: Best-case (upper), median (middle) and worst-case (lower) solution for the Darcy flow with semi-weak boundary conditions.

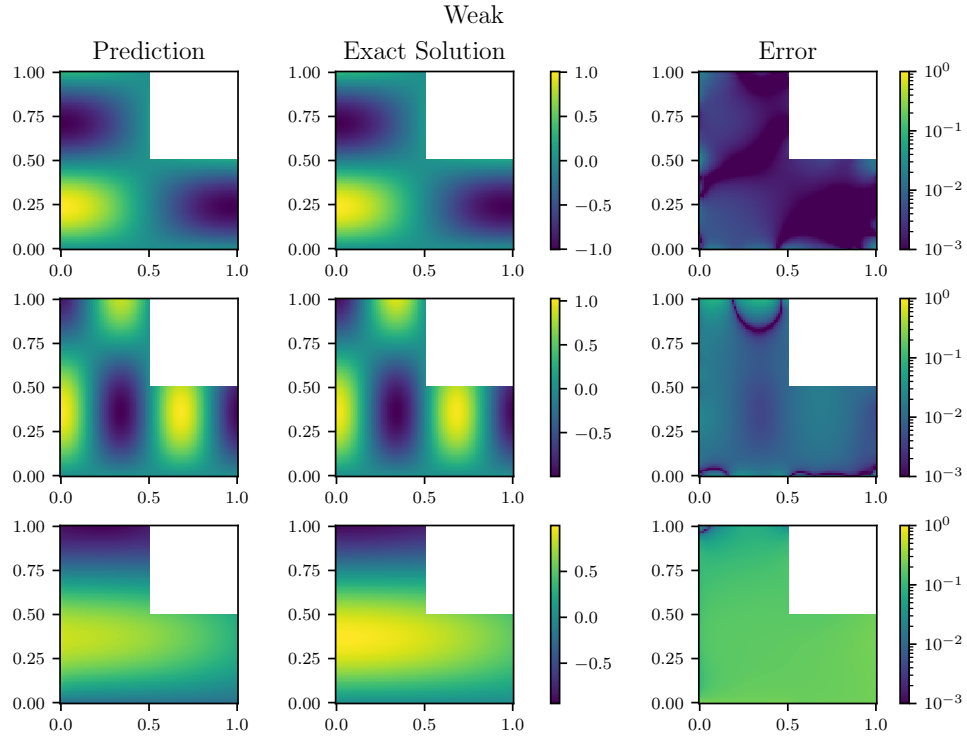


Figure 10: Best-case (upper), median (middle) and worst-case (lower) solution for the Darcy flow with weak boundary conditions.

868 on  $\Gamma_{\text{out}} = \Gamma_3$ . We set  $\mathbf{u}_0(x, y) = [u_0(y), 0]^T$  and  $\nu = 0.001$ . Only PINN-like training is investigated  
869 in this case. Note that we rescale the pressure by  $\tilde{p} := \frac{p}{\sqrt{\nu}}$  to reduce the ratio between velocity  
870 derivative and pressure in the outflow condition, which provided benefits to the accuracy of the exact  
871 boundary conditions. Since no analytical solution is available, we compute a reference solution  
872 numerically using FEniCSx Baratta, et al. (2023); Scroggs, et al. (2022a,b); Alnaes, et al. (2014).<sup>4</sup>  
873 We consider the boundary conditions in the form of (14) and (15). For all  $\Gamma_i$ , we pose the boundary  
874 conditions with respect to the same basis vectors  $b_i^{(1)}(\mathbf{x}) = (1, 0, 0)^T$ ,  $b_i^{(2)}(\mathbf{x}) = (0, 1, 0)^T$  and  
 $b_i^{(3)}(\mathbf{x}) = (0, 0, 1)^T$ .

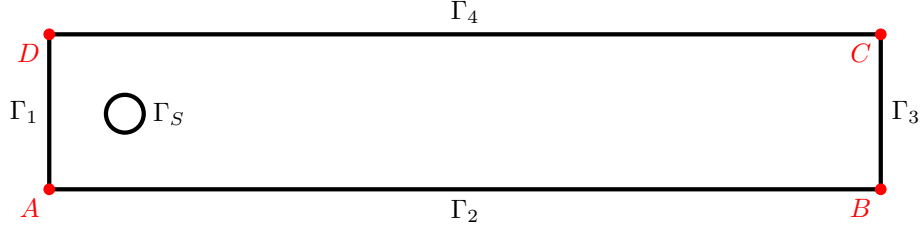


Figure 11: Computational domain for the Navier-Stokes test case.

875

876 **GLSS.** The solution structure generated by Algorithms 2 and 3 are

$$\begin{aligned}
 \begin{pmatrix} u(x, y) \\ v(x, y) \\ \tilde{p}(x, y) \end{pmatrix} &= w_1(x, y) \begin{pmatrix} u_0(y) \\ 0 \\ \Psi_1^{\tilde{p}}(x, y) \end{pmatrix} + w_2(x, y) \begin{pmatrix} 0 \\ 0 \\ \Psi_2^{\tilde{p}}(x, y) \end{pmatrix} \\
 &+ w_3(x, y) \begin{pmatrix} \Psi_3^u(x, y) - \bar{\phi}_3(x, y) \nabla \bar{\phi}_3(x, y) \cdot \nabla \Psi_3^u(x, y) + \bar{\phi}_3(x, y) \Psi_3^{\tilde{p}}(x, y) / \sqrt{\nu} \\ \Psi_3^v(x, y) - \bar{\phi}_3(x, y) \nabla \bar{\phi}_3(x, y) \cdot \nabla \Psi_3^v(x, y) \\ \Psi_3^{\tilde{p}}(x, y) \end{pmatrix} \\
 &+ w_4(x, y) \begin{pmatrix} 0 \\ 0 \\ \Psi_4^{\tilde{p}}(x, y) \end{pmatrix} + w_S(x, y) \begin{pmatrix} 0 \\ 0 \\ \Psi_S^{\tilde{p}}(x, y) \end{pmatrix} \\
 &+ \phi_1(x, y) \phi_2(x, y) \phi_3(x, y)^2 \phi_4(x, y) \phi_S(x, y) \begin{pmatrix} \Psi^u(x, y) \\ \Psi^v(x, y) \\ \Psi^{\tilde{p}}(x, y) \end{pmatrix}, \quad (80)
 \end{aligned}$$

877 where

$$\Psi_3^u(x, y) = \phi_B(x, y) \phi_C(x, y) \bar{\Psi}_3^u(x, y), \quad (81)$$

$$\Psi_3^v(x, y) = \phi_B(x, y) \phi_C(x, y) \bar{\Psi}_3^v(x, y), \quad (82)$$

$$\Psi_1^{\tilde{p}}(x, y) = \frac{\phi_D(x, y)}{\phi_A(x, y) + \phi_D(x, y)} \Psi_A^{\tilde{p}}(x, y) + \frac{\phi_A(x, y)}{\phi_A(x, y) + \phi_D(x, y)} \Psi_D^{\tilde{p}}(x, y), \quad (83)$$

$$\Psi_2^{\tilde{p}}(x, y) = \frac{\phi_B(x, y)}{\phi_A(x, y) + \phi_B(x, y)} \Psi_A^{\tilde{p}}(x, y) + \frac{\phi_A(x, y)}{\phi_A(x, y) + \phi_B(x, y)} \Psi_B^{\tilde{p}}(x, y), \quad (84)$$

$$\Psi_3^{\tilde{p}}(x, y) = \frac{\phi_C(x, y)}{\phi_B(x, y) + \phi_C(x, y)} \Psi_B^{\tilde{p}}(x, y) + \frac{\phi_B(x, y)}{\phi_B(x, y) + \phi_C(x, y)} \Psi_C^{\tilde{p}}(x, y), \quad (85)$$

$$\Psi_4^{\tilde{p}}(x, y) = \frac{\phi_D(x, y)}{\phi_C(x, y) + \phi_D(x, y)} \Psi_C^{\tilde{p}}(x, y) + \frac{\phi_C(x, y)}{\phi_C(x, y) + \phi_D(x, y)} \Psi_D^{\tilde{p}}(x, y) \quad (86)$$

878 The unknown functions to be learned are  $\Psi^u$ ,  $\Psi^v$ ,  $\Psi^{\tilde{p}}$ ,  $\Psi_S^{\tilde{p}}$ ,  $\bar{\Psi}_3^u$ ,  $\bar{\Psi}_3^v$ ,  $\Psi_A^{\tilde{p}}$ ,  $\Psi_B^{\tilde{p}}$ ,  $\Psi_C^{\tilde{p}}$  and  $\Psi_D^{\tilde{p}}$ .

<sup>4</sup>We use a modified version of the code provided in the corresponding tutorial: [https://jsdokken.com/dolfinx-tutorial/chapter2/ns\\_code2.html](https://jsdokken.com/dolfinx-tutorial/chapter2/ns_code2.html).

879 **OP.** According to (26) and (27), the solution structure for the variables  $u$  and  $v$  is

$$\begin{aligned} \begin{pmatrix} u(x, y) \\ v(x, y) \end{pmatrix} &= w_1(x, y) \begin{pmatrix} u_0(y) \\ 0 \end{pmatrix} + w_2(x, y) \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ &+ w_3(x, y) \begin{pmatrix} \bar{\Psi}^u(2.2, y) - \bar{\phi}_3(x, y) \bar{\Psi}^{\tilde{p}}(x, y) / \sqrt{\nu} \\ \bar{\Psi}^v(2.2, y) \end{pmatrix} \\ &+ w_4(x, y) \begin{pmatrix} 0 \\ 0 \end{pmatrix} + w_S(x, y) \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ &+ \phi_1(x, y) \phi_2(x, y) \phi_3(x, y)^2 \phi_4(x, y) \phi_S(x, y) \begin{pmatrix} \Psi^u(x, y) \\ \Psi^v(x, y) \end{pmatrix}, \end{aligned} \quad (87)$$

880 and for  $\tilde{p}$  it would be

$$\tilde{p}(x, y) = \bar{\Psi}^{\tilde{p}}(x, y) + \phi_1(x, y) \phi_2(x, y) \phi_3(x, y)^2 \phi_4(x, y) \phi_S(x, y) \Psi^{\tilde{p}}(x, y). \quad (88)$$

881 However, we will use a simplified version of the solution structure for  $\tilde{p}$  that we define by

$$\tilde{p}(x, y) = \bar{\Psi}^{\tilde{p}}(x, y) + \phi_3(x, y) \Psi^{\tilde{p}}(x, y). \quad (89)$$

882 Without doing this, PINO has problems learning the solution properly. The functions  $\bar{\Psi}^u$ ,  $\bar{\Psi}^v$  and  $\bar{\Psi}^{\tilde{p}}$   
883 are defined in a way such that they satisfy their corresponding Dirichlet conditions according to (29),  
884 i.e.

$$\begin{aligned} \bar{\Psi}^u(x, y) &= \frac{\phi_S(x, y)}{\phi_1(x, y) \phi_2(x, y) \phi_4(x, y) + \phi_S(x, y)} u_0(y) \\ &+ \phi_1(x, y) \phi_2(x, y) \phi_4(x, y) \phi_S(x, y) \tilde{\Psi}^u(x, y), \end{aligned} \quad (90)$$

$$\bar{\Psi}^v(x, y) = \phi_1(x, y) \phi_2(x, y) \phi_4(x, y) \phi_S(x, y) \tilde{\Psi}^v(x, y), \quad (91)$$

$$\bar{\Psi}^{\tilde{p}}(x, y) = \tilde{\Psi}^{\tilde{p}}(x, y). \quad (92)$$

885 The unknown functions that need to be approximated are  $\Psi^u$ ,  $\Psi^v$ ,  $\Psi^{\tilde{p}}$ ,  $\tilde{\Psi}^u$ ,  $\tilde{\Psi}^v$  and  $\tilde{\Psi}^{\tilde{p}}$ .

## 886 **K Solutions for the Navier-Stokes equations**

887 Figure 12 shows the solutions to the Navier-Stokes benchmark problem produced by the PINN-  
888 style training of the PINO using the four different ways to prescribe boundary conditions as well  
889 as the numerical reference at the top. While all solutions agree qualitatively, for weak and semi-  
890 weak boundaries, the peak in pressure right in front of the obstacle is too weak and the velocity is  
891 underestimated.

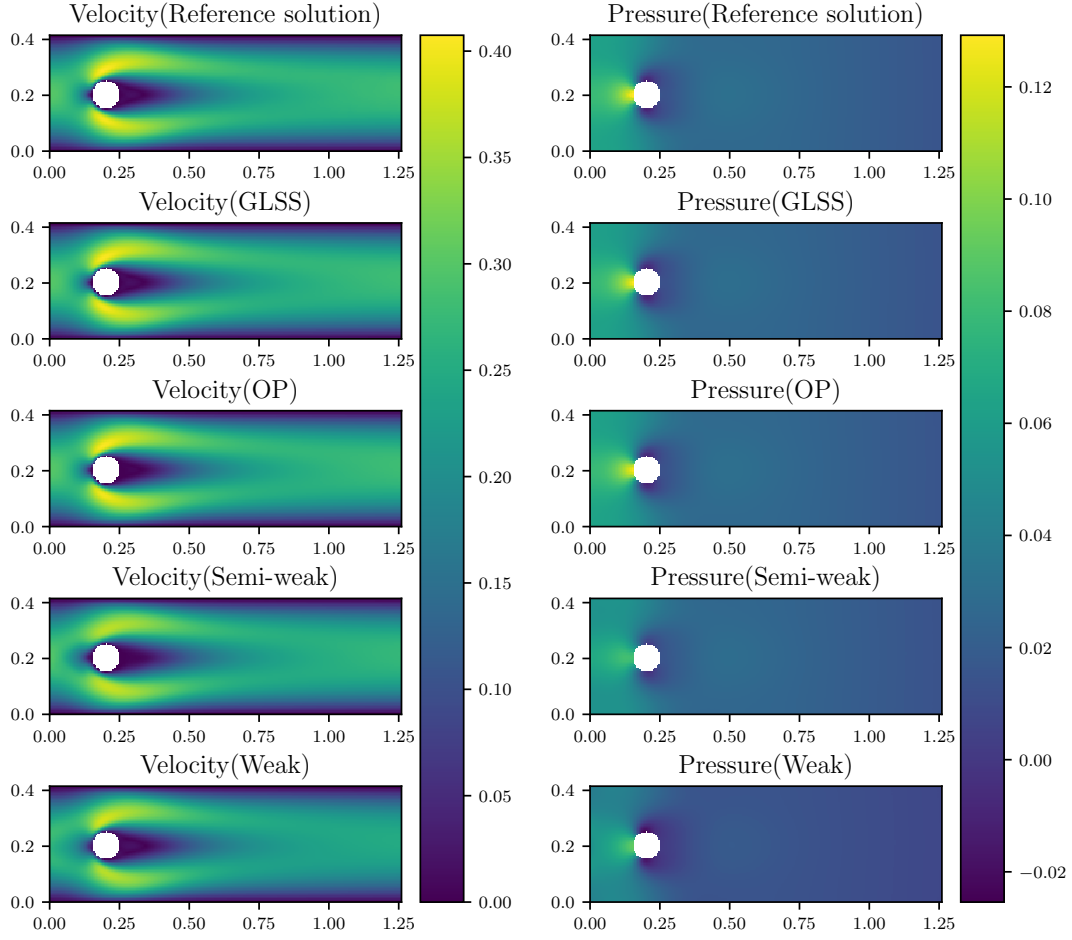


Figure 12: Velocity (left) and pressure (right) produced by the FNO using the four different ways to prescribe boundary conditions. The uppermost figure shows the numerical reference solution.