Substructuring of Arbitrary Domain Decomposition Methods

Martin J. Gander and Frédéric Nataf

1 History of substructuring in domain decomposition

Substructuring domain decomposition methods and iterative substructuring methods referred originally to a few specific methods, see e.g. [13]. The purpose of this note is to briefly present this historical development, and then to show that in fact all domain decomposition methods with exact subdomain solves can be written in substructured form, see e.g. [4] for two level- and [1] for non-linear Schwarz methods, and this can be beneficial for the run-time of domain decomposition methods when Krylov acceleration is used, because the memory requirements are drastically reduced [3].

In Civil Engineering, HARDY CROSS introduced in 1930 an interesting iterative method for solving structural problems [5], see Figure 1 for the physical intuition he had. The unknowns in the method are the moments m_j at joints, and the method corresponds to a Gauss-Seidel iteration to update one moment after the other, e.g.

> To the mathematically inclined the method will appear as one of solving a series of normal simultaneous equations by successive approximation. From an engineering viewpoint it seems simpler and more useful to think of the solution as if it were a physical occurrence. The beams are loaded or otherwise distorted while the joints are held against rotation; one joint is then allowed to rotate with accompanying distribution of the unbalanced moment at that joint and the resulting moments are carried over to the adjacent joints; then another joint is allowed to rotate while the others are held against rotation; and the process is repeated until all the joints are "eased down" into equilibrium.

Fig. 1 The Hardy Cross method from 1932.

Martin J. Gander
Section de Mathématiques, Université de Genève, e-mail: martin.gander@unige.ch

Frédéric Nataf
Laboratoire J.L. Lions, Université Pierre et Marie Curie, e-mail: nataf@ljll.math.upmc.fr

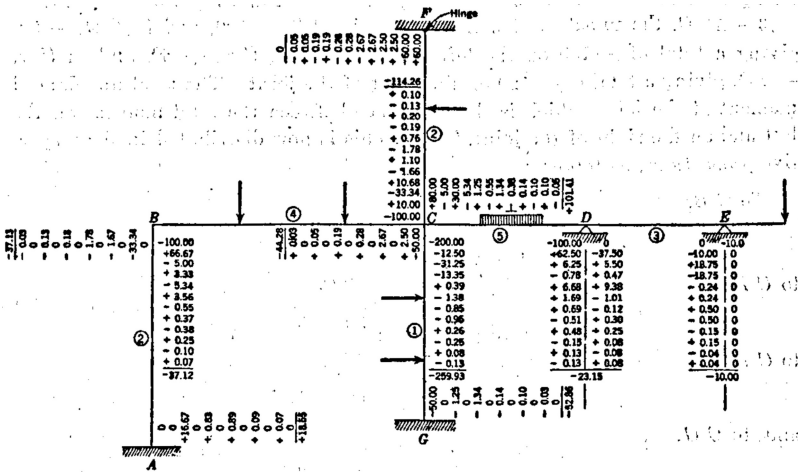


Fig. 2 Example of Hardy Cross from 1932.

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \Longrightarrow \begin{bmatrix} A_{11} & \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} m_1^{n+1} \\ m_2^{n+1} \end{bmatrix} = \begin{bmatrix} -A_{12} & \\ & \end{bmatrix} \begin{bmatrix} m_1^{n} \\ m_2^{n} \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \quad (1)$$

if there were only two moments in the system. In Figure 2, we show an original example of Hardy Cross with many moments, and how he computed the corrections. He starts with initial moment estimates, e.g. 0 at A, $(0,-100)$ at B etc., and then computes in alternating fashion how moments at joints have to be updated until convergence. The moment corrections are tabulated, and then summed. The Hardy Cross method is therefore an iterative method, and one has to know how beams (subdomains) react to loads to execute it, the beams themselves are not simulated.

In Aerospace Engineering, Janusz Przemieniecki introduced in 1963 a substructuring method where now also the substructures (subdomains) must be simulated [12], see also Figure 3:

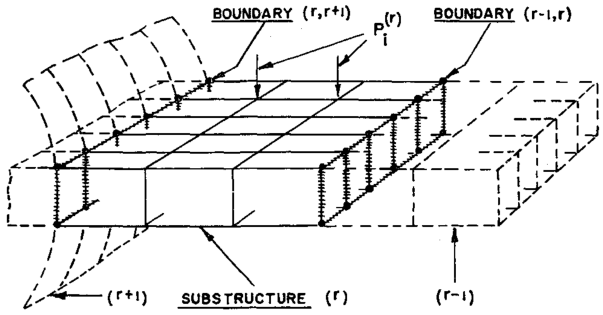


Fig. 3 Substructures of Przemieniecki from 1963.

> "The necessity for dividing a structure into substructures arises either from the requirement that different types of analysis have to be used on different components, or because the capacity of the digital computer is not adequate to cope with the analysis of the complete structure."

We see that the motivation is now quite different from Hardy Cross, including local solvers and distributed computing, and Przemieniecki describes his methods as follows:

> "In the present method each substructure is first analyzed separately, assuming that all common boundaries with adjacent substructures are completely fixed: these boundaries are then relaxed simultaneously and the actual boundary displacements are determined from the equations of equilibrium of forces at the boundary joints. The substructures are then analyzed separately again under the action of specified external loading and the previously determined boundary displacements."

In the notation of Przemieniecki, from the finite element system for the entire structure $KU = P$, unknowns are reordered into interior subdomain unknowns ('i'), and interface unknowns ('b' for 'boundary'),

$$\begin{bmatrix} K_{bb} & K_{bi} \\ K_{ib} & K_{ii} \end{bmatrix} \begin{bmatrix} U_b \\ U_i \end{bmatrix} = \begin{bmatrix} P_b \\ P_i \end{bmatrix}. \tag{2}$$

Fixing the interface unknowns U_b, one obtains for the interior unknowns $U_i = K_{ii}^{-1}(P_i - K_{ib}U_b)$. Introducing this into the equations for interface unknowns yields

$$(K_{bb} - K_{bi}K_{ii}^{-1}K_{ib})U_b = P_b - K_{bi}K_{ii}^{-1}P_i, \tag{3}$$

which is simply the Schur complement system. This system is then solved by a direct method by Przemieniecki, and once the interface values are known, the substructures can be computed, see the above quote. The method is therefore not iterative.

Wlodzimierz Proskurowski and Olof Widlund then introduced in 1976 a new Schur complement technique for capacitance matrix methods [11]:

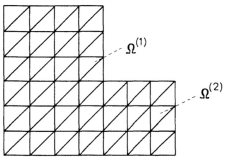
> "This new formulation leads to well-conditioned capacitance matrix equations which can be solved quite efficiently by the conjugate gradient method."

The key point here is that now the Schur complement system is solved by a Krylov method, not by a direct method, and thus the method is iterative.

Soon thereafter, in 1982, Max Dryja, the first winner of the Olof Widlund prize in domain decomposition, then introduced the seminal idea of preconditioning this Schur complement system in substructuring domain decomposition [7]:

> "The system is solved by generalized conjugate gradient method with $K^{1/2}$ as the preconditioning."

Max Dryja used an L-shaped domain Ω decomposed into two rectangles $\Omega_1 := (0, a_1) \times (0, b_2)$ and $\Omega_1 := (a_1, a_2) \times (0, b_1)$, see Figure 4, and the key matrix K here is the discrete Laplacian operator on the subdomain interface, whose square root allowed Max Dryja to get a condition number estimate which does not depend on the mesh size!



$$\gamma_0(K^{1/2}y,y)_{R^p} \leqq (Cy,y)_{R^p} \leqq \gamma_1(K^{1/2}y,y)_{R^p}$$

γ_0 *and* γ_1 *are the following constants*

$$\gamma_0 = r/(1+(1+r^2)^{1/2}), \qquad \gamma_1 = (1+2r(1+2\sigma))\,b_2/(4r\,b_1)$$
$$r = (32)^{1/2}(a_2-a_1)/b_1.$$

Fig. 4 Max Dryja inventing preconditioned iterative substructuring in 1982.

Bruno Després introduced in 1991 in his seminal PhD thesis [6] on what we call now optimized Schwarz methods a substructured formulation of a non-overlapping Schwarz method with Robin transmission conditions for Helmholtz problems; we quote directly from his PhD thesis in French:

$$\begin{aligned}
&(-\Delta-\omega^2)u_k^{n+1} = f && \text{dans } \Omega_k,\\
&(\frac{\partial}{\partial\nu_k}+i\omega)u_k^{n+1} = (-\frac{\partial}{\partial\nu_j}+i\omega)u_j^n && \text{sur } \Sigma_{kj}, \forall j,\\
&(\frac{\partial}{\partial\nu_k}+i\omega)u_k^{n+1} = 0 && \text{sur } \Gamma_k.
\end{aligned}$$

Notons $x = (x_{kj})$ le vecteur constitué des quantités de transmissions.

$$x_{kj} = \frac{1}{h}\hat{p}_{jk} + i\omega\hat{T}u_{jk}.$$

Le système précédent peut alors se mettre sous la forme

$$(I - \Pi T)x = g. \tag{7.36}$$

La matrice I est la matrice identité.

La matrice T permet de calculer les quantités de transmissions rentrantes à partir des quantités de transmissions sortantes et d'une équation de Helmholtz discrétisée à l'intérieur de chaque élément.

- **L'algorithme de décomposition de domaine qui résoud l'équation (7.36) n'est autre que $x^{n+1} = g + \Pi T x^n$.**

Equation (7.36) is the Schwarz substructured system, and the last equation is Schwarz for the first time written as an iteration on interface unknowns x^n only.

Only three years later, Frédéric Nataf, François Rogier and Eric De Sturler introduced substructured overlapping optimized Schwarz methods [10]. They considered a domain decomposition into strips, see Figure 5, and an optimized Schwarz method which can be made nil-potent, a groundbreaking result they prove in two ways, as seen directy from their manuscript:

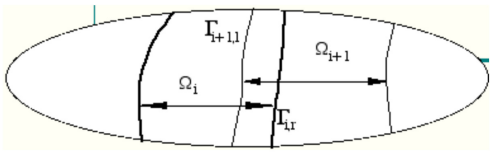


Fig. 5 Strip decomposition considered for substructured optimized Schwarz.

$$\begin{array}{c} \mathcal{L}(u_i^{n+1}) = f \text{ in } \Omega_i \\ (\frac{\partial}{\partial \vec{n}_{i,l}} - \Lambda_{i,l})(u_i^{n+1}) = (\frac{\partial}{\partial \vec{n}_{i,l}} - \Lambda_{i,l})(u_{i-1}^{n}) \text{ on } \Gamma_{i,l} \ (2 \le i \le N) \\ (\frac{\partial}{\partial \vec{n}_{i,r}} - \Lambda_{i,r})(u_i^{n+1}) = (\frac{\partial}{\partial \vec{n}_{i,r}} - \Lambda_{i,r})(u_{i+1}^{n}) \text{ on } \Gamma_{i,r} \ (1 \le i \le N-1) \\ \mathcal{C}(u_i^{n+1}) = g \text{ on } \partial\Omega \cap \partial\Omega_i \end{array} \tag{3}$$

Proposition 2.4 *The Schwarz algorithm (3) achieves convergence in N iterations, where N is the number of subdomains.*

We give two proofs. The first one is direct. The second one is based on an interpretation of (3) as an algorithm for unknowns defined on the boundaries of the subdomains. It is an introduction to the Schur method.

The second proof uses a substructured formulation of the Schwarz method, using the letter $h_{i,r \text{ or } l}$ for the interface unknowns, which gives in their manuscript

From (3), we have for $n \ge 1$

$$\begin{array}{ll} h_{2,l}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{2,l}} - \Lambda_{2,l})(S_1(h_{1,r}^n, 0, 0) + S_1(0, f, g)) \\ h_{3,l}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{3,l}} - \Lambda_{3,l})(S_2(h_{2,l}^n, 0, 0, 0) + S_2(0, h_{2,r}^n, 0, 0) + S_2(0, 0, f, g)) \\ & \vdots \\ h_{N,l}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{N,l}} - \Lambda_{N,l})(S_{N-1}(h_{N-1,l}^n, 0, 0, 0) + S_{N-1}(0, h_{N-1,r}^n, 0, 0) + S_{N-1}(0, 0, f, g)) \\ h_{N-1,r}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{N-1,r}} - \Lambda_{N-1,r})(S_N(h_{N,l}^n, 0, 0) + S_N(0, f, g)) \\ h_{N-2,r}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{N-2,r}} - \Lambda_{N-2,r})(S_{N-1}(h_{N-1,l}^n, 0, 0) + S_{N-1}(0, h_{N-1,r}^n, 0, 0) + S_{N-1}(0, 0, f, g)) \\ & \vdots \\ h_{1,r}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{1,r}} - \Lambda_{1,r})(S_2(h_{2,l}^n, 0, 0) + S_2(0, h_{2,r}^n, 0, 0) + S_2(0, 0, f, g)) \end{array}$$

The authors even show the Schur complement like system to which the substructured overlapping Schwarz method at the limit corresponds, which is simply obtained by taking the limit in the above system as the iteration index n goes to infinity, see also (4) and (5) below. For a first substructured optimized Schwarz method with coarse correction, see [9].

The key ideas for substructuring a domain decomposition method are therefore to decompose, like in all domain decomposition methods, the domain of computation Ω into subdomains Ω_j, which were historically non-overlapping for substructuring. The domain decomposition iteration is then reformulated as an iteration on interface unknowns only, which were historically moments, then Dirichlet traces, and then Robin or more generalized traces. The resulting interface systems are solved by iteration, historically Gauss-Seidel, then by Conjugate Gradients, possibly with a preconditioner for Dirichlet coupling, and finally by Schwarz iterations.

2 General concepts and examples

Domain decomposition methods for linear problems can all be written as stationary iterations of the form (see e.g. [13, Section 1.3 and 1.4], and also the examples below)

$$\mathbf{u}^{n+1} = \mathbf{u}^n + M^{-1}(\mathbf{f} - A\mathbf{u}^n), \tag{4}$$

where $\mathbf{u}^n$ can be interface values or subdomain volume solutions, and M represents the domain decomposition method, which can contain also a coarse space. This

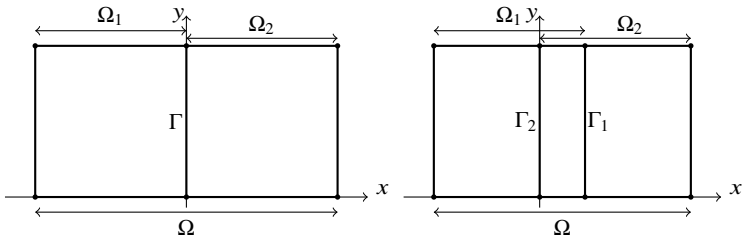


Fig. 6 Non-overlapping decomposition (left) and overlapping one (right).

iteration can be accelerated by a Krylov method: one then solves the DD iterative system (4) at the fixed point using a Krylov method. The system at the fixed point, i.e., when $n \to \infty$ and thus $\mathbf{u}^{n+1}$ and $\mathbf{u}^n$ cancel, is simply the preconditioned system

$$M^{-1}A\mathbf{u} = M^{-1}\mathbf{f}. \tag{5}$$

Solving this with a Krylov method gives much better convergence than the stationary domain decomposition iteration (4), since the error $\mathbf{e}^n := \mathbf{u} - \mathbf{u}^n$ for (4) satisfies

$$\mathbf{e}^{n+1} = \mathbf{e}^n - M^{-1}A\mathbf{e}^n = (I - M^{-1}A)^{n+1}\mathbf{e}^0, \tag{6}$$

and a Krylov method finds a much better residual polynomial than $(I - M^{-1}A)^{n+1}$,

$$\mathbf{e}^{n+1} = p_{n+1}(M^{-1}A)\mathbf{e}^0, \tag{7}$$

with $p_{n+1}(M^{-1}A)$ much smaller than $(I - M^{-1}A)^{n+1}$. For example Conjugate Gradients minimizes the energy norm $||\mathbf{e}^n||_{M^{-1/2}AM^{-1/2}}$, and GMRES minimizes the residual $||M^{-1}(\mathbf{f} - A\mathbf{u}^n)||_2$, see e.g. [2, Chapter 4.1].

Note that this same idea of acceleration also applies to non-linear problems: to accelerate a non-linear domain decomposition iteration $\mathbf{u}^{n+1} = G(\mathbf{u}^n)$ (or any non-linear fixed point iteration), one simply solves the fixed point equation $F(\mathbf{u}) := \mathbf{u} - G(\mathbf{u}) = 0$ by Newton's method, which is called non-linear preconditioning [8].

We now show several examples on how domain decomposition iterations can be substructured and then accelerated by Krylov methods. We start with the Dirichlet-Neumann method for a Poisson problem and two subomains, as shown in Figure 6 on the left. The method solves alternatingly Dirichlet and Neumann problems,

$$\begin{array}{llll} \Delta u_1^n = f & \text{in } \Omega_1, & \Delta u_2^n = f & \text{in } \Omega_2, \\ u_1^n = u_\Gamma^{n-1} & \text{on } \Gamma, & \partial_x u_2^n = \partial_x u_1^n & \text{on } \Gamma, \\ u_1^n = g & \text{on } \partial\Omega \cap \partial\Omega_1, & u_2^n = g & \text{on } \partial\Omega \cap \partial\Omega_2, \end{array} \tag{8}$$

and uses a relaxation to update the Dirichlet transmission condition,

$$u_\Gamma^n = \theta u_\Gamma^{n-1} + (1-\theta)u_2^n(\Gamma). \tag{9}$$

Using the Dirichlet to Neumann operator DtN, and the Neumann to Dirichlet operator NtD, we can write this method in substructured form, namely

$$u_\Gamma^n = \theta u_\Gamma^{n-1} + (1-\theta)\mathrm{NtD}_2(f, g, \mathrm{DtN}_1(f, g, u_\Gamma^{n-1})). \tag{10}$$

To use Krylov acceleration, we would solve this iteration at the fixed point using a Krylov method. By linearity the iteration at the fixed point yields the linear system

$$(I - \mathrm{NtD}_2(0, 0, \mathrm{DtN}_1(0, 0, \cdot))u_\Gamma = \mathrm{NtD}_2(f, g, \mathrm{DtN}_1(f, g, 0)). \tag{11}$$

Similarly, the Neumann-Neumann method for this example would be

$$\begin{array}{llll} \Delta u_i^n = f & \text{in } \Omega_i, & \Delta\psi_i^n = 0 & \text{in } \Omega_i, \\ u_i^n = u_\Gamma^{n-1} & \text{on } \Gamma, & \partial_{n_i}\psi_i^n = \partial_{n_1}u_1^n + \partial_{n_2}u_2^n & \text{on } \Gamma, \\ u_i^n = g & \text{on } \partial\Omega \cap \Omega_i, & \psi_i^n = 0 & \text{on } \partial\Omega \cap \Omega_i, \end{array} \tag{12}$$

with the interface updating relaxation

$$u_\Gamma^n = u_\Gamma^{n-1} - \theta(\psi_1(\Gamma) + \psi_2(\Gamma)). \tag{13}$$

This iteration can be written in the substructured form,

$$u_\Gamma^n = u_\Gamma^{n-1} - \theta \sum_{i=1}^{2} \mathrm{NtD}_i \left(\sum_{j=1}^{2} \mathrm{DtN}_j(f, g, u_\Gamma^{n-1}) \right), \tag{14}$$

and the solution can again be accelerated by solving with a Krylov method the Neumann-Neumann system at the fixed point,

$$\sum_{i=1}^{2} \mathrm{NtD}_i \left(\sum_{j=1}^{2} \mathrm{DtN}_j(0, 0, \cdot) \right) u_\Gamma = -\sum_{i=1}^{2} \mathrm{NtD}_i \left(\sum_{j=1}^{2} \mathrm{DtN}_j(f, g, 0) \right). \tag{15}$$

Finally, a Schwarz method for this problem and the overlapping decomposition in Figure 6 (right) would be

$$\begin{array}{llll} \Delta u_1^n = f & \text{in } \Omega_1, & \Delta u_2^n = f & \text{in } \Omega_2, \\ u_1^n = u_2^{n-1} & \text{on } \Gamma_1, & u_2^n = u_1^n & \text{on } \Gamma_2, \\ u_1^n = g & \text{on } \partial\Omega \cap \partial\Omega_1, & u_2^n = g & \text{on } \partial\Omega \cap \partial\Omega_2. \end{array}$$

To obtain a substructured formulation, we introduce the interface unknowns $\lambda^n := u_2^n|_{\Gamma_1}$, and then obtain the substructured iteration

$$\lambda^n = \mathrm{DD}_{21}(f, g, \mathrm{DD}_{12}(f, g, \lambda^{n-1})),$$

where DD_{ij} is the name for the subdomain solves and Dirichlet traceing. This iteration can again be accelerated by applying a Krylov method to the preconditioned substructured system

$$(I - \mathrm{DD}_{21}(0,0,\mathrm{DD}_{12}(0,0,\cdot))\lambda = \mathrm{DD}_{21}(f,g,\mathrm{DD}_{12}(f,g,0)).$$

For a more general substructured formulation of Schwarz methods, see [1, 3, 4].

3 Conclusions

We have seen that classical iterative domain decomposition methods can all be written in substructured form, and iterations in substructured form or in volume form are equivalent, provided exact subdomain solvers are used, see e.g. [1, 4]. Krylov acceleration in substructured form is cheaper with Krylov methods that do not have short recurrences (e.g. GMRES), because then the Krylov vectors to be stored are only of the dimension of the interfaces, not the volume unknowns [3]. It is easy to generate a substructured domain decomposition method from a volume one, one just has to apply restrictions and prolongations with interface data, see e.g. [1].

References

1. Chaouqui, F., Gander, M. J., Kumbhar, P. M., and Vanzan, T. Linear and nonlinear substructured Restricted Additive Schwarz iterations and preconditioning. *Numerical Algorithms* **91**(1), 81–107 (2022).
2. Ciaramella, G. and Gander, M. J. *Iterative methods and preconditioners for systems of linear equations*. SIAM (2022).
3. Ciaramella, G., Gander, M. J., Van Vriekingen, S., and Vanzan, T. A performance comparison of classical volume and new substructured one- and two-level Schwarz methods in PETSc. In: *Domain Decomposition Methods in Science and Engineering XXVII*. Springer (2023).
4. Ciaramella, G. and Vanzan, T. Substructured two-grid and multi-grid domain decomposition methods. *Numerical Algorithms* **91**(1), 413–448 (2022).
5. Cross, H. Analysis of continuous frames by distributing fixed-end moments. *Transactions of the American Society of Civil Engineers* **96**(1), 1–10 (1932).
6. Després, B. *Méthodes de décomposition de domaine pour la propagation d'ondes en régime harmonique. Le théorème de Borg pour l'équation de Hill vectorielle*. Ph.D. thesis, Paris 9 (1991).
7. Dryja, M. A capacitance matrix method for Dirichlet problem on polygon region. *Numerische Mathematik* **39**(1), 51–64 (1982).
8. Gander, M. J. On the origins of linear and non-linear preconditioning. In: *Domain Decomposition Methods in Science and Engineering XXIII*, 153–161. Springer (2017).
9. Japhet, C. *Conditions aux limites artificielles et décomposition de domaine: Méthode OO2 (Optimisé d'ordre 2). Application à la résolution de problèmes en mécanique des fluides*. Ph.D. thesis, CMAP, Ecole Polytechnique, Paris (1997).
10. Nataf, F., Rogier, F., and de Sturler, E. Optimal interface conditions for domain decomposition methods. Tech. rep., CMAP, Ecole Polytechnique, Paris (1994).
11. Proskurowski, W. and Widlund, O. On the numerical solution of Helmholtz's equation by the capacitance matrix method. *Mathematics of Computation* **30**(135), 433–468 (1976).
12. Przemieniecki, J. S. Matrix structural analysis of substructures. *AIAA Journal* **1**(1), 138–147 (1963).
13. Toselli, A. and Widlund, O. *Domain Decomposition Methods - Algorithms and Theory*, vol. 34. Springer (2004).