The HTFETI method variant gluing cluster subdomains by kernel matrices representing the rigid body motions

Alexandros Markopoulos, Lubomír Říha, Tomáš Brzobohatý, Ondřej Meca, Radek Kučera, and Tomáš Kozubek

IT4Innovations National Supercomputing Center, VŠB - Technical University of Ostrava, 17. listopadu 15/2172, Ostrava, Czech Republic
{alexandros.markopoulos,lubomir.riha}@vsb.cz

Abstract. The proposed algorithm called the Hybrid Total Finite Element Tearing and Interconnecting method (HTFETI) is a variant of the TFETI domain decomposition method suitable for large-scale problems with hundreds of thousands of subdomains. The floating subdomains are gathered into several groups belonging to individual clusters. We use the new idea consisting in gluing the cluster subdomains using kernel matrices defined by the rigid body motions. This technique reduces the size of the coarse problem. While the size of the coarse problem depends linearly on the number of subdomains in the classical TFETI method, it depends linearly on the number of clusters in the HTFETI method. The zero weighted averages across the interfaces of neighbouring subdomains (an alternative to the constraints enforcing the continuity across the corners used, e.g., in the FETI-DP method) improve conditioning of the resulting system of linear equations.

1 Introduction

The history of the FETI (Finite Element Tearing and Interconnecting) method [6] is longer than twenty years and over the years, numerous variants have been developed (FETI-DP method [2, 5], T(otal)FETI method [4] etc). The important impulse for development of new FETI variants was given by the implementation on more sophisticated computer architectures, where parallel processors are grouped into clusters. From the point of view of minimal communications, it is reasonable to copy the computer architecture into the FETI method that lead to the hybrid (two-level) FETI methods. The FETI–FETI-DP method proposed in [7, 8] combines the classical FETI method used on the global level with the FETI-DP method used on the clusters. In this paper, we deal with the TFETI–TFETI method that uses the TFETI method on both levels [3, 10]. It will be denoted as the H(ybrid)TFETI method. The new approach presented in this paper is called HTFETI$_{ker}$. In this method the gluing of subdomains (belonging to one cluster) is done using kernels of the local subdomains. This technique accelerates iterations like in the case of the transformation of basis discussed in [7].

In the numerical experiments we compare HTFETI$_{ker}$ with HTFETI$_{cor}$ method where the subdomains belonging to one cluster are glued by the Lagrange multipliers (LMs) corresponding to the corner nodes. Such method is similar to the FETI–FETI-DP method. The basic idea of the two-level FETI method is graphically explained in the following benchmark, in which we introduce also respective notation.

In order to simplify the presentation of the method, we use a simple cube benchmark with a hierarchical decomposition and discretization depicted in Fig. 1.

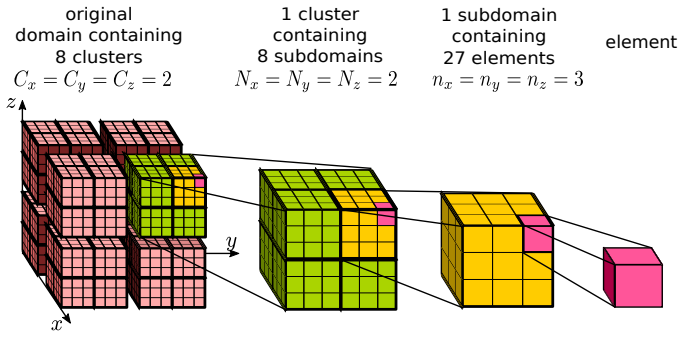

Fig. 1. Two levels of decomposition: 2 clusters ($C = 2$), 2 subdomains ($N = 2$), 3 elements ($n = 3$) in each space dimension

This hierarchical decomposition and discretization consists of three levels:

- Level 1 - decomposition into clusters is controlled by parameters C_x, C_y, and C_z (numbers of clusters in x, y, and z direction). Each cluster occupies one computational node.
- Level 2 - each cluster is decomposed into the subdomains controlled by parameters N_x, N_y, and N_z (numbers of subdomains in x, y, and z direction).
- Level 3 - each subdomain is discretized uniformly by hexahedral finite elements handled by parameters n_x, n_y, n_z (numbers of elements in x, y, and z direction).

If, for example, the number of clusters in all directions is the same $C_x = C_y = C_z = 2$, the description in the text is simplified to $C = 2$. This simplified notation is also applied to subdomains N and elements n.

2 Cluster constraints

2.1 Types of subdomains-gluing

In the following part we are going to focus on the constraints among subdomains in the cluster. All the details of the HTFETI method and also the derivation of

the algorithm can be found in [10]. The notation used in this section relates to the same paper.

Compared to the FETI method, in the described hybrid variant the neighboring subdomains are grouped into clusters using additional constraints. Together, with the commonly used joining of subdomains via corner nodes known in the FETI-DP method, we present a new technique based on the kernels of stiffness matrices. Such an approach requires a robust algorithm for factorizing singular matrices but, on the other hand, it simplifies implementation of the HTFETI method. Implicitly, it enforces zero averages across the faces between the neighbouring subdomain.

For simplification, let us use the cluster consisting of two subdomains Ω_j and Ω_k (see Fig. 2). The stiffness matrix of I-th cluster then will be

$$\tilde{\mathbf{K}}_I = \begin{pmatrix} \mathbf{K}_{j:k} & \mathbf{B}_{c,j:k}^\top \\ \mathbf{B}_{c,j:k} & \mathbf{O} \end{pmatrix} = \begin{pmatrix} \mathbf{K}_j & \mathbf{O} & \mathbf{B}_{c,j}^\top \\ \mathbf{O} & \mathbf{K}_k & \mathbf{B}_{c,k}^\top \\ \mathbf{B}_{c,j} & \mathbf{B}_{c,k} & \mathbf{O} \end{pmatrix} \tag{1}$$

which corresponds to Eq. (14) in [10] if interval $j : k$ consists of j and k only. Here, $\mathbf{K}_j$ and $\mathbf{K}_k$ are stiffness matrices, $\mathbf{B}_{c,j}$, $\mathbf{B}_{c,k}$ are linear constraints keeping both subdomains together, and $\mathbf{O}$ is a zero matrix with the appropriate size. In the next subsections, let us explain how to choose the blocks $\mathbf{B}_{c,j}$, $\mathbf{B}_{c,k}$.

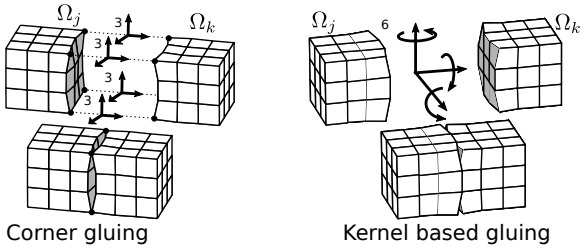

Fig. 2. Two-subdomain bonding, corners versus kernels: 3 forces per corner node (12 LM in total) / 3 forces and 3 moments per interface (6 LM in total).

Corner strategy - HTFETI$_{cor}$ method Using this method, $\mathbf{B}_{c,j}$, $\mathbf{B}_{c,k}$ are signed booleans matrices that enforce the connectivity across the corner nodes (see Fig. 2 left). The structure is similar as matrix of constraints in the FETI method (commonly denoted as $\mathbf{B}$).

Kernel strategy - HTFETI$_{ker}$ method The kernel strategy glues the domains Ω_j and Ω_k in a weaker sense using the kernel $\mathbf{R}_j$ of matrix $\mathbf{K}_j$. Instead of enforcing relative zero displacements in particular nodes belonging to the interface $\Gamma_{jk} = \overline{\Omega}_j \cap \overline{\Omega}_k$, we prescribe constraints acting onto all DOFs belonging to the face Γ_{jk}. The number of these constraints is determined by the defect d of $\mathbf{K}_j$ (and $\mathbf{K}_k$) that is $d = 6$ for the three-dimensional linear elasticity problems.

Let $\mathbf{Q}_j$ be an appropriate permutation matrix separating $\mathbf{R}_j$ into two parts:

$$\mathbf{Q}_j\mathbf{R}_j = \begin{pmatrix} \mathbf{R}_{j,\Omega_j\setminus\Gamma_{jk}} \\ \mathbf{R}_{j,\Gamma_{jk}} \end{pmatrix},$$

where $\mathbf{R}_{j,\Gamma_{jk}}$ is given by the rows of $\mathbf{R}_j$ belonging to the interface $\Gamma_{jk} = \overline{\Omega}_j \cap \overline{\Omega}_k$ and $\mathbf{R}_{j,\Omega_j\setminus\Gamma_{jk}}$ contains by the remaining rows. It is required that $\mathbf{R}_{j,\Gamma_{jk}} \in \mathcal{R}^{m\times d}$, where $m \geq d$. In the case of a three-dimensional linear elasticity problem, this requirement is always satisfied if the common interface between two neighboring subdomains is given by at least three nodes not lying in one line. The parameter m is then equal to 9 (number of all degrees of freedom belonging to this set of nodes). Then we define $\mathbf{B}_{c,j}$ and $\mathbf{B}_{c,k}$ as follows:

$$\mathbf{B}_{c,j}^\top = \mathbf{Q}_j^\top \begin{pmatrix} \mathbf{O} \\ \mathbf{R}_{j,\Gamma_{jk}} \end{pmatrix}, \quad \mathbf{B}_{c,k}^\top = \mathbf{Q}_k^\top \begin{pmatrix} \mathbf{O} \\ -\mathbf{R}_{j,\Gamma_{jk}} \end{pmatrix}, \tag{2}$$

where the permutation matrix $\mathbf{Q}_k$ maps the rows of $-\mathbf{R}_{j,\Gamma_{jk}}$ (in Ω_k) onto the corresponding rows of $\mathbf{R}_{j,\Gamma_{jk}}$ (in Ω_j). The non-sigularity of the matrices $\mathbf{B}_{c,j}\mathbf{R}_j$ and $\mathbf{B}_{c,k}\mathbf{R}_k$ guarantees that the subdomains $\boldsymbol{\Omega}_j$ and $\boldsymbol{\Omega}_k$ are properly glued together [9]. The gluing condition is schematically depicted in the right Fig. 2. The presented idea can be simply extended to the clusters with more than two subdomains. The approach is also applicable to non-singular matrices (transient problems).

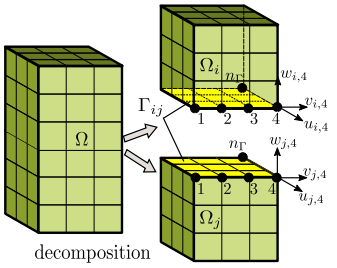

Fig. 3. Domain decomposition of Ω body into 2 subdomains Ω_i and Ω_j.

Example: Constraints assembled from the analytically computed kernel. Let us explain some general ideas regarding the analytical form available for kernels in linear elasticity . Let the nodes shared by $\boldsymbol{\Omega}_i$ and $\boldsymbol{\Omega}_j$ lying on the interface $\Gamma_{i,j} = \overline{\boldsymbol{\Omega}}_i \cap \overline{\boldsymbol{\Omega}}_j$ depicted in Fig. 3 be indexed by the set $\mathcal{G} = \{1, 2, \cdots, n_{\Gamma_{i,j}}\}$. Let the displacement vector of the g-th node $\mathbf{x}_g = (x_g,\ y_g,\ z_g) \in \boldsymbol{\Gamma}_{i,j}$ be denoted $\mathbf{u}_{i,g} = (u_{i,g},\ v_{i,g},\ w_{i,g})$ with respect to Ω_i and $\mathbf{u}_{j,g} = (u_{j,g},\ v_{j,g},\ w_{j,g})$ with respect to Ω_j. It follows from the mechanical arguments that two subdomains are kept together by 3 forces and 3 moments acting across the whole interface Γ_{ij} that avoids mutual movements and rotations. It can be achieved by zero averages

of displacements

$$\sum_{g=1}^{n_\Gamma}(u_{i,g}-u_{j,g})=0,\ \sum_{g=1}^{n_\Gamma}(v_{i,g}-v_{j,g})=0,\ \sum_{g=1}^{n_\Gamma}(w_{i,g}-w_{j,g})=0,$$

and rotations

$$\sum_{g=1}^{n_\Gamma}((u_{i,g}-u_{j,g})\cdot y_g-(v_{i,g}-v_{j,g})\cdot x_g)=0,$$

$$\sum_{g=1}^{n_\Gamma}((u_{i,g}-u_{j,g})\cdot z_g-(w_{i,g}-w_{j,g})\cdot x_g)=0,$$

$$\sum_{g=1}^{n_\Gamma}((v_{i,g}-v_{j,g})\cdot z_g-(w_{i,g}-w_{j,g})\cdot y_g)=0$$

across $\Gamma_{i,j}$. It also guarantees that the subdomains are sufficiently and optimally bonded together with the minimal number of constraints.

Apart from the natural *accelerating property*, there is also another significant feature of kernel-based $\mathbf{B}_{c,i}$. Since its constraints enforce the equality across the interface on average, the Dirichlet preconditioner acts on the whole interface as well and it is completely adopted from (T)FETI method in an unchanged form.

2.2 Rank of the cluster constraint matrix $\mathbf{B}_{c,j:k}$

Sufficient mutual gluing of all cluster subdomains realized by kernels requires 6 constraints per interface between two neighboring subdomains. The comparison with the corner strategy will be shown on the cube problem. Since the matrix $\mathbf{B}_{c,j:k}$ is always assembled without linearly dependent constraints, the rank and number of rows are equal.

Academic problem For the sake of clarity, the cube problem is uniformly decomposed into subdomains by setting: $C=1$ and $N=2,3,\cdots,10$. Thanks to a simple cube geometry and the uniform discretization and decomposition, we can derive the dependency between the number of subdomains N and the rank of the cluster matrix $\mathbf{B}_{c,j:k}$. If the corner strategy is used, three following situations can occur. The node is shared by two subdomains (then it produces $3\cdot 1$ LM), by four subdomains ($3\cdot 3$ LM) or by eight subdomains ($3\cdot 7$ LM). In the first case, the subdomains are glued using corner nodes, the dimension of $\mathbf{B}_{c,j:k}$ is

$$\operatorname{rank}\left(\mathbf{B}^{cor}_{c,j:k}\right)=21(N-1)^3+54(N-1)^2+36(N-1). \tag{3}$$

In the case $\mathbf{B}_{c,j:k}$ is assembled via parts of the kernels, each common interface generates 6 LM and the dimension is

$$\operatorname{rank}\left(\mathbf{B}^{ker}_{c,j:k}\right)=18N^2(N-1). \tag{4}$$

The ratio between "corner" and "kernel" case for $N \to \infty$ is

$$\lim_{N\to\infty} \frac{\operatorname{rank}\left(\mathbf{B}^{cor}_{c,j:k}\right)}{\operatorname{rank}\left(\mathbf{B}^{ker}_{c,j:k}\right)} = \frac{7}{6} \approx 1.1667. \tag{5}$$

In the numerical tests presented later we have used a variant with 1000 subdomains ($N = 10$) for each cluster. The kernel strategy exhibits an interesting property because it provides fewer iterations, although in the corner strategy (in this particular case) the matrix $\mathbf{B}^{cor}_{c,j:k}$ contains 23.5% more constraints.

3 Numerical test

The described algorithms were implemented into our ESPRESO (ExaScale PaRallel FETI SOlver) package developed at IT4Innovations National Supercomputing Center in Ostrava, the Czech Republic [11, 1].

For these computations we used facilities of IT4Innovations Czech national supercomputing center (www.it4i.cz), namely Salomon cluster. The Salomon cluster consists of 1008 compute nodes. Each node contains 24 core Intel Xeon E5-2680v3 processors and 128 GB RAM. The interconnect is a 7D Enhanced hypercube InfiniBand.

We varied the decomposition and discretization parameters on a cube benchmark test in order to demonstrate the scalability of our method. The cube (30 mm) is made of steel with the following parameters: Young's modulus $E = 2.1 \cdot 10^5$ MPa, Poisson's ratio $\mu = 0.3$, density $\rho = 7850$ kg/m^3, and gravity constant $g_{x_1} = 9.81$ m/s^2. The cube is fixed on the plane $x = 0$, and loaded by its own weight in the x direction.

The problem is solved by the projected preconditioned conjugate gradient method. The iterations are stopped after the relative preconditioned residual is reduced by stopping criterion to preconditioned residual $\varepsilon = 1 \cdot 10^{-4}$. The first test shows weak scalability for the benchmark depicted in Fig. 1 with one cluster, a fixed number of DOFs on each subdomain, and a variable number of subdomains. The considered parameters are: $C = 1$, $N = 2,\ 3,\ \cdots,\ 12$ and $n = 10$. The initial and last variant contain $27,783$ DOFs and $5,314,683$ DOFs, respectively. The linear system is preconditioned by the Dirichlet preconditioner. In Fig. 4 left, the problem is decomposed uniformly. Naturally, the TFETI method provides the best results. For the HTFETI$_{ker}$ method, the number of iterations slightly increases with the increasing number of subdomains N^3. The hybrid variant with corners (the HTFETI$_{cor}$ method) exhibits the worst results of all three methods. On the other hand when METIS is used as the decomposer (Fig. 4 right), the TFETI method can lose the scalability due to the irregular interface. The HTFETI$_{cor}$ method is also influenced by the decomposition, but the HTFETI$_{ker}$ method keeps the relatively same performance (a slightly increasing number of iterations) as in the uniform decomposition case.

Result of similar tests with a larger number of DOFs per subdomain (parameters: $C = 1$, $N = 2,\ 3,\ \cdots,\ 6$, $n = 20$, DOFs ranging from $206,763$ to

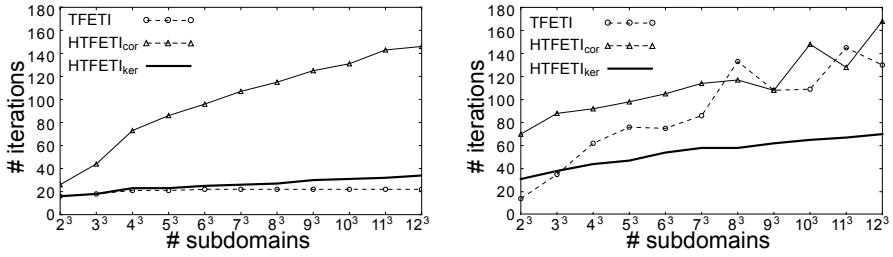

Fig. 4. Decomposition: uniform-left, METIS-right; $C = 1,\ N = 2, 3, \cdots, 12,\ n = 10$. Number of unknowns ranges from $27,783$ to $5,314,683$.

$5,314,683$) are displayed in Fig. 5. For a uniform decomposition, the TFETI and HTFETI$_{ker}$ method exhibit an equal number of iterations. It implies that if the interface is large enough (in this case 20×20 nodes versus 10×10), the TFETI method can be replaced by the HTFETI$_{ker}$ method containing one cluster. However, the HTFETI$_{ker}$ method is more expensive in preprocessing and partially also during the iterations. On the other hand, as it was already observed, when METIS is used, the TFETI method loses scalability faster, and therefore the utilization of the HTFETI method can be meaningful.

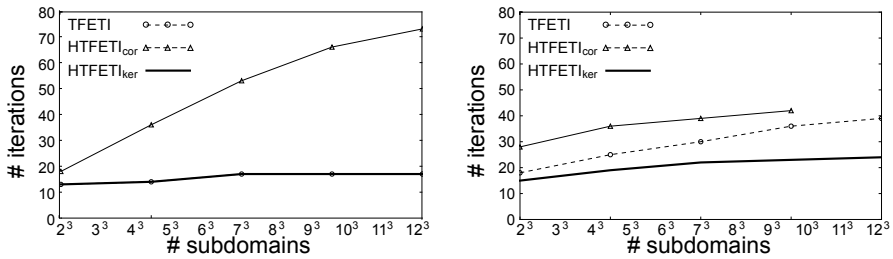

Fig. 5. Decomposition: uniform-left, METIS-right; $C = 1,\ N = 2, 3, \cdots, 6,\ n = 20$. Number of unknowns from $206,763$ to $5,314,683$.

The next set of numerical experiments in Fig. 6 shows weak scalability with the lumped preconditioner (the number of iterations on the left, solver time on the right) up to $1,259,712$ subdomains and 10.4 billion unknowns. Because of the very large number of subdomains, the TFETI method cannot be used for all the settings, and for this reason, it is not included in this comparison. However, both diagrams show weak scalability of the HTFETI method. It is also seen that the variant based on kernels requires three times fewer iterations compared to the case with corners.

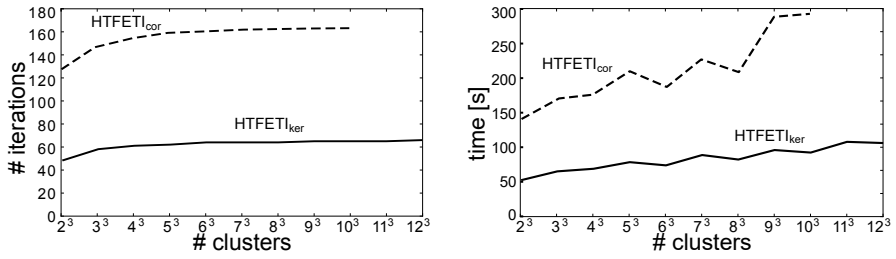

Fig. 6. HTFETI, uniform decomposition; C = 2,3,...,12, N = 9, n = 14. Number of unknowns from $4,858,2831$ to $10,390,538,091$.

4 Conclusion

This work presents the Hybrid variant of the Total FETI method.The main idea stems from the work published in [7], where the FETI and FETI-DP method are combined. Here, the presented version is the TFETI-TFETI method that uses the TFETI method on both levels. In the newly proposed variant, the subdomains are not glued together by corners but through the whole interface between each neighboring pair of subdomains via the kernels of the stiffness matrices. The numerical tests show efficiency of our algorithm. The very promising results were obtained for non-uniform decompositions. The Hybrid TFETI method based on kernels exhibits better weak scalability compared to the TFETI method.

5 Acknowledgment

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science - LQ1602 and from the Large Infrastructures for Research, Experimental Development and Innovations project IT4Innovations National Supercomputing Center LM2015070.

References

1. ESPRESO - Exascale Parallel FETI Solver, http://espreso.it4i.cz `http://espreso.it4i.cz`
2. A. Klawonn, O.B.W., Dryja, M.: Dual-primal feti methods for three-dimensional elliptic problems with heterogeneous coefficients. SIAM Journal on Numerical Analysis 40, 159–179 (2002)
3. Brzobohatý, T., Jarošová, M., Kozubek, T., Menšík, M., Markopoulos, A.: The hybrid total FETI method. In: Proceedings of the Third International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering. Civil-Comp, Ltd. (2011)
4. Dostál, Z., Horák, D., Kučera, R.: Total FETI - an easier implementable variant of the FETI method for numerical solution of elliptic PDE. Commun Numer Meth En 196, 1155–1162 (2006)

5. Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K., Rixen, D.: FETI-DP: a dual-primal unified FETI method, part i: A faster alternative to the two-level FETI method. International Journal for Numerical Methods in Engineering 50(7), 1523–1544 (2001)
6. Farhat, C., Roux, F.X., Oden, J.T.: Implicit parallel processing in structural mechanics. Elsevier Science SA (1994)
7. Klawonn, A., Rheinbach, R.: Highly scalable parallel domain decomposition methods with an application to biomechanics. ZAMM 1, 5–32 (2010)
8. Klawonn, A., Lanser, M., Rheinbach, O.: Toward extremely scalable nonlinear domain decomposition methods for elliptic partial differential equations. SIAM J. Sci. Comput. 37(6), C667–C696 (jan 2015), http://dx.doi.org/10.1137/140997907
9. Kučera, R., Kozubek, T., Markopoulos, A.: On large-scale generalized inverses in solving two-by-two block linear systems. Linear Algebra and its Applications 438(7), 3011–3029 (apr 2013)
10. Merta, M., Riha, L., Meca, O., Markopoulos, A., Brzobohaty, T., Kozubek, T., Vondrak, V.: Intel xeon phi acceleration of hybrid total feti solver. Advances in Engineering Software 112(Supplement C), 124 – 135 (2017), http://www.sciencedirect.com/science/article/pii/S0965997816302745
11. Říha, L., Brzobohatý, T., Markopoulos, A., Meca, O., Kozubek, T.: Massively parallel hybrid total feti (htfeti) solver. In: Proceedings of the Platform for Advanced Scientific Computing Conference. pp. 7:1–7:11. PASC '16, ACM, New York, NY, USA (2016), http://doi.acm.org/10.1145/2929908.2929909