



Hierarchical Multi-Resolution Finite Element Model for Soft Body Simulation

Matthieu Nesme*

François Faure†

Yohan Payan‡

GMCAO-TIMC & EVASION-GRAVIR / IMAG-INRIA

Abstract

The complexity of most surgical models has not allowed interactive simulations on standard computers. We propose a new framework to finely control the resolution of the models. This allows us to dynamically concentrate the computational force where it is most needed.

Given the geometrical model of an object to simulate, we first compute a bounding box and then recursively subdivide it where needed. The cells of this octree structure are labelled with mechanical properties based on material parameters and fill rate. An efficient physical simulation is then performed using hierarchical hexahedral finite elements. The object surface is used for rendering and to apply boundary conditions.

Compared with traditional finite element approaches, our method dramatically simplifies the task of volume meshing and increases the propagation of the deformations.

1 Introduction

1.1 Context

Soft body simulation is a growing research domain, for communities such as Computer Aided Surgery, Virtual Reality or Computer Graphics. Computer aided surgery (CAS) aims at assisting surgeons for the realization of diagnostic and therapeutic gestures in a rational and quantitative way in order to increase safety and accuracy [Taylor et al. 1996]. While the first designed systems focused on orthopaedics, researchers addressed more recently anatomical structures that cannot be considered as "rigid" as they are mainly composed of biological soft tissues. The corresponding CAS systems therefore need to take into account the displacements of the structures as well as their deformations. In most cases, authors propose to build biomechanical models of the anatomical structures and use these models to predict, in the most accurate way, the tissue deformations induced by the surgical gesture.

Virtual Reality (VR), in its interactions with the Medical community, has recently provided surgical simulation systems (Cotin et al., 1996). As for the flight simulators used to train pilots, the idea is that these surgical VR systems could be a great help in the learning and training processes, allowing the surgeon to acquire, for example, some difficult hand-eye coordinations, to repeat several times the most difficult gestures or to choose the best surgical procedure for a given pathological case. As for the CAS systems, deformable models have been included into the simulators, with constraints in term of robustness and computation times.

Computer Graphics (CG) has developed methods for the visually plausible animation of complex physical objects such as clothes and hairs. Considerable speedups have been obtained for stiff flexible bodies using implicit time integration, which allows arbitrary time

steps to be performed. Significant advances have also been made using hierarchical modeling and the control of levels of detail.

These three communities now converge towards the same needs, in terms of soft body modelling: accuracy, robustness and interactivity (i.e. fast computation times). Indeed, from one side the CAS community is now looking for models that could be per-operatively used, with possible real-time re-planning of the surgical gesture. From the other side, the VR and CG communities now focus on the accuracy of the deformations, in order to be as realistic as possible, in comparison with real data. In this framework, some recent works, coming from these communities, try to provide mechanical models with innovative implementations that preserve a continuous modelling context (with, for most of the works, a numerical resolution through the Finite Element Method) while proposing improvements, in terms of robustness and computation times. In addition, models built in the CAS or VR contexts need to be adapted to each patient anatomy. This point is particularly challenging (and time consuming) when a patient-specific Finite Element mesh needs to be defined.

Next part tries to summarize all of these recent works (part 1.2), while a new modelling approach is introduced in part 2. An example of implementation is presented in part 3 before providing some results (part 4).

1.2 Related Work

1.2.1 Deformable Models

In order to improve the computational efficiency of continuous biomechanical models, researchers have proposed new approaches concerning (1) the Finite Element discretization, (2) the dynamical integration and (3) the numerical resolution methods.

Because of the need for speed, the first interactive methods were based on precomputed matrix inversion [Cotin et al. 1996]. To extend these methods to large deformations frameworks, a non-linear computation of the strains is used in [Debunne et al. 2001; Picinbono et al. 2003]. Recently proposed methods favor a new approach based on the decomposition of the displacement of each element into a rigid motion and a pure deformation tractable linearly in the local frame [Eitzmuß et al. 2003; Hauth and Straßer 2004; Müller and Gross 2004; Nesme et al. 2005]. These methods allow a large displacements and rotations framework.

In the animation community, implicit integration methods have become popular, thanks to the iterative solution based on conjugate gradient presented by [Baraff and Witkin 1998]. Although these methods permit large time step, they become "expensive" when a fast propagation of the deformation is suited, since they require a lot of iterations to solve the system accurately. On the contrary, explicit integrations do not use iterations but require small time steps to maintain stability. Therefore, if a fast propagation of the deformations is needed, both implicit and explicit integration schemes remain computationally expensive. To face this problem, hierarchical methods have been proposed, providing an improvement of the propagation of the deformations (see for example the hierarchi-

*e-mail: matthieu.nesme@imag.fr

†e-mail: francois.faure@imag.fr

‡e-mail: yohan.payan@imag.fr

cal solvers proposed by [Terzopoulos and Fleischer 1988; Wu and Tendick 2004]).

In order to adapt the numerical solution schemes to the adequate level of details, authors have proposed to adapt the Finite Element (FE) mesh according to the actual state of the model (in terms of displacement, strain or stress). They propose therefore a multi-resolution FE approach ([Debunne et al. 1999; Debunne et al. 2001; Wu et al. 2001]). The idea is for example to define, for a given anatomical structure, different FE meshes, from a very coarse one to a full refined one. If boundary conditions induce small deformations inside the structure, the coarse mesh is sufficient for providing accurate FE discretization. On the contrary, a dense mesh is used where the deformation is high.

1.2.2 Patient Specific Models

Building a patient specific FE mesh is usually a delicate and time consuming task. From imaging exams such as Computed Tomography or Magnetic Resonance, data can be collected through image segmentation, providing for example the external contour of an organ as well as intrinsic sub-structures. From these geometrical data, 2D or 3D FE meshes are built and used to discretize the mechanical formulation. Arguing against the time consuming component of this patient-specific FE mesh elaboration, authors have proposed to use a reference FE mesh (an "atlas", built once from a reference patient geometry) and to match this mesh to each new patient morphology ([Couteau et al. 2000; Castellano-Smith et al. 2001; Clatz et al. 2005]). These methods reduce the time needed to build the patient-specific FE models, but they are still very specific and may alter the quality of the elements during the registration phase.

2 Contribution

Our approach proposes to merge a multi-resolution description with a Hierarchical FE integration. It is supposed to provide a numerical scheme that can be used for any type of mechanical description, from a small deformation framework to hyperelasticity. The objective is only to gather some methods already proposed in the literature in order to improve the propagation of the deformations as well as the efficiency of the computation according to the mechanical and geometrical state of the soft body. The specificity of the method is that a global 3D mesh is defined from a classical octree division of a bounding box including the soft body. Therefore, no FE mesh is needed to specifically model the 2D or 3D geometry of the body. Indeed, the FE computations are applied to the 3D mesh defined by the octree. To improve the propagation of the deformations, a hierarchical basis is defined to interpolate the FE computations, from the global parent cell defined by the bounding box to each child cell of the octree. Moreover, the multi-resolution scheme is used to decide, for a given state of the body, which levels of the octree should serve as basis for (1) FE computation and (2) 3D rendering: for example, only regions with high strains level should use a dense octree level for FE computation, while regions that are not displayed on the screen because they are not seen by the camera should use a coarse octree mesh level for the 3D rendering.

2.1 Octree Mesh and Multi-resolution

The first step consists in defining the complete 3D octree mesh as presented in figure 1-a. Starting from a cubic bounding box of the body, an iterative algorithm is used to divide each "parent" cube in order to generate eight "child" cubes. The cubes that do not

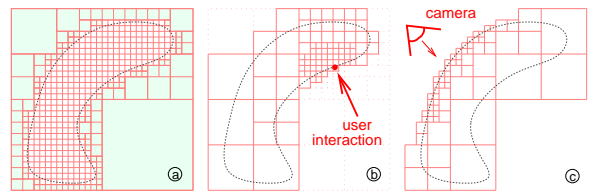


Figure 1: An example of adaptivity. (a) leaves of the octree mesh = the finest level of details. (b) mechanical leaves = the finest mechanical level. (c) geometric leaves = the finest geometric level.

contain any part of the body are removed from the octree mesh. The remaining cubes are again divided, so that each of them will generate 8 new cubes. A maximal level of division N_{max} is defined once, leading to the "maximal density" octree mesh (figure 1-a). Using this octree mesh architecture as a baseline, two intermediate resolutions will be defined at each time step of the global computation of the system, namely the octree resolution N_{FE} used for the FE mesh interpolation (figure 1-b) and the octree resolution N_{Rend} used for the rendering display (figure 1-c). These two resolutions can change during the solving of the system, according to the changes in the boundary conditions as well as the location of the camera that looks at the scene. For example, one condition to define whether a given level of the octree mesh is suitable for a given point of the body consists in looking at the strain rate. If it is sufficiently low, this means that the current resolution is sufficient. On the contrary, if a high strain variation is observed, a denser mesh is preferable around this region of the body leading to the use of the child cells of the actual octree element. Once the N_{FE} level is reached, the corresponding octree 3D mesh is used for the FE computation. In order to limit the influence of cells that would contain a small amount of the body (cells located at the surface of the body), it is proposed to ponderate the rheology of these cells by their filling ratio as illustrated in figure 4.

2.2 Hierarchical FE Bases

A function decomposed in a hierarchical basis is modeled using a rough approximation based on a few broad-range sample points, along with a number of recursively narrower-range sample points encoding local detail added to the approximation. Each value of the function is thus the sum of shape functions with various radius of influence. This approach allows one to easily control the level of detail by simply inserting or dropping control values where desired [Stollnitz et al. 1996]. Another nice feature of this approach is to considerably speed up the convergence of shape optimization, as shown in geometric modeling [Gortler and Cohen 1995]. It has been successfully applied to finite element methods [Grinspun et al. 2002; Capell et al. 2002].

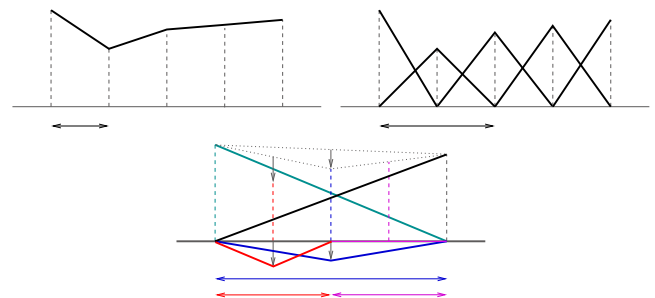


Figure 2: Illustrations of the finite element (top-left), basis function (top-right) and hierarchical basis function (bottom) points of view with linear interpolation. In the basis function point of view, the solution corresponds to a combinaison of functions associated with the nodes. In the hierarchical point of view, the influence support of these functions varies.

In the case of the octree mesh introduced above, the position stored for each vertex is relative from its parents position. Therefore, only vertices from the root cell (i.e. the cubic bounding box that includes the body) have real position in the 3D space. At start, all others child cells have a null relative position which only depends on their parents. Figure 2 illustrates the FE interpolation that will be provided with hierarchical linear functions.

3 An Implementation

This section proposes an example of the implementation of the previously presented hierarchical approach. This implementation uses the Cauchy's deformation tensor and a viscoelastic material, and is therefore valid only for small displacements and small deformations.

3.1 Shape functions

The fundamental principle of finite element is to interpolate desired values $u(p)$ using values u_i defined at the nodes of the elements:

$$\mathbf{u}(\mathbf{p}) = \sum_i \mathbf{H}_i(\mathbf{p}) \mathbf{u}_i \quad (1)$$

with $\mathbf{H}_i(\mathbf{p})$ the interpolation matrix for vertex \mathbf{p} concerning the element node number i .

To simplify computations, interpolations are done in natural coordinates in a local frame $(r, s, t) \in [-1, 1]^3$. To adapt computations in the global frame, the Jacobian operator \mathbf{J} which relates the natural coordinate derivatives to the local coordinate derivatives is needed. $\frac{\partial h}{\partial r} = \mathbf{J} \frac{\partial h}{\partial x}$.

As exprimed below, the interpolation matrix \mathbf{H} depends on the position of the lattice vertices influencing on the interpolated object vertex. Eight types of interpolation matrices exists, one for each node position in a cell:

$$\mathbf{H}_i = \begin{bmatrix} h_i & 0 & 0 \\ 0 & h_i & 0 \\ 0 & 0 & h_i \end{bmatrix}, \quad h_i(r, s, t) = \frac{1}{8} (1 \pm r)(1 \pm s)(1 \pm t)$$

3.2 Mechanics

3.2.1 Equations of Motion

The dynamic evolution of the deformations is modeled as the classical system of secondary order ordinary differential equation:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}$$

To build matrices \mathbf{M} , \mathbf{D} , \mathbf{K} and vector \mathbf{f} , the standard method used to simulate viscoelastic solids is considered. A classical approximation for the damping matrix \mathbf{D} is to use $\mathbf{D} = \alpha\mathbf{K} + \beta\mathbf{M}$. External forces are computed as

$$\mathbf{f} = \int_V \mathbf{H}^T \mathbf{f} dV$$

For example the gravity is applied like $\mathbf{f}_{gravity} = \int_V \mathbf{H}^T \rho \mathbf{g} dV$, with ρ the material density and \mathbf{g} the acceleration field. To integrate the volume over the natural coordinates the volume differential dV must be written in natural coordinate: $dV = \det \mathbf{J} dr ds dt$

Our approach induces differences with classical formulation concerning the displacement u which is not defined in global space coordinates, but is defined hierarchically. Only displacements of the vertices of the root cell are in space coordinates. The displacement of others vertices is relative from their parents. To build the mass matrix \mathbf{M} and the stiffness matrix \mathbf{K} , not only finest elements are considered. Indeed, for each elements along the hierarchy we take into account all nodal functions that influence the considered element as exprimed in algorithms 1 and 2. \mathbf{C} is the stress-strain matrix relating the material properties and the strain-displacement matrix \mathbf{B}_i is obtained by differentiation of h_i with respect to local coordinate and premultiplying the result by the inverse of the Jacobian operator:

$$\mathbf{B}_i = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial h_i}{\partial r} & 0 & 0 \\ 0 & \frac{\partial h_i}{\partial s} & 0 \\ 0 & 0 & \frac{\partial h_i}{\partial t} \\ \frac{\partial h_i}{\partial s} & \frac{\partial h_i}{\partial t} & \frac{\partial h_i}{\partial r} \\ 0 & \frac{\partial h_i}{\partial t} & \frac{\partial h_i}{\partial s} \\ \frac{\partial h_i}{\partial t} & \frac{\partial h_i}{\partial r} & 0 \end{bmatrix}$$

Algorithm 1 BUILD MATRICES \mathbf{K} AND \mathbf{M}

```

for each cell do
  for each vertex  $i$  defined at level of cell do
    INTEGRATE( $\mathbf{B}_i, \mathbf{B}_i, \mathbf{H}_i, \mathbf{H}_i, \mathbf{C}_{cell}, \mathbf{J}_{cell}$ )
  for each vertex  $j \neq i$  defined at level of cell do
    INTEGRATE( $\mathbf{B}_i, \mathbf{B}_j, \mathbf{H}_i, \mathbf{H}_j, \mathbf{C}_{cell}, \mathbf{J}_{cell}$ )
  end for
  for each ancestor of cell do
    for each vertex  $j$  defined at level of ancestor do
      take function  $h_j$  between range of cell in ancestor // detail in section 3.2.2
      INTEGRATE( $\mathbf{B}_i, \mathbf{B}_j, \mathbf{H}_i, \mathbf{H}_j, \mathbf{C}_{cell}, \mathbf{J}_{cell}$ )
      INTEGRATE( $\mathbf{B}_j, \mathbf{B}_i, \mathbf{H}_j, \mathbf{H}_i, \mathbf{C}_{cell}, \mathbf{J}_{cell}$ )
    end for
  end for
end for
end for
// Note that some computations can be omitted in considering the symmetric aspect
of matrices  $\mathbf{K}$  and  $\mathbf{M}$ :  $\mathbf{K}_{i,j} = \mathbf{K}_{j,i}^T$  and  $\mathbf{M}_{i,j} = \mathbf{M}_{j,i}^T$ 

```

Algorithm 2 INTEGRATE($\mathbf{B}_i, \mathbf{B}_j, \mathbf{H}_i, \mathbf{H}_j, \mathbf{C}, \mathbf{J}$)

$$\mathbf{K}_{i,j} = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \mathbf{B}_i^T \mathbf{C} \mathbf{B}_j \det \mathbf{J} dr ds dt$$

$$\mathbf{M}_{i,j} = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \mathbf{H}_i^T \mathbf{H}_j \det \mathbf{J} dr ds dt$$

3.2.2 Range Definition of the Interpolation Function

Using natural coordinates, all integrations are performed between -1 and 1 , corresponding to the range of the current cell. However the functions based on ancestor cells have an influence radius higher than the size of the child cells. We thus have to use a modified version of ancestor functions, such as h_j illustrated in 1D in figure 3.

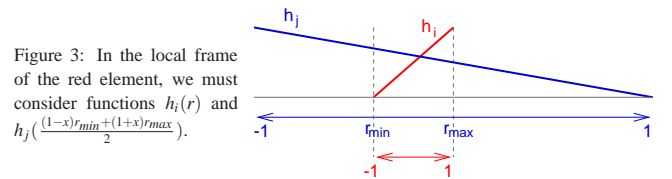


Figure 3: In the local frame of the red element, we must consider functions $h_i(r)$ and $h_j(\frac{(1-x)r_{min} + (1+x)r_{max}}{2})$.

4 Results

Figure 4 shows an octree mesh for a surfacic liver model.

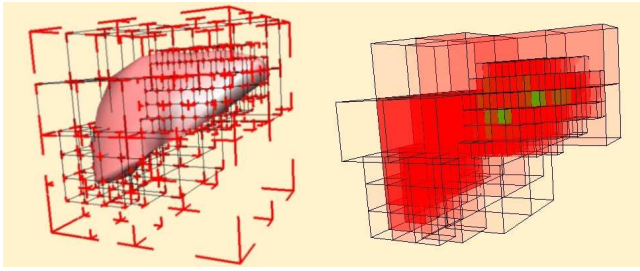


Figure 4: A multiresolution octree-mesh for a liver: the entire mechanical hierarchy and the density of mechanical leaves.

Table 1 compares the number of iterations necessary to converge using the nodal approach (i.e. the classical non-hierarchical one) against the hierarchical approach. Two examples are considered for several numbers of elements, the first one consists in a cubical fixed beam subject to gravity, while in the second a force is applied to a corner of the beam. As expected, the convergence is faster using the hierarchical approach. When the corner is pulled, the other end moves directly, whereas in the nodal model it is necessary to propagate the deformation along all elements.

Number of elements		1	8	64	512
example 1 (gravity)	nodal	1	13	55	146
	hierarchical	1	11	27	47
example 2 (boundary force)	nodal	8	50	87	198
	hierarchical	6	24	37	52

Table 1: Number of CG iterations until convergence, on two examples on a cubical fixed beam.

This faster propagation is useful in case of real-time simulation when only few iterations can be performed at each step. Using hierarchy, a small number of iterations (approximately ten) provides a much more accurate result, as illustrated in figure 5 that plots the convergence speed of the second example of table 1.

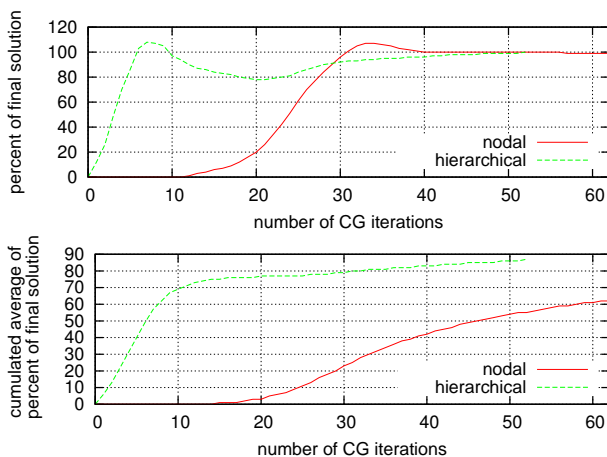


Figure 5: Convergence speed.

5 Conclusion

We proposed in this paper a hierarchical multiresolution technique to animate soft bodies. This new approach based on an octree mesh permits to work on various geometrical representations of an object without needing to provide a volumetric mesh of this object. Using the hierarchical approach improves the propagation and permits to simulate more rigid materials. Despite it, we do not yet obtain better results in terms of computation time because an optimized structure is difficult to set up. To make this work usable, it will be necessary to integrate criteria of adaptivity (automatic definition of N_{FE} and N_{Rend} values), and to take into account effective boundary conditions.

References

- BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *SIGGRAPH '98*.
- CAPELL, S., GREEN, S., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2002. A multiresolution framework for dynamic deformations. In *SCA '02*.
- CASTELLANO-SMITH, A., HARTKENS, T., SCHNABEL, J., HOSE, D., LIU, H., HALL, W., TRUWIT, C., HAWKES, D., AND HILL, D. 2001. Constructing patient specific models for correcting intraoperative brain deformation. In *Lecture Notes in Computer Science*, vol. 2208, 1091–1098.
- CLATZ, O., SERMESANT, M., BONDIU, P.-Y., DELINGETTE, H., WARFIELD, S. K., MALANDAIN, G., AND AYACHE, N. 2005. Realistic simulation of the 3d growth of brain tumors in mr images including diffusion and mass effect. In *IEEE Transactions on Medical Imaging*, vol. 24(10), 1334–1346.
- COTIN, S., DELINGETTE, H., CLEMENT, J.-M., TASSETTI, V., MARESCAUX, J., AND AYACHE, N. 1996. Volumetric deformable models for simulation of laparoscopic surgery. In *Computer Assisted Radiology*.
- COUTEAU, B., PAYAN, Y., AND LAVALLÉE, S. 2000. The mesh-matching algorithm: an automatic 3d mesh generator for finite element structures. In *Journal of Biomechanics*, vol. 33/8, 1005–1009.
- DEBUNNE, G., DESBRUN, M., BARR, A. H., AND CANI, M.-P. 1999. Interactive multiresolution animation of deformable models. In *Eurographics Workshop on Computer Animation and Simulation*.
- DEBUNNE, G., DESBRUN, M., CANI, M.-P., AND BARR, A. H. 2001. Dynamic real-time deformations using space and time adaptive sampling. In *SIGGRAPH '01*.
- ETZMUSS, O., KECKEISEN, M., AND STRASSER, W. 2003. A Fast Finite Element Solution for Cloth Modelling. *Proc Pacific Graphics*.
- GORTLER, S. J., AND COHEN, M. F. 1995. Hierarchical and variational geometric modeling with wavelets. In *S3D '95*.
- GRINSPUN, E., KRYSL, P., AND SCHRÖDER, P. 2002. Charms: a simple framework for adaptive simulation. In *SIGGRAPH '02*.
- HAUTH, M., AND STRASSER, W. 2004. Corotational simulation of deformable solids. In *Proc WSCG*.
- MÜLLER, M., AND GROSS, M. 2004. Interactive virtual materials. In *Proc Graphics Interface*.
- NESME, M., PAYAN, Y., AND FAURE, F. 2005. Efficient, physically plausible finite elements. In *Eurographics (short papers)*, 77–80.
- PICINBONO, G., DELINGETTE, H., AND AYACHE, N. 2003. Non-linear anisotropic elasticity for real-time surgery simulation. *Graph. Models*.
- STOLLNITZ, E. J., DEROSE, T. D., AND SALESIN, D. H. 1996. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann Publishers, Inc.
- TAYLOR, R., LAVALLÉE, S., BURDEA, G., AND MOSGES, R. 1996. *Computer integrated surgery: Technology and clinical applications*. Cambridge, MA: MIT Press.
- TERZOPOULOS, D., AND FLEISCHER, K. 1988. Modeling inelastic deformation: viscoelasticity, plasticity, fracture. In *SIGGRAPH '88*.
- WU, X., AND TENDICK, F. 2004. Multigrid integration for interactive deformable body simulation. In *ISMS*, 92–104.
- WU, X., DOWNES, M. S., GOKTEKIN, T., AND TENDICK, F. 2001. Adaptive non-linear finite elements for deformable body simulation using dynamic progressive meshes. In *EG 2001 Proceedings*.