Proceedings of the International Association for
Shell and Spatial Structures (IASS) Symposium 2013
„BEYOND THE LIMITS OF MAN"
23-27 September, Wroclaw University of Technology, Poland
J.B. Obrębski and R. Tarczewski (eds.)

Development of a Grasshopper Add-on for Interactive Exploration of Shapes in Equilibrium

Masaaki Miki[1], Jun Mitani[2] and Takeo Igarashi[3]

[1] The University of Tokyo, Japan/ JST ERATO, Japan, *masaakim@acm.org*
[2] University of Tsukuba, Japan / JST ERATO
[3] The University of Tokyo, Japan/ JST ERATO

Summary: We developed a set of Grasshopper components which enables interactive physics simulation based on the finite element method (FEM). We named this add-on Ricecooker. As a computational method, it is an explicit nonlinear FEM solver that can treat material and geometric nonlinearities. Unlike the usual FEM software, various parameters such as weight coefficients, material coefficients, and the coordinates of the fixed nodes can be changed during a computation through an interactive interface with real-time visualization. Some nonlinear constraint conditions can be also defined b using the provided components, and the constraint values such as the lengths of rigid bars can be also changed during a computation.

Keywords: *Dynamic relaxation method, Grasshopper, Shape-finding, Finite element method, Interactive interface*

1. INTRODUCTION

Recently, in the field of architectural design, some architects radically challenge to highly complex geometries. With no doubt, this is based on the advancement of the construction technologies, decreasing initial implementation cost of machine tools that are used for rapid prototyping such as 3D printers and laser cutters, and some other innovations related to the digital fabrication. With this new stream, the combination of the Rhinoceros ® (Fig. 1) and Grasshopper ® (GH) (Fig. 2) is used by preference in architectural design processes, especially in the very early stages.

The Rhinoceros is a commercial CAD software which is specialized in NURBS-based modeling but also supports various geometry types such as mesh surfaces and three-dimensional solids. The GH is an add-on for the Rhinoceros that allows us a component-based generative modeling. With GH, we can define a complex rule to generate geometries just by visually connecting components provided. Still, only few engineers are using the GH because the GH only supports geometric manipulations and the engineers are specialized in at least one theorem related to physics, e.g. mechanics, illumination, and thermodynamics.

Incidentally, the GH allows us to develop custom components. We believe that the GH custom components are ideal for engineers from some reasons as a platform to quickly implement our special computational methods. First, from inside a GH custom component, the developer can access to all the geometry definitions and geometric manipulations provided by the Rhinoceros. Second, the user-interface provided with a GH custom component is always common with the GH standard components. This means that a GH custom component is always very friendly with architects who have been already familiar with the GH operations. Finally, because the GH custom components are described by C#, old Fortran subroutines can be called from them via C++ wrapping codes. The old Fortran subroutines are indeed obsolete but we still need them in some specific cases. Consequently, we claim that the position of the GH custom components is just on the boundary between the engineer's and architect's sides and hence we expect that the GH custom components written by engineers can bridge the engineer's works and the architect's works. The Kangaroo [1] and the Ladybug [2] are known as the pioneers of such custom components that provides engineering functionalities but are mainly used by architects.

In this paper, we explain Ricecooker [3] (Fig. 3), which is a set of custom GH components developed by us. This enables interactive exploration of shapes in equilibrium. For example, both shape-finding problems in tension structures and large-deformation problems in hyper-elastic solids can be solved by using the Ricecooker. This functionality is quite similar to that provided by the Kangaroo indeed, but our components are directly connected to the finite element method (FEM). Even though there is a gap between the Kangaroo and FEM, this gap can be filled up because the Kangaroo is classified as a component that is based on the Dynamic Relaxation (DR) method [4] and the DR method is potentially to be used as an explicit solver in an FEM.

One might compare the Ricecooker with physics simulation engines that are built in some commercial three-dimensional modelling software such as Maya ® and 3D Studio Max ®. As described above, the major difference of our component with those is that the theoretical accuracy of the solutions is guaranteed by the FEM and quadrilateral and brick elements are provided. Note that in most physics simulation engines, only triangle and tetrahedron elements are supported. If it is compared to commercial FEM software, the Ricecooker is superior to them in that it provides an interactive change of various parameters in real-time. As an FEM, the Ricecooker is an explicit FEM solve that can treat geometric and material nonlinearities.

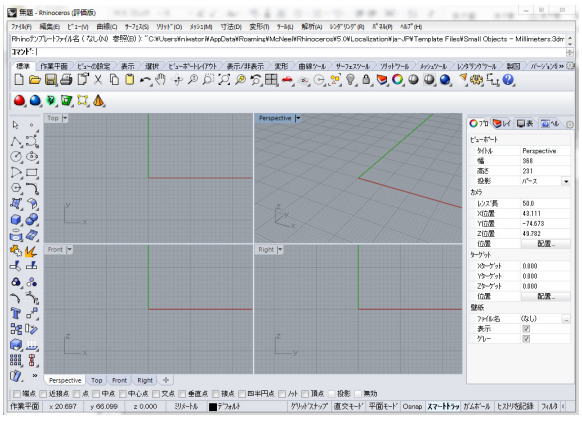


Fig. 1 Rhinoceros®

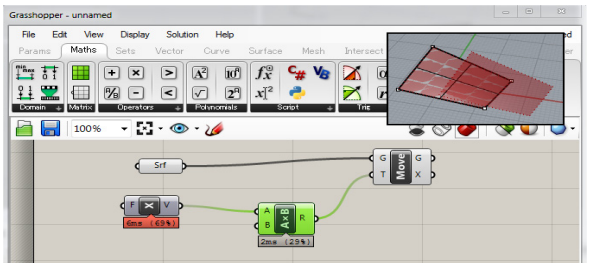


Fig. 2 Grasshopper®

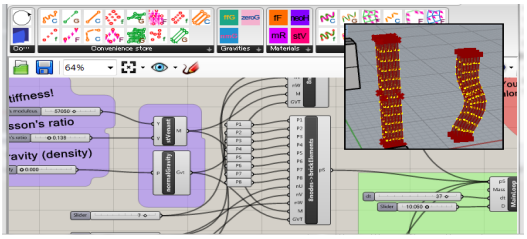
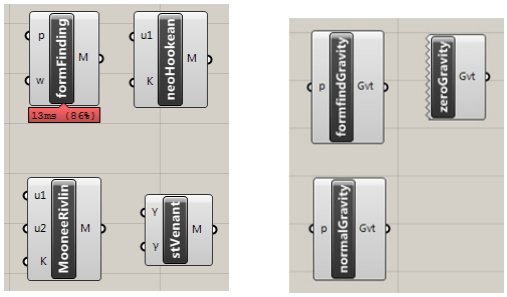


Fig. 3 Ricecooker (Developed by us)

2. IMPLEMENTATIONS

2.1. Material and gravity definitions



(a) Material definitions (b) Gravity definitions

Fig. 4 Material and gravity definitions

Because the Grasshopper has a component-based architecture, most components developed by us require inputting a material and gravity definitions. Both are provided in the form of components as shown in Fig. 4. Because these definitions are dimension-free, if they are input to a three-dimensional element such as a tetrahedron or a brick element, they automatically calculate an integral over a three-dimensional domain numerically and if they are input to a two-dimensional element such as a triangle or a quadrilateral element, an integral over a two-dimensional domain is numerically computed and so on.

As shown in Fig. 4 (a), the Ricecooker has four hyper-elastic material definitions. The three of them are the typical hyper-elastic materials, St. Venant solid, Neo-Hookean solid and Mooney-Rivling solid. Parameters in such materials such as the Young's modulus and the Poisson's ratio can be changed by a user during an iterative computation. The last material is a special material that can be used for shape-finding analyses of lightweight tension structures. If this material definition is input to one-dimensional elements, such as line elements

$$\sum_j w_j L_j^p(\boldsymbol{x}) \tag{1}$$

is added to the total strain energy function, where w_j and L_j are a weight coefficient (a real value) and the length of the j-th line element respectively, and p is the power. Both w_j and p can be changed during an iterative computation. We agree that one might think that p=2 is always used but we think that a number that is greater than 2 such as 4 is sometimes useful in shape-finding analyses. The first author this paper once reported such examples, in which p=4 works better than p=2 [5]. When this material definition is input to two-dimensional elements such as triangle and quadrilateral elements,

$$\sum_j w_j S_j^p(\boldsymbol{x}) \tag{2}$$

is added to the total strain energy function, where S_j is the area of the j-th triangle or quadrilateral element. When p is set to 1, the sum of the element areas is minimized and then, a discrete minimal surface would be obtained. Even this works fine, the solution is not unique because if a node moves along the surface, the total surface area does not change. To let the solution be unique, $p > 1$ can be used. From our experience, even if we set p to 2, the obtained solution seems to be always sufficiently close to a minimal surface.

As shown in Fig. 4 (b), the Ricecooker has three gravity definitions. The first one is used to just ignore the effect of gravity. The "normalGravity" is used in most cases, in which the volume force is proportional to the material density. In a large-deformation analyses of hyper-elastic bodies, the material density may change as the elements deform. However, in form-finding analyses, the material density should not change as the elements deform. In such a case, the "formfindingGravity" can be used instead. Because these gravity definitions are also dimension-free, a surface load can be easily defined by a combination of the "normalGravity" component and two-dimensional elements such as triangle and quadrilateral elements.

2.2. Elements

The Ricecooker supports two different types of elements. The first ones are first order elements such as line, triangle and tetrahedron elements. We call this group as simplex elements. The second ones are second order isoparametric elements such as line, quadrilateral and brick elements. An element in the first group has only one integrating point. An element in the second group has 3, 9, or 27 integrating points.

Most components provided with the Ricecooker are to generate elements automatically based on the input geometries and some parameters. For example, a "four nodes to quadrilateral elements" component shown in Fig. 5 has four point type objects, two integer numbers and the aforementioned material and gravity definitions as input parameters. Then, the component distributes quadrilateral (two-dimensional isoparametric) elements between the input points based on the specified integer numbers. The material and gravity definitions are connected to all the elements generated by the component. The interpolating function used is the same shape functions used in the isoparametric elements.

Fig. 6 shows another typical component that generates brick elements based on 8 point type objects input to the component.

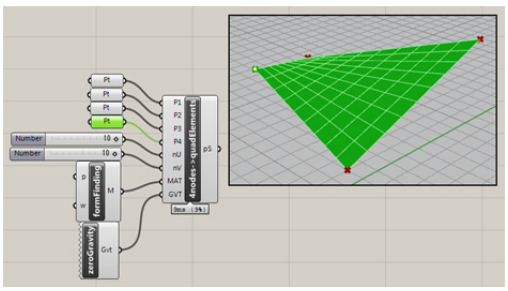


Fig. 5 A typical component provided with the Ricecooker

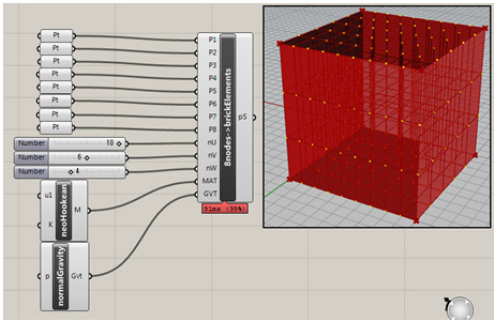


Fig. 6 Another typical component provided with the Ricecooker

2.3. Rigid constraint conditions

The Ricecooker can constrain a length, a area or a volume of an element to a specific value. This constraint value can be changed during an iterative computation. The Ricecooker can also constrain a sum of element lengths, element areas, or element volumes. Currently, only components that can constrain lengths are provided. In Fig. 7, a component that can constrain a distance between two points and a component that can constrain the total length of a polyline are shown. The lengths can be changed during an iterative computation.

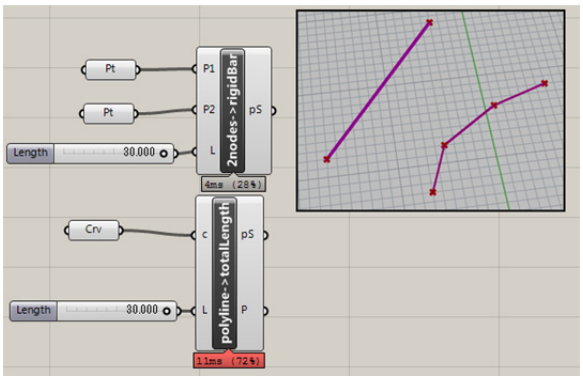


Fig. 7 Constraint conditions

2.4. Main loop

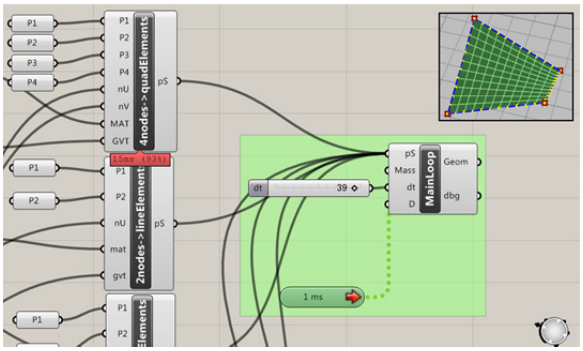


Fig. 8 A Mainloop component and particle systems

Every component that generates finite elements outputs a particle system (denoted by pS). All the particle systems output from components must be connected to a "Mainloop" component so that the Ricecooker can gather all information about finite elements (Fig. 8). The GH does not support iterative computations originally but a user can connect a Timer component to the Mainloop component so that one cycle defined in the Mainloop component is called repeatedly.

In the context menu of a Mainloop component, there is a menu named "Go". When it is hit, the one cycle is called iteratively by the Timer. The one cycle is one step of a simple Dynamic Relaxation (DR) method [4]. It is important that at the end of each one cycle, the current shape of the model under computation is displayed by Rhinoceros. In order to solve equality constraint conditions, we extended the DR method. In the extension, we use a Moore-Penrose type pseudo inverse matrix twice in each step of the DR method. The detail of this technique is excluded from this paper, because it is under submitting to a journal but its basic idea is similar to the multiplier estimate used in the projected gradient method [5]. In other words, even though the projected gradient method does not usually refer to Lagrange multipliers, the projected gradient method can be redefined as a steepest descent method in which Lagrange multipliers are estimated in each step by using the pseudo inverse matrices.

Because the DR method is based on the Newtonian mechanics, when the hysteresis of convergence is visualized in real-time, it is likely a real object moving based on the laws of physics such that users may feel comfortable when using the Ricecooker. Besides, although the convergence rate of the DR method is not very high, the computational cost of one cycle is considerably low or almost minimum in various computational schemes; hence the DR method is a suitable choice to develop an interactive interface that allows changing parameters in real-time.

2.5. Controllable parameters

During an iterative computation, some parameters in the material and gravity definitions can be changed via sliders as shown in Fig. 9. Also, a user can fix some nodes by using some specific components provided and such fixed nodes can be moved by a mouse operation during an iterative computation as shown in Fig. 10.

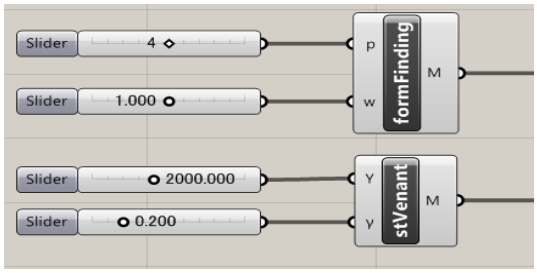


Fig. 9 Parameters

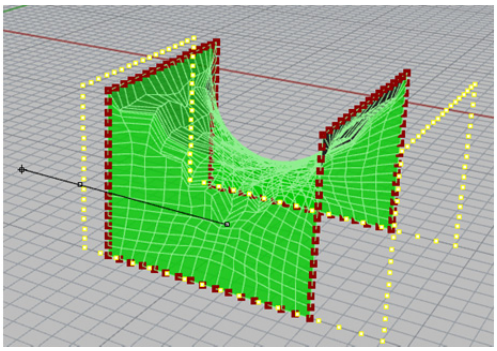


Fig. 10 Moving fixed nodes

3. EXAMPLES

When a curve is approximated by a set of line elements and their lengths are constrained to an equal value and an equally distributed nodal forces is applied to it, a discrete catenary can be obtained as shown in Fig. 11. During a computation, the lengths of the line elements and the coordinated of fixed nodes (both ends) can be changed interactively in real-time.

Fig. 12 shows an iterative computation to find a shape of minimal surface under constraint conditions. It is known that a soap film always minimizes its surface area and the shape of a soap film between two circles is known as catenoids. In Fig. 12, 8 strings of which lengths are constrained are installed to a minimal surface such that the obtained

surface is a minimal surface but a minimal surface under constraint conditions and it is different from catenoids (In Japan, this is described by K. Kawaguchi in [7]).

Fig. 13 shows a parameter study of a tensioned membrane structure, Tanzbrunnen Koln. In this analysis, the lengths of the rigid bars were treated as constraint conditions and each length can be changed during a computation.

Fig. 14 shows a large-deformation analysis of a Neo-Hookean solid. A equally distributed surface load is applied on top of the cube and its magnitude could be changed interactively during the computation in real-time.

4. SUMMARY

We described a set of custom Grasshopper components developed by us, the Ricecooker. This allows us doing physics simulation in a CAD software and in which continuum mechanics and nonlinear materials are fully considered. The Ricecooker is based on the nonlinear FEM and geometric and material nonlinearities can be treated. Additionally, without terminating computation, a user can change various parameters interactively in real-time such that the user can study various possible shapes that are in equilibrium. These characteristics are the major advantage to the conventional FEM software. By using the Ricecooker, shap-findings of catenaries, minimal surfaces, and tension structures, and large-deformation analyses of hyper elastic bodies can be performed.

5. REFERENCES

[1] Kangaroo, Grasshopper Groups, http://www.grasshopper3d.com/group/kangaroo

[2] Ladybug, Grasshopper Groups, http://www.grasshopper3d.com/group/ladybug

[3] Ricecooker, Grashopper Groups, http://www.grasshopper3d.com/group/ricecooker

[4] Day A. S., "An introduction to dynamic relaxation", The engineer, Vol. 219 (1965), pp. 218–221.

[5] M. Miki and K. Kawaguchi, "Extended Force Density Method for Form Finding of Tension Structures," *Journal of the International Association for Shell and Spatial Structures*, vol. 51, no. 4, pp. 291–303, 2010.

[6] J. B. Rosen, "The gradient projection method for nonlinear programming. Part I. Linear constraints," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 1, pp. 181–217, 1960.

[7] K. Kawaguchi, W. Ke, M. Miki, "Minimal Surface with Constraint Conditions and Steepest Descent Method", J. Struct. Constr. Eng. (published in Japanese), vol. 73, no. 632, pp. 1773–1777, 2008.

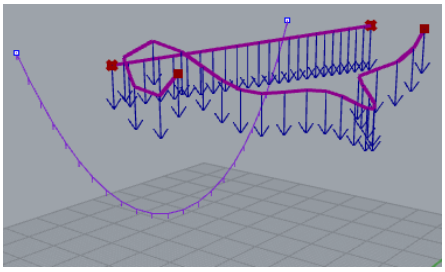


Fig. 11 Shape-finding of a catenary

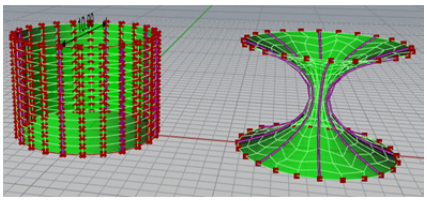


Fig. 12 A minimal surface under constraint conditions

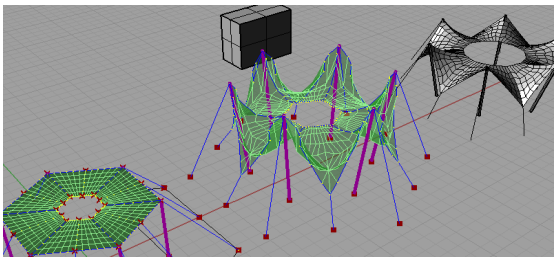


Fig. 13 Shape-finding of the Tanzbrunnen Koln

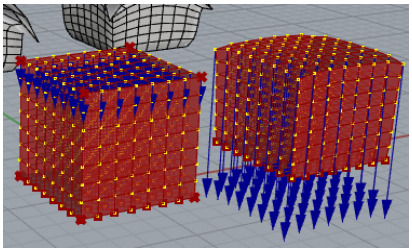


Fig. 14 Large-deformation analysis of a hyper-elastic solid

APPENDIX A. FORMULATIONS

In this section, we briefly describe the FEM formulations used by us.

A.1 Shape and displacement

Let us compose an n-dimensional column vector from all the x, y, and z coordinates of the nodes in a deformed configuration as

$$\boldsymbol{x} = [x_1 \cdots x_n]^T. \tag{3}$$

We do not eliminate the coordinates of the fixed nodes from $\boldsymbol{x}$. Bothe coordinates of free and fixed nodes are contained in $\boldsymbol{x}$. Similarly, we pack all the coordinates of the nodes in a reference configuration as

$$\bar{\boldsymbol{x}} = [\bar{x}_1 \cdots \bar{x}_n]^T. \tag{4}$$

In this paper, a $\bar{}$ put on a symbol indicates that the symbol is treated as a constant and does not subject differentiation. Normally, an undeformed state is chosen as the reference configuration, on which the stress tensor field vanishes. A displacement vector field is defined by $\boldsymbol{u} = \boldsymbol{x} - \bar{\boldsymbol{x}}$. Only two of $\{\bar{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}\}$ are independent and we choose $\{\bar{\boldsymbol{x}}, \boldsymbol{x}\}$ consistently although $\{\bar{\boldsymbol{x}}, \boldsymbol{u}\}$ is chosen in many textbooks. This is because we consider the shape itself as the unknown variable.

We implemented two different types of elements. The first ones are first order elements such as line, triangle and tetrahedron elements (Fig. 15 (a)). We call this group as simplex elements. The second ones are second order isoparametric elements such as line, quadrilateral and brick elements (Fig. 15 (b)). An element in the first group has only one

integrating point. An element in the second group has 3, 9, or 27 integrating points.

Let $X = (\theta^1, \cdots, \theta^N)$ be the local coordinate of a point on an *N*-dimensional element. Let $\boldsymbol{r} = [x\ y\ z]^T$ be the global coordinate (three-dimensional Cartesian coordinate) of the point *X*. For either a simplex element or an isoparametric element, we can write

$$\boldsymbol{r}(X) = \boldsymbol{N}(X)\boldsymbol{x}, \tag{5}$$

where $\boldsymbol{N}(X)$ is a $3\times n$ sparse matrix that is a composition of shape functions. The global coordinate of a point *X* in the reference configuration can be also represented in the same manner as

$$\bar{\boldsymbol{r}}(X) = \boldsymbol{N}(X)\bar{\boldsymbol{x}}. \tag{6}$$

In either the simplex elements or the isoparametric elements, we use embedded coordinate system so that the integral domain does not change during deformation and some tensors such as the Green-Lagrange strain become simpler forms.

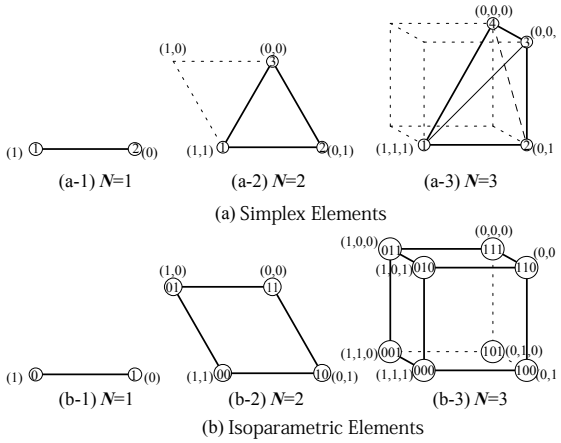


Fig. 15 Elements (local coordinates are represented with () and node indices are represented with ○)

A.2 Riemannian metric and Green-Lagrange strain

On a point $X = (\theta^1, \cdots, \theta^N)$ in an element, the covariant base vectors adjunct with the local coordinate system can be computed by

$$\boldsymbol{g}_i(X) = \boldsymbol{C}_i\boldsymbol{x}, \tag{7}$$

where

$$\boldsymbol{C}_i(X) = \partial_i\boldsymbol{N}. \tag{8}$$

On each point in the element, a *Riemannian* metric, which is an *N*×*N* symmetric matrix, is computed by

$$g_{ij} = \boldsymbol{x}^T\boldsymbol{B}_{ij}\boldsymbol{x}, \tag{7}$$

where $\boldsymbol{B}_{ij} = \frac{1}{2}\left(\boldsymbol{C}_i^T\boldsymbol{C}_j + \boldsymbol{C}_j^T\boldsymbol{C}_i\right)$. In the following, we use the *Einstein* summation convention. Additionally, we regard g_{ij} as representing the matrix itself as well as its (*i*, *j*) component. In the reference configuration, the *Riemannian* metric is also computed by

$$\bar{g}_{ij} = \bar{\boldsymbol{x}}^T\boldsymbol{B}_{ij}\bar{\boldsymbol{x}} \tag{9}$$

Let us define the inverse matrices of the *Riemannian* metrics by

$$g^{ij} = \left(g_{ij}\right)^{-1},\ \bar{g}^{ij} = \left(\bar{g}_{ij}\right)^{-1}. \tag{10}$$

Additionally, we define *N*-dimensional volume elements by

$$\mathrm{dv^N} = \sqrt{\det g_{\mu\nu}}\, d\theta^1 \cdots d\theta^N\text{, and} \tag{11}$$

$$\mathrm{d\bar{v}^N} = \sqrt{\det \bar{g}_{\mu\nu}}\, d\theta^1 \cdots d\theta^N. \tag{12}$$

A length, a area or a volume of the *j*-th element is given by

$$v_j^N = \int_{\Omega_j} \mathrm{dv}^N, \tag{13}$$

where Ω_j represents the integral domain of the *j*-th element. The variation of v_j^N is given by

$$\delta v_j^N = \frac{1}{2}\int_{\Omega_j} g^{\alpha\beta}\delta g_{\alpha\beta}\ \mathrm{dv^N}, \tag{14}$$

where δ is a variational operator. Here, we used an identity $\delta\sqrt{\det g_{\mu\nu}} = \frac{1}{2}g^{\alpha\beta}\delta g_{\alpha\beta}\sqrt{\det g_{\mu\nu}}$. By replacing $g^{\alpha\beta}$ with general functions of g_{ij} and $\bar{g}_{ij}$, we get a general form of δv_j^N and it is

$$\delta w_j^N = \frac{1}{2}\int_{\Omega_j} T^{\alpha\beta}\,\delta g_{\alpha\beta}\mathrm{dv^N}, \tag{15}$$

where $T^{\alpha\beta}$ is a general function expressed in terms of $\{g_{ij}, \bar{g}_{ij}, g^{ij}, \bar{g}^{ij}\}$. If we regard $T^{\alpha\beta}$ as a component of *Cauchy* stress tensor, δw_j can be regarded as a virtual work done by the stress acting on the element. Actually, because the component of a *Green-Lagrange* strain tensor is expressed as

$$E_{ij} = \frac{1}{2}(g_{ij} - \bar{g}_{ij})\,, \tag{16}$$

we get $\frac{1}{2}\delta g_{ij} = \delta E_{ij}$ and by using the relation between $T^{\alpha\beta}$ and the 2nd Piola-Kirchhoff stress tensor $S^{\alpha\beta}$

$$S^{\alpha\beta} = T^{\alpha\beta}\frac{\sqrt{\det g_{\mu\nu}}}{\sqrt{\det \bar{g}_{\mu\nu}}}, \tag{17}$$

we can rewrite δw_j^N as

$$\delta w_j^N = \int_{\Omega_j} S^{\alpha\beta}\,\delta E_{\alpha\beta}\mathrm{d\bar{v}^N}. \tag{18}$$

This is nothing but the elemental virtual work that repeatidly appears in the nonlinear FEM.

When a function $f\left(g_{ij}, \bar{g}_{ij}\right)$ such that $S^{\alpha\beta} = 2\frac{\partial f}{\partial g_{\alpha\beta}}$ exists, we can write

$$\delta w_j^N = \delta\left[\int_{\Omega_j} f\left(g_{ij}, \bar{g}_{ij}\right)\mathrm{d\bar{v}^N}\right]. \tag{19}$$

In this case, $f\left(g_{ij}, \bar{g}_{ij}\right)$ represents a strain energy density in the element. Normally, we start with Eq. (19) and derive Eq. (18). In this work, we started with Eq. (13) and derived Eq. (15). Then, we use Eq. (15) consistently. This is because we take some problems into consideration that do not require setting up reference configurations. The minimal surface problem is representative of such problems. However, $T^{\alpha\beta}\left(g_{ij}, \bar{g}_{ij}\right)$ should be chosen as that strain energy density $f\left(g_{ij}, \bar{g}_{ij}\right)$ truly exists. Such materials that strain energy densities can be found are called hyper-elastic materials. We implemented St.Venant, Neo Hookean, and Mooney-Rivlin solids. Each of them can be explicitly expressed in terms of four matrices $\{g_{ij}, \bar{g}_{ij}, g^{ij}, \bar{g}^{ij}\}$.

A.3 Equivalent nodal force vector

Let us define the gradient vector of a real-valued function of $\boldsymbol{x}$ by

$$\nabla f = \left[\frac{\partial f}{\partial x_1} \quad \cdots \quad \frac{\partial f}{\partial x_n}\right]. \tag{20}$$

Then, the gradient vector of g_{ij} can be computed by

$$\nabla g_{ij} = 2\boldsymbol{x}^T\boldsymbol{B}_{ij}. \tag{21}$$

An equivalent nodal force vector that can balance with the given *Cauchy* stress field can be obtained by just altering δ with ∇ in δw^N and it is expressed as

$$\boldsymbol{\omega}_j(\boldsymbol{x}) = \frac{1}{2}\int_{\Omega_j} T^{\alpha\beta}\,\nabla g_{\alpha\beta}\mathrm{dv^N}. \tag{22}$$

In general, Eq. (22) cannot be computed analytically and hence the Gaussian quadrature is commonly used. The numerically approximated $\boldsymbol{\omega}_j(\boldsymbol{x})$ obtained by the Gaussian quadrature is

$$\tilde{\boldsymbol{\omega}}_j(\boldsymbol{x}) = \frac{1}{2}\sum_U \phi_U\left[T^{\alpha\beta}\nabla g_{\alpha\beta}\sqrt{\det g_{\mu\nu}}\right]_{(j,I_U)}, \tag{23}$$

where U and ϕ_U are the index of an integrating point and its weight coefficient. Additionally, $[\circ]_{(j,I_U)}$ represents the value inside the square brackets at the U-th integrating point in the j-th element.

A.4 Geometric and material nonlinearities

Let us examine the 2nd order derivatives of $g_{\alpha\beta}$ in order to clarify what kind of nonlinearities are considered.

First, we rewrite $\tilde{\boldsymbol{\omega}}_j$ as

$$\tilde{\boldsymbol{\omega}}_j(\boldsymbol{x}) = \frac{1}{2}\sum_U \phi_U \left[S^{\alpha\beta}\nabla g_{\alpha\beta}\sqrt{\det \bar{g}_{\mu\nu}}\right]_{(j,I_U)} \tag{24}$$

Second, $\tilde{\boldsymbol{\omega}}_j(\boldsymbol{x})$ is differentiated and we get

$$\nabla\tilde{\boldsymbol{\omega}}_j(\boldsymbol{x}) = \frac{1}{2}\sum_U \phi_U \left[\left(S^{\alpha\beta}\nabla^2 g_{\alpha\beta} + \frac{\partial S^{\alpha\beta}}{\partial g_{\gamma\eta}}\nabla g_{\gamma\eta}^T \nabla g_{\alpha\beta}\right)\sqrt{\det \bar{g}_{\mu\nu}}\right]_{(j,I_U)}. \tag{25}$$

Here, $\nabla^2 f$ is the Hessian of a real-valued function $f(\boldsymbol{x})$. The $\nabla g_{\alpha\beta}$ is corresponding to a strain-displacement matrix in a linear FEM (often denoted by $\boldsymbol{B}$). The $\frac{\partial S^{\alpha\beta}}{\partial g_{\gamma\eta}}$ is corresponding to a stress-strain matrix (often denoted by $\boldsymbol{D}$). As the result, the second term in [ ] is corresponding to the linear stiffness matrix often expressed by $\boldsymbol{B}^T\boldsymbol{D}\boldsymbol{B}$. In addition, the first term in [ ] is nothing but the matrix that is called geometric stiffness matrix in the nonlinear FEM. This means that $\nabla\tilde{\boldsymbol{\omega}}_j(\boldsymbol{x})$ is nothing but the matrix that is called tangent stiffness matrix in the nonlinear FEM. Hence, if $\tilde{\boldsymbol{\omega}}_j$ is used, even though geometric stiffness matrix is not computed explicitly, it means that the geometric stiffness is taken into account.

Additionally, because the $\tilde{\boldsymbol{\omega}}_j$ only contains $S^{\alpha\beta}$ and does not contain $\frac{\partial S^{\alpha\beta}}{\partial g_{\gamma\eta}}$, $S^{\alpha\beta}$ does not need to be continuous with respect to g_{ij}. In other words, if only a one to one mapping from g_{ij} to $S^{\alpha\beta}$ or $T^{\alpha\beta}$ is given, $\tilde{\boldsymbol{\omega}}_j$ can be computed. Hence, the Ricecooker (or the DR method in general) can treat various types of nonlinear materials such as shown by Fig. 16 (a)-(d). As an exception, materials that depend on hysteresis as shown by Fig. 16 (e) cannot be treated by the Ricecooker. In addition, the Ricecooker can treat a rigid material as shown by Fig. 16 (f). This is described in the body.

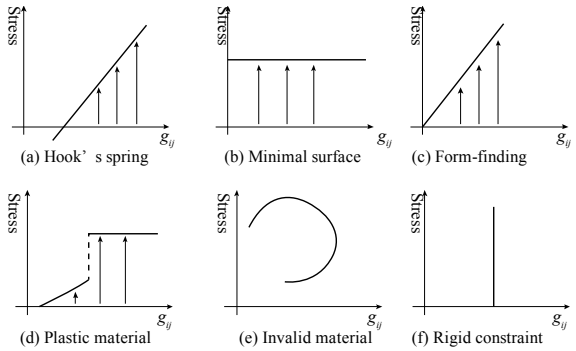


Fig. 16 Various nonlinear materials