V International Conference on Textile Composites and Inflatable Structures
STRUCTURAL MEMBRANES 2011
E. Oñate, B. Kröplin and K.-U.Bletzinger (Eds)

Direct Minimization Approaches on Static Problems of Membranes

M. MIKI*, K. KAWAGUCHI†

* Department of Architecture, School of Engineering, the University of Tokyo
komaba 4-6-1, merugo-ku, 153-8505, Tokyo, Japan
email: mikity@iis.u-tokyo.ac.jp, Web page: http://mikity.wikidot.com/

† Institute of Industrial Science, the University of Tokyo
komaba 4-6-1, merugo-ku, 153-8505, Tokyo, Japan
email: kawaken@iis.u-tokyo.ac.jp , Web page: http://space.iis.u-tokyo.ac.jp/

Key words: Principle of Virtual Work, Three Term Method, Membrane, Continuum Mechanics, Dual Estimate, Direct Minimization, Galerkin Method

Abstract. Within this work, direct minimization approaches on static problems of membranes are discussed. In the first half, standard direct minimization methods are discussed. Some form-finding analyses of tension structures are also illustrated as simple direct minimization approaches. In the second half, the principle of virtual works for cables, membranes, and 3-dimensional bodies are examined and they are approximated in a common way by using Galerkin method. Finally, some examples that direct minimization approaches can solve are reported.

1 INTRODUCTION

It is widely known that direct minimization approaches are sometimes very effective for solving simple static problems [2], such as minimal surface problem. This work aims to propose a common framework that can solve various types of static problems by direct minimization approaches. In this framework, not only simple problems but also general problems of continuum bodies can be solved by direct minimization approaches.

In chapter 2, standard direct minimization methods are discussed. The dual estimate is also introduced to include constraint conditions into direct minimization approaches. In chapter 3, the principle of virtual works for cables, membranes, and 3-dimensional bodies are examined and they are approximated in a common way by Galerkin method. In chapter 5, it is pointed out that some principle of virtual works can be solved by direct minimization methods even though their objective functions that are to be minimized are not clear.

2. DIRECT MINIMIZATION APPROACHES

2.1 Three Term Method without Constraint Conditions

For form-finding of cable-net structures, let us discuss the following stationary problem:

$$\Pi(\boldsymbol{x}) = \sum_{j=1}^{m} w_j L_j{}^2(\boldsymbol{x}) \rightarrow \text{stationary} , \tag{1}$$

where w_j, L_j are the weight coefficient and the length of the *j*-th cable respectively. In addition,

$$\boldsymbol{x} \equiv \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^T, \text{ and } \nabla f \equiv \begin{bmatrix} \dfrac{\partial f}{\partial x_1} & \cdots & \dfrac{\partial f}{\partial x_n} \end{bmatrix}, \tag{2}$$

where $\{x_1, \cdots, x_n\}$ is a set of unknown variables and ∇f is the gradient of a function *f*.

It has been reported by the authors [1] that Eq. (1) is the problem that simply represents the Force Density Method [3]. While making use of inverse matrix is proposed in the original article, direct minimization approach can also solve the same problem.

In the following discussion, suppose $\{x_1, \cdots, x_n\}$ represents the Cartesian coordinates of the free nodes and remark that those of the fixed nodes are eliminated beforehand and directly substituted into each $L_j(\boldsymbol{x})$.

The stationary condition (to be solved) of Eq. (1) is expressed as follows:

$$\nabla \Pi = \sum_j 2 w_j L_j \nabla L_j = \boldsymbol{0}. \tag{3}$$

When this problem is solved by direct minimization methods, $\nabla \Pi$ is used as the standard search direction. The simplest direct minimization recursive strategy is given by

$$\begin{aligned} \boldsymbol{r}_{Current} &= \frac{\nabla \Pi^T}{|\nabla \Pi|} \circ (\boldsymbol{x} = \boldsymbol{x}_{Current}), \\ \boldsymbol{x}_{Next} &= \boldsymbol{x}_{Current} - \alpha \boldsymbol{r}_{Current}, \end{aligned} \tag{4}$$

which is usually called the steepest decent method; however, in the relation with the following discussion, we shall call Eq. (4) as **2-term method**. In Eq. (4), $\boldsymbol{r}_{Current}$ is normalized because too large $\boldsymbol{r}_{Current}$ or too small $\boldsymbol{r}_{Current}$ sometimes causes trouble. In addition, α is a unique parameter which can be adjusted manually for controlling the step-size.

By the way, the following remedy of Eq. (4) sometimes provides a remarkable improvement of global convergence efficiency:

$$\begin{aligned} \boldsymbol{r}_{Current} &= \frac{\nabla \Pi^T}{|\nabla \Pi|} \circ (\boldsymbol{x} = \boldsymbol{x}_{Current}), \\ \boldsymbol{q}_{Next} &= 0.98 \boldsymbol{q}_{Current} - \alpha \boldsymbol{r}_{Current}, \\ \boldsymbol{x}_{Next} &= \boldsymbol{x}_{Current} + \alpha \boldsymbol{q}_{Next}, \end{aligned} \tag{5}$$

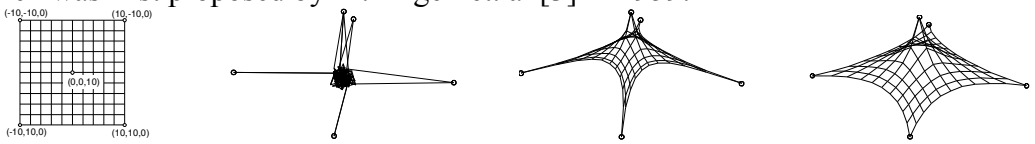
which gives the simplest **3-term method**.

Fig. 1(a) shows a numerical example for verification which can be solved by either **2-term method** or **3-term method**. The model consists of 220 cables and 5 fixed nodes. The first author has tested giving random numbers for initial configuration of $\{x_1 \cdots x_n\}$, which were ranging from -2.5 to 2.5 as shown in Fig. 1(b). Then, when $\{w_1 \cdots w_m\} = \{1 \cdots 1\}$, Fig.1 (c) was obtained and the corresponding minimum value of Π was 160.214. When 4 times greater weight coefficients were assigned onto the boundary cables, Fig. 1 (d) was obtained and the corresponding minimum value was 188.09.

The history of Π in the **3-term method** corresponding to Fig. 1 (c) is shown by Fig. 2 (α

was constantly set as 0.2). When α was gradually decreased manually, $|\nabla\Pi|$ also decreased gradually as shown in Fig.3.

It is important to note that Eq. (5) has a close relation with the **family methods with three term recursion formulae**. In 1982, M. Papadrakakis stated that two popular direc minimization methods, the Dynamic Relaxation method and the Conjugate Gradient Method, can be classified under one category, the family methods with three term recursion formulae [4], which was first proposed by M. Engeli et. al [3] in 1959.



(a) Connections (b) Initial Step (c) $\sum L^2 \rightarrow \min$ (d) $\sum wL^2 \rightarrow \min$

Figure 1: Form-finding of Cable-net Structure

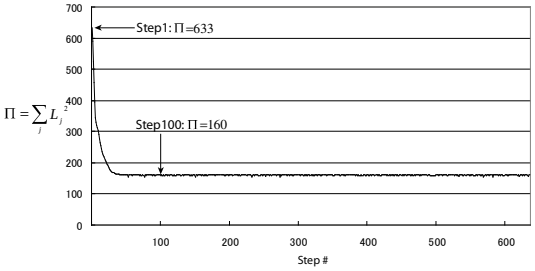


Figure 2: History of Π (α=0.2) (by 3-term method)

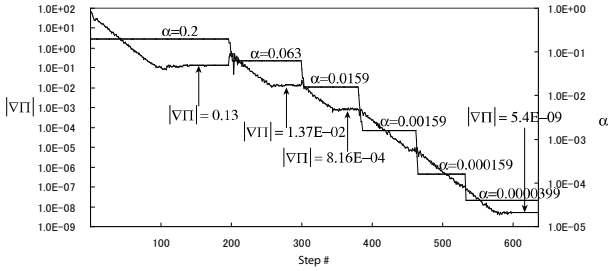


Figure 3: History of $|\nabla\Pi|$ (by 3-term method)

By using either **2-term method** or **3-term method**, both Fig. 1(c) and (d) can be obtained. However, the behavior of the **3-term method** is always very impressive, smooth, and seems to be the best method for solving various types of static problems by direct minimization approaches. Therefore, the scope of the **3-term method** must be examined and every numerical example illustrated below was solved by the **3-term method**.

2.2 Three Term Method with Constraint Conditions

(a) Connections of members (b) Result ($\sum L_j^4 = 18000$)

Figure 4: Form-Finding of Simplex Tensegrity

For form-finding of *Simplex Tensegrities* that consist of 3 struts (compression) and 9 cables (tension), let us discuss

$$\Pi(\boldsymbol{x},\boldsymbol{\lambda}) = \sum_{j=1}^{m} w_j L_j^{\,4}(\boldsymbol{x}) + \sum_{k=1}^{r} \lambda_k (L_{m+k}(\boldsymbol{x}) - \overline{L}_{m+k}) \rightarrow \text{stationary}, \tag{6}$$

where the first sum is taken for all the length of the cables and the second sum is taken for all

the length of the struts. The connections over the members are shown by Fig. 4(a). Such a modified functional is obtained by applying the *Lagrange* multiplier method to a minimization problem with equality constraint conditions. The supplemented multipliers are denoted by λ_k and the prescribed length of the struts are denoted by $\bar{L}_{m+k}$. When $\{w_1,\cdots,w_9,\bar{L}_{10},\cdots,\bar{L}_{12}\}=\{1,\cdots,1,10,\cdots,10\}$, Fig. 4 (b) was obtained as the unique solution and the corresponding minimum value of $\sum_{j=1}^{m} w_j {L_j}^4(\boldsymbol{x})$ was 18000. (See Ref. [1] for further detail)

To solve such a problems, let

$$\boldsymbol{x}\equiv\begin{bmatrix}x_1 & \cdots & x_n\end{bmatrix}^T,\ \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}\equiv\begin{bmatrix}\frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n}\end{bmatrix},\ \boldsymbol{\lambda}\equiv\begin{bmatrix}\lambda_1 & \cdots & \lambda_r\end{bmatrix},\ \frac{\partial f(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}}\equiv\begin{bmatrix}\frac{\partial f}{\partial \lambda_1} & \cdots & \frac{\partial f}{\partial \lambda_r}\end{bmatrix}^T, \text{ and} \tag{7}$$

$$\nabla f\equiv\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}. \tag{8}$$

Then, the stationary condition of Eq. (6) is represented by

$$\nabla\Pi=\boldsymbol{0}\ \&\ \frac{\partial\Pi}{\partial\boldsymbol{\lambda}}=\boldsymbol{0}\ . \tag{9}$$

Let us discuss the first condition. Due to the supplemented multipliers $\{\lambda_1\cdots\lambda_r\}$, $\nabla\Pi$ is still unknown even if $\boldsymbol{x}$ is given. The simplest idea to determine $\nabla\Pi$ uniquely is making use of general inverse matrix. Let us rewrite $\nabla\Pi=\boldsymbol{0}$ into the following form:

$$\nabla\Pi=\nabla\Pi_w(\boldsymbol{x})+\boldsymbol{\lambda}\boldsymbol{J}_\lambda=\boldsymbol{0}\ \Leftrightarrow\ \boldsymbol{\lambda}\boldsymbol{J}_\lambda=-\nabla\Pi_w, \tag{10}$$

For example, in this case,

$$\Pi_w(\boldsymbol{x})=\sum_j w_j {L_j}^4(\boldsymbol{x}),\ \nabla\Pi_w=\sum_j 4w_j {L_j}^3\nabla L_j,\ \text{and}\ \boldsymbol{J}_\lambda=\begin{bmatrix}\nabla L_{m+1}\\ \vdots\\ \nabla L_{m+r}\end{bmatrix}. \tag{11}$$

Eq. (10) is not satisfied unless a solution is given as $\boldsymbol{x}$. Nevertheless, by using *Moore-Penrose* type pseudo inverse matrix $\boldsymbol{J}_\lambda^+$, $\boldsymbol{\lambda}$ can be determined by

$$\boldsymbol{\lambda}=-\nabla\Pi_w(\boldsymbol{x})\cdot\boldsymbol{J}_\lambda^+, \tag{12}$$

which provides basically a least squared solution of Eq. (10). This strategy seems rather rough but it works fine because when Eq. (12) turns to a least norm solution, Eq. (10) is simultaneously satisfied and vice versa. A popular numerical environments such as Matlab®, Scilab®, and Octave® provide pinv() for this purpose. A popular linear algebra package Lapack® provides direct least squared solvers such as dgels().

As the result, 1 to 1 mapping between $\boldsymbol{x}$ and $\nabla\Pi$ can be defined by

$$\nabla\Pi=\nabla\Pi_w(\boldsymbol{x})+\boldsymbol{\lambda}\boldsymbol{J}_\lambda\circ(\boldsymbol{\lambda}=-\nabla\Pi_w(\boldsymbol{x})\cdot\boldsymbol{J}_\lambda^+), \tag{13}$$

which enables **2-term method** and **3-term method** feasible. This strategy is often called the **dual estimate** in the context of linear programming [7]. Here, one might notice that Eq. (13)

immediately reduces to the **projected gradient**, in which $\boldsymbol{\lambda}$ vanishes; however, $\boldsymbol{\lambda}$ is always calculated explicitly in this work because Eq. (13) simply represents the **composition of forces** (see Fig. 5 (a)).

Let us discuss the second condition in Eq. (9). For example, in this case, it expands as

$$\frac{\partial \Pi}{\partial \boldsymbol{\lambda}} = \boldsymbol{0} \Leftrightarrow \begin{cases} (L_{m+1}(\boldsymbol{x}) - \overline{L}_{m+1}) = 0, \\ \vdots \\ (L_{m+r}(\boldsymbol{x}) - \overline{L}_{m+r}) = 0. \end{cases} \tag{14}$$

The simplest idea to satisfy Eq. (14) is to solve

$$\boldsymbol{J}_\lambda \cdot \Delta \boldsymbol{x} = -\boldsymbol{r} \text{ , where } \boldsymbol{J}_\lambda = \begin{bmatrix} \nabla L_{m+1} \\ \vdots \\ \nabla L_{m+r} \end{bmatrix}, \boldsymbol{r} = \begin{bmatrix} L_{m+1} - \overline{L}_{m+1} \\ \vdots \\ L_{m+r} - \overline{L}_{m+r} \end{bmatrix}. \tag{15}$$

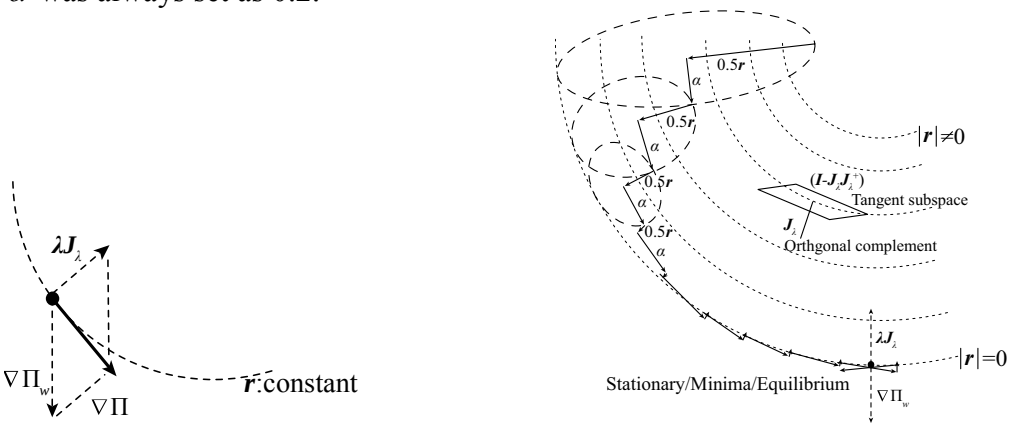
Again, $\boldsymbol{J}_\lambda^{+}$ plays an important role as follows

$$\Delta \boldsymbol{x} = -\boldsymbol{J}_\lambda^{+} \cdot \boldsymbol{r} \text{ ,} \tag{16}$$

which gives the least norm solution of Eq. (15). It is highly recommended to scale $\Delta \boldsymbol{x}$ when it is substituted into $\boldsymbol{x}_{Current}$, e.g.

$$\boldsymbol{x}_{Current} := \boldsymbol{x}_{Current} + 0.5\Delta \boldsymbol{x} \text{ .} \tag{17}$$

Fig. 5 (b) shows an over-view of above mentioned strategies. In each step, if Eq. (17) is performed once just before Eq. (4) or Eq. (5) is performed, $\boldsymbol{x}_{Current}$ would gradually approaches to the hyper-surface on which the prescribed constraint conditions are satisfied. By using either **2-term** or **3-term method**, Fig.4 (b) can be obtained. It was even giving random numbers for the initial configuration of $\{x_1 \ \cdots \ x_n\}$ as well as the previous example. In the analysis α was always set as 0.2.



(a) Composition of forces (b) Two types of search directions

Figure 5: Overview of Minimization under Constraint Conditions

Fig. 6 shows another example, which illustrates a form-finding analysis of a tension structure that consists of cables, membranes, struts and fixed points. In this analysis, the

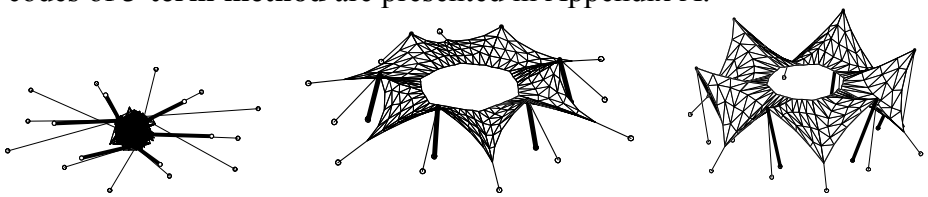
following stationary problem was solved by abovementioned strategies:

$$\Pi(\boldsymbol{x},\boldsymbol{\lambda}) = \sum_j w_j {L_j}^4(\boldsymbol{x}) + \sum_k w_k {S_k}^2(\boldsymbol{x}) + \sum_l \lambda_l (L_l(\boldsymbol{x}) - \bar{L}_l) \to \text{stationary}\,, \tag{18}$$

where the first sum is taken for all the line elements, the second is for all the triangle elements, and the third is for all the struts. In addition, S_k is a function to give the area of *k*-th triangle element. Fig. 6 (a) shows the initial configuration of $\boldsymbol{x}$ and it was given by random numbers as well as the previous examples.

It is important to note that Fig.6 (b) and (c) were obtained by only **3-term method** and it was almost impossible to solve the same problem by **2-term method**. It is because of that **2-term method** always traces "bumpy" objective function precisely whereas **3-term method** does roughly. By using the **3-term method**, unless the process terminated, the process can continue forever and the form can be varied at any moment by varying $w_j, w_k, \bar{L}_l$.

Pseudo codes of **3-term method** are presented in Appendix A.



(a) Initial Step (b) Variation 1 (c) Final decision

Figure 6: Form-Finding of Tanzbrunnen Koln (F. Otto, 1959)

3. PRINCIPLE OF VIRTUAL WORK

3.1 Principle of Virtual Work for General Membranes

In the following discussion, *Einstein* summation convention and standard notations of tensor algebra are used. Let us discuss the surface area of a surface, which is given by

$$a = \int_a \mathrm{da}\ ,\ \mathrm{da} \equiv \sqrt{\det g_{ij}}\, d\theta^1 d\theta^2\ (1 \le i, j \le 2)\,, \tag{19}$$

where $g_{ij}, \theta^1, \theta^2$ represents the *Riemannian* metric and the local coordinates defined on the surface respectively.

Using $\delta\sqrt{\det g_{ij}} = 1/2\, g^{ij}\delta g_{ij}\sqrt{\det g_{ij}}$ where $g^{ij} \equiv \left(g_{ij}\right)^{-1}$, the minimal surface problem is expressed as

$$\delta a = \frac{1}{2}\int_a g^{ij}\delta g_{ij}\,\mathrm{da} = 0\,. \tag{20}$$

By the way, let us discuss a self-equilibrium membrane whose boundary is fixed. The **principle of virtual work** for such a membrane is expressed as

$$\delta w = \frac{1}{2}\int_a t\sigma^{ij}\delta g_{ij}\,\mathrm{da} = 0\quad (1 \le i, j \le 2)\,, \tag{21}$$

where t, σ^{ij} represents the thickness and the Cauchy stress tensor respectively. Here, δg_{ij} is

used instead of the variation of strain tensor δe_{ij}, due to the essential identity of them.

Using $\sigma^{ij} = \sigma^{i}{}_{\cdot k} g^{kj}$, the **principle of virtual work** for such a membrane is rewritten as

$$\delta w = \frac{1}{2}\int_a t\sigma^{i}{}_{\cdot k} g^{kj} \delta g_{ij} \mathrm{da} = 0 \quad (1 \le i, j, k \le 2) . \tag{22}$$

When a new stress tensor T^{i}_{k} is defined by $T^{i}{}_{\cdot k} \equiv t\sigma^{i}{}_{\cdot k}$, we have

$$\delta w = \frac{1}{2}\int_a T^{i}{}_{\cdot k} g^{kj} \delta g_{ij} \mathrm{da} = 0 \quad (1 \le i, j, k \le 2) . \tag{23}$$

On the other hand, Eq. (20) can be rewritten as

$$\delta a = \frac{1}{2}\int_a \delta^{i}{}_{\cdot k} g^{kj} \delta g_{ij} \mathrm{da} = 0 \quad (1 \le i, j, k \le 2) , \tag{24}$$

which can be a simple demonstration of the essential identity of minimal surface and uniform stress surface.

3.2 Principle of Virtual Work for N-Dimensional Riemannian Manifolds

The length of a curve, the area of a surface, and the volume of a body are respectively given by:

$$l = \int_l \mathrm{dl} \ , \ a = \int_a \mathrm{da} \ , \ v = \int_v \mathrm{dv} \ , \text{ where} \tag{25}$$

$$\mathrm{dl} \equiv \sqrt{\det g_{ij}}\, d\theta^1 (i, j = 1) \ , \ \mathrm{da} \equiv \sqrt{\det g_{ij}}\, d\theta^1 d\theta^2 (1 \le i, j \le 2) \ , \ \mathrm{dv} \equiv \sqrt{\det g_{ij}}\, d\theta^1 d\theta^2 d\theta^3 (1 \le i, j \le 3) \tag{26}$$

By using the concept of *N*-dimensional *Riemannian* manifold, they can be unified. The volume element, the volume, and the variation of the volume for an *N*-dimensional *Riemannian* manifold *M* are respectively given by

$$\mathrm{dv}^N \equiv \sqrt{\det g_{ij}}\, d\theta^1 \cdots d\theta^N \ , \ v^N \equiv \int_M \mathrm{dv}^N \ , \text{ and } \delta v^N = \frac{1}{2}\int_M g^{ij} \delta g_{ij} \mathrm{dv}^N \quad (1 \le i, j \le N) . \tag{27}$$

Then, the variational problem of the volume of *M* is defined by

$$\delta v^N = \frac{1}{2}\int_M g^{ij} \delta g_{ij} \mathrm{dv}^N = 0 \Leftrightarrow \frac{1}{2}\int_M \delta^{i}_{\cdot k} g^{kj} \delta g_{ij} \mathrm{dv}^N = 0 \quad (1 \le i, j, k \le N) . \tag{28}$$

By the way, the **principle of virtual works** for self-equilibrium cables, membranes, deformable bodies are expressed as follows:

$$\delta w^1 = \frac{1}{2}\int_l A\sigma^{i}{}_{\cdot k} g^{kj} \delta g_{ij} \mathrm{dl} = 0 \quad (i, j, k = 1) , \tag{29}$$

$$\delta w^2 = \frac{1}{2}\int_a t\sigma^{i}{}_{\cdot k} g^{kj} \delta g_{ij} \mathrm{da} = 0 \quad (1 \le i, j, k \le 2) , \tag{30}$$

$$\delta w^3 = \frac{1}{2}\int_v \sigma^{i}{}_{\cdot k} g^{kj} \delta g_{ij} \mathrm{dv} = 0 \quad (1 \le i, j, k \le 3) , \tag{31}$$

where A,t respectively denote the sectional area of the cable and the thickness of the membrane. Then, when a new stress tensor $T^{i}{}_{\cdot k}$ is defined for each dimension separately by

$$T^{i}{}_{\cdot k} \equiv A\sigma^{i}{}_{\cdot k}\,(N=1)\,,\ T^{i}{}_{\cdot k} \equiv t\sigma^{i}{}_{\cdot k}\,(N=2)\,,\ \text{and}\ T^{i}{}_{\cdot k} \equiv \sigma^{i}{}_{\cdot k}\,(N=3)\,, \tag{32}$$

a common form of Eq. (29), (30), (31) is found and it is expressed as

$$\delta w^{N} = \frac{1}{2}\int_{M} T^{i}{}_{\cdot k}\, g^{kj}\delta g_{ij}\mathrm{d}v^{N} = 0\ (1 \le i, j, k \le N)\,, \tag{33}$$

which is the **principle of virtual work** for *N*-dimensional *Riemannian* manifold *M*. This can be naturally read that $T^{ij} = T^{i}{}_{\cdot k}\, g^{kj}$ is a general force which act within *M* and tend to produce small change of g_{ij} [7]. Additionally, Eq. (28) is a special case of Eq. (33) such that $T^{i}{}_{\cdot k} = \delta^{i}{}_{\cdot k}$.

3.3 Galerkin Method

Suppose a form of a curve, a surface, or a body is represented by *n* independent parameters such as $\boldsymbol{x} = [x_1 \ \cdots \ x_n]^T$. Let us define gradient vector of a function *f* with respect to $\boldsymbol{x}$ by

$$\nabla f \equiv \left[\frac{\partial f}{\partial x_1} \ \cdots \ \frac{\partial f}{\partial x_n}\right]. \tag{34}$$

In such a case, at most *n* independent δg_{ij} can satisfy Eq. (33). Such a set of δg_{ij} is provided by weighted residual method family. The Galerkin method is one of them. Using the Galerkin method, δg_{ij} is altered into

$$\delta\tilde{g}_{ij} \equiv \nabla g_{ij} \cdot \delta\boldsymbol{x}\,. \tag{35}$$

where $\delta\boldsymbol{x}$ is the variation of $\boldsymbol{x}$, or just an arbitrary column vector, namely $\delta\boldsymbol{x} \equiv [\delta x^1 \ \cdots \ \delta x^n]^T$.

Then, the **discrete stationary condition** is deduced as follows:

$$\delta\tilde{w}^{N} = \frac{1}{2}\int_{M}\left(T^{i}{}_{\cdot k}\, g^{kj}\delta\tilde{g}_{ij}\right)\mathrm{d}v^{N} = 0 \Leftrightarrow \delta w^{N} = \left(\frac{1}{2}\int_{M} T^{i}{}_{\cdot k}\, g^{kj}\nabla g_{ij}\mathrm{d}v^{N}\right)\cdot\delta\boldsymbol{x} = 0 \tag{36}$$

$$\Leftrightarrow \boldsymbol{\omega} = \frac{1}{2}\int_{M} T^{i}{}_{\cdot k}\, g^{kj}\nabla g_{ij}\mathrm{d}v^{N} = \boldsymbol{0}\,. \tag{37}$$

For general problems of statics, the following can be used:

$$\left(\frac{1}{2}\int_{M} T^{i}{}_{\cdot k}\, g^{kj}\nabla g_{ij}\mathrm{d}v^{N}\right)\cdot\delta\boldsymbol{x} = \boldsymbol{p}\cdot\delta\boldsymbol{x} \Leftrightarrow \boldsymbol{\omega} = \frac{1}{2}\int_{M} T^{i}{}_{\cdot k}\, g^{kj}\nabla g_{ij}\mathrm{d}v^{N} - \boldsymbol{p} = \boldsymbol{0}\,, \tag{38}$$

where $\boldsymbol{p}$ represents the nodal forces.

3.4 N-dimensional Simplex Elements

When the form of a structure is represented by *m* independent elements such as line elements, triangle elements, and tetrahedron elements, Eq. (37) expands to

$$\boldsymbol{\omega} = \sum_j \frac{1}{2}\int_j T^{\alpha}{}_{\cdot\gamma} g^{\gamma\beta} \nabla g_{\alpha\beta} \mathrm{dv}^N = \boldsymbol{0}\ , \tag{39}$$

where the sum is taken for all the elements and integrals are calculated separately within each element. Let us define

$$\boldsymbol{\omega}_j{}^N (T^{\alpha}{}_{\cdot\gamma}) \equiv \frac{1}{2}\int_j T^{\alpha}{}_{\cdot\gamma} g^{\gamma\beta} \nabla g_{\alpha\beta} \mathrm{dv}^N\ , \tag{40}$$

so that the **discrete stationary condition** can be simply represented by

$$\boldsymbol{\omega} = \sum_j \boldsymbol{\omega}_j{}^N = \boldsymbol{0} \ \text{ or } \ \boldsymbol{\omega} = \sum_j \boldsymbol{\omega}_j{}^N - \boldsymbol{p} = \boldsymbol{0}\ . \tag{41}$$

The simplest idea to calculate Eq. (40) is making use *N*-dimensional Simplex elements which are shown by Fig. 7. Within each dimensional element, the base vectors $\boldsymbol{g}_1 \cdots \boldsymbol{g}_N$ are constant and they are given by

$$\boldsymbol{g}_i = \boldsymbol{p}_i - \boldsymbol{p}_{i+1} \quad (1 \le i \le N)\ , \tag{42}$$

where $\boldsymbol{p}_i$ represents the global Cartesian coordinate of *i*-th node. Then,

$$g_{ij} = \boldsymbol{g}_i \cdot \boldsymbol{g}_j\ . \tag{43}$$

In almost cases, $T^{i}{}_{\cdot k}$ is given by a function of *Riemannian* metric, then $T^{i}{}_{\cdot k}$ is also constant within each element. Therefore,

$$\boldsymbol{\omega}_j{}^1 (T^{\alpha}{}_{\cdot\gamma}) = \frac{1}{2} L_j T^{1}{}_{\cdot 1} g^{11} \nabla g_{11}\Big|_j\ ,\ \boldsymbol{\omega}_j{}^2 \left(T^{\alpha}{}_{\cdot\gamma}\right) = \frac{1}{2} S_j \left(\begin{bmatrix} T^{1}{}_{\cdot 1} & T^{1}{}_{\cdot 2} \\ T^{2}{}_{\cdot 1} & T^{2}{}_{\cdot 2} \end{bmatrix} \begin{bmatrix} g^{11} & g^{12} \\ g^{21} & g^{22} \end{bmatrix} : \begin{bmatrix} \nabla g_{11} & \nabla g_{12} \\ \nabla g_{21} & \nabla g_{22} \end{bmatrix} \right)_j ,$$

$$\boldsymbol{\omega}_j{}^3 \left(T^{\alpha}{}_{\cdot\gamma}\right) = \frac{1}{2} V_j \left(\begin{bmatrix} T^{1}{}_{\cdot 1} & T^{1}{}_{\cdot 2} & T^{1}{}_{\cdot 3} \\ T^{2}{}_{\cdot 1} & T^{2}{}_{\cdot 2} & T^{2}{}_{\cdot 3} \\ T^{3}{}_{\cdot 1} & T^{3}{}_{\cdot 2} & T^{3}{}_{\cdot 3} \end{bmatrix} \begin{bmatrix} g^{11} & g^{12} & g^{13} \\ g^{21} & g^{22} & g^{23} \\ g^{31} & g^{32} & g^{33} \end{bmatrix} : \begin{bmatrix} \nabla g_{11} & \nabla g_{12} & \nabla g_{13} \\ \nabla g_{21} & \nabla g_{22} & \nabla g_{23} \\ \nabla g_{31} & \nabla g_{32} & \nabla g_{33} \end{bmatrix} \right)_j , \tag{44}$$

where ':' symbol is defined by $a^{ij} : b_{\alpha\beta} \equiv a^{ij} b_{ij}$. In addition, L_j, S_j, V_j denote the length, the area, and the volume of each element, namely

$$L_j = \sqrt{\det g_{\alpha\beta}}\Big|_j\ ,\ S_j = \frac{1}{2}\sqrt{\det g_{\alpha\beta}}\Big|_j\ ,\ V_j = \frac{1}{6}\sqrt{\det g_{\alpha\beta}}\Big|_j\ . \tag{45}$$

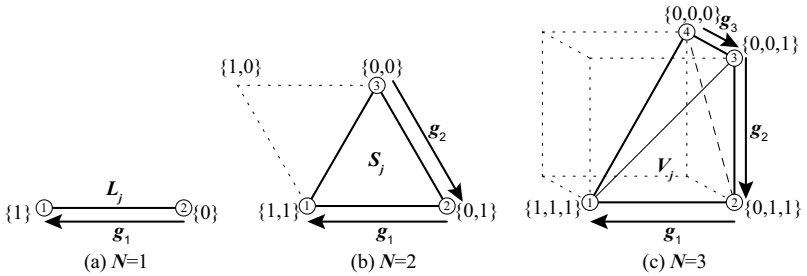


Figure 7: Simplex Elements

To conclude this section, it is important to note that

$$\boldsymbol{\omega}_j^{\,1}(\delta^{\alpha}_{\cdot\gamma}) = \nabla L_j\,,\ \boldsymbol{\omega}_j^{\,2}(\delta^{\alpha}_{\cdot\gamma}) = \nabla S_j\,,\ \text{and}\ \boldsymbol{\omega}_j^{\,3}(\delta^{\alpha}_{\cdot\gamma}) = \nabla V_j\,. \tag{46}$$

Therefore, the examples illustrated in chapter 2 are the special and simple cases of Eq. (41) such that each $T^{i}{}_{\cdot k}$ is given as just a scalar multiple of $\delta^{i}{}_{\cdot k}$. It seems obvious that Such simple cases can be solved by direct minimization methods. Conversely, if such special cases can be solved by direct minimization methods, it may be also possible that general cases such that $T^{i}{}_{\cdot k}$ are not a scalar multiple of $\delta^{i}{}_{\cdot k}$ can be solved by direct minimization methods. Since $\boldsymbol{\omega}$ is just a mixture of gradient vectors, Eq. (4) or Eq. (5) may still feasible when $\nabla\Pi$ is altered into $\boldsymbol{\omega}$.

5 ST. VENANT BODY

To solve the **principle of virtual works** by direct minimization approaches, an explicit expression of $T^{i}_{\cdot k}$ that provides 1 to 1 mapping between g_{ij} and $T^{i}_{\cdot k}$ must be prescribed, which is called constitutive law. Let us examine

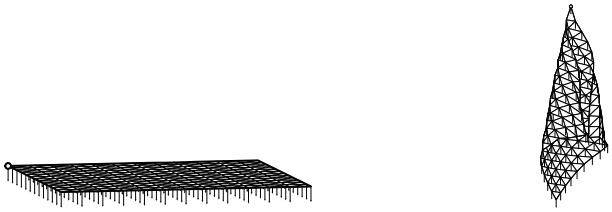
$$T^{i}_{\cdot k} = Eg^{il}(g_{lk} - \overline{g}_{lk})\,, \tag{47}$$

which gives a simplest one (St. Venant body when Poisson ratio is 0). Here, note that $\overline{g}_{lk}$ is the *Riemannian* metric on the undeformed state which is measured in advance. In the analyses illustrated below, unlike the above discussed examples, the initial configurations of $\{x_1 \ \cdots \ x_n\}$ were given by just the undeformed state and were not given by random numbers.

Fig. 8 and 9 show two natural forms of handkerchief under gravity whose dimensions are 8.0-8.0. For both results,

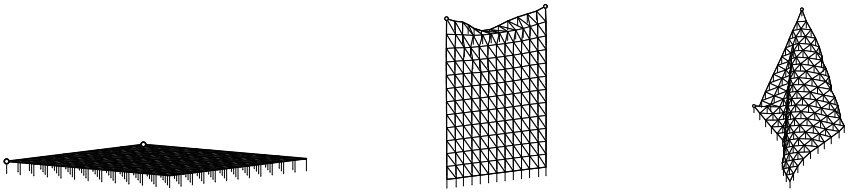
$$\boldsymbol{\omega} = \sum_j \boldsymbol{\omega}_j^{\,2}\left(T^{i}_{\cdot k}\right) - \boldsymbol{p} = \boldsymbol{0} \tag{49}$$

was solved by the **3-term method**, in which $\nabla\Pi$ is altered into $\boldsymbol{\omega}$. The prescribed $E, \boldsymbol{p}, \alpha$ for both results were 50, $[0 \ \ 0 \ \ -0.1 \ \ 0 \ \ 0 \ \ -0.1 \ \cdots \ 0 \ \ -0.1]$, and 0.2.



(a) Undeformed shape (b) Result

Figure 8: Natural Forms of Handkerchief under Gravity Hanged by 1-Point



(a) Undeformed shape (b) Result1 (c) Result2

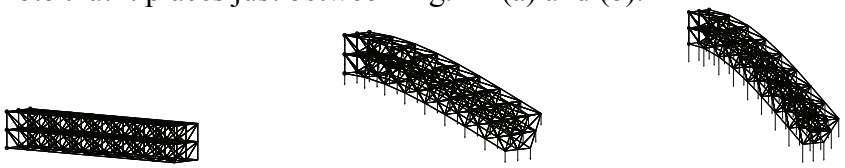
Figure 9: Natural Forms of Handkerchief under Gravity Hanged by 2-Points

Fig. 10 shows a large deformation analysis of a cantilever whose dimension is 2.0-2.0-12.0. Fig. 11 also shows a large deformation of the same body after bucking. For obtaining both the results,

$$\boldsymbol{\omega} = \sum_j \boldsymbol{\omega}_j^{\,3}\left(T_k^i\right) - \boldsymbol{p} = \boldsymbol{0} \tag{50}$$

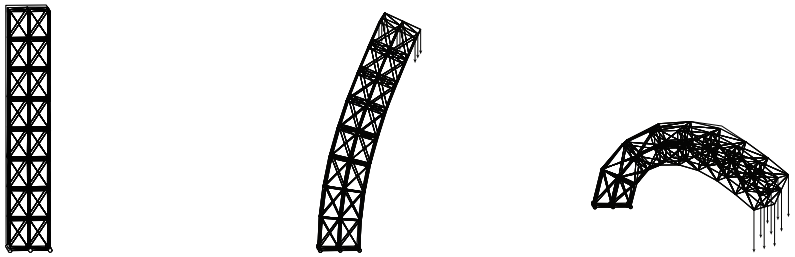
was solved by the **3-term method**. The prescribed E and α were 50 and 0.2. In Fig. 10, p represents the values which were set to the components of $\boldsymbol{p}$ that represents the z-components of all the nodal forces. In Fig 11, p denotes the values which were set to z-components of only 9 nodes and they place on top of the body.

For the body shown by Fig. 11, the Euler's buckling load is $p_{cr} = 1.14$, then its division by 9 is 0.126 and note that it places just between Fig. 11 (a) and (b).



(a) p=0.00 (b) p=0.05 (c) p=0.1

Figure 10: Large Deformation of Cantilever under Gravity



(a) p=0.10 (b) p=0.20 (c) p=1.0

Figure 11: Large Deformation after Buckling

6. CONCLUSIONS

2-term method and **3-term method** were described for not only simple minimization problems but also for minimization problems under constraint conditions. The **principle of virtual work** for *N*-dimensional *Riemannian* manifolds was also formulated. Finally, some **principle of virtual works** that can be solved by the **3-term method** were shwon, which imply the potential ability of the **3-term method** for solving various types of static problems.

7. ACKNOWLEDGEMENTS

This research was partially supported by the Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for JSPS Fellows, 10J09407, 2010

8. REFERENCES

[1] M. Miki, 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.3 (2010) 291-303.
[2] R. M. O. Pauletti, D. M. Guirardi, Direct area minimization through dynamic relaxation, Proceedings of the International Association for Shell and Spatial Strucutres (IASS) Symposium, (2010), 1222-1234.
[3] H.J. Schek, The force density method for form finding and computation of general networks, Computer Methods In applied Mechanics and Engineering. 3 (1974) 115–134.
[4] M. Engeli, T. Ginsburg, R. Rutishauser, E. Stiefel, Refined Iterative Methods for Computation of the Solution and the Eigenvalues of Self-Adjoint Boundary Value Problems, Basel/Stuttgart, Birkhauser Verlag, 1959.
[5] M. Papadrakakis, A family of methods with three-term recursion formulae, International Journal for Numerical Methods In Engineering. 18 (1982) 1785–1799.
[6] J. L. Lagrange (author), Analytical mechanics, A. C. Boissonnade and V. N. Vagliente (translator), Kluwer, (1997).
[7] I. I. Dikin, Iterative solution of problems of linear and quadratic programming, Soviet Mathematics Doklady. 8 (1967) 674-675

Appendix A. Pseudo Codes

(a) 3-term method

$\boldsymbol{x}_1 :=$ random numbers; $\boldsymbol{p}_1 := \boldsymbol{0}$; $\boldsymbol{q}_1 := \boldsymbol{0}$;
$Current := 1$; $Next := 2$;
while($true$){
  read $\alpha, \{w_1, \cdots, w_m\}$ from GUI;
  $\nabla\Pi := \sum_j 2w_j L_j(\boldsymbol{x}_{Current})\nabla L_j$;
  $\boldsymbol{p}_{Current} := \dfrac{\nabla\Pi}{|\nabla\Pi|}^T$;
  $\boldsymbol{q}_{Nexxt} := 0.98\boldsymbol{q}_{Current} - \alpha\boldsymbol{p}_{Current}$;
  $\boldsymbol{x}_{Next} := \boldsymbol{x}_{Current} + \alpha\boldsymbol{q}_{Next}$;
  $Current = Current + 1$; $Next = Next + 1$;
  visualize($\boldsymbol{x}_{Current}$);
}

(b) 3-term method under constraint conditions

$\boldsymbol{x}_1 :=$ random numbers; $\boldsymbol{p}_1 := \boldsymbol{0}$; $\boldsymbol{q}_1 := \boldsymbol{0}$;
$Current := 1$; $Next := 2$;
while($true$){
  read $\alpha, \{w_1, \cdots, w_m\}, \{\bar{L}_1, \cdots, \bar{L}_r\}$ from GUI;
  $\boldsymbol{J}_\lambda := \begin{bmatrix} \nabla L_{m+1} \\ \vdots \\ \nabla L_{m+r} \end{bmatrix}$; $\boldsymbol{r} := \begin{bmatrix} L_{m+1}(\boldsymbol{x}_{Current}) - \bar{L}_{m+1} \\ \vdots \\ L_{m+r}(\boldsymbol{x}_{Current}) - \bar{L}_{m+r} \end{bmatrix}$;
  $\Delta\boldsymbol{x} = -\boldsymbol{J}_\lambda^+ \cdot \boldsymbol{r}$;
  $\boldsymbol{x}_{Current} := \boldsymbol{x}_{Current} + 0.5\Delta\boldsymbol{x}$;
  $\nabla\Pi_w := \sum_{j=1}^{m} 4w_j L_j^3(\boldsymbol{x}_{Current})\nabla L_j$; $\boldsymbol{J}_\lambda := \begin{bmatrix} \nabla L_{m+1} \\ \vdots \\ \nabla L_{m+r} \end{bmatrix}$;
  $\boldsymbol{\lambda} := -\nabla\Pi_w \cdot \boldsymbol{J}_\lambda^+$;
  $\nabla\Pi := \nabla\Pi_w + \boldsymbol{\lambda}\boldsymbol{J}_\lambda$;
  $\boldsymbol{p}_{Current} := \dfrac{\nabla\Pi^T}{|\nabla\Pi|}$;
  $\boldsymbol{q}_{Nexxt} := 0.98\boldsymbol{q}_{Current} - \alpha\boldsymbol{p}_{Current}$;
  $\boldsymbol{x}_{Next} := \boldsymbol{x}_{Current} + \alpha\boldsymbol{q}_{Next}$;
  $Current = Current + 1$; $Next = Next + 1$;
  visualize($\boldsymbol{x}_{Current}$);
}