



Title	事例2 : 水中で振動する弾性体の挙動について
Author(s)	出口, 弘; 富田, 和雄; 森澤, 伸哉 他
Citation	大阪大学大型計算機センターニュース. 1995, 95, p. 20-25
Version Type	VoR
URL	https://hdl.handle.net/11094/66083
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

水中で振動する弾性体の挙動について

大阪大学大型計算機センター研究開発部

出口 弘

deguchi@center.osaka-u.ac.jp

大阪大学大学院工学研究科船舶海洋工学専攻

富田 和雄、森澤 伸哉、広野 義和

k_tomita@naoe.eng.osaka-u.ac.jp

1 はじめに

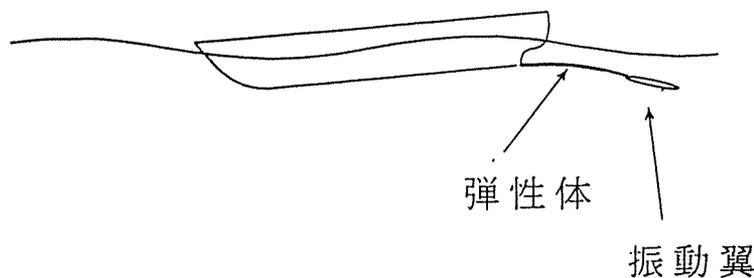


図1 振動翼推進

現在研究が進められている推進システムのなかに、Bio-Propulsion(生体推進システム)がある。これは、波浪エネルギーを推進力に変えるエネルギーの転換方法であり、船体に水中翼を取り付け、波浪中の船体動揺によって水中翼が振動し、推進力を発生するというものである。本研究では、図1のようにさらに船体と水中翼の間に弾性体を介したモデルを考えている。そして、船体の動揺により水中翼及び弾性体が動揺するようにし、その運動形状を魚類の遊泳運動に似たものにするにより、大きな推進力を得ようとするものである。そのためには、振動翼と弾性体の挙動を知ることが重要となる。今回この弾性体の挙動をベルヌーイ・オイラー梁の運動方程式を用いることにより求め、Explorerを使って可視化画像を作成した。そしてそれらの画像をVideoDesktopを用いてビデオを制御することにより動画像化した。

2 弾性板の振動

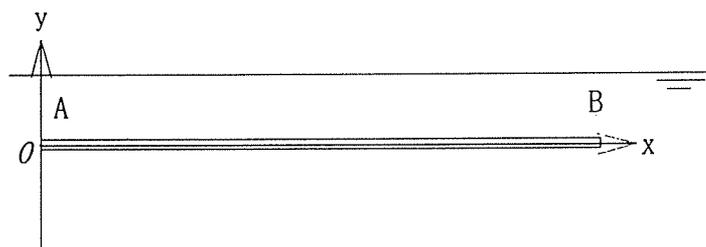


図2 座標系

図2のように水中でA点は固定端、B点は自由端の片持ち弾性板を、固定端A点で上下に強制動揺させた時の弾性板の挙動を調べる。その挙動は、次式のベルヌーイ・オイラー梁の運動方程式と呼ばれるもので与えられる。弾性板の任意点 x における y 方向の変位を $y(x,t)$ とすると、

$$\frac{\gamma_a A_a}{g} \frac{\partial^2 y(x,t)}{\partial t^4} + EI \frac{\partial^4 y(x,t)}{\partial t^4} = -\frac{1}{2} C_D \frac{\gamma_w d}{g} \frac{\partial y(x,t)}{\partial t} \left| \frac{\partial y(x,t)}{\partial t} \right| - C_M \frac{\gamma_w \pi d^2}{g} \frac{\partial^2 y(x,t)}{\partial t^2} \quad (2,1)$$

となる。ここに、 E は弾性板のヤング率 (kg/cm^2)、 I は弾性板の断面2次係数 (cm^4)、 γ_a は弾性板の単位体積重量 (kgf/cm^3)、 A_a は弾性板の断面積 (cm^2)、 d は弾性板の幅 (cm)、 γ_w は水の単位体積重量 (kgf/cm^3)、 g は重力 (cm/s^2)、 C_D は抗力係数、 C_M は付加質量係数である。この運動方程式を解いて関数 $y(x,t)$ を求め、弾性板の自重による初期たわみを重ね合わせれば、弾性板の挙動がわかる。

まず、(2,1)式の運動方程式を $y(x,t)$ について解くと次式となる。

$$y(x,t) = \eta \sin \omega_0 t + \sum_{j=1}^{\infty} \left[\frac{\cos \alpha_j x - \cosh \alpha_j x}{\cos \alpha_j l + \cosh \alpha_j l} - \frac{\sin \alpha_j x - \sinh \alpha_j x}{\sin \alpha_j l + \sinh \alpha_j l} \right] \left\{ \frac{-\beta_j \eta c' \omega_0 \omega_j^2}{(\omega_j^2 - \omega_0^2)^2 + (c' \omega_0)^2} \cos \omega_0 t + \frac{\beta_j \eta \omega_0^2 (\omega_j^2 - \omega_0^2 - c'^2)}{(\omega_j^2 - \omega_0^2)^2 + (c' \omega_0)^2} \sin \omega_0 t \right\} \quad (2,2)$$

ただし

$$\begin{aligned} j &= 1, 2, 3 \dots \\ \omega_j^2 &= \frac{4EIg}{(4\gamma_a A_a + C_M \gamma_w \pi d^2)} \\ c' &= -\frac{8}{3\pi} U_0 \frac{2C_D \gamma_w D}{(4\gamma_a A_a + \gamma_w \pi d^2)} \\ \beta_j &= \frac{\int_0^l X_j(x) dx}{\int_0^l X_j^2(x) dx} \\ X_j(x) &= \frac{\cos \alpha_j x - \cosh \alpha_j x}{\cos \alpha_j l + \cosh \alpha_j l} - \frac{\sin \alpha_j x - \sinh \alpha_j x}{\sin \alpha_j l + \sinh \alpha_j l} \end{aligned} \quad (2,3)$$

ここに、 η は固定端Aの上下強制動揺の振幅、 ω_0 は固定端A点の上下強制動揺の角速度、 α_j は片持ち梁の振動数方程式より決定される値、 U_0 は弾性板の自由端B点での速度である。

次に、弾性板の自重による初期たわみを表す式 $y_0(x)$ は次式となる。

$$y_0(x) = -\frac{(\gamma_a - \gamma_w) A_a}{EIg} \{ (l-x)^4 - 4l^3(l-x) + 3l^4 \} \quad (2,4)$$

そして、固定端A点で上下に強制動揺させた時の弾性板の挙動表す関数 $Y(x,t)$ は、(2,2)式と(2,4)式をたしあわせた、

$$Y(x,t) = y(x,t) + y_0(x) \quad (2,5)$$

となる。

これらの式から求めた値を可視化した図を次ページに示す。

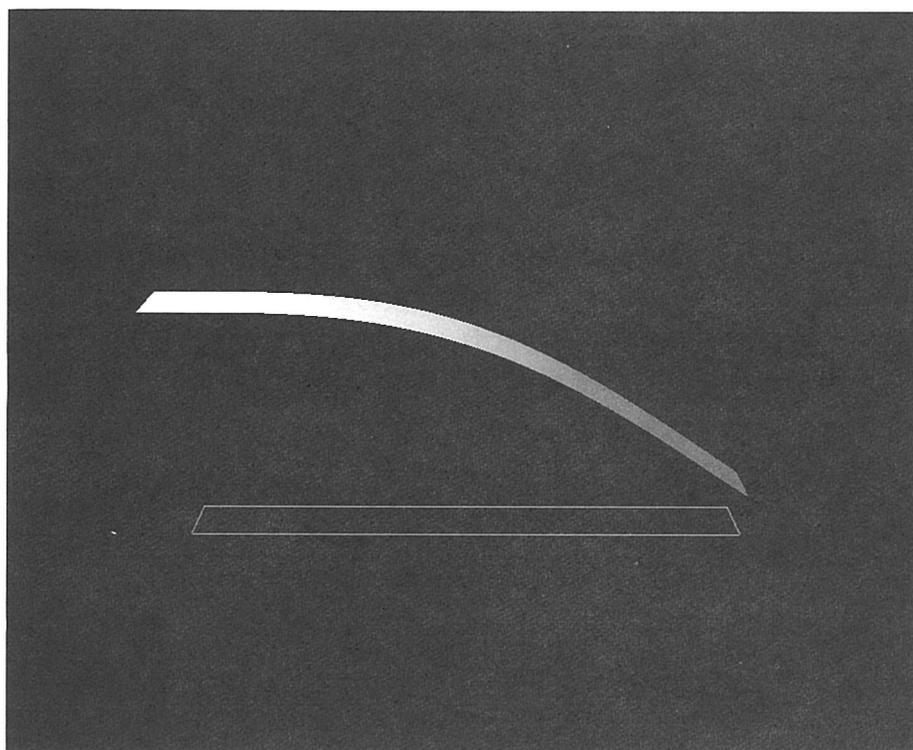


図 2 水中で振動する弾性体挙動の可視化

3 入力モジュールの説明

可視化に用いたデータは、以下のフォーマットでアスキーコードの文字列として表現されている。

```
40      x 軸方向のデータの数
4       y 軸方向のデータの数
5.431522 弾性板の変位のデータ
5.178239
5.030367
4.667614
4.350275
4.020709
(以下略)
```

データは、長方形を 40×4 のセルに等分割して表現しているので、Explorer では 2 次元

定型格子データとして扱うことができる。データの並びは、最初に各軸方向のデータの個数、その続きに、対応する格子点での弾性板の変位のデータが並んでいる。

弾性板の変位データは、格子点での基準面からの弾性板の距離を表している。下図はデータを読み込み、Explorer 型データタイプとして出力する read2d モジュールを Data Scribe で作成したところである。

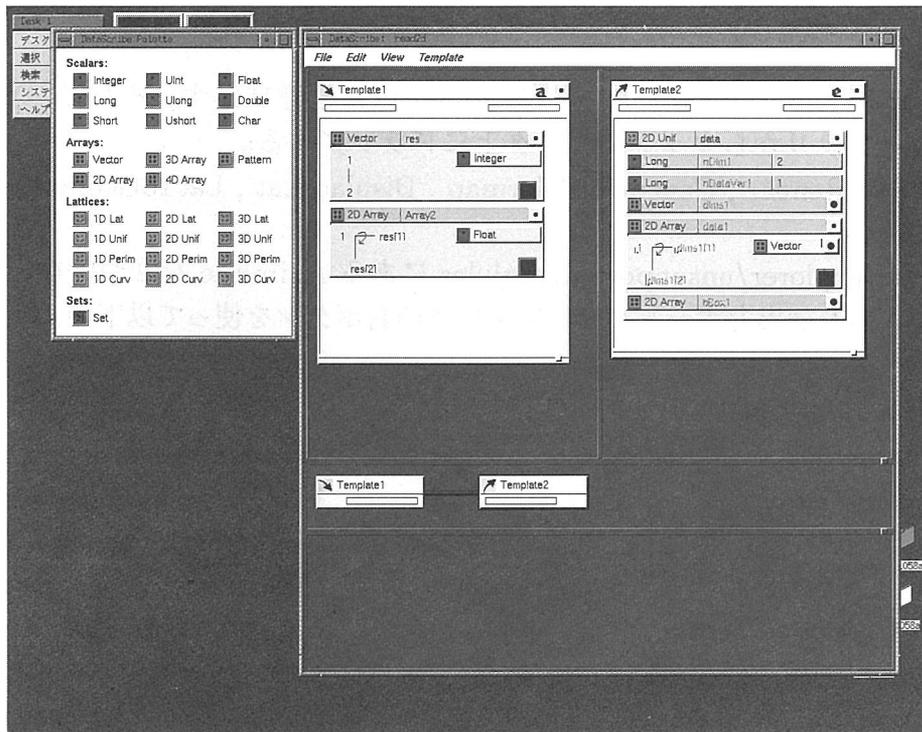


図 3 入力モジュール read2d

3.1 入力テンプレート

Template メニューで、入力テンプレートを作成したのち、以下のように編集する。

- (1) まず、データの個数を読み込むために、Vector をドラッグする。そして、名前を res に変更し、そのインデックスをデフォルトの 1-N から 1-2 にする。
- (2) 各格子点でのデータを読み込むために、パレットから 2D Array をドラッグする。
- (3) 2D Array のインデックスをそれぞれ res[1]、res[2] に変更する。(図 3 参照) そして、その要素を Float に変更する。

3.2 出力テンプレート

出力テンプレートとして 2D Unif をドラッグし、名前を data とする。また、nCoord の値に 1 を入力する。

3.3 接続と保存

入力テンプレートの `res` と出力テンプレートの `dims1` を接続する。同様に入力テンプレートの `Array2` と `Template2` の `data1` を `2` を接続する。そして、できたモジュールを `read2d` として保存する。

4 マップの説明

まず、Data Scribe で作成した `read2d` モジュールを呼び出す。そして、次に述べるモジュールライブラリからマップエディタ上にドラッグする。

`WireFrame` , `Render` , `Generate Colormap` , `DisplaceLat` , `LatToGeom`
`WriteImg` , `For`

また、`/usr/explorer/unsupported/mojules` にある `AnimFile` を 2 つ呼び出す。

これらの入力及び出力ポートパッドをマウスの右ボタンを使って以下のように接続する。

モジュール名/出力ポートパッド	モジュール名/入力ポートパッド
<i>AnimFile/out_file</i>	<i>read2d/Template1</i>
<i>read2d/data-Lattice</i>	<i>WireFlame/Input-Lattice</i>
	<i>DispleceLat/Function-Lattice</i>
	<i>DispleceLat/Displacement-Lattice</i>
<i>DispleceLat/Output-Lattice</i>	<i>LatToGeom/Input-Lattice</i>
<i>GenerateColormap/Colormap-Lattice</i>	<i>LatToGeom/Colormap-Lattice</i>
<i>LatToGeom/Output-Geometory</i>	<i>Render/Input-Geometory</i>
<i>Render/Sync-parameter</i>	<i>Render/snap</i>
<i>Render/Snapshot-Lattice</i>	<i>WriteImg/Input-Lattice</i>
<i>AnimFile<2>/out_file</i>	<i>WriteImg/File name</i>
<i>WriteImg/filename</i>	<i>For/Loopin Value 0</i>
<i>For/current value</i>	<i>AnimFile/value</i>

`AnimFile` モジュールとは、`in_filename` と `ext` の間に `Value` の数字をはめ込んだファイル名を出力するモジュールである。今回のマップでは入力するデータファイルのファイル名 (`y21t...`) を `AnimFile` モジュールに、出力するイメージファイルのファイル名 (`y21t...rgb`) を `AnimFile<2>` モジュールに入力する。また、`WriteImg` モジュールのファイルフォーマットは `SGI` にしておく。

弾性板のたわみの度合いは、`DisplaceLat` モジュールのダイヤルをマウスで回すことにより変更する。そして、色は `GenerateColormap` モジュールのカラーバンドを調整することにより変更する。

また、`For` モジュールの `Start Value`、`End Value`、`Step Value` に、作成するファイル名のスタート番号、エンド番号、ステップ数を入力し、`For` モジュールの `Refire` をクリッ

クすることにより、イメージファイルの自動作成が開始される。

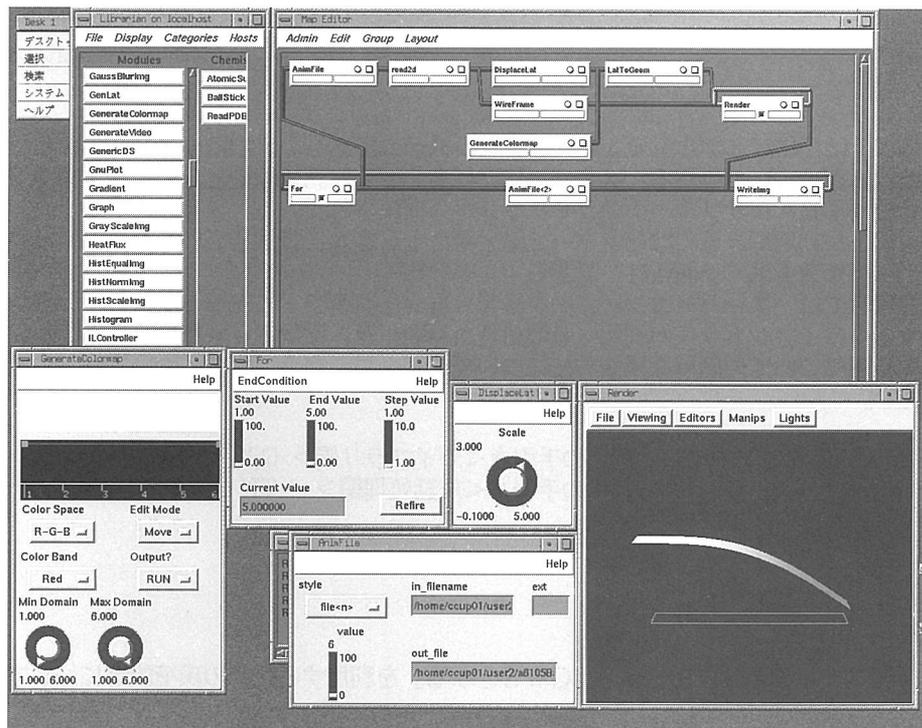


図 4 可視化マップ exeimage.map

5 アニメーションの作成

作成したイメージファイル (y21t1.rgb・・・y21t30.rgb) を、ビデオ制御ユーティリティ VideoDesktop を使うことによってコマ撮りし、アニメーションを作成する。

●詳細はセンターニュース本号の”ONYX によるアニメーションの作成方法について”を参照して下さい。

謝辞

本文の内容に関して懇切なる御指導を頂いた本学内藤林助教授と画像の作成の際に適切な助言を下された本学博士後期課程 1 年の木原一氏、清水保宏技官に感謝致します。

参考文献

1. 出口 弘：汎用可視化ツール *Explorer* の使い方、大阪大学大型計算機センターニュース、22-2(1992)、P66-80
2. 出口 弘、富田 和雄：ONYX によるアニメーションの作成方法について、大阪大学大型計算機センターニュース、24-4(1995)