

FEMを用いた紙の破断シミュレーション

佐藤 崇雄[†] 藤澤 誠[†] 三浦 憲二郎^{††}

近年 CG 製作のために物理法則に基づくシミュレーションを用いる手法に関する研究が盛んに行われている。紙は日常生活において身近な存在ではあるが布に比較して折り目の表現方法や破断の取り扱い、質感の表現が煩雑であるため CG への応用を意図した研究があまり盛んに行われていない。そこで、本研究は紙に関する研究の第 1 歩として、物理法則に基づいた紙の破断シミュレーションを行いその結果を CG に利用することを目的とする。

破断シミュレーションは、まず紙を三角形メッシュでモデル化しそのモデルに対して有限要素法を用いて変位と応力を計算する。得られた応力より破断位置と亀裂の方向を計算し、亀裂を表現するようにメッシュの再構築を行う。さらに、亀裂により生成されたメッシュの境界として折れ線で表現された破断面に対して、turbulence function を用いて面を複雑化しそれをテクスチャとして生成、そのテクスチャをマッピングすることで破断時に顕在化する紙の質感を表現する。破断面を turbulence function によって複雑化することにより、破断形状はメッシュの大きさにあまり依存せず、メッシュの要素サイズを比較的大きくすることで破断シミュレーションをほぼリアルタイムで実現した。

Paper Fracture Simulation Based on FEM

TAKAO SATO,[†] MAKOTO FUJISAWA[†] and KENJIRO T. MIURA^{††}

Recently researches on techniques to use simulations based on the physical laws for CG animations are becoming popular and popular. Although paper exists ubiquitously around us in daily life, it is not straightforward to express its folding lines, fractures, and textures and no much effort has been made on researches on paper for CG. Hence the purpose of this paper is to simulate paper fracture based on the physical laws and use the results for CG animations as the first step to the CG-oriented paper simulation.

The fracture simulation of paper is performed at first by modeling a piece of paper with triangular meshes and applying the FEM to calculate their stresses and strains. Then we determine the starting position of the fracture and the direction of the crack and rebuild the structure of the meshes to express the fracture cross sections. Furthermore for the cross section originally expressed by piece-wise linear segments, by use of the turbulence function we generate natural-looking geometrically complicated cross sections as textures and map them to the meshes to represent the paper characteristics that become clear when the paper is broken. The geometry of the fracture cross section remains similar even though the mesh size is changed because of the noise-mixed texture mapping, that allows us to use relatively large-sized meshes and simulate the paper fracture in almost real time.

1. 緒言

近年、CG 製作のために物理法則に基づいたシミュレーションを用いる手法についての研究が盛んに行われている。水の流れや炎、人の服装の動きなどがリアルに再現され、映画やゲームのアニメーションなど様々な分野で利用されている。しかし、水や衣類などと同様に日常身近に存在する「紙」に関するシミュレーション研究は、他の流体などの研究に比べるとほとんど行われていない。紙のように 2 次元で形状を表現できる物体の研究として、布

のシミュレーションがあり、布の研究は盛んに行われているが、紙の研究は布ほど盛んではない。

そこで、本研究では物理法則に基づいた紙の破断シミュレーションを行い、その結果を CG に利用することを目的とする。

2. システム概要

本研究における紙のシミュレーションの処理手順を、図 2(a) に示す。

まずはじめに、紙を三角形メッシュでモデル化し、そのモデルに対して有限要素法を適用する。運動方程式としては剛性方程式を用い、モデルにかかる応力と変位を計算する。次に、有限要素法より得られた変位と応力より、破断アルゴリズムで破断位置と方向を計算する。そして、破断により生成された亀裂に応じてメッシュを再構築し、

[†] 静岡大学大学院理工学研究科
Graduate School of Science and Engineering, Shizuoka University

^{††} 静岡大学創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University

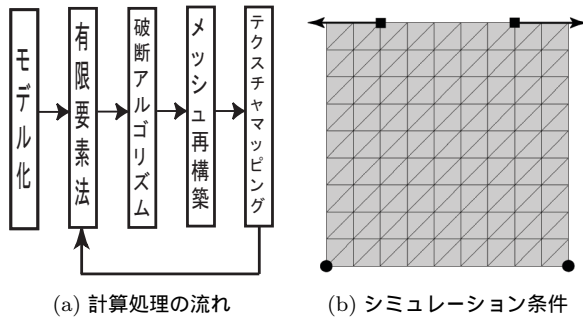


図 1 シミュレーション概要

最後の描画時に、モデルに対しテクスチャマッピングを行うことで、破断時に顕在化する紙の質感を表現する。この有限要素法からテクスチャマッピングまでの処理を 1 ステップとし、このステップを複数回繰り返すことにより破断をシミュレーションする。

本論文に示すシミュレーション結果は、特に言及が無い場合図 1(b) と同様の条件で行った。四角形で示した節点より矢印の方向に等しく荷重を加え、底辺の丸で示した節点で固定し、他では拘束なしの自由とする。

3. 原理

第 2 章で説明した処理の流れに従って、各処理ごとに詳しく述べていく。

3.1 有限要素法

紙を図 3 の節点と要素を持つ三角形メッシュでモデル化し、それにかかる変位と応力を計算する。本研究では力学計算に有限要素法を用いる。

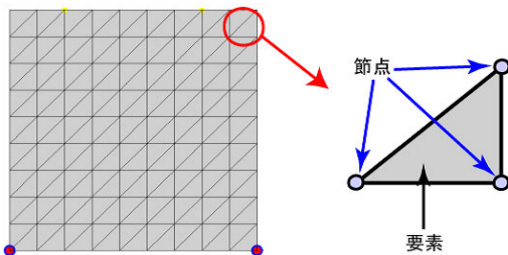


図 2 紙のモデル化

3.1.1 運動方程式

有限要素法に対して本研究で用いる運動方程式は、 F を荷重、 K をバネ定数、 u を変位とする式 (1) の剛性方程式である。

$$\{F\} = [K]\{u\} \quad (1)$$

バネ定数は、ヤング率 E 、断面積 A 、節点間の距離 l とする式 (2) で求める。断面積 A に関しては、すべてにお

いて同一で計算を行う。

$$k = \frac{EA}{l} \quad (2)$$

モデル化された要素について、三角形の 1 辺を 1 つのトラス要素と考える (図 4)。このトラス要素について、剛性マトリクスを三角形の各辺に対して求め、三角形の剛性マトリクスを得る。この計算を繰り返し、三角形の剛性マトリクスを求め、さらに、三角形のモデル全体との対応関係を見ることによりモデルの全体剛性マトリクスを計算する。全体剛性マトリクスと荷重・変位ベクトルを用いることで、各要素にかかる応力を計算し各節点の変位を計算する。得られた変位から各節点を移動させ、紙を変形させる。



図 3 トラス要素

3.2 破断アルゴリズム

有限要素法により節点の変位と要素の応力が計算されたが、それだけでは紙が塑性変形するのみで破断されない。そこで、有限要素法より得られた要素にかかる応力を用いて、紙の破断位置と方向を計算する。

3.2.1 破断方法

破断には「メッシュの稜線部分からの破断」と「節点からの破断」の 2 つの破断方法が考えられる。メッシュの稜線部分から破断を行う場合、図 5(a) において破断方向を破線で示すように、要素にかかる応力より要素の重心を通る様に破断を行うことになるが (図 5(b))、図 5(c) に示した状況においてどのように破断を行うかが問題となる。また、節点と要素の数が節点より破断を行った場合よりも多くなり、計算時間がより多くかかることになる。そのため、本研究ではメッシュの再構築の行きやすさや計算時間の点から、節点より破断を行うこととする。

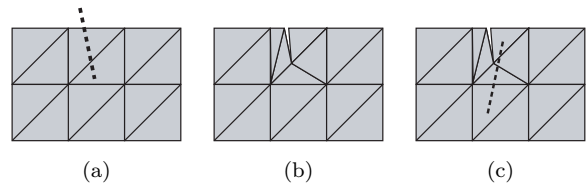


図 4 メッシュの稜線部分から破断を行った場合

有限要素法より得られるのは要素にかかる応力である。そのため、節点にかかる応力を求めるには、要素にかかる応力 σ を節点にかかる応力へと変換する必要がある。節点にかかる応力の計算は、O'Brien 等の 3 次元での手法 [2] を 2 次元へと応用した。

3.2.2 要素にかかる応力の分解

要素にかかる応力を引張と圧縮に分解する．要素の応力の引張成分 σ^+ と，圧縮成分 σ^- は以下の式 (3), (4) で与えられる．

$$\sigma^+ = \sum_{i=0}^2 \max(0, v^i(\sigma)) m(\hat{\mathbf{n}}^i(\sigma)) \quad (3)$$

$$\sigma^- = \sum_{i=0}^2 \min(0, v^i(\sigma)) m(\hat{\mathbf{n}}^i(\sigma)) \quad (4)$$

式 (3), (4) 中の $v^i(\sigma)$ は σ の i 番目の固有値で， $\hat{\mathbf{n}}^i(\sigma)$ はその固有ベクトルである．また $m(\mathbf{a})$ は $\mathbf{a} \in \mathbb{R}^3$ について，固有値として $|\mathbf{a}|$ を持つ 3×3 の対称行列となる．

$$m(\mathbf{a}) = \begin{cases} \mathbf{a}\mathbf{a}^T/|\mathbf{a}| & \mathbf{a} \neq \mathbf{0} \\ 0 & \mathbf{a} = \mathbf{0} \end{cases} \quad (5)$$

これにより，要素の応力をそれぞれの成分に分解できる．

3.2.3 節点の引張力と圧縮力

要素が節点に加える引張力 f_i^+ と圧縮力 f_i^- を計算する．引張力は次式で計算される．

$$f_{[i]}^+ = -\frac{s}{2} \sum_{j=1}^3 p_{[j]} \sum_{k=1}^2 \sum_{l=1}^2 \beta_{j1} \beta_{ik} \sigma_{kl}^+ \quad (6)$$

ここで，式 (6) の s は要素の面積であり， β は要素の関係する節点のワールド座標 $p_{[i]}$ より次式で計算される．

$$\beta = \begin{bmatrix} p_{[1]} & p_{[2]} & p_{[3]} \\ 1 & 1 & 1 \end{bmatrix}^{-1} \quad (7)$$

圧縮力に関しても同様の計算を行う事で $f_{[i]}^-$ を求める．これにより，節点にかかる引張力・圧縮力が求まる．

3.2.4 分離テンソル

破断方向を決定するために分離テンソルを求める．分離テンソル ζ は各節点に作用する引張応力と圧縮応力から得られ，次式によって計算される．

$$\zeta = \frac{1}{2} \left(-m(f^+) + \sum_{f \in \{f^+\}} m(f) + m(f^-) - \sum_{f \in \{f^-\}} m(f) \right) \quad (8)$$

ζ の最も大きな正の固有値が，ある閾値 よりも大きい場合，材料はその節点から破断する．仮に複数の固有値が閾値 よりも大きい場合は，その中で最大の固有値を持つ節点から破断する．破断のワールド座標における方向は，その固有値に対応する ζ の固有ベクトルに垂直な方向である．図 6 に示す矢印について，長さは各節点における固有値の大きさを示しており，その向きは固有値に対応する固有ベクトルを示す．

図 6 の結果の荷重・拘束条件について，前述したように加える荷重の大きさは左右とも同じである．しかし破断方向の分布は左右対称ではない．この左右非対称の結果が得られるの原因としては，モデル自体の形状が左右非対称であることが挙げられ，メッシュの切り方と関連していることがわかる．

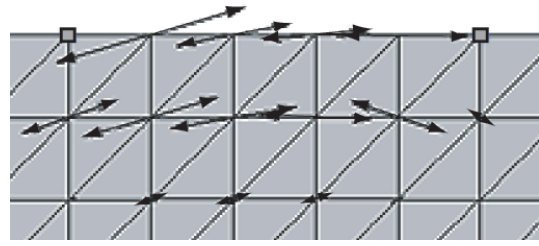


図 5 固有値と固有ベクトル

3.3 メッシュ再構築

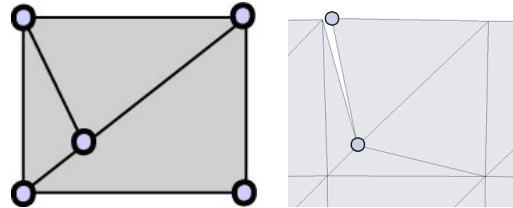
これまでの計算により，破断が起こる節点と破断が進行する方向が計算された．この破断が起こる節点と方向を用いてメッシュの再構築を行う．しかし，破断方向はメッシュの稜線と一致するとは限らない．そのため，破断方向に沿ったメッシュの再構築を行う．

3.3.1 再構築方法

再構築は，破断する節点を 2 つの異なる節点へ分割し，破断方向と要素の境界との交点に新しい節点を生成し，これら 2 つの新しい節点 (図 7(b) の丸で示した点) に関する要素の，各節点の対応付けにより行う．

3.3.2 注意点

再構築を行う際に，節点が他の要素の辺上にあると，隣接する要素が四辺形となり，三角形要素としてモデル化できない (図 7(a))．従って，そのような状態を避けるために再構築に際しては破断する要素と隣接するメッシュの再構築も行う．



(a) 避ける事例 (b) 再構築

図 6 1 回目の再構築

1 回目の破断ではここまでの再構築で十分であるが，2 回目以降の破断では更に再構築する要素が増える．2 回目の破断において，図 7(b) に示すように破断に直接関係のあるメッシュのみの再構築で十分であるが 2 回目以降は，破断されない周囲のメッシュの再構築も行う必要がある．もし 2 回目以降を 1 回目のものと同様に処理すると，図 8(a) のような要素同士が重なる不具合が生じる．図 8(a) の場合において，破断の起点となっている節点は A で示された節点であるが，1 回目の再構築処理より新しい点である B 色と C 色の点が生成され，要素 1 と要素 5 と C の節点に関する要素が再構築される．2 回目以降の処理で

は、ここから要素 3・4 の持つ A の節点を B 色の節点へ割り当てなおす処理を行う。これにより図 3.7(b) の示す正しい再構築を行うことができる。なお、この再割り当ての処理は常に 2 つの要素の節点の割り当てを行えばよいというものではない。破断の起点となる節点と関係する要素の数は、図 8(a) に示したような常に 4 つというわけではなく、図 9(b) の次の破断計算後の再構築においては、関係する要素数は 7 つとなる。そのため再割り当て処理においては、破断起点の節点が関係する要素の数も考慮する。要素の再構築は、基本的にこの 2 つの処理で行われる。

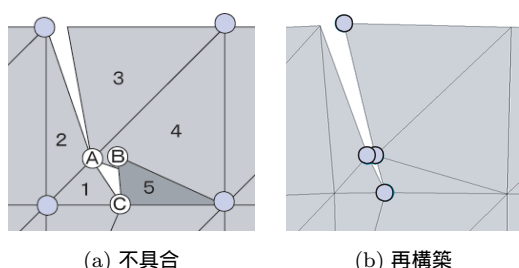


図 7 2 回目以降の再構築

ここまでの処理で、破断の挙動に関する計算とそれに関連するモデルの再構築は終了である。以後、全体剛性マトリクスを再計算し、ここまでのアルゴリズムを繰り返すことで紙の破断における挙動の計算を進行させていく。

3.4 テクスチャマッピング

メッシュを再構築した紙の破断面は、図 9(a) の通り三角形メッシュの 1 辺によって表現されるため、破断面は直線で表現される。しかし、実際の紙の破断面においては繊維のほつれ等があり、破断面が直線になることはほとんどない(図 9(b))。そのため、破断面におけるメッシュのテクスチャ表現を考える。



図 8 破断面比較

3.4.1 テクスチャ処理方法

破断面の表現は断面に沿ってテクスチャを貼り、そのテクスチャに対しノイズ関数を用いることで破断面を表現

する。ノイズ関数を用いる際には、左右の破断面で同じノイズ関数の数値を共有することにより、左右の破断面の形状を合わせる。これによりノイズ関数を用いたことによる、見た目としてのテクスチャの過不足を防ぐことができる上に、より自然な破断の表現を行うことができる。

3.4.2 破断面に沿ったテクスチャの貼り方

破断面に沿って貼るテクスチャに関して、テクスチャの形状はまずはじめに破断面に沿って長方形のテクスチャを貼る。図 10(a) から分かるように、破断面に沿って長方形のテクスチャを貼るだけでは、破断面の稜線の角度によってテクスチャの形状が出てしまう問題が起こる。そのため、角度によってテクスチャの形状が現れないようにするために、隣接する破断面のテクスチャ形状との関係をもて形状を修正し、テクスチャの再構築を行う。テクスチャ形状の修正は、新たな破断面とそれに関係する破断面の長方形テクスチャの交点を求めることで(図 10(b))、図 10(c) のような形に修正する。修正したテクスチャへノイズ関数を載せ(図 10(d))、破断面の外側を背景色にする(図 10(e))。この一連の流れで破断面に沿ってテクスチャを貼っていく。

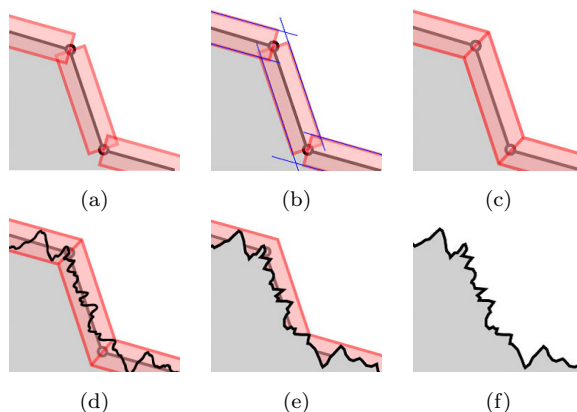


図 9 テクスチャの貼る手順

3.4.3 ノイズ関数

テクスチャに対して用いるノイズ関数について述べる。ノイズ関数としては一般的に白色雑音や Perlin ノイズ [3][4] などがあるが、それらを用いた結果を図 11 に示す。

図 11 に示したように Perlin ノイズを用いた場合には、ノイズの周期性が出てくる。また白色雑音を用いた場合には、破断面の直線的な形状が残ってしまう。そのため、両ノイズ関数を破断テクスチャへ用いることは、破断面の表現として不適である。その為、本研究ではノイズ関数として turbulence function [3][4] を用いる。

3.4.4 turbulence function

turbulence function は次式で計算される。



(a)Perlin ノイズ (b) 白色雑音

図 10 ノイズ

$$n_t(\mathbf{x}) = \sum_i^n \text{abs} \left(\frac{\text{noise}(2^i \mathbf{x})}{2^i} \right) \quad (9)$$

ここで、noise(x) は位置 x における Perlin のノイズであり、abs は絶対値を表す。turbulence function は式 (9) より分かるように Perlin ノイズの周期を変えたものを n 回だけ重ね合わせたものである。 n の値の変化による turbulence function を用いたテクスチャ表現の変化を図 12 に示す。



(b) $n = 1$ (c) $n = 3$ (d) $n = 5$

図 11 turbulence function を用いた結果

図 11, 12 より turbulence function を用いると、Perlin ノイズを用いるよりも周期性が目立たなくなるとともに、メッシュの稜線が目立たなくなり、より自然な破断面が得られる。重ね合わせる数である n の値の大きさについて、シミュレーション環境における画面の解像度との関係から、解像度より小さなノイズは断面の形状として現れないことになる。そのため n の値はあまり大きくても、解像度以下のノイズが増えるのみで外見上の変化は無くなるので、 n の値は適切な値以上を与えていれば十分である。

turbulence function によって形成されるこの形状は、擬似的なフラクタル形状となる。多くの自然物はこのフラクタル性を持っているため、人工物ではあるが天然の素材から作られた紙の破断面の表現に turbulence function を用いるのは、フラクタル性のない一般的なノイズ関数を用いるより適当である。

4. 結 果

4.1 シミュレーション結果

節点へかかる応力を計算し、亀裂を進展させることがシミュレートできた。また、生じた破断面に対してノイズ関数を用いることにより、より自然な破断面を表現す

ることができた。

4.2 計算時間

計算精度を上げるには、単純に要素の分割を小さく・多くすることが精度の高い挙動を得るために必要となる。また、破断面におけるメッシュの直線形状が出てしまい表現が不自然になる問題も、メッシュを小さくモデルを細かく分割する事で、破断挙動の正確さより、自然なシミュレーションを破断面テクスチャ無しで行うことができる。しかし、それを行うことは計算点の増加に繋がり、1 ステップ当たりの計算時間が増えることになる。図 13 は、節点数 98 における 2 ステップ毎の計算結果を 10 ステップ分示したもので、また、図 14 は図 13 の節点数の約 10 倍の 1058 節点数で精度を上げ、[Celeron2.4G][GeForce FX 5200] の環境において同程度の破断量まで計算した結果である。また、要素数と計算時間の関係比較は次の通りである。

要素数	節点数	計算時間 (1 ステップあたり)
64	98	約 0.7 秒
144	242	約 1.8 秒
256	450	約 4.9 秒
400	722	約 12.2 秒
576	1058	約 22.2 秒

得られた結果と計算時間に関する考察をすると、破断面テクスチャを用いない場合での結果 (図 13) は、精度を上げて計算した結果の方がメッシュの直線形状が目立たなくより自然な破断面であるが、これらに破断面テクスチャを用いた場合の結果 (図 14) では、破断の挙動の精度の違いはあるが、両結果ともほぼ同じような結果が得られている。しかし破断面テクスチャを用いた場合において重要なのは、この結果を得るまでの計算時間が両結果で大幅に違うことである。これは破断挙動の精度以外の点において少ないメッシュ数でほぼリアルタイムに、メッシュが細かいものとほとんど変わらない結果が得られることを示している。

全体的な挙動の結果としては、両結果とも破断の量が同程度であるにもかかわらず挙動の結果に若干の違いが見られる。この結果の原因としては、シミュレーションの経過に時間軸を用いているわけではなく、計算回数を用いていることが挙げられる。紙の変形に用いる値はモデルに荷重を加えた後、剛性方程式より計算を行いその結果を用いている。また得られた変位に対して弾性変形を考慮していないため、モデルの変形は塑性変形となる。それに対する破断は、メッシュ形状に依存する。そのため同程度の量の切り込みを入れるにあたって、メッシュが細かい場合と粗い場合では、計算回数の差によるモデルへ加える荷重の回数とそれによる塑性変形の量が図 13 の計算結果の差として現れたと考えられる。

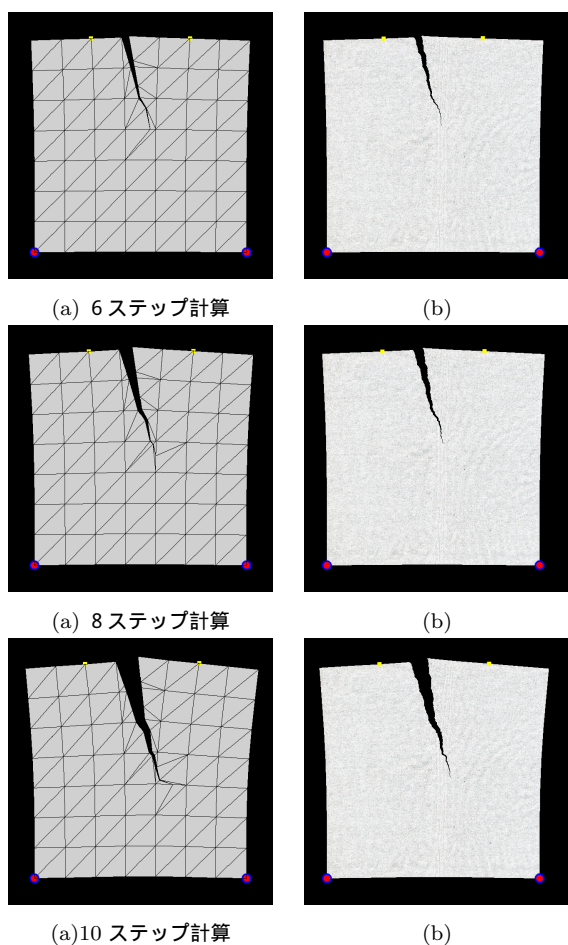


図 12 節点数 98 での結果

5. 結 言

5.1 ま と め

本研究において、紙の特徴の一つである破断面のほつれをモデルのメッシュの大きさに依存せずに表現することができた。更に、破断挙動の精度を要求しないのであれば、精度の高い計算を行ったものとほぼ外見が変わらない破断シミュレーションを、ほぼリアルタイムに近い計算速度で実現した。

5.2 今後の課題

今後の課題としては、三角形メッシュが規則的に生成され、また各節点における材料特性が全ての節点において一様であるため、紙の特徴である繊維の不規則性等が考慮されていない点が挙げられる。これらの紙のモデル化をどのように解決するかが、より自然で正確な紙の破断シミュレーションにおいて重要となってくる。そしてこの問題の解決策は同時に第 3 章で述べた、計算結果の左右非対称性の解決策ともなる。

また、4.2 節の計算時間の項でも述べたように、モデルへ切り込みを入れるにあたり、計算精度によって計算回

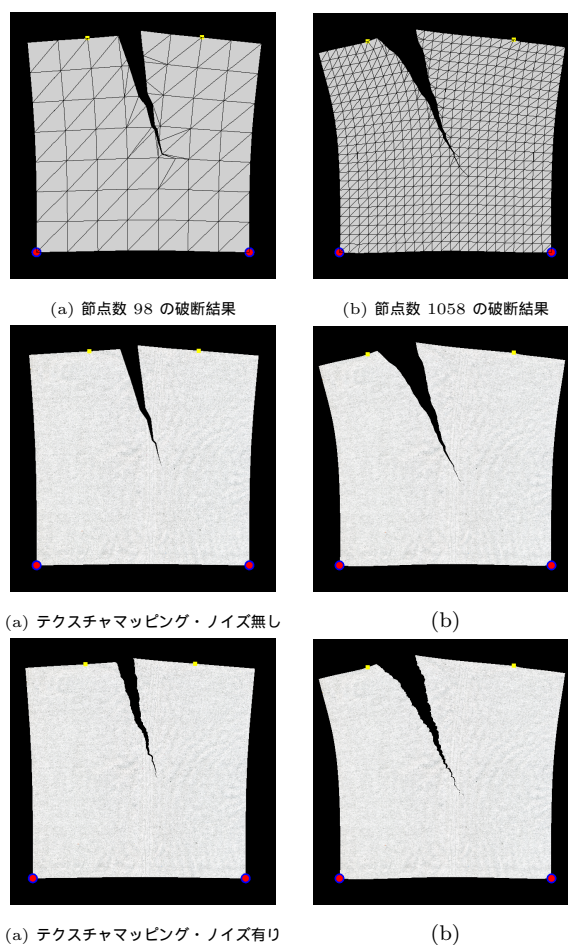


図 13 節点数 98・1058 での計算結果比較

数の差から得られる結果が変化してしまう問題がある。

現段階では 2 次元の引張りによる破断のシミュレーションのみであり、3 次元的なせん断による破断を行う事ができない。日常生活における紙の破断パターンは、本研究の 2 次元の引張りによる破断よりも、3 次元的なせん断による破断のパターンの方がより多くの生活場面において見られる紙の破断パターンであるので、3 次元的なせん断をどのようにシミュレーションし、実現させるかが今後の課題である。

参 考 文 献

- 1) Nelson S.-H. Chu et al, "Real-Time Ink Dispersion in Absorbent Paper," SIGGRAPH2005, 2005.
- 2) James F. O'Brien et al, "Graphical Modeling and Animation of Brittle Fracture," SIGGRAPH99, 1999.
- 3) Ken Perlin, "An Image Synthesizer," SIGGRAPH85, 19(3), 1985.
- 4) Ken Perlin and Eric M. Hoffert, "Hypertexture," SIGGRAPH89, 23, 1989.