2021 年度　卒　業　論　文

主曲率による浸食作用を伴う地形自動生成に関する研究

指導教員：渡辺 大地 教授

メディア学部 ゲームサイエンスプロジェクト
学籍番号　M0118053
小川　和樹

2022 年 2 月

2021 年度　　卒　業　論　文　概　要

論文題目

主曲率による浸食作用を伴う地形自動生成に関する研究

メディア学部 **学籍番号：M0118053**	**氏名**	小川　和樹	**指導教員**	渡辺 大地 教授

キーワード	Tessellation Shader、地形生成、パーリンノイズ、浸食

近年、PC や家庭用ゲーム機などのハードウェアの性能が上がり、広大な土地を移動するオープンワールドのゲームが増えてきている。その中で、自然景観を作ることは雰囲気作りに欠かせない要素で、河川という地形は現実や世界各地に多く存在している。また、ゲームの中でも美しいとされる場所も多く様々なコンテンツで利用されている。河川という形状は、河川の三要素といった自然現象や天候など様々な要因によって複雑な形状をしている。河川を表現する為に、建築や岸工事のための水流や、地形変異をシミュレートする研究が多く行われてきた。そこで本研究では、パラメトリック曲面の一つである Gregory 曲面とテッセレーションシェーダとパーリンノイズを使い、再送信する情報量の少ない地形の生成を提案した山本らの研究を元として、主曲率を使って浸食を伴う地形のランダム生成を目的とした。その結果、ランダムに生成された地形の斜面に沿って一点を通る様々な曲線における曲率を計算することによって、曲率の最大値と最小値を算出し、細かい凹凸がある地形を表現することが可能となった。

目 次

第 1 章

はじめに

1.1 研究背景

近年、PC や家庭用ゲーム機などのハードウェアの性能が上がり、多くの処理が可能となった。ハードウェア性能の向上により、ゲームでは画面内に多くの建物やキャラクターなどの 3D モデルを表示することができる。PS4 や PS5 で発売された、Marvel's Spider-Man[1] や PC でも遊べる ARK: Ultimate Survivor Edition[2]、マインクラフト [3] などのオープンワールドゲームと呼ばれる、広大な地形を移動して遊ぶことができる作品が多く制作されている。しかし、ゲームはリアルタイムに処理を行い、快適なプレイの実現が必要な分野である。ハードウェアの性能が向上し、多くの処理ができるようになったが、広大な地形を全て高精細に描画を行ったり、ポリゴン数の多い 3D モデルを多く表示してしまうと、計算処理にかかる時間が多くなるため、快適なプレイが実現できなくなるという問題が生じる。そのため、表示する地形やキャラクターの表示数やポリゴン数を調整する必要がある。3D ゲームでは計算処理を減らす方法として 3D モデルの詳細度制御を行っている。一般的に Level of Detail（以降「LOD」）と呼ばれる方法であり、事前にポリゴン数の違う 3D モデルを複数体用意しておき、カメラと 3D モデルの距離に合わせて表示するモデルを切り替える。カメラから離れた位置にある 3D モデルはポリゴン数を抑えたものを使用して表示を行い、カメラから近い位置にある 3D モデルにポリゴン数が多く詳細なものを使うことで、画面の見た目を大きく損なわずに計算処理を軽くしている。

GPU を使い 3D モデルを表示する場合、GPU に 3D モデルの情報を一度だけ送り込み、GPU 内のメモリ上に送信した情報を保存しておく。その後、GPU のメモリ上に保存した 3D モデルの情報を使い、表示を行っていく。また、表示するモデルの頂点数やポリゴンを構成するトポロジーに変更がある場合、変更の起こった部分の情報を新たに GPU に送信しなければならない。そして、多くの情報を CPU から GPU へ送信してしまうと処理が重くなってしまうため、再送

信する情報は減らす必要がある。LOD ではカメラと表示する 3D モデルの距離に応じて、表示する 3D モデルを切り替えている。そのため、カメラとモデルの距離が変わり、表示する 3D モデルの切り替えが発生した場合、GPU に新しい 3D モデル情報の送信する必要がある。LOD を使いポリゴン数を抑える場合、事前に GPU のメモリ上に保存していた情報と比べ、頂点数やトポロジーが大きく変化するため、モデルの情報を全て再送信することになる。

オープンワールドゲームなどの広大な地形を生成する場合、描画処理を行う前に広大な地形をいくつかの段階にわけて分割して CPU 上で準備しておき、カメラとの距離に応じて分割した地形を GPU へと送信して表示を行う。FARCRY6[4] では、最初に地形生成に必要な形状データであるハイトマップやノーマルマップなどを何段階かのミップマップとして準備する。次に描画処理を始める前に CPU 内で何キロもの大きな地形を繰り返し分割していく。分割するたびに分割した地形情報の保存を行い、複数の段階で分割した地形を保存しておく。次に広大な地形のどこにカメラがいるのか位置の算出し、先ほど分割し保存しおいた情報から適切な地形を選んでいく。カメラ位置を元に、カメラと地形の距離が遠い時には分割回数が少なく一つ一つが大きい地形を選び、カメラ地形の距離が近い時には分割回数が多く一つ一つが小さい地形を選ぶ。分割した地形と選んだ分割段階に対応する最初に準備した地形生成に必要な形状データを GPU に送る。そして、GPU に送られた情報をもとに、カメラに映らない部分の地形の描画処理を無視する。最終的に送られた情報をもとに地形を生成することでカメラ距離に応じた描画処理を行っている。しかし、この方法ではより細かな地形を表示しようとした場合、さらに細かい分割を行いその段階の情報を保持しておき、それに対応する地形生成に必要な形状データの準備が必要となる。そのため、準備する情報量も増えてしまい、表示する地形が細かくなればなるほど GPU への多くの情報の送信が必要となる。

地形に関する研究は、昔から行われていて非整数ブラウン運動を用いた生成方法 [5] や中点変位

法を用いた手法 [6] などのフラクタル手法を使ったものがある。また、等高線や標高値などから地形モデル生成する手法 [7] や等高線と B-spline 曲線を用いた生成手法 [8] のように、事前に用意した情報から地形を生成する方法も存在する。

地形は、火山活動、水の流れ、風などの自然現象により、地表面が変形を受けることによって形成される。その中でも水の流れは、地形の形成に大きな役割をしている。雨となって地表面に降り注いだ水は、山の高いところから低いところへ流れ、やがて川となり海へ向かう。その過程で、水は地表面を浸食し、削り取った土や砂を運搬する。そして、徐々に細かい粒子に砕いていき、再び地表面に積まれる。これが繰り返されることによって、川から海に繋がる河口付近には細かい砂や泥が堆積した三角州が形成され、人々の暮らしの土台となっている。

河川に関する研究は、河口周辺部の波浪場及び流れ場の計算モデルの研究を行った椹木ら [9] や計算負荷を減らしつつ，広域の河川区間に対する三次元流動計算を実現するために，新しいモードスプリット法と並列化処理を組み込んだ計算効率性の高い三次元河川流モデルを開発した加藤ら [10]、試験施工区間における融雪及び夏季出水時の流れと河床変動の再現、計算結果を現地河床形状と比較し、モデルの適用性について研究を行った横山ら [11]、侵食を考慮した河川形状の生成シミュレーションを行った戸沼ら [12] がいる。これらの研究は、川の流れが早いところと遅いところを分析しているが、CG による描画は考慮していない。その他に、釧路川の蛇行について、平面二次元モデルと三次元モデルを適用し、その妥当性を検証した掛川ら [13] の研究がある。掛川らは、蛇行部の川の流れを CG を使ってシュミュレートしているが、川の蛇行部に焦点を当てているため、山岳地帯の侵食を表現することはできない。

GPU 側で高速に地形の自動生成することを実現した手法として山本ら [14] がある。しかし、山本らの手法では平坦な地形を生成することには優れているが、山岳地帯特有である河川や水の流れなどの影響で、浸食のある地形を生成することは難しい。

本研究では、浸食を伴う地形のランダム生成を目的とする。山岳地帯特有である凹凸のある地形を生成するために、「曲率」を手法として着目した。曲線は、道路のカーブやジェットコースターのループなどにも使われているもので、曲線あるいは曲面上のある点での曲がり具合を表す量のことをいう。この曲率を山本ら [14] の手法を元に地形の自動生成に組み合わせることで、浸食を表現した。その結果、ランダムに生成された地形の斜面に沿って一点を通る様々な曲線における曲率を計算することによって、曲率の最大値と最小値を算出し、細かい凹凸がある地形を表現することが可能となった。

1.2 論文構成

本論文は全 5 章にて構成する。2 章では、山本手法について説明を述べる。3 章では本手法について述べる。4 章では本手法を用いた実行結果について述べる。そして、5 章にてまとめを述べる。

第 2 章

山本手法

2.1 山本手法

自然形状を作るうえで問題になるのが、複雑な自然形状データを CPU から GPU に転送する時間がかかる。例えば、マインクラフトのワールド生成の時に起きる膨大なデータを読み込むロード時間が数十秒かかってしまうことや、カメラ移動時に自然形状が遅れて発生してしまう問題がある。その問題を解決したのが、山本手法である。山本らは「テッセレーションシェーダを用いた描画要素数の動的制御による地形高速化に関する研究」[14] でモデルの形状データの情報量を減らすことに成功した。本研究は、この山本手法を基本としており、本手法の解説には山本手法の内容が不可欠であるため、本章では山本手法の概要を述べる。

2.2 パラメトリック曲面

山本手法では、パラメトリック曲面と Gregory 曲面を使って地形の自動生成を行っている。この章では、山本手法について説明を行う。まず、パラメトリック曲面はいくつかの制御点を設定することにより定義される形状のことである。一組のパラメータを与えると対応する位置が算出でき、それらを繋ぎ合わせることで出来上がる曲面がパラメトリック曲面である代表的なものとして Coons[15] がインタラクティブに形状を扱う曲面表現式として提案した Coons 曲面や Bezier[16][17] が自動車の設計を行うために作成した Bezier 曲面などがある。このパラメトリック曲面の制御点を GPU に送り、送った制御点情報から GPU 内で実際の曲面を計算して形状を生成や変更を行うため、カメラ距離の変更時に再送信の情報量を減らすことができる。本研究では複数枚のパラメトリック曲面を生成し、繋ぎ合わせることで広大な地形の表現を行う。そのため、複数のパラメトリック曲面を繋ぎ合わせたため、本研究では使用するパラメトリック曲面として Gregory 曲面 [18][19] を選択した。

2.3 Gregory 曲面

Gregory 曲面は、Gregory がおこなった一般 Coons 曲面の拡張を Bezier 曲面に用いた曲面である。Bezier 曲面は、パラメトリック曲面の一つであり、制御点と呼ぶ位置ベクトルのみで定義する曲面形状のことである。$n \times m$ 次の Bezier 曲面の曲面表現式は式 (2.1) のようになる。

$$\mathbf{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_i^n(u) B_j^m(v) \mathbf{P}_{ij} \tag{2.1}$$

P_{ij} は Bezier 曲面の制御点を表し、u と v は $0 \leqq u, v \leqq 1$ である。制御点は u 方向に $n+1$ 個、v 方向に $m+1$ 個、合計 $(n+1)(m+1)$ 個存在している。また、$B_i^n(u)$、$B_j^m(v)$ は Brnstein 基底関数であり、式 (2.2) は Bernstein 基底関数をあらわしたものである。

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \tag{2.2}$$

ここで式 (2.3) は 2 項係数である.

$$\binom{n}{i} = {}_nC_i = \frac{n!}{i!(n-i)!} \tag{2.3}$$

Bezier 曲面はこの制御点を移動することで、形状を変更することが可能である。しかし、隣り合う Bezier 曲面同士を滑らかに接続するには複雑な計算が必要となってしまう。

そこで Bezier 曲面を拡張したものが Gregory 曲面である。双 3 次 Gregory 曲面は 20 個の制御点で表現し、制御点は $\mathbf{P}_{ijk}(i = 0...3, j = 0...3, k = 0, 1)$ にて表す。式 (2.4) は Gregory 曲面の曲面表現式を表したものである。

$$\mathbf{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_i^n(u) B_j^m(v) \mathbf{Q}_{ij}(u,v) \tag{2.4}$$

$B_i^n(u)$ と $B_j^m(v)$ は式 (2.2) の Bernstein 基底関数である。

第 3 章

提案手法

3.1　提案手法

まず初めに山本手法を使って、広大な土地を表現する。次に曲面で生成した形状に対してノイズを使い細かい形状を加えて、地形の形状を設定する。生成された地形の斜面に沿って、ある一点を通る様々な曲線における曲率について計算を行う。最後に、曲率の最大値と最小値を算出し、地形に凹凸を表現する。

以上の内容を山本手法であるパラメトリック曲面とテッセレーションシェーダ、パーリンノイズを用いて、少ない情報の再送信により、地形の生成を行う。処理の説明に関して、CPU 側の処理と GPU 側の処理に分かれて説明していく。CPU 側の処理は GPU に送信するための 3 つの情報を準備して送信するという内容になっている。送信する処理は 3 つあり、各パラメトリック曲面の制御点、各曲面の領域情報、生成する地形の範囲情報の 3 つである。GPU 側の処理では送られてきた情報を使い GPU 内で曲面による生成する地形の形状を計算し、パーリンノイズを使用し細かい形状を加えて地形の形状を設定する。生成された地形の形状の斜面に沿って曲率を計算を行い、最大値と最小値を計算し、地形に凹凸を表現する。以下に CPU 側と GPU 側の処理をまとめたものを表しておく。

3.2 曲面

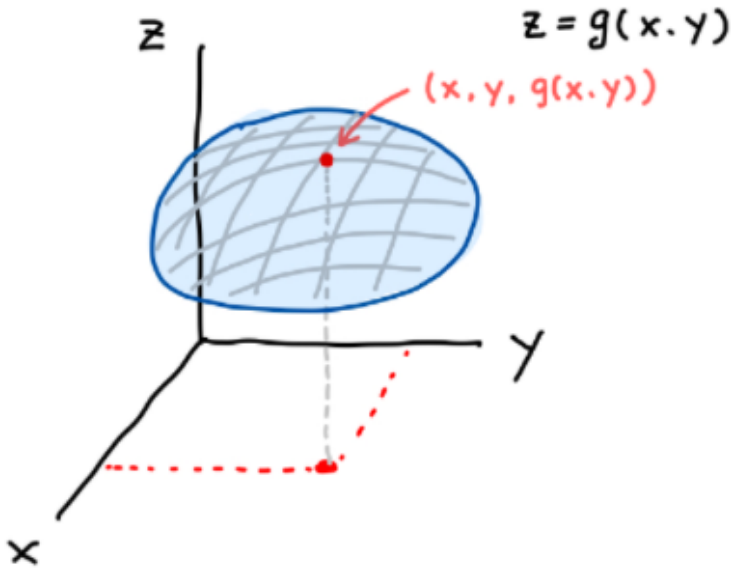

図 3.1　$z = g(x, y)$

図 3.1 は xyz 直交座標系での曲面の模式図である。$z = g(x, y)$ で表される曲面 S を考える。この曲面上の点 P の位置ベクトルは

$$\mathbf{P} = (x, y, g(x, y)) \tag{3.1}$$

となる。

点 P から x 方向に微小量 Δx だけ動いた点を点 Q、点 P から y 方向に微小量 Δy だけ動いた点を点 R とする。図 3.2 は、わかりやすくした図解である。

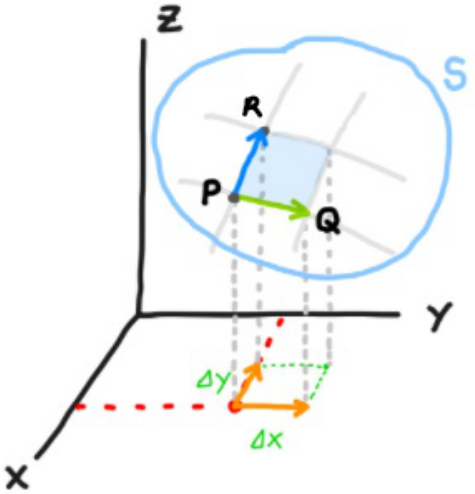

図 3.2　点 Q と点 S を動かした図

点 Q と点 R の位置ベクトルはそれぞれ、次のようになる。

$$\begin{aligned} &\text{点 Q:}\mathbf{P} + \left(\Delta x,\ 0,\ \frac{\partial g}{\partial x}(x,y)\Delta x\right) \\ &\text{点 R:}\mathbf{P} + \left(0,\ \Delta y,\ \frac{\partial g}{\partial y}(x,y)\Delta y\right) \end{aligned} \tag{3.2}$$

3.3　単位法ベクトル

図 3.3 は法線である。PQ と PS 両方に垂直な向きは法線方向になる。

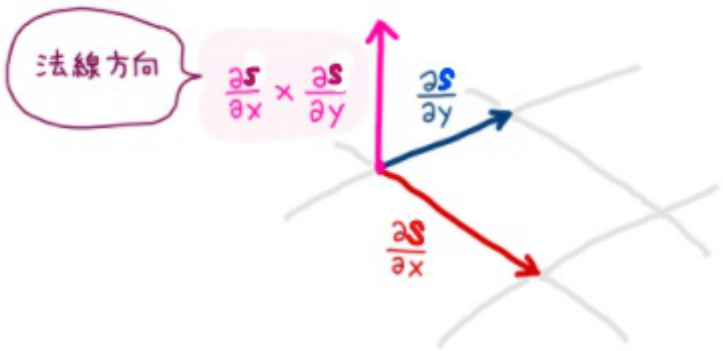

図 3.3　法線方向

よって、

$$\frac{\partial \mathbf{S}}{\partial x} \times \frac{\partial \mathbf{S}}{\partial y} \tag{3.3}$$

は法線方向を向くことがわかる。大きさを 1 にするために、そのベクトルの大きさそのもので割ると、単位法線ベクトル $\mathbf{n}$ がわかる。

$$\mathbf{n} = \frac{\frac{\partial \mathbf{S}}{\partial x} \times \frac{\partial \mathbf{S}}{\partial y}}{\| \frac{\partial \mathbf{S}}{\partial x} \times \frac{\partial \mathbf{S}}{\partial y} \|} \tag{3.4}$$

3.4 第 1 基本量と第 2 基本量

第 1 基本量は曲面 S の微分だけから定まるものである。第 2 基本量は曲面の法ベクトル n が必要である。これは、曲面の接線方向の情報が「曲面の内部情報 (曲面の接戦近似)」と見なせるのに対し、曲面の法線方向の情報は「曲面の外部情報 (曲面 R^3 の中でどのように存在するかという情報)」を表すものである。よって、「第 1 基本量は曲面の内在的量であり、第 2 基本量は曲面の外在的量である」と表現できる。

曲面 $\mathbf{S}(u, v)$ に対して、

$$\begin{aligned} E(u,v) = \frac{\partial \mathbf{S}}{\partial u}(u,v) \cdot \frac{\partial \mathbf{S}}{\partial u}(u,v) = |\frac{\partial \mathbf{S}}{\partial u}(u,v)|^2 \\ F(u,v) = \frac{\partial \mathbf{S}}{\partial u}(u,v) \cdot \frac{\partial \mathbf{S}}{\partial v}(u,v) \\ G(u,v) = \frac{\partial \mathbf{S}}{\partial v}(u,v) \cdot \frac{\partial \mathbf{S}}{\partial v}(u,v) = |\frac{\partial \mathbf{S}}{\partial v}(u,v)|^2 \end{aligned} \tag{3.5}$$

を、曲面 $\mathbf{S}$ の第 1 基本量と呼ぶ。

また、曲面 $\mathbf{S}(u, v)$ に対して、

$$
\begin{aligned}
L(u,v) &= \frac{\partial^2 \mathbf{S}}{\partial u^2}(u,v) \cdot \mathbf{n}(u,v) \\
M(u,v) &= \frac{\partial^2 \mathbf{S}}{\partial u \partial v}(u,v) \cdot \mathbf{n}(u,v) \\
N(u,v) &= \frac{\partial^2 \mathbf{S}}{\partial v^2}(u,v) \cdot \mathbf{n}(u,v)
\end{aligned} \tag{3.6}
$$

を、曲線 $\mathbf{S}$ の第 2 基本量と呼ぶ。

3.5 主曲率

曲面上の点 P における法線（接平面に垂直な線）を含む平面を考える。この平面で曲面を切断すれば、その切り口は平面上の曲面になり、点 P における曲面の曲率が定まる。法線を軸として平面を回転させると、切り口となる曲面の形が変化する。それに応じて、点 P における曲率の値も変わるが、滑らかな曲面ならば最大値と最小値が存在する。最大値と最小値を、点 P における曲面の「主曲率」という。

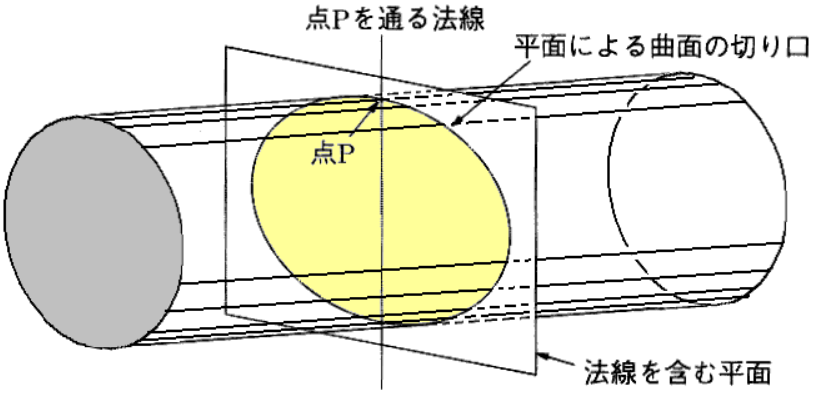

図 3.4　円柱側面の曲率 [20](p.131 の図 6 より転載)

図 3.4 は、半径 r の円柱面である。法線は円柱の軸と直行する直線で、法線を含む平面による円柱の切り口は楕円になる。曲率の最大値は平面が円柱軸に直行するときの $\frac{1}{r}$、最小値は平行に

なるときの 0 である。また、半径 r の球面の場合、法線は球の中心を通る直線で、法線を含む平面による球の切り口は常に円になるので、主曲率は最大値と最小値とも $\frac{1}{r}$ である。

第 1 基本量と第 2 基本量の式で求めた、E、F、G と L、M、N で二次方程式 (3.7) の解が主曲率となる (P129〜P133)。

$$(F^2 - EG)k^2 - (2FM - GL - EN)k + (M^2 - LN) = 0 \tag{3.7}$$

この式の実数解のうち、絶対値が大きいほうを取り出し、それが + だった場合のみ地形を凹ませることとする。

3.6 主曲率による高さの変更

主曲率で求めた式 k を用いて、式 (3.8) により新たな P_z を求める。

$$P'_z = P_z - \alpha k \tag{3.8}$$

この α の値は、パーリンノイズの強弱を決める数値である。これを計算することによって、地形が変化する。

第 4 章

出力結果と検証

4.1　開発環境

前章で述べた手法を実装し、検証を行った。開発言語は C^{++} を用いて、OS は Windows10 を使用した。また、実装は OpenGL をベースとする FK ライブラリ [21] 上で行った。開発および検証に用いた PC のスペックを表 4.1 に示す。

表 4.1　実行環境

OS	Windows10
CPU	Intel(R)Core i5-11600KF CPU @ 3.9GHz
メモリー	16GB
GPU	Geforce RTX 3080ti

4.2　出力結果

今回は、テッセレーションシェーダ内で本手法を使い地形生成を行った。

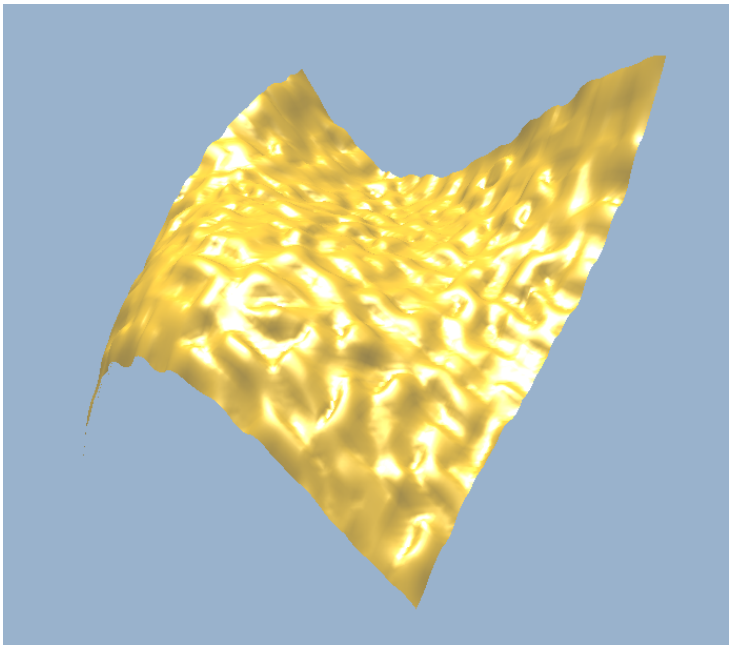

図 4.1　出力結果 1

図 4.1 は、山型に大きく膨らんでるところや谷型に凹んでいるところがわかり、侵食作用が進

んだ山岳地帯特有の地形を表している。

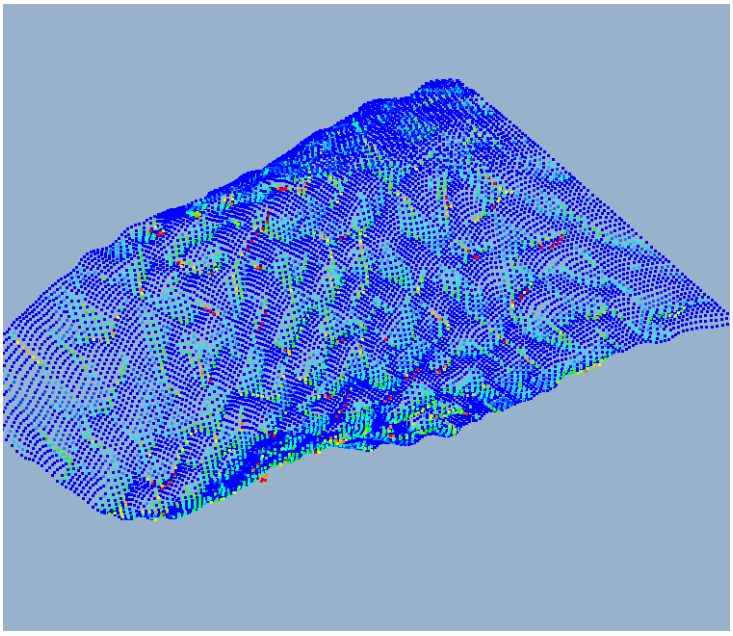

図 4.2　出力結果 2

図 4.2 は、曲面上の各点の曲率を疑似カラーにより着色した様子である。図 4.2 をよくみると青や水色、赤と色分けされているのがわかる。これは、山型や平らになっているところは青で表し、谷型で特に凹みが大きいところが赤く表示されるようになっている。また、水色や黄色は谷型で少し凹んでいるところを表している。

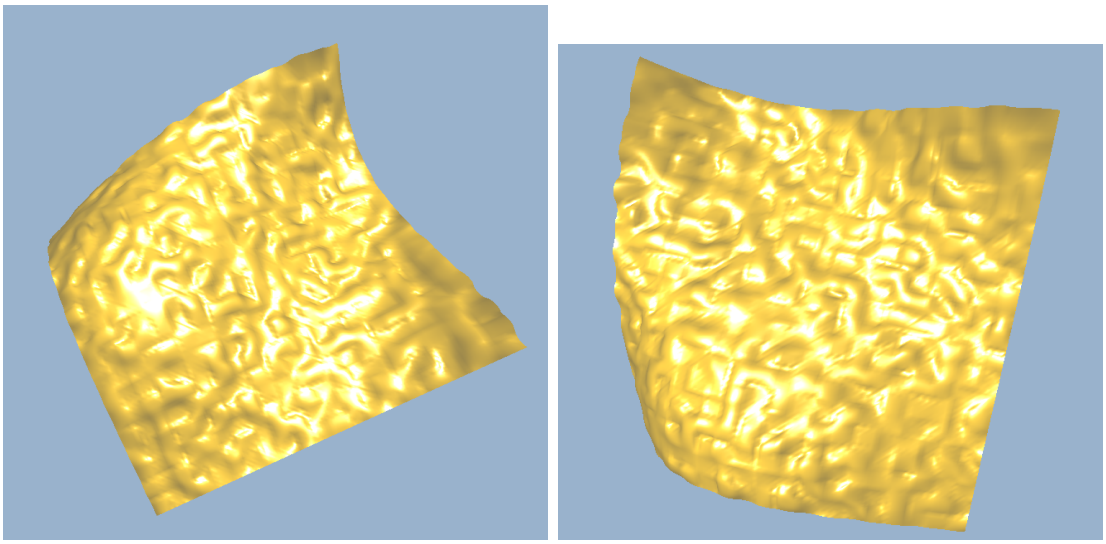

図 4.3　パーリンノイズを決まった形に変更させたもの

図 4.3 は、パーリンノイズの生成時に用いる乱数列を変化させ、ノイズのパターンを変化させたものである。これは、地形全体のパーリンノイズを決まった形に固定させることができる。

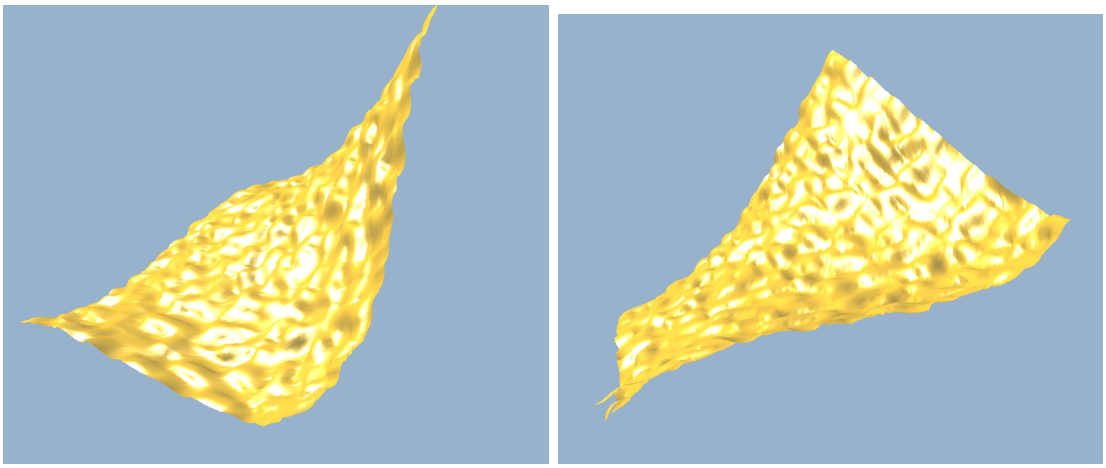

図 4.4　Gregory 曲面の形を変更したもの

図 4.4 は、Gregory 曲面の生成時に用いる乱数列を変化させ、曲面の形を変化させたものである。これは、グレゴリ曲面の形を変更することができる。

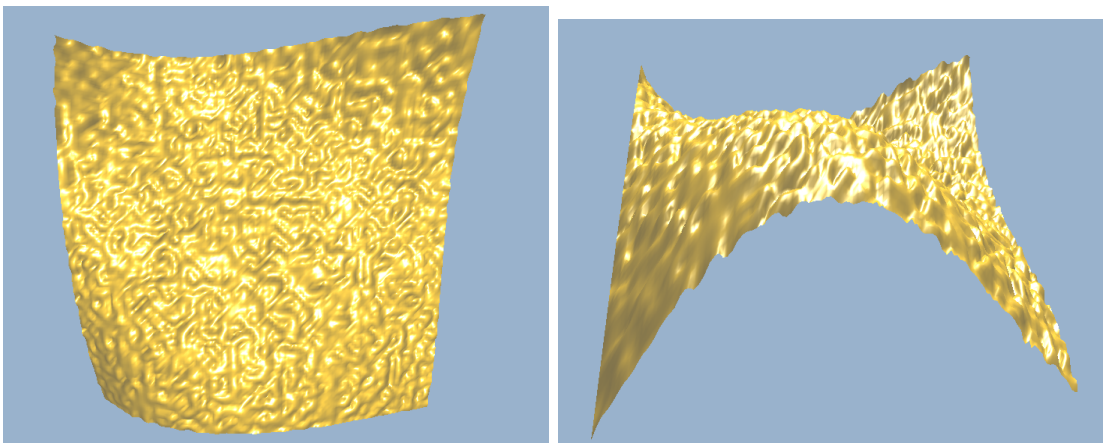

図 4.5　パーリンノイズの分割数を増やしたもの

図 4.6 は、曲面 1 つ 1 つのパーリンノイズの分割値を増やしたものである。図 4.1 よりも地形の凹凸が増していることがわかる。

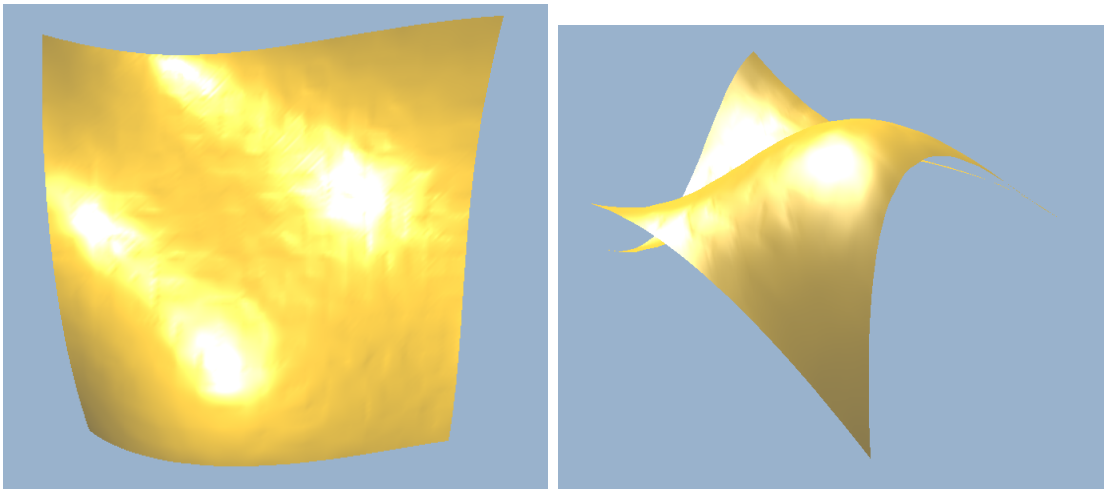

図 4.6　パーリンノイズの強さの変更

図 4.6 は、パーリンノイズの強弱を決めるもので、値を小さくすることによって凹凸が小さくなったり、大きくすることによって凹凸を大きくすることができる。

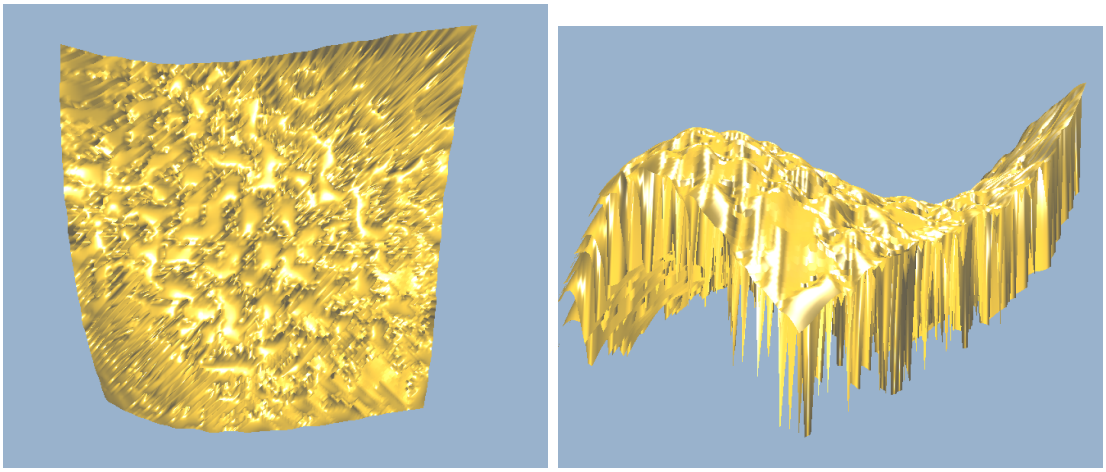

図 4.7　凹み具合を調節

図 4.7 は、主曲率が高いところの凹み具合を調整するもので、図 4.1 よりも地形の凹みが激しくなる。

4.3　考察

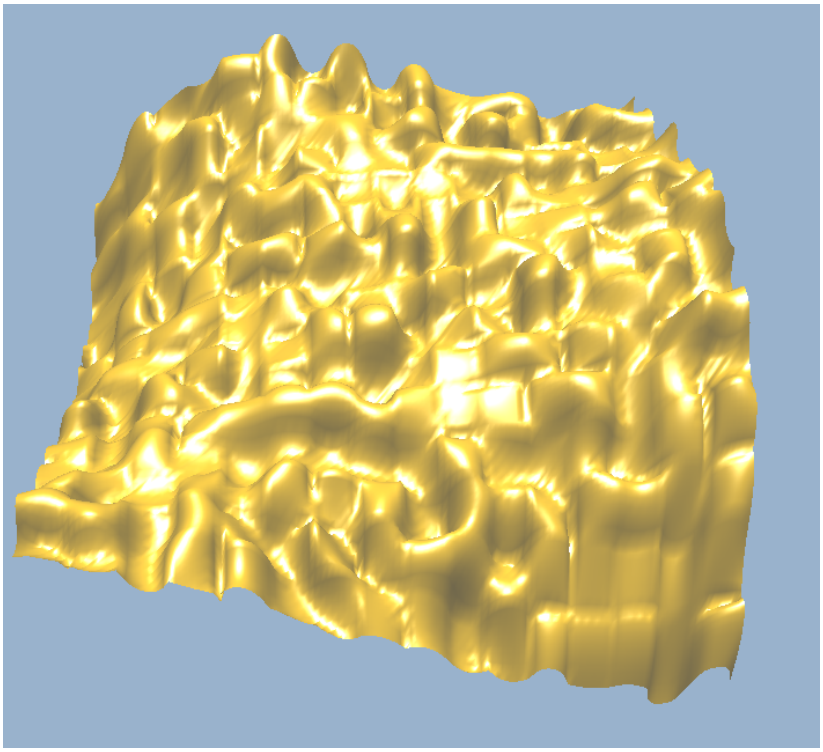

図 4.8　浸食作用が進んだ山岳地帯特有の地形

本研究では、主曲率による浸食作用を伴う地形自動生成に関する研究を行った。プログラムを組んで実行してみた結果、山岳地帯特有である凹凸のある地形を作成することが可能となった。さらに、主曲率 k が $+$ になった時だけ地形に凹凸を表現することによって、地形の形状はバランスの良いものを作成することができると考えられる。しかし、山になっているところから、時間が経つにつれて浸食が進んで図 4.8 になるというシュミュレートは、今回の研究では再現することは難しいと考えられる。

第 5 章

まとめ

近年ではハードウェアの性能が上がり、広大な地形を移動し遊ぶことができるオープンワールドと呼ばれるゲームが多く存在している。一般的に GPU を使用する場合はモデルの情報を一度送信してメモリ上に保存し、メモリ情報を繰り返し使い表示を行う。モデルの形状データや頂点数が変わる場合にモデルの再送信を行うが、再送信する情報量はなるべく減らす必要がある。

生成する地形の領域や位置を変更したい場合、少ない情報の再送信によって、動的に地形の生成や変更が行える山本らの研究を元にし、本研究では主曲率を使い浸食を伴う地形の生成を目的とした。ランダムに生成された地形の斜面に沿って一点を通る様々な曲線における曲率を計算することによって、曲率の最大値と最小値を算出し、細かい凹凸がある地形を表現することができる。また、地面のへこんでいるところとでっぱっているところを色分けすることによって、主曲率が高いところと低いところの分析が一目見ただけで容易となった。

本手法では、地形をパーリンノイズを使ってランダムに生成しているため、ある一部分だけを精密にデザインしたい場合の調整が難しい。また、仮に川を生成して時間が経つにつれて地面が削れていく様子を再現したい場合に、この手法で再現するのは難しい。

謝辞

本研究を進めるために多くのご指導やアドバイスをしてくださった渡辺先生、阿部先生や研究室のメンバーに感謝したいと思います。

プログラミングの支援や論文、研究のアドバイスなど数多くご指導、支援してくださった渡辺先生に、感謝しています。半年間研究の内容がなかなか定まらずに右往左往していた時期もありましたが、そんな中でも優しく接してくださいました。渡辺先生に心から感謝しています。

ポスター発表のときに的確にアドバイスをしてくださった阿部先生に感謝しています。ありがとうございます。

研究室のメンバーは、緊急事態宣言が何回も発令されてほぼほぼ顔合わせることはありませんでした。それでも、互いに協力し発表の時こうしたほうがいいよねとか、ここ直したほうが良くなるよねというアドバイスありがとうございます。自分では当たり前だと思って、気づかなかったことに気づけたのは研究室のメンバーがいてくれたからだと思います。ありがとうございました。

最後に、産んでから 22 年間育ててくれた両親に感謝します。不自由なく生活することができたのは、2 人が頑張ってくれたおかげです。ありがとう。

参考文献

[1] Marvel spider-man. https://www.jp.playstation.com/games/marvels-spider-man/. 参照：2022.01.16.

[2] Spike chunsoft ark:ultimate survivor edition. https://www.spike-chunsoft.co.jp/ark/. 参照：2022.01.16.

[3] Minecraft. https://www.minecraft.net/ja-jp. 参照：2022.01.16.

[4] Ubisoft farcry6. https://www.ubisoft.com/ja-jp/game/far-cry/far-cry-6. 参照：2022.01.16.

[5] 石川貴之, 青木由直. フラクタルによる図形の生成とその応用. テレビジョン学会技術報告, Vol. 8, No. 46, pp. 49–54, 1985.

[6] Alain Fournier, Don Fussell, and Loren Carpenter. Computer rendering of stochastic models. *Commun*, Vol. 25, No. 16, pp. 371–384, 1982.

[7] 安居院猛, 宮田一乗, 中嶋正之. 三次元山岳形状の等高線からの自動作成法. 電子情報通信学会論文誌 D, Vol. 69, No. 12, pp. 1905–1912, 1986.

[8] 張志毅, 今野晃市, 徳山喜政. 周期的 b-spline 曲線を利用した山岳地形の 3 次元モデル生成手法. 情報処理学会研究報告グラフィクスと CAD（CG）, No. 86, pp. 1–6, 1986.

[9] 椹木亨, 李宗隻, 出口一郎. 河口周辺の海浜流及び地形変動モデルに関する研究. 海岸工学講演会論文集, Vol. 31, pp. 411–415, 1984.

[10] 二瓶泰雄, 加藤祐一, 佐藤慶太. 広域河川流計算のための新たな三次元流動モデルの開発と洪水流計算への応用. 土木学会論文集, Vol. 2005, No. 803, pp. 115 – 131, 2005.

[11] 横山洋, 渡邊康玄, 鈴木優一. 分岐・合流流れを有する河川における河床変動計算に関する研究. 北海道開発土木研究所月報 615 号, pp. 2–9, 2004.

[12] 戸沼葉弓, 赤木康宏, 北嶋克寛. 侵食を考慮した河川形状の生成シミュレーション. 情報処理学会 研究報告グラフィクスと CAD(CG), Vol. 2008, No. 14, pp. 115–120, 2008.

[13] 掛川恵梨子. 蛇行河川を対象とした平面二次元モデルと三次元モデルの適用性比較. 平成 24 年度 土木学会北海道支部 論文報告集, Vol. 69, No. B-29, pp. 1–4, 2012.

[14] 山本馨加. テッセレーションシェーダを用いた描画要素数の動的制御による地形描画高速化に関する研究. 情報処理学会研究報告デジタルコンテンツデザイン（DCC）, Vol. 2021-DCC-27, No. 16, pp. 1–8, 2021.

[15] S.A.Coons. *Surfaces for computer-aided design of space figures*. Massachusetts Institute of Technology, Electronic Systems Laboratory, 1964.

[16] P.Bézier. *Definition numerique des courbes et surfaces*, Vol. 11. Automatisme, 1966.

[17] P.Bézier. *Definition numerique des courbes et surfaces(ii)*, Vol. 12. Automatisme, 1967.

[18] 千代倉弘明, 鳥谷浩志. 3 次元 CAD の基礎と応用. 共立出版, 1991.

[19] H.Chiyokura and F.Kimura. Design of solids with free-form surfaces. *Computer Graphics*, Vol. 17, No. 3, pp. 289–298, 1983.

[20] 吉田伸夫. 完全独習相対性理論. 講談社, 2016.

[21] Fine Kernel Project. Fine kernel tool kit. `https://gamescience.jp/FK`. 参照：2022.01.16.