

Dynamic DDM による氾濫シミュレーションの高速化

山口 悟 史[†] 岩 村 一 昭[†]

氾濫シミュレーションにかかる処理量を低減させる計算手法 Dynamic DDM を提案し、その有効性を検証した。Dynamic DDM は、シミュレーション実行中に計算する領域を水の流れに合わせて動的に拡大、縮小させる手法である。実利用を想定した実験では、従来手法の 1/14 の処理量で水害をシミュレートできた。Dynamic DDM は処理量を低減することが明らかになった。提案手法を用い、さらに GIS を用いることで、地球の任意の地点で氾濫シミュレーションが可能なソフトウェアが実現できた。今後結果の検証結果が蓄積されることにより、このような広域を対象とするシミュレーションシステムが実利用されていくと考えられる。

Fast Flood Simulation Method Using Dynamic DDM

SATOSHI YAMAGUCHI[†] and KAZUAKI IWAMURA[†]

We propose a Dynamic Domain Defining Method (Dynamic DDM) to reduce the calculation cost of flood simulation, and validate its speeding-up effect using the actual flood simulation system. The proposed calculation method expands or shrinks its calculation domain during the simulation, and excludes the unflooded areas from the calculation domain so that the calculation cost is minimized. The experiments of the actual system show that the method is about 14 times faster than conventional method, and therefore it is very suitable for the fast flood simulation.

1. はじめに

近年、集中豪雨や台風等による水害が頻発し、効果的な被害軽減策が求められている。被害軽減のためには、起こりうる水害をあらかじめ把握しておく必要がある。このような背景から「氾濫シミュレーションシステム」が提案されている^{1)~4)}。このシステムに堤防が決壊する地点等を入力すると、浸水領域や水深分布が書かれた地図が出力される。地図を見れば、たとえば決壊地点により物件が受ける浸水被害が異なることを理解できる。

氾濫シミュレーションシステムの特徴は、入力から出力までにかかる一連の処理が高速になされることである。したがって、たとえば堤防が決壊する地点を様々に変え、そのそれぞれについて浸水被害を推定することが短時間でできる。実際に、このようなシステムを用いて水害による損害リスクを総合的に診断するサービスが開始されている^{5),6)}。また、たとえばある地点で堤防が決壊しそうなときに、実際に決壊した場合に避難が必要な地域を推定するために活用できると

考えられ、河川管理者や市町村から高い関心が寄せられている。

氾濫シミュレーションシステムのユーザにとっては、短時間で高精度にシミュレーション結果が得られることが望ましい。さらに、シミュレーションに与える決壊シナリオが自由に変更できることが望ましい。ここで決壊シナリオとは、堤防決壊の発生する地点、決壊幅、流入量あるいは河川水位の時間変化を記述した情報である。細かい格子 (grid) を使えば、氾濫現象を精度良く再現できることが広く知られている。また、シミュレーションの対象とする領域を広くすれば、その領域についての任意の決壊シナリオを想定できる。ところが、細かい格子で広域を覆うとシミュレーションの処理量が増え、シミュレーションに長時間を要するようになる。

シミュレーション結果を短時間で出力することを目的として、以下の手法が提案されている。栗城ら¹⁾は、その場でシミュレートせずあらかじめシミュレートしておく手法を提案した。彼らのシステムでは、あらかじめ作成した 306 通りの決壊シナリオから 1 つの決壊シナリオをユーザが選択すると、そのシミュレーション結果が画面に出力される。安田ら²⁾は、あらかじめシミュレーション対象の領域を決め、その領域に適切

[†] 日立製作所中央研究所

Central Research Laboratory, Hitachi, Ltd.

な格子を生成する手法を提案した．彼らは那珂川下流域（茨城県）について 3,058 個の多角形からなる格子を経験的知識に基づいて作成した．そして，この格子を利用した場合，細かい格子を用いた場合と同様のシミュレーション結果を少ない処理量で計算できたことを報告した．いずれの手法も，あらかじめ決壊シナリオを決める，あるいはシミュレーション対象の領域を決めることによりその後の処理を軽減させている．

一方，近年のコンピュータの処理能力の向上により，領域を限定すれば氾濫現象をその場でシミュレートすることが可能になってきた．さらにシミュレーションに適した高精度の地形データ⁷⁾が多くの地域で取得され始めており，従来ではデータがないために不可能であった領域についてもシミュレーションが可能になってきた．これらの背景から，著者らは任意の領域についてその場でシミュレートする手法が必要と考えた．

氾濫シミュレーションシステムにおいて，浸水する領域は堤防決壊の地点周辺に限られる．したがって，広域をシミュレーションの対象とした場合，大半の領域は浸水しないことになる．浸水しない領域は計算する必要がない．そこで著者らは，計算する領域を水の流れに合わせて動的に拡大，縮小させればよいと着想した．実際に処理量の低減効果を発揮するためには，この単純な着想を計算機の特性に合わせて実装する必要がある．その計算手法を，ここでは Dynamic Domain Defining Method (Dynamic DDM) と名付けた．

本論文では氾濫シミュレーションシステムの処理量を低減させることを課題とする．その解決策として，計算手法 Dynamic DDM を提案する．まず 2 章で氾濫シミュレーションシステムの概要を説明する．続いて 3 章で Dynamic DDM を提案する．Dynamic DDM の有効性を 4 章で検証する．5 章で Dynamic DDM が有効となる条件について考察する．6 章で結論を述べる．

2. 氾濫シミュレーションシステム

2.1 システム構成

図 1 に氾濫シミュレーションシステムの構成を示す．氾濫シミュレーションシステムは GIS (Geographical Information System; 地理情報システム) と氾濫シミュレータからなる．ユーザは GIS が提供する表示画面を通じてシミュレータを操作する．図 2 に氾濫シミュレーションシステムの動作画面例を示す．ユーザが決壊する地点と決壊幅を指定すると，堤防決壊地点（画面右側）から市街地（画面左側）に向け水が流れる様子が可視化される．このように，ユーザは GIS を

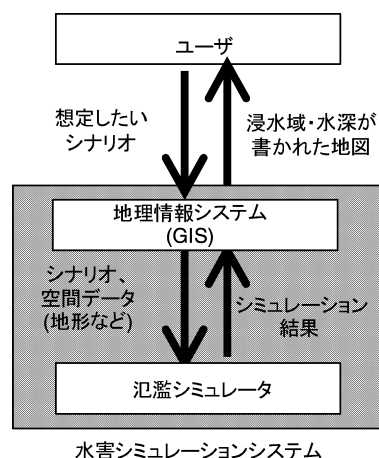


図 1 氾濫シミュレーションシステムの構成

Fig. 1 Architecture of Flood simulation system.

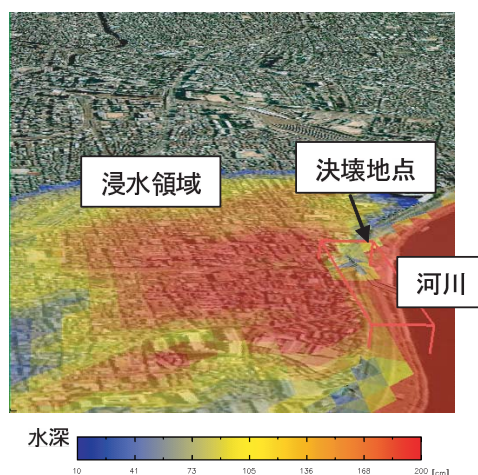


図 2 氾濫シミュレーションシステムの動作画面

Fig. 2 A screenshot of the system during user operation.

通じて決壊シナリオを入力でき，そのシミュレーション結果を短時間で得られる．図 3 は建物立体情報に水面を重ねて表示した例である．この例では，シミュレーション結果に基づき水面の高さが時間変化する．著者らのユーザテストの結果から，このアニメーションのような表示が水害の的確な認知に効果を発揮することが示唆された⁴⁾．GIS とシミュレータを連動させることで分かりやすい表示と使いやすい操作を実現できると考えられる．

2.2 シミュレータ

シミュレータは，出力する結果が実際の現象と精度良く一致することが重要である．精度に大きな影響を与える要素として，氾濫シミュレータとして用いる物



図 3 氾濫シミュレーションシステムによる都市景観表示
Fig. 3 A screenshot of the system displaying cityscape.

理モデルと格子サイズがあげられる．物理モデルとして浅水方程式を用いると高精度な結果が期待できることが知られている．たとえば，福岡ら⁸⁾は物理モデルとして浅水方程式 (shallow water equations) を用いた氾濫シミュレーション結果と，都市を模した実験装置における計測結果とを比較した．実験装置は実際の都市を 1/100 のスケールで再現した幅 1.5 m，長さ 6 m の模型であり，地形をモルタルで，家屋をスポンジで再現してある．彼らは，シミュレーションと模型実験とにおける水深分布が良好に一致したことを報告している．また，用いる格子は 50 m よりも細かい必要があると考えられる．山口・岩村⁹⁾は，2004 年 7 月に福井市で発生した水害における現地調査¹⁰⁾と，格子サイズ 10～300 m で行ったシミュレーション結果とを比較した．図 4 に格子サイズ 10 m および 100 m で行ったシミュレーションによる浸水域と，現地調査による浸水域の比較を示す．さらに，図 5 に格子サイズと浸水深の誤差の関係を示す．浸水深の誤差は，現地調査により得られた浸水深からシミュレーションにより得られた浸水深を引いた値とした．この誤差を現地調査を行った 146 地点について算出し，その RMSE (root mean square error; 2 乗平均平方根誤差) とバイアスを算出した．図 5 から，格子を 50 m より粗くすると大幅に浸水深誤差の RMSE が増大する傾向が見られる．格子を 50 m より粗くすると浸水深誤差が増大するこの傾向は，山本⁷⁾の報告でも見られる．そこで，以後本論文ではシミュレータで高い精度を実現するため，浅水方程式を用い，かつ 50 m よりも細かい格子を用いることとする．

浅水方程式を以下に示す¹¹⁾．

連続の式：



図 4 水害の現地調査¹⁰⁾とシミュレーションとの浸水域に関する比較
Fig. 4 Comparison between site investigation and simulation result about flood area.

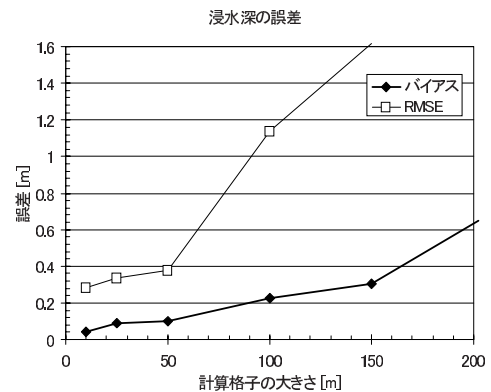


図 5 水害の現地調査¹⁰⁾とシミュレーションとの浸水深に関する比較
Fig. 5 Comparison between site investigation and simulation result about flood area.

$$\frac{\partial h}{\partial t} + \frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} = q \quad (1)$$

x 軸方向の運動方程式：

$$\frac{\partial M}{\partial t} = -gh \frac{\partial H}{\partial x} - \frac{gn^2}{h^{4/3}} M \sqrt{u^2 + v^2} \quad (2)$$

y 軸方向の運動方程式：

$$\frac{\partial N}{\partial t} = -gh \frac{\partial H}{\partial y} - \frac{gn^2}{h^{4/3}} N \sqrt{u^2 + v^2} \quad (3)$$

ただし， h ：水深， H ：水位 ($H = h + h_g$ ， h_g ：標高)， M ， N ：それぞれ x ， y 方向の単位幅あたりの流量， u ， v ：それぞれ x ， y 方向の流速 ($u = M/h$ ， $v = N/h$)， g ：重力加速度， n ：マンニングの粗度係数， q ：単位時間・単位面積あたりの流入量である．マンニングの粗度係数 (Manning's roughness coefficient) とは壁面や底面の表面特性を表す量で，粗度が大きいほど流れに働く摩擦力が大きくなる¹²⁾．

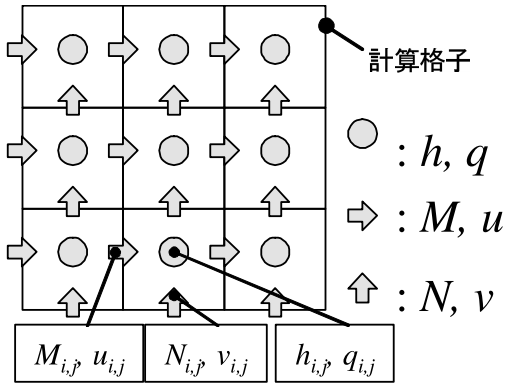


図 6 格子と変数の配置

Fig. 6 Arrangement of grids and variables.

浅水方程式 (1) ~ (3) を離散化する．さらにこの離散化した式を式変形し，ある時刻の水深 h と流速 u , v を代入すると，微小な時間 Δt 後の水深 h^* と流速 u^* , v^* が算出できる形式にする．離散化にはリーブフロッグ法 (leapfrog method), スタッガードグリッド (staggered grid) を用いた¹³⁾．変数の配置を図 6 に示す．スタッガードグリッドを採用したため， $h_{i,j}$, $q_{i,j}$ に対し $u_{i,j}$, $M_{i,j}$ は $-\Delta x/2$ ずれて， $v_{i,j}$, $N_{i,j}$ は $-\Delta y/2$ ずれて配置される．決壊シナリオは浅水方程式では q または h により表現される．堤防決壊の地点および幅は格子 (i,j) に変換され，流入量は q に，河川水位は h に変換される．

離散化した浅水方程式を以下に示す．

連続の式：

$$h_{i,j}^* = h_{i,j} + q_{i,j} \Delta t - \Delta t \left(\frac{M_{i+1,j} - M_{i,j}}{\Delta x} + \frac{N_{i,j+1} - N_{i,j}}{\Delta y} \right) \quad (4)$$

x 軸方向の運動方程式：

$$M_{i,j}^* = \frac{1}{1 + a_{i,j}} \left((1 - a_{i,j}) M_{i,j} - \Delta t g h_m^* \frac{H_{i,j}^* - H_{i-1,j}^*}{\Delta x} \right) \quad (5)$$

$$u_{i,j}^* = M_{i,j}^* / h_m^* \quad (6)$$

ただし，

$$a_{i,j} = \frac{\Delta t g n_m^2}{2 h_m^{*4/3}} \sqrt{u_{i,j}^2 + v_{i,j}^2} \quad (7)$$

であり， h_m , n_m , v_m はそれぞれ以下のように算出した．

$$h_m = \frac{h_{i,j} + h_{i-1,j}}{2} \quad (8)$$

$$n_m = \frac{n_{i,j} + n_{i-1,j}}{2} \quad (9)$$

$$v_m = \frac{v_{i,j} + v_{i-1,j} + v_{i,j+1} + v_{i-1,j+1}}{4} \quad (10)$$

なお，式 (4) および式 (5) の空間差分は，スタッガードグリッドにおける 1 次精度差分である． y 軸方向の運動方程式についても x 軸方向の運動方程式と同様に導出した．

本論文で用いた離散化方法は陽解法 (explicit method) であるので，計算を発散させないために CFL (Courant-Friedrichs-Lewy) 条件を満たす必要がある．CFL 条件とは，時間 Δt の間に現象が伝わる距離は 1 格子未満でなければならない，という条件である．ここでは，

$$\Delta t < \frac{1}{U_{i,j}} \frac{\Delta x \Delta y}{\sqrt{\Delta x^2 + \Delta y^2}} \quad (11)$$

$$U_{i,j} = 2\sqrt{gh_{i,j}} + \sqrt{u_{i,j}^2 + v_{i,j}^2} \quad (12)$$

をすべての格子 (i,j) について満たす値を Δt に設定した．

浅水方程式において水深 h が 0 またはごく小さい場合，流速 u , v に大きい誤差が生じることが知られている．これは， h を分母に持つ運動方程式 (2), (3) 右辺第 2 項である摩擦項が，ごく小さい水深の場合に大きな誤差を含むためである．これを防ぐため，水深がごく小さい値 h_ε よりも小さい箇所は流速を 0 にする方法が一般的にとられている^{14)~16)}．ここでは変化に富む地形に対応できるように Ying ら¹⁶⁾ にならい $h_\varepsilon = 10^{-3} \text{ m}$ とし，これ未満の水深の格子では流速 u , v を 0 にした．

離散化した浅水方程式は，必ずしもすべての格子で解く必要がない． h^* , u^* , v^* を求める式 (4) ~ (6) の右辺にある変数 h , u , v , q のすべてが 0 の場合，つねに $h_{i,j}^* = 0$, $(u_{i,j}^*, v_{i,j}^*) = (0, 0)$ である．したがって，格子 (i,j) における変数 h , u , v , q と，その格子隣接する 4 格子における変数 h , u , v のすべてが 0 である場合，浅水方程式を解く必要がない．逆に，格子 (i,j) における変数 h , u , v , q と，その格子隣接する 4 格子における変数 h , u , v のいずれかが 0 でない場合， h^* , u^* , v^* が 0 でない可能性がある．この場合は浅水方程式を解く必要がある．浅水方程式を解く解く必要がある格子を「浸水格子」，解く必要のない格子を「非浸水格子」とすると，まずその格子が浸水格子であるか非浸水格子であるかを判別し，浸水格子の場合のみ浅水方程式を解けばよい．通常シミュレーションの開始時にはほとんどの格子が非浸水格子であるため，この処理により特にシミュレーション開始直後の処理量を抑制できる．

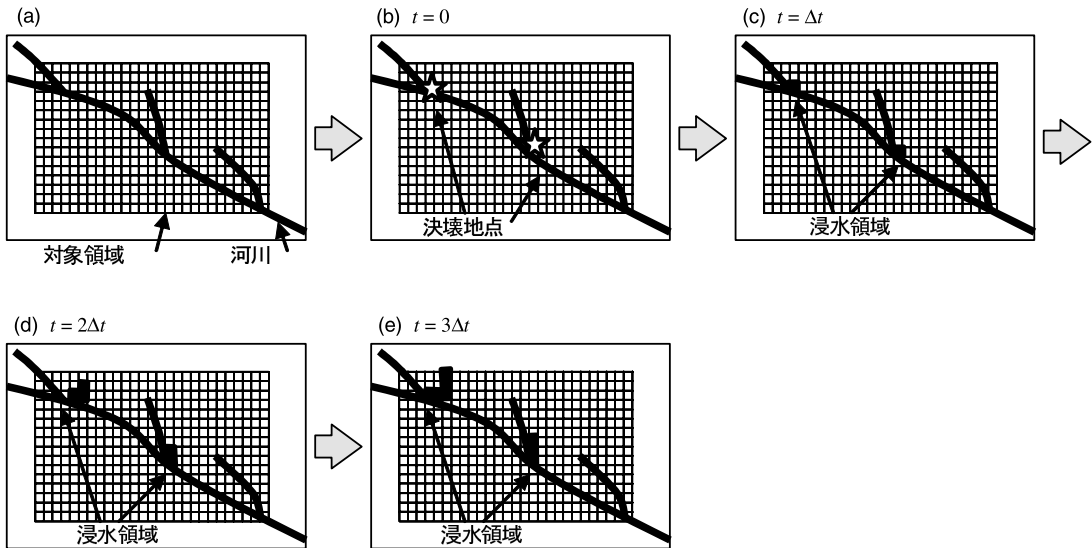


図 7 従来のシミュレーション手法の処理の流れ

Fig. 7 Schematic view of time advancing process using conventional method.

3. 高速計算手法

3.1 従来手法

シミュレーションでは、離散化した浅水方程式 (4) ~ (6) を繰り返し解き任意の時刻の水深や流速を得る。この処理をここでは時間発展処理と呼ぶ。まず、従来の時間発展処理を模式図 (図 7) により説明する。シミュレーション開始前にシミュレーション対象とする領域に格子を配置する (図 7(a))。決壊シナリオが入力されると、システムはシミュレーションを開始する (図 7(b))。開始後は、対象領域内のすべての格子について浸水格子・非浸水格子のどちらであるかを判別し、浸水格子の場合のみ浅水方程式を解く。シミュレーション開始から時間 $t = \Delta t$ だけ経過した状態を (図 7(c)) に示す。同様に、 $t = 2\Delta t$ の状態を図 7(d) に、 $t = 3\Delta t$ の状態を図 7(e) に示す。このように時間進行処理を繰り返すことにより、任意の時刻における氾濫流の状態を算出する。

図 8 は従来の時間発展処理のフローチャートである。ステップ A1 では、浸水する可能性の有無を判別する。浸水する可能性がある場合はステップ A2 で浅水方程式を解く。

従来の時間発展処理の処理量 C_c は以下の式で表現できると考えられる。

$$C_c = A_1 N_{\text{total}} + A_2 N_{\text{wet}} \quad (13)$$

ただし、 N_{total} : シミュレーション対象の全格子数、 N_{wet} : 浸水格子数、 A_k : 図 8 に示したステップ A_k の処理量である。ここで、 $N_{\text{wet}} \leq N_{\text{total}}$ である。浸

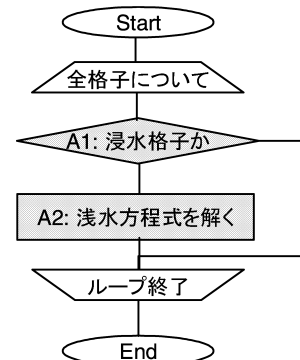


図 8 従来のシミュレーション手法の時間発展処理 (フローチャート)

Fig. 8 Flowchart of time advancing process using conventional method.

水する領域があらかじめ分かっているならば、シミュレーション対象の領域を浸水領域に限定し $N_{\text{total}} \simeq N_{\text{wet}}$ とできる。従来の時間発展処理の処理量 C_c は、 N_{total} を最小にすることで以下で示される理想値 C_{min} に近づくといいえる。

$$C_{\text{min}} = (A_1 + A_2) N_{\text{wet}} \quad (14)$$

ここで、浸水格子数を計算対象となる格子数で割った値 $R_{\text{fill}} = N_{\text{wet}}/N_{\text{total}}$ を充填率と定義する。この充填率が 1 に近いほど処理量 C_c は理想値 C_{min} に近づき、0 に近いほど処理量 C_c と理想値 C_{min} との差が大きくなる。

3.2 従来手法の問題点

従来手法では、図 7(a) に示すとおりシミュレーショ

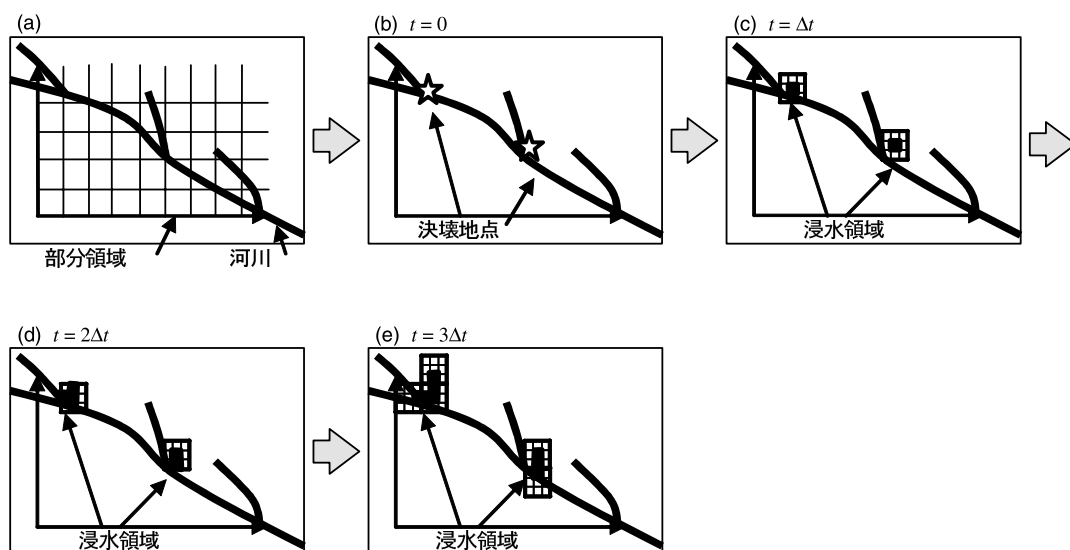


図9 提案手法 Dynamic DDM の処理の流れ

Fig. 9 Schematic view of time advancing process using Dynamic DDM.

ン前にあらかじめシミュレーションの対象領域を決定する必要がある。この対象領域が浸水する領域に比べ広い場合、充填率が0に近い場合処理量が多い。処理量を小さくするためには、浸水する領域をあらかじめ把握しておき浸水する領域と同じ領域を対象領域に設定すればよい。

ところが一般に、シミュレートする前に浸水する領域が把握できていることは少ない。たとえば、システムに複数地点が時間差を持って決壊するシナリオが与えられた場合を想定する。このようなシナリオには、決壊する地点の位置、決壊する順番、決壊地点からの流入量等、無限の組合せが存在する。あらゆる決壊シナリオについて浸水する領域をあらかじめ把握しておくことは非常に困難であるといえる。

そこで従来は、任意の決壊シナリオが与えられた場合でも浸水領域が確実に含まれるような広大な領域を計算していた。たとえば、横塚³⁾は面積533 km²の広大な領域を計算領域に設定している(日本国内では浸水面積が1~100 km²程度の水害が多い)。従来手法では充填率 R_{fill} が0に近い場合処理量が多いといえる。

3.3 提案手法

氾濫シミュレータの高速化のためには、充填率 R_{fill} を自動的に1に近づける手法が有効であると考えられる。充填率 R_{fill} を1に近づけるためには、ある格子が浸水格子であるか非浸水格子であるかを検査する必要がある。この検査の処理量が、充填率 R_{fill} を1に近づけることで削減される処理量よりも小さいことが

重要である。検査の処理量を削減するため、提案手法では検査を格子ごとではなく複数の格子からなる部分領域(subdomain)ごとに行う。

提案手法の処理の流れを模式図(図9)により説明する。まず、シミュレーション対象領域をあらかじめいくつかの部分領域に分割しておく(図9(a))。決壊シナリオが入力される(図9(b))と、その決壊地点を含む部分領域を計算対象とする。計算対象とされた部分領域に限定して従来手法と同様に流体方程式を解く。図9(c)に、計算開始から Δt 秒経過した状態を示す。しばらくすると部分領域の外側に水が流出する(図9(d))。この場合、新たに流入される側の部分領域も計算対象にする(図9(e))。逆に、部分領域内のすべての格子が非浸水格子になる可能性がある。この場合、その部分領域を計算対象でなくする。提案手法の特徴は、シミュレーション実行中に計算領域が部分領域単位で拡大、縮小することである。

図10に提案手法の処理の流れを示す。ステップB1では、決壊シナリオを参照し、決壊地点に該当する格子を含む部分領域を計算対象とする。続いて、計算対象とした部分領域について時間発展処理(ステップA1~A2)を行う。ステップB2では、浸水格子を含む部分領域を計算対象とし、ステップB3では非浸水格子のみからなる部分領域を計算対象でなくする。提案手法は、従来手法のフローチャート図8にステップB1~B3を追加した手法といえる。

ステップB1では、決壊シナリオを参照し、決壊シナリオにより変数 q あるいは h が変更される格子

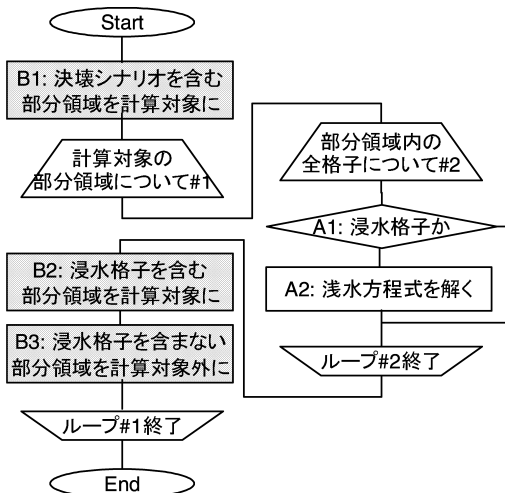


図 10 提案手法 Dynamic DDM の時間発展処理 (フローチャート)

Fig. 10 Flowchart of time advancing process using Dynamic DDM.

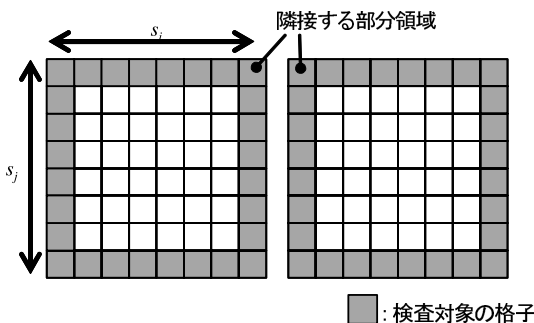


図 11 部分領域と格子の配置

Fig. 11 Arrangement of subdomains and grids.

(i, j) を求める。次に、この格子 (i, j) を含む部分領域 (b_i, b_j) を計算対象とする。本論文では、図 11 に示すように東西に s_i 格子、南北に s_j 格子からなる均一な大きさの部分領域を配置した。このとき格子 (i, j) を含む部分領域 (b_i, b_j) は

$$b_i = \text{floor}(i/s_i), b_j = \text{floor}(j/s_j) \quad (15)$$

により特定できる。ただし $\text{floor}()$ は小数部分を切り捨て整数にする関数である。

ステップ B2 では、ステップ A2 の浸水格子判別方法を応用し、浸水部分領域を判別する。ステップ A2 では、ある格子の変数 h, u, v が 0 であるかを検査し、0 でない場合はその格子と、その格子に隣接する 4 格子の合計 5 格子が浸水格子であるとした。これと同様にステップ B2 では、部分領域の最も外側の格子についてステップ A2 と同じ検査をする。検査対象の格子を図 11 に示す。検査対象の格子はいずれも、隣

接する 4 格子のうち 1 格子または 2 格子が別の部分領域に含まれる格子である。これらの検査対象の格子におけるいずれかの変数が 0 でない場合、隣接する部分領域を計算対象とする。

ステップ B3 では、検査対象の部分領域について、内部に 1 格子以上の浸水格子を含むか否かをチェックする。浸水格子を 1 格子も含まない場合、その部分領域を計算対象でなくする。具体的には、部分領域内の格子における変数 h, u, v, q 、および部分領域に隣接する格子における変数 h, u, v のすべてが 0 の場合に、その部分領域を計算対象でなくする。

浸水格子を含む部分領域については、従来手法と同じ時間発展処理 (ステップ A1 ~ A2) により算出されるため、提案手法と従来手法とで同じ結果が得られる。一方、浸水格子を含まない部分領域についてはステップ B1 ~ B3 により計算対象外になるため、提案手法ではこの部分領域についてのシミュレーション結果を得ることはできない。ただし、浸水格子を含まない部分領域では、すべての格子で変数 (h, u, v) の値が 0 であることが明らかである。したがって、浸水格子を含む部分領域についてはシミュレーションで得られた値を、浸水格子を含まない部分領域については $h = 0, (u, v) = (0, 0)$ を代入すれば、提案手法でも従来手法と同じ結果が得られる。提案手法は従来手法と同じ結果を少ない処理量で出力できるといえる。

提案手法の処理量を推定する。ステップ B1 の処理量は決壊シナリオ中の決壊地点数と式 (15) を実行する処理量の積であり、無視できる程度の大きさである。ステップ B2 の処理量は、計算対象になっている部分領域数 N_{sub} と部分領域内の検査対象の格子数 N_{mon} の積に、ステップ B3 の処理量は N_{sub} と部分領域内の全格子数 N_{in} の積に比例する。したがって、提案手法の処理量 C_p は以下の式で表現できると考えられる。

$$C_p = A_2 N_{\text{wet}} + B_2 N_{\text{mon}} N_{\text{sub}} + (A_1 + B_3) N_{\text{in}} N_{\text{sub}} \quad (16)$$

ただし、 B_k : 図 10 ステップ Bk の処理量である。この式には、 N_{total} に依存する項がない。シミュレーション対象とする領域の大きさとは無関係に処理量が決まるといえる。

4. 検証実験

4.1 実験方法

提案手法が処理量を低減させる効果があることを検証するため、3 種類の実験 (実験 1 ~ 3) を行った。実験では、計測用の計算機においてシミュレーションにかかった計算時間を処理量とした。氾濫シミュレーション

表 1 計測に使用した計算機の仕様

Table 1 Spec of Computer used in the experiments.

CPU	Intel® Pentium® M Processor 1.86 GHz
OS	Microsoft® Windows® XP Professional
チップセット	Mobile Intel® 915 PM Express Chipset
メモリ	1024 MB DDR2 SDRAM Memory
ビデオカード	NVIDIA Quadro® FX Go1400

ンシステムが実際に使われる環境を想定し、計測には市販されているノート PC を使用した。表 1 に計測用の計算機の仕様を示す。

実験 1 では、実際の氾濫シミュレーションシステムにおける提案手法の有効性を検証した。ユーザが氾濫シミュレーションシステムを使用する際、図 2 および図 3 のような 3 次元画像をシミュレーション実行中に表示する機能、およびシミュレーション結果をハードディスク等に保存する機能等は必ず使用されると考えられる。これらの機能を利用した場合、本論文で高速化の対象とした氾濫シミュレータの時間発展処理のほかにも多くの処理が発生する。その場合でも、提案手法が有効であるかを検証した。従来手法、提案手法どちらの計算時間も決壊シナリオに依存するので、一般的な傾向を知るために 3 つの異なる決壊シナリオに対して計算時間を計測した。すべての決壊シナリオで、過去に実際に決壊した 1 地点において堤防が幅 50 m にわたり決壊した場合を想定し、堤防決壊から 10 時間後までのシミュレーションにかかる時間を測定した。決壊シナリオ 1 では、決壊時刻から 10 時間後まで、河川水位が周囲の標高より 0.5 m 高かったとした。同様に、決壊シナリオ 2 では河川水位が 1.0 m、決壊シナリオ 3 では 2.0 m 高かったとした。市町村スケールの水害に強い関心を持っているユーザを想定し、比較する従来手法の格子数を $N_{\text{total}} = 250,000$ とした。実験で用いた格子サイズ $\Delta x, \Delta y$ が 50 m であるので、25 km 四方 (625 km²) に相当する。地形は国土地理院発行の「数値地図 50 m 標高」を用い、粗度は水田や畑地の代表的な値 ($n=0.03$) を全域に割り当てた。計算時間はシステムが実際に使用される環境を設定して計測した；シミュレーション内の時間で 10 分ごとの途中結果を画面に 3 次元画像として出力し、結果データをハードディスクへと格納した。開始から終了までの時間が 10 分を超える場合もあったため、ストップウォッチを使って計算時間を求めた。

実験 2 および実験 3 では、本論文で高速化の対象とした氾濫シミュレータの時間発展処理のみについて、従来手法と提案手法のそれぞれの計算時間を計測した。実験 2 では、従来手法の格子数 N_{total} に対す

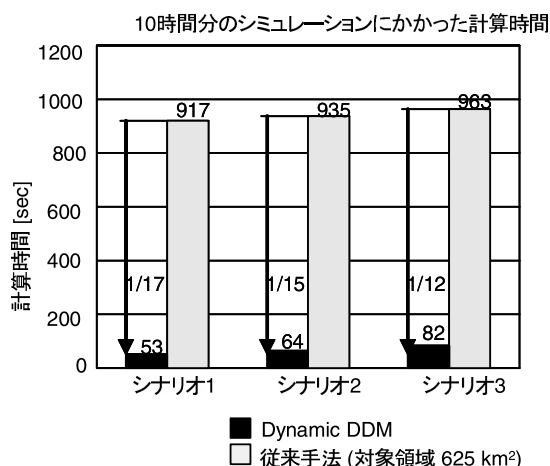


図 12 シナリオごとの計算時間

Fig. 12 Calculation time of each scenario.

る処理量の変化を調査した。格子数 N_{total} を 40,000, 90,000, 160,000, 250,000 の 4 種類に設定し 1 回の時間発展処理にかかる計算時間を計測した。計算領域の形状は、東西、南北に同数の格子を配置した正方形としたため、それぞれ 1 辺 200, 300, 400, 500 格子からなる計算領域を設定したことになる。計測には、1 点から時間一定の流量で水が流入する決壊シナリオを用いた。流入地点をシミュレーション対象領域の中心に設定した。用いた地形は完全に平坦とした。計算時間は理想的な環境設定で計測した；シミュレーション実行中は、画面出力もハードディスクへのアクセスもしないようにした。1 回の時間発展処理が $10^{-2} \sim 10^{-4}$ 秒程度であったため、実行に費やした CPU クロックサイクル数を CPU 周波数で割ることにより計算時間を求めた。

同様に、実験 3 では提案手法の部分領域の格子数 N_{in} に対する処理量の変化を調査した。部分領域の格子数 N_{in} を 256, 1024, 4096 の 4 種類に設定し 1 回の時間発展処理にかかる計算時間を計測した。部分領域の形状は、東西、南北に同数の格子を配置した正方形としたため、それぞれ 1 辺 16, 32, 64 格子からなる部分領域を設定したことになる。用いた決壊シナリオ、地形、計算時間の計測等は実験 2 と同様にした。

4.2 実験結果

実験 1 の結果を図 12 に示す。従来手法が $N_{\text{total}} = 250,000$ の場合、提案手法はその 1/12 ~ 1/17、平均 1/14 の処理量になった。決壊シナリオ 1 ~ 3 では破堤 10 時間後の浸水面積がそれぞれ 17.7 km², 28.7 km², 42.4 km² になった。実験で推定したシナリオは日本国内における中程度の水害を想定したことになる。提案

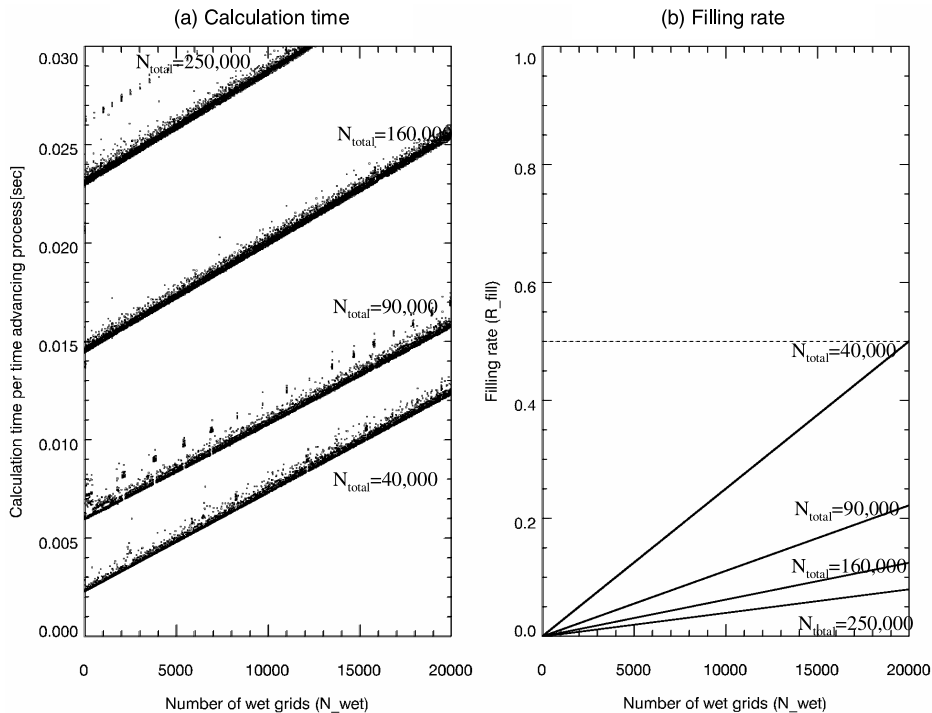


図 13 実験 2 の測定結果。(a) 時間発展処理 1 回あたりの計算時間, (b) 充填率
Fig. 13 Result of Experiment 2. (a) Calculation time, (b) Filling rate.

手法と従来手法の計算時間の比 (Dynamic DDM の計算時間: 従来手法の計算時間) は, シナリオ 1 では 1:17, シナリオ 3 では 1:12 であり, シナリオ 1 の方が比が大きい. 浸水面積が小さいほどコスト低減効果が高かったといえる. 実験で用いた格子サイズ Δx , Δy が 50m であるので, それぞれのシナリオで 10 時間後に浸水格子数 N_{wet} はそれぞれ 7,100, 11,500, 17,000 であったことになる. また, 10 時間分のシミュレーションを $\Delta t = 5 \text{ sec}$ で行ったため時間発展処理は 7,200 回行ったことになる. なお, このシミュレーションにおいてはつねに CFL 条件 (式 (11)) が満たされていることを確認した. さらに, 従来手法と提案手法とで同等の結果が得られていることをシステムの表示画面により確認した.

実験 2 の結果を図 13 に示す. 図 13 (a) は計算時間に関する結果であり, 横軸に浸水格子数 N_{wet} を, 縦軸に 1 回の時間発展処理にかかった計算時間をとった. 図 13 (b) は充填率 R_{fill} に関する結果であり, 横軸は図 13 (a) と同じであり, 縦軸に充填率 R_{fill} をとった. 従来手法では, N_{wet} が同じ場合格子数 N_{total} が大きいほど処理量が大きかった. また, N_{wet} が 0 に近いときでも無視できない計算時間がかかっていた. また, 充填率 R_{fill} は定義から明らかなように浸水格子

数 N_{wet} に比例して直線的に増大した.

実験 3 の結果を図 14 に示す. 凡例は図 13 と同じである. 図 14 (a) に示すように, 提案手法では N_{wet} が 0 に近いとき計算時間は 0 に近かった. また, N_{wet} が 10,000 格子以上のとき, $N_{in} = 1024$ の場合の計算時間が最小であった. 図 14 (b) に示すように, 充填率は N_{wet} が大きくなるにつれ増大する傾向を示し, 5,000 格子より大きい場合は N_{in} が小さいほど充填率が高かった. たとえば N_{wet} が 10,000 格子の場合, 部分領域のサイズ N_{in} が 256 のときは充填率 0.8, 1,024 のときは充填率 0.4, 4,096 のときは充填率 0.2 であった.

5. 考 察

Dynamic DDM は, 従来手法に比べ処理量が小さかった. 途中結果を画面に 3 次元画像として出力する等, 実利用を想定した場合の実験 (実験 1) では, 従来手法に比べ 1/12 ~ 1/17, 平均 1/14 の処理量であった. 氾濫シミュレーションシステムに提案手法 Dynamic DDM を適用することで, 処理量を低減できると考えられる.

実験 3 の結果から, 式 (16) を近似式として最小 2 乗法により係数を求めた. 近似式の係数を表 2 に示す. 近似式と計測結果との RMSE は 0.00027 と, 1 回の

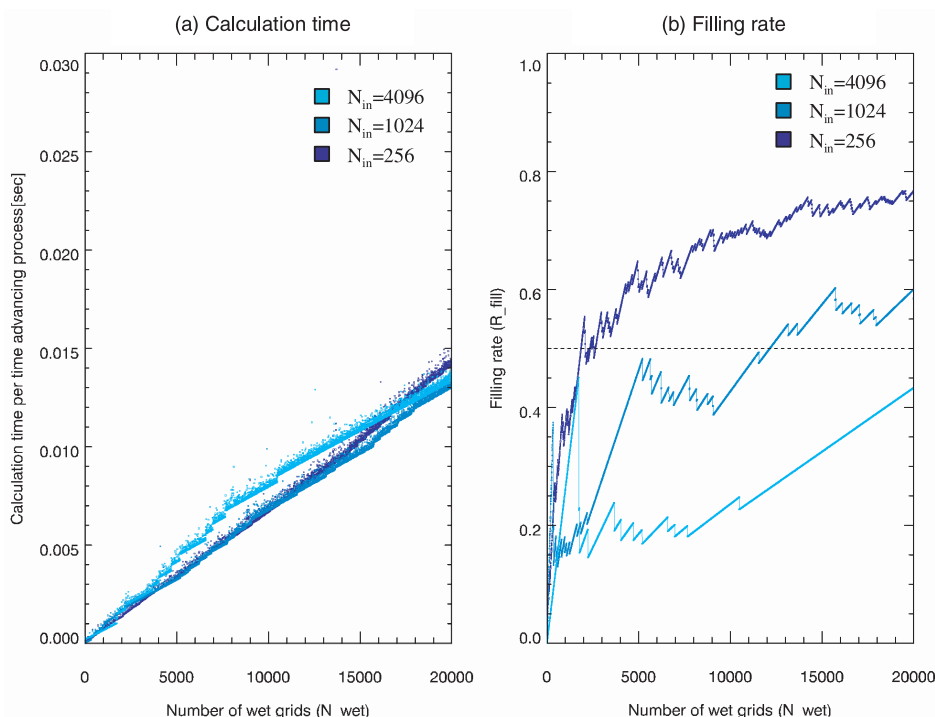


図 14 実験 3 の測定結果 . (a) 時間発展処理 1 回あたりの計算時間, (b) 充填率
Fig. 14 Result of Experiment 3. (a) Calculation time, (b) Filling rate.

表 2 提案手法の処理量の係数

Table 2 Coefficient of Dynamic DDM's calculation cost.

処理内容	格子あたり計算時間 (秒)
B_2	6.7×10^{-7}
$A_1 + B_3$	4.9×10^{-8}
A_2	5.3×10^{-7}

時間発展処理にかかった時間と比較して十分小さかった。これより, Dynamic DDM の処理量は式 (16) により精度良く見積もることができるといえる。

同じ浸水領域を処理する場合, 従来手法では充填率 R_{fill} が 1 に近いことが望ましい。Dynamic DDM で充填率 R_{fill} を 1 に近づけるためには, 図 14 (b) に示されたとおり部分領域サイズ N_{in} を小さくすればよい。ところが表 2 から, 検査のための処理量 B_2 は, 非浸水格子を計算対象に含む場合に発生する処理量 $A_1 + B_3$ の 14 倍以上であることが分かる。検査の処理量を抑えつつ充填率 R_{fill} を高められるよう, 部分領域のサイズ N_{in} を適切に選ぶことが重要である。図 14 (b) から, Dynamic DDM では充填率が 0.5 程度のときの処理量が最も少ない。日本国内で多く発生する浸水面積が $1 \sim 100 \text{ km}^2$ (浸水格子数 $400 \sim 40,000$) 程度の水害をシミュレートする場合は, 部分領域のサイズ N_{in} が 1024 が最適であると考えられる。

一般的に, 氾濫シミュレーションシステムのユーザは浸水領域が時間とともに拡大する過程をシミュレートしたいと考えており, かつシミュレーションを行う前に浸水領域かが分かっていることはまれである。浸水領域があらかじめ分かっていない場合, 従来手法では計算領域 N_{total} を広大な領域に設定するため, 充填率 R_{fill} が低い。また, 一般的にシミュレーション中に浸水領域が変化するため, 充填率が時間変化する。通常, シミュレーションの初期では浸水している領域がほとんどないため充填率 R_{fill} が低く, 時間経過とともに浸水領域が拡大し充填率が高くなる。充填率をつねに高い状態に維持することは, 従来手法では困難であるといえる。これに対し Dynamic DDM は計算対象となる領域が自動的に決定され, 充填率が高い状態に維持される。実験結果から, 浅水方程式に Dynamic DDM を適用した場合は充填率 0.5 程度のとき処理量は最小になった。充填率が 0.5 程度になる適切な部分領域サイズ N_{in} を設定することで, Dynamic DDM は氾濫シミュレーションシステムの高速化に寄与するといえる。

Dynamic DDM の計算対象となる領域が自動的に決定される特徴を利用すると, 水害を手軽にシミュレートできるシステムが作成できる。Dynamic DDM を

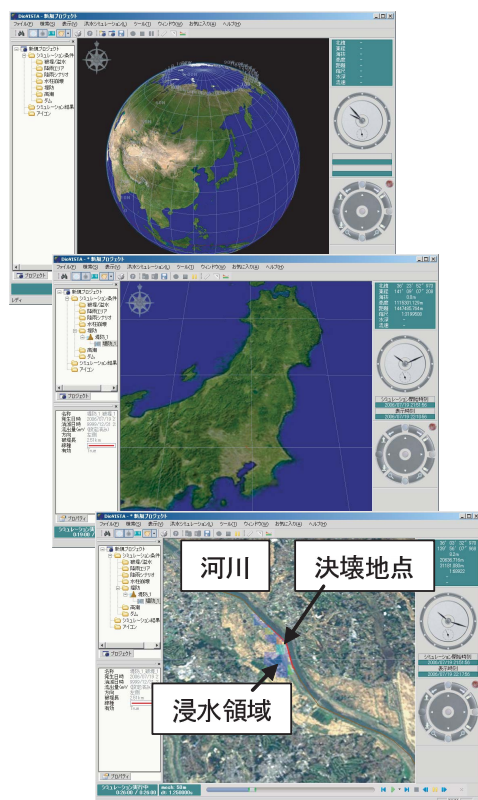


図 15 地球上の任意の場所でシミュレーション可能なシステムの動作画面

Fig. 15 Screenshot of global flood simulator.

利用して作製したシステムの動作画面を図 15 に示す．従来手法と異なりこのシステムではユーザが計算領域 N_{total} を入力する必要がない．決壊シナリオを入力するだけで地球上の任意の地点における水害をシミュレートできる．また，1 つの決壊シナリオのシミュレーションに要する時間は数分である．シミュレーションは従来，シナリオを想定し，必要なデータを収集し，シミュレーションを実行し，結果を可視化し，結果を検証するという手続きがなされてきた．Dynamic DDM を利用すると，シナリオの想定と結果の検証の中間にある手続きを自動化できる．このシステムで用いた地形データソースを表 3 に示す．地形データソースごとに得られる精度は異なるため，結果の利用には注意を要する．たとえば「数値地図 5 m メッシュ」の地形データによるシミュレーション結果は高精度（たとえば図 4）であると考えられる．これに対し，「SRTM topography」の地形データによるシミュレーション結果は少なくとも鉛直方向に 1 m，水平方向に 90 m を超える精度はないと考えられる．今後シミュレーション結果の検証データをデータソースごとに蓄積するこ

表 3 システムで用いた標高データ
Table 3 Topography data used in the system.

データソース	作成者	範囲	精度
SRTM	NASA	全球	水平 90 m
Topography			鉛直 1 m
数値地図	国土地理院	日本全国	水平 50 m
50 m メッシュ			鉛直 1 m
数値地図	国土地理院	大都市圏	水平 5 m
5 m メッシュ			鉛直 0.15 m

とで，このような広域を対象とするシミュレーションシステムが実利用されていくと考えられる．

Dynamic DDM は，水害に限らず，任意の地点から発生する可能性があり，かつ周囲に伝播する現象のシミュレーションに適用できると考えられる．たとえば，室ら¹⁷⁾の火災延焼シミュレーションに適用可能である．彼らのシミュレーションは，都市空間を微小な立方体格子で表現し，延焼が格子単位で進展していくセルオートマトンモデルを用いている．Dynamic DDM を適用するためには，シミュレーション対象領域をあらかじめ部分領域に分割しておけばよい．ただし，本研究のモデルは 1 回の時間発展処理で浸水領域が 1 格子以上拡大することを許していないのに対し，彼らのモデルは 1 回の時間発展処理で複数格子先まで延焼することを許しているため，最も外側の格子だけでなくさらに内側の格子も検査対象とする必要がある．火災延焼と同様に，Dynamic DDM は伝染病の蔓延等のシミュレーションにも適用可能と考えられる．

6. 結 論

本論文では，氾濫シミュレーションシステムの処理量を低減させる計算手法 Dynamic DDM を提案し，その有効性を検証した．

Dynamic DDM は，計算する領域を水の流れに合わせて部分領域単位で動的に拡大，縮小させる手法である．途中結果を画面に 3 次元画像として出力する等，実利用を想定した実験では，従来手法の約 1/14 の処理量で水害をシミュレートできた．Dynamic DDM は処理量を低減することが明らかになった．Dynamic DDM の処理量低減効果は部分領域の決め方に依存する．実験から，日本国内で多く発生する浸水面積が 1 ~ 100 km²（浸水格子数 400 ~ 40,000）程度の水害をシミュレートする場合は，部分領域のサイズ $N_{\text{in}} = 1024$ が最適であると考えられる．

Dynamic DDM を利用すると，地球上の任意の地点における水害を数分でシミュレートできるシステムを作成できる．このシステムはデータの収集，シミュレーションの実行，結果の可視化が自動化されている

ため、ユーザはシナリオの想定、および結果の検証に集中できる。今後シミュレーション結果の検証データを蓄積することで、このような広域を対象とするシミュレーションシステムが実用されていくと考えられる。

Dynamic DDM は、任意の地点から発生する可能性があり、かつ周囲に伝播する現象のシミュレーションに応用できると考えられる。たとえば、水害以外に、火災、伝染病等のシミュレーションに有効と考えられる。

謝辞 図 2 で用いた地図の作成にあたっては、国土地理院長の承認を得て、同院発行の数値地図 25000 (地図画像)、数値地図 2500 (空間データ基盤)、数値地図 50 m メッシュ (標高) および数値地図 5 m メッシュ (標高) を使用した (承認番号平 17 総使、第 205 号)。図 15 は、本論文の技術を応用して株式会社日立エンジニアリング・アンド・サービスが作製したソフトウェア DioVista[®] FloodSimulator により作製した。衛星画像は NASA's Earth Observatory を利用した。

参 考 文 献

- 1) 栗城 稔, 末次忠司, 小林裕明: ハザード・シミュレータの水防・避難活動への活用, 土木技術資料, Vol.37, No.11, pp.26-31 (1995).
- 2) 安田浩保, 白土正美, 後藤智明, 山田 正: 水防活動の支援を目的とした高速演算が可能な浸水域予測モデルの開発, 土木学会論文集, No.740/II-64, pp.1-17 (2003).
- 3) 横塚尚志: 動く洪水ハザードマップについて, 平成 15 年度河川情報シンポジウム講演集, 財団法人河川情報センター, pp.5.1-5.11 (2003).
- 4) 山口悟史, 室 啓朗, 岩村一昭: 水害危険情報のわかりやすい提示を目的とした浸水シミュレータの開発, 水工学論文集, Vol.49, pp.421-426 (2005).
- 5) ニッセイ同和損害保険株式会社: 「3D 洪水シミュレーション技術」を活用した「水災リスク診断サービス」を開始 (2005). ニュースリリース.
- 6) 山口悟史, 岩村一昭, 直野 健, 高山恒一: 三次元地図を用いた高速水害シミュレーションシステム, 日立評論, Vol.88, No.4, pp.62-67 (2006).
- 7) 山本一浩: レーザ計測データ利用による氾濫解析手法の提案, 平成 17 年度管内技術研究発表会, 国土交通省近畿地方整備局 (2005).
- 8) 福岡捷二, 川島幹雄, 松永宜夫, 前内永敏: 密集市街地の氾濫流に関する研究, 土木学会論文集, No.491/II-27, pp.51-66 (1994).
- 9) 山口悟史, 岩村一昭: 2004 年足羽川洪水事例による氾濫シミュレーションの精度検証, 平成 18 年度全国大会第 61 回年次学術講演会, 土木学会 (2006).
- 10) 山本博文: 福井市街足羽川左岸および鯖江市河和田地区における浸水被害について, 平成 16 年 7 月新潟・福島, 福井豪雨災害に関する調査研究, 平成 16 年度科学研究費補助金 (特別研究促進費 (1)) 研究成果報告書, pp.121-136 (2005).
- 11) 栗城 稔, 末次忠司, 海野 仁, 田中義人, 小林裕明: 氾濫シミュレーション・マニュアル (案), 土木研究所資料第 3400 号, 土木研究所 (1996).
- 12) 水理委員会 (編): 水理公式集平成 11 年度版, 土木学会 (1999).
- 13) 岩佐義朗 (編): 数値数理学, 丸善 (1995).
- 14) 福岡捷二, 松永宜夫: 密集市街地における洪水氾濫流解析と氾濫流制御の試み, 水工学論文集, Vol.36, pp.311-316 (1992).
- 15) 川池健司, 井上和也, 林 秀樹, 戸田圭一: 都市域の氾濫解析モデルの開発, 土木学会論文集, No.698/II-58, pp.1-10 (2002).
- 16) Ying, X., Khan, A.A. and Wang, S.S.Y.: Upwind conservative scheme for the Saint Venant equation, *Journal of Hydraulic Engineering*, Vol.130, No.10, pp.977-987 (2004).
- 17) 室 啓朗, 伊藤永一, 岩村一昭: 立体都市データを用いた空間シミュレーション方式とその火災延焼予測への適用, 第 16 回研究報告, 情報処理学会数値モデル化と問題解決研究会, pp.1-6 (1997).

(平成 18 年 8 月 17 日受付)

(平成 18 年 10 月 6 日再受付)

(平成 18 年 12 月 1 日再々受付)

(平成 18 年 12 月 8 日採録)



山口 悟史

昭和 54 年生。平成 15 年東北大学大学院理学研究科地球物理学専攻修士課程修了。同年 (株) 日立製作所中央研究所入社。氾濫シミュレーションシステムの研究開発に従事。

土木学会, American Society of Civil Engineers, 日本海洋学会各会員。



岩村 一昭

昭和 34 年生。昭和 58 年早稲田大学大学院理工学研究科機械工学専攻修士課程修了。同年 (株) 日立製作所中央研究所入社。ジオマティクスの研究開発に従事。電子情報通信学会, 日本シミュレーション学会各会員。