

Java 言語を用いた河川不定流のオブジェクト指向プログラミング

宇都宮大学工学部 学生会員 アマガラン ガンバヤル
宇都宮大学工学部 正会員 池田裕一

1. はじめに

今後の水理計算の方向としては、限られた専門技術者が行う数値解析業務だけでなく、一般のユーザーにも使用でき、周辺のデータベースやインターネットなど周囲の資源（リソース）を活用したアプリケーションの構築も重要になってくるであろう。Java 言語はその際のプログラミング言語の有力な候補であり、インターネットで結ばれた大規模な分散処理系の構築から、パソコン単位のアプリケーション、そして携帯端末での小規模アプリケーションまで、多様なプラットフォームでの多彩なアプリケーションの構築をサポートする。また、最近提唱されているオブジェクト指向プログラミングもサポートしている。

オブジェクト指向プログラミングでは、プログラムを動作させたい世界がどのような「もの」(オブジェクト)で構成されていくか検討し、いくつかのオブジェクトの相互作用によりアプリケーションを動作させようというもので、人間の認識により近いアプローチが可能である。

そこで本研究では、水理学の典型的な問題である河川の 1 次元不定流に対して、オブジェクト指向プログラミングを適用し、Java 言語による実装を試みた。

2. 差分方程式

今回は、水理計算のほうはなるべく簡単にするため、河床勾配と断面特性が一様な開水路（長方形断面）において、上流端の流量波形が与えられた際の、不定流計算を考える。境界条件として、まず上流端では流量時系列を以下の式で与える。

$$Q_0(t) = Q_b + (Q_p - Q_b) \left\{ \frac{t}{t_p} \exp \left(1 - \frac{t}{t_p} \right) \right\}^C \quad \dots (1) \quad \left(\begin{array}{l} Q_b: \text{基底流量、} Q_p: \text{ピーク流量、} \\ t_p: \text{ピーク時間、} C: \text{定数} \end{array} \right)$$

そして x 方向ならびに t 方向の刻み幅をそれぞれ $\Delta x = x_L / N$, $\Delta t = T_L / M$ とし (x_L, T_L は全体の計算距離および時間), $x_i = i \Delta x$, $t_j = j \Delta t$ の各点 (格子点) における水深 H と流量 Q を求めることにする。簡単のために, $x = x_i$, $t = t_j$ における量 Y の値を Y_i^j と書くことにする。すると、差分方程式は以下ようになる。

$i = 0 ;$	$i = 1, 2, N ;$	$i = N ;$
$\begin{aligned} h_0^{j+1} &= h_0^j - F_{q0}^j \Delta t \\ Q_0^{j+1} &= Q_0(t_{j+1}) \end{aligned} \quad \dots (2)$	$\begin{aligned} h_i^{j+1} &= h_i^j - F_{qi}^j \Delta t \\ Q_i^{j+1} &= \frac{Q_i^j - (F_{pi}^j + F_{vi}^j) \Delta t}{1 + F_{fi}^j \Delta t} \end{aligned} \quad \dots (3)$	$\begin{aligned} h_N^{j+1} &= h_N^j - F_{qN}^j \Delta t \\ Q_N^{j+1} &= Q_i^j - F_{vN}^j \Delta t \end{aligned} \quad \dots (4)$

式中の F_q, F_v, F_p, F_f の具体的な式形については、紙面の都合で省略する。参考文献³⁾を参照されたい

3. アプリケーションの構築

アプリケーションを構築するに当たり、MVC アーキテクチャを適用する。つまりアプリケーション全体を河川不定流計算の根幹部分 (Model)、計算パラメータの入力ならびに計算結果の出力にかかわる部分 (View)、Model と View を連携・制御する部分 (Controller) とに分ける。そのオブジェクトの構成を図 1 に示す。

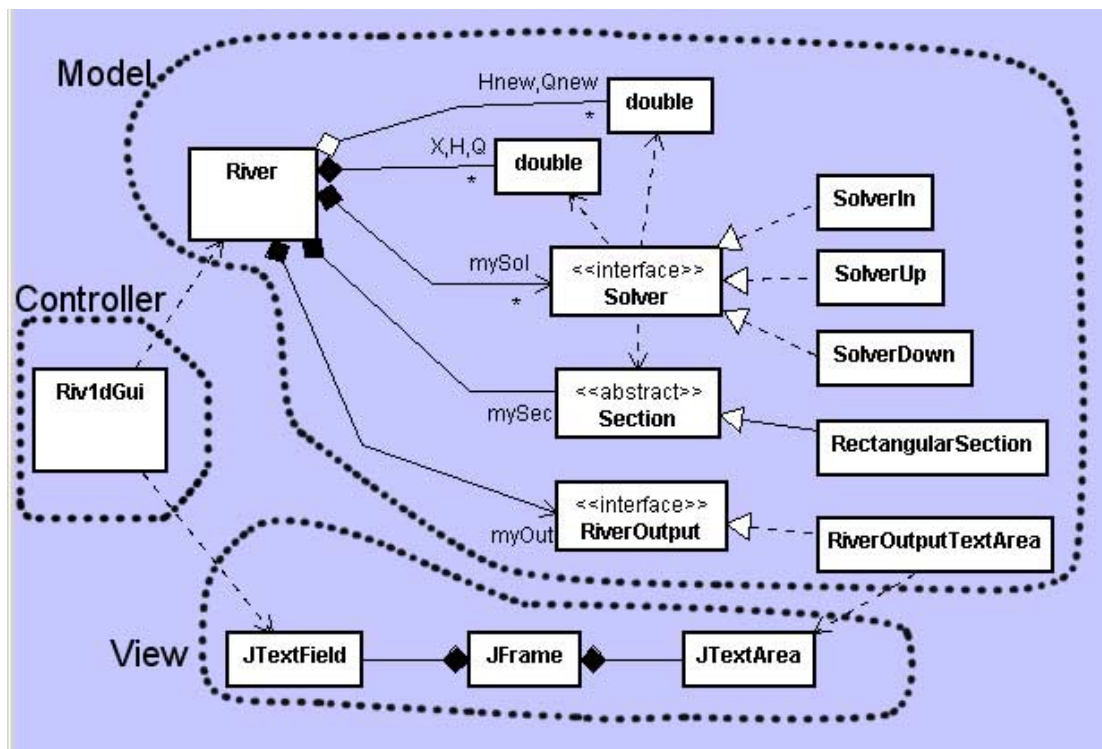


図1 GUIを用いたアプリケーションのオブジェクト構成

Model 部分は River という名のオブジェクトでまとめることにした。この River には、格子点の x 方向距離、水深、流量を格納する配列と、不定流計算を行うオブジェクト (Solver)、断面特性を計算するオブジェクト (Section)、計算結果を出力するオブジェクト (RiverOutput) から構成される。

このなかで、Solver の動作は格子点ごとによって異なるようにする必要がある (式(2)~(4)参照)。このとき Java 言語では、場合分けをして計算しない。Solver と共通のインタフェースを有する数種類のオブジェクトをつくり、それぞれの格子点の性質に応じて配置する手法をとる。今回の場合は、図 1 中の SolverUp、SolverIn、SolverDown オブジェクトでそれぞれ、上流端「式(2)」と領域内部「式(3)」と下流端「式(4)」での計算を行うようにする。これによって、「各点ごとに計算する」というプログラムを作成しても各点の性質に応じて異なる計算をさせることができる。これがオブジェクト指向プログラミングという「多態性」である。

計算結果の出力方法も同様で、RiverOutput に共通なインタフェースを有するものであれば、どのようなものも組み込むことができる。グラフ表示やファイル出力など必要に応じてオブジェクトをつくり、組み込めばよい (図 2)。

このように、River のなかでは各点の計算方法や出力方法をインタフェースとして定義すると、出力先や出力形式はインタフェースの仕様とはまったく独立していて、その実装に際してはかなり自由度が高く、River クラスの使い回しをよくしている。また、Controller 部分を変えることで、インターネット上のデータベースにアクセスして計算条件を得ることも可能となる。

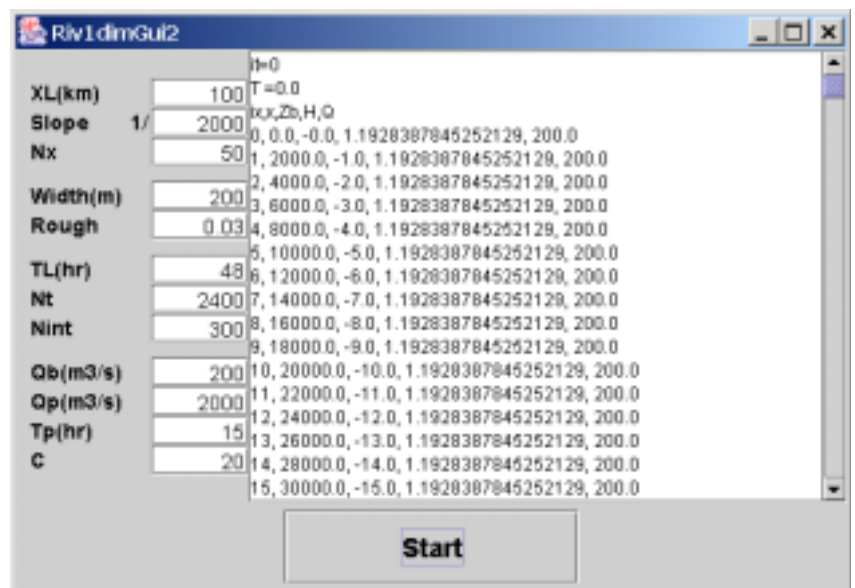


図2 GUIを用いたアプリケーションの実行例

参考文献

- 1) 高橋麻奈：やさしい Java 第 2 版，ソフトバンクパブリッシング，551p，2002．
- 2) 池田裕一：河川 1 次元不定流の計算プログラム，水理公式集プログラム例題集，土木学会水理委員会，2002．