

How to build GeoGebra from source code

藤本 光史 (福岡教育大学)

概要

GeoGebra is an interactive geometry software. The version 5 of GeoGebra has a CAS View, which is based on a computer algebra system Giac. In this paper, I will explain how to build GeoGebra from source code. Furthermore, I show a method of adding Giac's native functions to GeoGebra by changing source code.

1 GeoGebra の CAS View

GeoGebra はバージョン 5 より、CAS View が利用可能となった。CAS とは Computer Algebra System の略であり、CAS View は GeoGebra 上で数式処理システムを使って因数分解・微積分・連立方程式の求解などを可能にするものである。GeoGebra の数式処理エンジンは、バージョン 5 に至るまでにいくつかの変遷があったが、現在は Giac [3] が採用されている。

Giac は Université Grenoble Alpes の Bernard Parisse によって開発されたオープンソースの数式処理システムであり、C++ ライブラリとして提供されているため、他のソフトウェアに組み込んで利用することが可能である。Java アプリケーションである GeoGebra は JNI (Java Native Interface) を利用して Giac の C++ ライブラリを呼び出している。また、Web アプリケーション版の GeoGebraWeb は (Emscripten によって JavaScript に変換された) Giac.js [4] を利用している。

Giac が提供する数式処理機能は非常に多く、GeoGebra の CAS View で利用可能なものはそのごく一部である。そこで、本稿ではソースコードから GeoGebra をビルドする方法を解説し、ソースを修正することで現在の GeoGebra に未実装の Giac 関数を追加する方法について述べたい。

2 GeoGebra のビルドに必要なツール

GeoGebra をソースコードからビルドするには、いくつかのツールが必要である。ここではそれらについて説明する。

2.1 JDK

GeoGebra 自体が Java で開発されているため、Java SE Development Kit(JDK) を必要とする。最新のソースコードのビルドには、JDK8 が推奨されている。

2.2 Eclipse

Eclipse は様々なプログラミング言語に対応したオープンソースの統合開発環境であり、Java アプリケーションの開発に広く利用されている。GeoGebra も Eclipse での開発を前提としている。本稿では Eclipse 4.6 を利用する。

2.3 Google Web Toolkit SDK

Google Web Toolkit は Google による Java を用いた Web アプリケーションフレームワークである。このツールを用いることで、Java で記述された GeoGebra のソースから iOS や Android で動く GeoGebraWeb を生成できる。インストール方法は以下の通りである。

1. Eclipse のメニューから [Help] → [Install New Software] を選択する。
2. [Work with:] のボックスに <https://dl.google.com/eclipse/plugin/4.6> を入力する。
3. 表示されたアイテムから “Google Plugin for Eclipse 4.4/4.5/4.6” と “Google Web Toolkit SDK 2.7.0” を選択してインストールする。

2.4 Subclipse

GeoGebra のソースコードはバージョン管理システム Subversion で管理されている。Subclipse は Eclipse 用の Subversion クライアントプラグインである。インストール方法は以下の通りである。

1. Eclipse のメニューから [Help] → [Eclipse Marketplace] を選択する。
2. [Find:] のボックスに **Subclipse** を入力し、[Enter] キーを押す。
3. 表示されたアイテムから “Subclipse 1.10.13” を選択してインストールする。
4. 【Linux/macOS の場合】 Eclipse のメニューから [Window] → [Preferences] → [Team] → [SVN] を選択し、さらに [SVN interface:] の client として “SVNKit (Pure Java)” を選択する。

2.5 Gradle

GeoGebra はソースコードからビルドする際、上で挙げたもの以外に Giac などの様々な外部ツールやライブラリをダウンロードしながらビルドしていく。これらのライブラリの依存関係を管理するためにビルドツールの Gradle が利用される。インストール方法は以下の通りである。

1. Eclipse のメニューから [Help] → [Eclipse Marketplace] を選択する。
2. [Find:] のボックスに **GradleIDE** を入力し、[Enter] キーを押す。

3. 表示されたアイテムから “Gradle IDE Pack 3.8.x+1.0.x” を選択してインストールする。

3 ソースコードの取得

GeoGebra のソースコードを得る手順は少し複雑である。

1. Eclipse のメニューから [File] → [Import] → [SVN] → [Checkout Projects from SVN] を選択し、[NEXT] ボタンをクリックする。
2. [Create a new repository location] を選択して [NEXT] ボタンをクリックする。
3. [Url:] のボックスに <http://dev.geogebra.org/svn/trunk/> を入力し、[NEXT] ボタンをクリックする。
4. 表示されたアイテムから “geogebra” フォルダを選択し、[NEXT] ボタンをクリックする。
5. “Check out as a project in the workspace” を選択し、[Project name:] のボックスに **geogebra** を入力し、[Check out HEAD revision] のチェックを外してから [Revision:] のボックスに **49152** を入力して*1、[FINISH] ボタンをクリックする。

この操作で約 380MB のソースコードが workspace にダウンロードされる。

4 ソースコードのビルド

Gradle でビルドする前に、一旦、Eclipse の Package Explorer から geogebra プロジェクトを右クリックし、削除する。このとき、ディスクからプロジェクト自体を削除しないように注意する。以上の準備の後、ようやくビルドすることが可能となる。

1. Eclipse のメニューから [File] → [Import] → [Gradle (STS)] → [Gradle (STS) Project] を選択し、[NEXT] ボタンをクリックする。
2. [Root folder:] のボックスに `c:\workspace\geogebra` を入力する*2。
3. [Build Model] ボタンをクリックすると、ホームディレクトリの `.gradle` フォルダに約 220MB のファイルがダウンロードされる。
4. 表示されたアイテムから “input3D/stub” と “test” を除くすべてを選択し、[FINISH] ボタンをクリックするとビルド作業が開始される。

5 ビルドエラーの解消

ビルドの途中で以下の 2 種類のエラーが発生する場合がある。対処方法は以下の通りである。

*1 GeoGebra の HEAD リビジョンは予期しないエラーが発生する場合が多く安定性に欠ける。ここでは、2016 年 11 月 1 日時点で最も安定したリビジョンを用いる。

*2 この部分は geogebra のソースコードを実際にダウンロードしたフォルダを指定する。

- Access restriction error
“desktop” プロジェクトを右クリックし、[Properties] → [Java Compiler] → [Errors/Warnings] → [Deprecated and restricted API] を選択し、[Forbidden reference (access rules):] において [Error] を [Warning] に変更する。
- build path error
エラーが出たプロジェクトを右クリックし、[Build Path] → [Configure Build Path] を選択し、[Order and Export] タブにおいて、見つけれないライブラリ (GWT SDK や JRE) にチェックを入れる。また、[Libraries] タブにおいて、PATH に誤りのある (×マークがついた) .jar ファイルの PATH を修正する。

以上ですべてのビルドエラーを解消できる。この時点で workspace のディスクサイズは約 550MB になる。

6 GeoGebra の実行

ビルドできた GeoGebra バイナリを実行するには、以下のようにする。

6.1 デスクトップ版 GeoGebra の実行

1. “desktop” プロジェクト → src/main/java → org.geogebra.desktop にある “GeoGebra3D.java” を探す。
2. “GeoGebra3D.java” を右クリックし、[Run As] → [Java Application] を選択する。

6.2 Web アプリケーション版 GeoGebra の実行

1. Eclipse のメニューから [Run] → [Run Configuration] を選択し、[Web Application] タイプの configuration を新規に作成する。
2. [Name:] ボックスに **GeoGebraWeb** を入力し、Main タブの [Project:] に **web** を、[Main class:] に **com.google.gwt.dev.DevMode** を入力する。
3. Server タブの “Run built-in server” にチェックを入れる。
4. GWT タブの [Development Mode] において “Super Development Mode” を選択し、[Available Modules:] から “MML2LaTeX” を削除する。
5. Arguments タブの [VM arguments:] に **-Xmx2048m** に変更する*³。
6. [Run] ボタンをクリックし、Eclipse の下部に現れる [Development Mode] タブ中の **app.html** をダブルクリックすると、“web3d” プロジェクトが自動的にビルドされ、Web ブラウザ内に GeoGebraWeb が起動する。ただし、workspace のディスクサイズが約

*³ これは実行マシンのメモリが 2GB ある場合であり、少ない場合は 1024m などとする。

1,100MB になるので注意が必要である。

7 CAS flow in GeoGebra

ここでは CAS View に入力されたコマンドが実行される際の内部の流れを説明する。9970 を超える最小の素数を求めるコマンド `NextPrime[9970]` を例に説明する。

1. CAS View で `NextPrime[9970]` を入力する。
2. `org.geogebra.common.cas.GeoGebraCAS` クラスの `evaluateGeoGebraCAS()` メソッドによって、"`NextPrime[9970]`" が `ValidExpression` クラスのインスタンスに変換される。
3. `org.geogebra.common.cas.giac.CASgiac` クラスの `evaluateGeoGebraCAS()` メソッドによって、Giac コマンド文字列 "`nextprime(9970)`" に変換される。
4. `org.geogebra.common.jre.cas.giac.CASgiacJre` クラスの `evaluateCAS()` メソッドから `evaluate()` メソッド → `evalRaw()` メソッドが順次実行され、計算結果 9973 を得る。

8 数式処理関数を CAS View に追加する方法

前節のステップ 3 で CAS View に入力した文字列が Giac コマンド文字列に変換されるが、その変換ルールは `org.geogebra.common.cas.giac.Ggb2giac` クラスに記述されている。つまり、新しい数式処理関数を CAS View に追加するには、この部分を修正すればよい。以下では具体例として、中国剰余定理に関する `ichinrem` 関数を追加する方法を述べる。

1. “common” プロジェクト → `src/main/java` → `org.geogebra.common.cas.giac` にある “`Ggb2giac.java`” を探す。
2. 40 行目から始まる `getMap()` メソッド内に以下を追加する：

```
p("IchinRem.N", "ichinrem(%)" );
```

“`IchinRem.N`” の `IchinRem` が CAS View コマンドとなり、`N` が引数の個数を表す。中国剰余定理のように引数が複数で個数が不定の場合、このように `N` を指定する。この修正を行った上で、リビルドすると、CAS View で

```
IchinRem[{3,5},{2,7},{4,11}]
```

と入力すると

```
{268, 385}
```

という結果を得ることができるようになる。(これは $x \equiv 3 \pmod{5}$, $x \equiv 2 \pmod{7}$, $x \equiv 4 \pmod{11}$ を満たす解が $x \equiv 268 \pmod{385}$ であることを示している。)

また、上の修正のみで Web アプリケーション版である GeoGebraWeb の CAS View においても IchinRem コマンドが利用可能となる。ただし、コマンド入力補完機能を利用するには上の修正に加えて対応箇所を修正する必要がある。以上のように、“Ggb2giac.java”の `getMap()` メソッドに、`p(A, B)` という形で変換ルールを追加することで CAS View に様々な関数を追加することができる。

参考文献

- [1] GeoGebra Setup, <https://dev.geogebra.org/trac/wiki/Setup>
- [2] GeoGebraWeb, <https://dev.geogebra.org/trac/wiki/Web>
- [3] Giac, <http://www-fourier.ujf-grenoble.fr/~parisse/>
- [4] Giac.js, <https://www-fourier.ujf-grenoble.fr/~parisse/webxcas.html>