

有限体積法を高速化するための 領域特化言語のC++への埋め込み

伊藤 正勝, 宮島 敬明, 藤田 直行

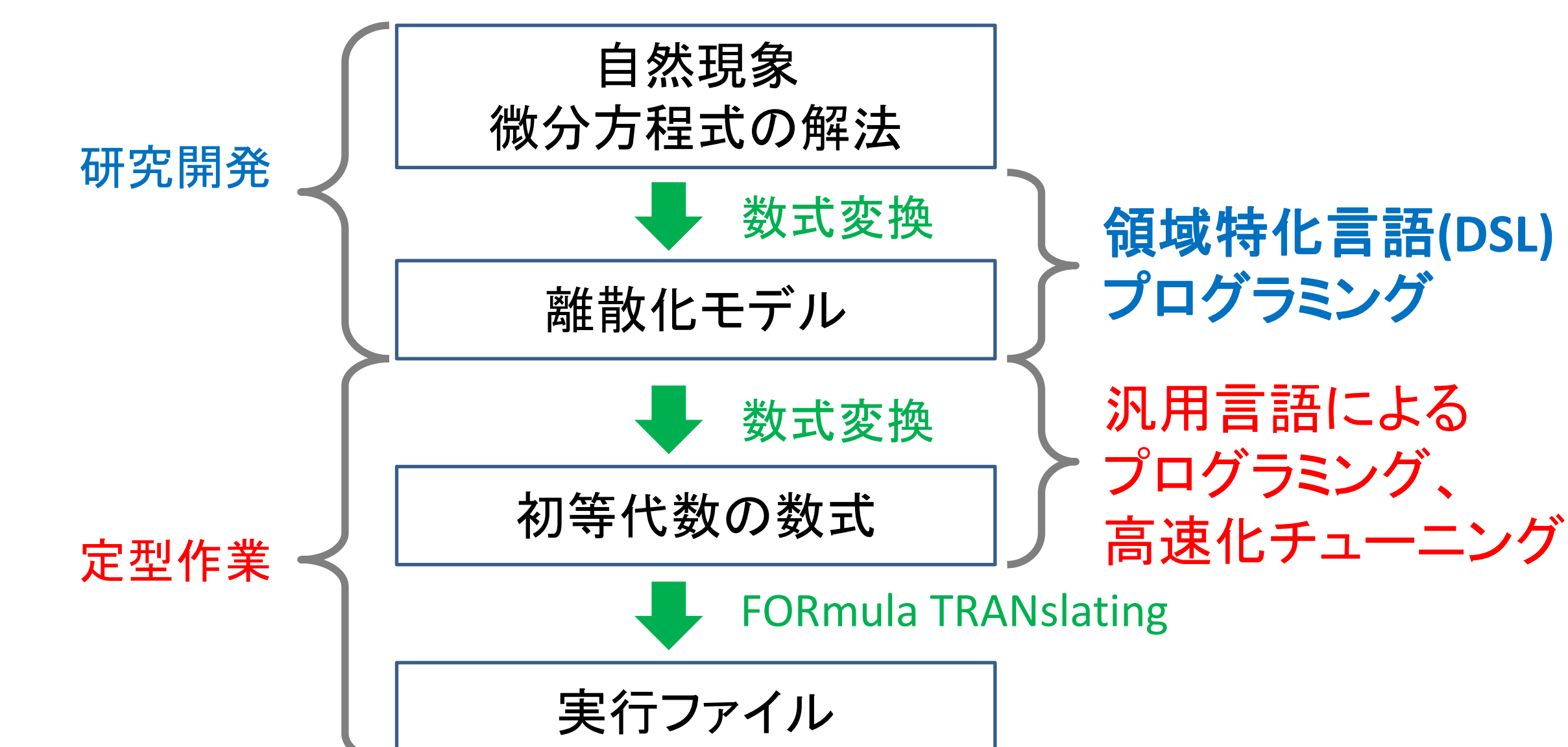
宇宙航空研究開発機構 航空技術部門 数値解析技術研究ユニット

我々は、数値流体力学のシミュレーション開発において、ソースコードを書くために、モデルの数式を初等的で冗長な数式に変換せざるをえないことが、開発と高速化の妨げになっていると考えている。

そこで、モデル数式をそのままソースコードとするために、領域特化言語を設計してC++に埋め込んでいる。我々の領域特化言語は、流体现象のモデリング手法である有限体積法に特化しており、そのコンパイラは有限体積法の意味において、ソースコードをハードウェアに近い低レベルのコードに変換してC++コンパイラに渡す。今回は、流体力学の重要現象である拡散と対流のうち、拡散現象を扱えるように、試作版²を作成した。

序：科学シミュレーションのアプリ開発

数式変換としてのプログラミング



なぜDSL?

- 計算科学者が研究開発に専念するためには、コンパイラに低次の数式変換を任せて、プログラミングでは高次の数式変換を扱う必要がある。

なぜ埋め込み型?

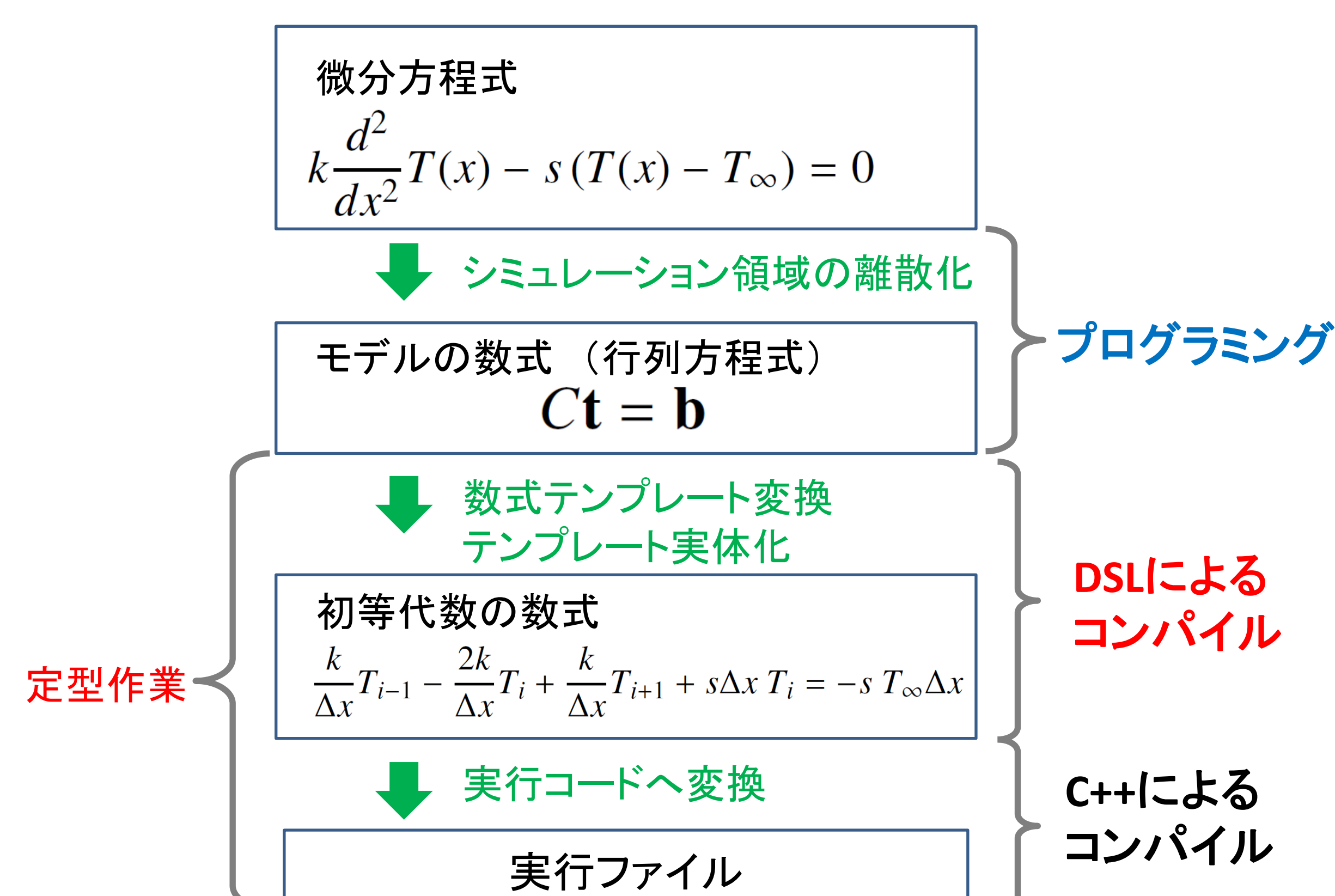
- 独立型 DSL**：実装**コスト高い**
 - Fortran：モデリング手法固有の数式変換はサポート外
- 埋め込み型 DSL**：実装**コスト低い**
 - モデリング手法ごとにDSLが開発されつつある。
 - 有限要素法：Feel++, COOLFLuid, ...
 - 有限体積法 → 本発表

有限体積法テンプレートの開発*

- 埋め込み型 DSL：ホスト言語は C++
 - C++ クラスライブラリと同じようにコンパイルできる。
- 数式テンプレート変換によるDSL
 - 有限体積法の意味論モデルで数式評価
 - Boost.Protoライブラリによる実装

低レベルの数式変換はDSLコンパイラに

- 例) 有限体積法で一次元熱伝導問題を解く場合



有限体積法の意味論モデルとは?

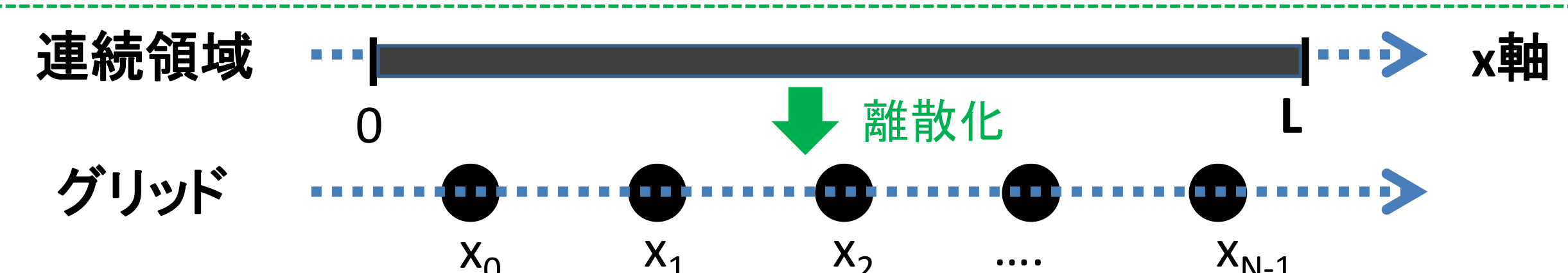
- 定型的な数式変換を自動化するためのモデル
- テンプレートメタプログラミングで数式テンプレート変換として実装

有限体積法テンプレートの使用法

ソースコードを、DSLコンパイラが**有限体積法の意味論**に基づいて、低レベルのコードに変換し、C++コンパイラが実行コードに変換する。

```
FVM::Grid grid( N, L );
```

長さLの連続領域をグリッドに

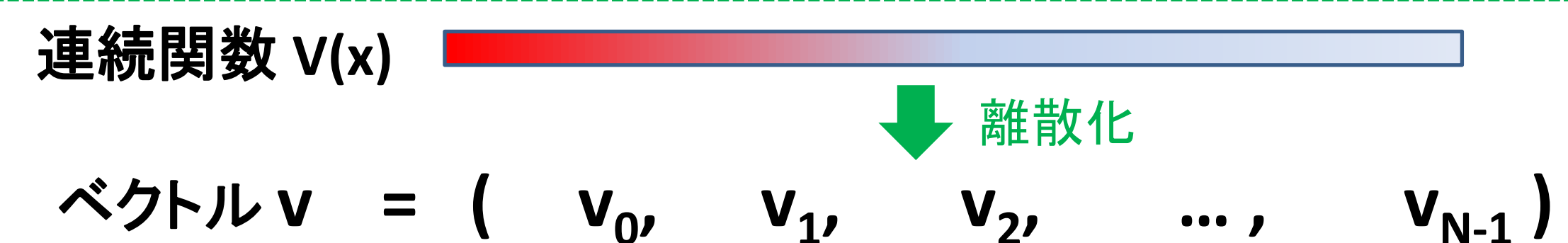


```
grid.addDirichletBoundary(
    -1, 0, TB );
grid.addNeumannBoundary(
    N, N-1, 0.0 );
```

左端の境界温度をT_Bで固定
右端の境界は断熱

```
LA::Vector b =
    grid.discretizeFunction(
        - s * TI );
```

定数項をベクトル **b** へ変換

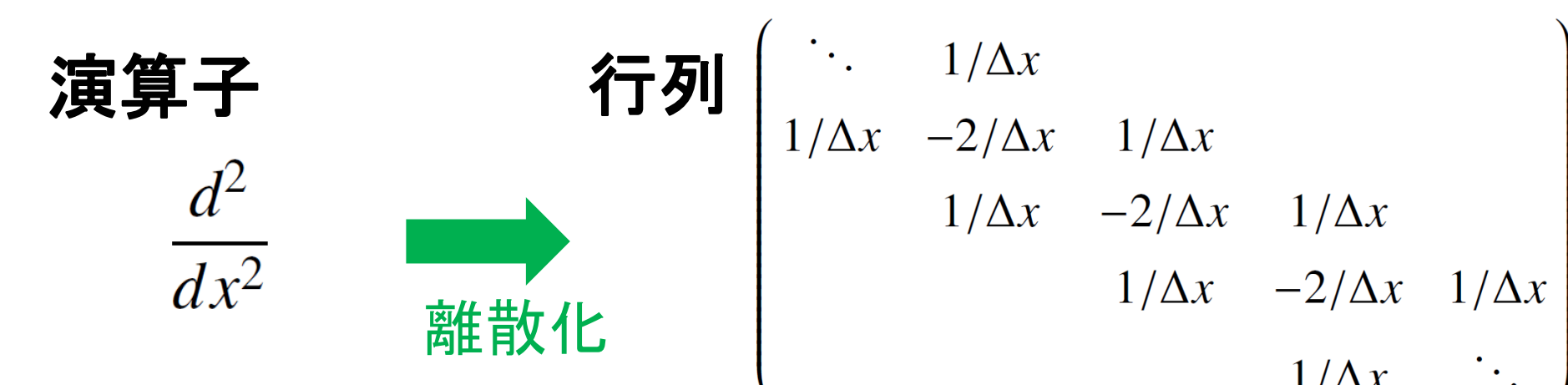


```
auto opr = proto::deep_copy(
    k * FVM::diffOpr *
        FVM::diffOpr
    - s * FVM::identityOpr );
```

微分方程式の演算子部分
 $k \frac{d^2}{dx^2} - s \hat{I}d$

```
LA::Matrix C =
    grid.discretizeOperator( opr );
```

演算子を行列 **C** へ変換



```
FVM::BoundaryCorrector
    bCorrector( grid, opr );

bCorrector.applyTo( C );
bCorrector.applyTo( b );
```

境界条件による
行列方程式の補正
Ct = b

まとめと今後の課題

「有限体積法テンプレート」を開発中

- C++ **数式テンプレート**ライブラリとしてのDSL
 - 行列演算での速度低下を回避
 - モデル数式を**低レベルのコード**に変換する必要がなくなる。
- 拡散だけでなく流体现象を扱えるように機能拡張中