

SICE プラントモデリング部会第1回研究会

数式処理システム Maple と 制御対象モデラ MapleSim について

サイバネットシステム株式会社
モデルベース開発推進室 山口(哲)

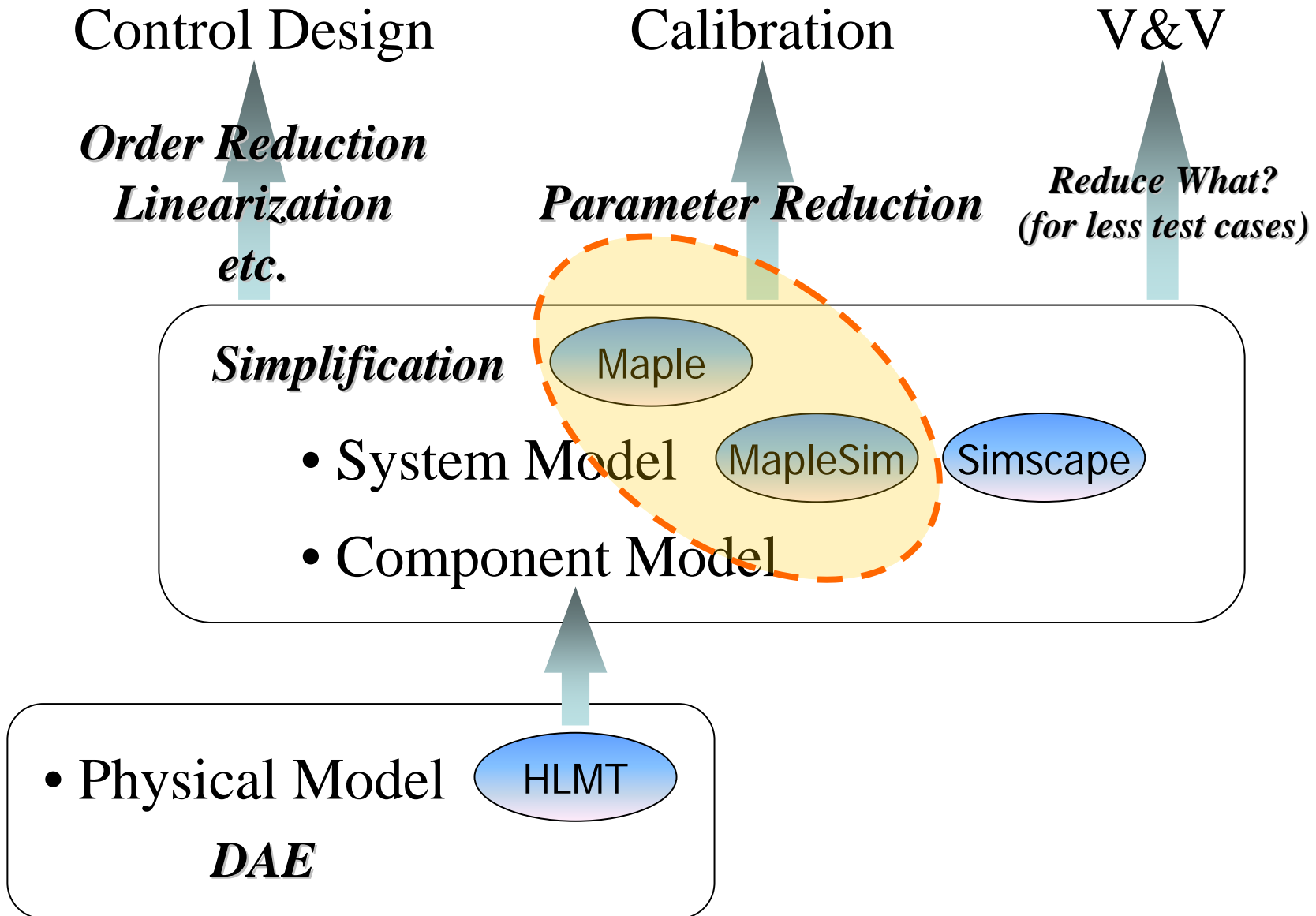
E-mail: infomaple@cybernet.co.jp
www.cybernet.co.jp/Maple

サイバネットシステム株式会社



発表内容

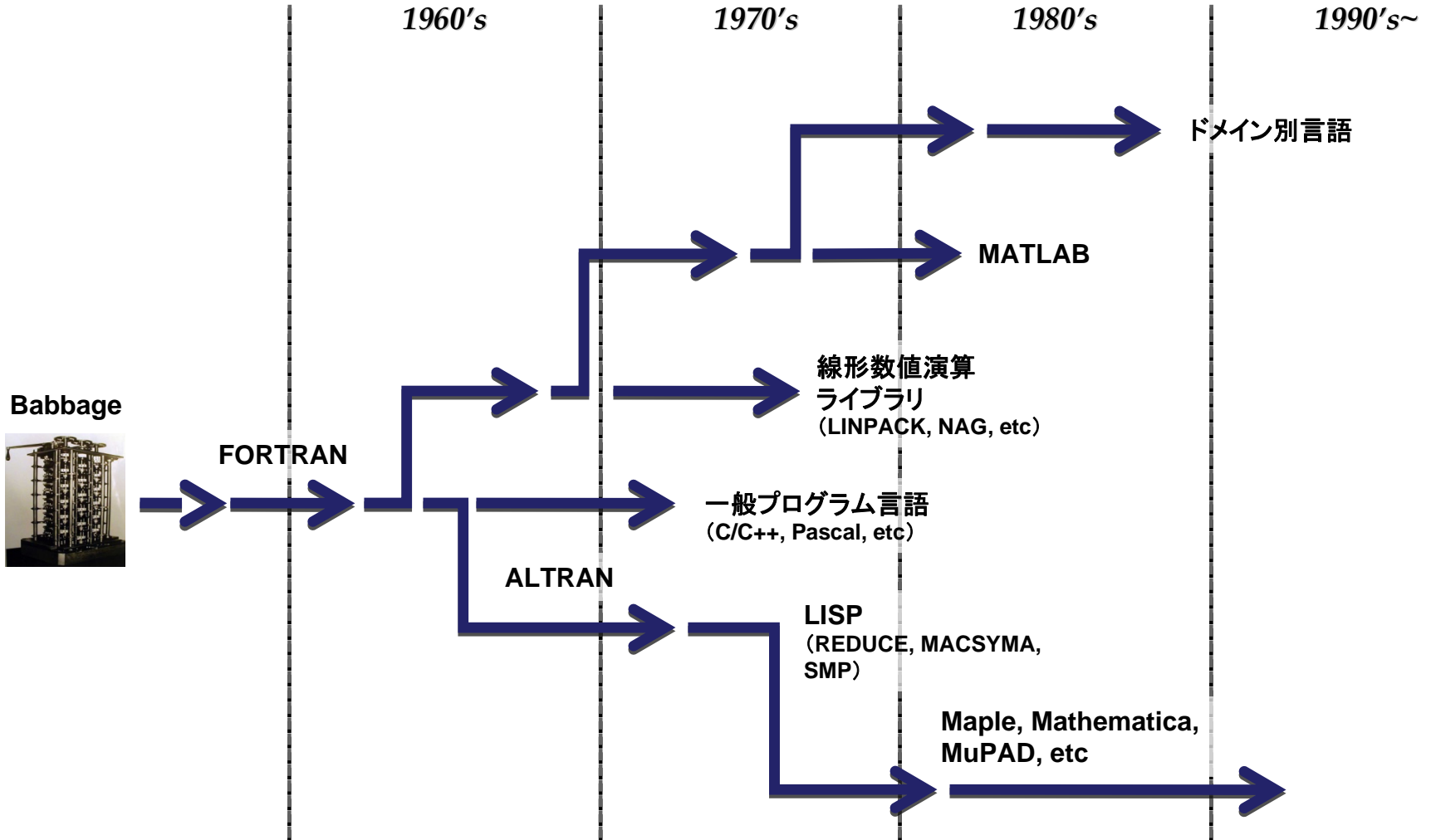
- 数式処理の歴史
- Maple 上の数式処理機能
- 微分方程式と数式処理
- 数式処理ベースの制御対象モデラ
- 数式処理の課題



発表内容

- **数式処理の歴史**
 - Maple 上の数式処理機能
 - 微分方程式と数式処理
 - 数式処理ベースの制御対象モデラ
 - 数式処理の課題

数式処理の歴史(1)



数式処理の歴史(2)

■ 数式処理とは...

- 学術的には「**計算代数(Computer Algebra)**」と呼ばれる分野
- 1965年 B. Buchberger「**グレブナ基底算法**」による連立方程式求解
- 主な算法:

✓ 代数演算(展開、因数分解、部分分数分解等)

✓ 数式変換(簡単化、項書換え等)

✓ 解析演算(微積分、級数展開、極限等)

✓ 微分方程式求解

✓ 任意精度演算

✓ 可視化

✓ 数式ライブラリ

✓ 言語変換(C/Java/VB 等) **数式処理システム**

数式処理 = 計算代数？

数式処理 = 記号計算？

計算代数 = 記号計算？

発表内容

- 数式処理の歴史
- **Maple 上の数式処理機能**
- 微分方程式と数式処理
- 数式処理ベースの制御対象モデラ
- 数式処理の課題

Maple について

- 開発元：カナダ Maplesoft 社
- 本拠地：オンタリオ州ウォーターロー市
- 従業員数：約130名
- 研究連携：
 - ウォーターロー大学、西オンタリオ大学、INRIA (仏)、他
- 主な歴史：
 - 1980年：K. Geddes, G. Gonet らにより『Maple』プロジェクト開始
 - 1988年：Waterloo Maple Software 社を設立
 - 1994年：Maple カーネルの一部を Mathcad へ OEM 提供 (~2006年)
 - 2003年：数学表記での入力可能な Maple 10 をリリース
 - 2008年：複合モデリング環境 MapleSim v1.0 をリリース
 - 2009年：サイバネットシステムが Maplesoft を100%子会社化
 - 2009年：サイバネット、キャッツ、Maplesoft による MBD 連携を発表

Mathematical
Application
Programming
Language
Environment

※ただし、公式ではない

Maple 上の数式処理機能

■ 計算代数機能

> $\text{solve}(ax^2 + bx + c = 0, x)$

$$\frac{1}{2} \frac{-b + \sqrt{b^2 - 4ac}}{a}, -\frac{1}{2} \frac{b + \sqrt{b^2 - 4ac}}{a}$$

(1)

方程式の求解

> $\int_0^{\pi} \sin(x)^5 + \exp\left(-\frac{x^2}{b}\right) dx$

$$\frac{16}{15} + \frac{1}{2} \sqrt{\pi} \sqrt{b} \operatorname{erf}\left(\frac{\pi}{\sqrt{b}}\right)$$

(2)

記号的定積分

> $\text{taylor}(\sin(\tanh(ax)), x=0, 8)$

$$ax - \frac{1}{2} a^3 x^3 + \frac{37}{120} a^5 x^5 - \frac{137}{720} a^7 x^7 + O(x^9)$$

(3)

級数展開

> $\text{factor}(x^{10} + 1)$

$$(1 + x^2)(x^8 - x^6 + x^4 - x^2 + 1)$$

(4)

因数分解

計算代数と記号計算の違い(1)

- 「記号計算＝計算代数」ではない
- 記号計算＝計算代数∪数式変形処理

例) 計算代数では対象としない変形処理

p を定義 :

$$> p := ax^2 + bx + c:$$

ℚ 上での因数分解 :

$$> factor(p)$$

$$ax^2 + bx + c$$

ℚ($\sqrt{b^2 - 4ac}$) 上での因数分解 :

$$> factor(p, \sqrt{b^2 - 4ac})$$

$$-\frac{1}{4} \frac{(-2ax - b + \sqrt{b^2 - 4ac})(2ax + \sqrt{b^2 - 4ac} + b)}{a}$$

$-4ac$

計算代数と記号計算の違い(2)

■ 計算代数

- 代数的構造に基づいた計算法を実現
- 対象: 多項式、行列、代数関数、etc, etc...
 - ✓ 具体的な次数・次元を持っていることが前提

■ 記号計算

- 代数計算に限らず、経験則を含めた式の変形・変換処理を実現
- 対象: 記号的変数の多項式、行列、代数関数、etc, etc...
 - ✓ 例) n 階微分、 n 次行列の LU 分解

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{pmatrix}$$

$$\begin{aligned} & \text{diff}(\sin(k \cdot t), t\$n) \\ & \quad \sin\left(k t + \frac{n \pi}{2}\right) k^n \\ & \text{diff}(f(t) \cdot g(t), t\$n) \\ & \quad \sum_{k=0}^n \binom{n}{k} \frac{d^{k l}}{dt^{k l}} f(t) \frac{d^{n-k l}}{dt^{n-k l}} g(t) \end{aligned}$$

※一部の記号計算は可能

Maple 上の数式処理機能

■ 計算量解析

- `codegen[cost]` コマンドで提供
- 加算・乗算、割当て、変数保存、関数評価の回数をカウント

■ 変数置換による計算効率化

- LargeExpressions パッケージで提供
 - ✓ `Veil` コマンドにより変数置換
 - ✓ `Unveil` コマンドにより置換内容の確認
- `codegen[optimize]` 等のコマンド内部でも利用可能

[codegen_cost_example.mw](#)

発表内容

- 数式処理の歴史
- Maple 上の数式処理機能
- **微分方程式と数式処理**
- 数式処理ベースの制御対象モデラ
- 数式処理の課題

微分方程式と数式処理

■ 数式処理を用いるメリット

- 厳密解の計算
- テイラー級数等の形式的近似解
- DAE における前処理の適用 (低インデックス化)

■ 数式処理と微分方程式の連携成果

- ODE クラスの自動分類
 - ✓ 1階: アーベル、対称性、リカッチ、他
 - ✓ 2階: ベッセル、エルミート、ラゲール、ヤコビ、他
 - ✓ 高階: 線形、非線形
- 階数低減

■ パラメトリックな制御系設計

- SyNRAC (富士通研 / 東大)

→ 式変形と代数関係によるアプローチ

微分方程式と数式処理

■ 記号ソルバ

- dsolve, pdsolve コマンド
- DEtools (ODE用), PDEtools (PDE用)を提供

■ 数値ソルバ

- dsolve, pdsolve コマンド
- 初期値問題:
 - ✓ rkf45, rosenbrock, gear, taylorseries 等
- DAE 問題:
 - ✓ rkf45_dae (DAE用拡張ソルバ) 等
 - ✓ 数式処理による低インデックス化の適用

微分方程式と数式処理

■ Maple による デモンストレーション

- 常微分方程式の厳密解の計算
- 数値解の計算
- Zero Crossing イベントの定義
- 低インデックス化の原理 (微分消去法)
- DAE 問題対処例

Maple の dsolve コマンドのデフォルト設定で計算 :

```
dsol1 := dsolve(dsys, numeric, maxfun = 0)
```

```
dsol1 := proc(x_rkf45_dae) ... end proc
```

(6.3)

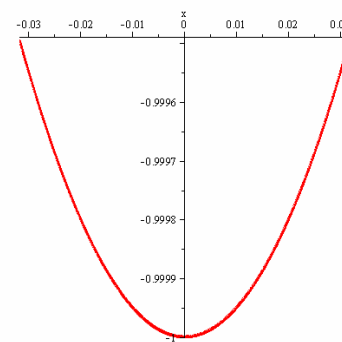
低インデックス化を適用、射影計算を無効 :

```
dsol2 := dsolve(dsys, numeric, differential = true, projection = false, maxfun = 0)
```

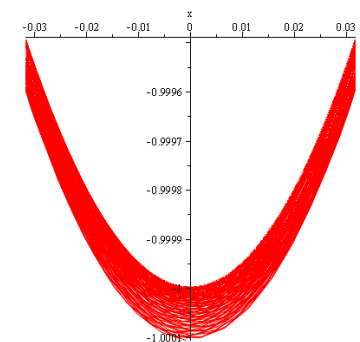
```
dsol2 := proc(x_rkf45_dae) ... end proc
```

(6.4)

```
plots[odeplot](dsol1, [x(t), y(t)], 0 .. 100, numpoints = 2000, frames = 100)
```



```
plots[odeplot](dsol2, [x(t), y(t)], 0 .. 100, numpoints = 2000, frames = 100)
```



発表内容

- 数式処理の歴史
- Maple 上の数式処理機能
- 微分方程式と数式処理
- **数式処理ベースの制御対象モデラ**
- 数式処理の課題

数式処理ベースの制御対象モデル『MapleSim』

■ Maple をベースにした複合物理モデリング環境

The screenshot displays the MapleSim 2 software interface. On the left, there are several toolbars and panels categorized by domain:

- 電気 (Electrical):** A toolbar with various electrical components like resistors, capacitors, and inductors.
- 制御 (Control):** A toolbar with control blocks such as integrators, differentiators, and gain blocks.
- 1D機械 (回転・並進) (1D Mechanical (Rotation/Translation)):** A toolbar with mechanical components like gears, shafts, and springs.
- 3Dマルチボディ (3D Multi-Body):** A toolbar with 3D mechanical components like fixed frames, flexible beams, and rigid bodies.
- 熱 (Thermal):** A toolbar with thermal components like heat capacitors, heat flows, and thermal conductors.

The main workspace shows a complex multi-domain model. It includes an electrical circuit with a controller, a 1D mechanical system with a spring and damper, and a 3D multi-body system with a rotating shaft and a probe. The model is interconnected, showing the flow of energy and information between domains. On the right side, there is a 'Parameters' panel with various settings like time step, solver options, and plot options.

→ 複合領域モデルを容易に構築

MapleSim の特徴 (1)

- 数式モデルの自動生成・Maple 環境との連携が可能

The image displays two windows from the MapleSim environment. The left window shows a circuit model with components like resistors, capacitors, and a motor, with probes labeled 'Probe1' and 'Probe2'. The right window shows the 'Linear System Analysis Template' in Maple, which contains a description, a model diagram, and a list of model equations. A pink callout box is overlaid on the right window, containing the text: 数式モデルの解析
コード生成
簡単化・最適化. Below the callout box, a graph shows a plot of a function over time.

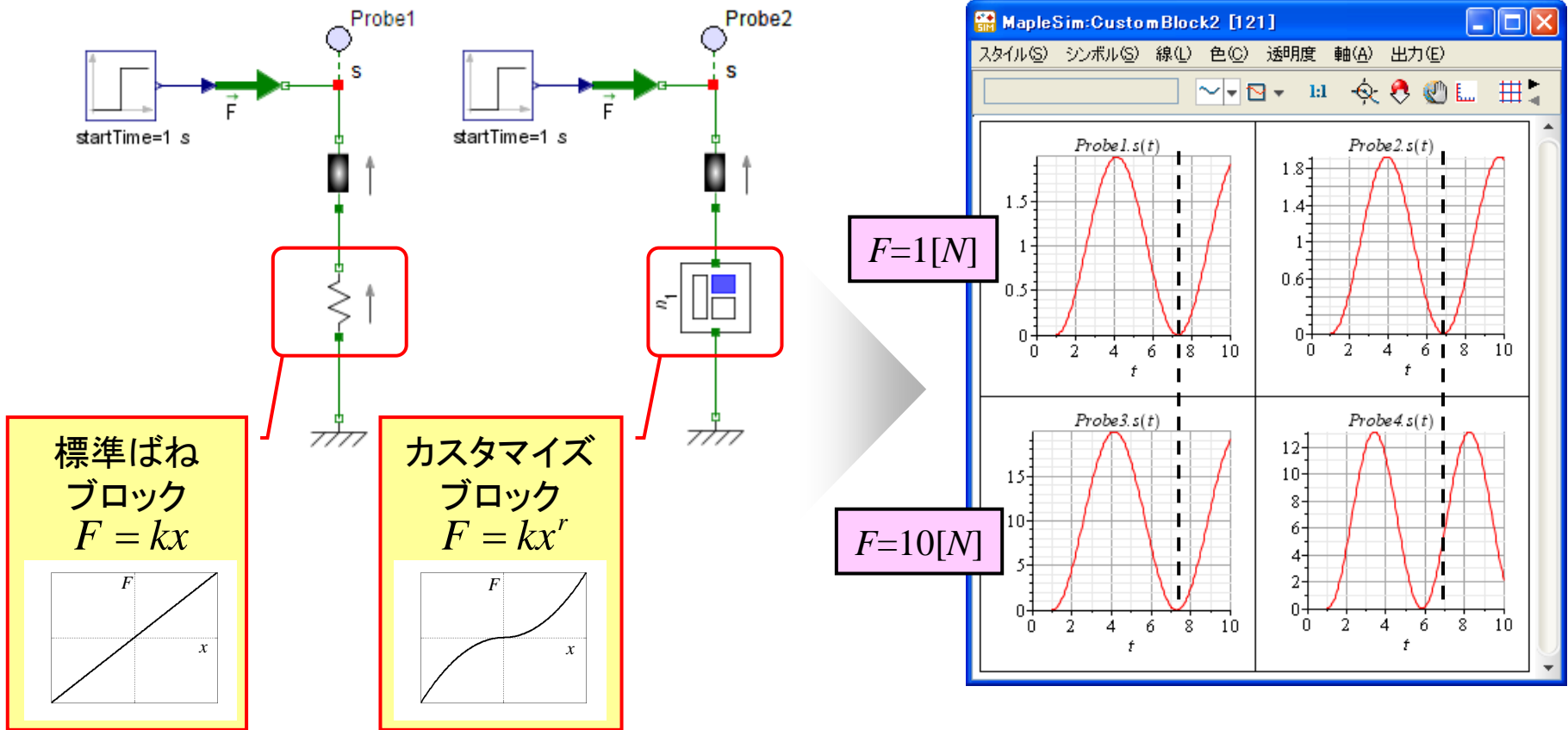
モデルから数式を自動生成
Maple環境へエクスポート



Maple 数式処理環境の活用

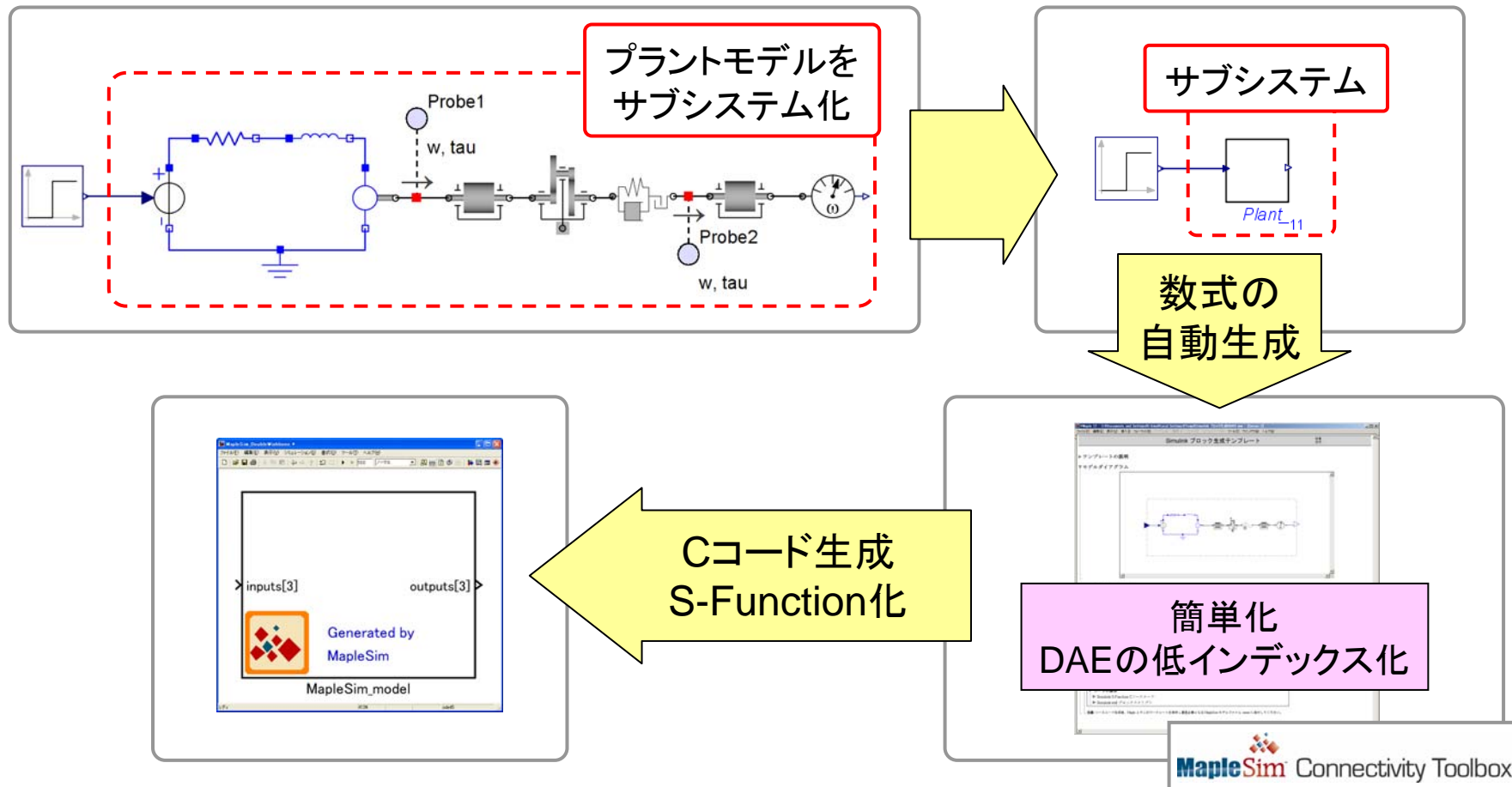
MapleSim の特徴 (2)

■ カスタムブロック (非線形バネの例)



MapleSim の特徴 (3)

■ MapleSim モデルのエクスポート



発表内容

- 数式処理の歴史
- Maple 上の数式処理機能
- 微分方程式と数式処理
- 数式処理ベースの制御対象モデラ
- **数式処理の課題**

数式処理の課題

■ 計算速度

- 数値計算◎ 数式計算×
- 対策：
 - ✓ 浮動小数化(近似代数演算)
 - ✓ 線形代数化(並列・マルチコア化)

■ 工学的観点での記号計算

- 中間式膨張
- 対策: 部分的な自動変数置換等による効率的計算過程の実装

■ 統合数式ライブラリ

- HLMT 連携上で必須

参考文献・論文等

■ 記号計算について

- “Computer Algebra’s Dirty Little Secret”, Stephen M. Watt, University of Western Ontario, 2008
- “Computing with Abstract Matrix Structures”, A. Sexton, V. Sorge, S. Watt, ISSAC 2009 Proceedings, Seoul, pp.325-332
- “Implicit Reduced Involutive Forms and Their Application to Engineering Multibody Systems”, W. Zhou, D. Jeffrey, G. Reid, C. Schmitke and J. McPhee, 2007
- “Linear Algebra Using Maple’s LargeExpressions Package”, J. Carette, W. Zhou, D. Jeffrey, M. Monagan, 2006, Maple Conference 2006

■ DAE について

- “Eliminating constraint drift in the numerical simulation of constrained dynamical systems”, D. Braun, M. Goldfarb, Compt. Methods Appl. Mech. Engrg., vol. 198 (2009), pp.3151-3160
- “微分代数方程式の解のべき級数展開法”, 平山 弘(神奈川工大), 京都大学数理解析研究所講究録1335 巻2003 年, pp.49-56