

物理エンジンの作り方 その2

株式会社コーエー

技術支援部シニアエキスパート

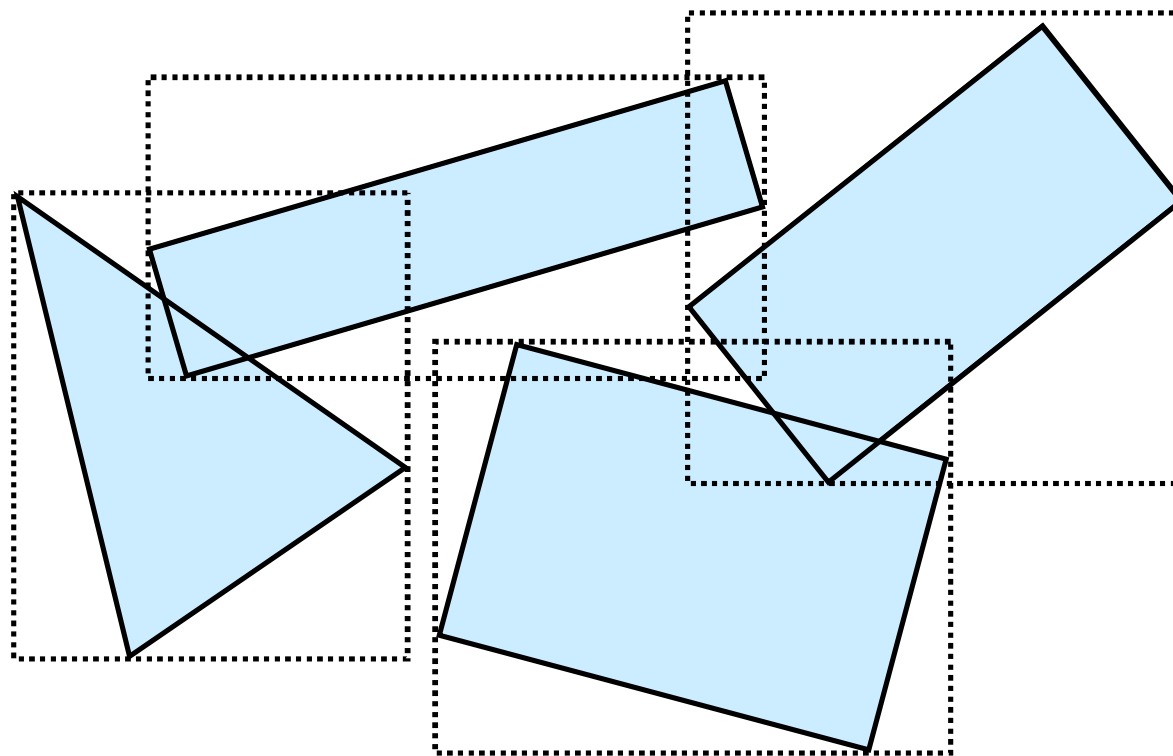
津田順平

並列化の困難さの程度

処理ステージ	並列化の難度
ブロードフェーズ	やや難
ナローフェーズ	容易
拘束式の組み立て	容易
拘束解決	困難
積分(運動方程式)	容易

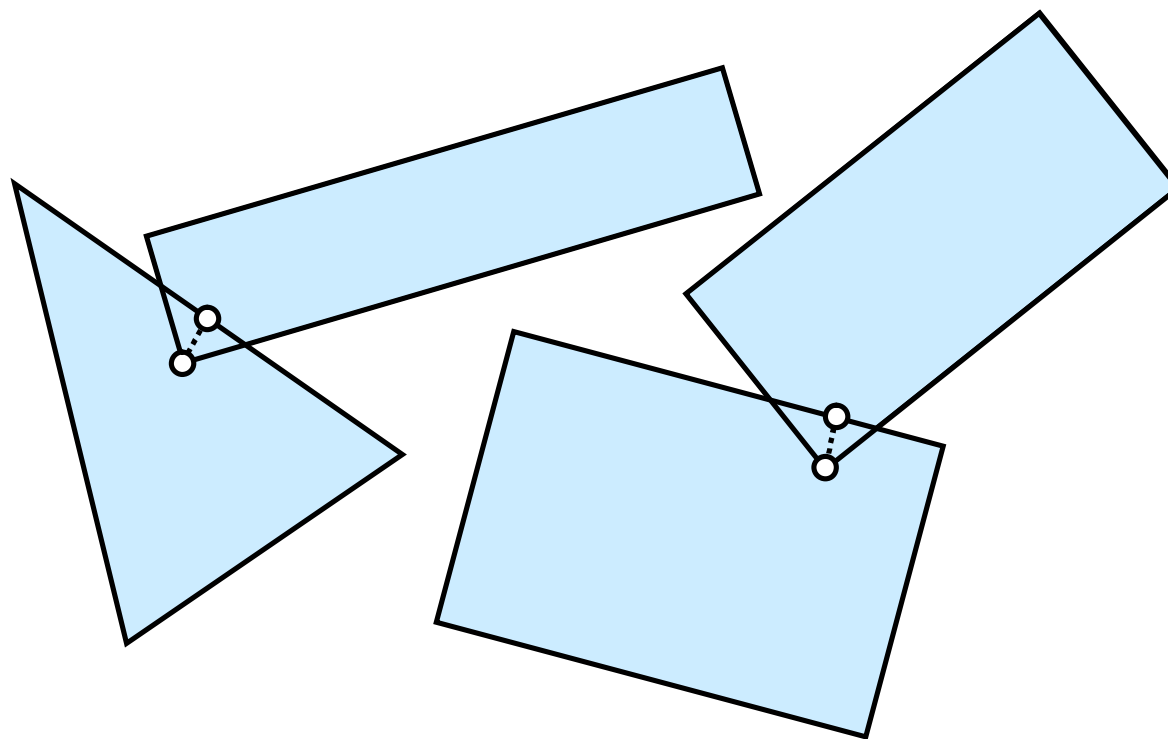
ブロードフェーズ

- 外接ボックスによる粗い衝突判定

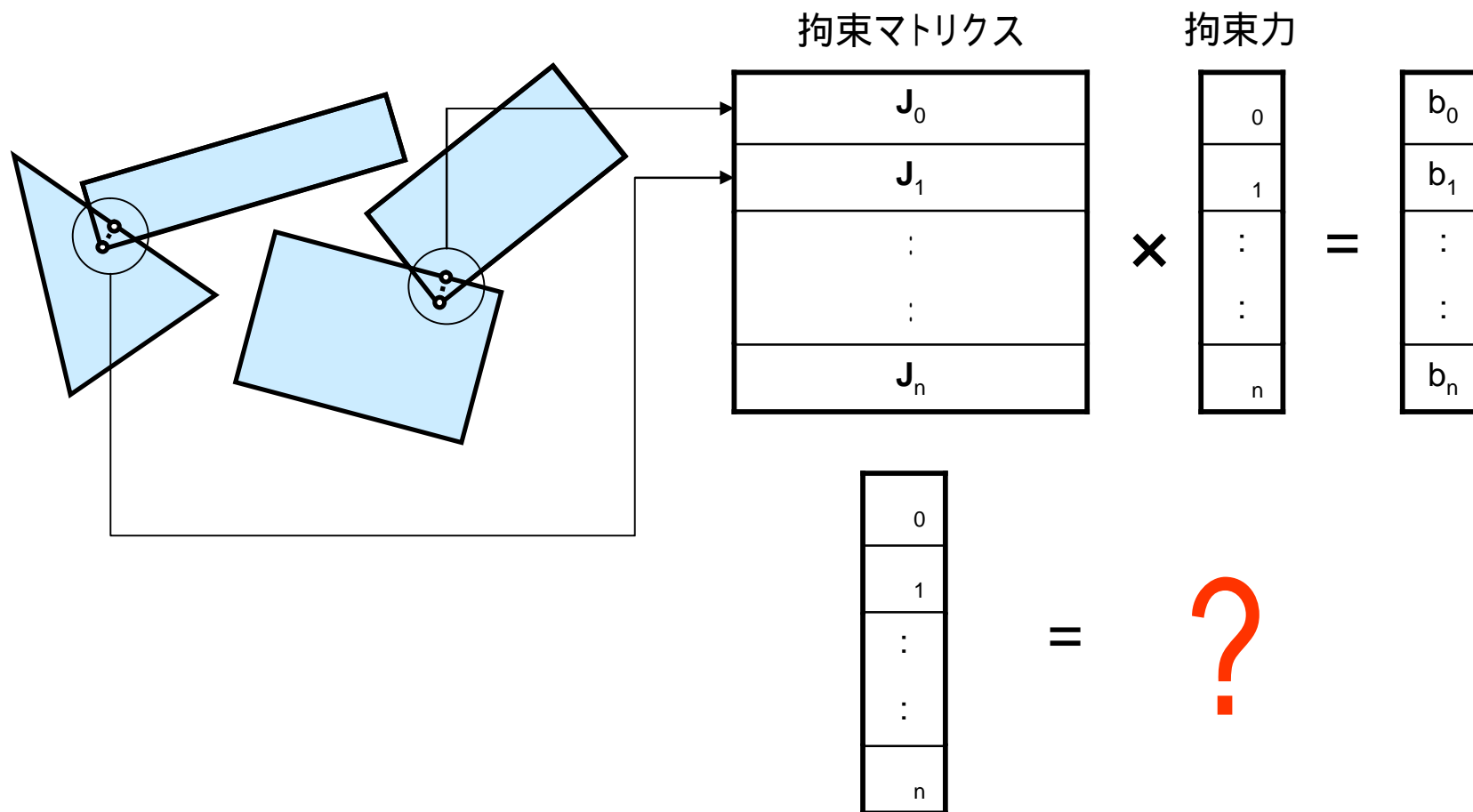


ナローフェーズ

- 貫通解消方向の算定を含む精密な衝突判定



拘束組み立て / 解決 (解析法)





積分

運動方程式を数値積分し、新しい位置、速度を求める
並進運動

$$\mathbf{F} = m \dot{\mathbf{v}}$$

(\mathbf{F} :力, m :質量, $\dot{\mathbf{v}}$:加速度)

回転運動

$$\boldsymbol{\tau} = \mathbf{I} \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I} \boldsymbol{\omega}$$

($\boldsymbol{\tau}$:トルク, \mathbf{I} :慣性テンソル, $\boldsymbol{\omega}$:角速度)



拘束の解き方

■ インパルス法

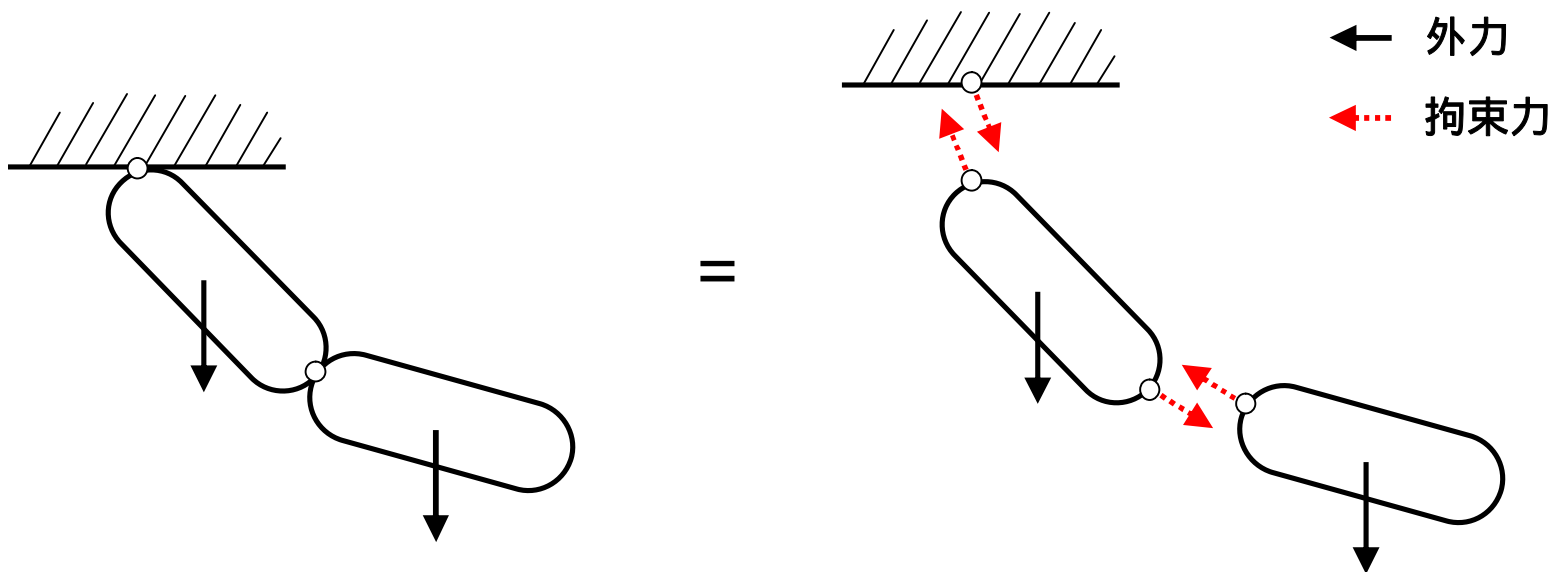
- [Guendelman03], [Mirtich95]
- 拘束を多数の小さな衝突(インパルス)として扱う

■ 解析法

- [Erleb05], [Baraff89]
- 接触点/ジョイントごとに拘束条件式を立てる
- それらを連立させた方程式を解き拘束力を求める

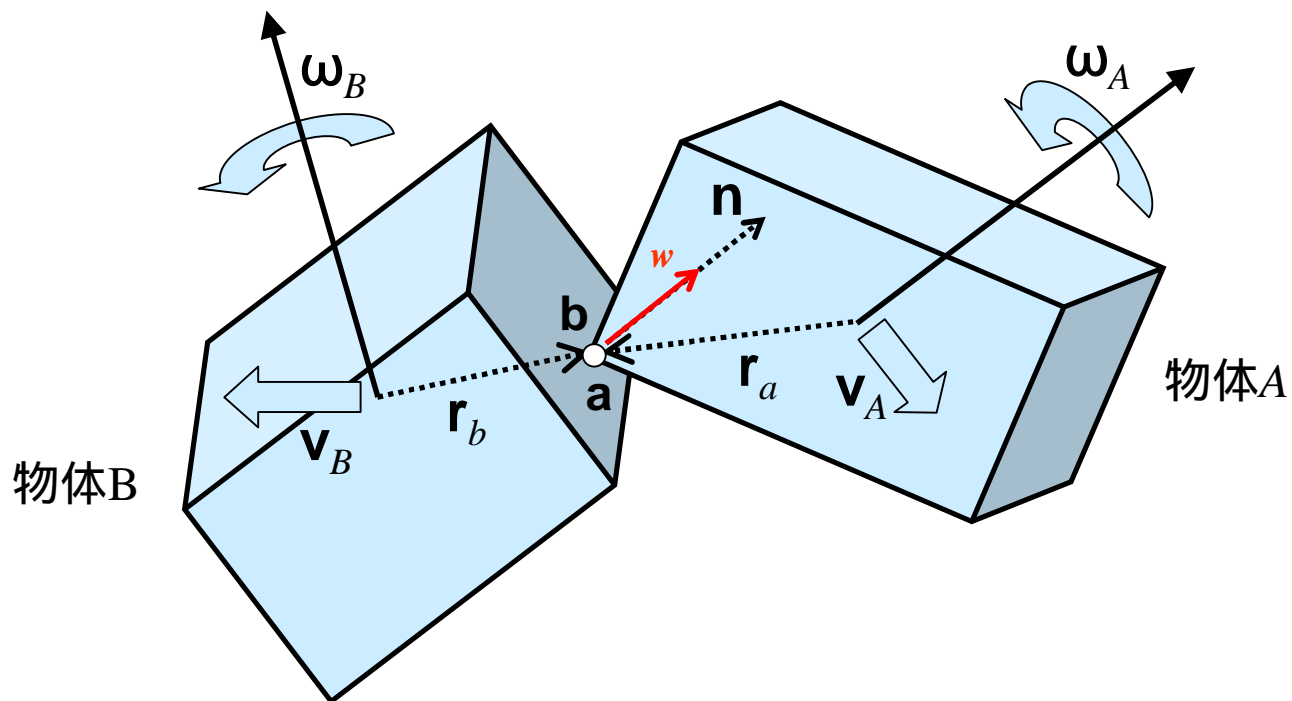
解析法

拘束を実現する力、「拘束力」を求める
拘束力を運動方程式の F 、 τ に算入
後は拘束なしの問題として解けばよい



拘束軸上の相対速度

物体A, B がそれぞれの所属点 a , b で接触している
B側の面法線 n 上での接触点 a , b の相対速度 w は？



拘束軸上の相対速度

$$\begin{aligned}w &= \mathbf{n} \cdot (\mathbf{v}_a - \mathbf{v}_b) \\&= \mathbf{n} \cdot (\mathbf{v}_A + \boldsymbol{\omega}_A \times \mathbf{r}_a - \mathbf{v}_B - \boldsymbol{\omega}_B \times \mathbf{r}_b) \\&= \mathbf{n} \cdot \mathbf{v}_A + \mathbf{r}_a \times \mathbf{n} \cdot \boldsymbol{\omega}_A - \mathbf{n} \cdot \mathbf{v}_B - \mathbf{r}_b \times \mathbf{n} \cdot \boldsymbol{\omega}_B \\&= (\mathbf{n}^T, (\mathbf{r}_a \times \mathbf{n})^T, -\mathbf{n}^T, -(\mathbf{r}_b \times \mathbf{n})^T) \begin{pmatrix} \mathbf{v}_A \\ \boldsymbol{\omega}_A \\ \mathbf{v}_B \\ \boldsymbol{\omega}_B \end{pmatrix} \\&= \mathbf{J}\mathbf{u} \text{ (相対速度)} \quad [\mathbf{J} \in \mathbb{R}^{1 \times 12}, \mathbf{u} \in \mathbb{R}^{12 \times 1}] \end{aligned}$$



拘束の表現

一般の場合でも拘束条件は

相対速度 0 (貫通なし、可動範囲制限など)

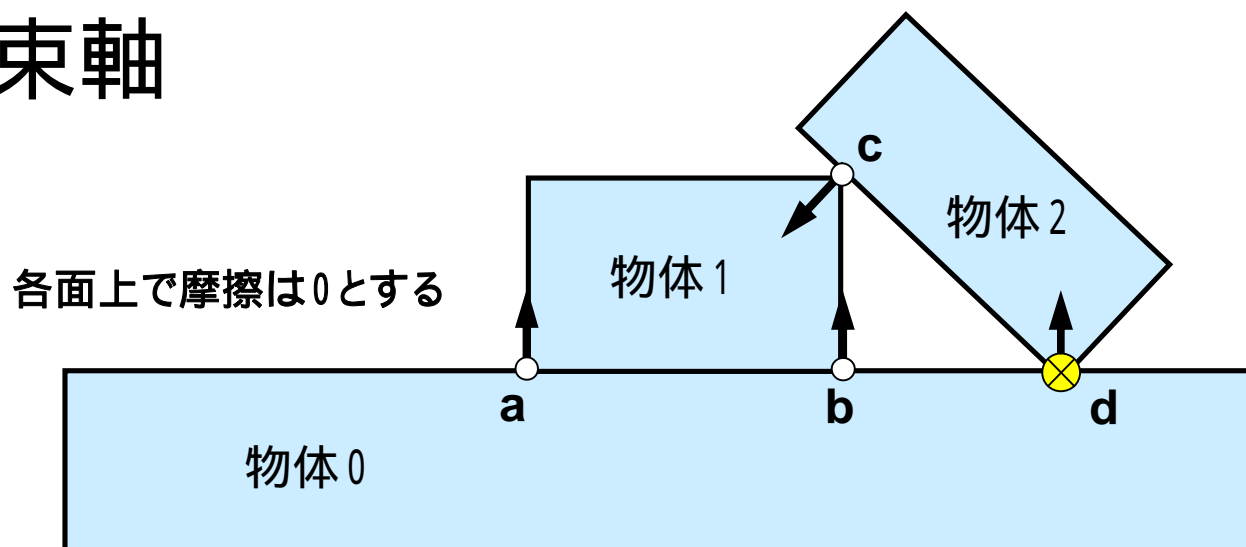
または

相対速度 $= 0$ (ジョイントなど)

として表現できる

多体の物理

- ⊗ 面上拘束 = 等式拘束
- 接触点 = 不等式拘束
- ↑ 拘束軸



拘束条件式 [Erleb05]

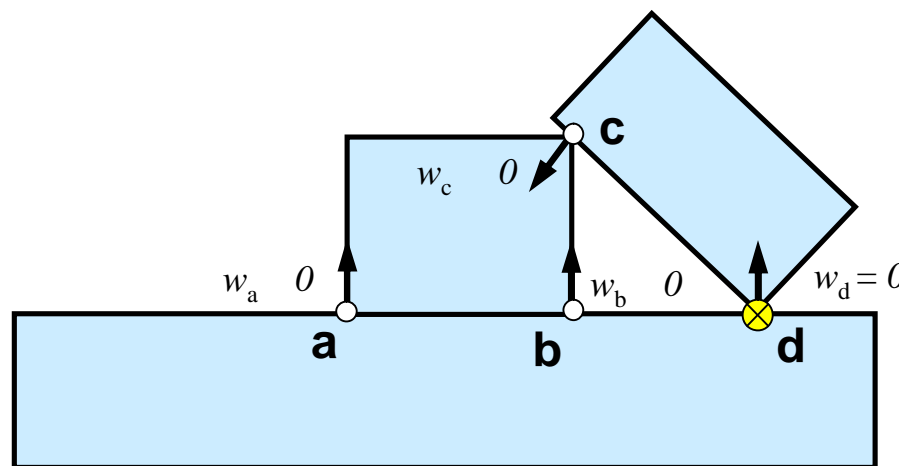
$$\begin{array}{l}
 w_a = \\
 w_b = \\
 w_c = \\
 w_d =
 \end{array}
 \begin{array}{c}
 \mathbf{J} \\
 \begin{array}{|c|c|c|}
 \hline
 & & 0 \\
 \hline
 & & 0 \\
 \hline
 & 0 & \\
 \hline
 0 & & \\
 \hline
 \end{array}
 \end{array}
 \left[\begin{array}{c}
 \mathbf{u}_{0+} \\
 \mathbf{u}_{1+} \\
 \mathbf{u}_{2+}
 \end{array} \right]
 \begin{array}{c}
 \mathbf{M}^{-1} \\
 \begin{array}{|c|c|c|}
 \hline
 \mathbf{M}_0^{-1} & 0 & 0 \\
 \hline
 0 & \mathbf{M}_1^{-1} & 0 \\
 \hline
 0 & 0 & \mathbf{M}_2^{-1} \\
 \hline
 \end{array}
 \end{array}
 \left[\begin{array}{c}
 \mathbf{J}^T \\
 \begin{array}{|c|c|c|c|}
 \hline
 & & & 0 \\
 \hline
 & & 0 & \\
 \hline
 0 & 0 & & \\
 \hline
 \end{array}
 \end{array}
 \right]
 \begin{array}{c}
 \left[\begin{array}{c}
 a \\
 b \\
 c \\
 d
 \end{array} \right]
 \begin{array}{l}
 + \mathbf{F}_{0,ext} \\
 + \mathbf{F}_{1,ext} \\
 + \mathbf{F}_{2,ext}
 \end{array}
 \end{array}
 \left. \vphantom{\begin{array}{c} \mathbf{J}^T \\ \begin{array}{|c|c|c|c|} \hline & & & 0 \\ \hline & & 0 & \\ \hline 0 & 0 & & \\ \hline \end{array} \end{array}} \right\} t
 \end{array}
 \left. \vphantom{\begin{array}{c} \mathbf{M}^{-1} \\ \mathbf{J}^T \end{array}} \right\} \begin{array}{l}
 \text{拘束力} \\
 \text{外力}
 \end{array}
 \end{array}
 \left. \vphantom{\begin{array}{c} \mathbf{M}^{-1} \\ \mathbf{J}^T \\ \left[\begin{array}{c} a \\ b \\ c \\ d \end{array} \right] \end{array}} \right\} \begin{array}{l}
 \text{各物体の加速度} \\
 \text{1ステップ後の各物体の速度}
 \end{array}
 \end{array}
 \left. \vphantom{\begin{array}{c} \mathbf{J} \\ \mathbf{M}^{-1} \\ \mathbf{J}^T \\ \left[\begin{array}{c} a \\ b \\ c \\ d \end{array} \right] \end{array}} \right\} \text{1ステップ後の各拘束軸上の相対速度}$$

拘束条件式

t を t に含め、整理すると...

$$\mathbf{A} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} + \mathbf{b} = \begin{matrix} w_a & 0 \\ w_b & 0 \\ w_c & 0 \\ w_d & = 0 \end{matrix}$$

方程式と不等式が混在！



LCP (線形相補問題)

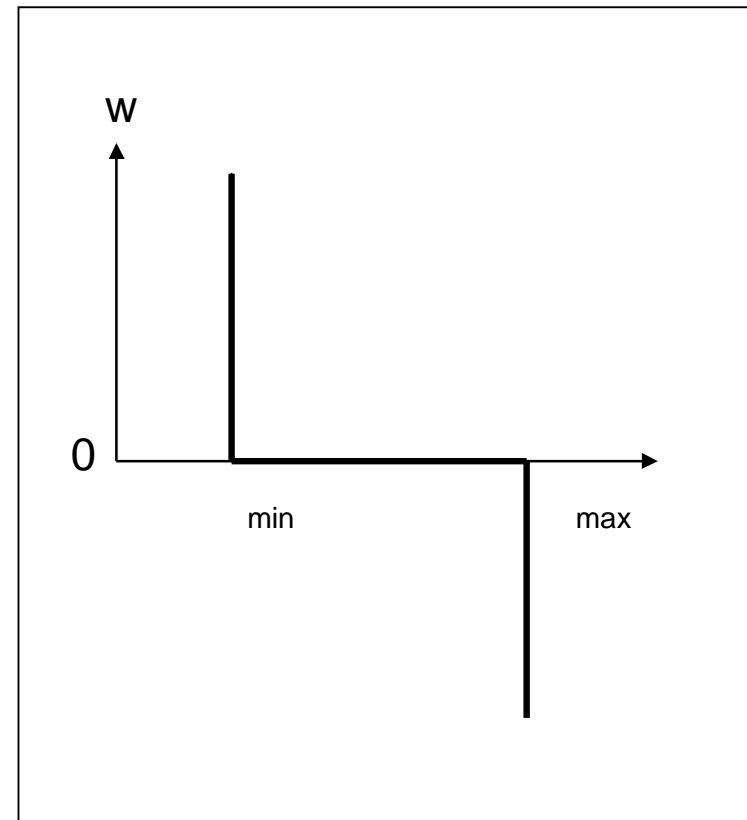
$w = A + b$ とする

以下を満たす w と λ を求める

$$w > 0 \text{ ならば } \lambda = \min$$

$$w = 0 \text{ ならば } \min < \lambda < \max$$

$$w < 0 \text{ ならば } \lambda = \max$$



詳細は [AIMMS07] など

Gauss-Seidel 法 (GSM)

- 線型方程式を以下のように変形

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

$$\begin{pmatrix} -a_{00} & 0 & 0 \\ 0 & -a_{11} & 0 \\ 0 & 0 & -a_{22} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & a_{01} & a_{02} \\ a_{10} & 0 & a_{12} \\ a_{20} & a_{21} & 0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & -a_{01}/a_{00} & -a_{02}/a_{00} \\ -a_{10}/a_{11} & 0 & -a_{12}/a_{11} \\ -a_{20}/a_{22} & -a_{21}/a_{22} & 0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_0/a_{00} \\ b_1/a_{11} \\ b_2/a_{22} \end{pmatrix}$$

Gauss-Seidel 法

変形した方程式を「更新式」と解釈
 k 回目から $k+1$ 回目の更新は

$$\begin{pmatrix} x_{0,k+1} \\ x_{1,k+1} \\ x_{2,k+1} \end{pmatrix} = \mathbf{A} \begin{pmatrix} x_{0,k} \\ x_{1,k} \\ x_{2,k} \end{pmatrix} + \mathbf{b}$$

更新を繰り返し $\mathbf{x}_{k+1} = \mathbf{x}_k = \mathbf{x}_{conv}$ になったとする
この \mathbf{x}_{conv} は元の方程式を満たすので求める解となる
(方程式から更新式への変形を逆にたどれば明らか)

G a u s s - S e i d e l 法

i 行目の更新に i - 1 行目までの更新結果を即時利用

$$x_{0,k+1} = a_{00}x_{0,k} + a_{01}x_{1,k} + a_{02}x_{2,k} + b_0$$

$$x_{1,k+1} = a_{00}x_{0,k+1} + a_{01}x_{1,k} + a_{02}x_{2,k} + b_1$$

$$x_{2,k+1} = a_{00}x_{0,k+1} + a_{01}x_{1,k+1} + a_{02}x_{2,k} + b_2$$

即時利用しない場合 (ヤコビ法) に比べて
1.5 ~ 2 倍程度, 収束速度が向上



投影型 Gauss-Seidel 法 [Cottle92]

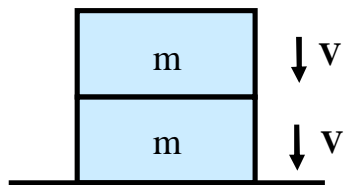
LCP を解く効率的な方法

繰り返しのたびに x_i をクランプ

$$x_i = \text{clamp}(x_i, \min_i, \max_i)$$

たったこれだけ！

インパルス法

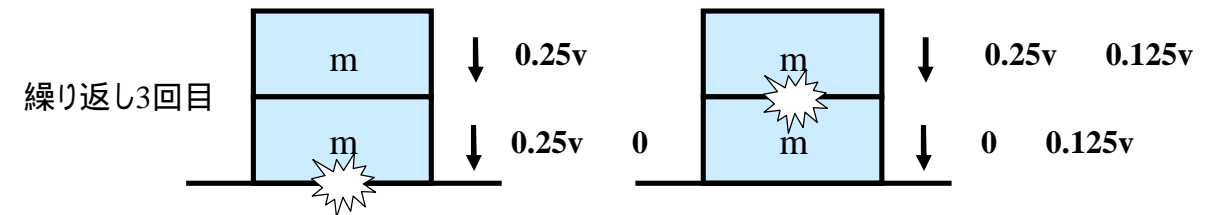
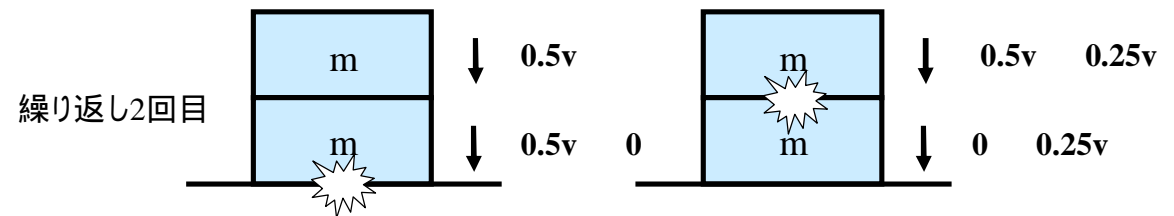
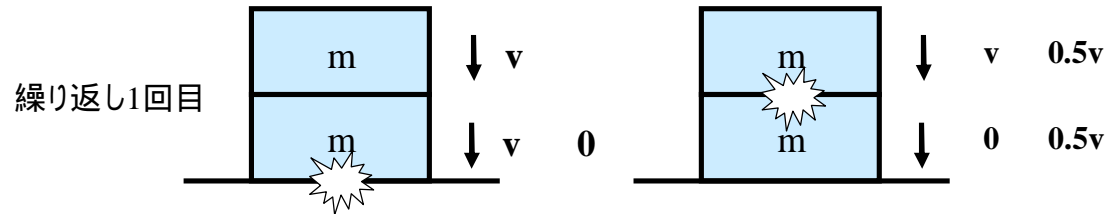


静止接触の例

運動量保存則

$$m_0 \mathbf{v}'_0 + m_1 \mathbf{v}'_1 = m_0 \mathbf{v}_0 + m_1 \mathbf{v}_1$$

2 オブジェクトを繰り返し衝突させ内力を伝播させる。



反発係数はすべて0とする

解析法 vs. インパルス法

- 拘束条件式を考える.

$$A\lambda + \mathbf{b} = \mathbf{J}u'$$

- 条件式の i 行目は

$$\sum_k A_{i,k} \lambda_k + b_i = \sum_k J_{i,k} u'_k$$

- \mathbf{J} の i 行目は 相互作用する2つの剛体に対応する12個の有効要素だけを含む

解析法 vs.インパルス法

- GSMを利用する場合、拘束方程式の1行を処理するということは2つの物体の相互作用によって発生する λ を求めることに等しい
- λ にはすでに t が乗じられている。つまりインパルスに等しい
- さらにGSMでは現在行の λ の計算にそれ以前の行の λ (=インパルス)が即時利用される。これは他のジョイント/接触点からの拘束力によって更新された新しい速度が現在行の λ の計算に使用されることを意味している

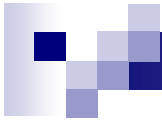


解析法 vs. インパルス法

実質的に、

繰り返し法を利用する解析法 = インパルス法

したがって以降の議論はいずれの方法にも適用可能



シミュレーションの並列化 (拘束解決)



拘束解決

- 拘束力は相互に依存しあっている.
 - GSMを利用する場合, 各行の処理は相互に依存.
- 独立した部分を見つけ出す必要がある.
 - 処理が独立した部分 = 並列処理可能部分
- シミュレーションアイランド?
 - 実際上はそれほど効果的ではない.
 - 現実には巨大なシミュレーションアイランドが高頻度で出現.



独立部分の抽出

- 多色順序付け (Multi Color Ordering)
- 赤-黒順序付け (Red-Black Ordering)
- セル状順序付け (弊社の方法).



多色順序付け (Multi Color Ordering)

- 色の数 = 独立部分 (並列処理可能部分) の数
- 色の数 = 必要な同期回数.
- 同一色グループでは拘束力は独立.
- ある色グループから他の色グループに処理が移るとき同期が必要.

多色順序付け(代数的)

係数マトリクスの要素 (i, j) が非ゼロなら, i, j に対応する未知数には異なる色を割り当てる.

[Mifune05] では以下のようなコードが提案されている.

```
float A[N][N];    //coefficient matrix.
int color[N];     //colors of rows.

for (i = 0; i < N; ++i) color[i] = -1;

for (i = 0; i < N; ++i) {
    m = 0;
    (*)
    for (j = 0; j < i; j++) {
        if (A[i][j] != 0 && color[j] == m) {
            ++m;
            goto (*);
        }
    }
    color[i] = m;
}
```

多色順序付け(代数的)

- 前述のコードを以下の係数マトリクスに適用

係数マトリクス A

	■	■		
■	■		■	
■		■		
		■		■
	■		■	

■ 非ゼロ要素

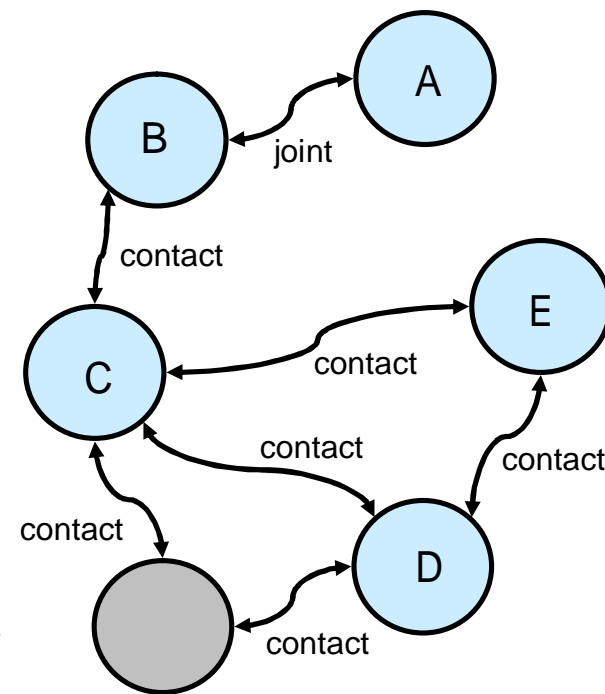
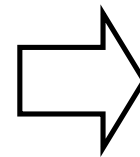
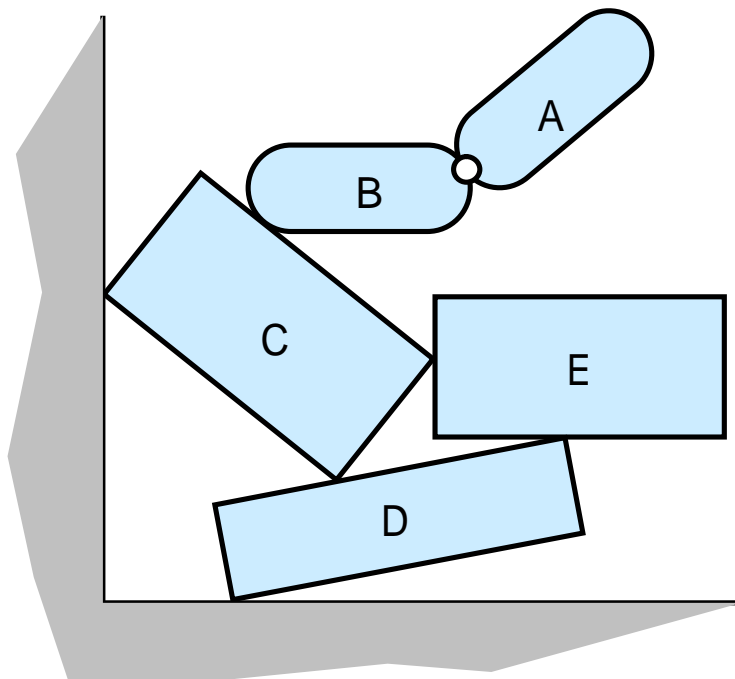


	■	■			color0
■	■		■		color1
■		■			color1
		■		■	color0
	■		■		color2

同一色の行は相互に独立している
ソルバはそれらを並列に処理可能.

接触グラフ

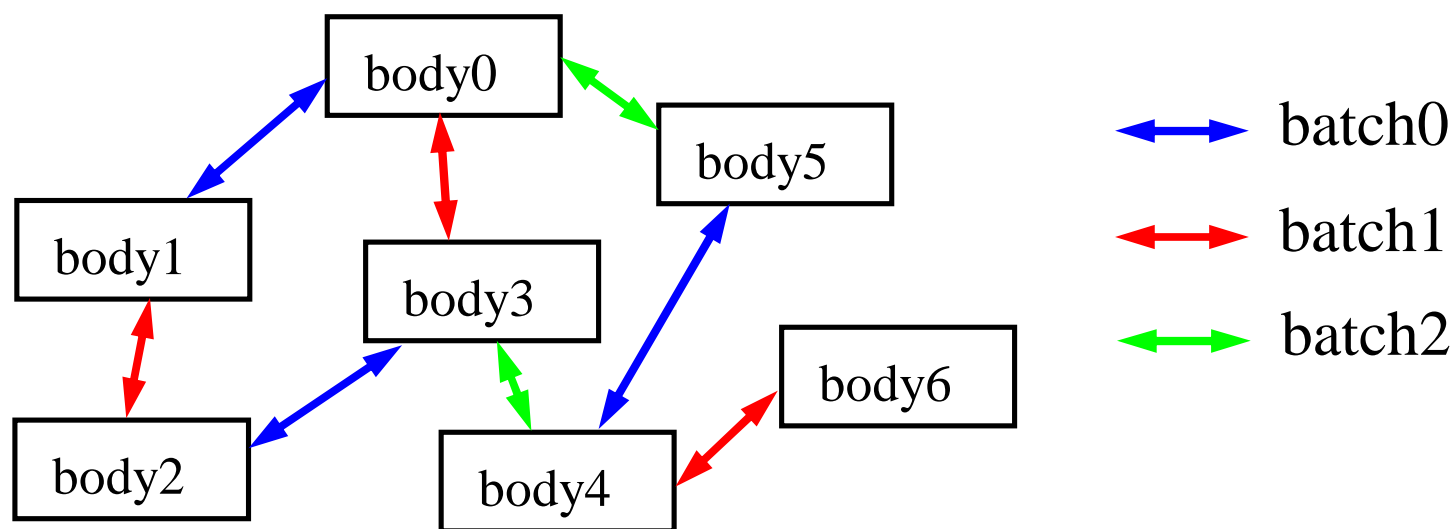
- オブジェクトの接続関係をグラフ構造として保持 [Hahn88]



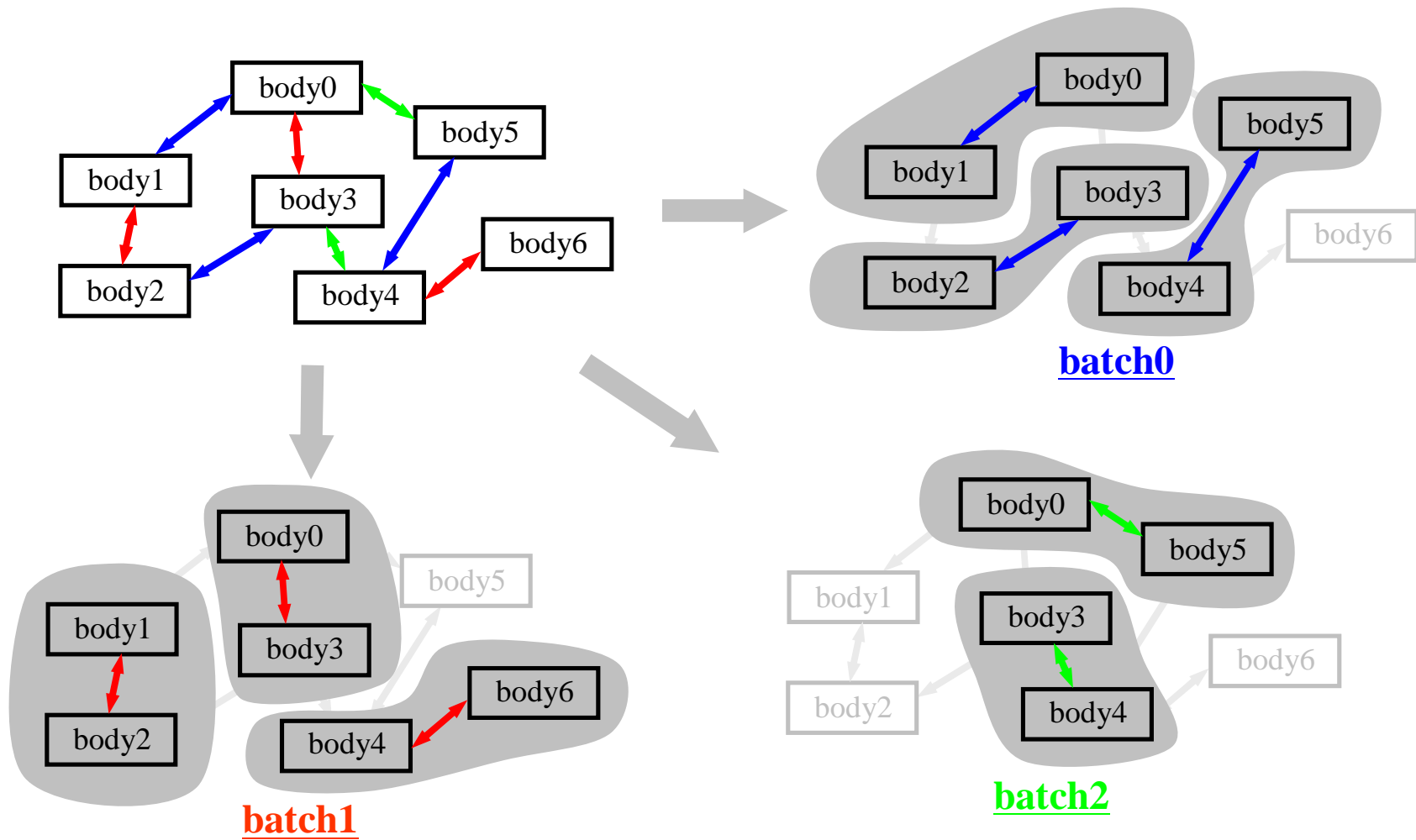
- 可動オブジェクト
- 不動オブジェクト

多色順序付け(幾何学的)

- 両端矢印の"リンク" はジョイントまたは接触を意味する.
- リンクを "バッチ" に分類する. [Chen05], [Keogh07]
- バッチの中の各リンクは独立.




多色順序付け





多色順序付け

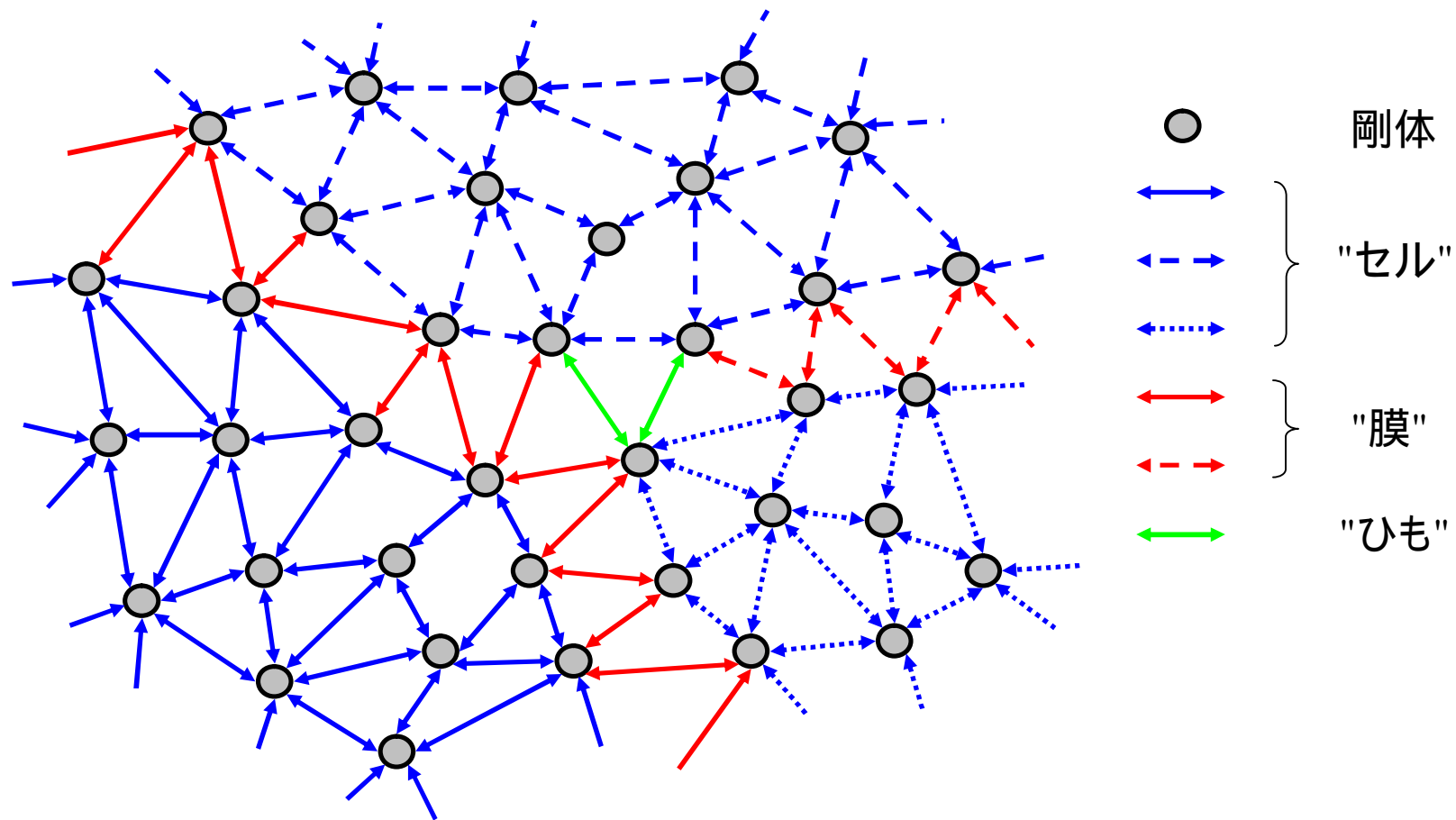
- アルゴリズムは比較的シンプル.
- シーンの複雑さに応じて同期コストが増大
 - ひとつの剛体周りのリンクはすべて異なる色でなければならない
 - ある剛体が n 個の剛体と接触しているなら,最低でも n 回の同期が必要となる.



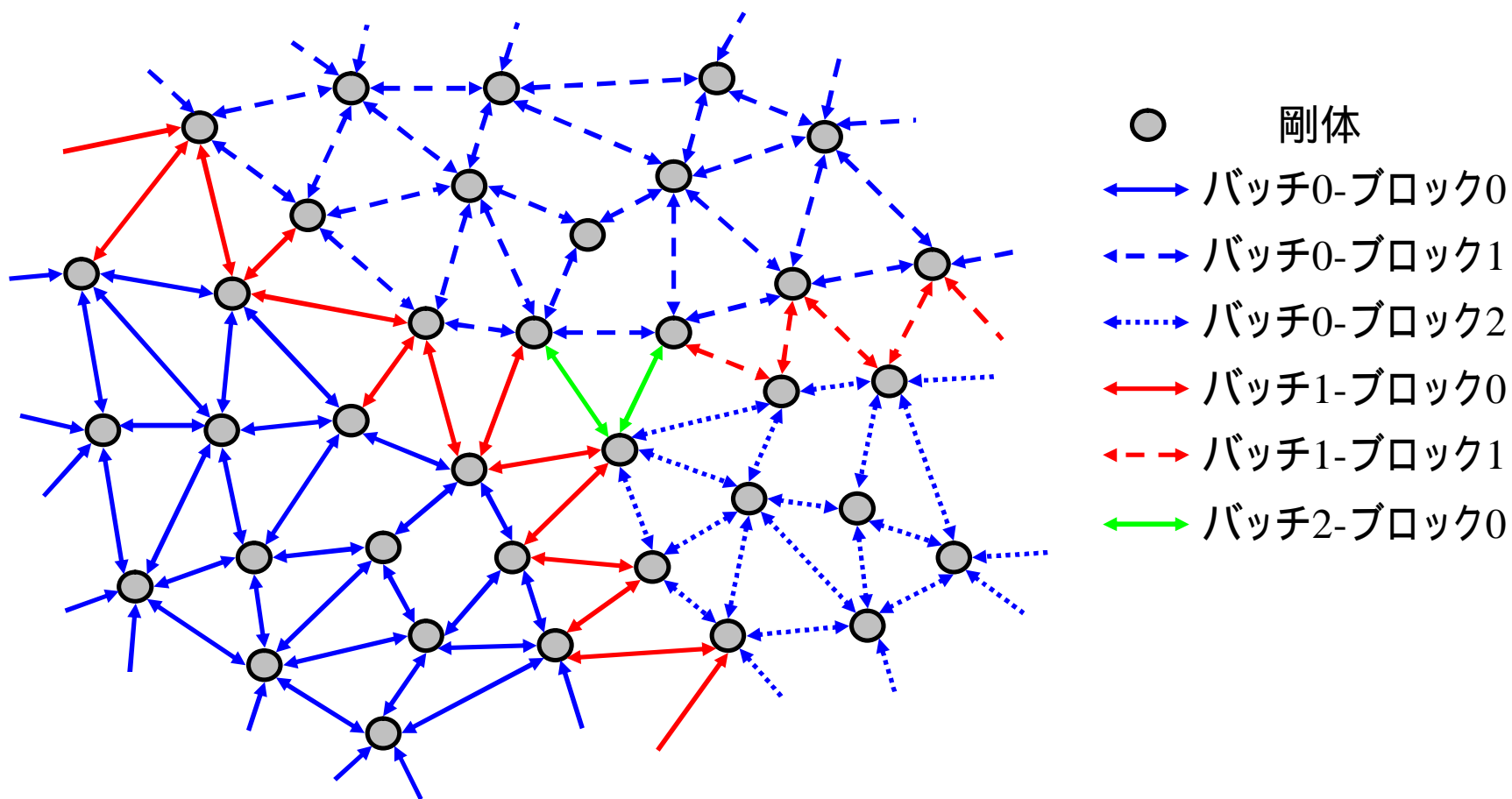
セル状順序付け (弊社の方法)

- 同期回数を可能な限り削減することが目的
- ブロック多色順序付けのバリエーション [Yosui07].
- 接触グラフを活用.
- 剛体周りに幅優先探索を行いリンクをブロック化.

セル状順序付け(Analogy)



セル状順序付け



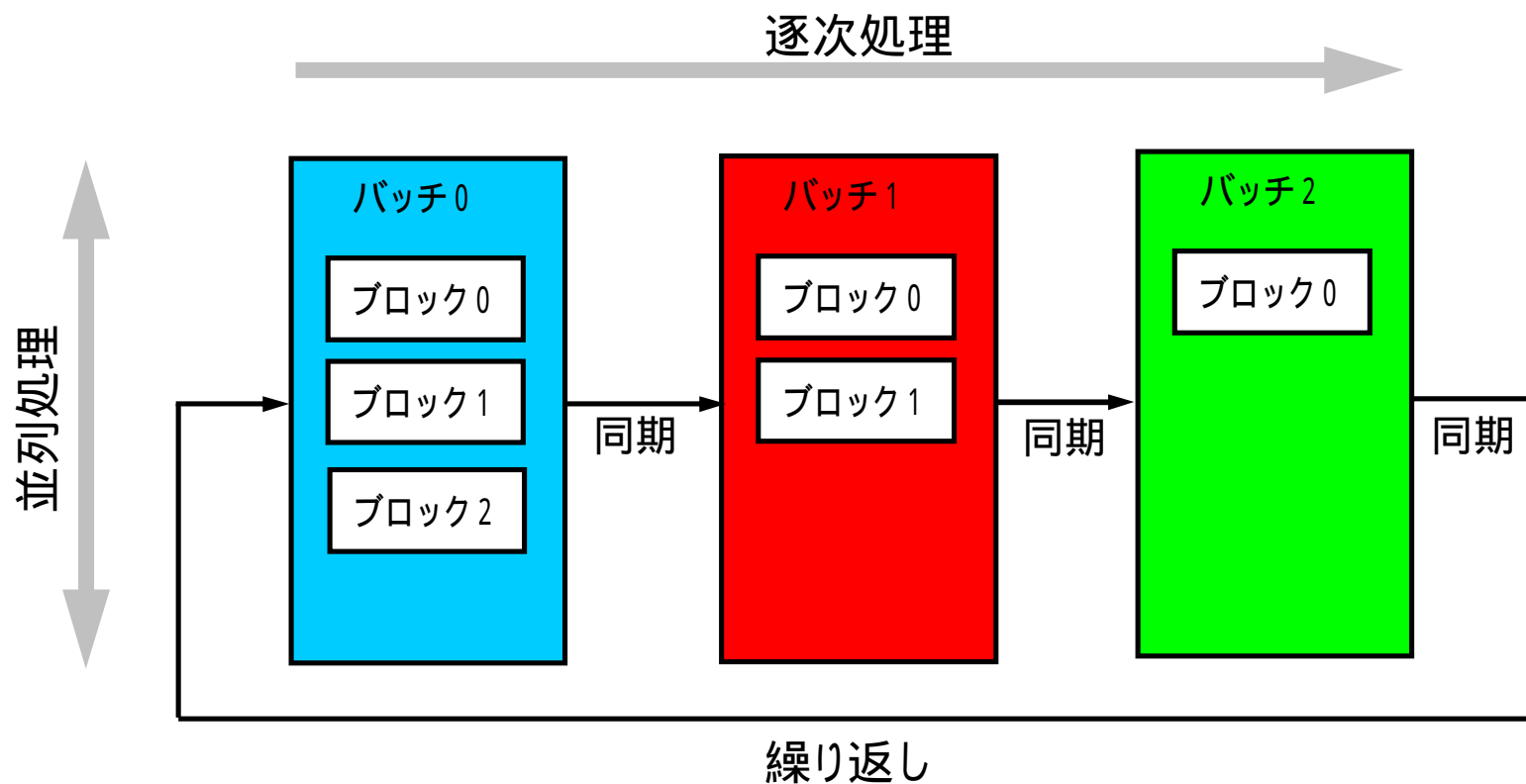


セル状順序付け

- 理論的には1回の繰り返し当たり4回の同期.
 - "ボリューム" "面" "線" "点"
 - 数万オブジェクトであっても原理的に4回
- 実際上は1回の繰り返し当たり3回の同期.
 - "セル(ボリューム)" "膜(面)" "ひも(線)"
 - 数百のオブジェクト数の場合

セル状順序付け

1つのバッチの中の各ブロックは並列に処理可能.





セル状順序付け

- より少ない同期コスト.
- バッチ内では各ブロックは並列処理可能.
 - ブロック内の各リンクは逐次処理.
- 各ブロックの粒度は柔軟に調整可能.
 - 現行のプラットフォームでは重要な特性.
- 接触グラフの探索のみで解決
 - 空間分割のような数値処理は不要.



References

- [AIMMS07] AIMMS Language Reference.
http://www.aimms.com/aimms/download/manuals/AIMMS3LR_MixedComplementarity.pdf
- [Baraff 89] David Baraff. Analytical method for dynamics simulation of nonpenetrating bodies. *Computer Graphics Vol. 23, No. 3, 223-232, 1989.*
- [Chen07] Yen-Kuang Chen et al. High-Performance Physical Simulations on Next-Generation Architecture with Many Cores.
<http://www.intel.com/technology/itj/2007/v11i3/8-simulations/4-methodology.htm>
- [Cottle92] R.W.Cottle et al. The Linear Complementarity Problem. *Academic Press.*
- [Erleb05] Kenny Erleben. Stable, robust, and versatile multibody dynamics animation.
PhD. thesis, Department of Computer Science, University of Copenhagen, Denmark, 2005.



References

- [Guendelman03] Eran Guendelman et al. Nonconvex rigid bodies with stacking. *ACM Tansaction on Graphics, Vol. 22 Issue 3, July 2003.*
- [Hahn88] James K. Hahn. Realistic Anmiation of Rigid Bodies. *Computer Graphics, Vol. 22 Number 4, August 1988.*
- [Keogh07] Chris Keogh. Physics in Games. <http://boombox.ucs.ed.ac.uk/physicspodcasts/genint/2007/resources/ChrisKeogh.pps>
- [Mifune05] T. Mifune et al. A Parallel Algebraic Multigrid Preconditioner Using Algebraic Multicolor Ordering for Magnetic Finite Element Analyses. <http://www.fz-juelich.de/nic-series/volume33/237.pdf>
- [Mirtich95] Brian Mirtich. Impulse-based simulation of rigid bodies. *Proceedings of the 1995 symposium on Interactive 3D graphics.*
- [Yosui07] Kuniaki Yosui et al. A Parellel Multigrid Solver for High Frequency Electromagnetic Field Analyses with Small-scale PC Cluster. *Electronics and ommunication in Japan, Vol. 127 Issue 8, 2007.*