

# 物理系を定義させるシミュレータ SimSym の提案

情報理工学院 数理・計算科学系 増原研究室  
18B04657 木内 康介  
2023年2月13日

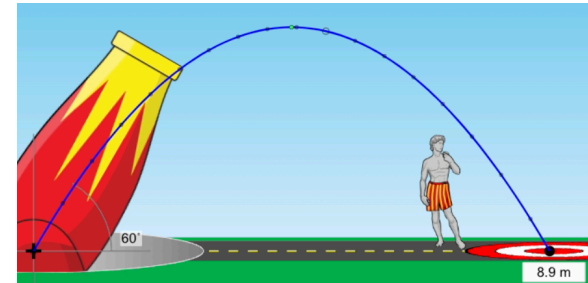
増原研究室の木内が、「物理系を定義させるシミュレータ SimSym の提案」というタイトルで発表いたします。よろしくお願いします。

# はじめに: 物理実験とシミュレータ

実験室での作業は理論的な概念を検証する最も重要な方法

[Holubova, 19]

シミュレータの活用 [Ajredini+, 14]



2

はじめに、物理実験とシミュレータについて紹介します。

物理学を学ぶ上で、実験を行うことは重要です。

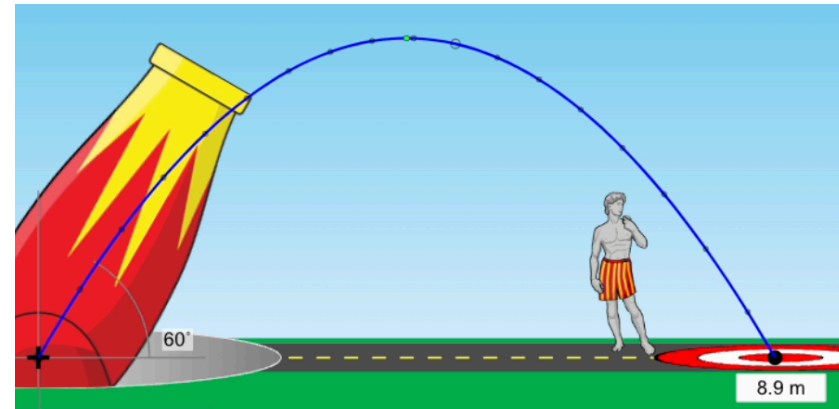
物理学はそもそも現実世界を解析する学問であり、物理学の理論は実験の結果に一致していることが要求されます。

Holubova は、「実験室での作業は理論的な概念を検証する最も重要な方法である」と述べています。

一方で、実験には時間と手間がかかります。そこで活用されているのが、シミュレータです。Ajredini の調査では、シミュレーションを利用した授業でも実際の実験と同様の学習効果を得ることができると報告されています。

## 既存のシミュレータの不足点

$$\begin{cases} x = v_{0x}t \\ y = v_{0y}t - \frac{1}{2}gt^2 \end{cases}$$



教育者が全て行う

3

しかし既存のシミュレータは、教育者が実験系を全て作成しています。そのため、学習者は可視化されている運動とその背景に存在する数式の間を理解することは容易ではありません。

## 実際の物理学習

$$\begin{array}{l} v = v_0 + at \\ \text{等加速度運動の公式} \quad x = v_0 t + \frac{1}{2} at^2 \end{array}$$

### 斜方投射を表す方程式

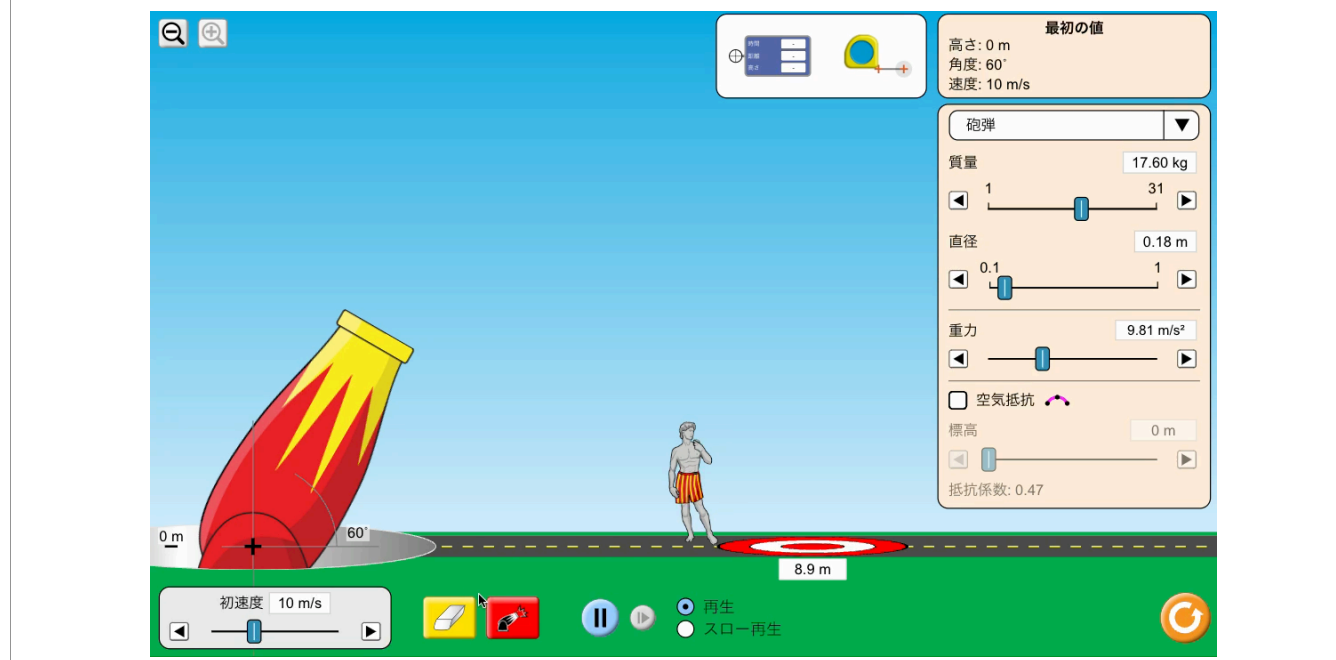
初速を  $(v_{0x}, v_{0y})$ 、重力加速度を  $g$  とする。

$$\begin{cases} v_x = v_{0x} \\ v_y = v_{0y} - gt \end{cases} \quad \begin{cases} x = v_{0x} t \\ y = v_{0y} t - \frac{1}{2} gt^2 \end{cases}$$

物理学の理論を学ぶ際には、物理法則や運動を方程式を活用して学びます。

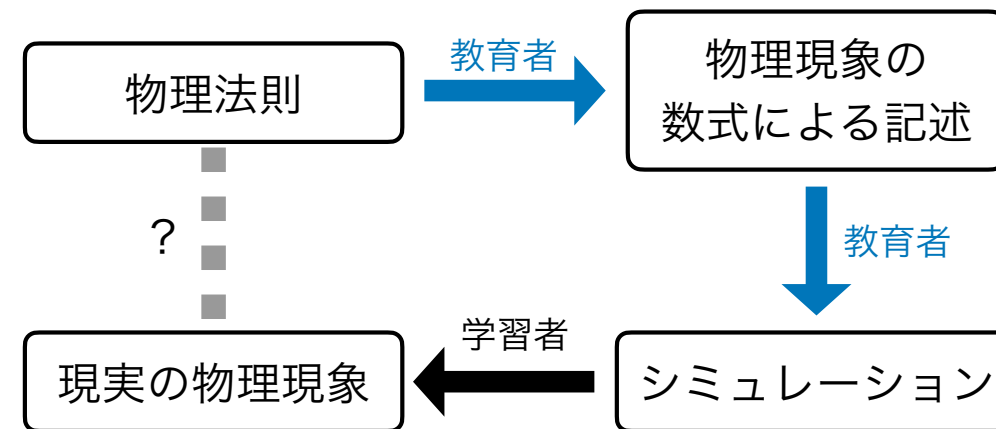
これは、斜方投射を表す方程式の例です。初速を  $(v_{0x}, v_{0y})$ 、重力加速度を  $g$  とすると、斜方投射の軌道はこのような方程式で表されます。

## シミュレータ PhET での斜方投射 [Perkins+, 06]



一方で、既存のシミュレータである PhET での斜方投射のシミュレーションを見てみます。質量や初速度を数値で入力し、それらに従って砲弾が運動、その軌跡が描画されます。しかし、軌跡がどのように求められたものなのかはわかりません。

## 既存のシミュレータによる理解

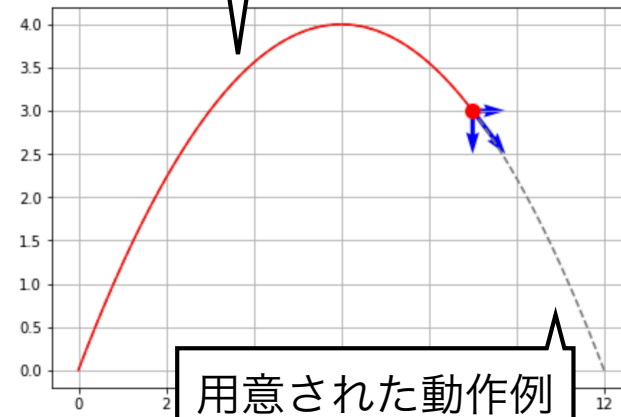


6

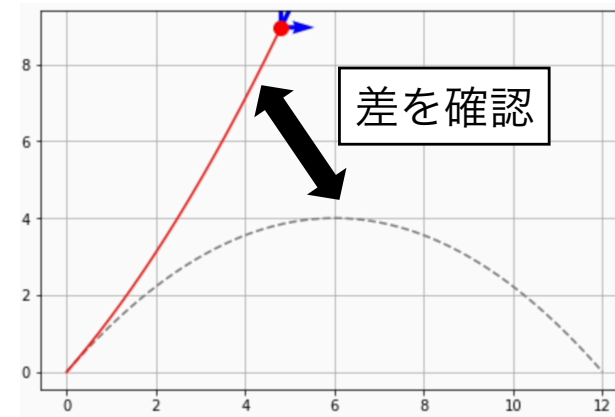
既存のシミュレータは、物理法則から現象を数式で記述し、シミュレーションに変換するまでを教育者が行なっています。学習者はシミュレーションを見ることで現実の物理現象を確認することができますが、それと物理法則との関係性は授業などで学ぶのみです。

# SimSym : 物理系を定義させるシミュレータ

学習者の定義に基づくシミュレーション



7



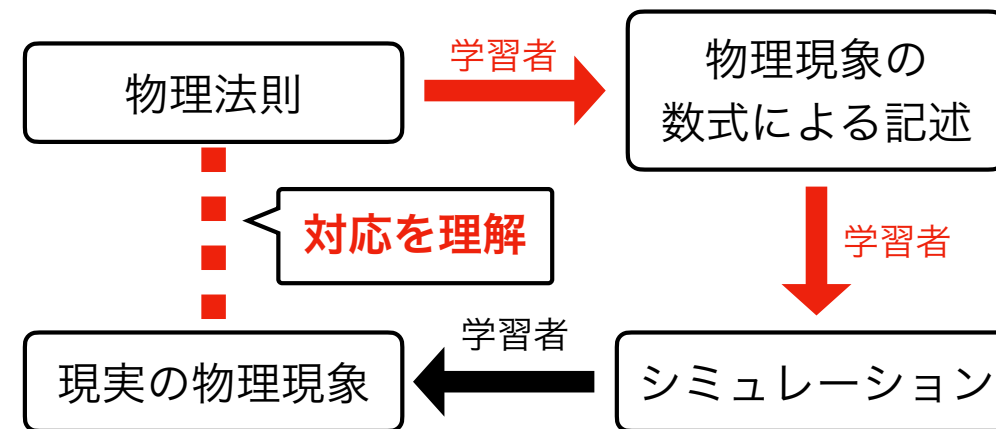
そこで本研究では、現実の物理現象と物理法則の関係性を理解できるシミュレータとして SimSym を提案します。

SimSym では、学習者が物理系を定義します。

つまり、学習者が定義した物体と方程式に基づく運動をシミュレートします。

また、現実の運動を正しく表現した動作例が存在し、これとシミュレーションの結果を比較することで、誤った定義をした際に気付くことができます。

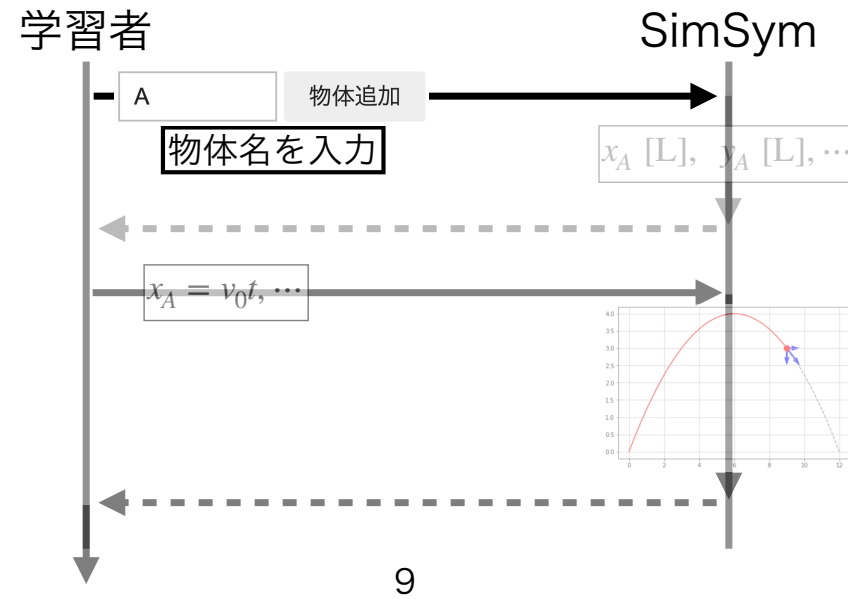
## SimSym による理解



SimSym では、物理法則から現象を数式で記述し、シミュレーションに変換するまでの工程も学習者が行います。これにより、従来は授業などで教わる必要があった現実の物理現象と物理法則の間の対応を学習者自らの経験を通して理解することができます。



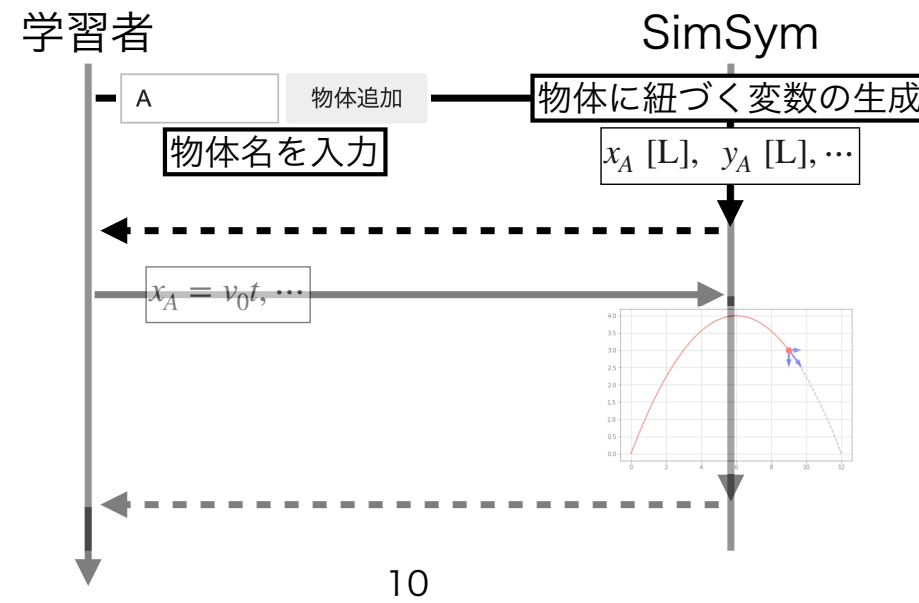
# SimSym を利用する流れ



それでは、SimSym を使う流れを説明します。

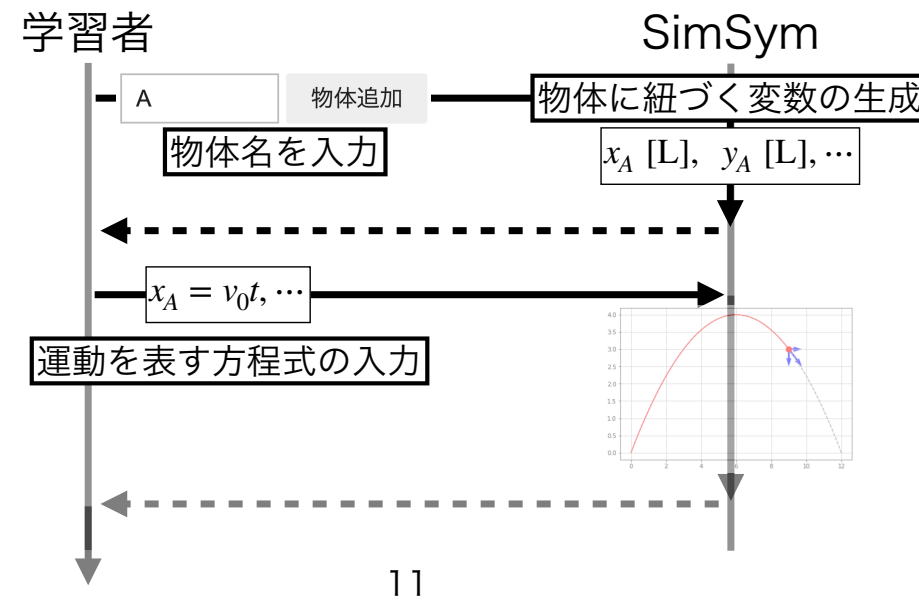
学習者が物体名を入力すると、

# SimSym を利用する流れ



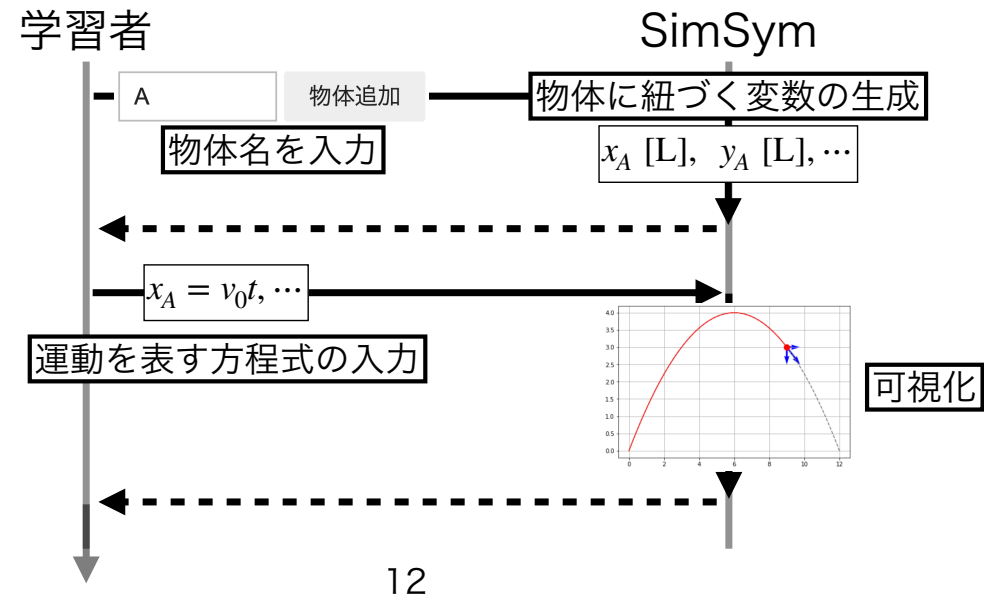
SimSym はその物体と、位置・速度などその物体に紐づく変数を生成し、学習者に提供します。

# SimSym を利用する流れ



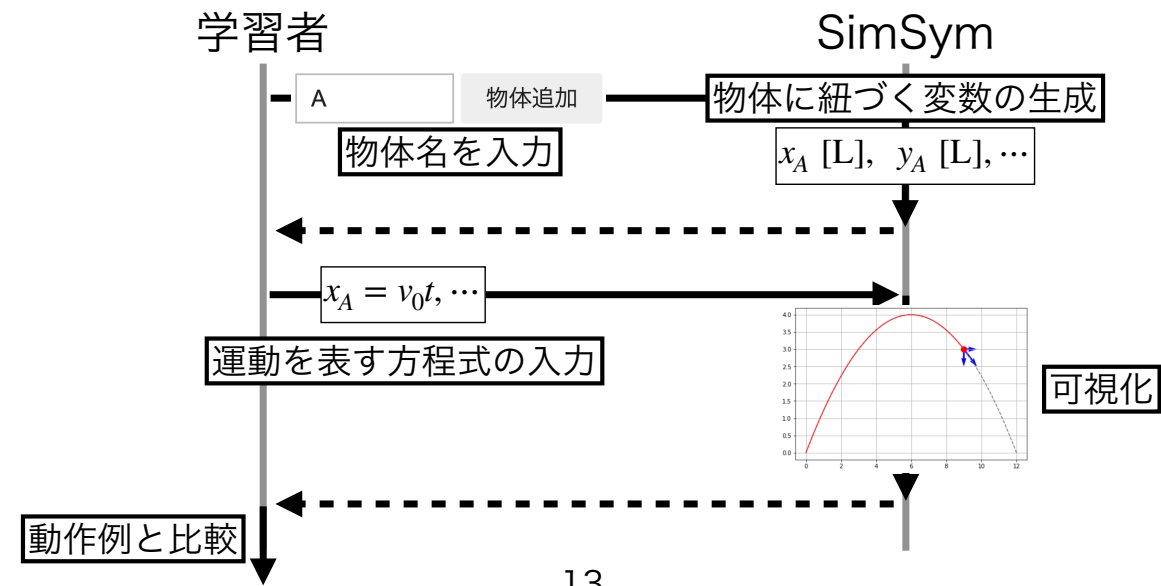
生成された変数を用いて運動を表す方程式を入力すると、

# SimSym を利用する流れ



その方程式に基づいて物体の運動が可視化されます。

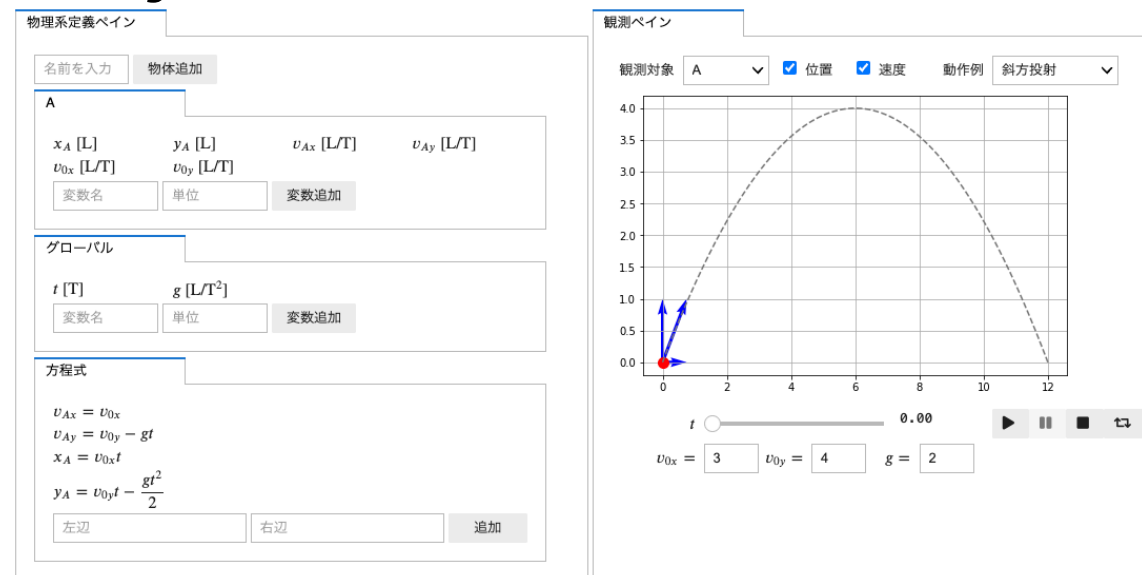
# SimSym を利用する流れ



13

学習者は、可視化された運動と動作例を比較することができます。

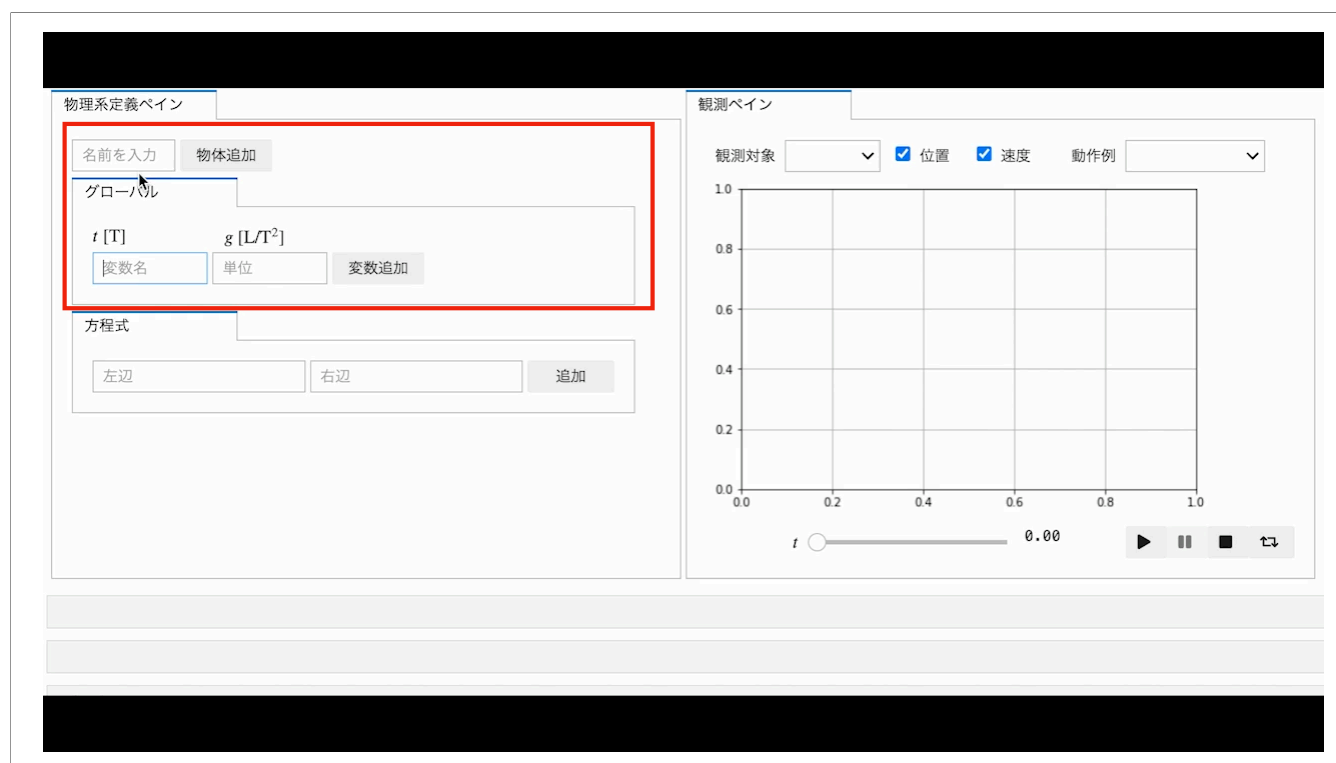
# SimSym の動作例：斜方投射



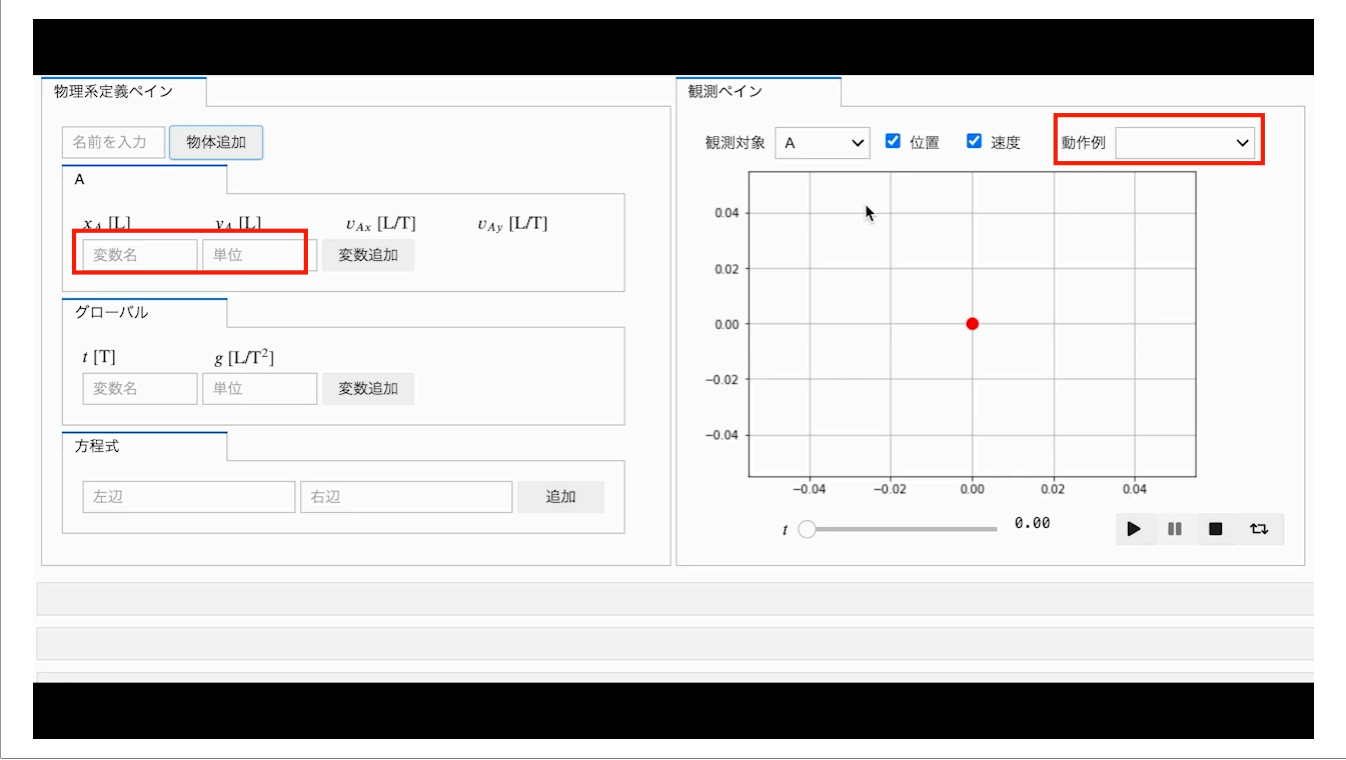
14

SimSym の具体的な動作例を見ていきます。今回は題材として、斜方投射を考えます。

こちらが入力が終わった完成図です。左半分が物理系を定義するペイン、右半分が観測を行うペインです。

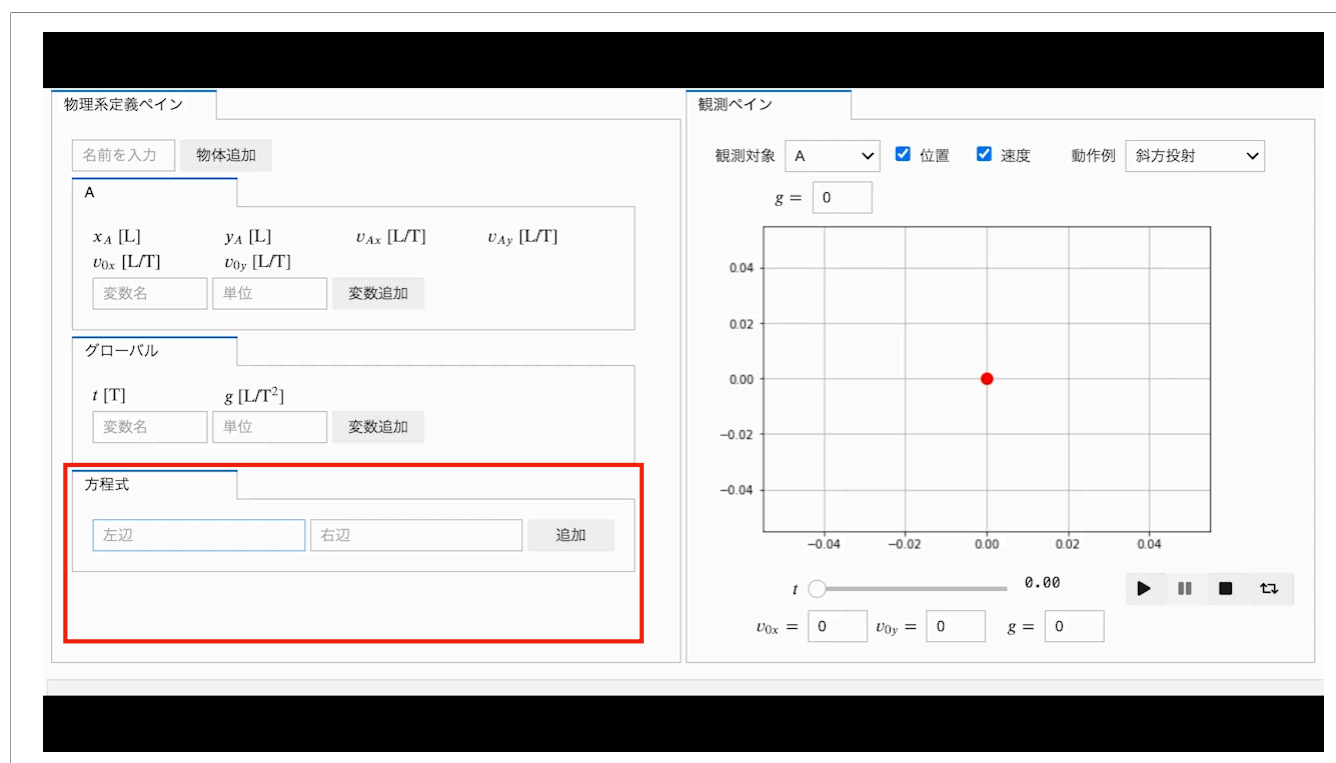


まず、物体名を A として物体を追加します。すると、物体 A に紐づく変数として位置と速度が自動で生成されます。

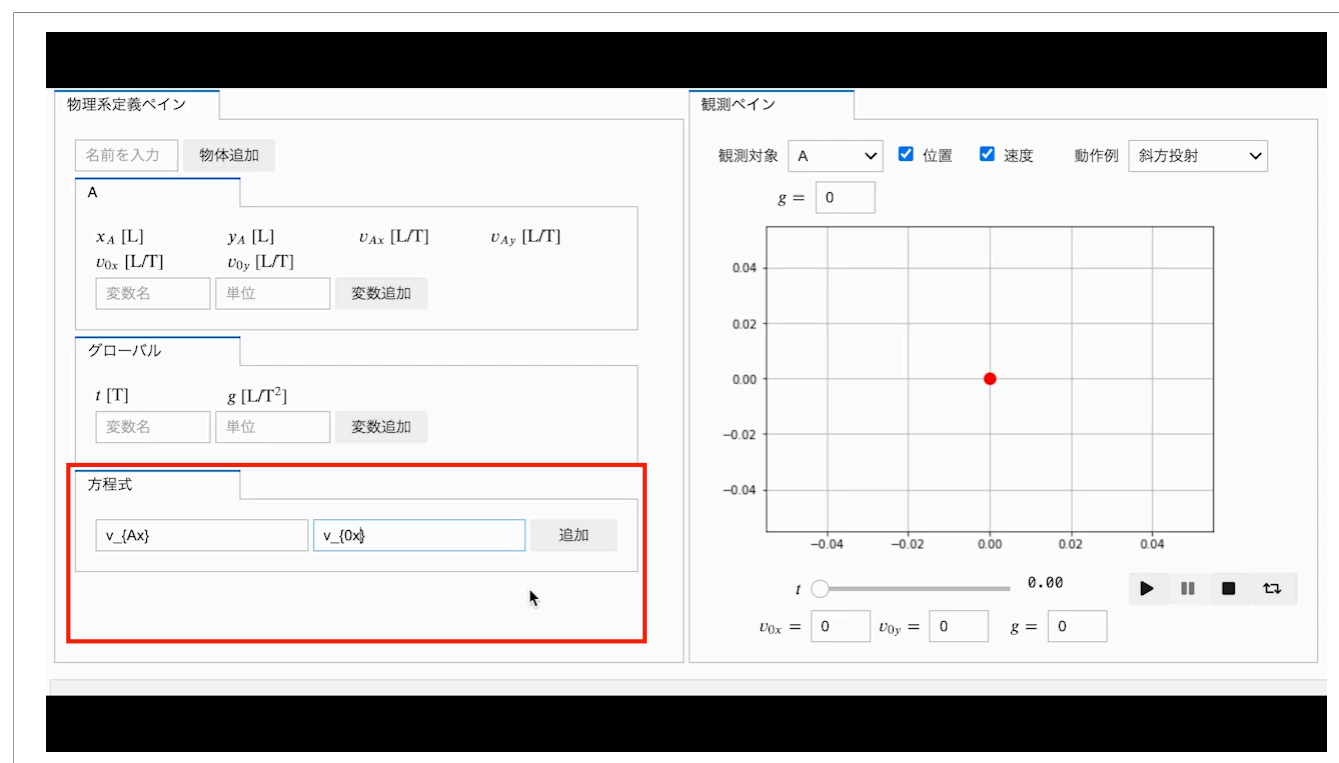


斜方投射の動作例を選択すると、x 軸方向の初速  $v_{0x}$  と、y 軸方向の初速  $v_{0y}$  が追加されます。

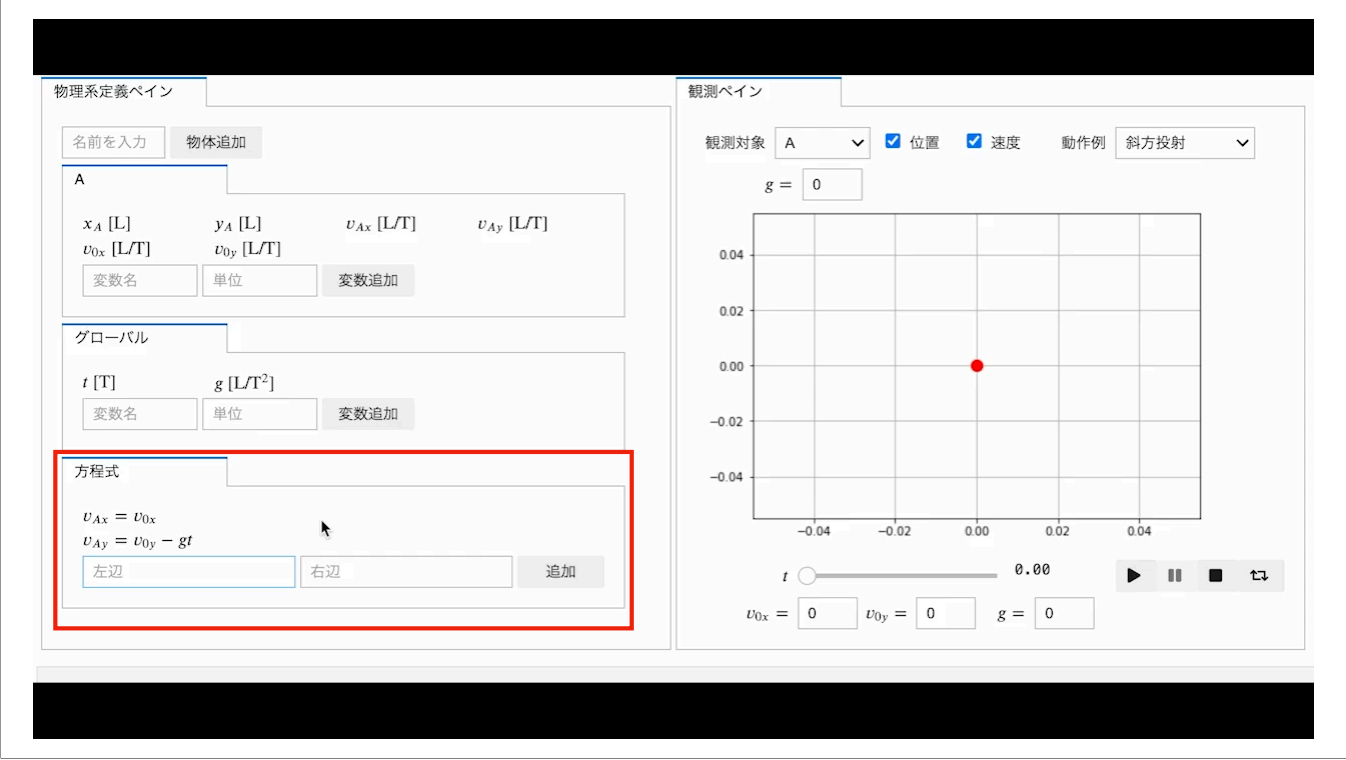




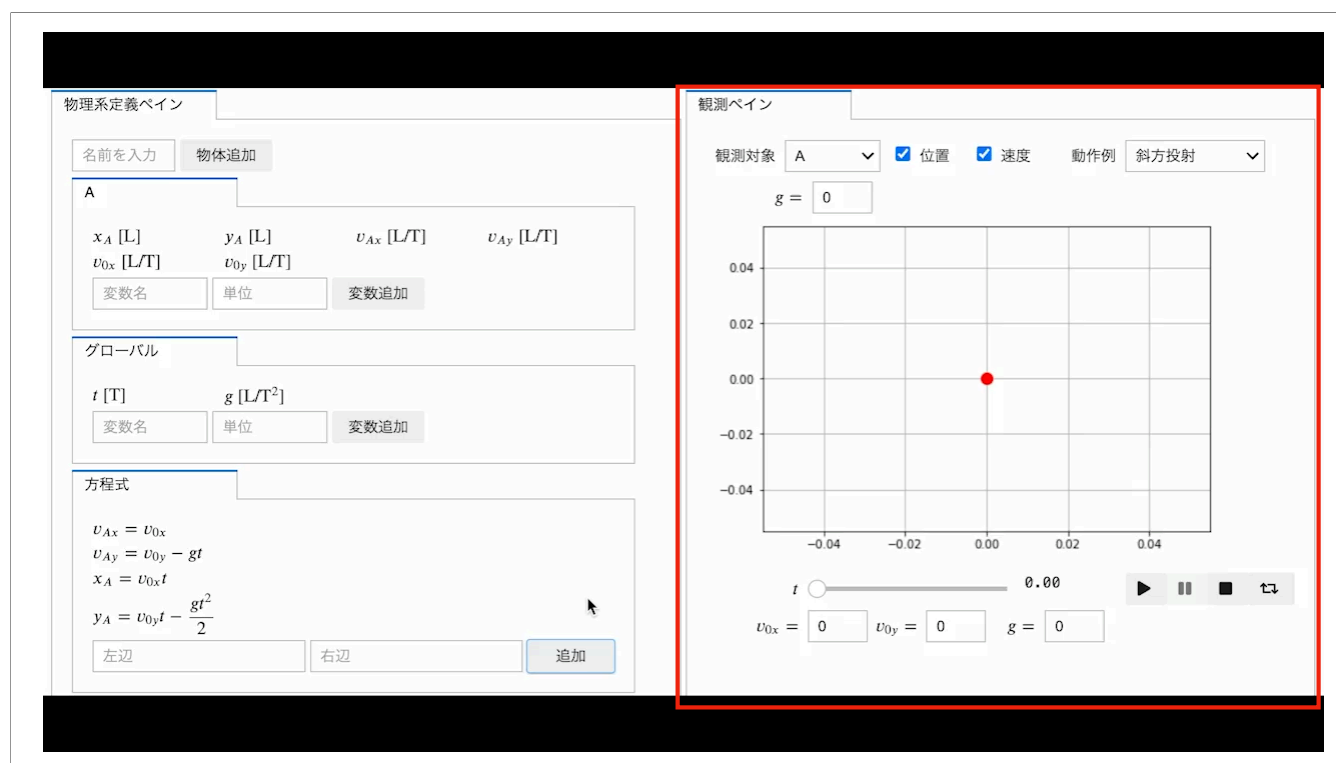
次に、追加した物体の速度・位置を表す方程式を立式していきます。  
x 軸方向は等速直線運動なので、 $v_{Ax} = v_{0x}$  となります。



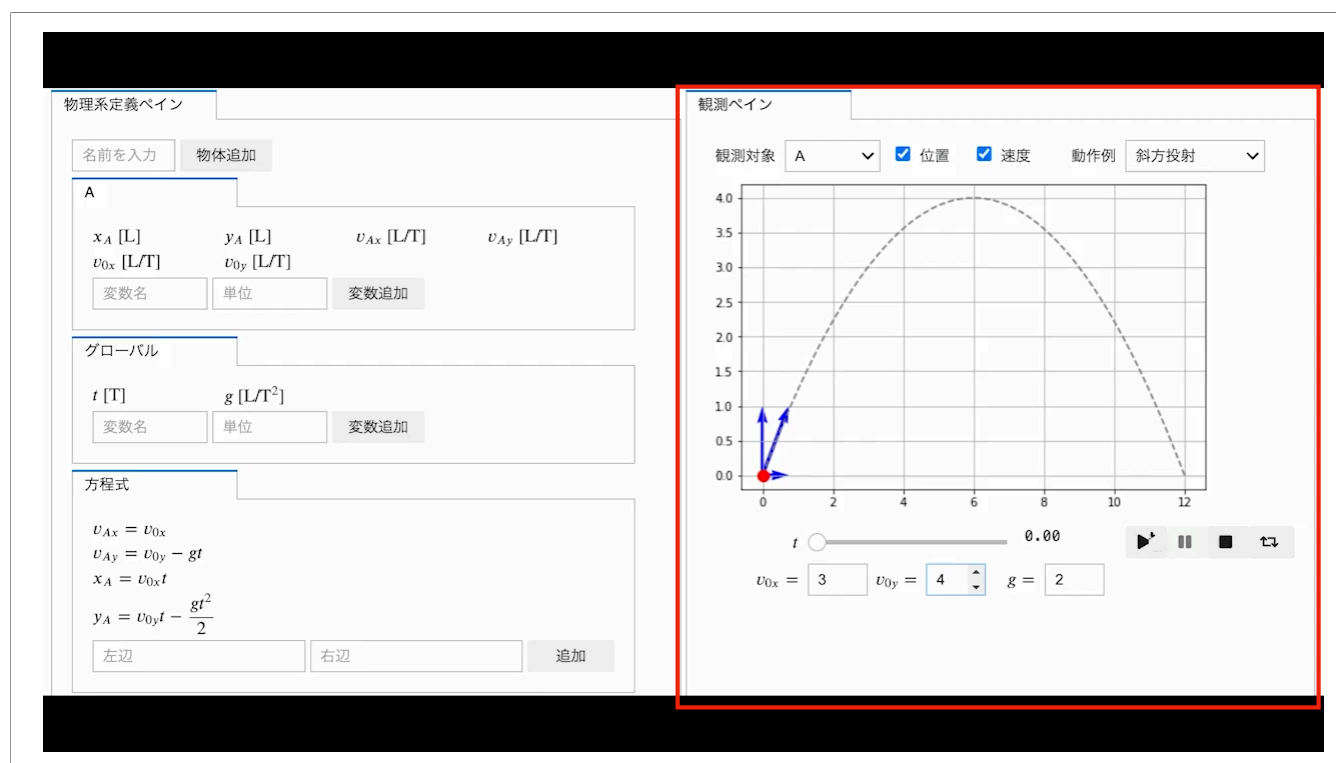
y 軸方向は重力加速度がかかるので、 $v_{Ay} = v_{0y} - gt$  となります。



同様に、x 座標と y 座標も定義します。



観測ペイン下部に、シミュレーションの描画に必要な変数が表示されるので、それらに数値を入力します。  
この際、入力した数値に応じて破線で表されている動作例も変化します。



この状態で時刻  $t$  を動かすことで、このようにシミュレーションが実行されます。軌道が動作例の破線と一致していることがわかります。以上が SimSym 上で斜方投射を定義する例です。

## 誤り1：符号のミス

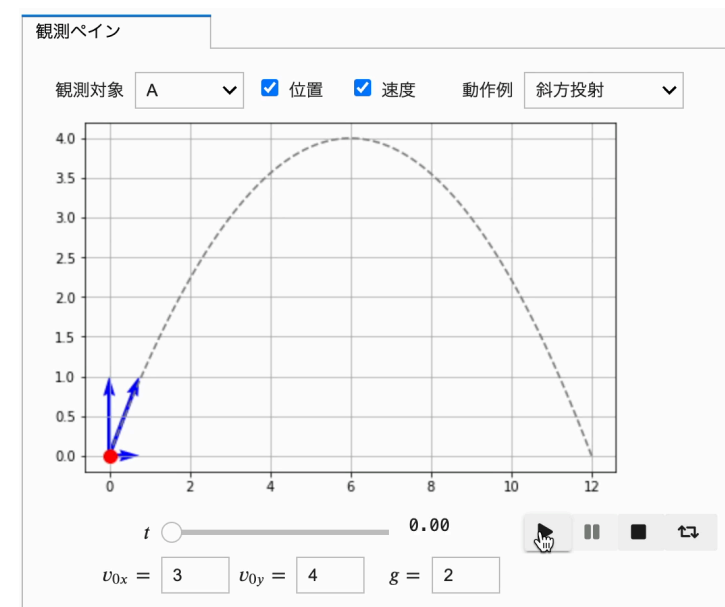
方程式

$$v_{Ax} = v_{0x}$$

$$v_{Ay} = v_{0y} \boxed{+} gt$$

$$x_A = v_{0x}t$$

$$y_A = v_{0y}t \boxed{+} \frac{gt^2}{2}$$



22

また、SimSym によって誤った定義をした際に気付くことができる例を紹介します。

例えば、y 軸方向の速度と位置の定義で符号を間違えた場合、シミュレーションの結果はこのようになります。これは動作例と大きく異なり上昇しているため、加速度の向きを間違えてしまったのではないかと推測できます。

## 誤り2：不正な次元

$$v_{0x} + t$$

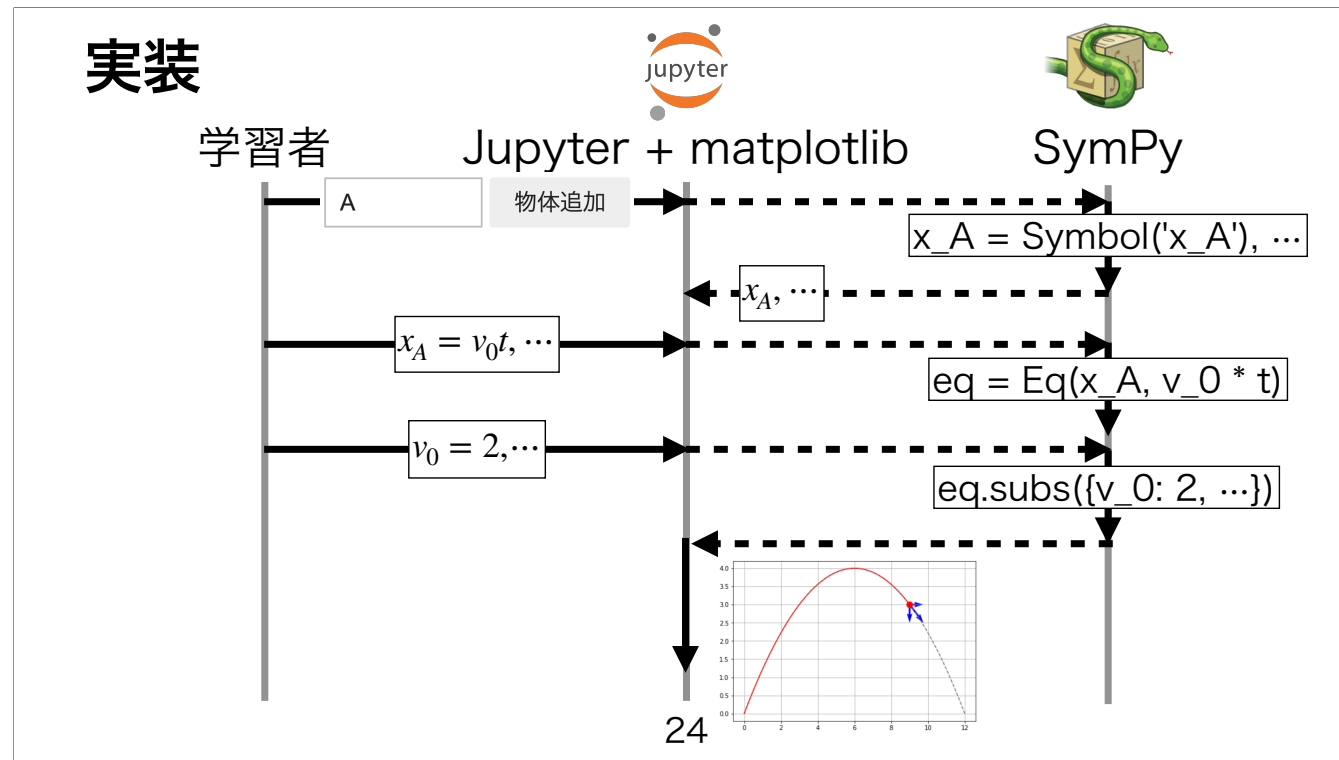
次元が不正なため定義できません:

$$\underline{[L/T]} + \underline{[T]}$$

23

また、速度 + 時間 のように次元が不正な式を定義しようとすると、警告され定義することができません。そのため、このような計算ミスを防ぐことができます。

## 実装

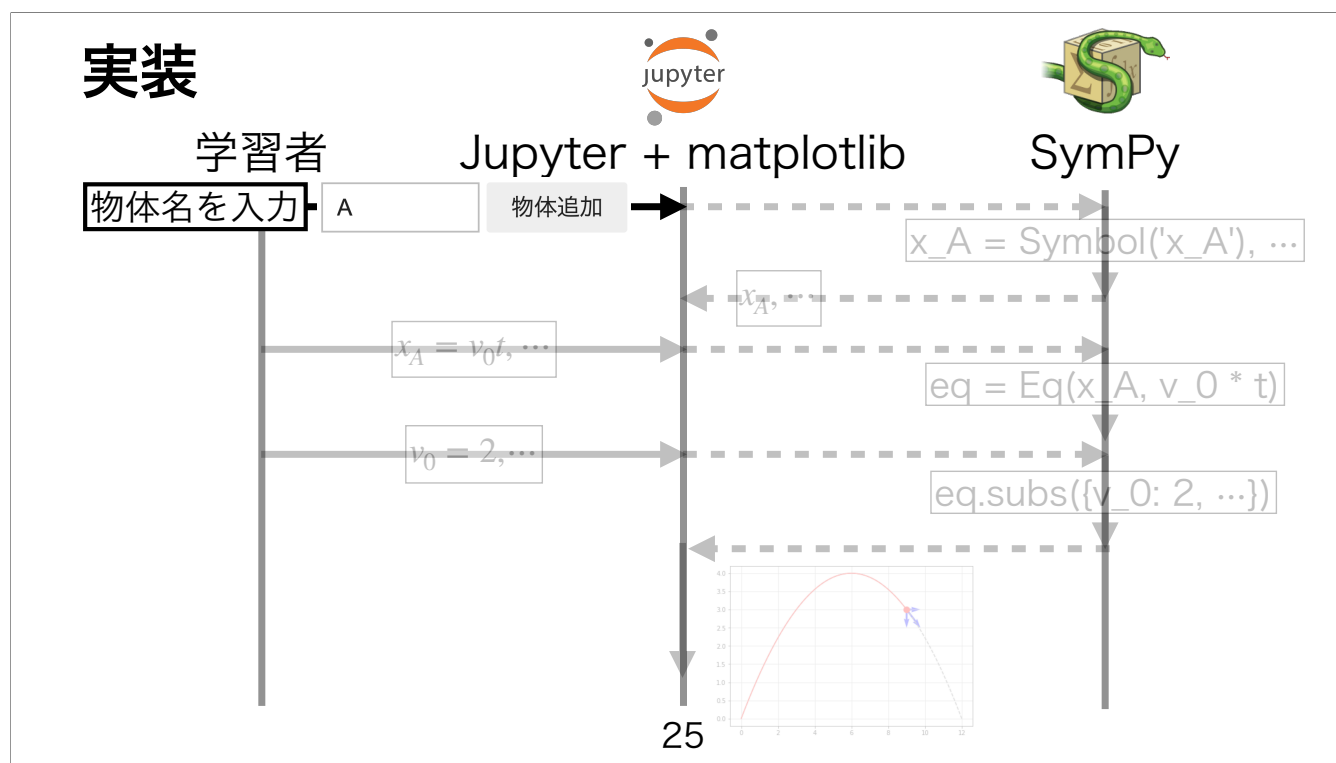


ここからは、実装について説明します。

現在、SimSym は全て Python で実装されています。

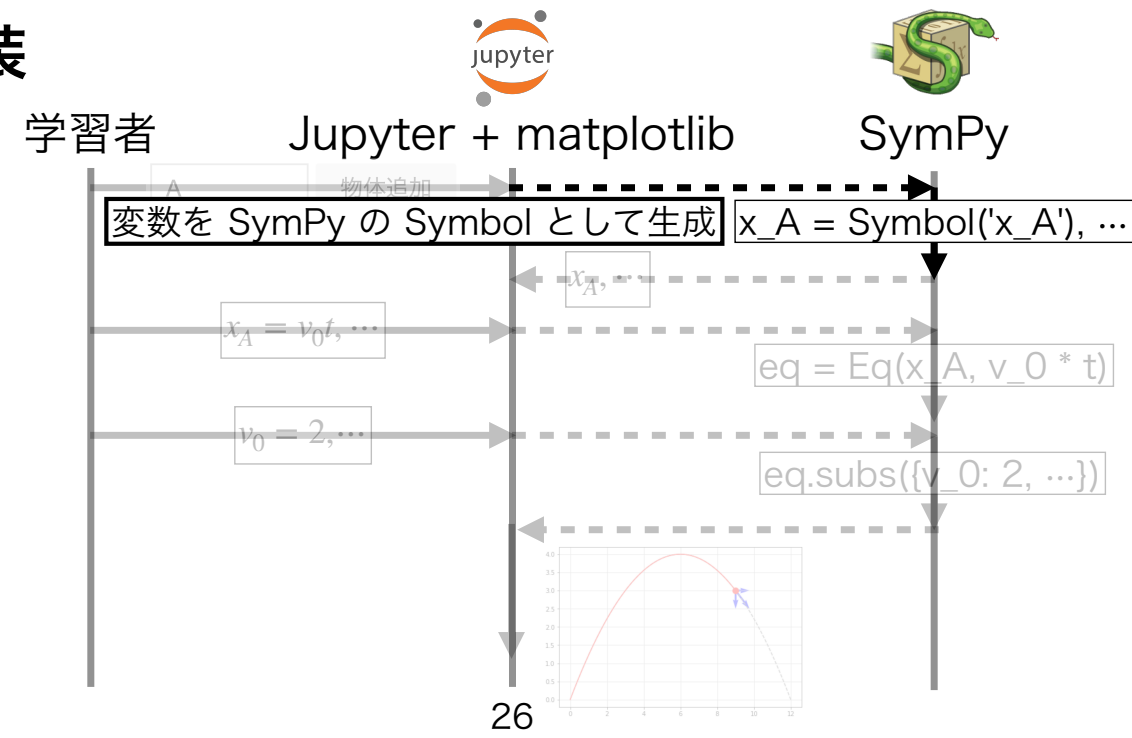
入力と表示に Jupyter Notebook、グラフの描画に matplotlibを、方程式の処理や数値計算に SymPy というライブラリを用いています。





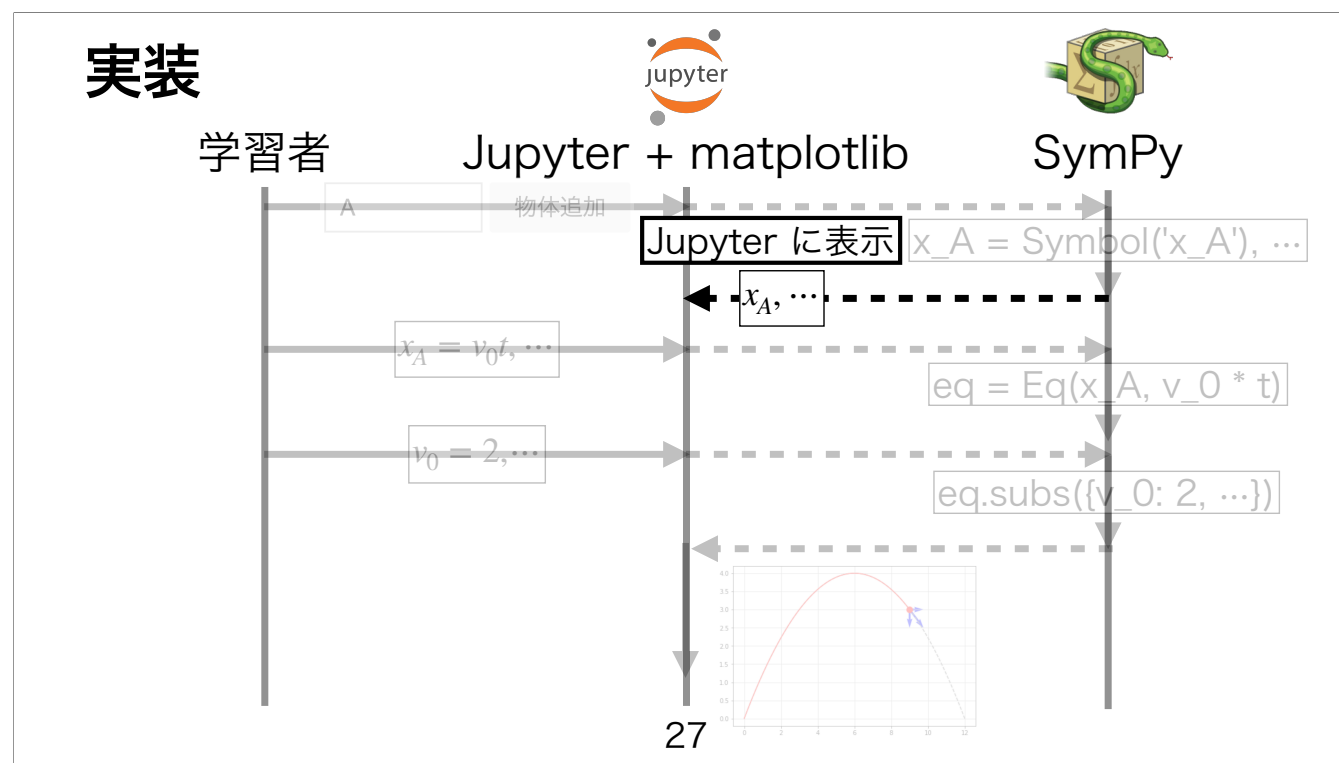
学習者が物体名を Jupyter に入力すると、

# 実装



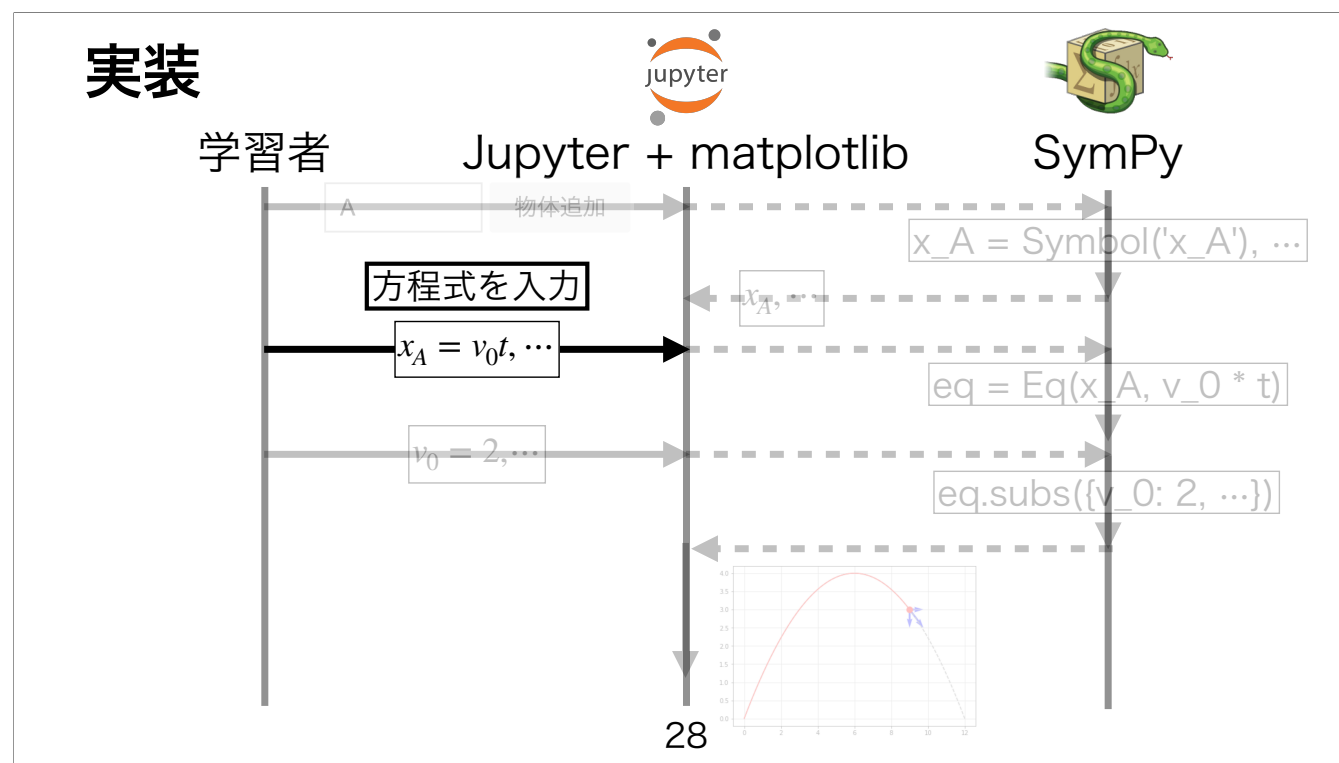
物体とそれに紐づく変数を SymPy で生成します。

## 実装



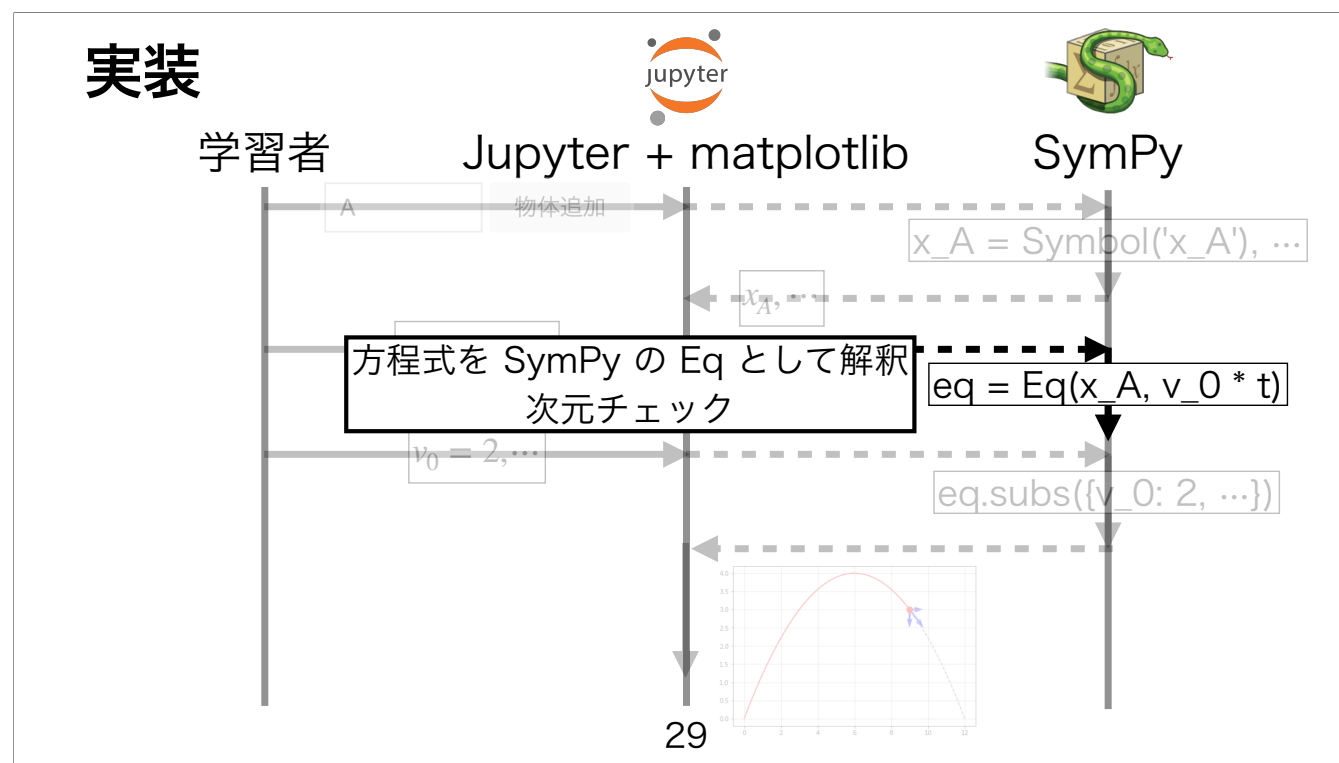
生成された変数は Jupyter 上に表示されます。

# 実装



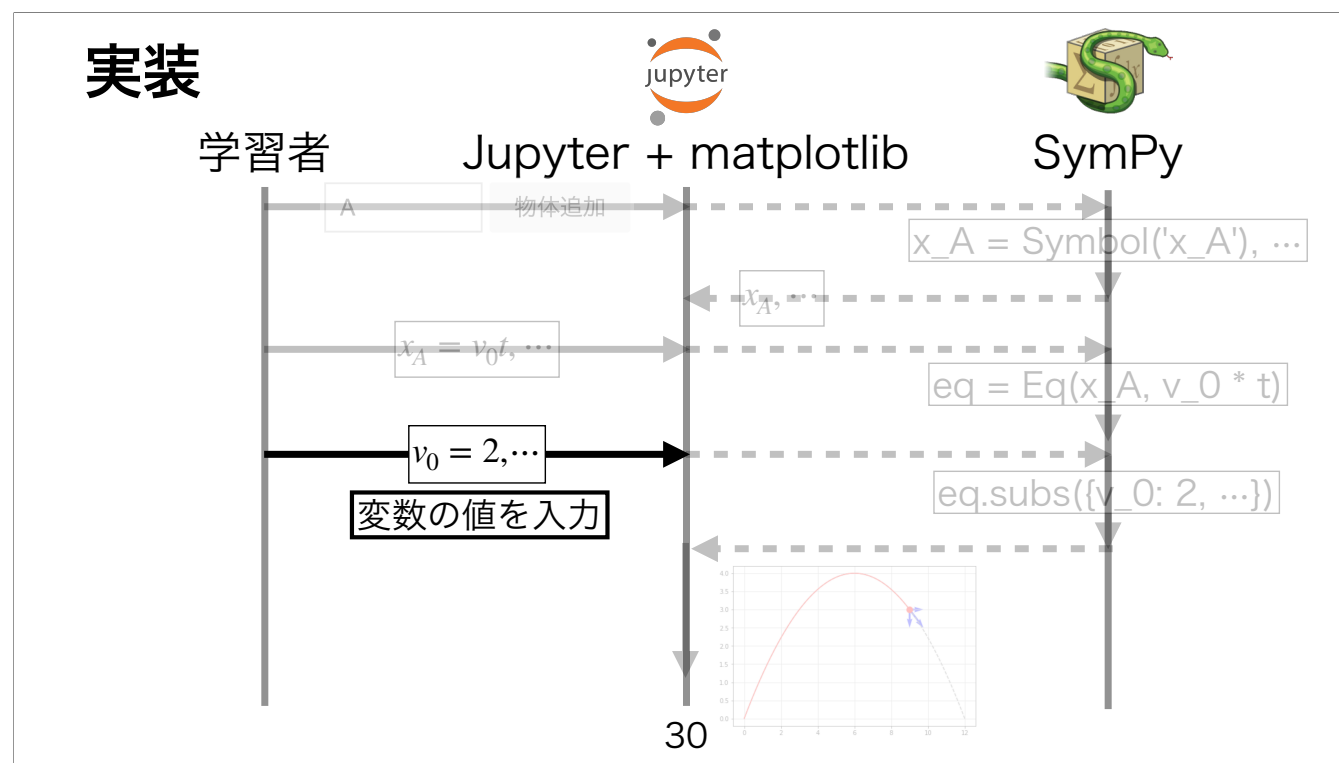
生成された変数を使った方程式を入力すると、

## 実装



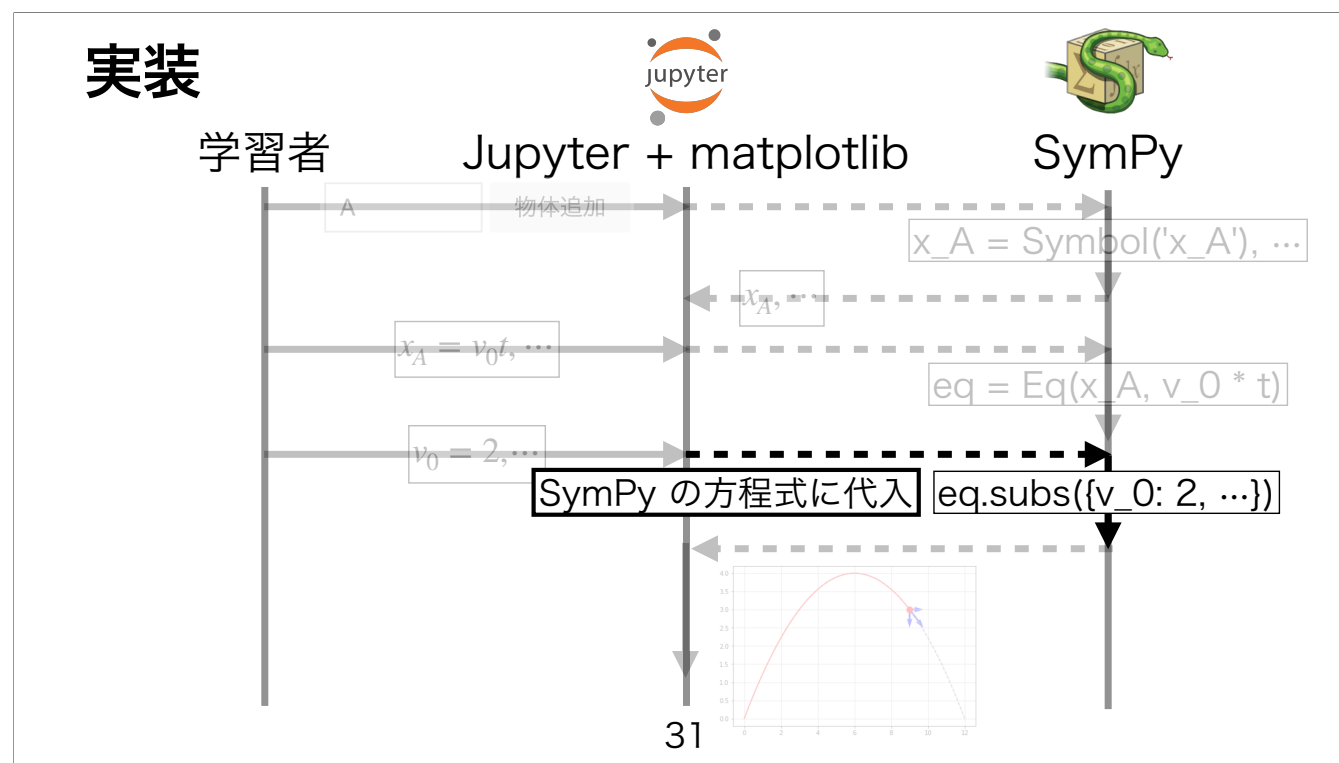
それが SymPy の方程式として解釈されます。この際、次元の検査も行われます。

## 実装

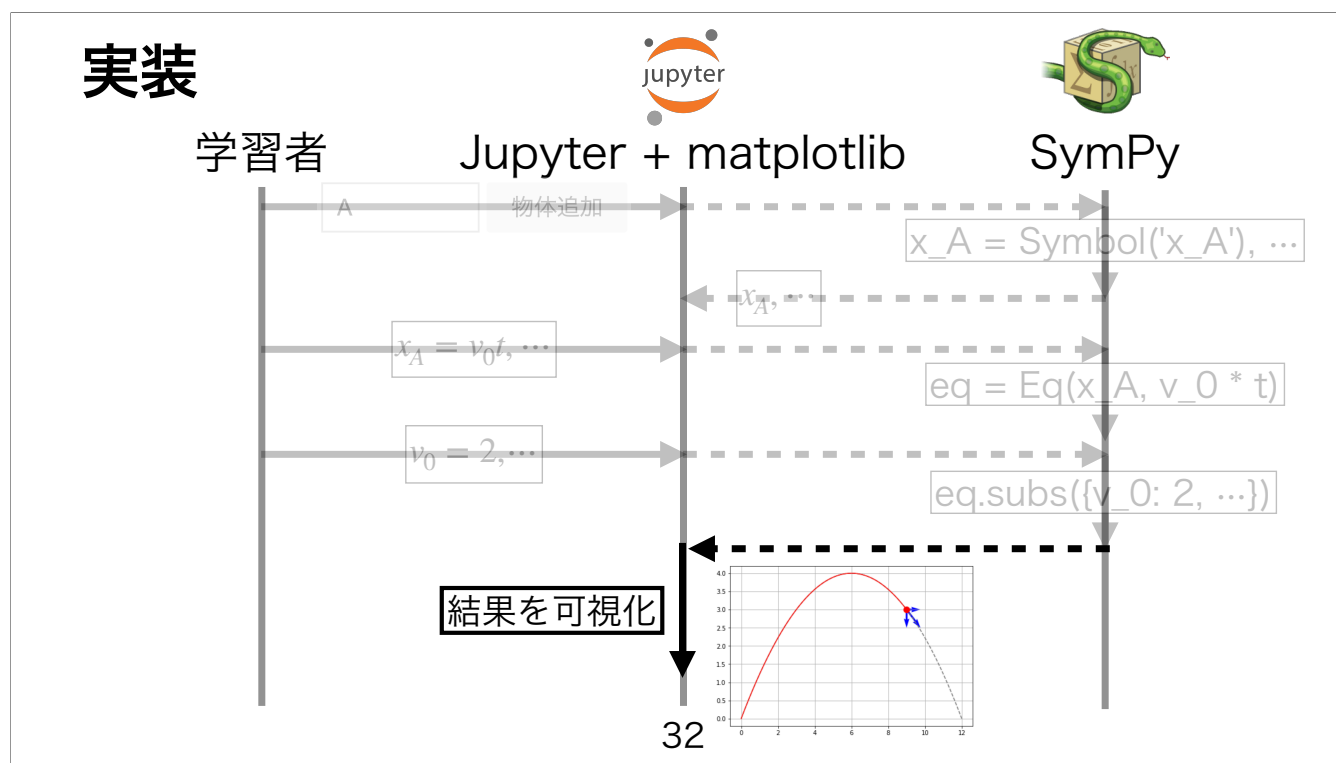


学習者が描画に必要な変数の値を入力すると、

## 実装

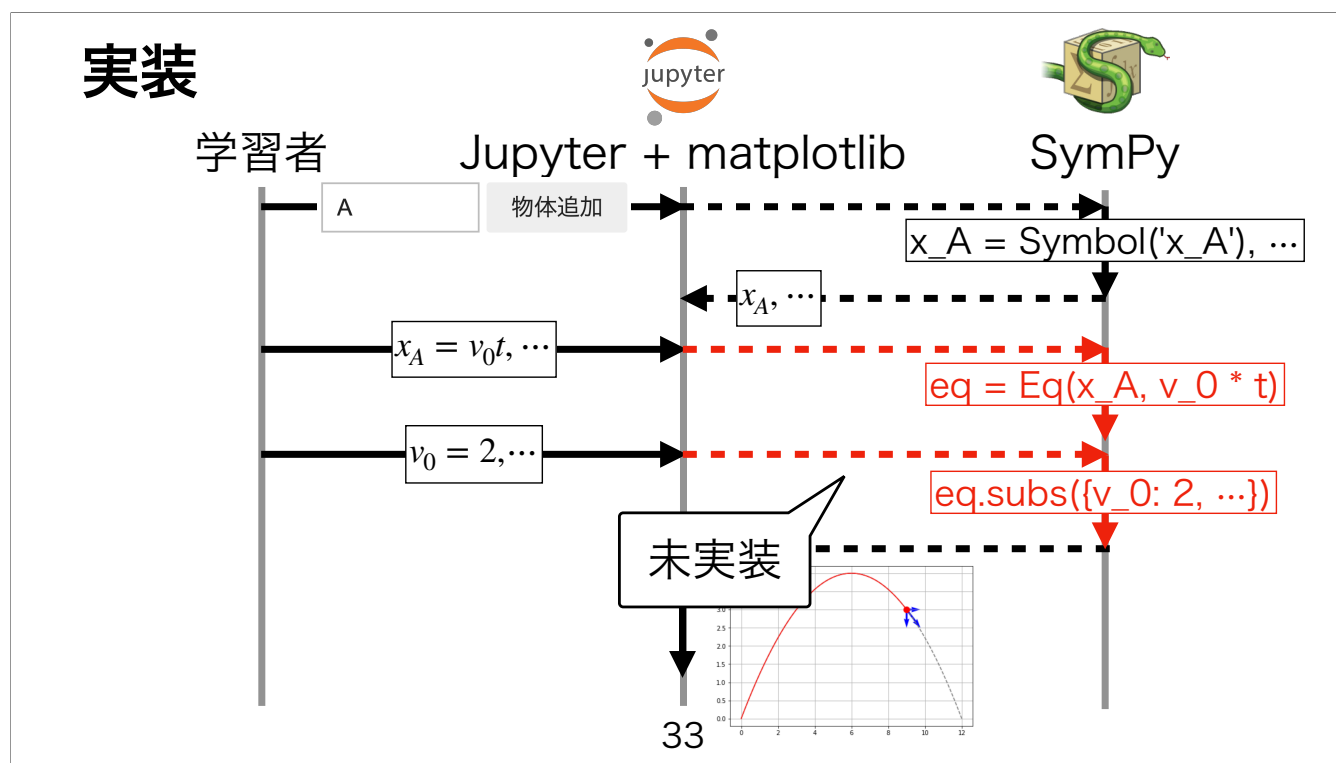


入力した値が方程式に代入され、数値計算が行われます。



その結果を元に、Jupyter と matplotlib で可視化を行います。





なおこれらのうち、入力された方程式を SymPy 上の方程式として解釈する部分、SymPy による次元の検査と数値計算はまだ実装できておらず、先程の例ではあらかじめ手動で入力してあります。

## 今後の課題

- ・実装
- ・発展
  - ・ 頻出する公式の提供・提案
  - ・ 場合分けへの対応
- ・学習効果の評価
  - ・ SimSym / PhET(既存のシミュレータ) で対照実験

34

今後の課題としては、まず先述した「入力された方程式を SymPy 上の方程式として解釈する部分」「SymPy による次元の検査と数値計算」を実装する必要があります。

また、発展させるアイデアがいくつかあります。

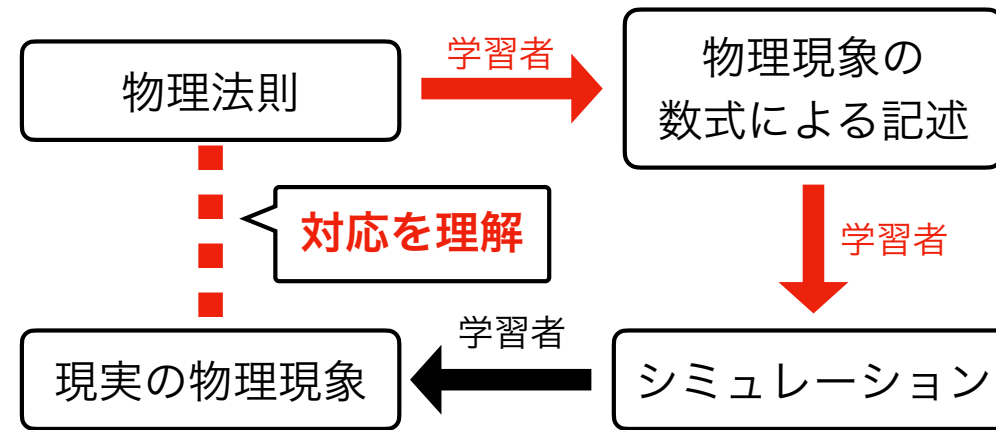
まず、運動方程式や等加速度運動の公式などの頻出する公式を SimSym で提供し、現在存在している変数から使えそうな公式を提案することができれば、学習者の入力の負担を軽減することができます。

また、現在は衝突現象など特定のタイミングで動作が変わるような現象を表現することができません。そのため、時刻によって方程式を変えるなどの場合分けに対応する必要があります。

これらの実装ができれば、学習効果を評価する必要もあります。SimSym を用いる授業と 既存のシミュレータである PhET を用いる授業を行い、テストの結果を比較する対照実験などが考えられます。

## まとめ

物理系を定義させるシミュレータ SimSym の提案



35

まとめです。

本研究では、学習者に物理系を定義させるシミュレータ SimSym を提案しました。

SimSym では、物理法則から現象を数式で記述し、シミュレーションに変換するまでの工程も学習者が行います。これにより、現実の物理現象と物理法則の間の対応を学習者自らの経験を通して理解することができます。

以上で発表を終わります、ご清聴ありがとうございました。

## 参考文献

- Renata Holubova. The impact of experiments in physics lessons – “why, when, how often?”. AIP Conference Proceedings, Vol. 2152, No. 1, p. 030007, 2019.
- Fadil Ajredini, et al. Real Experiments versus Phet Simulations for Better High-School Students’ Understanding of Electrostatic Charging. European Journal Of Physics Education, Vol. 5, No. 1, p. 59, February 2014.
- Katherine K. Perkins, et al. PhET: Interactive Simulations for Teaching and Learning Physics. The Physics Teacher, Vol. 44, No. 1, pp. 18–23, January 2006.