第138回 北数教高等学校部会 数学教育実践研究会

数学教師とコンピュータの30年

— 教材作成から生成AIまで —

清水 団 Dan Shimizu
城北中学校・高等学校 校長
令和8年8月1日（土）・オンライン開催

～2000	2001	2018	2019	2022	2024	2026
Canvas + MathType	pLaTeX	Julia	VSCode	LuaLaTeX	Typst	生成AI

INTRO はじめに

自己紹介

清水 団

Dan Shimizu（しみず・だん）

城北中学校・高等学校 数学科・校長

🏫 johoku.ac.jp
𝕏 x.com/dannchu
📝 zenn.dev/dannchu
💻 github.com/shimizudan

経歴

- 早稲田大学 教育学部 理学科 数学専修 卒業
- 早稲田大学大学院 理工学研究科 数学専攻 修士課程修了
- 城北中学校・高等学校 数学科教員として着任
- 教頭・入試委員長を歴任、ICT教育の環境整備にも尽力
- **2025年4月より校長に就任**

𝕏（Twitter）：@dannchu

数学教育・プログラミングの事例・報告などを発信

興味・関心

数学 ／ テクノロジー ／ Apple製品

AGENDA 本日の内容

本日の内容

1

PART 1 ・ 20分

数学教師としての教材作成遍歴

2000年から2024年まで——Canvas・LaTeX・Julia・Typst、ツールの変遷30年を振り返る

2

PART 2 ・ 25分

LaTeX・Julia・GitHub・Zenn・Typst

各ツールの授業・教材への具体的な活用事例を紹介

3

PART 3 ・ 25分

生成AIと数学教育

可能性・リスク・AI時代の数学教育のあり方を考える

4

PART 4 ・ 15分

Q & A

私の方が色々聞きたいです！

01

数学教師としての教材作成遍歴

20分 — 30年の軌跡を振り返る —

- 教材作成遍歴
- ツール活用事例
- 生成AIと数学教育
- Q & A

PART 1 教材作成遍歴

30年の歩み — 教材作成ツールの変遷

💬 「お正月は新しいことにチャレンジしたくなる」 — Zennより

PART 1 教材作成遍歴

～2000年ごろ：Canvas + MathType

～2000

ドロー系ソフト「Canvas」

- Mac 上で動くグラフィックソフト
- 図形・グラフを**ビジュアルに作成**
- 当時の最先端の教材作成環境

数式エディタ「MathType」

- Word などに数式を貼り込む形式
- 美しい数式の「印刷」を実現
- データが残っていないのが残念…

📝 当時の教材作成の流れ

① Canvas で図形・表を作成
② MathType で数式を入力
③ 組み合わせてプリント完成
④ 印刷・コピー → 配布

資料が残っていないが、**美しく伝えたい**という想いはこの頃から変わっていない

PART 1 教材作成遍歴

2001〜2017年：pLaTeX + TeXShop

2001–2017

Mac 上の pLaTeX 環境

- **pLaTeX**：日本語 LaTeX の標準
- **TeXShop**：Mac 用統合環境
- dvi → PDF の変換ワークフロー
- eps 画像の取り込みも活用

この時代に学んだこと

- コードで数式を記述する力
- 「美しさ」を追求する組版感覚
- 図もいろいろ描ける

```
¥documentclass[a4paper]{jsarticle}
¥usepackage{amsmath, amssymb}
¥usepackage[dvipdfmx]{graphicx}

¥begin{document}

¥[ ¥lim_{x ¥to 0}
   ¥frac{¥sin x}{x} = 1 ¥]

¥begin{align}
  f'(x) &= ¥lim_{h ¥to 0}
    ¥frac{f(x+h)-f(x)}{h}
¥end{align}

¥end{document}
```

約17年間にわたって使い続けた信頼できる環境

PART 1 教材作成遍歴

2018

2018年1月：Atom + Julia の出会い

なぜ Atom へ？

- **Julia言語**を使い始めたのがきっかけ
- テキストエディタを Atom に変更
- LaTeX も Julia も同じ環境で書ける

Julia言語とは

- 科学技術計算向けの言語
- **数学的な記法**に近い文法
- Python 並みの書きやすさ
- C言語に匹敵する速度

```
# Julia での数列の極限
using Plots

n = 1:100
aₙ = (1 .+ 1 ./n).^n

plot(n, aₙ,
  label="(1+1/n)^n",
  xlabel="n", ylabel="値",
  title="e への収束", lw=2)
hline!([exp(1)],
  label="e≈2.718", ls=:dash)
```

※ Atom は後に開発終了 → **VSCode へ移行のきっかけ**に

PART 1 教材作成遍歴

2019

2019年1月：VSCode へ移行

移行の理由

- Atom の将来性に不安
- VSCode 上で **Jupyter Notebook** が使える
- Julia も LaTeX もバッチリ動く
- 拡張機能のエコシステムが豊富

VSCode で実現した環境

- 📝 LaTeX：プリント・テスト作成
- 📊 Julia + Jupyter：数値実験
- 🔧 GitHub 連携：サイトの作成
- 🌐 Markdown：授業メモ

💬 「やっぱりお正月は新しいことにチャレンジしたくなる」
pLaTeX の dvi → PDF ワークフローは引き続き維持

Jupyter Notebook の威力

数式・コード・グラフ・説明を1つのノートにまとめて授業で提示

PART 1 教材作成遍歴

2022

2022年2月：LuaLaTeX へ移行

移行の動機

- フォントを**柔軟に設定**したい
- **UDフォント**（ユニバーサルデザイン）をプリントに使いたい
- dvi ファイルを経由しない直接 PDF 生成へ

実際のところ

- 処理速度は思ったほど速くならなかった
- フォント設定の自由度は大幅向上
- BIZ UDPMincho などが使えるように

```
¥documentclass{ltjsarticle}
¥usepackage{fontspec}
¥usepackage{unicode-math}

% UDフォントの設定
¥setmainfont{BIZ UDPMincho}
¥setsansfont{BIZ UDPGothic}

¥begin{document}

¥[ ¥sum_{k=1}^{n} k
   = ¥frac{n(n+1)}{2} ¥]

¥end{document}
```

> UDフォントで「読みやすさ」が向上 → **特別支援教育の観点からも重要**

PART 1 教材作成遍歴

2024

2024年6月：Typst へ移行 — 現在進行形

移行を決めた理由

① コンパイル速度が圧倒的に速い
② フォントを柔軟に設定できる
③ **Rust** ベースで高い信頼性
④ コードを挿入するのが簡単
⑤ ユーザーが増え、数学対応が充実

奥村晴彦先生も注目
LaTeX の第一人者も Typst に関心を示している

```
// typst のコード例
#set text(
  font: ("New Computer Modern",
         "BIZ UDPMincho"),
  size: 10pt, lang: "ja",
)

// 数式
$ sum_(k=1)^n k = (n(n+1))/2 $

// 列挙
#set enum(numbering: "(1)")
+ 次の問いに答えよ。
+ $7x - 5y = 1$ を解け。
```

Zenn に実践記事を公開中 → zenn.dev/dannchu

PART 1 教材作成遍歴

Typst の特徴 — LaTeX との比較

観点	LaTeX（pLaTeX / LuaLaTeX）	Typst
コンパイル速度	遅い（複数回必要）	**高速**（即座に反映）
学習コスト	高い	**比較的低い**
数式対応	完璧（歴史の重み）	充実してきた
フォント設定	LuaLaTeX で改善	**柔軟・簡単**
パッケージ	膨大	急速に増加中
日本語対応	成熟	改善中
エラーメッセージ	難解	**わかりやすい**

🎯 **結論**：新規プロジェクトは Typst、既存の資産は LaTeX を維持

02

LaTeX・Julia・GitHub・Zenn・Typst 活用事例

25分 — ツールを武器に変える —

- 教材作成遍歴
- **ツール活用事例**
- 生成AIと数学教育
- Q & A

PART 2 活用事例 — LaTeX

LaTeX の授業・教材活用

定期試験・問題作成

- 美しい数式・図形を高品質印刷
- 自由にスタイルファイルを作成できる
- 解答の表示・非表示も切り替えられる

授業プリント・ノート

- PDFで作成すれば、共有も便利
- LaTeXを利用する人にはコードも共有
- 生徒に送るときもPDFは便利

```
% 穴埋め問題の例
¥[ ¥lim_{x¥to 0}
   ¥frac{¥sin x}{x}
   = ¥underline{¥phantom{1}} ¥]

% 解答版（¥phantom を数値に）
¥[ ¥lim_{x¥to 0}
   ¥frac{¥sin x}{x} = 1 ¥]
```

💡 **LaTeX を書く = 数式の構造を理解する**

分数・積分・総和の「入れ子」がコードで可視化される

PART 2 活用事例 — LaTeX

pLaTeX で作成した教材（10点）

教材プリント

↗ ① 数と式：代数の基礎プリント
↗ ② 数と式（解答）：解答付き版
↗ ③ 解の配置：二次方程式の解の位置

定期テスト

↗ ④ 定期テスト：試験問題用紙
↗ ⑤ 定期テスト（解答）：解答・採点基準

演習・フローチャート

↗ ⑥ 漸化式：漸化式の解法まとめ
↗ ⑦ 積分フローチャート：積分計算の手順図
↗ ⑧ 微分計算：微分計算演習プリント
↗ ⑨ 積分計算：積分計算演習プリント
↗ ⑩ 一次変換：行列と一次変換

📐 **コードで記述した高品質な教材**
組版の美しさと再利用性を17年間維持

PART 2 活用事例 — Julia

Julia 言語とは

Julia の特徴

- **科学技術計算**向けに設計
- Python に近い**直感的な文法**
- C言語に匹敵する**実行速度**
- 数学の記法（添字・ギリシャ文字）がそのまま使える

なぜ数学教育に向いているか

```
# 数学の記法がそのままコードに
α = 0.5
∑(n) = sum(1/k^2 for k in 1:n)

# 配列も自然に書ける
A = [1 2; 3 4]    # 行列
```

活用場面

📈 **グラフ・視覚化** — 関数・数列のグラフをリアルタイム描画
🔢 **数値実験** — モンテカルロ法で π を推定、素数・整数問題の探索
📊 **統計・データ分析** — 入試データの可視化・分析

Jupyter Notebook との組み合わせ

数式・コード・グラフを1つのノートに → **授業で即座に使える**

PART 2 活用事例 — Julia

Julia が数学教育にもたらすもの

「計算する」ことで数学が「動く」

```
# π のモンテカルロ推定
N = 1_000_000
inside = count(
  _ -> rand()^2 + rand()^2 < 1,
  1:N
)
π_est = 4 * inside / N
println("π ≈ $π_est")
```

数列の収束・発散を「見る」体験

→

試行錯誤が即座に結果として返る

数学とプログラミングの対応

数学	Julia
Σ（和）	sum, for 文
場合分け	if 文
漸化式	再帰関数
行列	配列 [ ]

🎓 **教員・生徒がともに「探究者」になれる**

「先生も知らない答え」を一緒に探す授業が実現

PART 2 活用事例 — GitHub

GitHub を使った教材・スライドのWeb公開

Marp + GitHub Pages で公開

① slides.md を GitHub リポジトリに push
② GitHub Actions で自動ビルド → HTML スライドを生成
③ GitHub Pages として自動公開 → URL で誰でもアクセス可能

メリット

- ブラウザだけで閲覧可能（インストール不要）
- 更新が即座に反映される
- URL 共有だけで配布完了

GitHub Actions 設定例

```yaml
name: Deploy Slides
on:
  push:
    branches: [main]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - name: Build with Marp
        uses: marp-team/marp-action@v2
        with:
          input: slides.md
      - name: Deploy to GitHub Pages
        uses: peaceiris/actions-gh-pages@v4
```

本スライドも GitHub Pages で公開 → github.com/shimizudan

PART 2 活用事例 — Julia

Julia：発表・教材公開の実績（2022〜2025）

2022–2024

- ↗ 2022/12/31 データの分析（Pluto.jl）
- ↗ 2024/03/10 大学入試（数学）とJulia言語（Quarto）
- ↗ 2024/08/31 高校数学とJulia（Pluto.jl）

2025

- ↗ 2025/03/27 大学入試とJulia言語（Quarto）
- ↗ 2025/08/24–28 Julia言語と高校数学（PDF）

🌐 **すべて Web で無料公開**
Quarto / Pluto.jl などを利用

PART 2 活用事例 — Zenn

Zenn：GitHub 連携で記事を公開

Zenn とは

- 日本の技術者向け記事・本の公開プラットフォーム
- **Markdown** で執筆、数式（KaTeX）・コードハイライト対応
- 無料で誰でも公開可能

GitHub 連携ワークフロー

```
VSCode で .md を編集
    ↓
git push → GitHub リポジトリ
    ↓
Zenn が自動検知・公開
    ↓
zenn.dev/dannchu に即時反映
```

メリット

- ローカルで書いて push するだけで公開
- Git でバージョン管理・差分確認
- VSCode の拡張機能でプレビュー可能
- 記事・本どちらも同じ仕組みで管理

公開先

📝 **公開記事**：zenn.dev/dannchu
📦 **リポジトリ**：github.com/shimizudan

PART 2 活用事例 — Julia × Zenn

Julia × 東大入試数学（2026）：問題をJuliaで考える

6問すべてをJuliaで可視化・検証

```
# 第2問：∫₀¹ |t-x|/(1+t²) dt の最小値
using QuadGK, Optim

f(x) = quadgk(t -> abs(t-x)/(1+t^2), 0, 1)[1]
minf = optimize(f, 0.0, 1.0)
# → x ≈ 0.4142 のとき最小値 0.1882

# 第6問：f(n) = n³+10n²+20n が素数となるn
using Primes
filter(n -> isprime(f(n)), -100:100)
# → [-7, -3, 1]
```

📖 **Zenn 本：** u-tokyo-2026-julia

各問で使ったJuliaパッケージ

問	テーマ	パッケージ
第1問	座標空間の領域	LinearAlgebra / Plots
第2問	積分の最大・最小	QuadGK / Optim
第3問	確率・行列累乗	LinearAlgebra
第4問	自動微分・最適化	Zygote / Optim
第5問	3D回転体	Plots（PlotlyJS）
第6問	素数判定	Primes

> 💡 **数学教育への示唆**
>
> 「解く」のではなく**「考える」ツール**として Julia を使う — 入試問題を通じて多様なパッケージを体験

PART 2 活用事例 — Julia × Zenn

Julia × Zenn：活用事例シリーズ

📝 Zenn に公開した Julia 記事（抜粋）

↗ **積分から定義するπとe**：QuadGK で厳密な定義を数値検証
↗ **サイコロでπ推定**：浜松医大2025入試×モンテカルロ法
↗ **混合分布の誤解**：GMM vs (X+Y)/2 の違いを可視化
↗ **標本標準偏差の期待値**：E(S)＜σ の問題を Julia で確認
↗ **対数近似の危険性**：Measurements.jl で誤差伝播を可視化

📝 続く記事

↗ **クラスタリング**：k-means・DBSCAN・GMM で生徒データ分析
↗ **複素数変換の可視化**：f(z)＝z(z+1) を CairoMakie で描画
↗ **円順列・数珠順列**：バーンサイドの補題を Julia で実装
↗ **Unicode/LaTeX入力ガイド**：¥pi→π：数式をそのままコードに

🎯 **共通テーマ**：「教科書の数学」を Julia で動かし「差し支えない」を「確かにそうだ」に変える

PART 2 Julia 実践 ①

Julia 実践①：積分から定義するπとe

高校では「所与」の定数を積分で定義

```
using QuadGK

# πの定義：sin の弧長の逆関数から
# sin x = 1 となる弧長 = π/2
integrand_pi(t) = 1 / sqrt(1 - t^2)
half_pi, _ = quadgk(integrand_pi, 0, 1)
println("π ≈ ", 2 * half_pi)  # → 3.14159…

# eの定義：∫₁ᵉ (1/t) dt = 1
f(e_cand) = quadgk(t -> 1/t, 1, e_cand)[1] - 1
# 二分法で e を求める（省略）
```

🔢 数値で証明の「感触」を与える

高校生が「なぜπ≒3.14なの？」と聞いたとき、**積分の定義から計算してみる**体験ができる

定数	積分による定義	Julia で確認
π	半円の弧長	2 * quadgk(...)
e	面積が1の底	quadgk(1/t,1,e)=1
sin(π/6)	弧長が π/6 の点	→ **0.5**

✅ 教科書の「証明なし」の事実を**計算で体験する**

PART 2 Julia 実践 ②

Julia 実践②：入試問題をモンテカルロで解く

浜松医科大学 2025年入試問題

問題の核心：6面体サイコロで一様分布を生成し、モンテカルロ法でπを推定せよ

```
function dice_uniform()
    while true
        a, b = rand(1:6), rand(1:6)
        v = (a - 1) * 6 + b   # 1〜36
        v ≤ 32 && return (v - 1) / 32
    end
end

function estimate_pi(N)
    inside = count(1:N) do _
        x, y = dice_uniform(), dice_uniform()
        x^2 + y^2 < 1
    end
    4 * inside / N
end
```

シミュレーション結果と洞察

```
N =     1_000 → π ≈ 3.124
N =    10_000 → π ≈ 3.1488
N =   100_000 → π ≈ 3.14236
N = 1_000_000 → π ≈ 3.14103
```

⚠️ **「試行回数を増やせば精度が上がる」の限界**

誤差は O(1/√N)：精度を10倍にするには試行回数を**100倍**必要とする → モンテカルロ法の本質的な限界を**入試問題を通じて実感**できる

🎓 入試問題を「プログラムで動かす」ことで数学的洞察が深まる授業実践

PART 2 Julia 実践 ③

Julia 実践③：「差し支えない」の意味を確かめる

標本標準偏差の期待値は σ に等しいか？

```
using Statistics

# 母集団 N(0,1), 標本サイズ n=5
n, trials = 5, 100_000
results = [std(randn(n)) for _ in 1:trials]
println("E(S) ≈ ", mean(results)) # → 約 0.94
println("σ    = ", 1.0)
```

教科書：「大標本では差し支えない」
→
小標本では E(S) < σ が数値で明確に！

対数の近似値の危険性（Measurements.jl）

```
using Measurements

# 6^700 の最高位問題
# log₁₀(6) の誤差が 700 倍に増幅される
log6 = 0.77815 ± 0.000005
result = 700 * log6
println(result)
# → 544.705 ± 0.0035
# 小数部の不確実性が大きくなる！
```

🔍 Measurements.jl の力

± 記法で誤差が自動伝播 → 「この近似値を使っても大丈夫か？」を**定量的に評価**できる

PART 2 Julia 実践 ④

Julia 実践④：組合せ論と複素数変換の可視化

円順列・数珠順列（バーンサイドの補題）

```
using Combinatorics, Primes

# 赤4個・白6個の円順列を数える
function enkan(items)
    n = sum(values(items))
    total = sum(Primes.totient(k) *
        multinomial([(v÷gcd(k,n) for
                      (_,v) in items)...]...)
        for k in 1:n) ÷ n
end
# enkan({赤=>4, 白=>6}) → 22通り
# juzu({赤=>4, 白=>6}) → 16通り（数珠）
```

公式の検証を Julia でその場で確認

複素数変換 f(z) = z(z+1) の可視化

```
using CairoMakie

f(z) = z * (z + 1)  # 臨界点 z = -1/2

# 格子点を変換して描画
xs = range(-1.5, 1.5, length=30)
ys = range(-1.5, 1.5, length=30)
```

🎨 CairoMakie で格子の変換を描画

- 横線 → 曲線に変換される様子を可視化
- 臨界点 z = −1/2 付近の「折りたたみ」が視覚的に理解できる
- 複素関数論の「等角写像」の直感を養う

PART 2 Julia 実践 ⑤

Julia 実践⑤：生徒データのクラスタリング

40人のテスト結果を分析する

```
using Clustering, CairoMakie

# 数学・英語の点数（模擬データ）
scores = rand_student_scores(40)  # 省略

# k-means クラスタリング（3群）
result = kmeans(scores, 3)
labels = assignments(result)

# DBSCAN（密度ベース）
db = dbscan(scores, 0.5, 5)
```

4つのクラスタリング手法を比較

手法	特徴	授業での活用
k-means	群数を指定	習熟度別指導
DBSCAN	密度ベース	孤立した生徒の発見
階層型	樹形図	類似度の可視化
GMM	確率的所属	境界の曖昧な分類

🏫 教育的意義

「どの手法が正しいか」でなく「**目的に応じた手法の選択**」を学ぶ — データサイエンスの本質

PART 2 活用事例 — Julia

Julia の強み：数式をそのままコードに

LaTeX 形式でのUnicode入力

```
# ¥pi<Tab> で入力
π ≈ 3.14159265358979

# ¥theta<Tab>, ¥alpha<Tab>
θ = π/6
α = sin(θ)    # → 0.5

# 集合記号・演算子も入力可能
∑(x for x in 1:10)    # ¥sum<Tab>
∈, ∉, ⊆, ∪, ∩         # 集合記号
≤, ≥, ≠, ≈            # 比較演算子
```

数学との対応がほぼ 1:1

数学の記法	Julia コード
π	π
$\theta = \pi/6$	$\theta = \pi/6$
$\Sigma_{k=1}^{n} k$	sum(1:n)
$\forall x \in S$	all(x -> f(x), S)
$\sqrt{2}$	$\sqrt{2}$

✅ **生徒にとっての利点**

数学ノートの記法に近いコードを書ける → **「翻訳」の壁が低い** → 数学的思考をそのまま表現できる

PART 2 活用事例 — Typst

Typst の活用 — 数学テスト作成の実践

テスト作成の全体像

```
// ページ・フォント設定
#set page(paper: "jis-b4",
          flipped: true)
#set text(
  font: ("New Computer Modern",
         "BIZ UDPMincho"),
  size: 10pt, lang: "ja")

// 問題番号の設定
#set heading(numbering: "1. ")

// 2段組レイアウト
#columns(2)[
  = 次の問いに答えよ。
  + $7x - 5y = 1$ を解け。
]
```

Typst で実現したこと

問題用紙

B4横置き・2段組 — 美しい数式を高速コンパイル

解答用紙

設問ごとに解答欄を枠で区切り、レイアウトが自由自在

解答

解答用紙に解答を重ねる形式 — **1ソースから3種類の印刷物**

📖 **Zenn 記事：**数学テスト作成の実践

PART 2 活用事例 — Typst × CeTZ

Typst × CeTZ：Zenn 連載記事シリーズ

2024年6月の Typst 移行後、実践記録を Zenn に公開

📄 共通テスト数学テンプレート

解答欄・丸囲み数字・選択肢番号を Typst で再現

📐 CeTZ で正方形の面積問題

AP=5, BP=3, CP=7 の幾何図形を自動描画

🔵 CeTZ で円と直線

三角関数の最大・最小を座標平面で視覚化

📦 CeTZ で3D立方体

ortho() 関数で立体図形・証明問題を描画

🎯 CeTZ 遠近法の実装

perspective() 関数で奥行きのある立体表現

📈 微分の増減表とグラフ

曲線矢印で凹凸を表現する美しい増減表

PART 2 Typst 実践 ①

Typst 実践①：共通テスト数学テンプレート

Claude Code と協働で開発

共通テスト形式の再現を目指し、Claude Code と対話しながらテンプレートを構築

```
// 解答欄（マーク式）
#let answer-box(n) = box(
  width: 1.5em, height: 1.5em,
  stroke: 0.8pt,
  align(center+horizon)[
    #circle(radius: 0.6em)[#n]
  ]
)

// 選択肢番号
#let choice(n) = circle(
  radius: 0.6em, stroke: 0.5pt
)[#n]
```

実装した機能

機能	内容
answer-box	解答用マス目
choice	丸囲み選択肢番号
circle	丸囲み数字（①②③）
question	問題番号
notice-box	注意書きボックス
dialogue-box	会話形式の問題

LaTeX の exam クラスに相当する機能を Typst で一から設計
→
Typst Advent Calendar 2025 に掲載

PART 2 Typst 実践 ②

Typst 実践②：CeTZ で幾何学図形（正方形）

問題設定

正方形 ABCD の内部に点 P があり、

$AP = 5,\ BP = 3,\ CP = 7$

正方形の面積を求めよ。（答：58）

CeTZ で自動描画

```
#import "@preview/cetz:0.4.2"

#cetz.canvas(length: 2.0cm, {
  import cetz.draw: *
  // 正規化で点Pの座標を計算
  line("A","B","C","D","A")
  line("A","P"); line("B","P")
  line("C","P"); line("D","P")
  content("P", anchor:"east")[P]
})
```

コードで図形が描ける！

- 数値を変えれば図が自動で変わる
- 直角マーク・ラベルの位置も計算
- LaTeXのepsファイル不要

テスト問題への応用

幾何問題の図を Typst コード1つで生成 — 問題・解答・図を同一ソースで管理

normalize() 関数で方向ベクトルを計算 → 比例した座標を自動配置

PART 2 Typst 実践 ③

Typst 実践③：CeTZ で座標平面（円・直線）

問題：三角関数の最大・最小

$\sin\theta - \cos\theta$ の最大・最小を求めよ

ただし $\sin^2\theta + 2\sin\theta\cos\theta \leq 1$

答

$$-1 \leq \sin\theta - \cos\theta \leq \sqrt{7}/2$$

CeTZ で単位円と条件領域を描画

```
// 単位円（赤い半円）
arc((-1,0), start: 180deg, stop: 0deg,
    radius: 1, stroke: red)

// 条件領域（水色で塗りつぶし）
arc((0,-1), start: 270deg, stop: 90deg,
    radius: 1,
    fill: rgb(200,240,255,150),
    stroke: blue)

// 最大・最小を与える直線
line((-1.5,-0.5),(1.5,0.5))
```

座標平面・円・条件領域を数値で記述 → **美しい図が完成**

PART 2 Typst 実践 ④

Typst 実践④：CeTZ で3D立方体

問題：立体図形の証明

立方体 ABCD-EFGH において、対角線 AG は平面 BDE に垂直であることを証明せよ

```
#import "@preview/cetz:0.4.2"

#cetz.canvas({
  import cetz.draw: *
  ortho(x: 20deg, y: 30deg, {
    // 3D座標 (x,y,z) で頂点を定義
    line((0,0,0),(1,0,0))  // AB
    line((0,0,0),(0,1,0))  // AD
    // 隠れた辺は破線
    line((0,0,0),(0,0,1),
      stroke: (dash: "dotted"))
    // 平面BDEを半透明で表示
    fill(aqua.transparentize(70%))
  })
})
```

ortho(x:20deg, y:30deg) で

数学の教科書と同じ**斜投影図**

活用場面

- 立体の証明問題の図
- 切断面を色分けで強調
- ベクトル問題の図示

技術ポイント

3D座標 (x,y,z) をそのまま使える → 座標の計算が数学と1対1で対応

PART 2 Typst 実践 ⑤

Typst 実践⑤：CeTZ で遠近法の実装

標準の ortho() では足りない

数学的な斜投影図ではなく**実際の見え方（透視図法）**で描きたい！

perspective() 変換を自作

```
// 透視変換：奥のものほど小さく
#let perspective(pos, d: 3) = {
  let (x, y, z) = pos
  let scale = 1 / (1 + z / d)
  (x * scale, y * scale)
}

// Y軸まわりの回転
#let rotate-y(pos, angle) = {
  let (x, y, z) = pos
  let c = calc.cos(angle)
  let s = calc.sin(angle)
  (x*c + z*s, y, -x*s + z*c)
}
```

遠近法の原理

`scale = 1 / (1 + z/d)`
d が小さいほど遠近感が強い／奥の面から順に描く（後ろ→前）

数学と情報の融合

教科書の「一点透視図法」を Typst で数式的に実装

プログラミングで**「なぜ遠くのものは小さく見えるか」**を数学的に説明できる

PART 2 Typst 実践 ⑥

Typst 実践⑥：微分の増減表とグラフ

問題：三角関数の微分

$$y = \cos 2x - 2\sin x \quad (0 \leq x \leq 2\pi)$$

増減表を書き、グラフを描く

曲線矢印で凹凸を表現

```
// 上に凸で減少する矢印
#let arrow-right-down =
  box(cetz.canvas(length: 1em, {
    import cetz.draw: *
    arc((0,0),
        start: 90deg, stop: 0deg,
        radius: .8,
        mark: (end: ">",
               fill: black,
               scale: 0.5),
        stroke: 0.5pt)
  }))
```

4種類の曲線矢印

増加凸・増加凹・減少凸・減少凹を矢印の曲がりで表現 — 凹凸まで一目でわかる増減表に

増減表に組み込む

```
#figure(
  table(
    columns: 20,
    align: center + horizon,
    [$y$], [$1$],
    arrow-right-down,
    [極小値], ...
  ),
  caption: [増減表]
)
```

→ Typst Advent Calendar 2025 に掲載

03

生成AIと数学教育

25分 — AIとどう向き合うか —

- 教材作成遍歴
- ツール活用事例
- **生成AIと数学教育**
- Q & A

PART 3 生成AIと数学教育

生成AIで作ったもの — Claude Code（VSCode）を活用

Zenn の記事全般

ほぼすべての記事を Claude Code と協働で執筆 → **実演します**

正規分布のTシャツ作り

正規分布曲線をデザインしたオリジナルTシャツを Claude Code でコード生成 → **実演します**

🔧 共通の開発環境

VSCode + Claude Code（Claude Sonnet）

おにぎりリュックサック

↗ ゲーム発表スライド
↗ 夏合宿サイト

Julia と高校数学（夏期講習会）

授業用Webサイトを Claude Code で一から構築
shimizudan.github.io/julia-summer-course

PART 3 生成AIと数学教育

私の ChatGPT 活用法

ちょっと数学の問題をとく

- 解法の確認・別解を手軽に探す
- 計算過程を追いたいときに

```
Q: 次の不定積分を求めよ。
   ∫ x² sin x dx

→ 部分積分を2回使った
  ステップごとの解答を即座に確認
```

調べてみるときに使う

- 知らない定理・手法の概要を素早くつかむ
- 文献・キーワード探しの入口として

数学の概念の定義を聞く

- 「○○ってどういう定義でしたっけ？」
- 教科書の言葉と対応させながら確認
- 曖昧な記憶のリフレッシュに

⚠ 答えは必ず自分で検証する — **「それっぽい嘘」に注意**

PART 3 生成AIと数学教育

NotebookLM の活用 — 学校の AI ガイドライン作成

STEP 1　ChatGPT（チャッピー）で壁打ち

ガイドラインの骨子・文言を対話しながら作成 —「こういう場合はどう書く？」を繰り返す

STEP 2　NotebookLM でスライド化

完成した文章をソースとして読み込み、「スライドにして」→ 構造化・整理を自動化。文章だけでは伝わりにくい内容が視覚的に

成果物

城北中学校・高等学校 生成AI利用ガイドライン

NotebookLM とは

- Google が提供する AI ノートツール
- PDF・ドキュメント・URLを「ソース」として登録
- ソースに基づいた Q&A・要約・スライド生成が可能
- **ハルシネーションが少ない**（根拠が明示される）

使い分けのポイント

ツール	向いている用途
ChatGPT	発散・壁打ち・草案作成
NotebookLM	整理・構造化・資料化
Claude Code	コード生成・実装

PART 3 生成AIと数学教育

城北中学校・高等学校 生成AI利用ガイドライン — 概要

基本方針 — AI は教員の専門性を拡張するツール

① **最終確認** — 出力は必ず教職員が事実確認・修正
② **情報管理** — 個人情報・機密情報の保護、著作権遵守
③ **教育的配慮** — 生徒の学びへの影響を常に意識

許可サービス（学校メールアドレスで利用）

ChatGPT／Gemini／Claude／NotebookLM／Canva／Adobe CC
※ **オプトアウト設定（学習利用）は必ず OFF に**

2つの利用区分

	通常利用	機微情報を含む利用
対象	授業案・通信・プレスト	生徒氏名・成績・会議録
環境	学校メールで標準利用	有料版・セキュア環境のみ
ルール	最終確認・私用アカウント禁止	入力は必要最小限

✓ チェックリスト

- 学校用アカウントでログイン
- オプトアウト設定は完了している？
- 出力を鵜呑みにせず自ら確認・修正した？
- ヒヤリハット事例は報告をする！

PART 3 生成AIと数学教育

生成AIのリスクと向き合い方

⚠️ リスク・懸念

- **ハルシネーション**（自信ある誤り）
- 数学的証明の論理的飛躍
- 思考のアウトソーシング化
- 著作権・個人情報の問題
- 評価の難しさ（誰の答えか？）

✅ 向き合い方

- 「使わない」より**「正しく使う」教育**
- AI の答えを**検証する習慣**づくり
- 基礎計算力・論理力は引き続き重要
- 校内ルールの整備と教員間の共有
- 文科省ガイドラインの継続的確認

🎯 **重要**：AI は「使う道具」。**何を学ぶかの判断は人間が行う**

PART 3 生成AIと数学教育

AI 時代の数学教育 — 変わるもの・変わらないもの

変わるもの

- 計算・手続きの「訓練」の意味合い
- 暗記中心の学習スタイル
- 教員の「知識の独占」的役割
- 定型問題の評価（AI が解ける問題）

変わらないもの

- **論理的思考・証明する力**
- 「なぜ？」を問い続ける姿勢
- 数学的美しさを感じる感性
- 人と協働して問題を解く力

AI が「どのように」を担う時代に、

「なぜ・何のために」を問う力こそ数学教育の核心

04

Q & A

ご清聴ありがとうございました

Mail dan-shimizu@johoku.ac.jp

Zenn zenn.dev/dannchu

X @dannchu

本スライドは GitHub で公開しています

github.com/shimizudan

数学教師とコンピュータの30年 — 教材作成から生成AIまで
第138回 北数教高等学校部会 数学教育実践研究会・2026.8.1