

体験授業

数学のおもちゃ箱 KNOPPIX/Math  
体験数学とソフトウェア

村尾 裕一 (情報・通信工学科コンピュータサイエンスコース)

2010/07/18

## 本日のトピック ( topics )

数学のおもちゃ箱 KNOPPIX/Math ( の DVD ) を使って

- 数式の計算 ... 数式処理システム maxima で遊ぶ
- 幾何の作図 ... 対話式幾何学ソフト KSEG や GeoGebra で遊ぶ
- 式の証明 ( 数学的帰納法 ) ... 対話型証明支援システム Coq で遊ぶ

## KNOPPIX/Math とは

- KNOPPIX: Linux をインストールせずに起動・利用可能とする CD/DVD  
⇒ その CD/DVD 1 枚を持ち歩けば, どの Windows PC 上でも同じ環境で利用可能 (あるいは, 仮想マシン上でも → Mac でも)
- 数学関連のソフトウェアや文書を収集し収納したものが KNOPPIX/Math
- ソフトウェアの起動は
  - 画面左下の左端のアイコン → 「 $\sqrt{x}$  Math」のメニューから . あるいは,
  - 画面左下のペンギンのアイコン (KnxmLauncher) をクリックするとアイコン一覧が画面いっぱいに見れたり消えたりするので, そのアイコンから .
- どれが何なのかは, そのメニューの「KNOPPIX-Math-Start」を開くと, 簡単な説明・WEB サイトへのリンク・ドキュメントへのリンクがある

DVD イメージは福岡大の濱田龍義氏らの KNOPPIX/Math Project のページ (<http://www.knoppix-math.org>) から入手可 .

( 今回の配布物は標準の起動法の指定を変更してある )

## 数式処理システム maxima

- 数式の計算を記号のまま正確に行うソフトウェアが数式処理システム  
←→ コンピュータが得意なのは数値処理
- Macsyma : 米国 MIT(マサチューセッツ工科大学) の Project MAC で開発 ('60s~'80s...) されたソフト → DOE Macsyma → フリーソフト maxima
  - 人工知能の研究成果の代表格とも見なされた
  - 様々な先進的なアルゴリズム = 計算法も開発
  - きっかけは, MIT の Slagle による試み ('60 年代初頭) : 不定積分を 発見的に (heuristics) プログラム = 大成功 : 大学初年級程度の問題の 90% 程度を解いた ⇒ 不定積分を計算する Risch の アルゴリズム と Moses (MIT) による実現
- 同様のソフトウェア : Reduce, Risa/Asir, Axiom も収録 .  
商用では Maple, Mathematica など

## 数式処理システム (wx)maxima で遊ぶ

何種類かのインタフェースで利用できる (計算エンジンは同じ) が, wxMaxima を使おう

- 入力欄に式を入力 (して Shift-Enter), または
- 使いたい機能をメニューから選び (「Calculus」→「Integrate ...」(積分) など), 入力ウィンドウで呼び出す機能への引数 (アーギュメント) を入力.

乗算「 $x$  かける  $y$ 」は  $x*y$ , 冪乗「 $x$  の  $y$  乗」は  $x^y$ , 自然対数の底  $e$  は %e, 円周率は %pi と入力する. %in と %on は  $n$  番目の入力と出力の指定 (直前の結果は %でも指定可).

(1) 「%e^(%e^(%e^(%e^(%e^x))))」と入力してみよう.

(1.1) この式 (直前の式) を  $x$  で微分する: 「Calculus」→「Differentiate ...」でそのまま「OK」(または「differentiate(%, x)」と入力).

(1.2) その結果 (直前の結果) を  $x$  で積分してみよう: 「Calculus」→「Integrate ...」でそのまま「OK」(または「integrate(%, x)」と入力).

積分 を色々試してみよう.

(2) (先ずは, 良く知ってるものを)  $\int \log(x)dx$ .

(3)  $\int \frac{1}{x^3-1}dx$  や  $\int \frac{1}{x^4-1}dx$  を求めてみよう.

(3.1) 結果の式を微分して, 元の式に戻るか試してみよう.

(3.2) 元の式と同じ形にはならなかったですね. 通分等の式の変形をして, 式を簡単化 (simplify) しましょう: 左下の「Simplify」をクリックするか, 「Simplify」→「Simplify expression」とする.

Macsyma において, もうひとつの先進的な機能だった (今日では普通) のが 多項式の因数分解 です. いくつか試してみましよう.

(4) `factor(x^4-1)`, `factor(x^8-1)`, `factor(x^20-1)`, `factor(x^20-2*x^10+1)` とか色々試してみてください (「Simplify」→「Factor expression」でも可)

(5) (もう少し難しい例) `nusum(k^2,k,1,n)` とすると,  $\sum_{k=1}^n k^2$  を求めてくれます (または, 「Calculate」→「Calculate sum ...」で, 「Sum of」に式  $k^2$ , 「by variable」に変数  $k$ , 「from」に下限 1, 「to」に上限  $n$  を指定し, Nusum をチェックする).

## 対話式幾何学ソフト KSEG を試す

対話式幾何学ソフト というのは、コンパスや定規を使う代わりに、コンピュータと対話しながら図形を描くソフトウェアです。DVD にはこうしたソフトが何種類も収められていますが、先ず KSEG を使ってみましょう。

### KSEG の使い方

次の 3 点を押さえておけば、あとはやっていく内にわかってくることでしょう。

1. 右クリックで点を描画
2. 左クリックで点・線・円などの（構成部品）のオブジェクトを選択（複数選択する場合には、shift キーを押しながらクリック。ドラッグした領域内のすべてのものを選択することも可能）
3. 新たな図形を作成するために必要な構成部品を選択した後、メニュー（「新規」など）やアイコンボタンで作成

## KSEG を試す (1) : 基本練習

練習のために線分と円を描いてみましょう。

(a) 線分を描く

- (a.1) 適当な場所で右クリック（点が描かれる）
- (a.2) もう一ヶ所別の場所で右クリック（もうひとつ点が描かれる）
- (a.3) Shift キーを押しながら両方の点を左クリックし，2点を選択する
- (a.4) 「新規」→「線分」を選ぶと，線分が描かれる
- (a.5) 描かれた点や線分をクリックして選択し，マウスをドラッグする（左クリックしたまま動かす）と点や線を動かすことができるのを確認

描いた図全体の領域をドラッグして点と線分を選択して「編集」→「オブジェクトを削除」を選び，全部消しましょう。

(b) 円を描く

- (b.1) 適当な場所で右クリック（点が描かれる）
- (b.2) もう一ヶ所別の場所で右クリック（もうひとつ点が描かれる）
- (b.3) Shift キーを押しながら両方の点を左クリックし，2点を選択する（最初にクリックした点が円の中心になる）
- (b.4) 「新規」→「中心と点による円」を選ぶと，円が描かれる
- (b.5) 描かれた点や円をクリックして選択し，マウスをドラッグすると円の大きさが変化したり円を移動することができる



## KSEG を試す (2)

三角形を描いて, その五心を求め, それらの点は三角形の頂点を動かして必ず一点で交わることを確認しましょう. 五心とは次のような点です.

- (c) 三角形を描く : (c.1-3) 点を 3 つ描く . (c.4) その 3 点をドラッグして選択 . (c.5) 「新規」 → 「線分」 . 沢山の線を描くことになりますから , 「編集」 → 「色」 で三角形の線分に適当な色をつけておくと良いでしょう
- (d) 重心 三角形の 3 つの頂点とそれぞれの対辺の中点を結ぶ線が交わる点 .  
中点 : 線分を選び , 「新規」 → 「中点」
- (e) 内心 三角形の 3 つの内角の二等分線が交わる点 .  
角の二等分線 : 3 点を選び , 「新規」 → 「角の 2 等分線」 ( 2 番目の点の )
- (f) 垂心 三角形の 3 つの頂点からそれぞれの対辺に引いた垂線が交わる点 .  
垂線 : 点と線分/半直線/直線をひとつずつ選び , 「新規」 → 「垂線」
- (g) 外心 三角形の 3 辺の垂直二等分線が交わる点 .
- (h) 傍心 三角形のひとつの内角の二等分線と他の 2 つの外角の二等分線が交わる点 . 内角の採り方により 3 つある .

## KSEG を試す (3) : 放物線を描く

放物線は「定直線  $l$  と,  $l$  上にない定点  $F$  から等しい距離にある点の軌跡」と定義される.

- (i.1) X-Y 軸を描く : 2 点を水平に描き, その 2 点を選び直線 (X 軸) を描く (2 点を使うことはないので「編集」→「オブジェクトを隠す»). X 軸に乗らないように (上側に) 点  $F$  を描く. 点  $F$  と X 軸を選び垂線 (Y 軸) を描く.
- (\*) 点の名前  $F$  を表示しておくには : 点を選択し「編集」→「ラベルを表示». 名前を変更するには「編集」→「ラベルを変更」で修正.
- (i.2) 原点  $O$  : X 軸と Y 軸の 2 直線を選び「新規」→「交点」.
- (i.3)  $F$  と原点  $O$  に関して対称な点  $F'$  : 点  $F, O$  をこの順に選択し「変形」→「ベクトルを選択». 原点  $O$  を選択し「変形」→「変換」で表れる点が  $F'$ .
- (i.4) X 軸に平行で  $F'$  を通る直線  $l$  をひく : この点  $F'$  と Y 軸の直線を選択し「新規」→「垂線」.
- (i.5) 駆動点  $P$  : 直線  $l$  上で右クリックし点  $P$  を描く. この点  $P$  をドラッグすると直線  $l$  上を動く.
- (i.6) 線分  $FP$  の垂直 2 等分線をひく : 2 点  $F$  と  $P$  を選び「新規」→「線分」, 「新規」→「中点». この中点と線分  $FP$  を選び「新規」→「垂線」.
- (i.7) この垂直 2 等分線上の点は点  $F, P$  から等距離で, 点と直線の距離という場合, その点は直線の垂線上にあることになるので, 点  $P$  における直線  $l$  の垂線とその垂直 2 等分線の交点が放物線上の点 : 点  $P$  と直線  $l$  を選び「新規」→「垂線». その垂線と垂直 2 等分線を選び「新規」→「交点」.
- (i.8) 放物線の軌跡 : その交点と点  $P$  を選び「新規」→「軌跡». 点  $P$  をドラッグして動かしてみよう.
- (i.9) 点  $F$  をドラッグして動かしてみよう.

## KSEG を試す (4) : 楕円や双曲線を描く

楕円 : 楕円上の点と 2 つの焦点との距離の和が一定

- (j.1) 楕円の焦点となる 2 つの点  $O_1, O_2$  を描く .
- (j.2)  $O_1O_2$  よりも長い線分  $AB$  を描く .
- (j.3) 線分  $AB$  上に点  $P$  を描く (駆動点となる) .
- (j.4) 線分  $AP$  と  $BP$  を描く (図としては変わらないが, 以降で線分の長さを得るのに必要) .
- (j.4) 線分  $AP$  を (何度かクリックして) 選択し, さらに点  $O_1$  を選択した後「新規」→「中心と半径による円」を描く (半径は  $AP$ ) .
- (j.5) 同様にして, 点  $O_2$  を中心とする半径  $PB$  の円を描く .
- (\*) 点  $P$  をドラッグして動かしてみよう .
- (j.6) 2 つの円の交点 : 2 つの円を選択し「新規」→「交点」 .
- (j.7) 軌跡 : 交点の一方と点  $P$  を選択し「新規」→「軌跡」 (距離の和は  $AB$  で一定)

双曲線 : 双曲線上の点と 2 つの焦点との距離の 差 が一定

- $AB$  を線分でなく直線とし,
- 駆動点  $P$  をその直線上で (線分  $AB$  の外) 動かす

## 対話式幾何学ソフト GeoGebra も面白い

- 作図は上部のアイコン ( のメニュー ) から選んだ後 , マウス操作
- 左側に描かれたオブジェクトの一覧 . どれがマウスで選択されているかが一目でわかる

内接円 を描く :

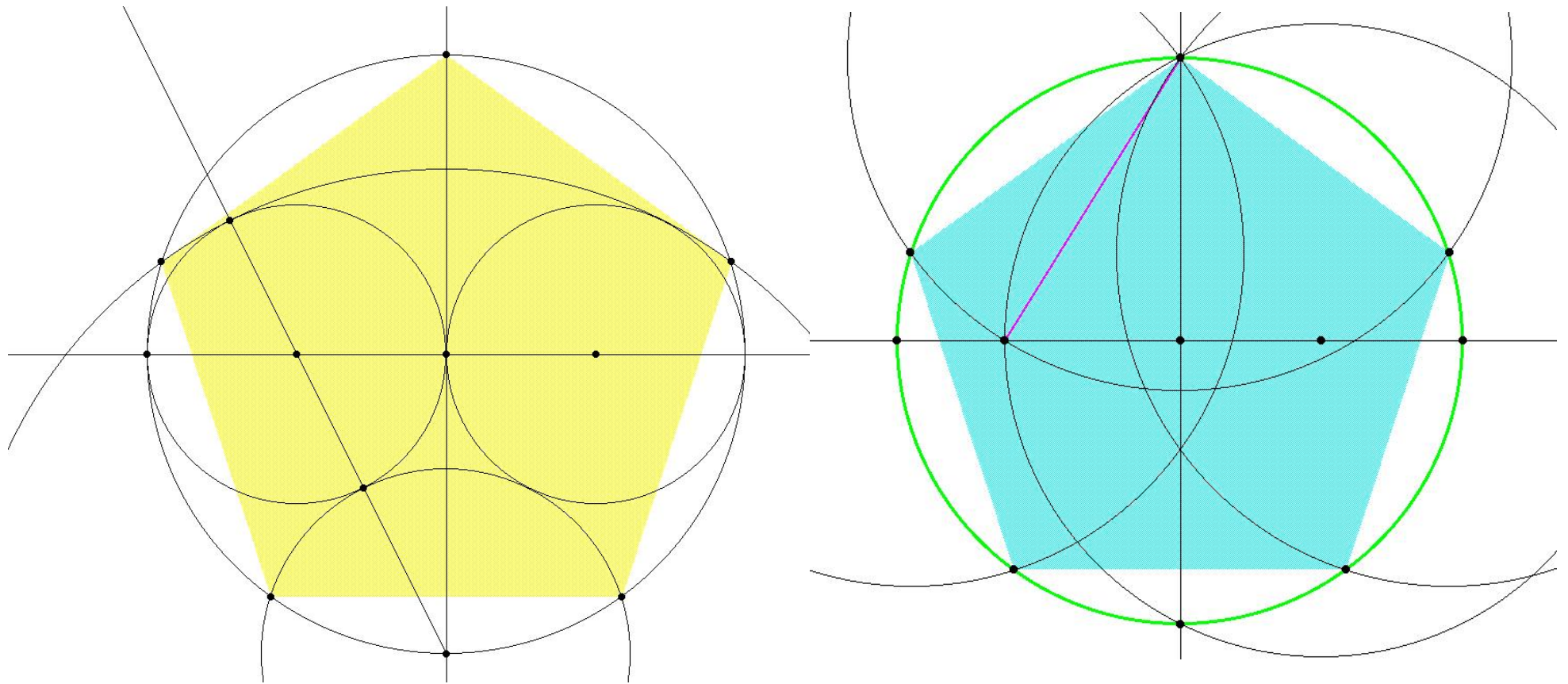
- (1) 5 番目のアイコンの [多角形] で , 3 点をクリックした後 , 最初の点をクリックして三角形を描く
- (2) 4 番目のアイコンのメニューから [角の二等分線] を選ぶ . 3 点をクリックすると 2 番目に選んだ点の内角の二等分線がひかれるので , この二等分線を 3 本ひく .
- (3) 2 番目のアイコンのメニューの [2 つのオブジェクトの交点] で , 2 本の二等分線を選ぶ . 交点が三角形の内心 ( 内接円の中心 ) .

- (4) 4 番目のアイコンのメニューの [垂線] を選び . 内心の点と三角形の一边を選ぶ .
- (5) [2 つのオブジェクトの交点] で , この垂線と選んだ辺の交点を描く .
- (6) 6 番目のアイコンの [中心と円周上の 1 点で決まる円] を選び . 内心の点とその交点を選ぶと内接円が描かれる .
- (\*) 左端のアイコンを選び , 三角形の頂点を選択してドラッグして動かしてみよう .

外接円 を描くには?

3 次元はもっと面白い ( コンピュータは大変 ): Surf とかを試したい...

【問題】正五角形を描くには?



## 機械証明・証明支援システム：Coq を試す

「証明をする」 = 「事実を元に、論理的に正しいことが既知である方法により、別の事実へと結びつけていく」.

- 問題 (扱うデータ, 操作内容および解法) を形式化
- 三段論法のような我々が正しいと信じて (理解して) いる論理 (logic) に基いて演繹・推論: 機械的に (機械証明) or 自動化したい  $\implies$  最終的なより所は定義と公理
- (例) 数ですら, 先ずは定義:
  - 〔自然数の定義〕 0 は自然数である; ある自然数を 1 だけ増やすたものも自然数である ( $n$  を自然数とすれば,  $(\text{Suc } n)$  も自然数.  $\text{Suc}$  は successor (= 後にくるもの) の略)
- 代表的なソフトウェア: Isabelle, HOL, Coq など
- 〔応用〕 アルゴリズムの正当性, ソフトウェアの検証, プロトコルの安全性, ...
- 幾何の定理を自動証明するソフトウェアも存在する

## 証明支援システム：Coq を試す $2 \sum_{k=0}^n k = n(n+1)$ の証明

- 級数の和  $\sum_{k=0}^n k$  を表す「sum n」を帰納的に定義する: ( $\text{Suc } p$  は「S p」と表す)

```

Fixpoint sum (n:nat) {struct n} : nat := (* {struct ...} は「nについて再帰」を示す *)
  match n with
  | 0 => 0 (* n が 0 ならば 0 *)
  | (S p) => (S p) + (sum p) (* n が (S p), つまり p+1 ならば (p+1)+(sum p) *)
  end.

```

- 事実 : `Theorem sumof_1ton: forall x:nat, 2*(sum x) = x*(x+1).` ..... `sumof_1ton` が名前
- 帰納法 (induction) による証明 : `induction x.` とすると, 次の2つを示せばよいことになる.
  - $x=0$  の場合 : 「 $2*(\text{sum } 0) = 0*(0 + 1)$ 」 `auto.` で OK.
  - $(S x)$  の場合 : 「 $2*(\text{sum } x) = x * (x + 1)$ 」という仮定 (IH<sub>x</sub> と名付けられる) の元で 「 $2*(\text{sum } (S x)) = (S x) * ((S x) + 1)$ 」 が成り立つこと.

補題を (Theorem の前に) 準備 :

```

Lemma sub: forall n:nat, sum (S n) = (S n) + (sum n)
  auto.
  Qed.

```

```

rewrite sub. (* 示したい論理式をこの sub を用いて書き換える *)
              (* 左辺は「2*((S x) + (sum x))」になる *)
Require Import Arith. (* 既存のものを導入 *)
rewrite mult_plus_distr_l. (* その中の規則 (多分, 分配則) を用いて書き換える *)
                          (* 左辺は「2*(S x) + 2*(sum x)」になる *)
rewrite IHx. (* 前提条件 IHx を用いて書き換える *)
              (* 左辺は「2*(S x) + x*(x + 1)」になる *)

Require Import ArithRing.
ring. (* 算術式の整理 *)
Qed.

```

## 数式処理関係の話題

世界のあちこちで研究が行われているが、日本では...

- 日本には数式処理専門の学会がある『日本数式処理学会』 <http://www.jssac.org/> (対象は学部卒業以上．教育分科会というのもあります)
- 高校での教育現場での利用例や生徒による研究例も：関西学院高等部，神戸大学発達科学部附属，...
- JSEC(Japan Science & Engineering Challenge) 高校生“科学技術”チャレンジ：2008年の文部科学大臣賞は
  - 「ヨセフス問題とパスカルの三角形」(関西学院高等部の生徒3名)
    - パズルや幾何に見られる新事実の発見
    - 数式による定式化
    - Mathematica を用いた可視化と証明
- Rubik キューブと群論 ← 群論に特化した数式処理システム GAP の応用．  
福岡教育大の藤本先生のページ <http://www.fukuoka-edu.ac.jp/fujimoto/rubik.html> 参照
- 数独を数式を使って解く試み (条件を数式で記述 → Gröbner 基底)  
⇔ 情報処理では backtrack して (試行錯誤で) 解くのが普通 (?)