

Teach Yourself KMC

独習KMC vol.4

京大マイコンクラブ 著



Hello, World!!
This is the 4th magazine of KMC
Contents are :

当たり判定

プログラミングコンテストの開き方

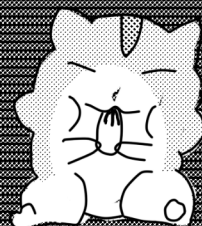
ミャンマー旅行と2012年中国反日示威活動中

の中国で同人誌刷ってきた記

色々と未経験な人に送るKMC新入生体験記

Fortran戦記 流体編

(漫画)KMC部員の日常



はじめに

この度は『独習 KMC vol.4』を手にとって頂きありがとうございます。この本は、京都大学のコンピュータ系サークルである京大マイコンクラブ*1の日頃の活動の様子やメンバーが持っている知識などを世に知ってもらうために制作されたものです。

我々 KMC の活動は、新勸期に配るビラでは「コンピュータを用いた様々な活動」と表現されるようになかなか広範であり、具体的に列挙してみるとゲームやツール制作、プログラミングコンテストへの挑戦、Web サービスの制作、イラスト制作、DTM*2、3D モデリング、電子工作、サーバ管理、またそれらに関する知識や技術を得るための読書会・勉強会 などといった事を行って日々精進したり遊んだりしています。

さて、こうして KMC が行っている様々な活動を挙げましたが、部員全員がこれら全てを手がけているわけではありません。多くの部員はそれぞれ自分の興味がある分野に絞って活動をしています。なので、部員全体が一つになって何かをするといった機会は、週 2 回行われる例会を除けば各種飲み会や合宿、学祭*3の運営といったイベントに限定されており、日頃の活動では各人や各グループが好き勝手な時間に好き勝手な方向に向かって歩んでいます。

その傾向はこの本のコンテンツにも現れており、目次を見ていただければ分かるように技術系のコラムから部員の個人的な旅行記、ゆるい 4 コマ漫画と見事に一貫性がありません。その一貫性の無さ、多様性こそが KMC であると思っております。この本の全てのコラムを心から楽しく読める人間は、あまりいないでしょう。多くの読者の方は内容の半分ほどが未消化のまま本を閉じるでしょう。それでいいのです。自分が良く分からない分野は横目に見つつ、自分の惹きつけられる分野を思いきり味わう。それこそが、KMC という存在の正しい楽しみ方なのです。この本の記事を一つでも楽しんでいただけたなら、或いはこの文章をここまで読んでくれた時点で、あなたは私たちの「仲間」であると言えるのかもしれない。

前置きがすっかり長くなってしまいました。それでは、我々のエッセンスを込めた『独習 KMC vol.4』、どうぞお楽しみください。

この本の内容が、たとえ一文でもあなたのメモリ領域に残る事を祈って。

京大マイコンクラブ 第 34 代代表 hideya

*1Kyoto-university Microcomputer Club の頭文字を取って KMC と略される。以下、文中でも KMC と表記する。

*2コンピュータを用いた作曲活動の事。

*3自作ゲームの展示が主。年によっては、部員オリジナルの体重計や処理系 (!) といったものも展示される。

目次

はじめに	i
色々と未経験な人に送る KMC 新入生体験記 (lastcat)	1
まずは	1
入部後	2
これから	4
部員のコラム	1
プログラミングコンテストの開き方 (cos)	5
はじめに	5
仲間を集める	6
問題を集める	6
入出力、想定解、問題文の作成	7
オンライン・オンサイト会場の用意	8
コンテストを開く	8
おわりに	9
当たり判定 (nojima)	10
はじめに	10
静止した図形の当たり判定	10
等速直線運動する図形の当たり判定	11
まとめ	14
Fortran 戦記 流体編 (astatine)	16
流体の数値計算	16
流体	17
格子設定	18
初期条件の設定	19
圧力の Poisson 方程式を解く	20

Navier-Stokes 方程式	23
新たな初期条件を求めて	24
最後の鍵	25
最後に	27
ミャンマー旅行と 2012 年中国反日示威活動中の中国で同人誌刷ってきた記 (hidesys) . .	29
出かける前に	29
ミャンマー	30
中国	34
あとがき	37
4 コマ漫画: KMC 部員の日常 (madaragi)	II

色々と未経験な人に送る KMC 新入生体験記

lastcat

まずは

文章概要

この文章は、春の入部時点で何の知識も経験もないままこんなバリバリの技術、創作系サークルに入った私が、その結果どうなったかを簡単にまとめたものです。

入部まで

春、私は晴れて大学に合格し、「京大マイコンクラブ（以下 KMC）」という多少あやしげな創作系サークルに入部した。もともとパソコンを触っているのは好きだったし（主にインターネットの海を遊泳するくらいだったが）、サークル紹介冊子や看板、ビラにあった、「コンピューターを使った『何か』をするサークルです」という文句が気に入ったのもあったかもしれない。ゆるい雰囲気サークルに入ろうとしていたので、ガチガチに「プログラミングやります!!」などと書かれていたら私の取った行動は全く別のものになっていただろう。新勤でご飯をおごってもらった後、大学内のサークル棟には向かわず、少し入り組んだ路地の中にある怪しい建物に連れて行かれたときは、大学で受けた「カルト宗教サークルについて」というオリエンテーションが頭をよぎった。しかし、長い話の後に壺を買わされた……ということは全くなく、部室で先輩たちが作ったゲームを楽しく遊ばせてもらい、素直に感心したのでその場のノリで入部を決めた。部室の机で私の横で入部届を書いているまだ顔も名前も知らない同じく新入生であろう者たちは、なにやらそれっぽい話を早速先輩と交わしているし、ズブの素人なのは自分一人のように思えてなかなか辛いスタートになったが仕方がない。

入部後

新入生プロジェクト

さて、KMCには「新入生プロジェクト」と呼ばれるプロジェクト群がある。サークルに入っただけで、右も左もC++もわからない新入生に（私は今でもC++はわからない）、先輩が色々レクチャーし、基礎的な技術も身につけながらサークルに馴染んでもらおうという趣旨で行われるプロジェクトだ。今年のものでいくつか例を挙げれば、「C#でゲームを作ろう 2012」、「DTM勉強会 2012」、「ICPC勉強会 2012」などなど……。新入生は各々自分の興味がある分野の新入生プロジェクトに加わり、先輩に叱咤激励されながら、無理のないペースで徐々に自らのレベルを上げていく事になる。もちろん私もご多分に漏れず、そうするつもりだった、のだが。

やっちまった

手を広げすぎた。

今年行われた新入生プロジェクトの数、7。私が参加したプロジェクトの数、6。つまり行われたほとんどすべての新入生プロジェクトに私は参加していた。参加しなかった残りの一つのプロジェクトは色々ゴタゴタがあって、経営シミュレーションゲームを作るはずが最終的にはちくわをゲットして点数を稼ぐ「ちくわ大明神」なる謎ゲーム製作プロジェクトに変貌していたので、「まともに行われた」という篩にかけると私はすべての新入生プロジェクトに参加していたといえるだろう。週1バイト、6サークル。このタイムスケジュールが私の前期の取得単位数に与えた影響についてはまた別の機会に話すとして、やはりどうみてもむちゃくちゃである。当時の心情を省みるに、やはり周りの新入生と比較しての圧倒的無力感とか以前から特に積み上げてきたものがないし、「取り敢えず全部やってみっかー」とかいうノリに従った結果こういうことになったという感じだった。しかしこの無理の結果、私は見事すべてのプロジェクトを修め、見事KMC部員としてふさわしい、オールラウンドの技術力を手に入れた！

……というのは幻想。

実際は『本当に』残念な話、何一つとして満足できるレベルに達しなかった。コードは書けない、絵も描けない、曲も書けない。客観的に見れば、中途半端に必要なツールだけが詰め込まれたノートPCだけがある、とても切ない感じになっていた。参加するプロジェクトを絞った友人達はそれぞれ対応した技術を手にして羽ばたき始めているのに、自分だけ入ったばかりの頃とほとんど変わっていない。さすがにコレはアカンで。私が他の新入部員と比較してなにか勝っているものがあつたとすれば、部室の入り浸り度、イベントの参加率、そしてその2つから導かれる「先輩との会話量」であろう。最初から人と話すことが特に好きな私に（大学受験直前期はあんまりにも話すので友人が私を見ると逃げていった）、常に話し相手がいる部室という空間を与えた結果、こういうことになったのにはある種必然といえる。一時期は「部室に

住んでいる」とまで言われた入り浸り度を先輩方が内心どう思っていたかはわからないが、とりあえず知名度だけはあった、そんな新入生。

ところが

前期も終わりかけの時、いつものように部室で深夜まで話し込んでいると、ある一人の先輩から話を持ちかけられた。

「そういえばなかなか割のいいアルバイトがあるんですけど、やってみませんか。」

私はすでに一つ、塾講師のアルバイトを抱えていて、それがたびたび無給出勤を強いてくる上に一年契約のせいでやめられず……というどうしようもなく厄介な代物だったので、労働ということに対してすっかり臆病になってしまっていた（読者の方々に忠告させてもらおうと、軽々しく職場を決めるのは絶対にやめたほうが良い）。しかし話を聞いてみると、今回はプログラミングができるのなら本当はかなり割のいい話のようで、少し心が動いた。ところがどっこい残念ながら前述の通り私はろくにコードも書けない。その旨を伝えると先輩は、

「じゃあ夏休み勉強しましょう。」

と軽くのたまった。

結局夏休み、私はひとつのプログラミング言語を詰め込む羽目になった。それがしかもよりにもよって一番ややこしい言語だったため、今から思い返すとあんまり身につけていなくて結構残念感があるのだが、基礎概念みたいなものを0から学ぶいい機会にはなった。また別の機会、部内でゲームを作るという企画がいくつか持ち上がった時、そのひとつのチームのリーダーの先輩から、

「君、ゲームのステージ曲書いてみない？」

という誘いを受けた。

DTMでの曲作りも、結局勉強会が完全に一から自分で一曲作る、という段階になったときに休んでしまい、かなり中途半端になっていた。あんまり触れていなかったためにツールの使い方も曖昧で、なんかうやむやで終わりそうな匂いが漂っていたところにこの申し出である。受けた。こたつの隣にDTM勉強会主催者の先輩を置いて、つきっきりで教えてもらい、ひたすらに作曲した。そして完成した。以下、先輩の一言。

「どっちかってーとこれタイトル曲っぽいしもうこれタイトル曲にしちゃってもう一回ステージ曲書こうか。」

ひたすらにもう一度作曲をした。1曲目の経験から勝手がわかったので2曲目は（ファミコン音源の3音縛りがあったといえものの3時間くらいで仕上げることができた。完全なオリジナルの作品を作るのは初めてだったので、かなり感慨深かった。そしてこの部誌である。今回はこうだった。

「君、表紙描く気ない？」

これから

感じたこと

今現在、この記事を執筆している時点でまだ表紙は完成していない（相当やばい）が、今回のコレもやはり絵描き入門のいい機会となった。さて新しいことを0から習っていくのを繰り返した中で少し気がついたことがある。それはこれら（プログラミング、作曲、絵描き）はすべて、特殊技能などではないということだ。すごく極端な言い方をすると誰かに指導してもらえば一日でも最低限気分くらいは味わえるし、あわよくばひとつくらい完成物が出来上がる。一度できてしまえば二度目からはたやすい。そうやって徐々に凝ったものを、満足できるものを……と追求していくうちに、気付けば自分も昔の自分が別世界の住人と決めつけていたモノを作れる人達のはしくれにはなっている。

to you

この文章を読んでいる人の中には、プログラミングに、作曲に、絵描きに興味があるけどよくわからないし手が出せないという人もいらっしゃると思う。そういう人はまず一度踏み込んでみて欲しい。おそらくネットでそれっぽいページをググるとか、どっかで本でも借りてきて読んでみることだろう。そしてほぼ確実に挫折する。そこまでは予定調和なので安心してほしい。最初から独学というのは一度つまった場合のやる気のrebootが大変困難なのでよほどの熱意がないと続かない。そこで気軽に頼れる先達の存在の必要性に気づくわけである。どんなコミュニティでもよいのでとにかく経験者のいる団体に属して、自分がその技術を身につける必要性のある状況に身を置くと良い。まずそこからすべてが始まる。そうやってスキルを上げていったあなたの作品と出会えたら、この記事を書いたかいもあったというものである。

最後に、KMCの外部ページより、

「京大マイコンクラブ (KMC) は京都大学を中心に活動しているコンピュータサークルです。

プログラムやゲームの開発に興味がある方の入会は常に大歓迎です。入会を希望される方や、詳しい説明をお聞きになりたい方は、一度4~5月の説明会の方にお越し頂くか、メールでお問い合わせ下さい。」

KMCの部員となることが現実的でない人も、「創作すること」を続けていけば、いつか何処かで道は交わることでしょう。

それでは、この記事を読んだあなたの技術力のさらなる発展を願って。

プログラミングコンテストの開き方

COS

はじめに

こんにちは、cos です。ICPC を引退したので問題を解くより作る方が多くなってきました。さて、このコラムではタイトルの通り、プログラミングコンテストの開き方についてお話をします。ここでいうプログラミングコンテストとは、3~12 個のタスクが与えられ、時間内にその仕様を満たすプログラムを速く正確に作るアルゴリズム部門のコンテストを指します。最近はこのプログラミングコンテストが Topcoder や AtCoder などの企業だけでなく、学生団体で開かれることも多くなってきました。具体的には KUPC、UTPC、IJPC などの単体のコンテストや JAG*1の夏合宿、会津合宿、立命合宿などの 3、4 日に渡る合宿まで開かれています。私自身はコンテストの主催に KMC 部内コンテストであるふか杯で 5 回、KUPC で 2 回、JAG で 4 回関わったので、ここで培った情報をまとめたいと思います。

さてコンテストを開くにあたって、全体の流れは以下のようになります。

1. 仲間を集め、いつごろにどのようなコンテストを開くかを決める。
2. 問題を集めて、問題セットを決める。
3. 入出力、想定解、問題文の作成する。
4. コンテストの宣伝、オンサイト会場の確保を行なう。
5. 本番、懇親会、反省会を行なう。

具体的な例として KUPC 2012 でのタイムラインは以下のようになっていました。問題を作ったり、問題自体の評価や校正などに時間がかかるため、1、2 ヶ月前から準備を行ないます。

*1ACM-ICPC Japan Alumni Group. ACM-ICPC の OB/OG の会。

1年くらい前から	人集め
4月	日程決め
4月下旬～5月	問題セットを出し合う
5月下旬	オンサイト会場*2の確保、webページの公開
6/2 (土)	問題セット決める
6/16 (土)	入出力と想定解と問題文のスケッチ done
6/28 (木)	校正
6/29 (金)	プラクティス
7/1 (日)	本番
7/7 (土)	反省会

さて以下の節ではこの個々の詳細について話をします。

仲間を集める

一人でコンテストを開くことは不可能なので一緒にコンテストを開く仲間を集める必要があります*3。なぜ一人だと不可能なのかというと、ミスジャッジを防ぐために一つの問題に対して二人以上が解答を書かなければならないからです。問題セットの原案を作る作業や、問題の難易度の評価も複数人でやった方が安定するでしょう。

仲間を集めるのは難しいかもしれません。特に問題を作ったことが無い人だと嫌厭してしまうかもしれません。そこで、必要に応じて原案はいいから想定解だけお願い、校正だけお願いなど部分的に担当をお願いして、引き受けてもらいやすくしましょう。特に校正などは、日本語の問題なので日頃から競技プログラミングを行っていない人でも引き受けてくれるかもしれません。

さて一緒にコンテストを開く仲間が集まったら、その仲間内だけで使える情報共有のシステムを構築しましょう。具体的には、軽い話をするための Skype や IRC などのチャットシステム、問題を集めたり重要なことを記録するための wiki、想定解や入出力を作成するための SVN や Git などのレポジトリが必要です。それらの準備が整ったら次は問題を集めましょう。

問題を集める

問題を集めて、コンテストのセットを作るためには原案が必要です。原案とは問題の本質の部分（と解法）を書いた物です。問題文の肉付けや解答のソースコードはまだ必要ありません。

原案を作る方法としてよく挙げられるのがタイトルドリブン*4、ストーリードリブン*5、ア

*3Target などの人外を除く。

*4タイトルドリブンの例：「しりとり」、「すべては1になる」など。

*5ストーリードリブンの例：「舞台装置の魔女」、「IRU vs SAKI」、「Acceleration of Network」、「宝探し」など。

ルゴリズムドリブン*⁶、問題変形ドリブン*⁷があります。私はよくゲームの設定を問題にして、それを解いてみて面白かったら採用ということをよくします。他には、今まで見てきた問題の制約条件をいじって新しい問題を作ったりもします。ここは人によって好みが変わると思いますが、重要なのは設定を考えて自分で解いてみるということです。気づいたら NP 困難だったとかでボツ問題になることがよくありますが、何度も作っているうちに良い問題が出てくるので頑張ってください*⁸。

さてそんなこんなで、皆で問題を考えあって wiki に原案を集めたら次は問題選定を行いません。難易度やアルゴリズムのバランスの関係で全部の問題を出題することは通常無いです。KUPC ではおもしろい問題や出したい問題から順に決めていって、バランスが取れるように出す問題・出さない問題を決め、足りない部分は新しい問題を作って対応しました。JAG では出題する問題を問題ストックから推薦しあって、最終的には多数決で決めるという形式になっています。もちろん、途中で問題セットや問題自体の不都合な点に気がいたら問題の差し替えを行なう場合もあります。そこらへんは、皆で話し合って柔軟に対処しましょう。

入出力、想定解、問題文の作成

問題セットが決まったら、次は各問題に対する想定解、誤答、入力、バリデータ、問題文*⁹を作成します。想定誤答は入力が弱くて嘘解法が通ってしまうという状況を防ぐために付け加えます。入力は手打ちで作る分とジェネレータで自動的に作成する場合があります。サンプル、コーナーケース*¹⁰、ランダムケース、ビューティフルケース*¹¹を用意しておけば十分と言われます。バリデータは入力が問題文の制約を満たしているかをチェックするために作ります。問題文をアップデートしたら忘れずにバリデータのアップデートも行なってください。

想定解、誤答、入出力、バリデータを作るのに rime*¹²というシステムがあるのでそれを使うとよいでしょう。それぞれの問題ごとに想定解、誤答、サンプル入力、ジェネレータ、バリデータを置いて、`rime.py test [ディレクトリ名]` と打てば自動で入出力の作成、バリデート、解答の一致・不一致のチェックを行なってくれます。解答の置き方やコンフィグファイルの書き方はサンプル (rime に入っている example) を見るのが早いでしょう。また、入出力の空白や改行の位置の正確なバリデートのためのライブラリとして `testlib`*¹³などもあるので必要に応じて使うと良いでしょう。

問題文は意味が一意に定まり、制約などが過不足なく書かれている必要があります。さらに、

*⁶アルゴリズムドリブンの例：「ハコの魔女」、「Do use segment tree」など。

*⁷問題変形ドリブンの例：「Mod 3 Knights Out」、「XOR 回廊」など。

*⁸自分の問題ストックを見るとボツになってないのだけでも、おもしろくないとか問題セットのバランスなどの理由で公開してない原案が 50 問程度あります。

*⁹スペシャルジャッジ (解が複数あるもの) の場合はチェッカーも必要。

*¹⁰最小ケース、最大ケース、縮退するケース、スターグラフなど。

*¹¹乙とか見た目が美しいケース。意外とこのケースだけで落ちることがある。

*¹²<https://github.com/nya3jp/rime>

*¹³<http://code.google.com/p/testlib/>

日本語の文章として美しくなるまで校正しようとする予想以上に大変なので作業時間は多めに見積もっておくとよいでしょう。問題文を置く場所については wiki やリポジトリ上など皆が見れる場所であればどこに置いても良いです。

オンライン・オンサイト会場の用意

問題セットの準備ができた次はコンテストを開きましょう。実際には問題作成と同時作業になりますが、ジャッジシステムを借りる交渉^{*14}やオンサイト会場の確保などが必要です。

ジャッジシステムを借りる候補としては Aizu Online Judge (AOJ) と AtCoder^{*15}があります。AOJ は会津大学の人によって運営されていますが、中の人を通さないと問題文と入出力のアップロード・テストができない^{*16}という問題があり、おすすめしづらいです。AtCoder は中の人 AOJ よりも多くて連絡を取りやすい上に、自分で問題文と入出力のアップロード・テスト^{*17}が行えるので大分扱いやすいです。しかし、ソースコードが強制的に公開されるのが嫌だという人には使いづらいかもしれません。個人的にはネットワークのアクセスなどの問題が起きておらず、自分たちで作業できる範囲が広い AtCoder の方が好きです。

オンサイト会場については、大学の講義室や企業の会議室などのスペースを借りることが多いですが、交渉する団体によってやり方が異なってくるので詳細は割愛します。なるべく早い段階で交渉を開始すること、確認を怠らないという基本的な事が重要だと思われまます。また会場のスペースが限られているとか、懇親会を行いたいとか、大学に届出を出さないといけないとかよくあるので参加登録を事前に行なっておきましょう。

コンテストを開く

ここまで準備できたらやっとなコンテストが開けます。さて本番当日ですが、一番重要なことはネットワークの確保^{*18}です。ここでトラブル^{*19}が起きると予定を繰り下げる必要があるので、ある程度トラブルが起きても問題ないように時間には余裕を持っておきましょう。

そしてコンテスト本番ですが、実はコンテスト中にやることは少ないです。クラー^{*20}が飛んできたら答えたり、提出された解答を見てにやにやしてください。もしかしたら、ミスジャッジとかデータが弱いとかでリジャッジが必要になるかもしれませんが、どうしようもないのでがんばって急いで修正しましょう。

^{*14}自分で作るという選択肢もありますが、オープンなコンテストだと大勢の人に使われているセキュリティの高い安定したシステムを使った方が無難です。

^{*15}企業が運営していますが、学生がボランティアで活動している分には基本的に無料で貸してくれます。

^{*16}rime から AOJ のコンフィグファイルへの変換スクリプトが用意されれば大分ましになると思うので誰か作ってください。

^{*17}画像や一部のチェッカーは除く。

^{*18}最近ではポータブルなネット環境が安定するようです。 http://imoz.jp/documents/121118_icpc.pdf

^{*19}プロキシの問題とか、DHCP でアドレスが配りきれないとか、Arena がダウンロードできないとかがよく起こる。

^{*20}参加者から出題者への質問。

コンテスト終了後は解説をしたり、懇親会を開いていろいろな人と話をすると良いでしょう。オンサイトのコンテストは半分はオフ会なので思う存分楽しみましょう。

当日はこんな感じですが、もしも来年もコンテストを開きたいと思うなら反省会を開きましょう。何か問題はなかったかとか、どの程度解かれたかとか、何人参加したかのログをとっておくと来年またコンテストを開くときの参考になります。記録を残しておいて来年の準備の前に見ることが重要です。

おわりに

さて今回はコンテストの開き方についての記事を書きました。初めは部内コンテストなど小さいものでも十分ですから、コンテストを開いてみると自分が解くのととは違った楽しみが見えてきます。ぜひ、みなさんもこの記事を参考にしてコンテストを開いてください。お待ちしております。

当たり判定

nojima

はじめに

シューティングゲームやアクションゲームには当たり判定が欠かせません。しかし、当たり判定の実装はそんなに簡単ではありません。例えば、高速に移動する物体が障害物をすりぬけたり、床にめりこんだキャラが変な方向に移動したりするなどのバグが頻発します*1。このコラムでは、2次元平面上の当たり判定におけるこれらの問題に対処する方法を考えていきます。

ゲームにおける各物体は、円や矩形などの領域を持っていて、それらの領域が交差した際に、2つの物体は衝突したと考えます。最もありふれた当たり判定の実装では、各フレームにおいて、物体が持つ領域の交差判定を行い、衝突したか否かを判定します。この方法は実装が容易ですが、上に述べたようなすりぬけ問題やめりこみ問題を解決するのは簡単ではないかもしれません。

もう少し凝った方法として、2つの連続するフレームの間はすべての物体は等速直線運動を行うことを仮定し、衝突する時刻を計算で求めるという方法があります。この方法では、すりぬけ問題やめりこみ問題は簡単に解決できます。

静止した図形の当たり判定

まず、単純な当たり判定の方法を考えていきます。

円と円

中心 (x_1, y_1) 半径 r_1 の円*2と、中心 (x_2, y_2) 半径 r_2 の円が交差するのは、以下の場合です。

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 \leq (r_1 + r_2)^2.$$

*1TAS 動画などでよく見かけると思います。

*2円ではなく円板と呼ぶのが正しいのかもしれませんが、この記事では円と呼ぶことにします。

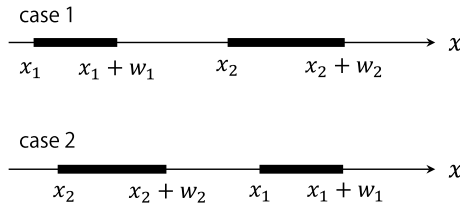


図1 交差しない場合 (1次元)

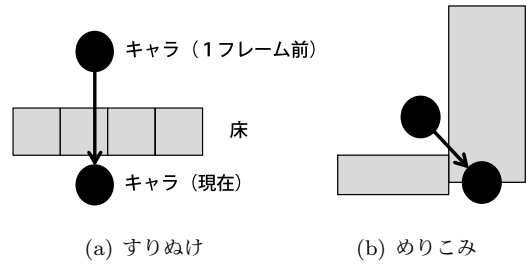


図2 単純な当たり判定の問題点

矩形と矩形

ここでは、 x 軸または y 軸と平行な辺を持つ矩形だけを考えます。

まず、1次元の場合から考えてみます。左端の座標がそれぞれ x_1, x_2 であり、幅がそれぞれ w_1, w_2 である2つの区間があります。この2つの区間が交差しないのは、以下の場合です (図1)。

$$x_1 + w_1 < x_2 \quad \text{or} \quad x_2 + w_2 < x_1.$$

これの否定を取ると、以下の判定式が得られます。

$$x_1 + w_1 \geq x_2 \quad \text{and} \quad x_2 + w_2 \geq x_1.$$

2つの矩形が交差するのは、 x 軸方向、 y 軸方向それぞれで交差する時のみなので、判定式は以下ようになります。

$$x_1 + w_1 \geq x_2 \quad \text{and} \quad x_2 + w_2 \geq x_1 \quad \text{and} \quad y_1 + h_1 \geq y_2 \quad \text{and} \quad y_2 + h_2 \geq y_1.$$

問題点

この方法では、フレームの間を一切考慮しないため、すりぬけ問題が発生します。図2(a)のように、キャラの動きが速すぎて床ブロックの幅を1フレーム未満の時間で通りすぎてしまう場合、床と衝突しているという判定ができず、床をすりぬけてしまいます。

また、めりこみの問題も発生します。あるフレームにおいて、キャラが床や壁にめりこんでいる場合は、めりこまないように正しい方向に押し戻す必要があるのですが、図2(b)のように、どの方向に押し戻すのが正しいのか決めるのが大変な場合があります。

等速直線運動する図形の当たり判定

この節では、等速直線運動を行う2つの図形の当たり判定を考えます。

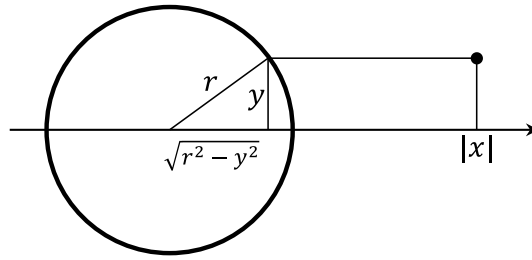


図3 円と点の衝突

時刻 $t = 0$ が一つ前のフレームで、時刻 $t = 1$ が現在のフレームであるとして、前のフレームから現在のフレームまでに2つの物体が衝突するかどうかを判定し、衝突する場合はその時刻を求めることにします。

この節では、「交差判定」と「衝突判定」を使い分けします。交差判定は、動きを考慮せず、静止した状態で領域が交差するかどうかを判定することを指します。衝突判定は、動きを考慮し、前のフレームから現在のフレームまでの間に、2つの移動する物体が交差するかどうかを判定することを指します。

円と円

まず、 $t = 0$ で2つの円が交差している場合は、衝突します。よって、以下では $t = 0$ で2つの円は交差していないものとします。

2つの円の衝突判定は、点と半径 $r_1 + r_2$ の円の衝突判定に帰着できます。また、相対速度を考慮することで、円の速度を0とすることができます。さらに、この問題は平行移動と回転に関して不変なので、円の中心が原点に一致するように平行移動し、相対速度が x 軸と平行になるように回転して考えることにします。

$t = 0$ における点の座標を (x, y) とし、相対速度の x 成分を v とします*3。また、 $r = r_1 + r_2$ とします。

まず、 $v = 0$ の時は、衝突しません*4。また、点が円から離れていく方向に動く場合は衝突しません。したがって、 $xv \geq 0$ ならば衝突しません。さらに、 $|y| > r$ の場合も衝突しません。

そうでない場合は、衝突します。この時の衝突時間は、図3より

$$t_C = \frac{|x| - \sqrt{r^2 - y^2}}{v}$$

です。

円の衝突判定アルゴリズムは、Algorithm 1 および Algorithm 2 のようになります。

*3仮定により、相対速度の y 成分は0です。

*4今、 $t = 0$ で衝突していないことを仮定しています。

Algorithm 1 CirclePoint

Input: 円の半径 r , 初期状態における点の座標 (x, y) , 相対速度 $v (> 0)$

Output: 円と点が $0 \leq t \leq 1$ で衝突する場合、その時刻を返す。そうでない場合、 ∞ を返す。

- 1: **if** $xv \geq 0$ **or** $|y| > r$ **then return** ∞
- 2: $t_C := \left(|x| - \sqrt{r^2 - y^2} \right) / v$
- 3: **return** (**if** $0 \leq t_C \leq 1$ **then** t_C **else** ∞)

Algorithm 2 CircleCircle

Input: 2つの円 (x_1, y_1, r_1) , (x_2, y_2, r_2) , 相対速度 (v_x, v_y)

Output: 2つの円が $0 \leq t \leq 1$ で衝突する場合、その時刻を返す。そうでない場合、 ∞ を返す。

- 1: **if** 初期状態で2つの円が交差している **then return** 0
- 2: $v := \sqrt{v_x^2 + v_y^2}$
- 3: **if** $v = 0$ **then return** ∞
- 4: $x' := (v_x(x_2 - x_1) + v_y(y_2 - y_1)) / v$
- 5: $y' := (v_x(y_2 - y_1) - v_y(x_2 - x_1)) / v$
- 6: **return** CirclePoint($r_1 + r_2, (x', y'), v$)

多角形と多角形

この小節では多角形と多角形の衝突判定を考えます。多角形と書いていますが、ゲームでは四角形の衝突判定ができれば十分なことがほとんどだと思います。アルゴリズム的に一般化可能なので、ここでは多角形で考えますが、実際に実装するときは、四角形などに限定して実装してもよいと思います。

円の場合と同様に、 $t = 0$ で交差している場合は、衝突します。よって、以下では $t = 0$ で2つの多角形は交差していないものとしします。

2つの多角形が衝突するとき、必ず一方のある辺が他方のある辺に衝突します。よって、2つの多角形の衝突判定は、2つの線分の衝突判定に帰着できます。2つの線分が衝突するとき、2つの線分のうち少なくとも1つは、頂点で他方の線分に衝突します。したがって、2つの線分の衝突判定は、線分と点の衝突判定に帰着できます。線分を固定して点を動かしたとき、点の軌跡は線分になります。よって、線分と点の衝突判定は、線分と線分の交差判定に帰着できます。

λ_1, λ_2 を媒介変数とする2つの直線

$$l_1 : \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \lambda_1 \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}, \quad l_2 : \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + \lambda_2 \begin{pmatrix} a_2 \\ b_2 \end{pmatrix}$$

の交点における λ_1, λ_2 の値は、 $\Delta = a_2 b_1 - a_1 b_2$ とすると

$$\lambda_1 = \frac{a_2(y_2 - y_1) - b_2(x_2 - x_1)}{\Delta}, \quad \lambda_2 = \frac{a_1(y_2 - y_1) - b_1(x_2 - x_1)}{\Delta}$$

Algorithm 3 SegmentPoint

Input: 線分 $(x_1, y_1) + \lambda(a_1, b_1)$, 点 (x_2, y_2) , 相対速度 (a_2, b_2) **Output:** 線分と点が $0 \leq t \leq 1$ で衝突する場合、その時刻を返す。そうでない場合、 ∞ を返す。

- 1: $\Delta := a_2 b_1 - a_1 b_2$
 - 2: **if** $\Delta = 0$ **then return** ∞
 - 3: $\lambda_1 := (a_2(y_2 - y_1) - b_2(x_2 - x_1))/\Delta$
 - 4: $\lambda_2 := (a_1(y_2 - y_1) - b_1(x_2 - x_1))/\Delta$
 - 5: **return** (**if** $0 \leq \lambda_1 \leq 1$ **and** $0 \leq \lambda_2 \leq 1$ **then** λ_2 **else** ∞)
-

Algorithm 4 PolygonPolygon

Input: 2つの多角形 P_1, P_2 , 相対速度 (v_x, v_y) **Output:** P_1 と P_2 が $0 \leq t \leq 1$ で衝突する場合、その時刻を返す。そうでない場合、 ∞ を返す。

- 1: **if** 初期状態で P_1 と P_2 が交差している **then return** 0
 - 2: $t_C := \infty$
 - 3: **for all** $s \in P_1$ の辺の集合 **do**
 - 4: **for all** $p \in P_2$ の頂点の集合 **do**
 - 5: $t_C \leftarrow \min\{t_C, \text{SegmentPoint}(s, p, (v_x, v_y))\}$
 - 6: **end for**
 - 7: **end for**
 - 8: P_1 と P_2 を入れ替え、 (v_x, v_y) の向きを反転して、3-7 と同様の処理を行う。
 - 9: **return** t_C
-

となります。ただし、 $\Delta = 0$ である場合は、2つの直線は平行であり、交点は存在しないとしてよいです*5。

この式を使って直ちに2つの線分 l_1, l_2 ($0 \leq \lambda_1, \lambda_2 \leq 1$) の交差判定が行えます。すなわち、この式で λ_1, λ_2 を求めた後、 $0 \leq \lambda_1, \lambda_2 \leq 1$ が満たされていた場合、2つの線分が交差していることが分かり、そうでない場合は、2つの線分は交差していないことが分かります。

多角形の衝突判定アルゴリズムは Algorithm 3 および Algorithm 4 のようになります。

まとめ

このコラムでは、2次元平面における移動物体の当たり判定を考えました。当たり判定は、単純な処理に見えますが、様々な工夫（と様々なバグ）があり、なかなか楽しい分野だと思います。

*5 2直線が重なる場合は交点が存在しますが、線分と点の衝突判定の際には無視して大丈夫です。なぜなら、そのような場合には、同じ時刻に衝突する辺と点の組であって $\Delta \neq 0$ であるものが存在するからです。

ます。このコラムでは扱いませんでしたが、動く壁の押し返しをどうするかなどを考えるのもおもしろいです。

また、最近使いやすい物理エンジンのライブラリ*⁶などがありますので、当たり判定に疲れた方は何も考えずにそっちを使うのもよいと思います。

*⁶Box2D (<http://box2d.org/>) が有名です。

Fortran 戦記 流体編

astatine

KMC の大半は情報や数学に関わる人たちであり、理学部や物理を学ぶ私はややめずらしい部類に入る。周りの物理学科を見ても、プログラミングとは疎遠な者が多く、私も紙の上で微分・積分・Fourier 変換・Taylor 展開の四則演算をしているほうが多い。そのため、私のプログラミングに関する知識のほとんどは KMC の例会や部員の話で得られたものであり、私にとっての KMC とは情報学やプログラミングの世界の縮図に近いものである。

KMC で使われているプログラミング言語は“Ruby”や“C++”、“C#”が多い。また、大学の課題とし“C”や“Lisp”を学んでいる部員も見ることができた。一方、古の言語である“COBOL”や“Fortran”なんて使っている部員は見当たらなかった。そのため、KMC をプログラミング世界の縮図と思い込んでいた私には、“C++”等に比べれば“Fortran”は石槍のようなものであり既に使われなくなった死んだ言語であると映った。だが、“Fortran”は生きていた。しかも、私のすぐく身近なところで。

まず、卒検の課題となる数値シミュレーションに“Fortran”が推奨された。次に、先生からこれを使えと渡された資料にも“Fortran”のコードがある。さらに、研究室の院生たちの「もっと Fortran を使えるようにならないとなあ〜」「Fortran 力を高めないと」という呟き。どうやらこの研究室、ひいては物理学徒の中では、プログラミングといえば“Fortran”らしい。来年からそこに所属する私も、“Fortran”からは逃げられないことだろう。

観念した私は、一人の“Fortran 戦士”となるべく石槍を掲げることにした。

流体の数値計算

今回、数値計算の対象として取り扱おうとするのは、一様な流れの中に置かれた一本の円柱の周りの流れは、どのように変化するのだろうかという問題だ。

この問題は、名のある現象の一つでカルマン渦と呼ばれるものである。川の流れの中に置かれた障害物の後ろにできる渦や、孤立した島の風下にできる渦、これらがカルマン渦と呼ばれるものである。身近に見られる例では、コーヒーに入れたクリームをスプーンでかき混ぜるときに見られる渦模様、これも同じカルマン渦である。この渦巻を計算機でも作り出したい。

流体の数値計算法には、「流れ関数-渦度法」や「有限要素法」など様々な方法がある。今回

は、スタッカード格子 (図 2) に若干馴染みがあったという理由で、「MAC 法 (Marker and Cell Method)」を選ぶことにした。

スタッカード格子は、圧力と速度の評価点を一緒にせず、食い違う形で設定したものだ。具体的には、圧力の評価点を格子の中心にとるが、速度の評価点は格子の枠の上にとることにする。この方法は、圧力差が速度差を生み出すという構造を自然に再現できるという強みがある。微分方程式を差分化するときなど、評価点以外の点での速度が必要になった際には、周りの評価点での値を平均化したものを用いることにする。

理論的な背景などの仔細は各ステップを追いながら説明していくとする。

流体

まず必要なのは、流体の性質に関する知識だろう。

流体とは何かを簡潔に言ってしまうと、固体ではない物体のことである。即ち液体や気体、プラズマのことを流体と呼ぶ。身近な流体としては、空気と水が挙げられる。

流体が固体と区別される点は、流動性と呼ばれる変形のしやすさにある。静止している流体に力を加えれば、流体は力を受けてその形を変えてしまう。一方、固体は力を加えてもその形を保つことができる。

また、物体を流体として考えられる場合は、その物体が膨大な分子の集まりである必要がある。例えば、上空の大気など非常に希薄な場合には、分子を個々の粒子として見たほうが適切になる。だが、宇宙に散らばるガスといった非常に希薄な場合でも、天文単位という長さで見れば、流体として扱うことができる。

対象とする問題によって、さらさらしたものや絡みつくように粘っこいものなど様々な性質を持つ流体が考えられる。その中で最も一般的なものは、非圧縮性と粘性という 2 つの性質を持つ非圧縮性粘性流体だろう。

非圧縮性とは、流体に圧力を加えても体積や密度がさほど変化しない性質のことを指す。水のような液体は、余程高い圧力を加えなければ密度が変化することはないために、非圧縮性流体として見ることができる。一方、空気のように圧力によって体積・密度が大きく変わる流体は、圧縮性流体と呼ばれる。ソニックブームなどの衝撃波は、この圧縮性が引き起こす現象である。

また、粘性は流体の粘り気を示す性質である。粘性を持つ流体では、速度が違う流れの間に力が働き、流れを均一化しようとする。この働きにより、粘性流体は他の物体との境界で相対速度が零 ($v = 0$) になる。水などの液体が身体にまとわりつくのも、この粘性に因る境界条件が働いているためである。

一方、粘性が存在しない流体は非粘性流体、または理想流体と呼ばれる。理想流体を考える場合、粘性を考えないために他の物体との境界条件が粘性流体の場合とは異なる。境界面の法線方向の速度が零でなければならない ($v_n = 0$) 点は、粘性流体と変わらない。だが、境界の接線方向の速度 (v_t) には、何ら制限が課されない (粘性流体の場合では、 $v_t = 0$ となる)。

この2つの性質を加味すると、非圧縮粘性流体の運動を表す基礎方程式は、非圧縮性を表す式(式1)と、運動方程式となる Navier-Stokes 方程式(式2)となる。この2つの式を組み合わせ、(Navier-Stokes 方程式の両辺に div を作用させて) 圧力に関する Poisson 方程式を作ることができる。

$$\text{div } \mathbf{v} = 0 \quad (1)$$

$$\frac{\partial \mathbf{v}}{\partial t} - (\mathbf{v} \cdot \text{grad}) \mathbf{v} = -\text{grad } p + \frac{1}{Re} \nabla^2 \mathbf{v} \quad (2)$$

$$\nabla^2 p = -\text{div}(\mathbf{v} \cdot \text{grad}) \mathbf{v} \quad (3)$$

Navier-Stokes 方程式(式2)に現れる“ Re ”は、Reynolds 数と呼ばれる物理的次元をもたない値である。この Reynolds 数は、流体の速度が速く長さのスケールが大きいほど大きな値となり、粘性が強いほど小さな値をとる。この数を導入する利点は、非圧縮性流体の物理的性質がこの数の値で全て表せるところにある。重力などの他の力を考えない限り、基礎方程式には他の定数は入らない。そのために、境界が幾何学的に相似であり Reynolds 数が同じであれば、流れは数学的には同等なものとなる。これを相似則と呼ぶ。相似則を用いることで様々な性質の流体や様々なスケールの問題を Reynolds 数を変化させるだけで一括して扱うことができる。

Navier-Stokes 方程式の中の $(\mathbf{v} \cdot \text{grad}) \mathbf{v}$ の部分が非線形項であり、この項が存在するために Navier-Stokes 方程式の解を解析的に求めることは難しい。流体に数値計算が用いられるのはそのためである。

格子設定

まず行うべきは格子の設定となる。数値計算のためには、連続な空間を格子に分割する必要がある。その際、対象とする問題によって適切な格子のとり方は変わってくる。今回の問題では、どんな格子のとり方がよいのだろうか。

まず、デカルト座標 (x, y) で格子を取る場合だ。この場合だと、方程式が簡単になるという利点がある。だが、円柱の境界はブロックで作った円のようになるために幾ら格子をとっても滑らかな境界は得られない。そのため、境界条件を考えるのが難しくなるという欠点がある。

一方、円柱の中心を原点として、極座標 (r, θ) で格子を取る場合は、円柱の表面は単に半径が一定 ($r = \text{const}$) という条件で表せる。そのため、デカルト座標に比べて境界は滑らかではつきりとしている。境界条件を考えるのも容易い。だが、難点としてはデカルト座標に比べて方程式がやや複雑になる。まあ、簡単な計算でなんとかなるけど。

他にも、デカルト座標に比べれば、極座標の方は汎用性に乏しいという欠点もある。極座標は境界が円であるときにしか効果を発揮できない。境界が楕円の場合を考えようと思ったら、また別な座標系、具体的には楕円座標系(双曲線と楕円を座標とする)で考えねばならないだろう。その時はまた簡単な計算をして、別のスキームを組み直す必要が出てきてしまう。

次に、格子の間隔のとり方の問題だ。一般的には、格子の間隔を細かく取ると、結果の精度

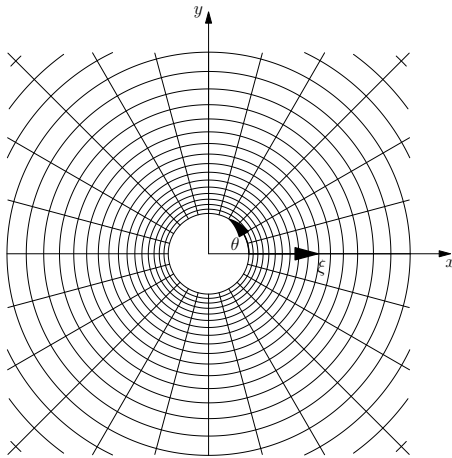
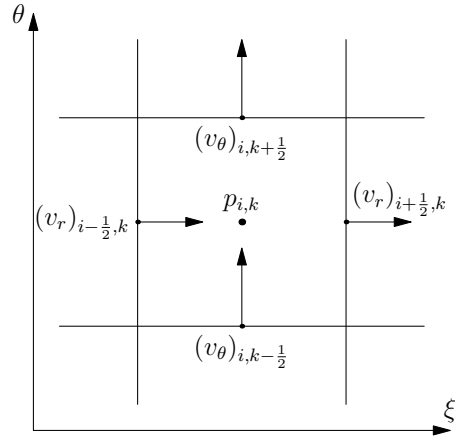
図1 (x, y) から (ξ, θ) への座標変換

図2 スタッカード格子

が良くなる。だが、細かく取り過ぎると、計算量が増えるし、時間差分を細かく取らなければならなくなる。

また、目的によっても格子のとり方は変わってくる。重要な変化が起こっていると考えられる円柱の近くでは、なるべく格子を細かくとって詳しく変化を見たい。だが、円柱より上流の方は割とどうでもいいので、この部分に格子を割り当て過ぎると、何も面白いことが起こっていないのに計算量だけは喰う無駄なことになってしまう。だからといって、格子間隔を一定にとらなければ、スキームが複雑化してバグの温床となることだろう。

これらの事情から、極座標の半径に指数関数を使い円柱近くに細かく格子を割り当てるため、式4に基づいて格子を分割する。具体的な方針を挙げると、半径に関しては $\Delta\xi$ が一定になるように $[0, \ln R]$ を均等に N 分割し、角度 θ に対しては $[0, 2\pi]$ の区間を均等に M 分割する。図で示せば図1のようになる。

円柱の半径を1としているのは、円柱の半径という情報は Reynolds 数に含まれているからである。円柱の半径を変えることは、Reynolds 数を変えることと同じなのである。同様に、時間間隔のとり方も Reynolds 数に含まれる情報となる。

$$(x, y) = (e^\xi \cos \theta, e^\xi \sin \theta) \quad \xi \in [0, \ln R], \theta \in [0, 2\pi] \quad (4)$$

初期条件の設定

格子の設定の次は、初期条件の設定に移る。

初期条件の設定は、数値計算が上手くいくかどうかを決める重要な要素である。問題設定に合った適切な条件も、これは物理的に適切かどうか疑わしいような条件も初期条件として数値的に当てはめることができってしまうからだ。

大体の傾向としては、初期状態が均一な状態を考えるか、あらかじめ解析的・数値的に分

かっている解を用いるか、の2つになるだろう。

全体が均一な初期状態で問題を考える場合は、初期条件自体を考えるのには苦労しない。その後の振る舞いは予測できない。予測できるのは、結果が上手くいかなかった場合に、どこが合っていてどこが間違っているのかが分からず疑心暗鬼になって、全てを投げ出したくなる衝動に駆られることだけだ。

一方、解析的・数値的に解が得られている場合だが、解析的に解が得られないから数値的に求めようとしているわけであり、数値的な解が欲しいから数値的に求めようとしているわけだから、完全な値を求めようとする「鶏が先か玉子が先か」という話になる。そもそも、完璧に条件を満たす解析的な解があれば、数値計算なんてする必要がないというのが事実だ。このことを踏まえると、何らかの妥協をした結果得られたものが、初期条件として考える解析的・数値的な解ということになる。

テスト用と割りきって考えるならば、解析的に扱うことのできる解を用いた方が、解析的に得られている結果と数値計算結果を照らし合わせて、スキームが合っているのかどうかを判別することができるという利点がある。このことを重視して、今回は多少の違いには目をつぶり理想流体のポテンシャル流を初期条件に用いることにした（本当の理由は、参考とした本に書いてあったからなんだけど……合ってるのかね）。

$$\mathbf{v} = \text{grad } \phi \quad (5)$$

$$\phi = \left(r + \frac{1}{r} \right) \cos \theta \quad (6)$$

$$v_r = \left(1 - \frac{1}{r^2} \right) \cos \theta \quad (7)$$

$$v_\theta = - \left(1 + \frac{1}{r^2} \right) \sin \theta \quad (8)$$

$$p = \frac{1}{2} - \frac{1}{2} (v_r^2 + v_\theta^2) + p_\infty \quad (9)$$

これは、遠方で一様流になる ($r \rightarrow \infty$ で $|\mathbf{v} - \mathbf{x}| \rightarrow 0$) 理想流体のポテンシャル流 ($\text{rot } \mathbf{v} = 0$) を表す式である。理想流体の流れを表す式であるため、円柱の境界で接線方向の速度が零にならない ($v_\theta \neq 0$) という難点を持つが、これ以上に適当な解析的な初期条件を見付けられなかったというのが実情だ。

この式を基に、各格子の各評価点に値を割り当てていった。グラフ化して確認してみても、特に問題は見られなかった。

圧力の Poisson 方程式を解く

格子設定、初期条件の設定を終えこれからは時間ステップに関わるスキームを組む段階へ移る。

まずは、与えられた速度場を数値的に満たす圧力場を求めることにする。

座標変換の結果、圧力の Poisson 方程式は式 10 のようになる。

$$\frac{\partial^2 p}{\partial \xi^2} + \frac{\partial^2 p}{\partial \theta^2} = \left(\frac{\partial v_r}{\partial \xi} \right)^2 + 2 \frac{\partial v_\theta}{\partial \xi} \frac{\partial v_r}{\partial \theta} \left(\frac{\partial v_\theta}{\partial \theta} \right)^2 + v_r^2 + 2v_r \frac{\partial v_\theta}{\partial \theta} - 2v_\theta \frac{\partial v_\theta}{\partial \xi} \quad (10)$$

境界条件は、遠方で圧力は一定になると考えてディリクレ境界条件 (式 12) を設定することにした。もう一方の境界である円柱との境界では、境界上では恒常的に速度が零 ($\mathbf{v} = 0$) となることを用いて、Navier-Stokes 方程式からノイマン境界条件 (式 11) が得られる。

$$\left. \frac{\partial p}{\partial \xi} \right|_{\substack{\xi=0 \\ \theta=\theta_k}} = \frac{1}{Re} \left. \frac{\partial^2 v_r}{\partial \xi^2} \right|_{\substack{\xi=0 \\ \theta=\theta_k}} \quad (11)$$

$$p(\xi \geq \ln R, \theta = \theta_k) = p_\infty \quad (12)$$

圧力場を求める上で問題となるのは、この方程式が定常現象を解くための方程式であることだ。

時間的に変化する非定常現象では、一つの数値セットを元にして方程式を使い、次の時刻の数値セットを求めることになる。その際、新たに求めた数値は、以前の時刻の数値セットの情報だけしか持たない。そのように考えれば、隣合う 2 点の値は兄弟のように似てはいるが独立した 2 つの値なのである。

だが、定常現象では大きく異なる。一つの数値セットは、とある方程式を解くものでなくてはならないこのことから、定常現象を数値的に解くとき、隣合う 2 点の値はもはや同じ生物の一部であるように、切っても切れない関係にある。この事情から、定常現象を表す方程式を解く際には、全ての値を“同時に”求める必要が出てくるのだ。

この問題を解く方法の一つに反復法というものがある。一つの近似値を基にして、方程式を使い修正を幾度も繰り返して近似値が求める解へ収束していくまで待つという方法だ。今回は、この反復法を用いて圧力場の Poisson 方程式を解くことにする。

式を差分化すると、方程式はとある $N \times M$ 元連立 1 次方程式になる。 $N = 5$, $M = 4$ の場合を具体的に書き下すと式 13, 14, 15 のようになる。

$$\begin{pmatrix} -\frac{1}{\Delta \xi} I & \frac{1}{\Delta \xi} I & & & \\ \frac{1}{\Delta \xi^2} I & X & \frac{1}{\Delta \xi^2} I & & \\ & \frac{1}{\Delta \xi^2} I & X & \frac{1}{\Delta \xi^2} I & \\ & & \frac{1}{\Delta \xi^2} I & X & \frac{1}{\Delta \xi^2} I \\ & & & \frac{1}{\Delta \xi^2} I & X \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{pmatrix} = \begin{pmatrix} \left. \frac{\partial p_1}{\partial \xi} \right|_{\xi=0} \\ \nabla^2 p_2 \\ \nabla^2 p_3 \\ \nabla^2 p_4 \\ \nabla^2 p_5 - \frac{1}{\Delta \xi^2} p_\infty \end{pmatrix} \quad (13)$$

$$X = \begin{pmatrix} -2 \left(\frac{1}{\Delta \xi^2} + \frac{1}{\Delta \theta^2} \right) & \frac{1}{\Delta \theta^2} & O & \frac{1}{\Delta \theta^2} \\ \frac{1}{\Delta \theta^2} & -2 \left(\frac{1}{\Delta \xi^2} + \frac{1}{\Delta \theta^2} \right) & \frac{1}{\Delta \theta^2} & O \\ O & \frac{1}{\Delta \theta^2} & -2 \left(\frac{1}{\Delta \xi^2} + \frac{1}{\Delta \theta^2} \right) & \frac{1}{\Delta \theta^2} \\ \frac{1}{\Delta \theta^2} & O & \frac{1}{\Delta \theta^2} & -2 \left(\frac{1}{\Delta \xi^2} + \frac{1}{\Delta \theta^2} \right) \end{pmatrix} \quad (14)$$

$$\mathbf{p}_i = \begin{pmatrix} \mathbf{p}_{i,1} \\ \mathbf{p}_{i,2} \\ \mathbf{p}_{i,3} \\ \mathbf{p}_{i,4} \end{pmatrix} \quad (15) \quad I \text{ は単位行列を表す}$$

これを略記すると式 16 のようになる。このままでは反復計算ができないので、次は係数行列 A を変形させる。具体的には、対角行列 D 、下三角行列 L 、上三角行列 U の 3 つに分割して考える。このように分割することで、式 16 は式 18 のように表すことができる。

$$A\mathbf{p} = \mathbf{b} \quad (16)$$

$$A = D + L + U \quad (17)$$

$$D = \begin{pmatrix} a_{1,1} & & & O \\ & \ddots & & \\ O & & a_{N \times M, N \times M} & \\ & & & \end{pmatrix} \quad L = \begin{pmatrix} 0 & & & O \\ a_{2,1} & 0 & & \\ \vdots & \ddots & \ddots & \\ a_{N \times M, 1} & \cdots & a_{N \times M, N \times M - 1} & 0 \end{pmatrix}$$

$$\mathbf{p} = -D^{-1}(L + U)\mathbf{p} + D^{-1}\mathbf{b} \quad (18)$$

この方程式を漸化式化すれば、ヤコビ法 (式 19) が得られる。また、方程式を上から順に解いていく事を考えれば、ヤコビ法よりも速く収束するというガウス＝ザイデル法 (式 20) がつくり出せる。

$$\mathbf{p}^{(\nu+1)} = -D^{-1}(L + U)\mathbf{p}^{(\nu)} + D^{-1}\mathbf{b} \quad (19)$$

$$\mathbf{p}^{(\nu+1)} = -D^{-1}(L\mathbf{p}^{(\nu+1)} + U\mathbf{p}^{(\nu)}) + D^{-1}\mathbf{b} \quad (20)$$

適当な初期値を $\mathbf{p}^{(0)}$ として、この漸化式を用いて $\mathbf{p}^{(1)}, \mathbf{p}^{(2)} \dots$ と求めていく。最終的にこの列が収束したもの、即ち $\mathbf{p}^{(\nu)}$ と $\mathbf{p}^{(\nu+1)}$ の差を何らかの形で評価したものが一定値以下になれば、その値が求める圧力場となる。

実装の簡単さと収束の速さに惹かれて、ガウス＝ザイデル法を実装してみたものの、実際に計算させてみると簡単に 2 万ステップを超えていき、それでも収束は十分ではなかった。そこで、さらに収束が速いとされる SOR 法 (Successive Over Relaxation、逐次加速緩和法) (式 21) を用いることとした。この方法は適当にパラメータ ω を変えることで、ガウス＝ザイデル法 ($\omega = 1$) よりも速い収束が見込める手法である。

$$\tilde{\mathbf{p}} = -D^{-1}(L\mathbf{p}^{(\nu+1)} + U\mathbf{p}^{(\nu)}) + D^{-1}\mathbf{b}$$

$$\mathbf{p}^{(\nu+1)} = (1 - \omega)\mathbf{p}^{(\nu)} + \omega\tilde{\mathbf{p}} \quad (21)$$

SOR 法を実装し、幾つか ω の値を変えて試してみたところ、 $\omega = 0.4$ ぐらいが一番収束が速いように感じられた。

これにより、今度は1万6千ステップ程度で $|\Delta p| \sim 10^{-4}$ 程度まで収束するようになった。だが、結果を見てみると明らかにおかしい。先ほど決めた初期条件と変わらない結果が現れるべきだというのは、現れた結果はそれとは似ても似つかないものであった。観察してみると、角度に依らない対称型をしていて、なんとなく形が $p(r) = -\ln r - r^{-4}$ に似ているような……。ああ、境界条件が狂っているらしい。今考えている理想流体のポテンシャル流では、 $v_\theta \neq 0$ であるため、今考えているノイマン境界条件 (式 11) とは全く異なる、 $\left. \frac{\partial p}{\partial \xi} \right|_{\xi=0} = 0$ という境界条件のもとで解を求めていたようだ。それならば、そのような結果になるのも頷ける。

ここで初期条件に合わせて境界条件を再設定してやるとちゃんと予測通りの結果が得られるかと思ったら、そうは簡単にいかない。何故だ何故だと考えていくと、誤差だと思っていたズレがどうも気にかかってくる。その差異を突き詰めると、スキームの上で考えている評価点と、理論上での評価点が $\frac{\Delta \xi}{2}$ だけずれていることが判明した。その分だけ、境界条件に修正を加えることにした。

ここまで手を加えてようやく期待通りの結果を出すようになった。これでスキームが正しく組めていることが示されて、一安心といったところだ。(こんな処理をしなければならないという時点で、あの初期条件に対する不信感は募るばかりであるが……)

Navier-Stokes 方程式

圧力場が求まったならば、次は Navier-Stokes 方程式を用いて次の時間ステップの速度場を求めることになる。

Navier-Stokes 方程式も座標変換の結果姿を変えて、式 22,23 のようになる。

$$\begin{aligned} \frac{\partial v_r}{\partial t} + e^{-\xi} \left[v_r \frac{\partial v_r}{\partial \xi} + v_\theta \frac{\partial v_r}{\partial \theta} - v_\theta^2 \right] \\ = -e^{-\xi} \frac{\partial p}{\partial \xi} + \frac{e^{-2\xi}}{Re} \left[\frac{\partial^2 v_r}{\partial \xi^2} + \frac{\partial^2 v_r}{\partial \theta^2} - v_r - 2 \frac{\partial v_\theta}{\partial \theta} \right] \end{aligned} \quad (22)$$

$$\begin{aligned} \frac{\partial v_\theta}{\partial t} + e^{-\xi} \left[v_r \frac{\partial v_\theta}{\partial \xi} + v_\theta \frac{\partial v_\theta}{\partial \theta} + v_r v_\theta \right] \\ = -e^{-\xi} \frac{\partial p}{\partial \theta} + \frac{e^{-2\xi}}{Re} \left[\frac{\partial^2 v_\theta}{\partial \xi^2} + \frac{\partial^2 v_\theta}{\partial \theta^2} - v_\theta + 2 \frac{\partial v_r}{\partial \theta} \right] \end{aligned} \quad (23)$$

こいつを差分化して実装してみたはいいが、圧力のポワソン方程式と違い組んだスキームを確かめる術がなかなか思いつかない。何故なら、考えている初期条件では Navier-Stokes 方程式の右辺が零になるからだ。各項を合わせた結果が零になるのだから、数値として現れるのは、方程式を差分化した際の誤差だけである。だが、この誤差が意味有りげなものに見えてきたのだ。他の初期条件を入れて確かめようとしても、私が解析的に考えられる流れは時間発展しない定常流だけであるから、どれを選んでも Navier-Stokes 方程式の右辺が零になる。

いろいろ考えた挙句、格子の間隔を10分の1にしたとき、数値結果が差分化の誤差評価 ($\sim e^{-\xi} O(\Delta \xi^2, \Delta \theta^2)$) に従って100分の1になるかどうかを見ることぐらいしかスキームが上

手く組めているか否かを判断する術を思いつかなかった。この判定法に基づけば、どうやら上手くスキームを組めたらしい。

速度の境界条件は、簡単に自由境界条件 $\frac{\partial v}{\partial \xi} = 0, \frac{\partial v}{\partial \theta} = 0$ を課してみることにした。

完成？

これで一応全てのスキームを組み終えたことになる。気がかりであった圧力の境界条件が時間経過により、初期条件に合わせたものから一般的な式 11 へ緩やかに移るように少々細工をした後、スキームを走らせて時間発展させてみることにした。

最初は期待通りであったが、だんだんと圧力場を求める際の収束が遅くなり……「NaN! NaN! NaN! NaN! NaN! NaN! NaN! NaN! NaN! NaN! NaN! NaN! ...」 \大崩壊／

まっ、まだ、慌てるような段階ではない。数値計算が最初からうまくいく方がおかしいのだ。考えられる要因から潰していこう。

まずはスキームの確認である。特に境界での処理に問題がないか、念入りに調査した。境界点では、境界の外側の点が存在しないために、普通の点と異なる方法で差分化を行わなければならない。また、境界点でのスキームの精度を他の点と同じく $O(\Delta \xi^2, \Delta \theta^2)$ 程度にしようと思ったら、Taylor 展開をして連立 1 次方程式を解くという過程を経て得られる適切なスキームをそれぞれの場合ごとに考える必要がある。

この検討の中で、勘違いしていた点が浮き彫りになった。特に深刻な影響を及ぼしていそうな間違いがあったのは、圧力の境界条件だった。以前に圧力場を求める際、初期条件と同じ結果を得るために円柱との境界条件に現れる $\left. \frac{\partial p}{\partial \xi} \right|_{\xi=0}, \left. \frac{\partial^2 v_x}{\partial \xi^2} \right|_{\xi=0}$ の評価点を $(\frac{\Delta \xi}{2}, \theta_k)$ に設定していたが、理論を再び考えた結果、やはりその評価点は $(0, \theta_k)$ でなければならないことが分かった。このことを踏まえて、 $\left. \frac{\partial p}{\partial \xi} \right|_{\xi=0}$ の差分化を $\xi = 0, \Delta \xi$ の 2 点から、 $\xi = 0, \Delta \xi, 2\Delta \xi$ の 3 点で行うように変え、 $\Delta \xi$ をより小さくするという対処をすることでこの問題を解決することができた。その場凌ぎの策は、思いついたときには妙案に思えたものの、やはり愚策であったようだ。

このようにしてスキームにいろいろと手を加えてみたのだが、それでも計算結果はすぐに NaN をだしてしまう。ここでようやく理想流体のポテンシャル流という初期条件に限界を感じるようになった。 $v_\theta \neq 0$ という壁を乗り越えられなかったのだ。いくら考えを巡らせても、 $v_\theta = 0$ へと緩やかに変化していく自然な仕組みを考えることができなかつた。そもそもが不自然な流れを仮定していたのだから、当然のことだ。

さようなら、ポテンシャル流君。君は実験材料としては優秀だったよ。

新たな初期条件を求めて

今まで使っていた、ポテンシャル流という境界条件を放棄したからには、次の初期条件を考えなければならない。

まず思いついたのは、全体的に静止している状態から初め、適当なパラメータ τ を用いて、

外縁の速度の境界条件を式 24 のように時間変化させることだ。この条件を物理的に解釈すると、計算領域より外側にある一様流が徐々に計算領域に染み入り、一様流へと次第に変化していくことを期待している。

$$\mathbf{v}_{\text{outer}}(t) = \begin{cases} \frac{t}{\tau} \mathbf{x} & (0 \leq t \leq \tau) \\ \mathbf{x} & (\tau \leq t) \end{cases} \quad (24)$$

だが、このような流れの設定に対し、圧力場は異様に大きな圧力の壁を作り出すという抵抗を行った。考えてみれば、圧力場は与えられた速度場を満たすように数値的につくり上げたものであるから、この不自然な圧力場が得られたのは当然の結果だ。この初期条件がもたらしたのは、流れを全体的に変えなければ、不自然な結果となるという教訓だけだった。

この教訓を基に新たな初期条件を考えると、適当なパラメータ τ を用いて、時間が条件 ($t \in [0, \tau]$) を満たす間、時間ステップ毎に計算領域全ての流れに、 $\Delta \mathbf{v}$ (式 25) だけの速度を加算していくという方法だ。

$$\Delta \mathbf{v} = \frac{\Delta t}{\tau} \mathbf{x} \quad (0 \leq t \leq \tau) \quad (25)$$

これを解釈すると、流れのない静的な流体の中に置かれた円柱が、時間 τ の間、一定の加速度で $-x$ 方向へ運動している様子を表している。この円柱は、速度が $-x$ となった時点で加速を止め、円柱は静止している流体の中を $-x$ という一定速度で動き続ける。この状態を円柱から見れば、速度 x の一様流の中に置かれているのと同じ状況になる。

これは、クリームを浮かべたコーヒーの中にあるスプーンを動かすことと同じような状況であり、カルマン渦列を見るためには最適な初期条件のように見える。しかし、恣意的な操作であるためにスキームに若干の手を加える必要がある。

まず、圧力のポワソン方程式に $e^{2\epsilon} \frac{\text{div} \mathbf{v}}{\Delta t}$ という項を右辺に付け加えるという操作だ。この操作によって、各時間ステップで非圧縮性 ($\text{div} \mathbf{v} = 0$) が保たれるように調整する働きをスキームに持たせることができる。

次に、速度の境界条件 ($\mathbf{v} = 0$) が確実に守られるように各時間ステップで毎回 $(v_\theta)_{0,k+\frac{1}{2}} = 0$ を設定することだ。これは理想流体のポテンシャル流ではやりたくてもできなかった操作である。

だがこれでも何かが足りないらしい。時間発展させていくごとに流れはおかしくなり、NaN の列を画面いっぱいに並べてくる。どうも、円柱の境界で非圧縮性が保たれていないらしい。どうやったらいいんだ……。どうやったらうまくいく……。

最後の鍵

少々悩んだ挙句、この問題を解決する鍵はとても簡単なものであった。

円柱の周りで非圧縮性が保たれていない原因は、時間発展していく間、 $(v_r)_{\frac{1}{2},k}$ が零に保たれないからである。ならば、無理矢理 $(v_r)_{\frac{1}{2},k} = 0$ を設定してしまえばいいのではないだろうか。

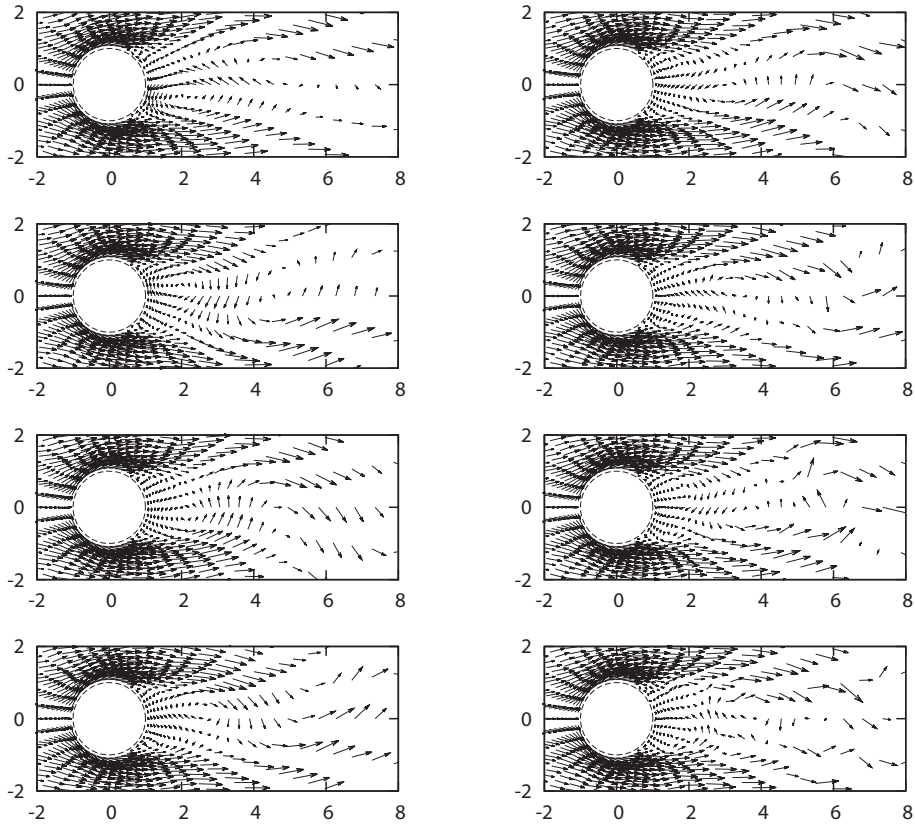


図3 $R = 50, N = 50, M = 40, Re = 500, p_\infty = 1$

はっとなって思い返せば、今までスキームの中で $(v_r)_{0,k} = 0$ という境界の情報を用いたことは何故かなかった。この情報を用いていないならば、数値計算を幾度行おうと求める結果が得られないのは、当然といえば当然だった。

この方策は、非常に上手くいった。今までの苦労が嘘のように、数値計算は安定し待望の渦を1つ2つと生み出していった。

最後に少し、外縁の速度の境界条件が、固定された流入と自由境界条件が滑らかに繋がるように調整を加えた。今まで、円柱によって一様流から歪められた流れが計算領域の外まで続くことなんて考えられもしなかったから嬉しい調整だった。そして、私は3時間計算機を回した。

最後に計算結果(図3)を挙げることにする。左側が上から $t = 20, 25, 30, 35$ での結果、右側は上から $t = 40, 45, 50, 55$ での結果である。より深い調整を行なっていきたいが、原稿のめ切りが近い……無念。

最後に

ただ面白そうだからという理由だけで個人的に挑んだカルマン渦列の数値シミュレーションについて、理論的な背景やドツボにハマった点を「Fortran 戦記」としてここに書き綴ることにした。

この数値シミュレーションは、Fortran のコードを書く練習として選んだものであるため、コードは全て Fortran で書いてある。そして、これが私が書いた初めての大規模で本格的な Fortran プログラムとなるだろう。

Fortran に対して、私は今までただ古い歴史を持つプログラミング言語という印象をもっただけであり、使ったこともなかった。それにもかかわらず、何故使ったこともない Fortran を嫌悪するようになってしまったかという、教科書などにプログラム例として載せられたコードに原因がある。それらのコードにはパンチカード時代の慣例が残っているためか、変数名・関数名は極限まで短くされた上に全て大文字でかかれ、とても人間が読める代物でなかった。さらに、インデントが一切行われておらず、goto が多用されているなどなど、憎しみを抱かせる心憎い演出がなされていた。

だが裏を返せば、これらのルールをきちんと守ってコードを書けば、Fortran の長所を存分に生かせるのである。Fortran は数値計算に特化しているだけはある、暗黙の型宣言やサブルーチンという機能、引数管理の方法がとても使いやすかった。配列を扱い、変数管理に気をつけねばならない数値計算において、これらの機能はとても魅力的なものである。

Fortran の難点もいくつかわかってきた。まず、文字列を扱う際にはどうしても他の言語に見劣りしてしまうことだ。これは、メモリ管理を適切に行うための代償と思えば仕方ないことである。次に、オブジェクト指向言語ではないことだ。数値計算のコードを書き進めるにつれ、変数やサブルーチンに渡す引数が増えるために、似たような名前や引数の順番の方に気を回さねばならなくなった。そのような状態に陥ると、オブジェクト指向が魅力的に映るようになり、速度と速度に関わる処理をクラスとしてひとまとめにしたいという衝動が現れるようになった。

このシミュレーションを通して、嫌悪から愛しさのような感情を抱くようになるまで Fortran と戯れることになった。しかし、実際には Fortran のコードよりも Fortran の計算結果を可視化するための gnuplot のコードを書くための Python のコードを書き続けていた時間のほうが長く、Python を書いていた印象のほうが強い。このことから、数値計算を行う上では、数値計算のコードの書きやすさよりも、計算結果のテストをどれだけ気軽に行える環境を整えられるかの方が重要なようである。

流体の数値シミュレーションに興味があれば、環境を整えて以下に挙げる参考文献等を基にして同様のことをやってみてはいかがだろうか（同じようなシミュレーション例を探してググったら、卒論とか修論とか書いてあるんですけど……）。

参考文献

- エリ・ランダウ、イエ・リフシッツ『理論物理学教程 流体力学 1・2』（東京図書）
- 数値流体力学編集委員会編『数値流体力学シリーズ 1 非圧縮性流体解析』（東京大学出版会）

ミャンマー旅行と 2012 年中国反日示威活動中の中国で同人誌刷ってきた記

hidesys

こんにちは。hidesys です。

僕はわりと旅行するのが好きでいろいろ回ってるんですが、そんな中でも今年四回生の夏にミャンマーと中国へ行ってついでに同人誌刷ってきた話を書きます。

出かける前に

東京：同人誌の原稿

(9/2) 明日から東京へ行ってミャンマーへ飛ぶのに、こみつくトレジャー 20 で売り子したりしててまったく準備していなかった。卒論もまったく手を付けていない。

夜行バスを降りてそのまま彼の家で就寝。明日からミャンマーへ飛ぶまでよろしくね。14 時に起きる。で、原宿行ったり新宿行ったり。東京観光の終わった後は基本的に引きこもって、同人誌書いてた。

最後の日も、ずっと原稿描いてたんだけど、彼に「呑みに行かない？」って誘われた。彼は呑むのが好きだったらしくて、断ったのを今になって後悔してる。



図1 マンダレーのローカルマーケット

中国：上海の臭豆腐と昆明の KFC

(9/8) まず中国へ飛ぶ。チェックイン締め切り 5 分前ぐらいにカウンターへ着いて結構ヤバかった。中国国際航空で成田から上海へ。中国国際航空のオレンジジュースはカラオケで出てくるのより不味いので青島ビール一択だった。

上海では乗り継ぎ 6 時間。ATM でした借金で郊外の住宅街まで出てみるんだけど、地下鉄の改札を出たらなんとも言えない臭いが充満しててオエってなった。出口へ出てみると臭豆腐

を揚げてた。臭いの原因はコレだ。北京の若者に人気なお洒落スポットで臭豆腐をこの前食べたけどほとんど臭いがしなかったので、やっぱり南の方の本場物は臭みがチガウなあと。

そのあと昆明へ。実は前日に昆明市のある雲南省で M5.7 の地震があったんだけど、特にスケジュールに狂いは出なかった。でも、客の中には救助しに来ました系の人がチラホラと居た。うむむ。空港の KFC で眠る。

ミャンマー

ミャンマー概説

東にラオス・タイ、西にバングラデシュ・インドがある、東南アジアの国。2007年に民主化が行われるまでは軍事政権の国だった。少数民族がわりと多くて実際はみんな混じって暮らしてるのだけれども、行政区分としては7つの州に国は分けられ、それぞれの地域に主要な民族がいるという仮定が為されている。政治自体がマジョリティのビルマ族中心で先導されてきたという歴史がある中で、ミャンマー人としての自覚を持っていながらも少数民族であるというアイデンティティを持った彼らが、政治的にはどのような、そして現実的にはどのような生活を送っているのかという点について興味があった。

ヤンゴン：旧首都・ビルマ経済の中心

(9/9) 市街地中心部にあるスレーパゴダ仏塔を観に行く。ホテルの人に「歩いて30分で着くよ」って言われたけど、普通に1時間かかった。暑いし体力消耗するしヤバい。途中で休みに入った安食堂カフェで、外人だということで珍しいのか、お客のおばちゃんに注文の仕方とかを指導してもらった。ありがたい。そのあと、ミャンマーで一番大きなシェダゴンパゴダに行く。ここは素直にタクシー使ったほうが良い。歩いたら地獄を見た。市内では京都市バスとか南海バスの中古バスがそのままの塗装で走ってた。ドア位置が反対に改造されてたけど。あと、朝食を摂りに行ったホテルの食堂で、中高時代の同級生と偶然遭遇したのが一番びっくりした。総合商社に就職が決まったんだって。すごいなー。

パテイン：ミャンマー第4の都市・米の集積港

(9/10) ミャンマーの旅では鉄道を使いたいな、と思っていて、モウラマインまで行くのに昼にヤンゴン駅に行った。でも、朝に列車は出た。一日1,2本しか無いのに。ロンリー・プラネットも地球の歩き方も当てにならないので、鉄道時刻表は現地を確認したほうが無難。しかたがないので別都市のパテインまでバスで。日本円に直すとだいたい350円。エーヤワディー・デルタを一直線にバスで突っ切りますぞ(^_^)

バス道中は道の整備状況が悪くて揺れまくり。でも、あたり一面が青空の下に広がる田んぼで、いいものを見たなあ、と思った。

パテインに着いて、チェックインしてから夕食。今回ホテルはどこも予約せずに行ってるん

だけど、残り 1 部屋とかのをギリギリ確保する事象が続いた。ついてる。旅行前にお墓と神社行っというてよかった。夕飯後にエーヤワディー川付近を歩く。川の高低差がそんなにないから、日本みたいに河川敷があるのではなくて、ただただ低い堤防があってその横に歩道と道路が敷いてある感じ。ヨーロッパみたい。そんなことを思ってたら、女の子の格好をしてる若い子たちに声を掛けられる。前年度行ったインドネシアでは女装は違法で旅中で 1 人しか見かけなかったけども、ミャンマーでは中学生ぐらいの子たちでさえつるんで遊んでたりするんだなあと。一緒に写真を撮ってもらった。この時ほど、ミャンマー語が話せばなあと考えたことはない。

(9/11) 次の朝の朝食はインド料理屋。固めに炊いたご飯と鶏肉の煮付けを交互に層状に重ねて煮汁を上からかけて蒸した料理が、すごく美味しかった。一緒に出てきた酸味のあるスープも、入ってた葉っぱが街路樹のものみたいだったことを除けば爽やかでコレもまた美味しかった。

昼にバスでヤンゴンへ戻る。途中でパンクして 1 時間ぐらい修理したり、速いバスにどんどん抜かされたりして結局着いたのは 18 時ぐらいだった。行きは 360 円だったのに帰りは 600 円だったんだけど、チケット屋まで付いてきてくれた人にコミッション取られたかな。マンガレーまでギリギリ乗り継げるかなという時間にヤンゴンに着いたので、西バスステーションから北バスステーションまでタクシーで飛ばす。バス発車数分前にチケット売り場までタクシーが付けてくれた。運ちゃんありがとう。昼食と夕食を食べそびれて空腹がヤバい。バスの 30 分の休憩でごはん。油と塩だけで炒めたご飯に豆を少し載せてカレースープをつけたのと、卵 3 つ分ぐらいの玉ねぎ・トマト・ダシの入ったオムレツを食べる。すごく美味しかった。150 円。で、眠る。

パティンは綺麗な装飾を施した傘が有名だから、おみやげ用に買っておけばよかった。

マンガレー：ミャンマー第 2 の都市・乾燥気候・最後の王朝

(9/12) バス着 6 時。チェックイン後眠る。14 時に起きる。マンガレー丘に登りに行く。ピックアップ*1 乗り場で若い少女に「日本人ですか？」って話しかけられて、そのままその人もピックアップに乗ってきた。16 歳ぐらいかなと思ったんだけど、25 歳だったので驚き。背も低くて若く見えるのに。そのままガイドをしてもらうことに。

“マンガレー丘に着く。一緒に歩く。観光名所をどんどん通りすぎながら、前にミャンマーにバックパックで旅行しに来た日本人のタカシの彼女になって、タイまで飛んで追っかけて*2一緒に旅行したり日本に 3 ヶ月住まわせてもらったりしたのに、1 ヶ月前に「お互い忘れるのがいいよ」みたいなメールを受け取って、クソ。ブロークンハート。みたいな話を延々とされる。やばい。タカシ悪い男やなホンマ。”

*1 軽トラの後ろにカゴを付けて人が乗れるようにしたもの。東南アジアでは小さな公共交通機関として使われているのをよく見かける。

*2 ミャンマー人の平均月収が日本円で 38500 円である中で、彼女は彼のために自腹でタイまでの航空券を買って飛んで行ったのだ。

とかブログに書いたら、あとで「セックスしたかっただけちゃう？」ってサークルメンバーにツッコまれた。今更「セックスしたかった」だなんて言われても、チン○の舐め方しか知らん。

ご飯を一緒に食べてから別れる。ホテルで休眠を取ってから、20時に地元のお祭りへバイク*3で向かう。仏教じゃない土着の精霊のお祭りだった。道が、歩いた分では1kmぐらい屋台でうめつくされてて、中には小さな遊園地の遊具も持ち込まれたりしててめちゃくちゃ人が居た。若い人たちが完全に興奮状態になって、寺の外で踊り狂っててすごかった。こういうのが生きてる土着の信仰なんだろう。

(9/13) 市場を見に行く。コンクリートで建てられたショッピングモール風の市場と、PC鉄筋で建てられた6階建ての市場と、地べたにゴザ敷いて物を売ってる市場とが並んでる。ショッピングモールみたいなところは中は全然テナントが入ってなくてヤバかった。昔からある市場の方が繁盛してた。男性向けスカートを買う。15時にバスターミナルへ行ったらバガンへのバスが22時にしかなくて、待合室で同人誌の原稿を進めた。

バガン 壮大なステップに広がる古跡群

さて夜行バスに乗るぞー、と思ったら、日本の古い市バスの車内に大量に穀物やらオニオンやら香辛料の入った麻袋を積み出して、その上に天井スレスレになりながら人間が三角座りしだした。狂ってる。で、そのまま雨季で水没してる道路にどんどん突っ込んで行く。ブルルン。地図上ではマングレーとバガンって近いんだけど、普通に山道で悪路だし道も水没してるしですごく時間がかかった。荷物として運ばれるの興奮するけど、実際こうやって麻袋に入れられて運搬されたら死ぬなあとか考えてたら6時にバガン近郊のニューウンーに着いた。

(9/14) 着いてから観光。見渡す限りのステップに多くの古跡が散見される。すごいなあ。

ミャンマーのバガン遺跡では砂で描いた絵を土産物として売っている人達がいる。そんな中で、「僕は砂ではなく石を砕いて絵を描くことを考案したんだ。それに見てよ、本に載ってた白黒のピカソの絵に色を付けて模写したんだ。ピカソの絵を売ってるのは、バガンでは僕だけさ。」って言ってる絵描きが居た。ピカソが、ピカソの時代に生きた人たちとは区別されて、時代性を超えてピカソとして取り出されるのはなぜか。それは、ピカソは他の人達と同様に現実的な生活を営みながらも、幻想世界*4では今まで人類が積み上げてきた芸術というものにもう一つ積み上げをしたからだ。一方、そのバガンの絵描きは、いくらバガンでたった一人だけの特殊な絵描きであるとしても、芸術に対する積み上げは行えていない。でも、彼は自信満々だった。その姿を見て、震えた。何が「ここにある生活がここにある*5」やねんと。僕のは学問に対する積み上げではなかったのかと。幻想領域から目を背けて現実生活を完

*3 バイクタクシーの略。運転手のバイクに二人乗りして移動する。

*4 マルクスの上部・下部構造論における上部構造を言い換えたもの。ここでの記述は、マルクスとフロイトの補完と融合をやったのけた吉本隆明の『共同幻想論』の考えに基づいている。『共同幻想論』については、前号の筆者コラムでも取り上げた。

*5 「幻想領域との関わり合い(=研究が上手く進んだとか)を抜きにした、朝早く起きた・おいしいご飯が食べられたなどの現実的生活が上手くいっている様子」を指す定型句として、数カ月前から筆者が日記やtwitterにて使用してきた。

全にコントロールしたところで、幻想領域に何かしら関与したいという衝動は覆い隠せないだろうと。研究をほったらかして現実的な生活を楽しんだところで一体何なんだ。学問なり芸術なり科技なりなんなり、人類が積み上げてきたものにどうにかして自分の爪痕を残したいという衝動を持つことは、自分がその時代に生まれてその時代の人達と同じように食べて生きて老いて死んでいくんだということに気付くぐらいショッキングなことだと、そう思った。

(9/15) 熱が出た。ヤンゴン行きのバスに 16 時に乗る。

モウラミヤイン

(9/16) ニャウンウー→ヤンゴン・アウンミンガラーバスセンター→モウラミヤン→タンビュッザワ→モウラミヤン と、26 時間ぐらいぶつつづけて移動した。ヤンゴンから西のパテインへ行ったときはデルタ地帯に田んぼが広がってたんだけど、西のモウラミヤンへ来たときはゴムの木畑がたくさん見えた。モウラミヤン⇄タンビュッザワの道際で大きなパイプを溶接しては地面に埋めてたんだけど、通信管かな？ 上水道管かな？ ガス管かな？

(9/17) 朝にモウラミヤインを一望できる、チャイタンラン・パヤーに登る。ミャンマーの仏教施設は神聖だから土足厳禁なんだけど、チャイタンラン・パヤーまで至る参道に犬のうんこ落ちまくりだった。全部避けて通る。

パゴー ヤンゴン近郊の観光地・モン族の都

モウラミヤインからバスでパゴーへ 17 時に着く。そのままバイクを拾ってパゴー特急観光。足裏に世界が書いてある*6寝大仏は絶対見に行きたかった。7 時ほどパヤーを回る。バイクのおっちゃんと雑談してたんだけど、“タイへビザなしで山越え密入国して闇労働しに行ったことがある。” って言ってた。そういう斡旋業者があるんだって。あと、“マレーシアへ出稼ぎに行ったときは、ちゃんとビザを取って行った。” って言われた。

そして中国

(9/18) 4 時起き。列車乗る。ヤンゴンへ。とても鉄道に乗ってるとは思えないような縦揺れがしてやばかった。ひどい縦揺れ。相撲取りのように重心を低くして手すりを持って床へ立ってるんだけど、そのままの格好で 10 cm ぐらい床から突き上げられて浮く。狂ってる。

ヤンゴン中央駅から市内環状線で 6 駅先のパイエット・セイク・コーンへ。空港の近くの駅。幹線道路から駅まで続く市場はすごく流行ってて、やっぱりヤンゴンは旧首都だけあってどこも人の数が多いんだなあと思った。すごい。

ウィスキーとかラム酒とかタバコとかのおみやげをいろいろ買って、ヤンゴン空港から昆明へ中国東方航空で飛ぶ。

*6 ミャンマー人は世界の原理わかっている。

中国

来中国：満州事変の日に入国

昆明空港へ夜中に着く。入国ですよ入国。

旅行へ出向く前には気付いてたんだけど、反日運動の活発な中国に九一八事変（満州事変）の日に入国する予定になってた。おおい。昆明空港の入境係員は英語を話さなくて、「ヤー、リーレンバンナー。」みたいなこと言われて、首をやれやれという感じで振られてから入境スタンプを押してもらった。少し待ってから、同じ飛行機で北京へ。おう。

北京：田舎な首都

恥の多い人生を送ってきました。

北京国際空港に着いたら 23 時で、バスに乗って市内に着いたら 25 時で、ホテルのレセプションで眠ってるおばちゃんを起こしたら中国語でめっちゃ怒られた。ほんまごめんなさい。それを見かねたレセプション横の PC でネットしてた白人に好意で部屋へ入れてもらったんだけど、やっぱり考え直してネカフェへ泊まりに行くことに。部屋を出たら白人が「Go there, and sleep!」って言うてくれてめちゃくちゃいい人だった。お礼を言いながらネカフェへ。……あつ、白人の部屋にメガネ忘れた。

(9/19) 白人の好意を踏みにじった上でメガネ取りに帰るのめちゃくちゃ躊躇する。8 時ぐらいに取りに行ったら、外人居なかった。レセプションのおばちゃんに謝る。ごめんなさい。

さて、ここで問題が。帰国には春秋航空の上海香川便を使おうと思ってたんだけど、春秋航空の規約を読むと、「荷物は手持ち・預けを含めて 18kg まで。1kg オーバーごとに 1100 円の超過料金を頂きます。」って書いてあった。冗談じゃない*7！ 同人誌刷ったら重いぞ。とりあえず新幹線で上海へ。5 時間半。8000 円。高い。

上海についてホテルへ。北京市街地と上海浦東新区はわりと土地勘がついてきた。ご飯食べに行く。あつ、高い割に美味しくない。上海だなあ。

上海：ただの都会

(9/20) さて、原稿描くぞ！ と思ってノーパソを開いたら、画面が真っ暗だった。モウラマインでノーパソを派手に落としたんだけど、それで壊れたらしい。困ったので上海の日本橋、『宝山路』へ出かけて行って修理してもらおう。バックライトの交換だから 100 元ぐらいかな？ と踏んでたんだけど、診断してもらったら LED 液晶だったみたいでパネルごと交換に。最初は 450 元って言われてたんだけど、日本人って言うことがバレて、追加で 200 元払わないと修理しないだの修理中に通りすがりのおっさんに「ニーシー、リーレンバンナー？ ファッキン・

*7 中国東方航空は、エコノミークラスでも 32 kg の預け荷物を 2 つまでと手荷物の持ち込み可能。

ジャップ!!」だの言われて大変だった。結局、追加で 100 元払う。計すると 9000 円ぐらいだったので、日本で自分で交換するよりも安くついた。

翌日には 1 冊目の原稿を上げて印刷所をいろいろ探して回る。昨日とはうって変わって、「日本人が漫画を印刷に来たゾ～」みたいな感じでフレンドリーに対応してくれた。職種によって層が違うのかなあとかなんとか思った。

そのあとは基本的にホテルにこもって 2 冊目の原稿を描いてた。

帰国

(9/25)9 時ちょいに上海から出る春秋航空で帰ることに。同人誌は別で郵送することにした。

最終日の上海で一番遅くまでやってる郵便局が閉まる 3 時間前に原稿を完成させて印刷所に持っていく。頭を下げたら速攻で製本してくれたのでほんとうに感謝する。で、タクシーの運ちゃんに「まだ開いてる郵便局に」って言ったら「そんなんない」ってあっさり返されて、夜の上海をさまよってたら全部の郵便局が閉まっている時間になってしまった。20 kg のエコ同人と旅行かばんを抱えたままこれからどうしようかと悩みにネカフェに行ったら、空港行きの最終電車もなくなってしまった。本当にやばい。

空港に近い駅まで行って同様に困ってる外人を捕まえてシェアタクシーで空港へ。上海浦東空港の眠たそうに目をこすってるインフォメに「郵便局っていつ開くの？」って聞いても不正確な答えしか帰ってこなくて焦った。最終的には、搭乗ゲートの閉まる 15 分前に国境内の郵便局に小包を出してギリギリで飛行機に乗る感じになった。

反日運動について

日本ではいろいろ大きく報道されてたみたいだけど、そういう場面には出くわさなかった。(ネカフェで身分証を求められて、パスポートを見せたら「あなたはここの国では危険だから気をつけるように」って助言された*8 ことはあるけど。) ああやってカッカしてるのは、政府に雇われて反日運動してる人たちか、それになんもわからんと乗せられてしまってるだけの人たちなんだろう。

あと、中国国内報道で、「尖閣諸島を日本が国有化」という報道が前面に押し出されてて、事実は“日本の領土で個人所有だった尖閣諸島を国が買い取った”なのに、“日本の領土でない尖閣諸島を日本が国有のものだと宣言した”といったようなミスリーディングな報道の仕方だったなあと思う。

しかし、19 日以降に政府が「反日運動やめましょう」って勧告したらピタッと止まったのが面白い。なかなかこんな国はないよ。

*8 スパイ映画みたいでかっこいい

中国で同人誌刷るのってどうよ

印刷代それ自体は頑張れば半分ちよいぐらいにできるけども、輸送費がかかって結局あんまり変わらなかったり。サーチコスト・コミュニケーションコスト・輸送時間による機会損失を考えたら、どう考えても日本で刷った方がよかった。まあ、時間が十分にあってかつたまま中国に行く用事のある人がパパッと済ます分にはワンチャンありそう。オワリ。

あとかぎ

■**花月霞 (hanazuki)** 『独習 KMC』は第 1 号より L^AT_EX を用いて組版を行なってきましたが、アドホックなコードが増えてきたため、今号の編集にあたり少しだけリファクタリングを担当しました。本誌で使用しているクラスファイルは [github.com:hanazuki/kmc-clsfles](https://github.com/hanazuki/kmc-clsfles) にて公開しています。そのうちビルドスクリプトなども公開していければと思います。

■**若林秀也 (hideya)** 人生で最も長い期間付き合うことになる人間は当然「自分自身」であり、自分の性格や能力をちゃんと把握して「自分と上手く付き合っていく」事が大事なんだなァ、と最近思っています。しかし全て自己完結して他人の割り込む隙が無いのもそれはそれで寂しいわけで、いやぁ人生って難しい。

■**@t_possum (possum)** 卒論を書くふりをしつつ部室の電子錠を再構築していました。一日中電子錠関係の作業をしていたため、今日は何をしたのかと聞かれた時、専門的でつまらないのではと答えるのが嫌になり、相手にも嫌な思いをさせました。説明が下手だと妹に評されたのを心に、相手に合わせた良い説明を心がけたいです。

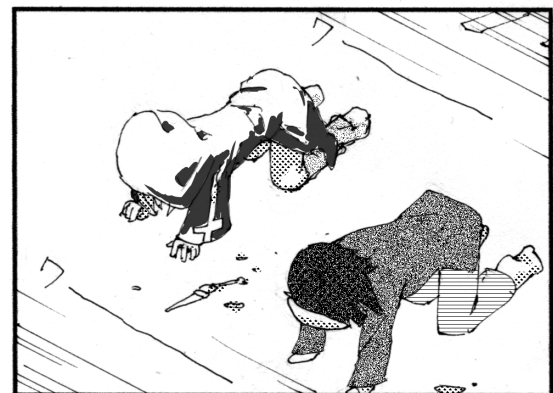
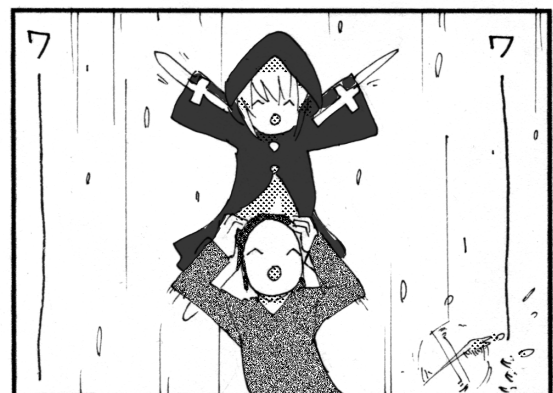
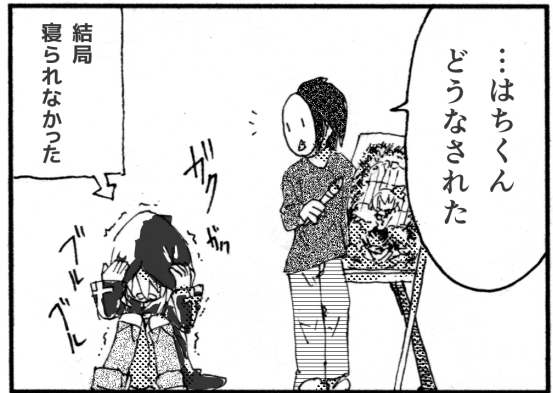
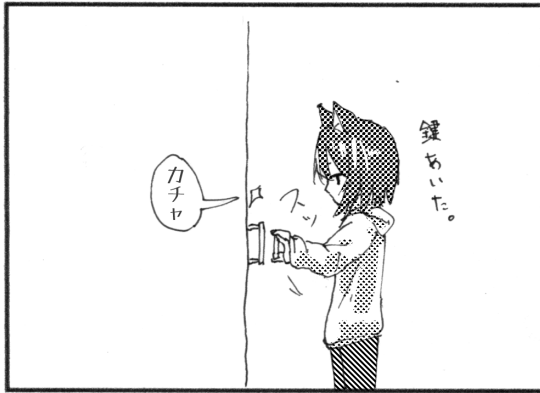
■**まだらぎ (madaragi)** 「そろそろキャラを動かしたい……」落書きでもいいから漫画を描こう！ そう思ったのは 7 月、前期の試験期間真っ最中のことでした。それから 5 ヶ月が経ち、僕は元気に部誌に漫画を描いています。内側から見た KMC の姿をお楽しみください。

■**@t_uda (tuda)** RT @t_uda: 今回記事は書いていませんが、例によって校正作業だけ少し関わらせて頂きました。執筆者の皆さんの T_EX 力・記事力(?) 向上に貢献できていれば幸いです。[すみませんこのあとかぎ使い回しです][#0 は自然数]

■**@85Astatine (astatine)** Fortran の記事が 1/4 を占めるとは思いもしなかった。このままでは KMC が Fortran という古代言語を操るサークルだと思われる。そんな誤解を招かぬように、今度は C++ で数値計算して情報サークル度を高めるように貢献せねばならない。私の戦いはこれからだ!!

0.8秒

AM3:00



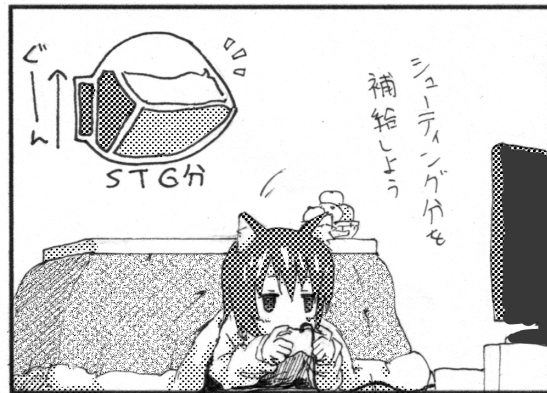
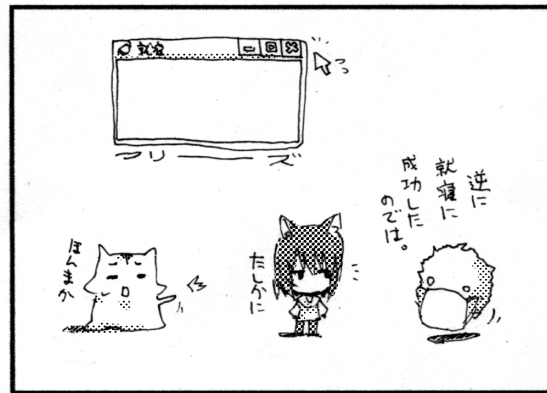
Possum KMC前々会計。内部で人格が分離しており、人と話したりするのは外務局の担当。今年は新しい電子錠管理システム「Becky」の製作に着手する。



chandos KMC2 回生の平部員。IRC でぶらっと謎の発言を残していく正体不明の生命体。マスクはひげを剃るのが面倒だから着けているらしい。

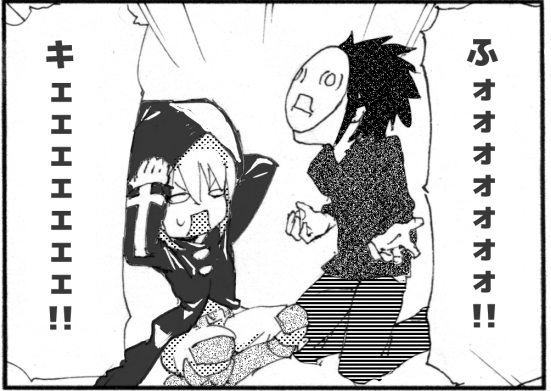


tobiuo 前々文レク。かつては「歩くコンテンツ」と呼ばれ、KMC 最大のコンテンツ力を発揮していた。リア充になるために数々の伝説を残したが、果たして...?





m a d a r a g i 2 回生の平部員絵を描いたり曲を作ったりゲームを作ったりしている。うちの子が可愛くて仕方がない。うおおおおあすううう!!!



ふオオオオオオ!!

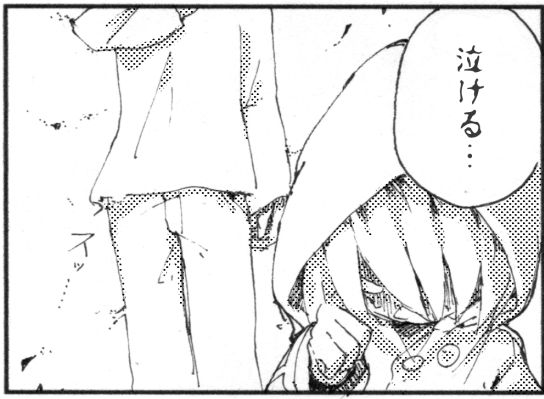
キエエエエエ!!



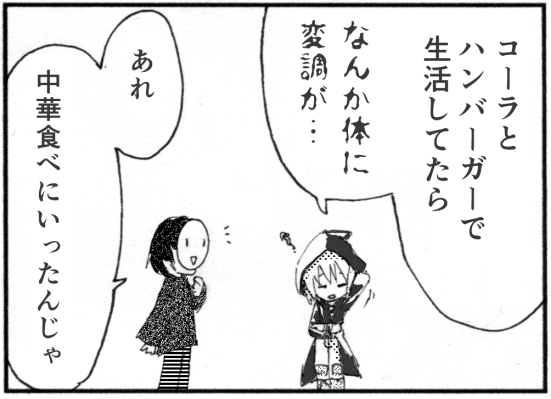
h a c h i 1 回生のお絵かきクラス住民とあるキャラに心底惚れ込んでおり、その愛はキャラの名前を逆にしたものを自身のコードネームにするほどである。



今日もまた
終わろうとしている...



泣ける...



コーラと
ハンバーガーで
生活してたら

なんか体に
変調が...

あれ

中華食べにいったんじゃ



泣いてないで

...失った時間を
追いかけるんだ



ジャンク
性活してたから

中華食べたくなつて
今日行ってきました



世界を崩したいなら
泣いた雫を活かせ

ナイス回文!?



...生活

生の字が性に
なるのは
よくあること

誰かなるとくれ

*なると=IRCでメンバーを追い出せ権利のこと。古くは名前に@がつくことからこう呼ばれる

吸血鬼アイドルをめざして

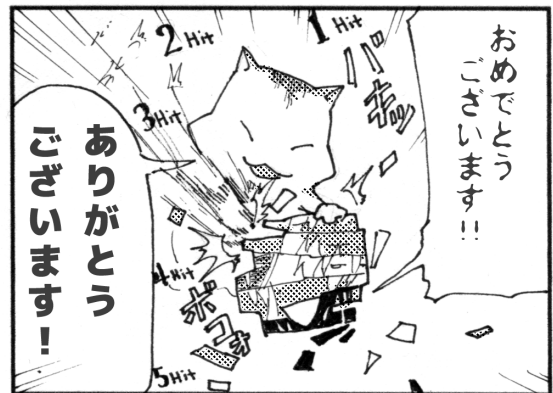
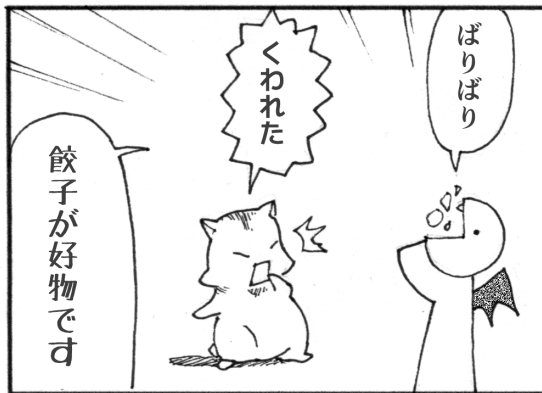
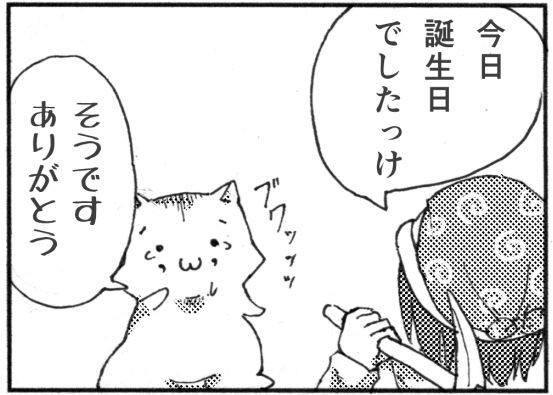
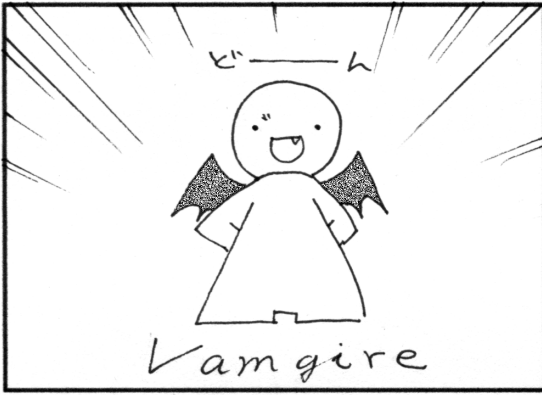
2X歳



g i r e K M C 文レク。京大ラジオ体操同好会初代会長であり、平日の朝6時半から京大クスノキ前でラジオ体操を行なっている。

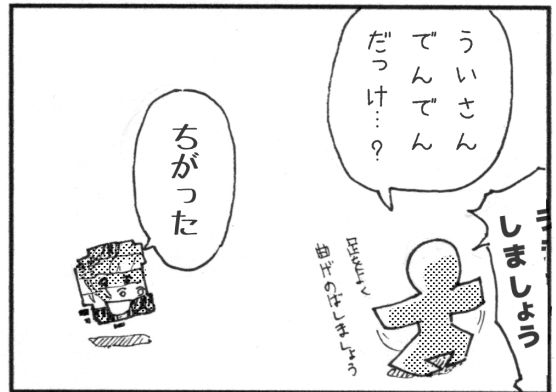
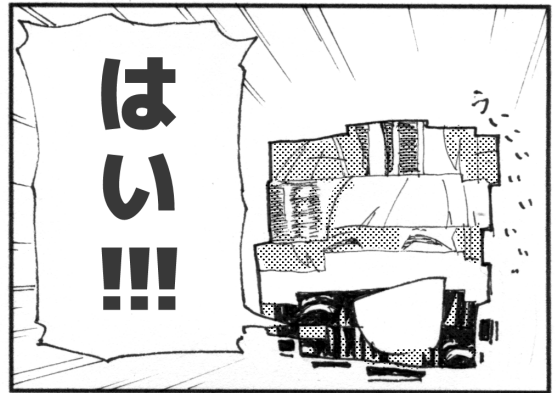


c r y s K M C 前副会長とあるゲームの電気タイプで黄色と黒と白と赤の可愛い生命体を溺愛している。妖精さんとパズルゲームが好き。

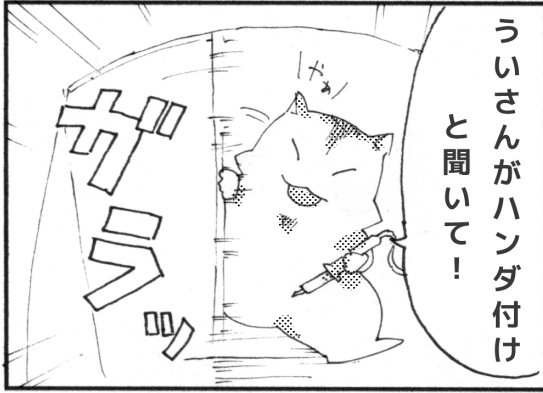
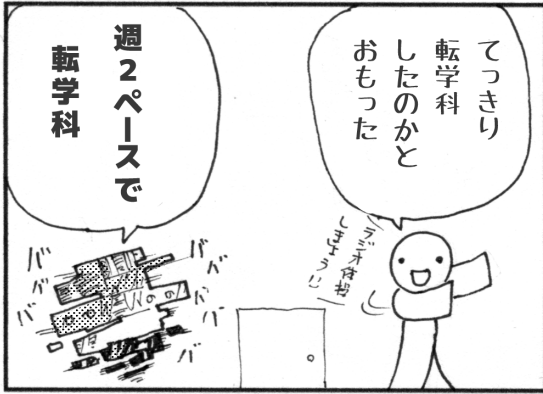




Moko M1のジャンガリアンハムスタ。かつてはKMCの絵の仕事をして一人で引き受けており現お絵かきクラスタの長老的存在である。コアが大好き



*でんでん=電気電子工学科のこと。学期の初めに説明会(ガイダンス)がある。



uureo 副NF担当という名の平部員。Iecana旧カニバリズム山脈など、Webサービス方面の活動をしている。出席ができなすぎて神になった。

国民的デビュー

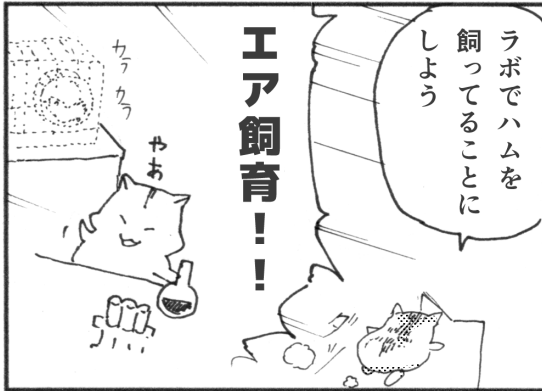
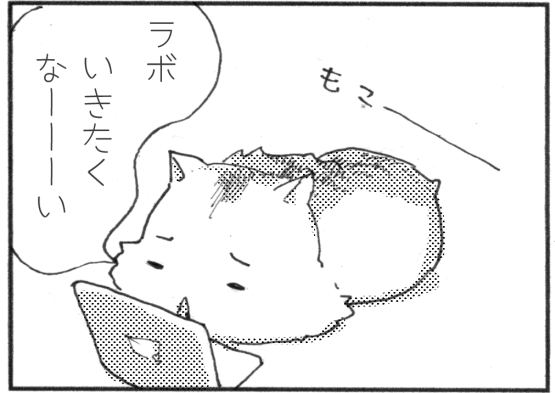
おまじない



tsutchoko KMC前代表。ゲーム「Ocean Wars」のリーダーはこの人。大きな声と特徴的な外見で会長力は53万を超える。厨二病に詳しい。



hideya KMC代表前会長。自身が手がけるゲームはレトロ&クールと評される。ATELAや玉入れの製作者。STGをこよなく愛する。

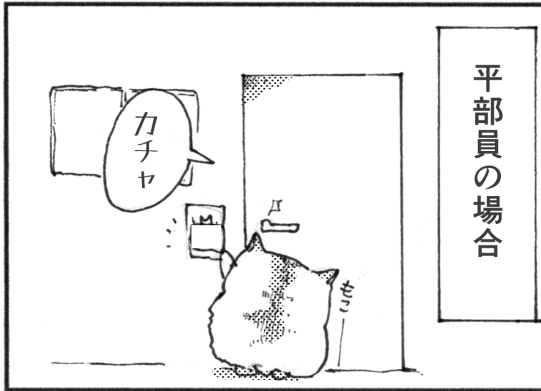


会長力



j f 7 1 2 (じえにぶあ)

KMC会長。よく部屋で寝ている。音屋で自作ゲーム「靈安京」のディレクター&BGM担当。ロックマンゼロを愛す。



KMC 部員 の 日常

1

madaragi



独習 KMC vol.4

2012 年 12 月 31 日 初版発行

著作・発行 京大マイコンクラブ

表紙デザイン lastcat

メールアドレス info@kmc.gr.jp

Web <http://www.kmc.gr.jp/>

落丁・乱丁の際は在庫がある限りお取り替えいたします。上記のメールアドレスまでご連絡ください。

独習KMC

vol.4

■ 利用分類

部誌

■ 製品分類

KMC

■ レベル

初級

中級

上級

<http://kmc.jp/>

