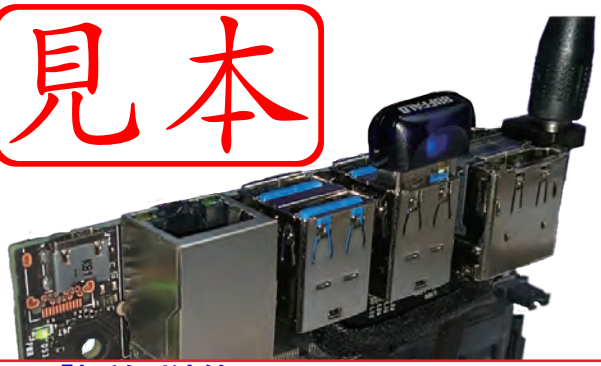
トラ技 特選記事シリーズ【04】

粒子群のふるまいを32コアでAIシミュレーション

超並列演算器 NVIDIA GPU 入門

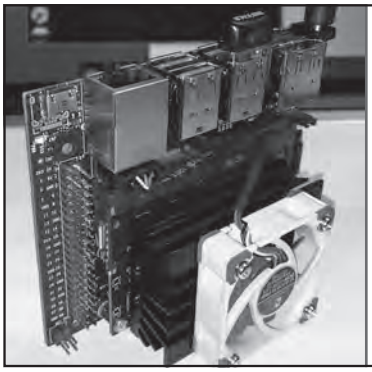
見本

桑野 雅彦



【CONTENTS】

CQ出版社



githubでソースコード&実行ファイル公開！

粒子群のふるまいを32コアでAIシミュレーション

超並列演算器 NVIDIA GPU入門

〈1〉始めの一歩！1個のコアで足し算

桑野 雅彦 Masahiko Kuwano

2019年本誌9月号特集では，NVIDIAのGPUスタータキットJetson NanoをつかってCUDA Cプログラミングで GPUを全力回転させ，円周率などの高速な行列演算に応用してみました．

本連載では，GPUの基本構造とCPUとの違いを理解しながら，GPUならではの並列処理プログラムを作成します．目標は，GPUの特徴を活かした物理シミュレータ・アプリの作成です(**写真1**)．応用すれば電界や天体運動のシミュレータを作ることもできます．アプリケーションは，Jetson Nanoのソフトウェア上で動きます．パソコンやCPUでもアプリケーションを動かせますが，並列演算を高速に実行できるGPUに比べると非力でしょう．

〈編集部〉

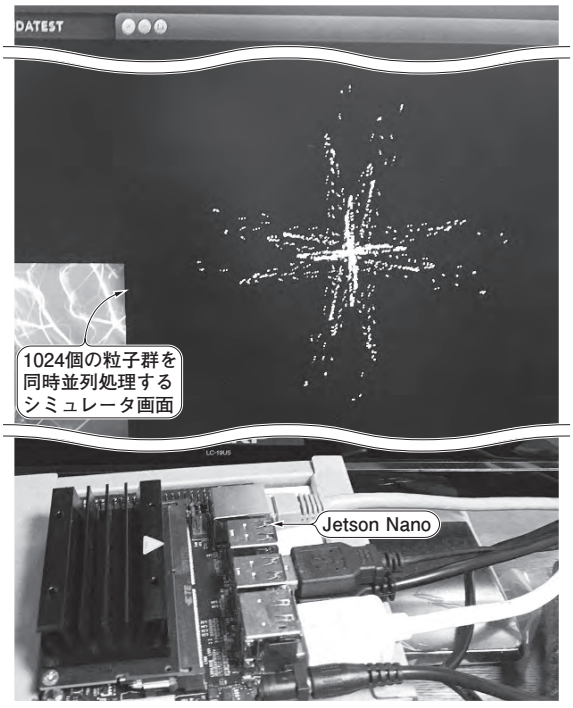


写真1　本連載で作成する物体運動シミュレータを動かした結果

たくさんの物体が互いに力を及ぼし合う挙動を計算する

本稿では，GPUのサンプル・プログラムとして，C言語の“Hello.c”に相当するものを作成します．連載をとおしてサンプルの作成と解説を進める中で，演算結果を動画で見ることができる物理シミュレータを作ります(注1)(**写真1**)．X-Windowの画面上にグラフィックを表示できます．GPUプログラムの記述によって処理速度が変わるようすがわかります．

● シミュレータの条件

▶条件①…物体の最大数は1024個

物体を10×10×10個の格子状に3次元配置した場合でも1000個です．1024個というのはこの手のシミュレーションでは小規模です．

▶条件②…2次元空間

3次元のデータを画面上で表示する場合，z軸成分を取り除くか，遠近法を使って奥行きを表現します．今回のサンプル・プログラムは試作なので2次元にしました．

▶条件③…働く力は距離の2乗に比例

物理シミュレータは，重力や電磁気力のような，実際の物体の特性や挙動を模擬するものが多いです．今回は距離の2乗に比例した力が働く(遠くにあるものほど大きな力が働く)という，実際ではありえない条件を設定しました．

▶条件④…物体どうしは衝突しない

物体シミュレーションには，物体が衝突したときの

注1▶GPUを使わないC言語版も用意

動作速度比較も兼ねて同じ処理をCPUで行うプログラム“gsim_cpu”も用意しました．QRコードか下記リンク先からダウンロードできます．C言語のプログラムなので，Linux上でX-Windowが動けば，コンパイルし直すだけで実行できます．筆者も試しにラズベリー・パイやノート・パソコン上のLinux(Lubuntuを使いました)で動かしました．Jetson Nanoをおもちでない方も興味が出たら一度試してみてください．

https://github.com/PastelMagic/CUDA_TRG202002

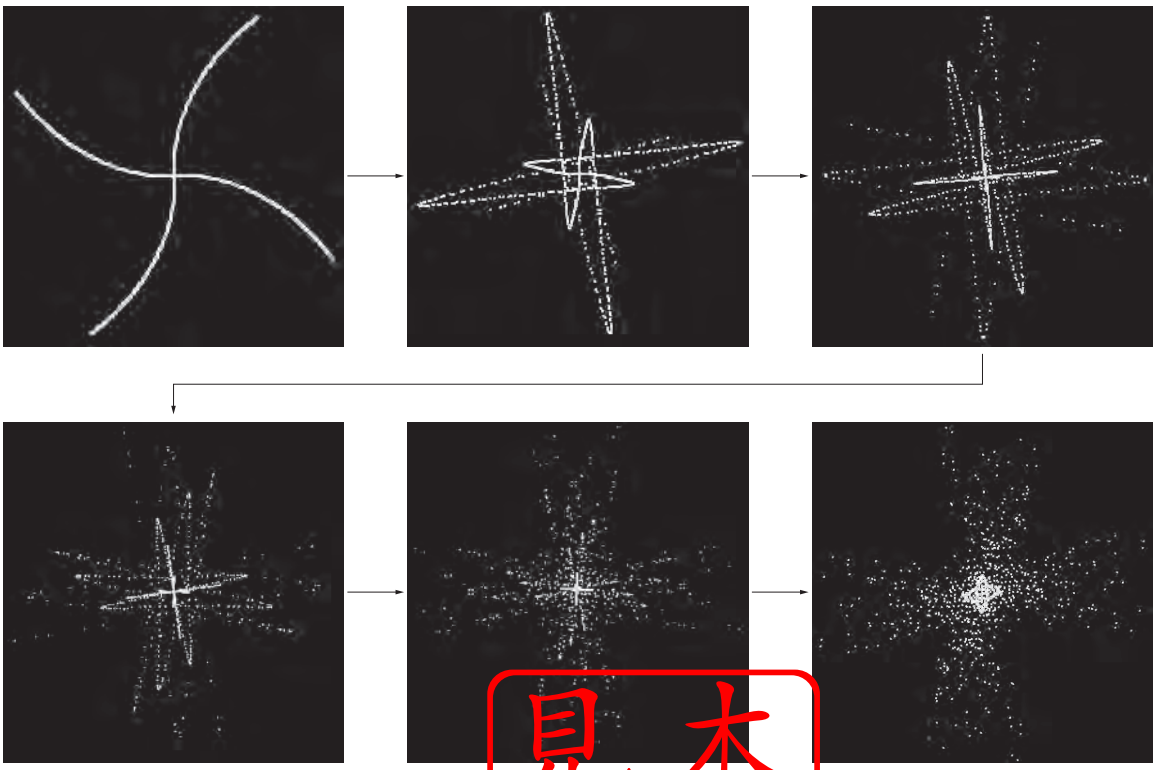


図1　実際にありえない物理条件を加えた物体の軌跡
1023個の物体が，距離に応じて質量を変えながら原点に拘束されている

計算処理が必要です．跳ね返ったり合体したりして1つの物体になる動きを考えます．働く力が「距離の2乗に反比例」の場合，距離がゼロに極めて近い状態では配慮が必要です．今回は衝突には配慮せず，互いに突き抜けるようにしました．働く力も距離の2乗に比例するので，距離がゼロなら働く力もゼロです．

▶条件⑤…ひねった配置にして個々の物体の質量を変えてみる

最初にプログラムを作ったとき，初期状態では単純に十字型の配置でしたが，動きが面白くなかったので，ひねった配置にしました．さらに違う重さの物体がある方が面白いと思い，初期状態では外側にある物体ほど重くしました．

● 物体の動きを計算するときの条件設定

物体の動きの計算や表示は，次の4つのステップを繰り返して実行します．力の働き方はまさに「空想世界」ならではの動きですが，実行手順は現実的な物理シミュレーションと同じです．物体が移動するたびに1024個すべてに次の処理が適用されます．

(1) 1つの物体が周囲の1023個の物体のそれぞれからどのような向きにどのような大きさの力を受けるのかを計算する．それらのベクトル和を求めると，最終的に受ける力の方向と大きさが決まる
(2) 受ける力を質量で割ると加速度，すなわち単位時間あたりどれだけの速度が増減するのかが決まる
(3) 現在の速度に加速度を加えると「単位時間経過後の物体の速度」がわかる
(4) 速度に単位時間を乗じると移動する先の位置が決まる

● 1024個の粒子群でシミュレーション

図1にシミュレーション結果を示します．

実際にはありえないような条件だらけですが，動かしてみると，初期状態の十字を少しひねったような形が縮小/伸長するような動きを繰り返し，そのうち十字型の軌道上をたくさんの点が振動するような動きになりました．あくまでもxy平面の2次元上でのシミュレーションですが，3次元方向に動いているかのように見えます．時間が進むと十字型に多くの点が飛び交うようになり，さらに進むと一面に多くの点がランダムに存在する星雲のような感じになっていきます．

最終的なランダムの状態にたどり着くまで形が次第に変わっていくので，形を見るだけでも，演算の進行具合がなんとなくわかるでしょう．