DOING PHYSICS WITH A COMPUTER IN HIGH SCHOOLS: DESIGNING AND IMPLEMENTING NUMERICAL EXPERIMENTS

Giorgio Pastore and Maria Peressi, *Department of Theoretical Physics, University of Trieste, Strada Costiera 11, I-34151 and CNR-INFM DEMOCRITOS National Simulation Center, Trieste, Italy*

Abstract
We report on a project devoted to teachers and last two-years students of High Schools, aimed at providing hands-on demonstrations of designing and implementing simple computer experiments to show their power in predicting and unveiling different physical phenomena.
The focus is on actual problem solving and on algorithms rather than on programming languages or on sophisticated "canned" computer simulations. The problem of Brownian motion is discussed here as a case study.

1. Introduction
Computer simulations are virtual experiments in which our representation of the physical reality, though necessarily schematic and simplified, can be tuned and varied at will. In the last decades, computer simulations has emerged as the third fundamental paradigm of science, beside theory and experiment. It is time for them to play a similar role in science education, to show their predictive power and how computers help unveil some of the most fascinating secrets of nature.

Besides an impressive numbers of Java applets concerning physical phenomena available from the Web, several simulations software packages and integrated environments are available and already used as an interactive and efficient learning tool. It is our opinion that in most cases, however, students cannot get a feeling of what is a computer simulation carrying on numerical experiments with a software ready-to-use; similarly, they cannot have a feeling of what is a real experiment if they do not have the possibility of doing their hands-on sessions in Lab.

It is our opinion that time is ripe to go beyond traditional computer aided instruction to a real computational physics laboratory with students from the last two years of High School. We propose therefore to design and implement *together with the students* simple computer experiments to study real-world problems, the focus being on actual problem solving. We focus on problems that students can afford using basic mathematical and physical knowledge normally available to them. We select basic numerical tools, focusing on algorithms rather than on specific programming languages or techniques.

In the activity already done with students of different High Schools in Friuli-Venezia Giulia (Italy), such problems have included: the motion of planets (universal gravitation, Kepler's laws), the physics of classic billiards and chaos (from circular billiards to the stadium billiards), the ray optics (light refraction), the fractal growth of surfaces (different deposition models), the Brownian motion.
We focus here on the latter, as a prototype of realistic problems that are difficult to solve analytically at high school level but can be easily solved using numerical simulations, without the necessity of introducing differential equations or integral calculus.

2. The Brownian motion
2.1. The physical system
In 1827 the botanist Brown observed that the specks of suspended pollen in water describe a zigzag, random motion. In 1905 Einstein supplied the explanation of the Brownian motion and, also for this, in 1921 he was awarded the prize Nobel. He realized that the motion of the specks of pollen is due to a stochastic force caused by many collisions with solvent particles much smaller and light (not visible on the chosen observation scale). This stochastic force can be dealt with statistical methods, without worrying about the details of the dynamics of the small molecules of the solvent.

A quite sophisticated mathematical treatment allowed Einstein to provide the key relation between diffusion coefficient D and solvent viscosity η:

$$D = k_B T/(6\pi\eta P), \quad (1)$$

where T is the temperature, $k_B=R/N_A$ is the Boltzmann constant, R the gas constant, N_A the Avogadro's number, and P the radius of spherical pollen particles.
We will follow a different approach due to Paul Langevin, a French physicist, which, one year after Einstein, published another paper obtaining the same relation with a complete different method.
Langevin wrote an equation of motion including a *random* force due to the solvent. Although stochastic differential equations are not an easy subject and certainly much beyond the possibility of even the brightest high school student, the conceptually simple form of Newtonian equations of motion, combined with the possibility of extracting the main consequences by numerical methods make this approach much easier than the original one by Einstein based on partial differential equations for the probability distribution. We adapted the analysis and the simple derivation of the numerical algorithm by B.G. De Groot (De Groot, 1999) to a more computationally based approach.

2.2. The algorithm

Following De Groot's analysis we first write the equations in one dimension of the elastic collision between a large particle of mass *M* and velocity *V* (*V'* after the collision) with a small (solvent) particle of mass *m* and velocity *v (v'* after the collision). The conservation of momentum and energy give:

$$MV+mv = MV'+mv' \quad (2)$$

$$MV^2/2+mv^2/2 = MV'^2/2+mv'^2/2 \quad (3)$$

Extracting *V'* and *v'* as a function of *V* and *v* and of the masses, taking into account that $m<<M$ and keeping the linear terms in *m/M*, we obtain the change of velocity of the heavy particle after the collision:

$$\Delta V=V'-V = (2m/M)v - (2m/M)V \quad (4)$$

After N collisions (that we assume to occur in a time interval Δt) the velocity variation is

$$\Delta V= (2m/M)(v_0+v_1+\ldots+v_{N-1}) - (2m/M)(V_0+V_1+\ldots+V_{N-1}) \quad (5)$$

where v_i and V_i are the velocity of the smaller particles and of the larger particle respectively before the collisions.
We can further simplify Eq. (5) introducing the reasonable assumption that all the velocities of the larger particle in all the collisions occurring between *t* and $t+\Delta t$ are the same, i.e. $V_i=V(t)$. Introducing the frequency collision $n=N/\Delta t$, the right hand side term in Eq. (5) can be written as $-2mn\Delta tV(t)$. We can therefore derive an expression for the force acting on the larger particle:

$$Ma= M\Delta V/\Delta t=2m(v_0+v_1+\ldots+v_{N-1})/\Delta t - 2mnV(t). \quad (6)$$

The equation of motion is therefore:

$$Ma = F_s - \gamma V(t) \quad (7)$$

where F_s is a *stochastic force*, i.e. the cumulative effect, in the time interval, of many collisions with smaller particles, to be dealt with statistical methods. The other term is instead a drag force, opposite to *V(t),* being γ positive. γ can be expressed (using Stokes' formula for a sphere of radius *P*) as:

$$\gamma=6\pi\eta P. \quad (8)$$

The most interesting feature of Eq. (7) is that both forces have the same origin, in the collisions with the smaller particles. Such common origin is on the basis of the connection between *fluctuations* (diffusion coefficient) and *dissipation* (viscosity). We discretize Eq. (7) in time to solve it numerically. Indicating with ΔV_s the velocity variation in the time interval Δt and labeling with *q* the discretized time, we obtain:

$$V_{q+1} = V_q - \gamma(\Delta t/M)V_q + \Delta V_s \ . \quad (9)$$

If we reduce Δt to time interval where only one collision occurs, we would have:

$$\Delta V_s = 2mv/M = 1/M\ v/|v|\ \sqrt{(2\gamma k_B T/n)} \quad (10)$$

where we have used the equipartition theorem of the kinetic theory of gases ($mv^2/2=k_BT/2$ for each cartesian component of the velocity). The ratio $v/|v|$ is -1 or +1. From statistical physics (this could require a preliminar "numerical experiment" with students) we know that the result of N such collisions is a random variable with a Gaussian distribution centered in 0 and with standard deviation $\sigma=\sqrt{(N/2)}$. If we indicate with *w* a random variable with a Gaussian distribution centered in 0 and with unitary standard deviation (many programming languages provide directly *w)*, the final result for the velocity V_q and the position X_q of the heavy particle at the time *q* is:

$$V_{q+1} = V_q - (\gamma/M)V_q\Delta t + w_q(\sqrt{(2\gamma k_B T\Delta t)})/M \quad (11)$$

$$X_{q+1} = X_q + V_{q+1}\Delta t. \quad (12)$$

This algorithm is the hearth of our numerical approach and can be easily implemented for iterative execution. This allows to prove by numerical experiments: (i) the linear behavior of the mean square displacement $<R^2>$ (R^2 is the modulus square of the displacement of the heavy particle in the *d*-dimensional space) with time and (ii) the validity of the Einstein relation between the slope of this line and the solvent parameters (temperature and drag coefficient):

$$<R^2> = (2d\ k_B T / \gamma)\ t \quad (13)$$

This quantity described the time evolution of the size of a drop of solute in a solvent which can be phenomenologically measured through the diffusion coefficient D:

$$<R^2> = 2dD\ t \quad . \quad (14)$$

2.3. A possible implementation

We have implemented the algorithm in Java, and we describe in the following the relevant parts. Concerning the variables used:

- **`vel, pos, posPrec`** are three arrays containing the Cartesian components of the velocity V, position X and position at the previous step of each Brownian particle.
- **`random.nextGaussian()`** provides the values of a random variable with a standard Gaussian distribution
- **`NDIM`** and **`npart`** are the spatial dimension and the number of heavy particles whose motion is calculated
- **`dt`** is the time interval Δt

…

```
double posPrec[]=new double[NDIM];
for (int ip=0;ip<nPart;ip++) {
      for (int jc=0;jc<NDIM;jc++) {
posPrec[jc]=pos[ip][jc];
vel[ip][jc] =
      vel[ip][jc] * ( 1 - gamma*dt/massa ) +
      random.nextGaussian()*Math.sqrt(gamma*kT*dt)/massa;
pos[ip][jc] += vel[ip][jc] * dt;
       }
```

We have also to assign specific values to parameters and constants. The physical parameters (mass **massa,** temperature **kT**, the drag coefficient **gamma**) are given by the specific physical situation we want to study. For a speck of pollen in water at room temperature, we have $k_BT=4\cdot 10^{-21}$J, $M=1.4\cdot 10^{-10}$kg, and a reasonable value of γ is $8\cdot 10^{-7}$Ns/m.

It is very simple in this formulation to introduce also the effect of an external force.

2.4. Running the code

The time step Δt is crucial and cannot be fixed a priori. A suggestion comes from rough numerical considerations. A too small value of Δt would not allow us to mimic the time evolution of the physical system on the time scale of the numerical simulation. On the other hand, a too large value would introduce serious numerical uncertainties. At zero or very low T the stochastic force should be negligible and the resulting motion should be a damped motion, with a drag proportional to the velocity. In order to describe well this situation with our algorithm, it is necessary for $\gamma\Delta t/M$ to be small with respect to unity (see Eq. (9)). Under this constraint, therefore, we suggest to test the code for increasing values of Δt until important changes in the diffusion coefficient are observed.

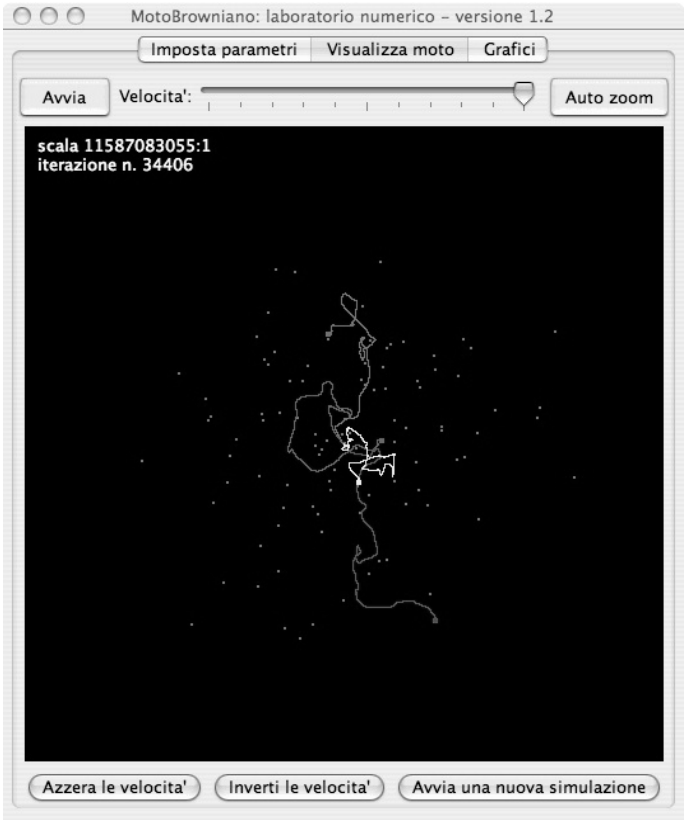

Figure 1: Snapshot of a numerical simulation of the Brownian motion in two dimensions of many large particles. The trajectories of four of them are shown. The parameters of the simulation are given in the text.

3. Suggestion for numerical experiments

We suggest in the following some possible questions that could be asked the students during an assisted hands-on session in computer lab.

1. Looking at the plots obtained, verify that the behavior of the mean square displacement is linear with time after an initial transient.

2. Estimate the slope of the line describing the mean square displacement as a function of time obtained from the numerical simulation for different values of *M, T* and γ and verify the consistency with the Einstein relation.

3. Verify the sensitivity of the results obtained on the time step Δt (try using a value twice or half …)

4. Apply the code to different physical situations, other than the pollen in water, by using some realistic values for different fluids (γ is related to η that can be easily found for different fluids) and suspended particles (P, M).

4. Conclusions

We have proposed the simulation to students of different types of High Schools. In our opinion they have been able to follow all steps of the analysis, including the implementation of the algorithm. Two hands-on sessions (about three hours each one) would be a reasonable choice in order to introduce the physical problem and the specific algorithm, to become a bit familiar with random numbers and finally to understand and make individual numerical experiments with a provided working code. Writing the code from scratch would be equally possible, the numerical core of the code is not longer than a few tens lines, but this would require previous training on computer programming. We would strongly encourage to offer students to develop programming skills in a computer programming language because such experience helps to develop the right attention to required data, logic sequence of instructions and hierarchy of tasks which is highly valuable for an algorithmic approach to physics. However, we stress that in our approach the key hands-on activities are connected to the algorithmic design, to the tuning of numerical parameters and to the critical exploiting of results of open source codes more than on bare programming.

References

Einstein A, (1905) On the Movement of Small Particles Suspended in Stationary Liquids Required by the Molecular-Kinetic Theory of Heat, Annalen der Physik (19), 549

De Grooth BG (1999), A simple model for Brownian motion leading to Langevin equation, American Journal of Physics (67), 1248-1252

A version of the Java software developed by the Authors and implementing the algorithm discussed here can be found in: http://www.democritos.it/edu/index.php