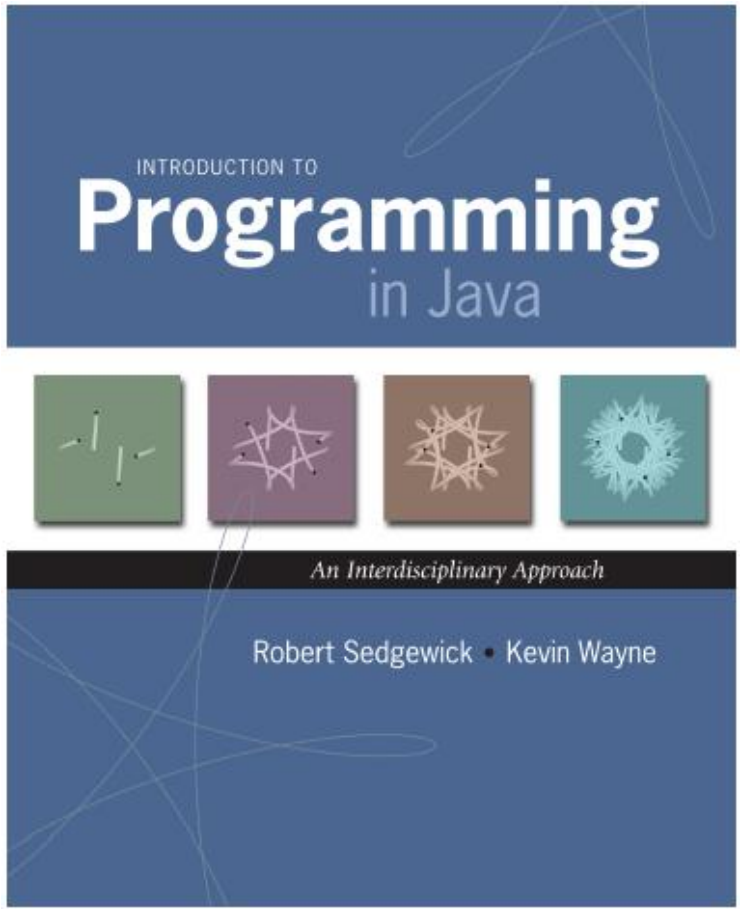
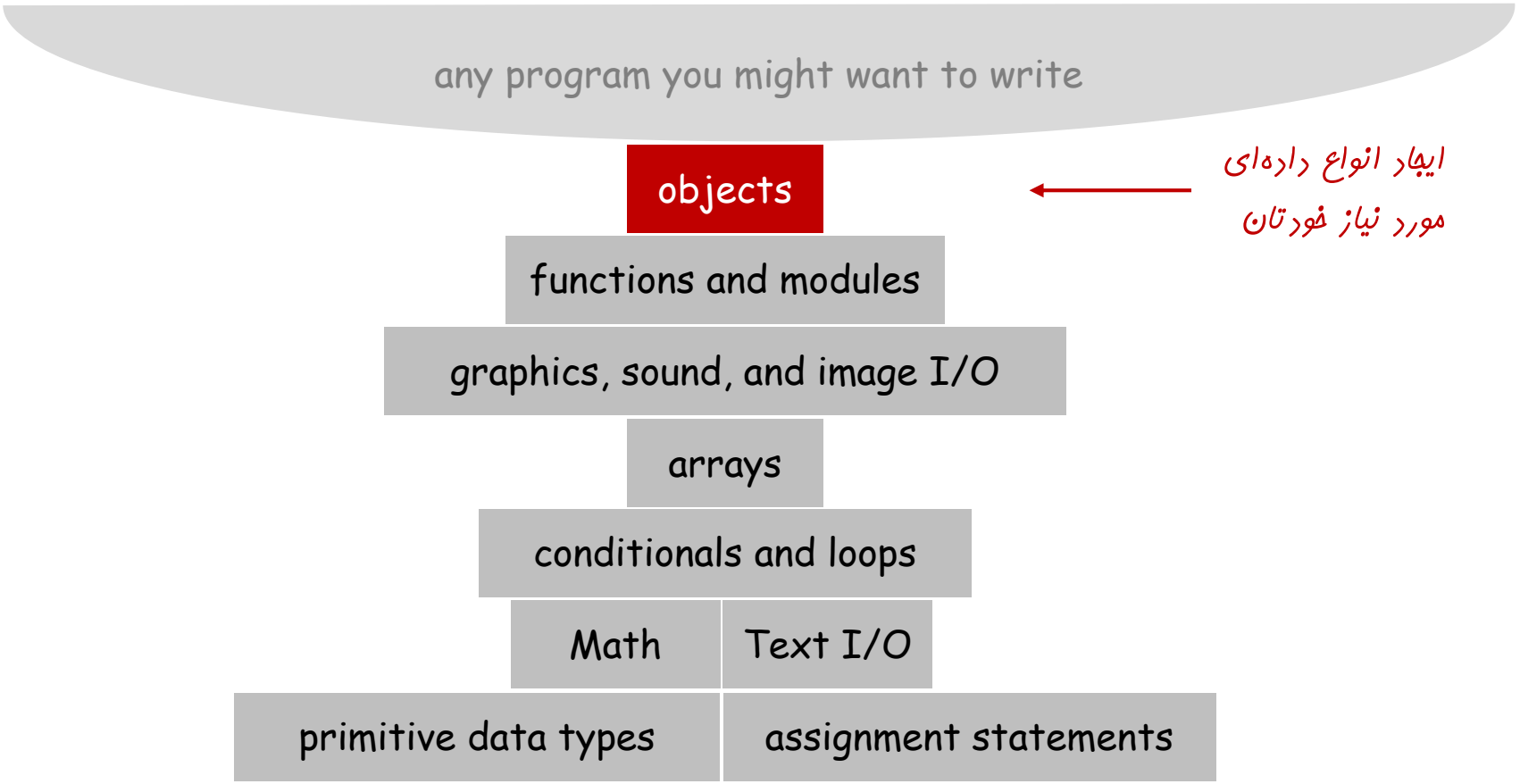
برنامه‌نویسی شی‌گرا:
شبیه‌سازی N-جسمی

سید ناصر رضوی **www.snrazavi.ir**
۱۳۹۶

۳-۴ شبیه‌سازی N-جسمی

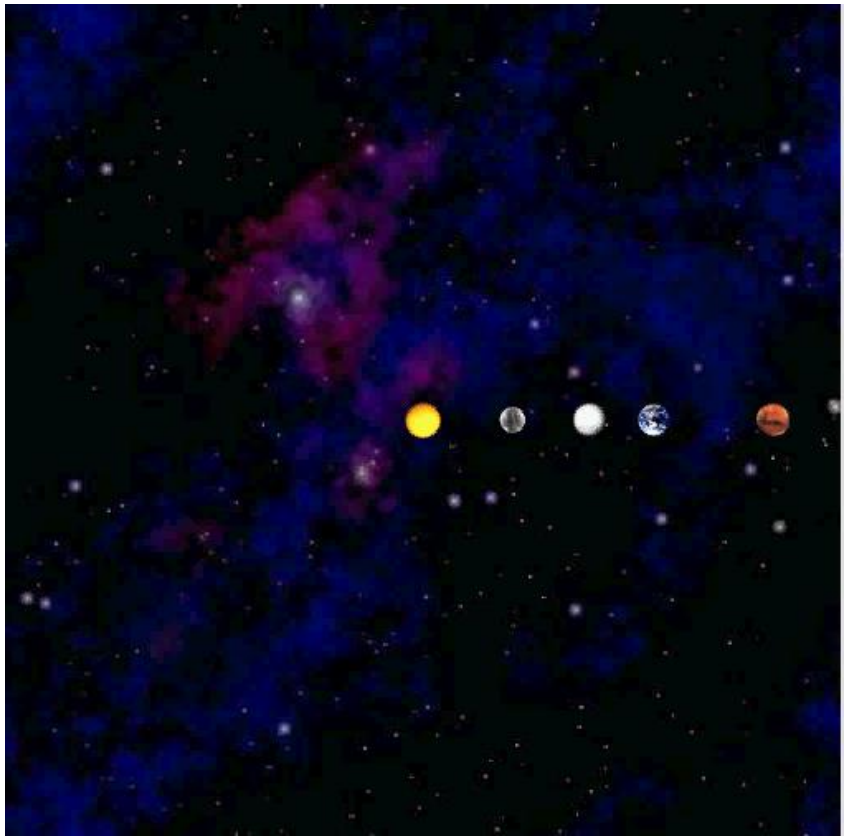


اجزای برنامه‌نویسی

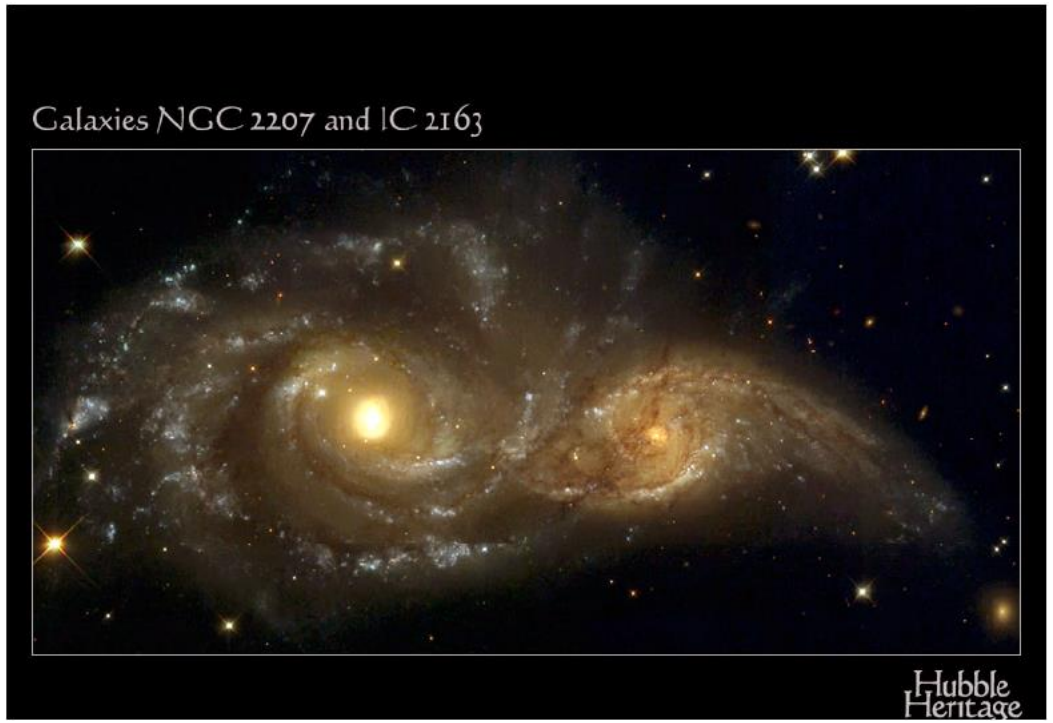


شبیه‌سازی N-جسمی

- هدف. تعیین چگونگی حرکت N ذره، در حالی که دو به دو تحت تأثیر نیروی جاذبه یکدیگر قرار دارند.



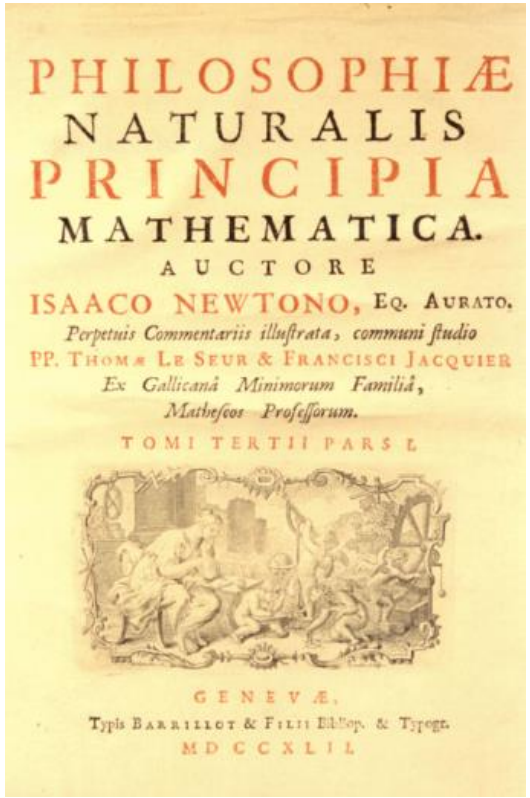
شبیه‌سازی N-جسمی: کاربردها



- کاربردها در اخترفیزیک.
  - مدار سیاره‌ها در سامانه خورشیدی.
  - دینامیک ستارگان در مرکز کهکشان.
  - دینامیک ستارگان در خوشه‌های کروی.
  - دینامیک ستارگان در برخورد دو کهکشان.
  - دینامیک کهکشان‌ها در زمان تشکیل خوشه.
  - تشکیل ساختارها در هستی.

مسئله N-جسمی

- هدف. تعیین چگونگی حرکت N ذره، در حالی که دو به دو تحت تأثیر نیروی جاذبه یکدیگر قرار دارند.
- زمینه. نیوتن در کتاب اصول خود، قوانین فیزیکی را بیان نمود.


PHILOSOPHIÆ
NATURALIS
PRINCIPIA
MATHEMATICA.
AUCTORE
ISAACO NEWTONO, Eq. Aurato.
Perpetuis Commentariis illustrata, communi studio
PP. Thomæ Le Seur & Francisci Jacquier
Ex Gallicanâ Minimorum Familiâ,
Matheseos Professorum.
TOMI TERTII PARS I.
GENEVÆ,
Typis Barrillot & Filii Bibliop. & Typogr.
MDCCXLII.

$$F = m\,a$$

قانون دوم نیوتن

$$F = \frac{G\,m_1 m_2}{r^2}$$

قانون جهانی جاذبه نیوتن


پوانکاره


دولونه


لاگرانژ


اویلر

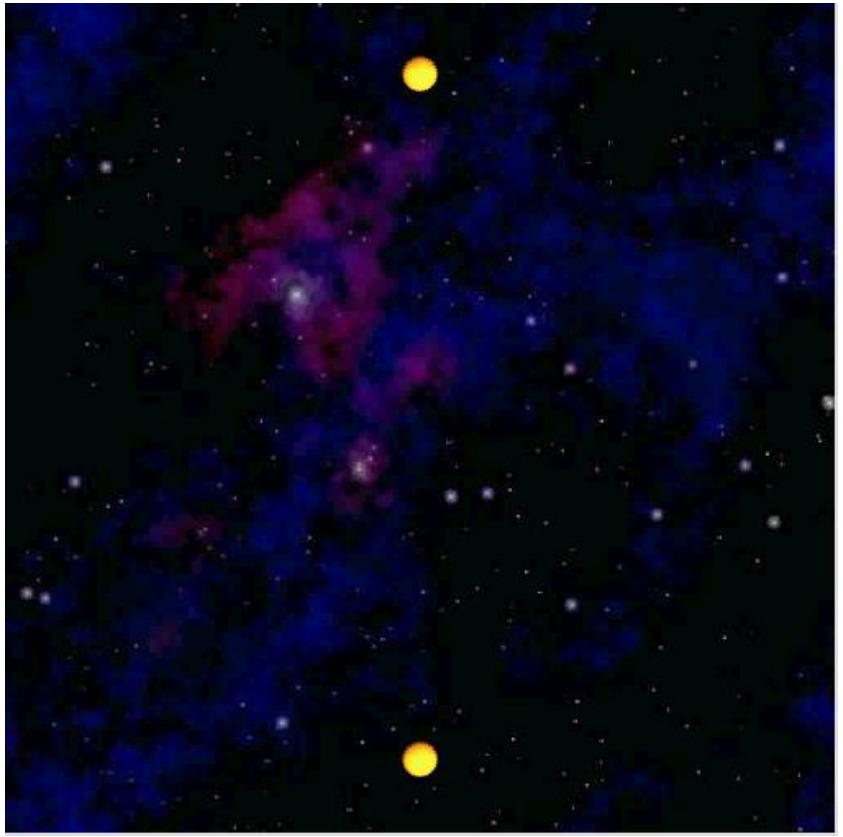

نیوتن


برنولی


کپلر

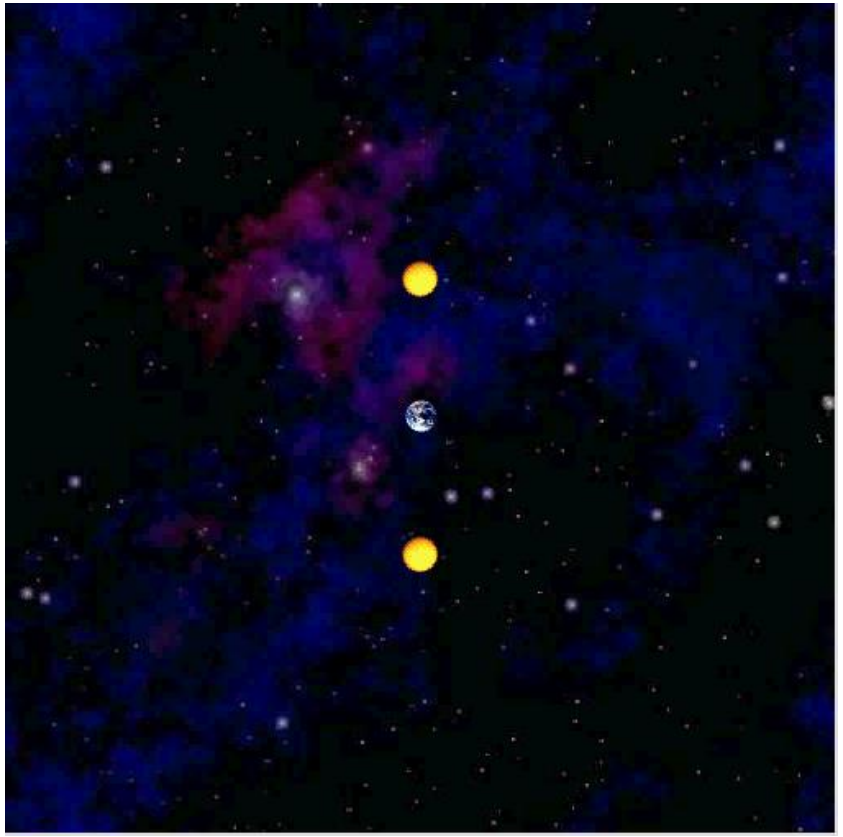
مسئله ۲-جسمی

- مسئله ۲-جسمی. با استفاده از قانون سوم کپلر به صورت تحلیلی قابل حل است.
  - دو جسم در مدارهای بیضوی حول یک مرکز ثقل مشترک گردش می‌کنند.



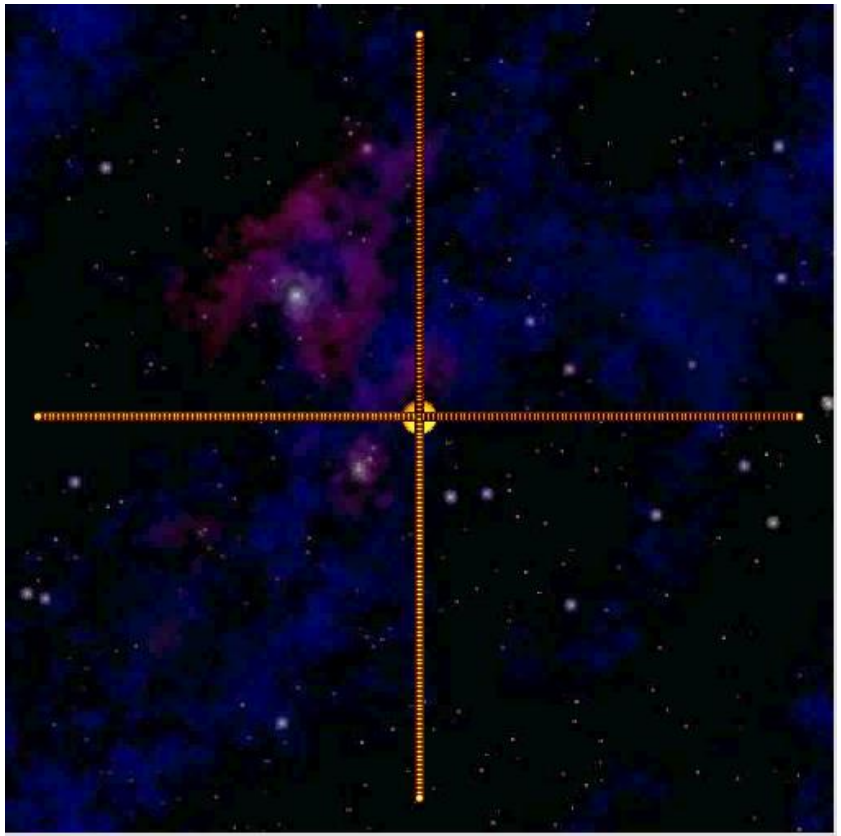
مسئله ۳-جسمی

- مسئله ۳-جسمی. عدم وجود راه‌حل برحسب توابع ابتدایی؛ مدارهای ناپایدار یا تناوبی!
- نتیجه. حل مسئله با استفاده از روش‌های محاسباتی.



شبیه‌سازی N-جسمی

- شبیه‌سازی N-جسمی. نهایت برنامه‌نویسی شی‌گرا:
  - شبیه‌سازی جهان هستی!



نوع داده‌ای جسم

- نوع داده‌ای جسم. برای مدل‌سازی یک ذره (جسم).

```
public class Body
```

```
        Body(Vector r, Vector v, double mass)
  void  move(Vector f, double dt)
  void  draw()
Vector  forceFrom(Body b)
```

اعمال نیروی **f**، *حرکت جسم برای* **dt** *ثانیه*

ترسیم جسم

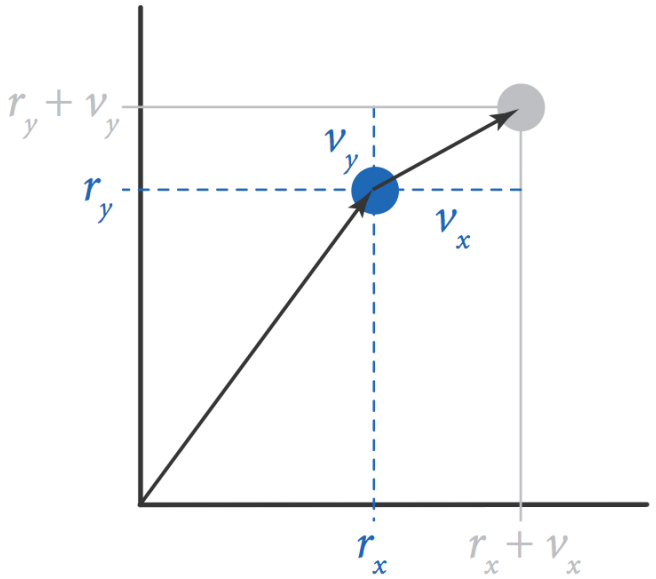
محاسبه نیروی وارد بر این جسم از طرف جسم **b**

- نوع داده‌ای بردار. نمایش موقعیت، سرعت و نیرو با استفاده از **Vector**.

```
public class Body {
   private Vector r;          // position
   private Vector v;          // velocity
   private double mass;       // mass
```

حرکت دادن یک جسم

- حرکت دادن یک جسم.
  - بدون وجود نیروهای دیگر، هر جسم در یک خط مستقیم حرکت می‌کند.



$$r_x = r_x + dt \cdot v_x$$

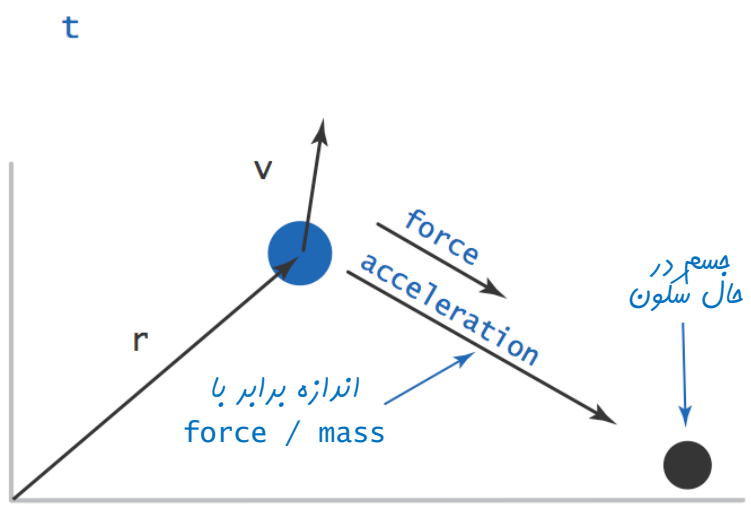
$$r_y = r_y + dt \cdot v_y$$

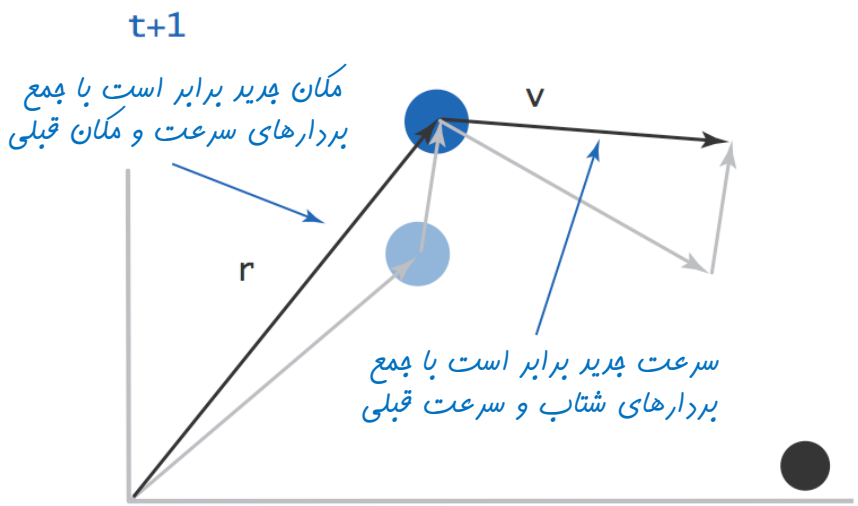
```
r = r.plus(v.times(dt));
```

حرکت دادن یک جسم

- حرکت دادن یک جسم.
  - با داشتن نیروی خارجی F، شتاب برابر است با $a = F/m$.
  - استفاده از شتاب به منظور محاسبه میزان تغییر سرعت.
  - استفاده از سرعت به منظور محاسبه میزان تغییر موقعیت.

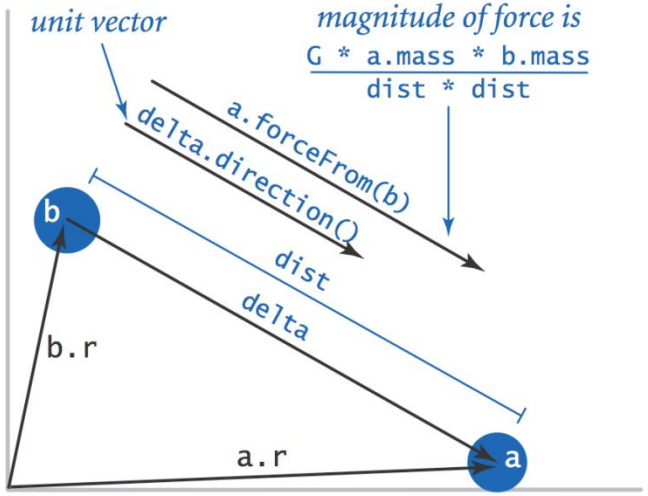
```
Vector a = f.times(1/mass);
v = v.plus(a.times(dt));
r = r.plus(v.times(dt));
```





نیروی بین دو جسم

- قانون جهانی جاذبه نیوتن.
  - اندازه نیرو برابر است با $F = Gm_1m_2/r^2$.
  - جهت نیرو در جهت خطی است که مرکز دو ذره را به هم وصل می‌کند.



Force from one body to another

```
double G = 6.67e-11;
Vector delta = a.r.minus(b.r);
double dist = delta.magnitude();
double F = (G * a.mass * b.mass) / (dist * dist);
Vector force = delta.direction().times(F);
```

نوع داده‌ای جسم: پیاده‌سازی


```
public class Body {
   private Vector r;                // position
   private Vector v;                // velocity
   private double mass;             // mass

   public Body(Vector r, Vector v, double mass) {
      this.r = r;
      this.v = v;
      this.mass = mass;
   }

   public void move(Vector f, double dt) {
      Vector a = f.times(1/mass);
      v = v.plus(a.times(dt));
      r = r.plus(v.times(dt));
   }

   public Vector forceFrom(Body that) {
      double G = 6.67e-11;
      Vector delta = that.r.minus(this.r);
      double dist = delta.magnitude();
      double F = (G * this.mass * that.mass) / (dist * dist);
      return delta.direction().times(F);
   }

   public void draw() {
      StdDraw.setPenRadius(0.025);
      StdDraw.point(r.cartesian(0), r.cartesian(1));
   }
}
```

متغیرهای نمونه

سازنده

مترها

نوع داده‌ای جهان

- نوع داده‌ای جهان. بیانگر جهانی متشکل از N ذره.

```
public class Universe
```

`Universe()`	
`void increaseTime(double dt)`	شبیه‌سازی گزشت dt ثانیه
`void draw()`	ترسیم جهان

```
public static void main(String[] args) {
   Universe newton = new Universe();
   double dt = Double.parseDouble(args[0]);
   while (true) {
      StdDraw.clear();
      newton.increaseTime(dt);
      newton.draw();
      StdDraw.show(10);
   }
}
```

حلقه اصلی شبیه‌سازی

نوع داده‌ای جهان

- نوع داده‌ای جهان. بیانگر جهانی متشکل از N ذره.

```
public class Universe
```

```
      Universe()
void  increaseTime(double dt)
void  draw()
```

شبیه‌سازی گزشت dt ثانیه

ترسیم جهان

```
public class Universe {

   private double radius;      // radius of universe
   private int N               // number of particles
   private Body[] orbs;        // the bodies
```

متغیرهای نمونه

طراحی مبتنی بر داده

سازنده

```
public Universe() {
   N = StdIn.readInt();
   radius = StdIn.readDouble();
   StdDraw.setXscale(-radius, +radius);
   StdDraw.setYscale(-radius, +radius);

   // read in the N bodies
   orbs = new Body[N];
   for (int i = 0; i < N; i++) {
      double rx   = StdIn.readDouble();
      double ry   = StdIn.readDouble();
      double vx   = StdIn.readDouble();
      double vy   = StdIn.readDouble();
      double mass = StdIn.readDouble();
      double[] position = { rx, ry };
      double[] velocity = { vx, vy };
      Vector r = new Vector(position);
      Vector v = new Vector(velocity);
      orbs[i] = new Body(r, v, mass);
   }
}
```

فایل ورودی

```
% more 4body.txt
4
5.0e10
-3.5e10 0.0e00 0.0e00  1.4e03 3.0e28
-1.0e10 0.0e00 0.0e00  1.4e04 3.0e28
 1.0e10 0.0e00 0.0e00 -1.4e04 3.0e28
 3.5e10 0.0e00 0.0e00 -1.4e03 3.0e28
```

اصل برهم نهی

- اصل برهم نهی.
  - نیروی خالص عمل کننده بر یک جسم برابر است با مجموع نیروهای فردی.

```
// compute the forces

for (int i = 0; i < N; i++) {
   for (int j = 0; j < N; j++) {
      if (i != j) {
         f[i] = f[i].plus(orbs[i].forceFrom(orbs[j]));
      }
   }
}
```

$$F_i = \sum_{i \neq j} \frac{G\, m_i\, m_j}{\left| r_i - r_j \right|^2}$$

نوع داده‌ای جهان: پیاده‌سازی


```
public class Universe {
   private final double radius;    // radius of universe
   private final int N;            // number of bodies
   private final Body[] orbs;      // array of N bodies

   public Universe() { /* see previous slide */ }

   public void increaseTime(double dt) {
      Vector[] f = new Vector[N];
      for (int i = 0; i < N; i++)
      f[i] = new Vector(new double[2]);
      for (int i = 0; i < N; i++)
         for (int j = 0; j < N; j++)
            if (i != j)
               f[i] = f[i].plus(orbs[j].forceTo(orbs[i]));
      for (int i = 0; i < N; i++)
         orbs[i].move(f[i], dt);
   }

   public void draw() {
      for (int i = 0; i < N; i++)
      orbs[i].draw();
   }
}
```

سازنده

به روز رسانی

ترسیم اجسام

اندکی جزییات بیشتر

- دقت. مقدار dt چقدر باید کوچک باشد؟ چگونه می‌توان از خطاهای اعداد اعشاری که بر روی هم جمع می‌شوند، اجتناب نمود؟

- کارایی.
  - جمع مستقیم: زمان متناسب با N^2.
    ⟸ برای N های بزرگ قابل استفاده نیست.
  - الگوریتم بارنز-هوت: زمان متناسب با $N \lg N$.
    ⟸ قابل استفاده برای تعداد بسیار زیاد اجسام.

- شبیه‌سازی ۳-بعدی. استفاده از بردارهای ۳-بعدی (کد مربوط به ترسیم تغییر می‌کند!).

- پارامتر نرم‌سازی. برای اجتناب از تصادم‌ها.

$$F_i = \sum_{i \neq j} \frac{G\ m_i\ m_j}{|r_i - r_j|^2 + \varepsilon^2}$$