

Создайте потоковые анимации частицы

Путь к снаряду в зависимости от времени

Необходимо отобразить путь снаряда в зависимости от времени с помощью 3D графика полей градиента.

Покажите путь следующего снаряда с помощью констант для скорости и ускорения, v_z и a . Вычислите z как высота, когда время варьируется от 0 до 1.

$$z(t) = v_z t + \frac{a}{2} t^2$$

```
vz = 10; % velocity constant
a = -32; % acceleration constant
t = 0:.1:1;
z = vz*t + 1/2*a*t.^2;
```

Вычислите положение в *направлении X* и *направлении Y*.

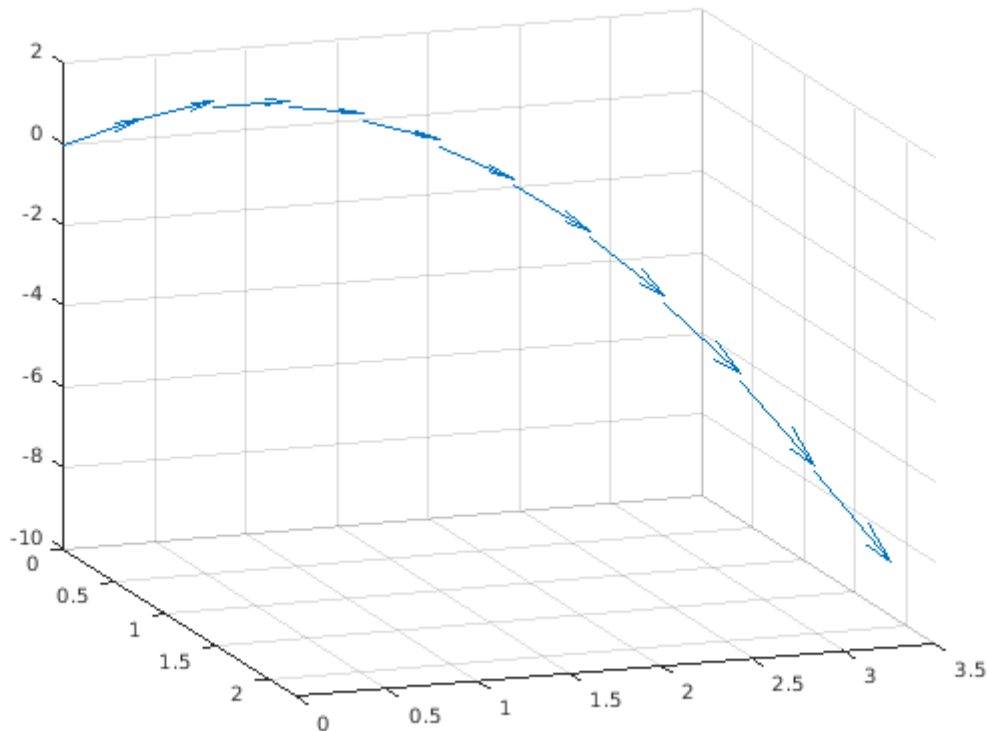
```
vx = 2;
x = vx*t;
```

```
vy = 3;
y = vy*t;
```

Вычислите компоненты векторов скорости и отобразите векторы с помощью 3-D графика полей градиента. Измените точку зрения осей к [70,18].

```
u = gradient(x);
v = gradient(y);
w = gradient(z);
scale = 0;

figure
quiver3(x,y,z,u,v,w,scale)
view([70,18])
```



Что могут показать анимации частицы

Потоковая анимация частицы полезна для визуализации направления потока и скорости векторного поля. “Частицы” (представленные любым из маркеров линии) прослеживают поток вдоль конкретной линии потоков. Скорость каждой частицы в анимации пропорциональна величине векторного поля в любой данной точке вдоль линии потоков.

1. Определение отправных точек области значений данных

Этот пример определяет область объема, чтобы построить путем определения соответствующих начальных точек. В этом случае потоковые графики начинаются в $x = 100$, и y охватывает 20 - 50 в $z = 5$ плоскостей, которые не являются полными границами объема.

```
load wind
[sx sy sz] = meshgrid(100,20:2:50,5);
```

2. Создание линий потоков, чтобы указать на пути к частице

Этот пример использует линии потоков (`stream3`, `streamline`) проследить путь анимированных частиц, который добавляет визуальный контекст для анимации.

```
verts = stream3(x,y,z,u,v,w,sx,sy,sz);
sl = streamline(verts);
```

3. Определение представления

В то время как все линии потоков запускаются в $z = 5$ плоскостей, значения некоторых постепенно снижаются к нижним значениям. Следующие настройки обеспечивают четкое представление об анимации:

- Точка зрения (`view`) выбранный показывает и плоскость, содержащую большинство линий потоков и спираль.

- Выбор соотношения сторон данных (`daspect`) из `[2 2 0.125]` обеспечивает большее разрешение в *z-направлении*, чтобы сделать потоковые частицы более легко видимыми в спирали.
- Установите пределы осей, чтобы совпадать с пределами данных (`axis`) и чертите поле оси (`box`).
 - `view(-10.5,18)`
 - `daspect([2 2 0.125])`
 - `axis tight;`
 - `set(gca, 'BoxStyle', 'full', 'Box', 'on')`

4. Вычисление потоковых вершин частицы

Определите вершины вдоль линии потоков, где частица будет чертиться. `interpstreamspeed` функция возвращает эти данные на основе вершин линии потоков и скорости векторных данных. Этот пример масштабирует скорости 0,05, чтобы увеличить число интерполированных вершин.

Установите оси `SortMethod` свойство к `childorder` таким образом, анимация запускается быстрее.

`streamparticles` функционируйте устанавливает следующие свойства:

- `Animate` к 10 запускать анимацию 10 раз.
- `ParticleAlignment` к `on` запустить все трассировки частицы вместе.
- `MarkerEdgeColor` к `none` чертить только поверхность кругового маркера. Анимации обычно запускаются быстрее, когда ребра маркера не чертятся.
- `MarkerFaceColor` к `red`.
- `Marker` к `o`, который чертит круговой маркер. Можно использовать другие маркеры линии также.
 - `iverts = interpstreamspeed(x,y,z,u,v,w,verts,0.01);`
 - `set(gca, 'SortMethod', 'childorder');`
 - `streamparticles(iverts,15,...`
 - `'Animate',10,...`
 - `'ParticleAlignment','on',...`
 - `'MarkerEdgeColor','none',...`
 - `'MarkerFaceColor','red',...`
 - `'Marker','o');`

