

In [1]:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 rng = np.random.default_rng()
```

在使用周期性边界条件时，两个粒子有无数多的像(image)，所以不能直接计算出所有像之间的距离，再寻找最小距离

可以先选定一个粒子的一个像作为参考，寻找与之最近的另一个粒子的像，两个像最近时距离一定小于1/2周期，可以以此为判断依据

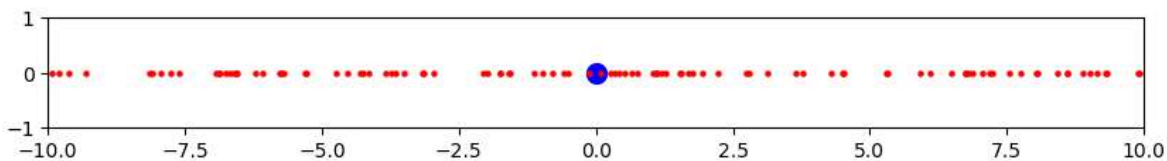
In [2]:

```
1 def pbc(r, ref, L):
2     while r - ref > L/2:
3         r -= L
4     while ref - r > L/2:
5         r += L
6     return r
```

选取一个参考像和一些随机的坐标

In [3]:

```
1 x_ref = 0
2 x_array = rng.uniform(-10, 10, 100)
3
4 plt.figure(figsize=(10, 1), dpi=100)
5 plt.scatter(x_ref, 0, s=100, color='blue')
6 plt.scatter(x_array, np.zeros_like(x_array), s=5, color='red')
7 plt.xlim((-10, 10))
8 plt.ylim((-1, 1))
9 plt.show()
```



假设周期长度为  $L = 5.0$ ，用上面的函数处理可以找到这些随机点距离参考像最近的像

In [4]:



```
1 L = 5.0
2 x_image = np.array([pbc(x, x_ref, L) for x in x_array])
3
4 plt.figure(figsize=(10, 1), dpi=100)
5 plt.scatter(x_ref, 0, s=100, color='blue')
6 plt.scatter(x_image, np.zeros_like(x_image), s=5, color='red')
7 plt.xlim((-10, 10))
8 plt.ylim((-1, 1))
9 plt.show()
```

