HTML 5模拟行波叠加实验记录

孙煜婷 17307110451

课题设计

通过动画代码画出行进的正弦波，设计调整振幅、频率、波长滑块按钮改变相关参数从而改变叠加波的形式。

对于振动方向相同、传播方向相同、频率不同的的两列波，$\varphi_1 = A_0 cos(\omega_1 t - k_1 z)$、$\varphi_2 = A_0 cos(\omega_2 t - k_2 z)$，叠加可得：

$$\varphi = 2A_0 cos(\frac{\Delta\omega}{2}t - \frac{\Delta k}{2}z)cos(\bar{\omega}t - \bar{k}z)$$

叠加形成波形如图1所示。

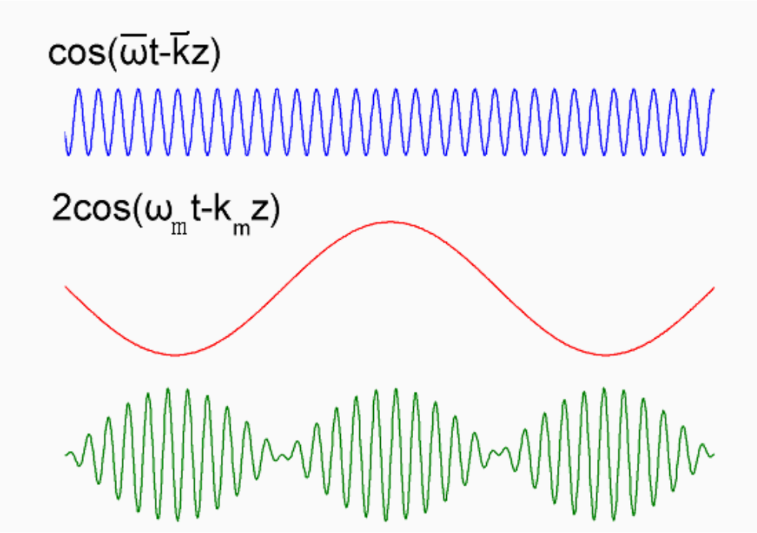


图1 振动方向相同、传播方向相同、频率不同的的两列波叠加

不同频率单色光叠加形成拍。

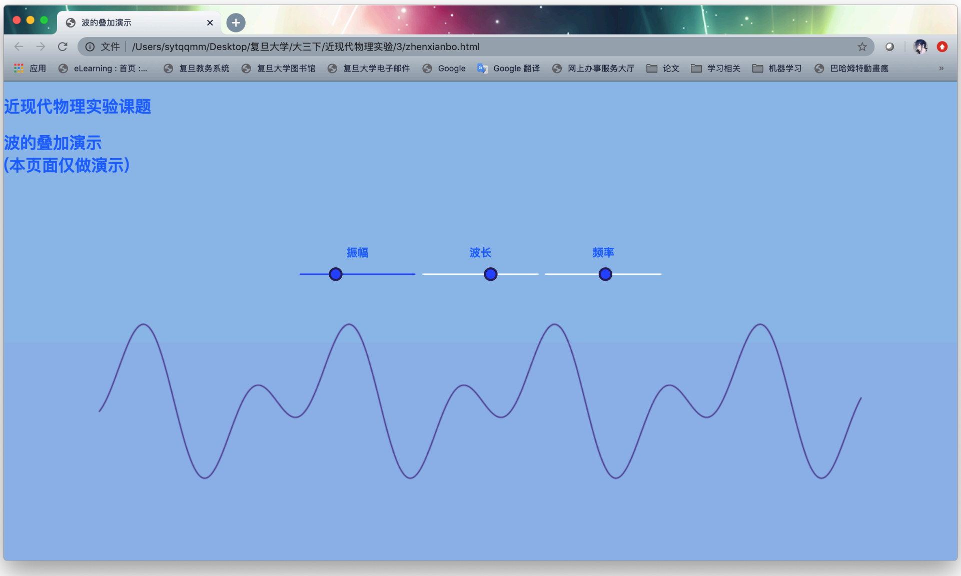
实验过程和结果

利用$class =" control"$，分类三个参数标签（$label$）：振幅、频率、波长，并设置取值范围（$type =" range"$）。将三个参数写入正弦波的驻波方程中。

设置滑块控制三个参数的取值。首先用$-webkit - appearance : none$去除系统默认滑块样式，这步操作是为了不使用原有的样式，使之后的自定义样式有效；然后用代码函数$input[type = range] :: - webkit - slider - runnable - track\{\}$定义滑块控件轨道，并给滑块添加样式$input[type = range] :: - webkit - slider - thumb\{\}$。

设置驻波动画，设定x横坐标参数，x坐标增加，对应驻波y坐标改变，画出动画。由于y坐标中振幅、频率、波长的参数由滑块控件确定参数值，当滑块滑动改变赋值，y的值也将改变，从而起到演示驻波行进以及参数改变时驻波样式改变的作用。

完整网页如图2所示。

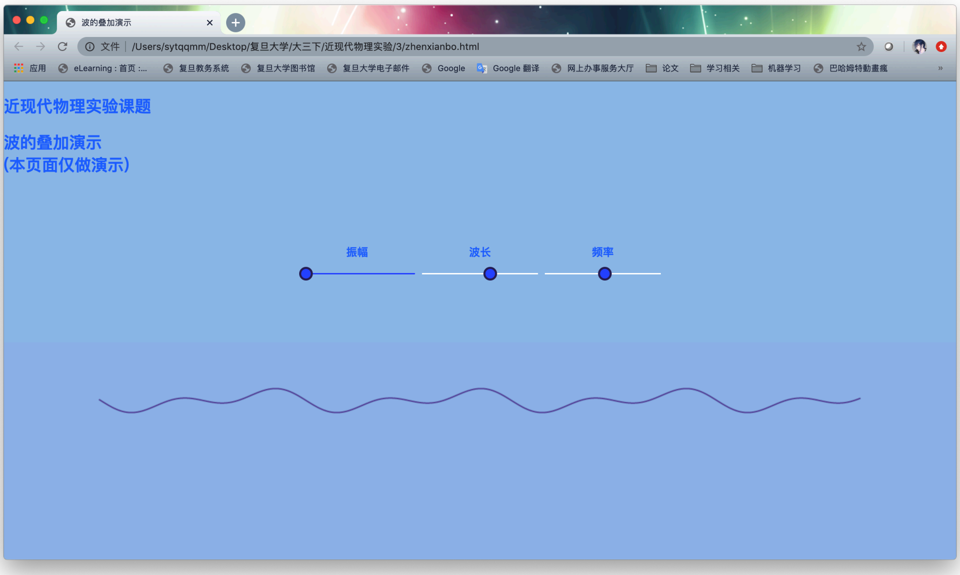


近现代物理实验课题

波的叠加演示
(本页面仅做演示)

图2 网页样式

拉动振幅滑块控件如图3所示。

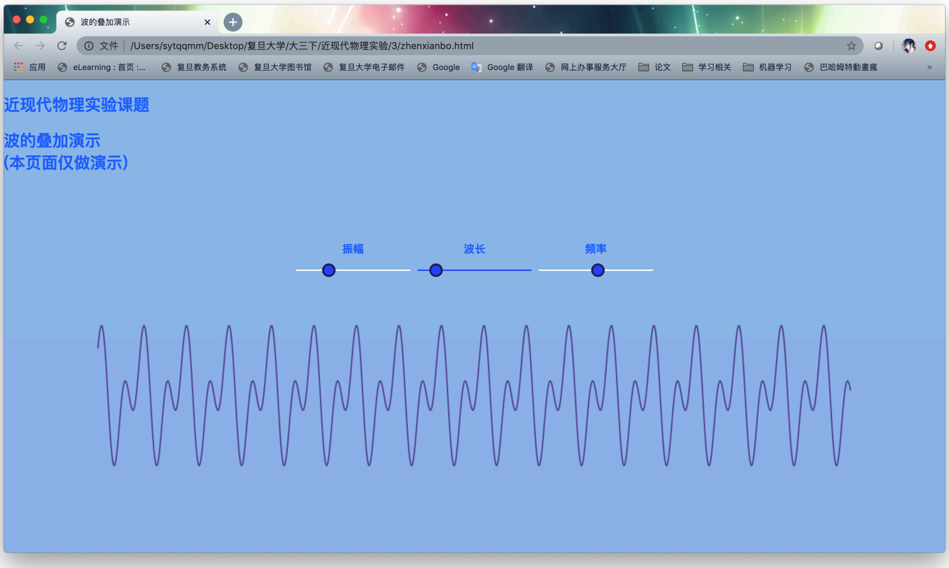


近现代物理实验课题

波的叠加演示
(本页面仅做演示)

图3 拉动滑块改变振幅大小

拉动波长滑块控件如图4所示。



近现代物理实验课题

波的叠加演示
(本页面仅做演示)

图4 拉动滑块改变波长大小

拉动频率滑块控件，叠加波行进快慢改变。

实验完整代码为：

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>波的叠加演示</title>


  <style>
    body {
      background-color:#0661c979;
      color: #225dff;
      margin: 0;
      overflow: hidden;
      font-weight: bold;
    }

    .controls {
 display: -webkit-box;
 display: -ms-flexbox;
```

```
  display: flex;
  -webkit-box-pack: space-evenly;
      -ms-flex-pack: space-evenly;
          justify-content: space-evenly;
  -webkit-box-orient: horizontal;
  -webkit-box-direction: normal;
      -ms-flex-direction: row;
          flex-direction: row;
  width: 550px;
  margin: 0 auto;
  margin-top: 100px;
}

.control {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  -webkit-box-orient: vertical;
  -webkit-box-direction: normal;
      -ms-flex-direction: column;
          flex-direction: column;
  -webkit-box-pack: center;
      -ms-flex-pack: center;
          justify-content: center;
  width: 170px;
  text-align: center;
}
```

/* 自定义滑块条参考资料：https://blog.csdn.net/u013347241/article/details/51560290 */

```
        input[type=range] {
            -webkit-appearance: none;
            width: 100%;
            margin: 10px 0;
            margin-top: 20px;
        }/* 去除系统默认滑块样式 */

        input[type=range]:focus {
            outline: none;
        }

        input[type=range]::-webkit-slider-runnable-track {
            width: 100%;
            height: 2.2px;
            cursor: pointer;
            border-radius: 2.3px;
            border: 0 solid #2243ff;
        }/*自定义滑动控件的轨道（track）*/

        input[type=range]::-webkit-slider-thumb {
            -webkit-box-shadow: 0 0 0 #291d53, 0 0 0 #523d96;
                box-shadow: 0 0 0 #291d53, 0 0 0 #523d96;
            border: 3px solid #291d53;
            height: 20px;
            width: 20px;
            border-radius: 20px;
            background: #2243ff;
            cursor: pointer;
            -webkit-appearance: none;
            margin-top: -8.9px;
        }/*定义滑块（thumb）样式*/

        input[type=range]:focus::-webkit-slider-runnable-track {
            background: #2243ff;
        }

    </style>
</head>
```

```
<body>
  <h2>近现代物理实验课题</h2>
  <h2>波的叠加演示<br>(本页面仅做演示)</h2>

  <section class="controls">
    <div class="control">
     <label for="amplitude">振幅</label>
     <input id="amplitude" type="range" min="10" max="200" value="30">
    </div>
    <div class="control">
     <label for="wavelength">波长</label>
     <input id="wavelength" type="range" min="3" max="500" value="150">
    </div>
    <div class="control">
     <label for="frequency">频率</label>
     <input id="frequency" type="range" min="1" max="30" value="10">
    </div>
   </section>

  <canvas id="canvas"></canvas>

  <script>
    var canvas = document.getElementById('canvas');
    var ctx = canvas.getContext('2d');

    var amplitudeRange = document.getElementById('amplitude');
    var waveLegthRange = document.getElementById('wavelength');
    var frequencyRange = document.getElementById('frequency');

    var posX = 0;
    var posY1 = 0;
    var posY2 = 0;
    var posY3 = 0;
    var diffX = 0;
```

```
updateCanvas();

var amplitude = Number(amplitudeRange.value);
var frames = 0;
var frequency = Number(frequencyRange.value);
var wavelength = Number(waveLegthRange.value);

amplitudeRange.addEventListener('change', function (evt) {
    amplitude = Number(evt.currentTarget.value);
});

waveLegthRange.addEventListener('change', function (evt) {
    wavelength = Number(evt.currentTarget.value);
});

frequencyRange.addEventListener('change', function (evt) {
    frequency = Number(evt.currentTarget.value);
});

animate();

window.addEventListener('resize', onResize);

function updateCanvas() {
    canvas.width = window.innerWidth;
    canvas.height = window.innerHeight / 2;
    diffX = canvas.width / 2.5;

    posX = canvas.width * .5;
    posY1 = canvas.height * .5;
    posY2 = canvas.height * .5;
    posY3 = canvas.height * .5;

    /*ctx.fillRect(0, 0, canvas.width, canvas.height);
    ctx.strokeStyle = '#523d96';
```

```
            ctx.lineWidth = 2;
            ctx.moveTo(posX, posY1);

            ctx.fillRect(0, 0, canvas.width, canvas.height);
            ctx.strokeStyle = '#523d96';
            ctx.lineWidth = 2;
            ctx.moveTo(posX, posY2);*/

            ctx.fillRect(0, 0, canvas.width, canvas.height);
            ctx.strokeStyle = '#523d96';
            ctx.lineWidth = 2;
            ctx.moveTo(posX, posY3);
        }

        /*function calcPosY1(x) {
            return Math.sin((x / wavelength * (Math.PI * 2)) + frames / frequency) * amplitude;
        }

        function calcPosY2(x) {
            return Math.sin((x / wavelength * (Math.PI * 4)) + 2*frames / frequency) * amplitude;
        }*/

        function calcPosY3(x) {
            return (Math.sin((x / wavelength * (Math.PI * 2)) + frames / frequency) * amplitude)
+(Math.sin((x / wavelength * (Math.PI * 4)) + 2*frames / frequency) * amplitude);
        }

        function animate() {
            frames++

            ctx.clearRect(0, 0, canvas.width, canvas.height);
            ctx.beginPath();
            ctx.lineWidth = 2;

            for (var x = posX - diffX; x < canvas.width / 2 + diffX; x++) {
                var y3 = calcPosY3(x);
                ctx.lineTo(x, y3 + posY3);
            }

            ctx.stroke();

            requestAnimationFrame(animate.bind(this));
        }

        function onResize() {
            updateCanvas();
        }
    </script>

</body>
</html>
```

简要分析

由于y坐标设置的形式就是驻波方程，尽管改变相关三个参数，驻波的基本形状却无法改变，不能演示各种样式的驻波。

除了驻波，还有多种正弦波叠加交互可以设计。