Initialization.

In[10]:=
```
fs = 18;
```

In[11]:=
```
ff = Times;
```

In[12]:=
```
mystyle[text_] := Style[text, FontSize → fs, FontFamily → ff]
```

First-taste example.

Defining boundary value problem.

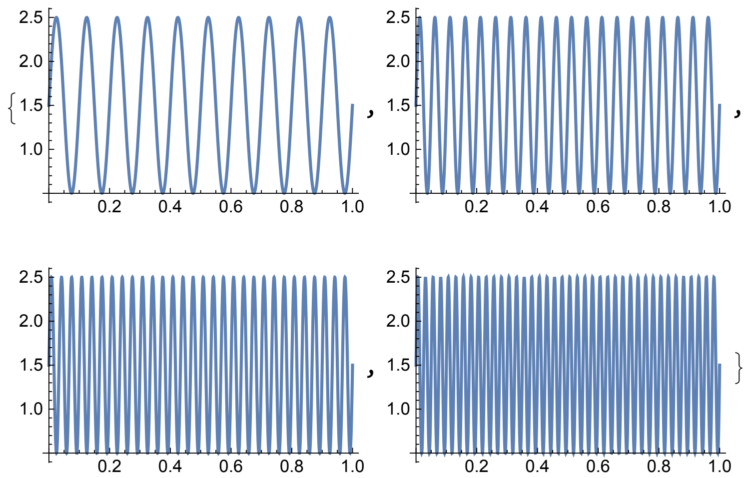
In[1]:=
```
L = 1;
```

In[2]:=
```
c[k_, x_] := 3/2 + Sin[2 π k x];
```

In[3]:=
```
f[x_] = x;
```

In[5]:=
```
Table[Plot[c[10 k, x], {x, 0, L}], {k, 1, 4}]
```

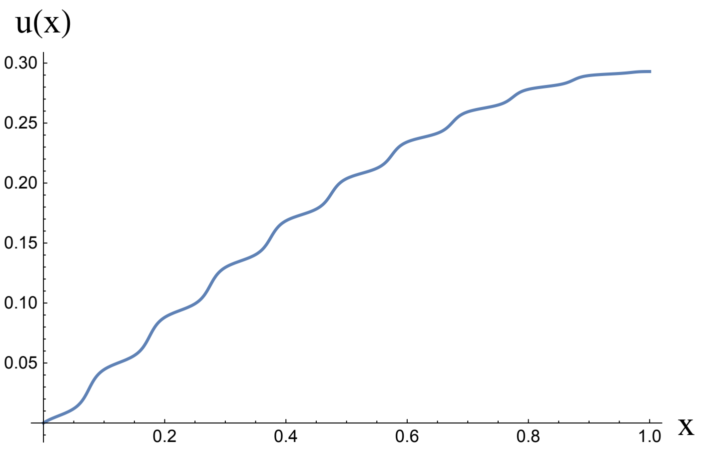
Out[5]=



In[62]:=
```
nsol = NDSolve[{D[c[10, x] u'[x], x] + f[x] == 0, u[0] == 0, u'[L] == 0},
   u[x], {x, 0, L}, AccuracyGoal → 20];
```

In[23]:=
```
Plot[u[x] /. nsol[[1]], {x, 0, L}, ImageSize → Medium,
 AxesLabel → {mystyle["x"], mystyle["u(x)"]}]
```

Out[23]=

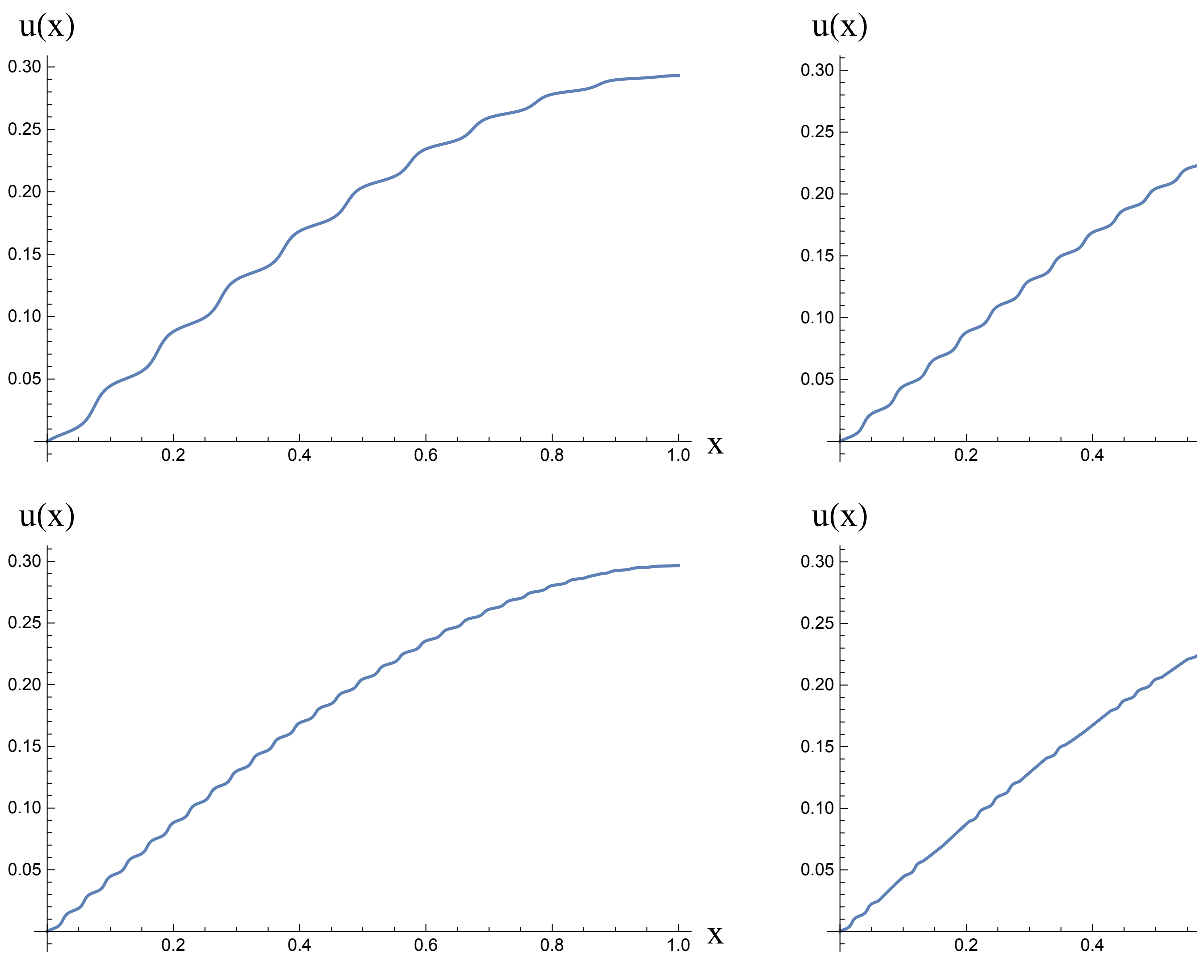


List of equations with small wavelengths of material parameters.

In[28]:=
```
nsolList = Table[NDSolve[
    {D[c[10 k, x] u'[x], x] + f[x] == 0, u[0] == 0, u'[L] == 0}, u[x], {x, 0, L}], {k, 1, 4}];
```

In[29]:=
```
GraphicsGrid[Partition[Table[
   Plot[u[x] /. nsolList[[j, 1]], {x, 0, 1}, AxesLabel → {mystyle["x"], mystyle["u(x)"]}],
   {j, Length[nsolList]}], 2], ImageSize → 800]
```

Out[29]=

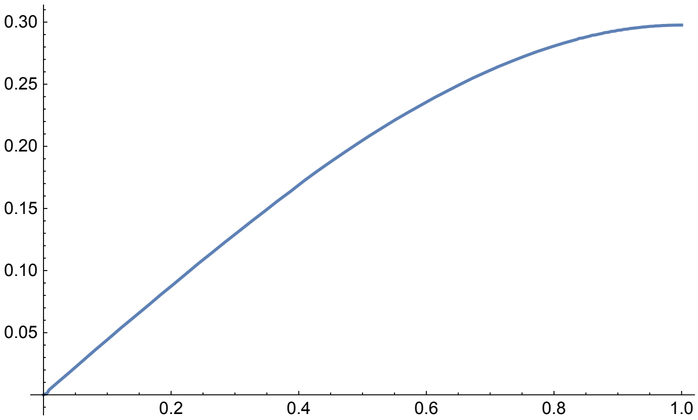


In[63]:=
```
nsol2 =
  NDSolve[{D[c[10 × 10, x] u'[x], x] + f[x] == 0, u[0] == 0, u'[L] == 0}, u[x], {x, 0, L}];
```

NDSolve: The scaled boundary value residual error of 11.878190169874852` indicates that the boundary values are not satisfied to specified tolerances. Returning the best solution found.

In[33]:=
```
Plot[u[x] /. nsol2[[1]], {x, 0, L}]
```

Out[33]=



In[26]:=
```
Clear[u, k]
```

In[60]:=
```
Re[ComplexExpand[
   DSolve[{D[c[k, x] u'[x], x] + f[x] == 0, u[0] == 0, u'[L] == 0}, u[x], {x, 0, L}]]];
```

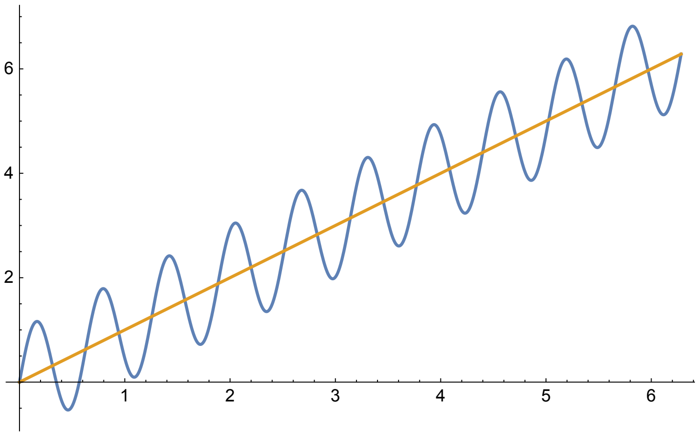
In[61]:=
```
g[x_] = Sin[10 x] + x;
```

In[35]:= $\lambda = 2\,\frac{\pi}{10};$

In[64]:= $\text{fbar}[x\_] = \frac{1}{\lambda}\int_{x-\lambda/2}^{\lambda/2+x} g[\tau]\,d\tau;$

In[65]:=
```
Plot[{g[x], fbar[x]}, {x, 0, 2 π}]
```

Out[65]=



Solve the microscale periodic boundary value problem.

```
k = 1 / 2 π;
λ = 1 / k;
c[k_, x_] := 3 / 2 + Sin[2 π k x];
```

```
strain = 0.025;
```

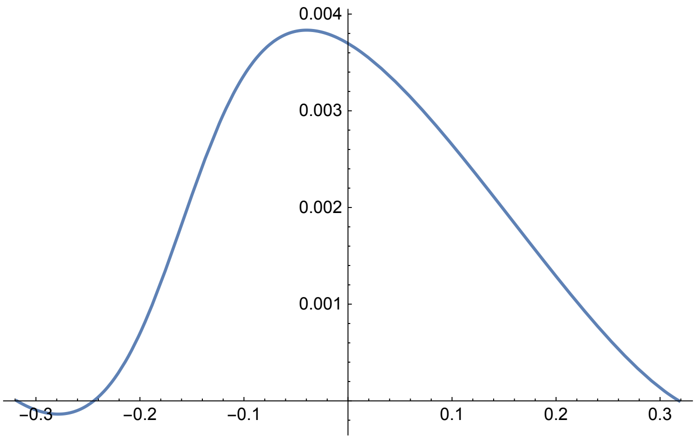
```
nsol = NDSolve[
  {D[c[k, x] (v'[x] + strain), x] == 0, v[-λ/2] == 0, v[λ/2] == 0}, v[x], {x, -λ/2, λ/2}]
```

```
{{v[x] → InterpolatingFunction[ Domain: {{-0.318, 0.318}}
                                Output: scalar ][x]}}
```

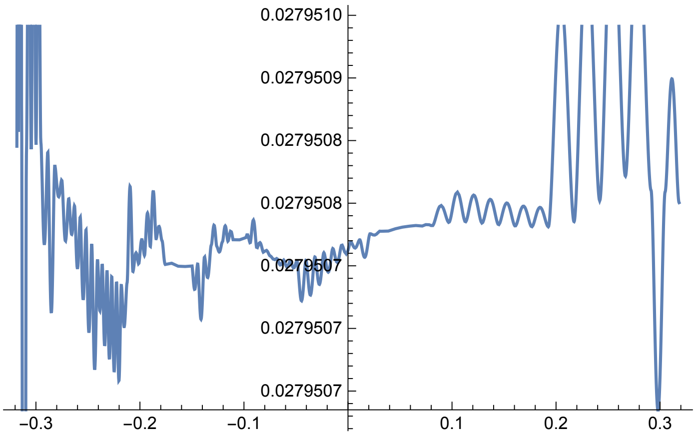
```
u[x_] = v[x] /. nsol[[1]];
```

```
σ[x_] = c[k, x] (D[u[x], x] + strain);
```

```
Plot[u[x], {x, -λ/2, λ/2}]
```




```
Plot[σ[x], {x, -λ/2, λ/2}]
```




```
1/λ NIntegrate[σ[x], {x, -λ/2, λ/2}]
```

```
0.0279508
```

```
1/1 (1 - r) /. {r → 0}
```