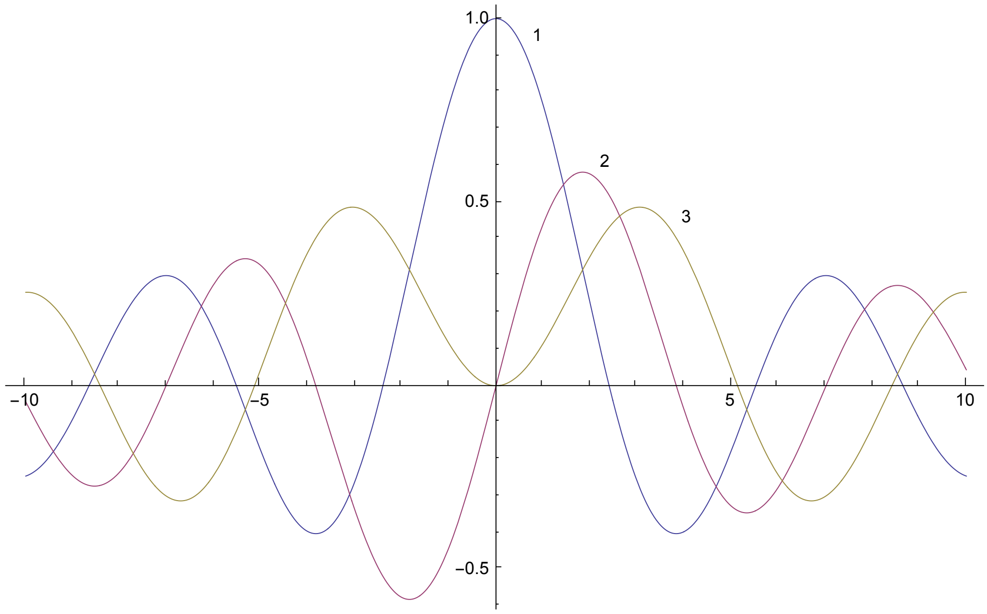
```
Clear[f]
```

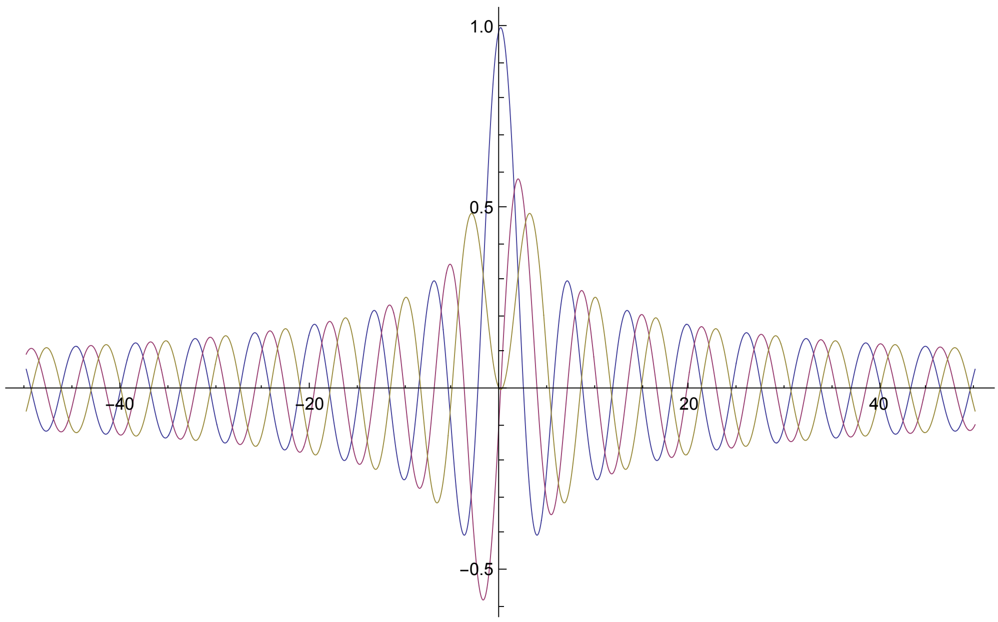
$$f[r\_] = f[r] \text{ /. } \mathbf{DSolve}\left[\left\{f''[r] + \frac{1}{r} f'[r] + \left(k^2 - \frac{m^2}{r^2}\right) f[r] == 0\right\}, f[r], r\right][[1]]$$

```
BesselJ[m, k r] C[1] + BesselY[m, k r] C[2]
```

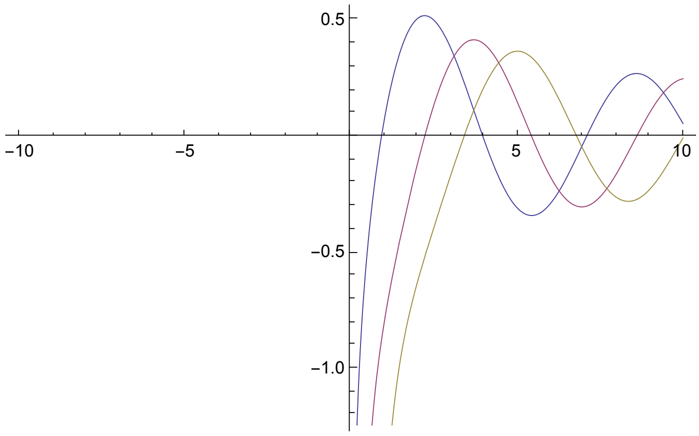
Note que a solução acima não é a mais geral, pois falta a soma em m.

```
Plot[{BesselJ[0, x], BesselJ[1, x], BesselJ[2, x]}, {x, -10, 10}]
Plot[{BesselJ[0, x], BesselJ[1, x], BesselJ[2, x]},
 {x, -50, 50}, PlotRange → All]
```





```
Plot[{BesselY[0, x], BesselY[1, x], BesselY[2, x]}, {x, -10, 10}]
```



Como f[0] é finito, vamos eliminar a contribuição das funções de Neumann

```
f[r_] = BesselJ[m, k r] Const
```

```
Const BesselJ[m, k r]
```

Vamos agora encontrar as raízes da função de Bessel, a fim de impor a condição f[R] = 0

```
Reduce[ BesselJ[m, k R] == 0, k]
```

Reduce::nsmet: This system cannot be solved with the methods available to Reduce. >>

```
Reduce[BesselJ[m, k R] == 0, k]
```

“Reduce” não resolve este problema, embora resolva um análogo:

```
Reduce[Sin[x] == 0, x]
```

```
C[1] ∈ Integers && (x == 2 π C[1] || x == π + 2 π C[1])
```

Para encontrar as raízes da função de Bessel, usa-se métodos numéricos, por exemplo:

```
N[BesselJZero[0, 1]]
N[BesselJZero[0, 2]] (* São a primeira e a segunda raizes de BesselJ[0,x]*)
```

```
2.40483
```

```
5.52008
```

```
FindRoot[BesselJ[0, x] == 0, {x, 1}]
(* Procura por raízes na vizinhança de x == 1*)

FindRoot[BesselJ[0, x] == 0, {x, 5}]
(* Procura por raízes na vizinhança de x == 5*)
```

```
{x → 2.40483}
```

```
{x → 5.52008}
```