Hydrogen wave functions

see **wikipedia**

```
# GNU scientific library, for Legendre and Laguerre
using GSL
```

```
# interactive Plots
using PlotlyJS
```

```
# Pluto: simple slider
using PlutoUI
```

```
# radial part of the wave function
function radial(r; qn)
    a0=1; #for convenience, or 5.2917721092(17)×10-11 m
    ρ  = 2 * r / (qn.n * a0) # reduced radial coodinate

    return sf_laguerre_n(qn.n  - qn.l - 1, 2 * qn.l + 1, ρ) * e^(-ρ/2) * ρ^qn.l
end;
```

```
# Hydrogen wave function in spherical coordinates
function psi(r, θ, ϕ, qn)

    # complex valued
    # phasefac = e^(im * qn.m * ϕ)

    # real-valued superpositions
    if (qn.m < 0)
        phasefac = sin(abs(qn.m) * ϕ)
    else
        phasefac = cos(abs(qn.m) * ϕ)
    end

    Yml = (-1)^qn.m * sf_legendre_Plm(qn.l, abs(qn.m), cos(θ))

    #norm = sqrt( (2/qn.n)^3 * factorial(qn.n - qn.l - 1) /
    #   (2 * qn.n * factorial(qn.n + qn.l) ) )

    return radial(r; qn = qn) * Yml * phasefac
end;
```

```
#Hydrogen wave function in cartesian coordinates
function psi_cart(x,y,z; qn)
    r=sqrt(x^2 + y^2 + z^2)
    θ=acos(z/r) # [0,π].
    #ϕ=atan(y/x) # [-π/2,π/2].   # we need to cover the whole sphere
    ϕ=atan(y, x) # [-π , π].

    return psi( r, θ, ϕ, qn)
end;
```

```
# evaluate wave function on 3D grid
begin
    xs = range(-15,15; length=50)
    X, Y, Z = mgrid(xs, xs, xs)  # X contains x-coordinates of all points, etc.
    values = @.  real(psi_cart(X,Y,Z; qn=qn))
end;
```

1

```
@bind nx Slider(1:4; show_value=true)
```

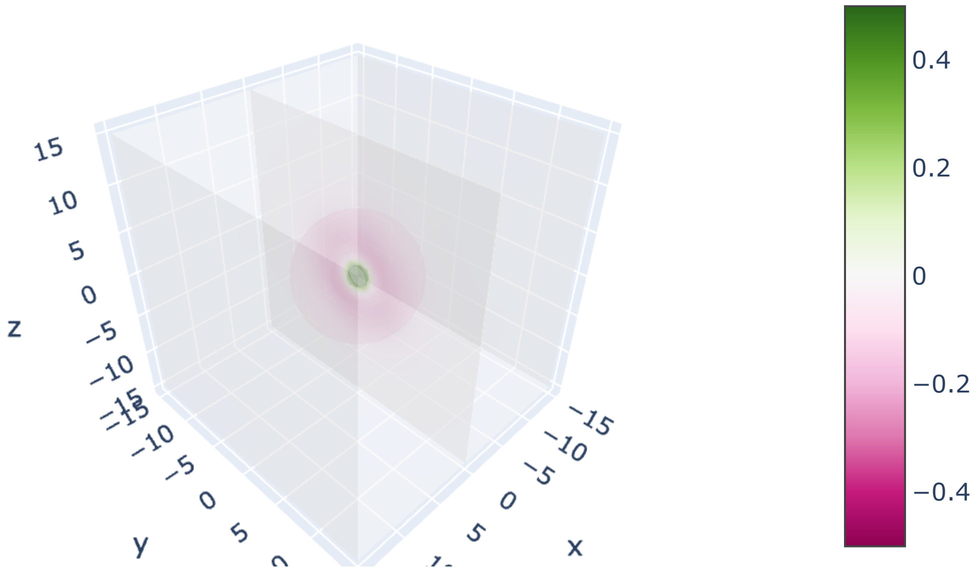
0

```
@bind lx Slider(0:nx-1; show_value=true)
```

0

```
@bind mx Slider(-lx:lx; default=0, show_value=true)
```

```
qn =  (n = 2, m = 0, l = 0)
```

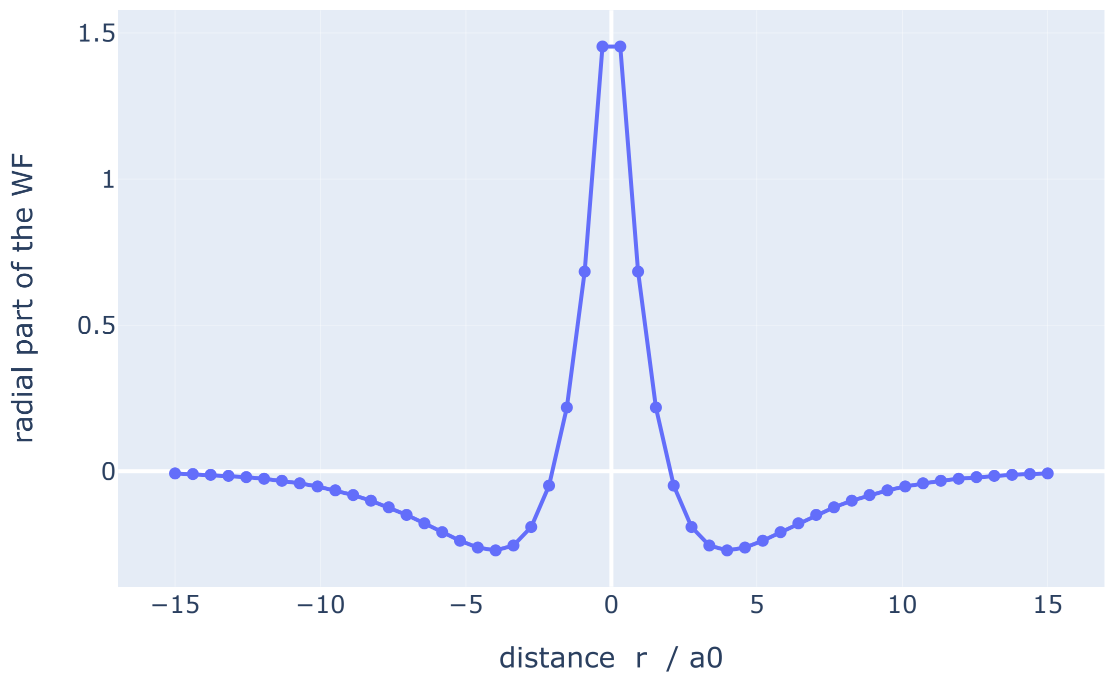


```
# this needs to be a capital P in Plot !
Plot(
    isosurface(

    x=X[:],  # unfold 3D arrays to one long 1D array
    y=Y[:],
    z=Z[:],
    value= values[:],

    isomin=-0.5,    # symmetric around 0
    isomax=0.5,

    opacity=0.3, # needs to be small to see through all surfaces
    surface_count=4, # even number
    slices_x=attr(show=true, locations=[0]), # slice along x=0 plane
),
    Layout(width=600, height=400) # adjust size in pixel
 )
```



```
# the radial part only
Plot(
    scatter(x=xs, y=radial.(abs.(xs); qn=qn), mode="markers+lines"),
    Layout(width=600, height=400,
        xaxis_title="distance  r  / a0", yaxis_title="radial part of the WF")
)
```