# ポテンシャルがある場合の1Dシュレーディンガー方程式の解

一次元系のシュレーディンガー方程式は、

$$\left(-\frac{\hbar^2}{2m}\frac{d^2}{dx^2}+V(x)\right)\psi(x)=\epsilon\psi(x)$$

である。 前回はポテンシャル$V(x)$がゼロの場合を解いたが、次はポテンシャルを入れて解いてみよう。

## 前回のおさらい

座標を離散化すると、シュレーディンガー方程式は、

$$\sum_j\left(-\frac{\hbar^2}{2m}d_j+V(x_j)\delta_{ij}\right)\psi(x_j)=\epsilon\psi(x_i)$$

と書くことができる。
ここで、

$$\frac{d^2}{dx^2}\psi|_{x=x_i}\rightarrow\sum_j d_j\psi(x_j)$$

である。
この方程式は、先ほどと同様に行列とベクトルで表現することができて、

$$\hat{H}\psi=\epsilon\psi$$

となる。これがシュレーディンガー方程式の行列表示である。
ここで、$\psi_n=U\mathbf{c}_n$を使えば、

$$\hat{H}U\mathbf{c}_n=\epsilon U\mathbf{c}_n$$
$$U^+\hat{H}U\mathbf{c}_n=\epsilon U^+U\mathbf{c}_n$$
$$\hat{H}^{'}\mathbf{c}_n=\epsilon\mathbf{c}_n$$

となる。
ベクトルはユニタリー変換で色々変換できるので、kでもxでも好きな表示で計算できる。

# 離散化座標表示での解

　実は前回のノートのコードで対応できる。

## ポテンシャルがない場合

ポテンシャルを入れる前に、波動関数がどのようになっているか見てみよう。 方程式は

$$\left(-\frac{d^2}{dx'^2}+V(x')\right)\psi(x')=\epsilon\psi(x')$$

と無次元化されている。 境界条件も前回と同じ

$$\psi(x_{-1})=0$$
$$\psi(x_{N+1})=0$$

である。

まず、ハミルトニアンを表す行列を以下のように用意する。

In [1]:

```
function make_H1d(N,a)
    mat_H = zeros(Float64,N,N)
    vec_V = zeros(Float64,N)

    for i in 1:N
        for dx in -1:1
            j = i + dx
            v = 0.0
            if dx == 0
                v = (2/a^2 + vec_V[i])
            elseif dx == 1
                v = -1/a^2
            elseif dx == -1
                v = -1/a^2
            end

            if 1 <= j <= N
                mat_H[i,j] = v
            end

        end

    end


    return mat_H
end
```

Out[1]:

make_H1d (generic function with 1 method)

最小固有値に属する波動関数を見てみよう。 まず、対角化し、

In [2]:

```
N = 1000
a = 0.01
mat_H = make_H1d(N,a)
ε,ψ = eig(mat_H)

println(ε[1])
```

0.09849886676979276

最小固有値を得た。次に固有ベクトルをプロットしよう。固有ベクトルの要素の二乗が波動関数の二乗である。

In [3]:

```
i = 1
println(ψ[:,i])
```

[0.000140286, 0.00028057, 0.000420851, 0.000561129, 0.0007014, 0.
000841665, 0.000981922, 0.00112217, 0.0012624, 0.00140263, 0.001
54284, 0.00168303, 0.00182321, 0.00196337, 0.00210351, 0.0022436
3, 0.00238373, 0.0025238, 0.00266385, 0.00280387, 0.00294387, 0.00
308384, 0.00322377, 0.00336368, 0.00350355, 0.00364338, 0.003783
19, 0.00392295, 0.00406267, 0.00420236, 0.004342, 0.0044816, 0.004
62116, 0.00476067, 0.00490014, 0.00503955, 0.00517892, 0.0053182
3, 0.0054575, 0.0055967, 0.00573586, 0.00587496, 0.006014, 0.00615
298, 0.0062919, 0.00643075, 0.00656955, 0.00670828, 0.00684694, 0.
00698554, 0.00712407, 0.00726252, 0.00740091, 0.00753922, 0.0076
7746, 0.00781562, 0.00795371, 0.00809172, 0.00822965, 0.00836749,
0.00850526, 0.00864294, 0.00878053, 0.00891804, 0.00905546, 0.009
1928, 0.00933004, 0.00946719, 0.00960424, 0.0097412, 0.00987807,
0.0100148, 0.0101515, 0.0102881, 0.0104245, 0.0105609, 0.0106972,
0.0108333, 0.0109694, 0.0111053, 0.0112412, 0.0113769, 0.0115125,
0.011648, 0.0117834, 0.0119186, 0.0120538, 0.0121888, 0.0123237,
0.0124585, 0.0125932, 0.0127277, 0.0128621, 0.0129964, 0.0131306,
0.0132646, 0.0133985, 0.0135323, 0.0136659, 0.0137994, 0.0139328,
0.014066, 0.0141991, 0.014332, 0.0144649, 0.0145975, 0.01473, 0.01
48624, 0.0149946, 0.0151267, 0.0152587, 0.0153904, 0.0155221, 0.01
56536, 0.0157849, 0.0159161, 0.0160471, 0.0161779, 0.0163086, 0.01
64392, 0.0165695, 0.0166997, 0.0168298, 0.0169597, 0.0170894, 0.01
72189, 0.0173483, 0.0174775, 0.0176065, 0.0177354, 0.0178641, 0.01
79926, 0.0181209, 0.0182491, 0.018377, 0.0185048, 0.0186324, 0.018
7598, 0.0188871, 0.0190141, 0.019141, 0.0192677, 0.0193942, 0.0195
205, 0.0196466, 0.0197725, 0.0198982, 0.0200237, 0.0201491, 0.0202
742, 0.0203991, 0.0205238, 0.0206483, 0.0207727, 0.0208968, 0.0210
207, 0.0211444, 0.0212679, 0.0213912, 0.0215142, 0.0216371, 0.0217
598, 0.0218822, 0.0220044, 0.0221264, 0.0222482, 0.0223698, 0.0224
911, 0.0226122, 0.0227331, 0.0228538, 0.0229742, 0.0230945, 0.0232
145, 0.0233342, 0.0234538, 0.0235731, 0.0236922, 0.023811, 0.02392
96, 0.024048, 0.0241661, 0.024284, 0.0244017, 0.0245191, 0.024636
3, 0.0247532, 0.0248699, 0.0249863, 0.0251025, 0.0252185, 0.025334
2, 0.0254496, 0.0255648, 0.0256798, 0.0257945, 0.0259089, 0.026023
1, 0.026137, 0.0262507, 0.0263641, 0.0264773, 0.0265902, 0.026702
8, 0.0268152, 0.0269273, 0.0270391, 0.0271507, 0.027262, 0.027373
1, 0.0274838, 0.0275943, 0.0277046, 0.0278145, 0.0279242, 0.028033
6, 0.0281427, 0.0282516, 0.0283601, 0.0284684, 0.0285765, 0.028684
2, 0.0287916, 0.0288988, 0.0290057, 0.0291123, 0.0292186, 0.029324
6, 0.0294303, 0.0295358, 0.0296409, 0.0297458, 0.0298504, 0.029954
6, 0.0300586, 0.0301623, 0.0302657, 0.0303688, 0.0304715, 0.03057
4, 0.0306762, 0.0307781, 0.0308797, 0.030981, 0.0310819, 0.031182
6, 0.0312829, 0.031383, 0.0314827, 0.0315822, 0.0316813, 0.031780
1, 0.0318786, 0.0319768, 0.0320746, 0.0321722, 0.0322694, 0.032366
3, 0.0324629, 0.0325592, 0.0326552, 0.0327508, 0.0328461, 0.032941
1, 0.0330358, 0.0331301, 0.0332241, 0.0333178, 0.0334111, 0.033504
2, 0.0335969, 0.0336892, 0.0337813, 0.033873, 0.0339643, 0.034055
4, 0.0341461, 0.0342364, 0.0343264, 0.0344161, 0.0345055, 0.034594
5, 0.0346832, 0.0347715, 0.0348595, 0.0349471, 0.0350344, 0.035121
3, 0.0352079, 0.0352942, 0.0353801, 0.0354657, 0.0355509, 0.035635
7, 0.0357203, 0.0358044, 0.0358882, 0.0359717, 0.0360548, 0.036137
5, 0.0362199, 0.0363019, 0.0363836, 0.0364649, 0.0365459, 0.036626
5, 0.0367067, 0.0367866, 0.0368661, 0.0369452, 0.037024, 0.037102
4, 0.0371805, 0.0372582, 0.0373355, 0.0374124, 0.037489, 0.037565
2, 0.0376411, 0.0377166, 0.0377917, 0.0378664, 0.0379407, 0.038014
7, 0.0380883, 0.0381616, 0.0382344, 0.0383069, 0.038379, 0.038450
7, 0.0385221, 0.038593, 0.0386636, 0.0387338, 0.0388037, 0.038873

1, 0.0389422, 0.0390108, 0.0390791, 0.039147, 0.0392146, 0.039281
7, 0.0393485, 0.0394148, 0.0394808, 0.0395464, 0.0396116, 0.039676
4, 0.0397408, 0.0398048, 0.0398684, 0.0399317, 0.0399945, 0.04005
7, 0.040119, 0.0401807, 0.0402419, 0.0403028, 0.0403633, 0.040423
3, 0.040483, 0.0405423, 0.0406012, 0.0406597, 0.0407177, 0.040775
4, 0.0408327, 0.0408896, 0.040946, 0.0410021, 0.0410577, 0.041113,
0.0411679, 0.0412223, 0.0412763, 0.04133, 0.0413832, 0.041436, 0.0
414884, 0.0415404, 0.041592, 0.0416432, 0.041694, 0.0417444, 0.041
7943, 0.0418438, 0.041893, 0.0419417, 0.04199, 0.0420379, 0.042085
4, 0.0421324, 0.0421791, 0.0422253, 0.0422711, 0.0423165, 0.042361
5, 0.042406, 0.0424502, 0.0424939, 0.0425372, 0.0425801, 0.042622
6, 0.0426646, 0.0427063, 0.0427475, 0.0427883, 0.0428286, 0.042868
6, 0.0429081, 0.0429472, 0.0429859, 0.0430241, 0.043062, 0.043099
4, 0.0431363, 0.0431729, 0.043209, 0.0432447, 0.04328, 0.0433149, 0.
0433493, 0.0433833, 0.0434169, 0.04345, 0.0434827, 0.043515, 0.043
5469, 0.0435783, 0.0436093, 0.0436399, 0.04367, 0.0436997, 0.04372
9, 0.0437579, 0.0437863, 0.0438143, 0.0438418, 0.043869, 0.043895
7, 0.0439219, 0.0439478, 0.0439731, 0.0439981, 0.0440226, 0.044046
7, 0.0440704, 0.0440936, 0.0441164, 0.0441388, 0.0441607, 0.044182
2, 0.0442032, 0.0442239, 0.044244, 0.0442638, 0.0442831, 0.044302,
0.0443204, 0.0443384, 0.044356, 0.0443731, 0.0443898, 0.0444061,
0.0444219, 0.0444372, 0.0444522, 0.0444667, 0.0444808, 0.0444944,
0.0445076, 0.0445203, 0.0445326, 0.0445445, 0.0445559, 0.0445669,
0.0445775, 0.0445876, 0.0445973, 0.0446065, 0.0446153, 0.0446237,
0.0446316, 0.0446391, 0.0446461, 0.0446527, 0.0446589, 0.0446646,
0.0446699, 0.0446747, 0.0446791, 0.0446831, 0.0446866, 0.0446897,
0.0446924, 0.0446946, 0.0446963, 0.0446976, 0.0446985, 0.044699,
0.044699, 0.0446985, 0.0446976, 0.0446963, 0.0446946, 0.0446924,
0.0446897, 0.0446866, 0.0446831, 0.0446791, 0.0446747, 0.0446699,
0.0446646, 0.0446589, 0.0446527, 0.0446461, 0.0446391, 0.0446316,
0.0446237, 0.0446153, 0.0446065, 0.0445973, 0.0445876, 0.0445775,
0.0445669, 0.0445559, 0.0445445, 0.0445326, 0.0445203, 0.0445076,
0.0444944, 0.0444808, 0.0444667, 0.0444522, 0.0444372, 0.0444219,
0.0444061, 0.0443898, 0.0443731, 0.044356, 0.0443384, 0.0443204,
0.044302, 0.0442831, 0.0442638, 0.044244, 0.0442239, 0.0442032, 0.
0441822, 0.0441607, 0.0441388, 0.0441164, 0.0440936, 0.0440704, 0.
0440467, 0.0440226, 0.0439981, 0.0439731, 0.0439478, 0.0439219, 0.
0438957, 0.043869, 0.0438418, 0.0438143, 0.0437863, 0.0437579, 0.0
43729, 0.0436997, 0.04367, 0.0436399, 0.0436093, 0.0435783, 0.0435
469, 0.043515, 0.0434827, 0.04345, 0.0434169, 0.0433833, 0.043349
3, 0.0433149, 0.04328, 0.0432447, 0.043209, 0.0431729, 0.0431363, 0.
0430994, 0.043062, 0.0430241, 0.0429859, 0.0429472, 0.0429081, 0.0
428686, 0.0428286, 0.0427883, 0.0427475, 0.0427063, 0.0426646, 0.0
426226, 0.0425801, 0.0425372, 0.0424939, 0.0424502, 0.042406, 0.04
23615, 0.0423165, 0.0422711, 0.0422253, 0.0421791, 0.0421324, 0.04
20854, 0.0420379, 0.04199, 0.0419417, 0.041893, 0.0418438, 0.04179
43, 0.0417444, 0.041694, 0.0416432, 0.041592, 0.0415404, 0.041488
4, 0.041436, 0.0413832, 0.04133, 0.0412763, 0.0412223, 0.0411679, 0.
041113, 0.0410577, 0.0410021, 0.040946, 0.0408896, 0.0408327, 0.04
07754, 0.0407177, 0.0406597, 0.0406012, 0.0405423, 0.040483, 0.040
4233, 0.0403633, 0.0403028, 0.0402419, 0.0401807, 0.040119, 0.0400
57, 0.0399945, 0.0399317, 0.0398684, 0.0398048, 0.0397408, 0.03967
64, 0.0396116, 0.0395464, 0.0394808, 0.0394148, 0.0393485, 0.03928
17, 0.0392146, 0.039147, 0.0390791, 0.0390108, 0.0389422, 0.038873
1, 0.0388037, 0.0387338, 0.0386636, 0.038593, 0.0385221, 0.038450
7, 0.038379, 0.0383069, 0.0382344, 0.0381616, 0.0380883, 0.038014
7, 0.0379407, 0.0378664, 0.0377917, 0.0377166, 0.0376411, 0.037565

2, 0.037489, 0.0374124, 0.0373355, 0.0372582, 0.0371805, 0.037102
4, 0.037024, 0.0369452, 0.0368661, 0.0367866, 0.0367067, 0.036626
5, 0.0365459, 0.0364649, 0.0363836, 0.0363019, 0.0362199, 0.036137
5, 0.0360548, 0.0359717, 0.0358882, 0.0358044, 0.0357203, 0.035635
7, 0.0355509, 0.0354657, 0.0353801, 0.0352942, 0.0352079, 0.035121
3, 0.0350344, 0.0349471, 0.0348595, 0.0347715, 0.0346832, 0.034594
5, 0.0345055, 0.0344161, 0.0343264, 0.0342364, 0.0341461, 0.034055
4, 0.0339643, 0.033873, 0.0337813, 0.0336892, 0.0335969, 0.033504
2, 0.0334111, 0.0333178, 0.0332241, 0.0331301, 0.0330358, 0.032941
1, 0.0328461, 0.0327508, 0.0326552, 0.0325592, 0.0324629, 0.032366
3, 0.0322694, 0.0321722, 0.0320746, 0.0319768, 0.0318786, 0.031780
1, 0.0316813, 0.0315822, 0.0314827, 0.031383, 0.0312829, 0.031182
6, 0.0310819, 0.030981, 0.0308797, 0.0307781, 0.0306762, 0.030574,
0.0304715, 0.0303688, 0.0302657, 0.0301623, 0.0300586, 0.0299546,
0.0298504, 0.0297458, 0.0296409, 0.0295358, 0.0294303, 0.0293246,
0.0292186, 0.0291123, 0.0290057, 0.0288988, 0.0287916, 0.0286842,
0.0285765, 0.0284684, 0.0283601, 0.0282516, 0.0281427, 0.0280336,
0.0279242, 0.0278145, 0.0277046, 0.0275943, 0.0274838, 0.0273731,
0.027262, 0.0271507, 0.0270391, 0.0269273, 0.0268152, 0.0267028,
0.0265902, 0.0264773, 0.0263641, 0.0262507, 0.026137, 0.0260231,
0.0259089, 0.0257945, 0.0256798, 0.0255648, 0.0254496, 0.0253342,
0.0252185, 0.0251025, 0.0249863, 0.0248699, 0.0247532, 0.0246363,
0.0245191, 0.0244017, 0.024284, 0.0241661, 0.024048, 0.0239296, 0.
023811, 0.0236922, 0.0235731, 0.0234538, 0.0233342, 0.0232145, 0.0
230945, 0.0229742, 0.0228538, 0.0227331, 0.0226122, 0.0224911, 0.0
223698, 0.0222482, 0.0221264, 0.0220044, 0.0218822, 0.0217598, 0.0
216371, 0.0215142, 0.0213912, 0.0212679, 0.0211444, 0.0210207, 0.0
208968, 0.0207727, 0.0206483, 0.0205238, 0.0203991, 0.0202742, 0.0
201491, 0.0200237, 0.0198982, 0.0197725, 0.0196466, 0.0195205, 0.0
193942, 0.0192677, 0.019141, 0.0190141, 0.0188871, 0.0187598, 0.01
86324, 0.0185048, 0.018377, 0.0182491, 0.0181209, 0.0179926, 0.017
8641, 0.0177354, 0.0176065, 0.0174775, 0.0173483, 0.0172189, 0.017
0894, 0.0169597, 0.0168298, 0.0166997, 0.0165695, 0.0164392, 0.016
3086, 0.0161779, 0.0160471, 0.0159161, 0.0157849, 0.0156536, 0.015
5221, 0.0153904, 0.0152587, 0.0151267, 0.0149946, 0.0148624, 0.014
73, 0.0145975, 0.0144649, 0.014332, 0.0141991, 0.014066, 0.013932
8, 0.0137994, 0.0136659, 0.0135323, 0.0133985, 0.0132646, 0.013130
6, 0.0129964, 0.0128621, 0.0127277, 0.0125932, 0.0124585, 0.012323
7, 0.0121888, 0.0120538, 0.0119186, 0.0117834, 0.011648, 0.011512
5, 0.0113769, 0.0112412, 0.0111053, 0.0109694, 0.0108333, 0.010697
2, 0.0105609, 0.0104245, 0.0102881, 0.0101515, 0.0100148, 0.009878
07, 0.0097412, 0.00960424, 0.00946719, 0.00933004, 0.0091928, 0.00
905546, 0.00891804, 0.00878053, 0.00864294, 0.00850526, 0.008367
49, 0.00822965, 0.00809172, 0.00795371, 0.00781562, 0.00767746, 0.
00753922, 0.00740091, 0.00726252, 0.00712407, 0.00698554, 0.0068
4694, 0.00670828, 0.00656955, 0.00643075, 0.0062919, 0.00615298,
0.006014, 0.00587496, 0.00573586, 0.0055967, 0.0054575, 0.0053182
3, 0.00517892, 0.00503955, 0.00490014, 0.00476067, 0.00462116, 0.0
044816, 0.004342, 0.00420236, 0.00406267, 0.00392295, 0.00378319,
0.00364338, 0.00350355, 0.00336368, 0.00322377, 0.00308384, 0.002
94387, 0.00280387, 0.00266385, 0.0025238, 0.00238373, 0.00224363,
0.00210351, 0.00196337, 0.00182321, 0.00168303, 0.00154284, 0.001
40263, 0.0012624, 0.00112217, 0.000981922, 0.000841665, 0.000701
4, 0.000561129, 0.000420851, 0.00028057, 0.000140286]

In [4]:

```
using Plots
gr()
```

Out[4]:

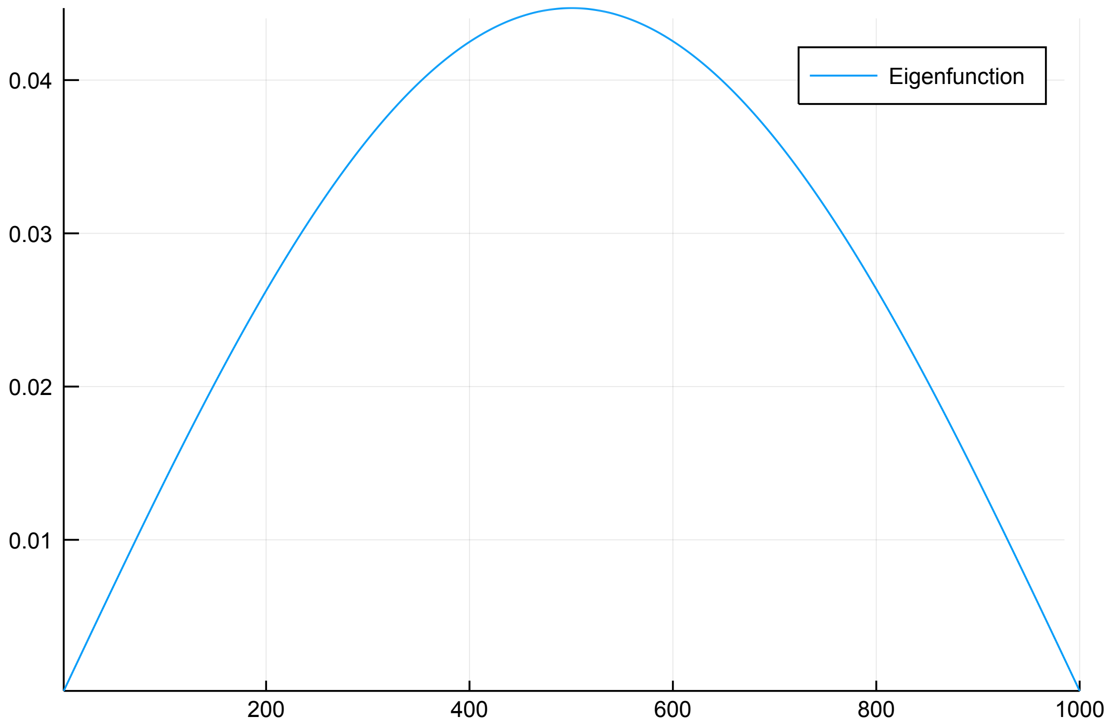
Plots.GRBackend()

In [5]:

```
xaxis = Int64[]
for i in 1:N
    push!(xaxis,i)
end

i = 1
plot(xaxis[1:N],ψ[1:N,i],label="Eigenfunction")
```

Out[5]:




見てわかるように、前回解いた時に得られた

$$\psi(x) = 2iC_1 \sin \frac{\pi}{L} x$$

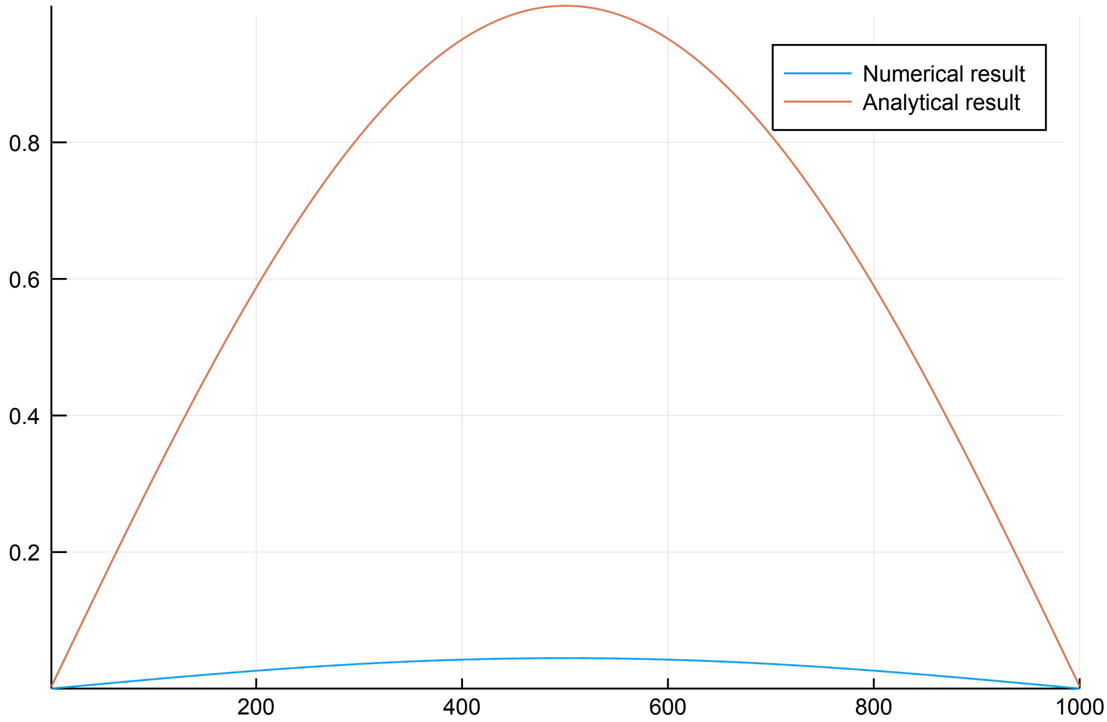
と等しい。ここで、$L = (N+1) * a$である（境界条件をみよ）。プロットすると、

In [6]:

```
aψ = zeros(Float64,N)
for i in 1:N
    xi = i*a
    aψ[i] = sin(xi*π/((N+1)*a))
end

i = 1
plot(xaxis[1:N],[ψ[1:N,i],aψ[1:N]],label=["Numerical result","Analytical result"])
```

Out[6]:




あれ、値がずれてしまった。これは、シュレーディンガー方程式の解は定数倍も解であるためである。
数値的に得られた解は

In [7]:

```
sum(dot(ψ[1:N,i],ψ[1:N,i]))
```

Out[7]:

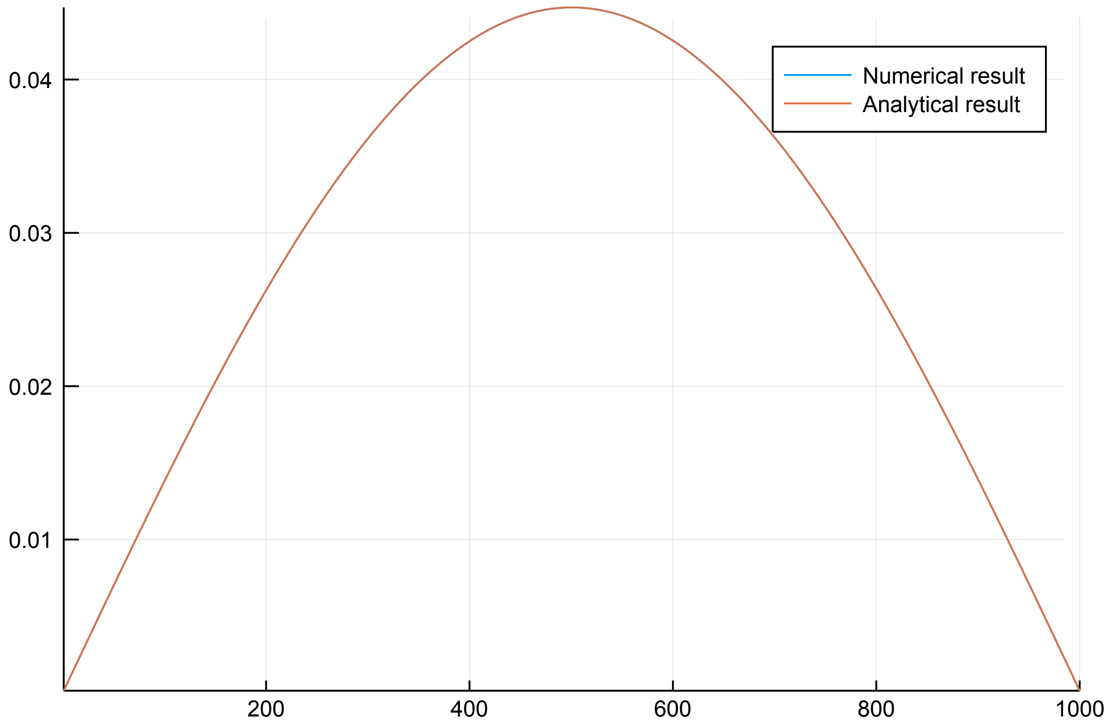
1.0000000000000004

内積をとると1に規格化されている。この内積は全空間での粒子を見出す確率であるので、まさに規格化条件となっている。
解析解の方も規格化を行うと、

In [8]:

```
aψ = zeros(Float64,N)
for i in 1:N
    xi = i*a
    aψ[i] = sin(xi*π/((N+1)*a))
end
C = sum(dot(aψ[1:N],aψ[1:N]))
aψ = aψ/sqrt(C)
i = 1
plot(xaxis[1:N],[ψ[1:N,i],aψ[1:N]],label=["Numerical result","Analytical result"])
```

Out[8]:



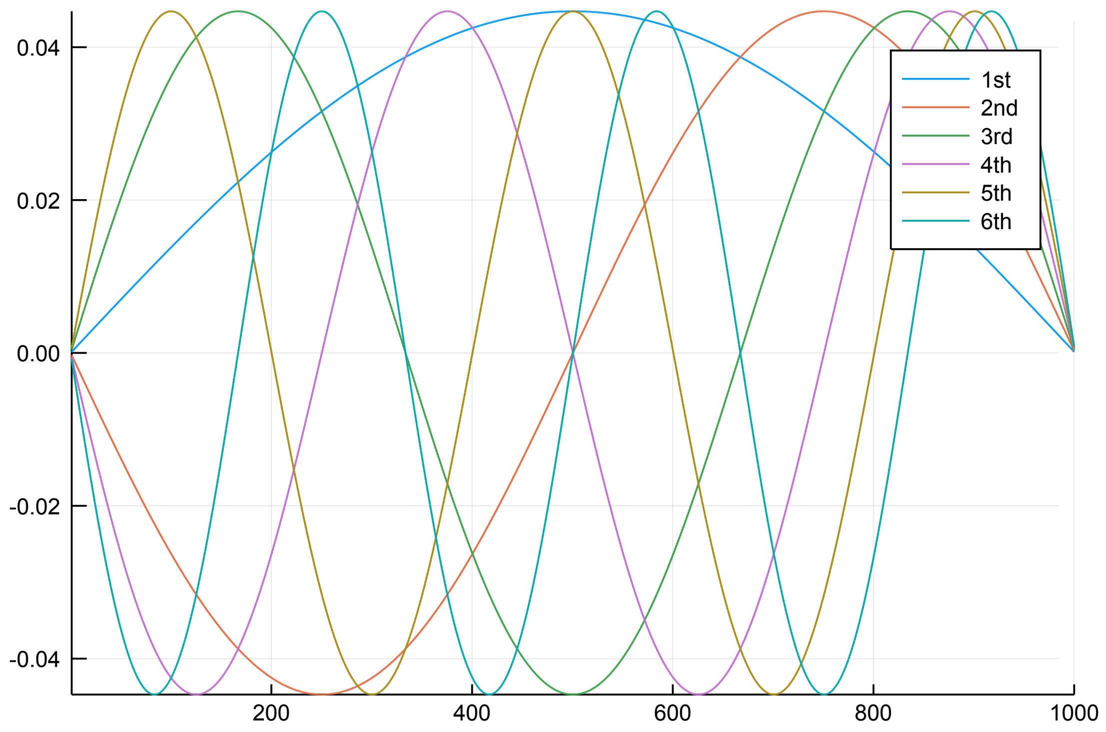

となり、完全に一致する。

ポテンシャルがない場合の解を、複数描くと、

In [9]:

```
plot(xaxis[1:N],ψ[1:N,1:6],label=["1st","2nd","3rd","4th","5th","6th"])
```

Out[9]:




どれもsin関数である。

## ポテンシャルがある場合

ポテンシャルがある場合を考えよう。まず、ポテンシャルを定義する。

In [15]:

```
function calc_V(N,V0)
    vec_V = zeros(Float64,N)
    dx = N/6
    for i in 1:N
        if N/2 - dx <= i <= N/2 + dx
            vec_V[i] = V0
        end
    end
    return vec_V
end
```

Out[15]:

calc_V (generic function with 1 method)

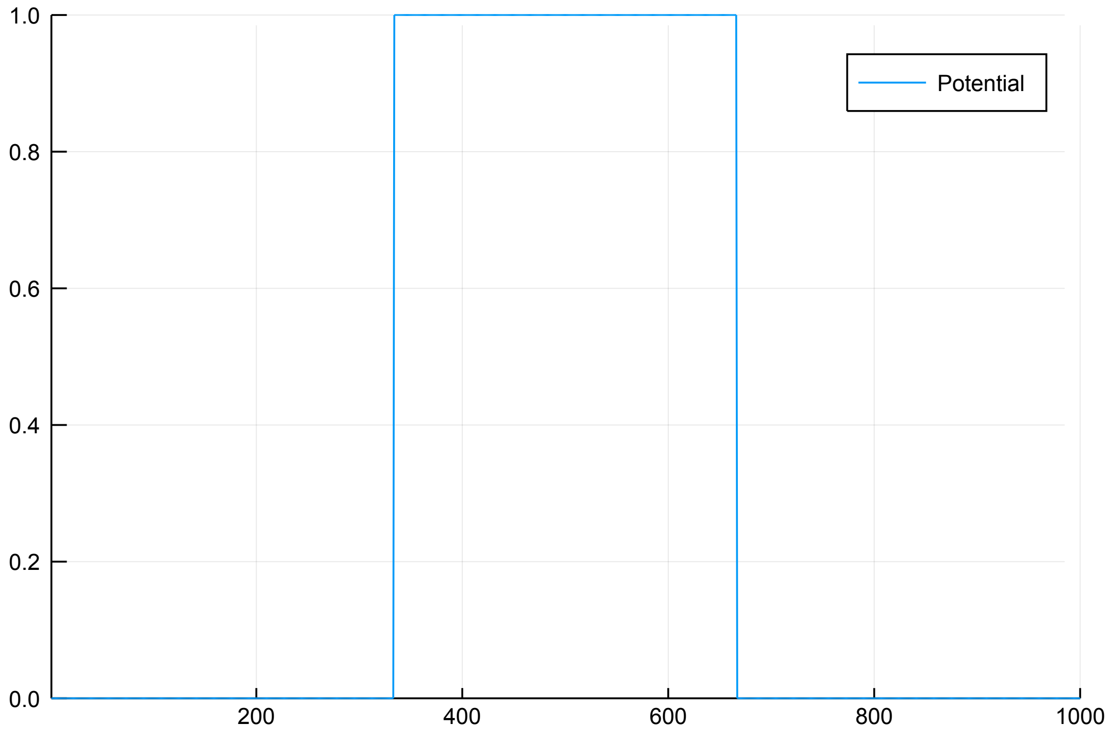
ここでは、$N/2 - dx \leq i \leq N/2 + dx$の範囲にポテンシャル$V0$があることにした。

In [16]:

```
xaxis = Int64[]
for i in 1:N
    push!(xaxis,i)
end

V0=1.0
vec_V = calc_V(N,V0)
plot(xaxis[1:N],vec_V[1:N],label="Potential")
```

Out[16]:




このポテンシャルがある場合の最小固有値とその固有関数を見てみよう。
そのために、ハミルトニアンを定義する。

In [17]:

```
function make_H1dv(N,a,V0)
    mat_H = zeros(Float64,N,N)
    vec_V = calc_V(N,V0)

    for i in 1:N
        for dx in -1:1
            j = i + dx
            v = 0.0
            if dx == 0
                v = (2/a^2 + vec_V[i])
            elseif dx == 1
                v = -1/a^2
            elseif dx == -1
                v = -1/a^2
            end

            if 1 <= j <= N
                mat_H[i,j] = v
            end

        end

    end


    return mat_H
end
```

Out[17]:

make_H1dv (generic function with 1 method)

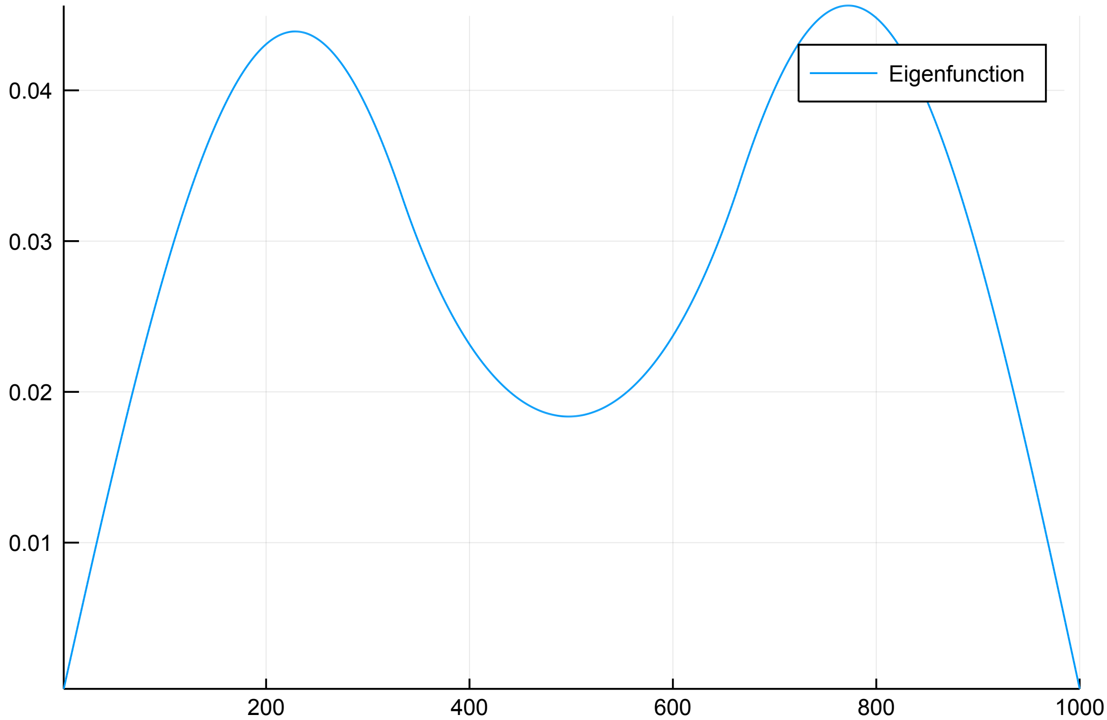
In [19]:

```
N = 1000
a = 0.01
V0 = 1.0
mat_H = make_H1dv(N,a,V0)
ε,ψ = eig(mat_H)

println(ε[1])
i = 1
plot(xaxis[1:N],ψ[1:N,i],label="Eigenfunction")
```

0.47212647457743806

Out[19]:




ポテンシャルによって、波動関数が凹んでいることがわかる。当然、その二乗である存在確率も低くなっている。
次に、ポテンシャルの強度を変えてみよう。

In [25]:

```
N = 1000
a = 0.01
groundstates = []
labels = []
for v in 1:10
    V0 = v*0.5
    mat_H = make_H1dv(N,a,V0)
    ε,ψ = eig(mat_H)
    println("Potential = ",V0," Minimum eigenvalue = ",ε[1])
    push!(groundstates,ψ[:,1])
    push!(labels,string(V0))
end
```

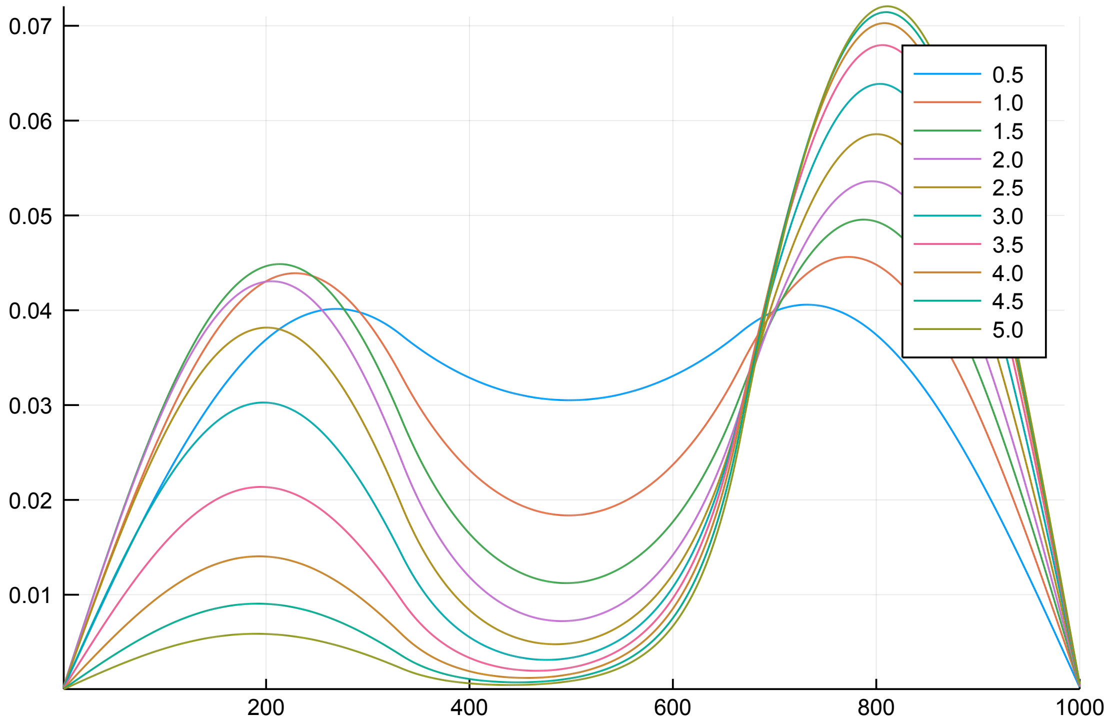
```
Potential = 0.5 Minimum eigenvalue = 0.3406271439458158
Potential = 1.0 Minimum eigenvalue = 0.47212647457743806
Potential = 1.5 Minimum eigenvalue = 0.542094653793089
Potential = 2.0 Minimum eigenvalue = 0.5841549613399831
Potential = 2.5 Minimum eigenvalue = 0.6125251020397613
Potential = 3.0 Minimum eigenvalue = 0.6332641494769529
Potential = 3.5 Minimum eigenvalue = 0.6493089489080786
Potential = 4.0 Minimum eigenvalue = 0.662273059034554
Potential = 4.5 Minimum eigenvalue = 0.6730976630913431
Potential = 5.0 Minimum eigenvalue = 0.682351937176906
```

対角化できたので、プロットしてみる。

In [26]:

```
plot(xaxis[1:N],groundstates,label=labels)
```

Out[26]:




本来、左右対称に出るべきもののような気がするが、なぜ非対称になってしまったのだろうか？
ポテンシャルを少し変えてみる。

In [27]:

```
function calc_V(N,V0)
    vec_V = zeros(Float64,N)
    dx = N/6
    center = (N+1)/2
    for i in 1:N
        if center - dx <= i <= center + dx
            vec_V[i] = V0
        end
    end
    return vec_V
end
```

Out[27]:

```
calc_V (generic function with 1 method)
```

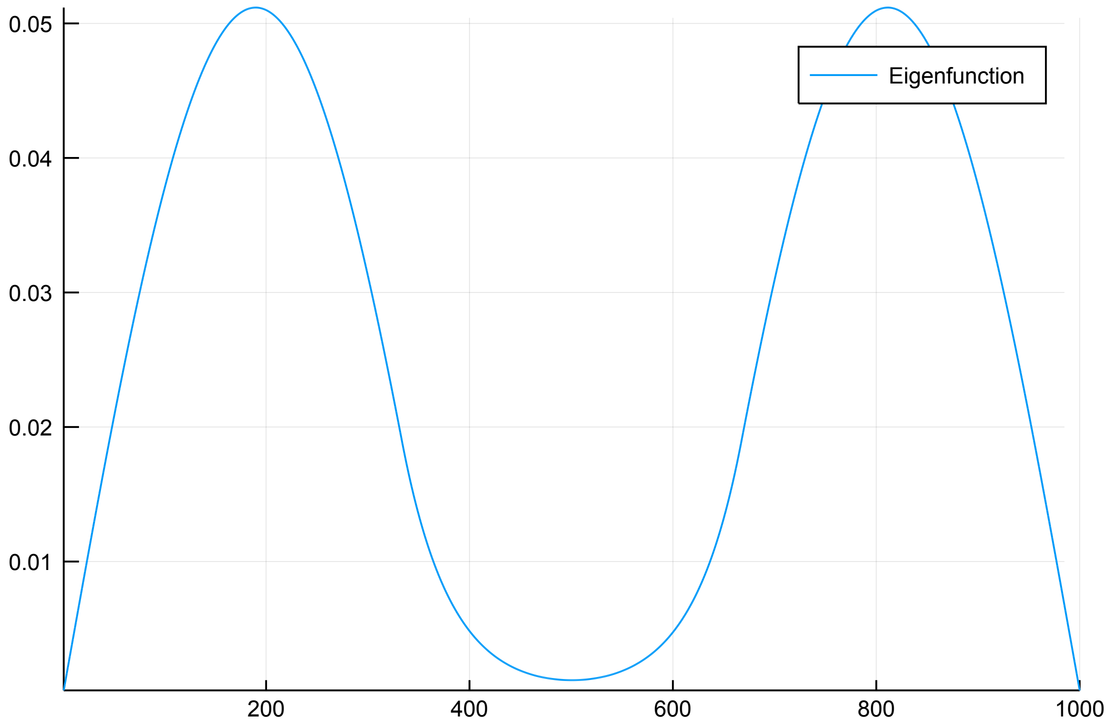
In [28]:

```
N = 1000
a = 0.01
V0 = 5.0
mat_H = make_H1dv(N,a,V0)
ε,ψ = eig(mat_H)

println(ε[1])
i = 1
plot(xaxis[1:N],ψ[1:N,i],label="Eigenfunction")
```

0.6856667068903135

Out[28]:




ほとんど左右対称になった。しかし、このくらいの微小なちがいでなぜこんなに変化してしまったのだろうか。
その謎は、$a$の大きさにある。ポテンシャルをもとの定義にして、$a$の大きさを半分にしてみよう。

In [32]:

```
function calc_V(N,V0)
    vec_V = zeros(Float64,N)
    dx = N/6
    center = (N)/2
    for i in 1:N
        if center - dx <= i <= center + dx
            vec_V[i] = V0
        end
    end
    return vec_V
end
```

Out[32]:

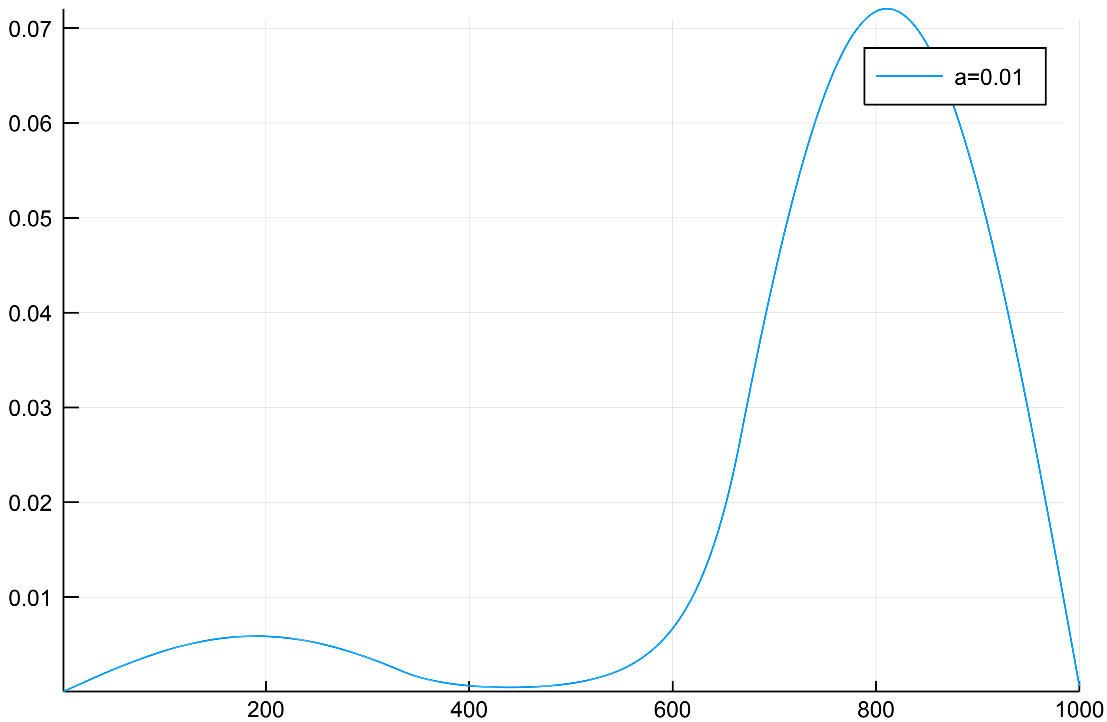
calc_V (generic function with 1 method)

In [44]:

```
N = 1000
a = 0.01
V0 = 5.0
mat_H = make_H1dv(N,a,V0)
ε,ψ = eig(mat_H)
println(ε[1])
plot(xaxis[1:N],ψ[1:N,1],label="a=0.01")
```

0.682351937176906

Out[44]:



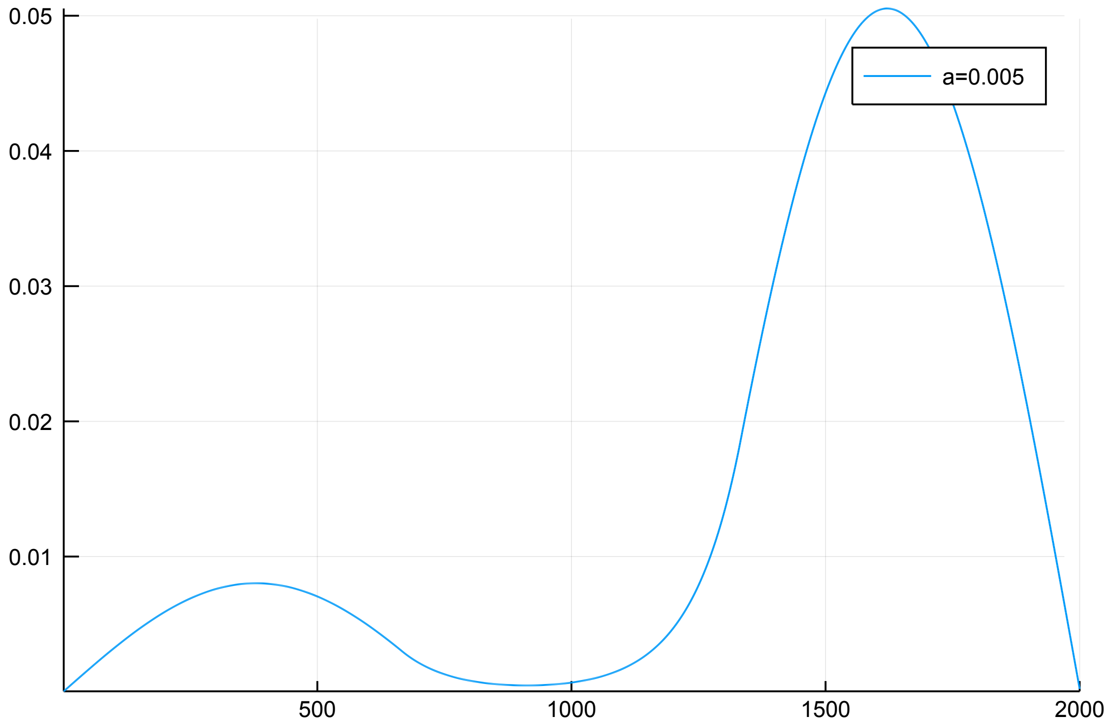
In [46]:

```
N = 1000*2
xaxis = Int64[]
for i in 1:N
    push!(xaxis,i)
end


a = 0.01/2
V0 = 5.0
mat_H = make_H1dv(N,a,V0)
println(ε[1])
ε,ψ = eig(mat_H)
plot(xaxis[1:N],ψ[1:N,1],label="a=0.005")
```

0.6850219113122353

Out[46]:




みてわかるように、$a$の大きさを小さくすると左右の均衡が少し改善された。
つまり、数値計算上のエラーであることがわかる。$a$が小さくなればさらに改善されるだろう。 矩形ポテンシャルをきちんと中心においた場合$center = (L+1)/2$であれば、改善される。 このような違いは、矩形のポテンシャルに不連続な変化があるからである。いま、差分化している微分演算子の精度が悪いのである。
これを確認するために、ポテンシャルの形状をなだらかにしてみよう。

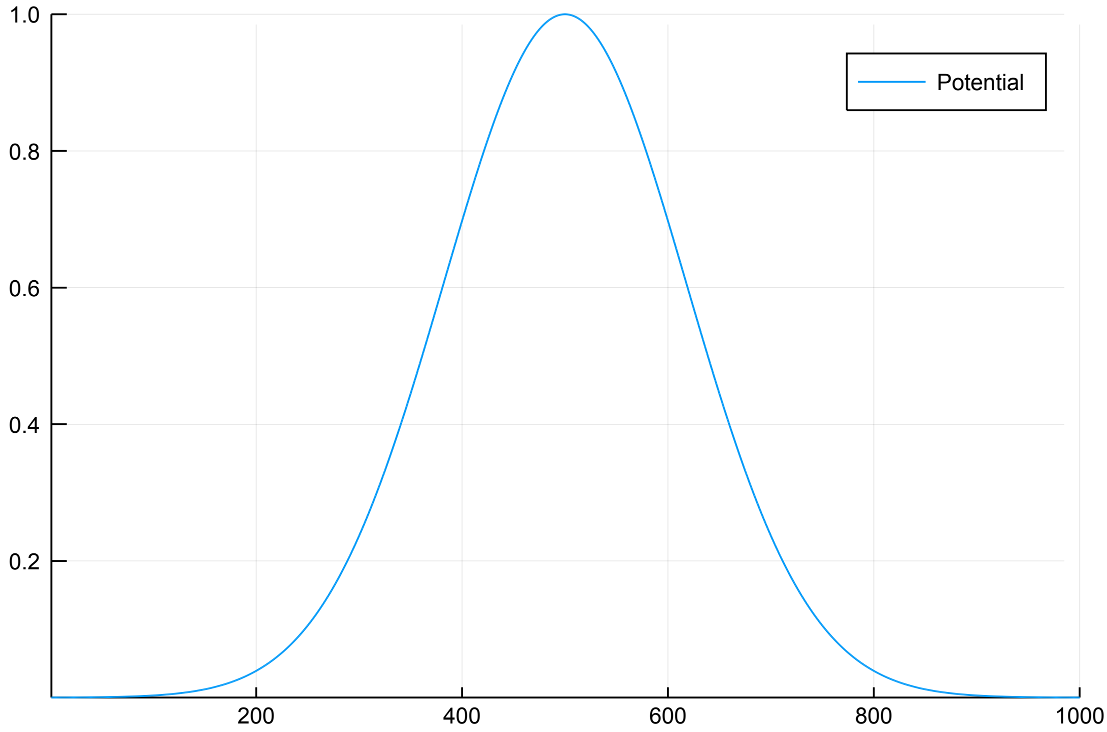
In [50]:

```
function calc_V(N,V0)
    vec_V = zeros(Float64,N)
    dx = N/6
    center = (N)/2
    for i in 1:N
        vec_V[i] = V0*exp(-(i-center)^2/(dx^2))
    end
    return vec_V
end

N = 1000
xaxis = Int64[]
for i in 1:N
    push!(xaxis,i)
end

V0=1.0
vec_V = calc_V(N,V0)
plot(xaxis[1:N],vec_V[1:N],label="Potential")
```

Out[50]:




ポテンシャルを、上のようにガウス関数にした。このとき、最小固有値の固有関数は

In [51]:

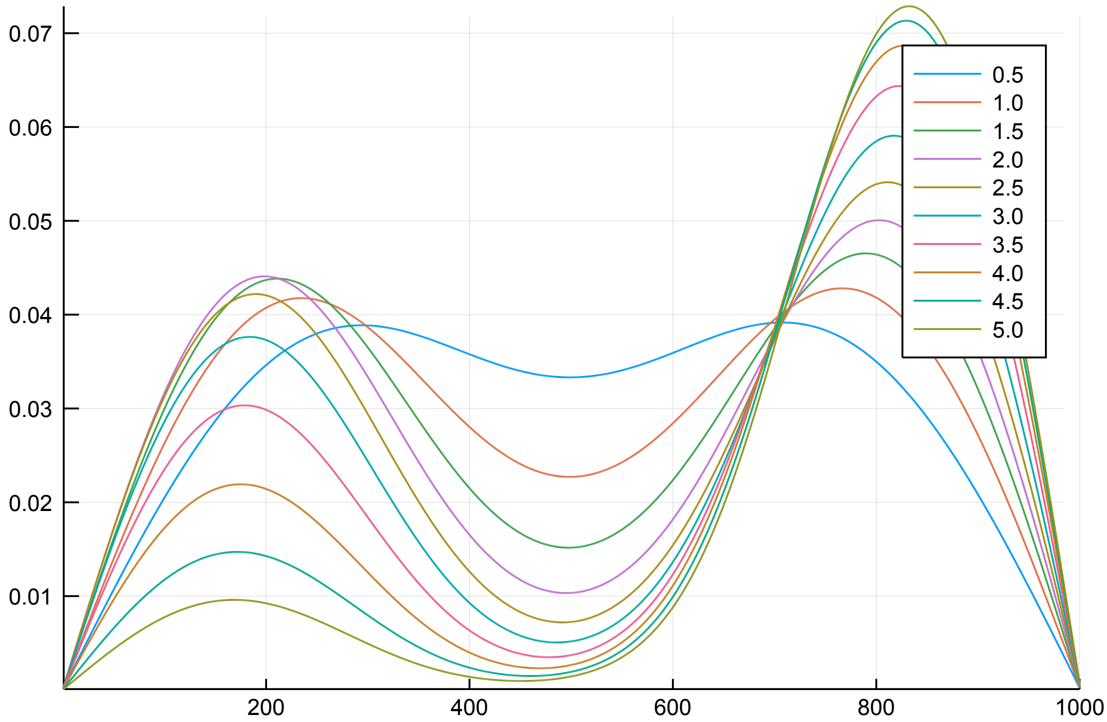
```
N = 1000
a = 0.01
groundstates = []
labels = []
for v in 1:10
    V0 = v*0.5
    mat_H = make_H1dv(N,a,V0)
    ε,ψ = eig(mat_H)
    println("Potential = ",V0," Minimum eigenvalue = ",ε[1])
    push!(groundstates,ψ[:,1])
    push!(labels,string(V0))
end
```

```
Potential = 0.5 Minimum eigenvalue = 0.3216661503241833
Potential = 1.0 Minimum eigenvalue = 0.47496666834409224
Potential = 1.5 Minimum eigenvalue = 0.5787666683372826
Potential = 2.0 Minimum eigenvalue = 0.6538526217544764
Potential = 2.5 Minimum eigenvalue = 0.7124623403946451
Potential = 3.0 Minimum eigenvalue = 0.7609213437478689
Potential = 3.5 Minimum eigenvalue = 0.8025976518969778
Potential = 4.0 Minimum eigenvalue = 0.83945543554293
Potential = 4.5 Minimum eigenvalue = 0.8727520445958912
Potential = 5.0 Minimum eigenvalue = 0.9033145515618101
```

In [52]:

```
plot(xaxis[1:N],groundstates,label=labels)
```

Out[52]:




先ほどよりもましになったが、相変わらず$V0 = 5$では、左右対称が崩れてしまっている。これは、ポテンシャルの変化が急激すぎることに起因している。最後に、ポテンシャルの中心を厳密に系の中心とおくと

In [20]:

```
function calc_V(N,V0)
    vec_V = zeros(Float64,N)
    dx = N/6
    center = (N+1)/2
    for i in 1:N
        vec_V[i] = V0*exp(-(i-center)^2/(dx^2))
    end
    return vec_V
end
```

Out[20]:

calc_V (generic function with 1 method)

In [21]:

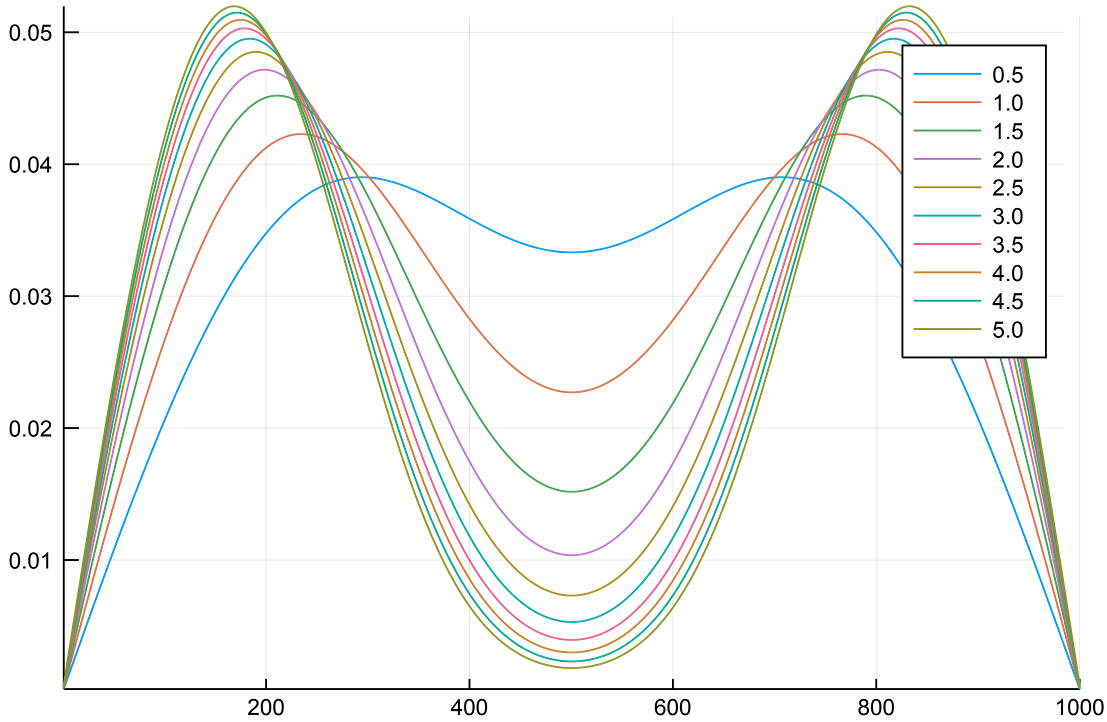
```
N = 1000
a = 0.01
groundstates = []
labels = []
for v in 1:10
    V0 = v*0.5
    mat_H = make_H1dv(N,a,V0)
    ε,ψ = eig(mat_H)
    println("Potential = ",V0," Minimum eigenvalue = ",ε[1])
    push!(groundstates,ψ[:,1])
    push!(labels,string(V0))
end
```

```
Potential = 0.5 Minimum eigenvalue = 0.3216680437594529
Potential = 1.0 Minimum eigenvalue = 0.47497702625749055
Potential = 1.5 Minimum eigenvalue = 0.5788014430373154
Potential = 2.0 Minimum eigenvalue = 0.6539436690715744
Potential = 2.5 Minimum eigenvalue = 0.7126667576418749
Potential = 3.0 Minimum eigenvalue = 0.7613285627430311
Potential = 3.5 Minimum eigenvalue = 0.8033158575112843
Potential = 4.0 Minimum eigenvalue = 0.8405613523837149
Potential = 4.5 Minimum eigenvalue = 0.8742498997697548
Potential = 5.0 Minimum eigenvalue = 0.905155534659135
```

In [22]:

```
plot(xaxis[1:N],groundstates,label=labels)
```

Out[22]:




きれいに左右対称になった。このことから、厳密に左右対称なものが欲しい場合には、数値計算でもきちんと左右対称にすべきであることがわかる。