See http://winvector.github.io/freq/minimax.pdf for background and details

From: http://mathoverflow.net/questions/177574/existence-of-solutions-of-a-polynomial-system

Fix $k \in \mathbb{N}, k \geq 1$. Let $p \in [0, 1]$ and $x = (x_0, \ldots, x_k)$ be a $(k+1)$-dimensional *real* vector, and define

$$S_k(p, x) = -x_0^2 + \sum_{i=0}^{k} \binom{k}{i} p^i (1-p)^{k-i} \cdot (x_i - p)^2.$$

Experiments show that for small values of k

$$\exists x \in \text{ interior of } [0, 1]^{k+1} . \ \forall p \in [0, 1] . \ S_k(p, x) = 0.$$

In other words, there are x_i's such that $S_k(x, p)$ is identically zero as a polynomial in p.
For a given k we can expand $S_k(x, p)$ as a polynomial in p and equate the coefficients to 0. For $k = 2$ we get

and this has two real solutions:

$$x = (\frac{1}{2}(-1 - \sqrt{2}), \frac{1}{2}, \frac{1}{2}(3 + \sqrt{2}))$$

and

$$x = (\frac{1}{2}(-1 + \sqrt{2}), \frac{1}{2}, \frac{1}{2}(3 - \sqrt{2})).$$

One of which satisifies our conditions.

The problem arises in statistics, see John Mount's blog post (http://www.win-vector.com/blog/2014/07/frequenstist-inference-only-seems-easy/) for background.

Question: Is there a solution for every k?

Addendum: John says he wants soltions in the interor of $[0, 1]^{k+1}$...

```
In [0]:
```

Solution submitted 8-4-2013 by me (John Mount), having a lot of trouble with links and formatting. Enough of that, submitting it here.

This is some background to the question and the solution (minus one check mentioned at the end).

Define

$$S(k, p, x) = \sum_{i=0}^{k} \binom{k}{i} p^i (1-p)^{k-i} (x_i - p)^2.$$

Define

$$f_k(k) = \text{argmin}_x \max_p S(k, p, x).$$

Then $f_k(k)$ is the minimax square-loss solution to trying to estimate the win rate of a random process by observing k results (Wald wrote on this). The neat thing is: we can show if th is a real solution x in $[0, 1]^{k+1}$ to $S(k, p, x) = x_0^2$ then $x = f_k(k)$. Meaning we avoided two nasty quantifiers. See this (https://github.com/WinVector/Examples/blob/master/freq/python/freqMin.rst) for some experimental examples.

We know there is only one connected component of solutions in the interior of the unit cube because these solutions represent extreme points of the minimax estimation problem. show that there is a diversity of gradients by reflecting coordinates of x around p (and thus we have an extreme point).

From the original problem we expect a lot of symmetries. Also, a change of variables $z = p/(1-p)$ makes collecting terms easier. In fact I now have a conjectured exact solution, I n only need a proof that it always works (cancels the p's, is real and in the interior of $[0, 1]^{k+1}$; I already have a proof that such a solution when it exists solves the original estimat problem). The conjectured solution for $k > 1$ (for $k = 1$ the solution is $[1/4, 3/4]$) is:

$$\begin{aligned}
f_k(0) &= (\sqrt{k} - 1)/(2(k-1)) \\
f_k(1) &= \sqrt{f_k(0)^2 + 2f_k(0)/k} \\
&\text{for } h > 1 : \\
f_k(h)^2 &= (k+2)(k+1)(f_k(0)^2)/((k+2-h)(k+1-h)) \\
&+2hf_k(h-1)(1-f_k(h-1))/(k+1-h) \\
&-h(h-1)((f_k(h-2)-1)^2)/((k+2-h)(k+1-h))
\end{aligned}$$

A Python implementation, demonstration and check of this solution up through $k = 8$ is given h (https://github.com/WinVector/Examples/blob/master/freq/python/explicitSolution.rst). So really all that is left to prove is the right hand side of $f_k(h)^2$ is always positive and in the inter of $[0, 1]$ for all k, h.

Note 8-4-2014: Vladimir Dotsenko finished the solution (http://mathoverflow.net/a/177820/56665) by adding the important insight that the $f_k(h)$ are evenly spaced when k is h constant. This lets him get a closed form solution for each $f_k(h)$ (without having to refer to ealier h).

In [1]:
```
import sympy


# expecting a dictionary solution
def isGoodSoln(si):
   def isGoodVal(x):
      xn = complex(x)
      xr = xn.real
      xi = xn.imag
      return (abs(xi)<1.0e-6) and (xr>0.0) and (xr<1.0)
   return all([ isGoodVal(xi) for xi in si.values() ])


# only good for k>=1
def solveKz(k):
   vars = sympy.symbols(['phi' + str(i) for i in range((k+1)/2)])
   if k%2!=0:
      phis = vars + [1-varsi for varsi in reversed(vars) ]
   else:
      phis = vars + [sympy.Rational(1,2)] + [1-varsi for varsi in reversed(vars) ]
   z = sympy.symbols('z')
   poly = sum([ sympy.binomial(k,h) * z**h * ((1+z)*phis[h] -z)**2 for h in range(k+1)]) - phis[0]**2 * (1+z)**(k+2)
   polyTerms = poly.expand().collect(z,evaluate=False)
   eqns = [ polyTerms[ki] for ki in polyTerms.keys() if (not ki==1) ]
   solns = sympy.solve(eqns,vars,dict=True)
   soln1 = [ si for si in solns if isGoodSoln(si)][0]
   solnv = [ soln1[vi] for vi in vars ]
   if k%2!=0:
      xs = solnv + [1-solni for solni in reversed(solnv) ]
   else:
      xs = solnv + [sympy.Rational(1,2)] + [1-solni for solni in reversed(solnv) ]
   return xs

# original substitution from inspecting tri-diagonal recurrance
# only good for k>=1
def conjectureKa(k,numeric=False):
   if k<=1:
      return [sympy.Rational(1,4),sympy.Rational(3,4)]
   phi = [ 0 for i in range(k+1) ]
   phi[0] = (sympy.sqrt(k)-1)/(2*(k-1))
   phi[1] = (sympy.sqrt((phi[0]**2+2*phi[0]/k).expand())).simplify()
   if numeric:
      for h in range(2):
         phi[h] = float(phi[h])
   for h in range(2,(k+1)):
      phi[h] = sympy.sqrt(( (k+2)*(k+1)*(phi[0]**2)/((k+2-h)*(k+1-h)) + 2*h*phi[h-1]*(1-phi[h-1])/(k+1-h) - h*(h-1)*((phi[h-2]
1)**2)/((k+2-h)*(k+1-h)) ))
   return phi

# simplified in pseudo-observation form
def conjectureK(k,numeric=False):
    sqrtk = sympy.sqrt(k)
    if numeric:
        sqrtk = float(sqrtk)
    return [(sqrtk/2 + h)/(sqrtk+k) for h in range(k+1) ]
```

In [2]:
```
p = sympy.symbols('p')
for k in range(1,9):
   print
   print 'k',k
   solnk = solveKz(k)
   print 'soln       ',solnk
   poly = sum([ p**h * (1-p)**(k-h) * sympy.binomial(k,h) * (solnk[h]-p)**2 for h in range(k+1) ]).expand()
   print 'check poly',poly
   conjk = conjectureK(k)
   print 'conjecture:',conjk
   print 'max difference:',max([ abs(complex(solnk[i]-conjk[i])) for i in range(len(solnk)) ])
   print '1/k for scale:',1/float(k)
   print
```

```
k 1
soln        [1/4, 3/4]
check poly 1/16
conjecture: [1/4, 3/4]
max difference: 0.0
1/k for scale: 1.0


k 2
soln        [-1/2 + sqrt(2)/2, 1/2, -sqrt(2)/2 + 3/2]
check poly -sqrt(2)/2 + 3/4
conjecture: [sqrt(2)/(2*(sqrt(2) + 2)), (sqrt(2)/2 + 1)/(sqrt(2) + 2), (sqrt(2)/2 + 2)/(sqrt(2) + 2)]
max difference: 2.36364252615e-125
1/k for scale: 0.5


k 3
soln        [-1/4 + sqrt(3)/4, sqrt(3)/12 + 1/4, -sqrt(3)/12 + 3/4, -sqrt(3)/4 + 5/4]
check poly -sqrt(3)/8 + 1/4
conjecture: [sqrt(3)/(2*(sqrt(3) + 3)), (sqrt(3)/2 + 1)/(sqrt(3) + 3), (sqrt(3)/2 + 2)/(sqrt(3) + 3), (sqrt(3)/2 + 3)/(sqrt(3)
+ 3)]
max difference: 9.45457010461e-125
1/k for scale: 0.333333333333


k 4
soln        [1/6, 1/3, 1/2, 2/3, 5/6]
check poly 1/36
conjecture: [1/6, 1/3, 1/2, 2/3, 5/6]
max difference: 0.0
1/k for scale: 0.25


k 5
soln        [-1/8 + sqrt(5)/8, 1/8 + 3*sqrt(5)/40, sqrt(5)/40 + 3/8, -sqrt(5)/40 + 5/8, -3*sqrt(5)/40 + 7/8, -sqrt(5)/8 + 9/8]
check poly -sqrt(5)/32 + 3/32
conjecture: [sqrt(5)/(2*(sqrt(5) + 5)), (1 + sqrt(5)/2)/(sqrt(5) + 5), (sqrt(5)/2 + 2)/(sqrt(5) + 5), (sqrt(5)/2 + 3)/(sqrt(5)
+ 5), (sqrt(5)/2 + 4)/(sqrt(5) + 5), (sqrt(5)/2 + 5)/(sqrt(5) + 5)]
max difference: 9.45457010461e-125
1/k for scale: 0.2


k 6
soln        [-1/10 + sqrt(6)/10, 1/10 + sqrt(6)/15, sqrt(6)/30 + 3/10, 1/2, -sqrt(6)/30 + 7/10, -sqrt(6)/15 + 9/10, -sqrt(6)/1
 + 11/10]
check poly -sqrt(6)/50 + 7/100
conjecture: [sqrt(6)/(2*(sqrt(6) + 6)), (1 + sqrt(6)/2)/(sqrt(6) + 6), (sqrt(6)/2 + 2)/(sqrt(6) + 6), (sqrt(6)/2 + 3)/(sqrt(6)
+ 6), (sqrt(6)/2 + 4)/(sqrt(6) + 6), (sqrt(6)/2 + 5)/(sqrt(6) + 6), (sqrt(6)/2 + 6)/(sqrt(6) + 6)]
max difference: 7.56365608369e-124
1/k for scale: 0.166666666667


k 7
soln        [-1/12 + sqrt(7)/12, 1/12 + 5*sqrt(7)/84, sqrt(7)/28 + 1/4, sqrt(7)/84 + 5/12, -sqrt(7)/84 + 7/12, -sqrt(7)/28 + 3
4, -5*sqrt(7)/84 + 11/12, -sqrt(7)/12 + 13/12]
check poly -sqrt(7)/72 + 1/18
conjecture: [sqrt(7)/(2*(sqrt(7) + 7)), (1 + sqrt(7)/2)/(sqrt(7) + 7), (sqrt(7)/2 + 2)/(sqrt(7) + 7), (sqrt(7)/2 + 3)/(sqrt(7)
+ 7), (sqrt(7)/2 + 4)/(sqrt(7) + 7), (sqrt(7)/2 + 5)/(sqrt(7) + 7), (sqrt(7)/2 + 6)/(sqrt(7) + 7), (sqrt(7)/2 + 7)/(sqrt(7) +
)]
max difference: 7.56365608369e-124
1/k for scale: 0.142857142857


k 8
soln        [-1/14 + sqrt(2)/7, 1/14 + 3*sqrt(2)/28, sqrt(2)/14 + 3/14, sqrt(2)/28 + 5/14, 1/2, -sqrt(2)/28 + 9/14, -sqrt(2)/1
 + 11/14, -3*sqrt(2)/28 + 13/14, -sqrt(2)/7 + 15/14]
check poly -sqrt(2)/49 + 9/196
conjecture: [sqrt(2)/(2*sqrt(2) + 8), (1 + sqrt(2))/(2*sqrt(2) + 8), (sqrt(2) + 2)/(2*sqrt(2) + 8), (sqrt(2) + 3)/(2*sqrt(2) +
8), (sqrt(2) + 4)/(2*sqrt(2) + 8), (sqrt(2) + 5)/(2*sqrt(2) + 8), (sqrt(2) + 6)/(2*sqrt(2) + 8), (sqrt(2) + 7)/(2*sqrt(2) + 8)
 (sqrt(2) + 8)/(2*sqrt(2) + 8)]
max difference: 3.78182804185e-124
1/k for scale: 0.125
```

In [3]:
```
p = sympy.symbols('p')
for k in range(1,21):
   print
   print 'k',k
   conjk = conjectureK(k,numeric=True)
   print 'conjecture:',conjk
   polyc = sum([ p**h * (1-p)**(k-h) * sympy.binomial(k,h) * (conjk[h]-p)**2 for h in range(k+1) ]).expand()
   print 'conjecture check poly',polyc
   print '1/k for scale:',1/float(k)
   print
```

```
k 1
conjecture: [0.25, 0.75]
conjecture check poly 0.0625000000000000
1/k for scale: 1.0


k 2
conjecture: [0.20710678118654754, 0.5, 0.7928932188134525]
conjecture check poly 4.44089209850063e-16*p**2 + 0.0428932188134525
1/k for scale: 0.5


k 3
conjecture: [0.18301270189221933, 0.3943375672974065, 0.6056624327025936, 0.8169872981077807]
conjecture check poly -1.33226762955019e-15*p**4 + 8.88178419700125e-16*p**3 - 4.44089209850063e-16*p**2 + 1.11022302462516e-1
*p + 0.0334936490538903
1/k for scale: 0.333333333333


k 4
conjecture: [0.16666666666666666, 0.3333333333333333, 0.5, 0.6666666666666666, 0.8333333333333334]
conjecture check poly -6.66133814775094e-16*p**5 + 1.66533453693773e-15*p**4 + 0.0277777777777778
1/k for scale: 0.25


k 5
conjecture: [0.15450849718747373, 0.29270509831248426, 0.43090169943749473, 0.5690983005625052, 0.7072949016875157, 0.84549150
8125263]
conjecture check poly 7.105427357601e-15*p**5 - 3.5527136788005e-15*p**4 + 8.88178419700125e-16*p**3 - 2.22044604925031e-16*p*
2 + 1.11022302462516e-16*p + 0.0238728757031316
1/k for scale: 0.2


k 6
conjecture: [0.1449489742783178, 0.2632993161855452, 0.38164965809277257, 0.5, 0.6183503419072274, 0.7367006838144547, 0.85505
0257216821]
conjecture check poly -7.105427357601e-15*p**7 + 1.4210854715202e-14*p**6 + 2.8421709430404e-14*p**5 - 1.4210854715202e-14*p**
 + 7.105427357601e-15*p**3 - 8.88178419700125e-16*p**2 + 1.11022302462516e-16*p + 0.0210102051443364
1/k for scale: 0.166666666667


k 7
conjecture: [0.1371459425887159, 0.24081853042051138, 0.3444911182523068, 0.4481637060841023, 0.5518362939158977, 0.6555088817
76932, 0.7591814695794886, 0.8628540574112842]
conjecture check poly 1.13686837721616e-13*p**7 - 2.8421709430404e-14*p**6 + 3.5527136788005e-15*p**3 + 4.44089209850063e-16*p
*2 + 5.55111512312578e-17*p + 0.0188090095685474
1/k for scale: 0.142857142857


k 8
conjecture: [0.13060193748187074, 0.22295145311140305, 0.3153009687409354, 0.4076504843704677, 0.5, 0.5923495156295323, 0.6846
90312590646, 0.777048546888597, 0.8693980625181293]
conjecture check poly 5.32907051820075e-14*p**9 + 1.77635683940025e-14*p**8 - 5.59552404411079e-14*p**7 - 1.4210854715202e-14*
**6 - 5.6843418860808e-14*p**5 + 1.4210854715202e-14*p**4 - 3.5527136788005e-15*p**3 + 8.88178419700125e-16*p**2 + 0.017056866
740185
1/k for scale: 0.125


k 9
conjecture: [0.125, 0.20833333333333334, 0.2916666666666667, 0.375, 0.4583333333333333, 0.5416666666666666, 0.625, 0.708333333
333334, 0.7916666666666666, 0.875]
conjecture check poly -4.54747350886464e-13*p**8 + 2.27373675443232e-13*p**6 + 5.6843418860808e-14*p**4 - 7.105427357601e-15*p
*3 - 4.44089209850063e-16*p**2 + 5.55111512312578e-17*p + 0.015625
1/k for scale: 0.111111111111


k 10
conjecture: [0.12012653667602108, 0.19610122934081686, 0.27207592200561265, 0.34805061467040843, 0.4240253073352042, 0.5, 0.57
```

```
9746926647957, 0.6519493853295915, 0.7279240779943873, 0.803898770659183, 0.8798734633239789]
conjecture check poly 4.54747350886464e-13*p**10 - 1.13686837721616e-13*p**9 - 9.66338120633736e-13*p**8 + 1.4210854715202e-14
p**7 + 5.6843418860808e-14*p**6 + 1.13686837721616e-13*p**5 + 5.6843418860808e-14*p**4 - 7.105427357601e-15*p**3 + 0.014430384
137754
1/k for scale: 0.1


k 11
conjecture: [0.11583123951777, 0.18568010505999363, 0.25552897060221724, 0.3253778361444409, 0.3952267016866645, 0.46507556722
88815, 0.5349244327711118, 0.6047732983133354, 0.6746221638555591, 0.7444710293977826, 0.8143198949400063, 0.8841687604822299]
conjecture check poly -1.36424205265939e-12*p**12 - 1.81898940354586e-12*p**11 + 2.72848410531878e-12*p**10 + 5.00222085975111
-12*p**9 + 1.02318153949454e-12*p**8 - 2.68585154117318e-12*p**7 - 9.09494701772928e-13*p**6 + 4.54747350886464e-13*p**5 - 5.6
43418860808e-14*p**4 + 1.06581410364015e-14*p**3 - 8.88178419700125e-16*p**2 - 5.55111512312578e-17*p + 0.013416876048223
1/k for scale: 0.0909090909091


k 12
conjecture: [0.11200461886989793, 0.17667051572491496, 0.24133641257993196, 0.30600230943494894, 0.370668206289966, 0.43533410
144983, 0.5, 0.564665896855017, 0.629331793710034, 0.693997690565051, 0.758663587420068, 0.8233294842750851, 0.887995381130102
]
conjecture check poly 1.81898940354586e-12*p**12 + 3.63797880709171e-12*p**11 + 7.27595761418343e-12*p**10 - 7.27595761418343e
12*p**9 + 1.45519152283669e-11*p**8 - 3.63797880709171e-12*p**7 - 1.81898940354586e-12*p**6 + 1.13686837721616e-13*p**4 - 1.42
0854715202e-14*p**3 - 8.88178419700125e-16*p**2 + 1.11022302462516e-16*p + 0.0125450346481911
1/k for scale: 0.0833333333333


k 13
conjecture: [0.10856463647766623, 0.16878546163494837, 0.22900628679223048, 0.2892271119495126, 0.3494479371067947, 0.40966876
2640769, 0.469889587421359, 0.5301104125786411, 0.5903312377359232, 0.6505520628932053, 0.7107728880504874, 0.7709937132077695
 0.8312145383650517, 0.8914353635223338]
conjecture check poly 4.54747350886464e-13*p**14 - 3.27418092638254e-11*p**12 + 8.54925019666553e-11*p**11 + 1.8644641386345e-
1*p**10 + 2.13731254916638e-11*p**9 - 1.36424205265939e-12*p**8 + 1.00044417195022e-11*p**7 - 9.09494701772928e-13*p**6 + 6.82
21026329696e-13*p**5 - 2.8421709430404e-13*p**4 + 2.8421709430404e-14*p**3 - 8.88178419700125e-16*p**2 + 0.0117862802935278
1/k for scale: 0.0769230769231


k 14
conjecture: [0.10544836102976697, 0.1618128808826574, 0.21817740073554784, 0.2745419205884383, 0.3309064404413287, 0.387270960
9421915, 0.4436354801471096, 0.5, 0.5563645198528905, 0.6127290397057809, 0.6690935595586713, 0.7254580794115617, 0.7818225992
44522, 0.8381871191173426, 0.894551638970233]
conjecture check poly 1.45519152283669e-11*p**14 - 5.82076609134674e-11*p**11 + 2.91038304567337e-11*p**9 + 1.81898940354586e-
2*p**6 + 1.4210854715202e-12*p**5 + 1.13686837721616e-13*p**4 + 0.0111193568438641
1/k for scale: 0.0714285714286


k 15
conjecture: [0.10260654807883632, 0.1555923416683248, 0.20857813525781332, 0.2615639288473018, 0.31454972243679025, 0.36753551
02627876, 0.4205213096157673, 0.47350710320525574, 0.5264928967947442, 0.5794786903842327, 0.6324644839737212, 0.6854502775632
98, 0.7384360711526983, 0.7914218647421867, 0.8444076583316752, 0.8973934519211638]
conjecture check poly 7.27595761418343e-12*p**16 + 2.18278728425503e-11*p**15 + 1.45519152283669e-11*p**14 + 2.7648638933897e-
0*p**13 + 4.51109372079372e-10*p**12 - 7.27595761418343e-11*p**11 - 1.55750967678614e-10*p**10 + 3.79714037990198e-11*p**9 + 2
5465851649642e-11*p**8 + 2.18278728425503e-11*p**7 + 4.54747350886464e-12*p**6 - 6.82121026329696e-13*p**5 + 1.4210854715202e-
3*p**4 - 3.5527136788005e-14*p**3 + 8.88178419700125e-16*p**2 + 0.0105281037086545
1/k for scale: 0.0666666666667


k 16
conjecture: [0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9]
conjecture check poly -2.90967250293761e-12*p**17 - 7.91544607636752e-11*p**16 - 8.3787199400831e-11*p**15 - 1.86219040188007e
10*p**14 - 8.3855411503464e-10*p**13 + 7.09405867382884e-10*p**12 + 8.54925019666553e-11*p**11 + 4.91127138957381e-11*p**10 +
.81898940354586e-11*p**9 + 3.27418092638254e-11*p**8 - 3.63797880709171e-11*p**7 - 4.09272615797818e-12*p**6 - 1.5916157281026
e-12*p**5 + 1.70530256582424e-13*p**4 - 7.105427357601e-15*p**3 + 1.77635683940025e-15*p**2 - 5.55111512312578e-17*p + 0.01
1/k for scale: 0.0625


k 17
conjecture: [0.0975970508005519, 0.14493857423578108, 0.19228009767101029, 0.23962162110623947, 0.28696314454146865, 0.3343046
797669783, 0.381646191411927, 0.4289877148471562, 0.4763292382823854, 0.5236707617176146, 0.5710122851528437, 0.61835380858807
, 0.6656953320233021, 0.7130368554585313, 0.7603783788937605, 0.8077199023289897, 0.855061425764219, 0.9024029491994481]
conjecture check poly 9.09494701772928e-12*p**18 + 1.01863406598568e-10*p**17 - 4.36557456851006e-11*p**16 - 8.18545231595635e
10*p**15 - 4.16548573412001e-9*p**14 - 7.89668774814345e-10*p**13 - 3.20142135024071e-9*p**12 + 3.72529029846191e-9*p**11 + 4.
6561898570508e-10*p**10 - 1.16415321826935e-10*p**9 + 2.47382558882236e-10*p**8 + 2.18278728425503e-11*p**7 + 3.63797880709171
-12*p**6 - 9.09494701772928e-13*p**5 + 2.27373675443232e-13*p**4 - 2.8421709430404e-14*p**3 + 1.77635683940025e-15*p**2 - 1.11
22302462516e-16*p + 0.00952518432496551
1/k for scale: 0.0588235294118


k 18
conjecture: [0.0953717849152731, 0.14033047548024274, 0.1852891660452124, 0.23024785661018204, 0.2752065471751517, 0.320165237
4012134, 0.36512392830509105, 0.4100826188700607, 0.45504130943503035, 0.5, 0.5449586905649697, 0.5899173811299393, 0.63487607
```

```
694909, 0.6798347622598786, 0.7247934528248483, 0.7697521433898179, 0.8147108339547876, 0.8596695245197573, 0.9046282150847269
conjecture check poly -1.37049482873408e-10*p**19 - 2.29107399718487e-10*p**18 - 2.19642970478162e-10*p**17 + 1.23463905765675
-9*p**16 - 1.12613633973524e-8*p**15 - 3.34694050252438e-10*p**14 - 2.85945134237409e-9*p**13 + 1.1423253454268e-9*p**12 - 1.3
422428578138e-9*p**11 + 1.54977897182107e-9*p**10 - 1.16415321826935e-10*p**9 - 8.36735125631094e-11*p**8 - 3.63797880709171e-
2*p**7 + 3.63797880709171e-12*p**6 - 4.54747350886464e-13*p**5 - 2.8421709430404e-13*p**4 - 2.8421709430404e-14*p**3 - 8.88178
19700125e-16*p**2 + 0.00909577735792511
1/k for scale: 0.0555555555556


k 19
conjecture: [0.09330274843168537, 0.13611298543887637, 0.1789232224460674, 0.2217334594532584, 0.26454369646044945, 0.30735393
46764045, 0.35016417047483145, 0.39297440748202245, 0.4357846444892135, 0.4785948814964045, 0.5214051185035955, 0.564215355510
865, 0.6070255925179775, 0.6498358295251685, 0.6926460665323596, 0.7354563035395506, 0.7782665405467416, 0.8210767775539326, 0
8638870145611236, 0.9066972515683146]
conjecture check poly 4.14729584008455e-10*p**20 + 7.42147676646709e-10*p**19 + 4.07453626394272e-9*p**18 - 5.355104804039e-9*
**17 - 1.86264514923096e-9*p**16 - 1.86264514923096e-8*p**15 - 5.58793544769287e-9*p**14 - 1.86264514923096e-9*p**10 + 4.65661
87307739e-10*p**8 - 5.82076609134674e-11*p**7 - 3.63797880709171e-12*p**5 + 4.54747350886464e-13*p**4 - 2.8421709430404e-14*p*
3 + 1.77635683940025e-15*p**2 - 5.55111512312578e-17*p + 0.00870540286490637
1/k for scale: 0.0526315789474


k 20
conjecture: [0.09137199881577841, 0.13223479893420056, 0.17309759905262273, 0.2139603991710449, 0.25482319928946706, 0.2956859
94078892, 0.33654879952631134, 0.37741159964473353, 0.41827439976315567, 0.45913719988157786, 0.5, 0.540862800118422, 0.58172
6002368443, 0.6225884003552665, 0.6634512004736887, 0.7043140005921108, 0.7451768007105329, 0.7860396008289551, 0.826902400947
773, 0.8677652010657995, 0.9086280011842216]
conjecture check poly -2.18278728425503e-10*p**21 + 5.93718141317368e-9*p**20 - 8.2072801887989e-9*p**19 - 3.72529029846191e-9
p**18 + 1.80152710527182e-8*p**17 - 2.0467268768698e-8*p**16 + 5.35210347152315e-8*p**15 - 3.49828042089939e-8*p**14 + 3.07591
08139604e-8*p**13 + 3.95812094211578e-9*p**12 - 6.54836185276508e-11*p**11 + 1.86264514923096e-9*p**10 - 9.31322574615479e-10*
**9 + 7.56699591875076e-10*p**8 + 1.18234311230481e-10*p**7 - 5.45696821063757e-11*p**6 - 2.72848410531878e-12*p**5 + 3.410605
3164848e-13*p**4 - 5.55111512312578e-17*p + 0.00834884216759061
1/k for scale: 0.05
```