Using the Matlab code of Lecture 15 we find that $C = 40$ channels at $PB$ (Gos) = 5% will support $\rho = 34.59$ Erlangs. We are told $\lambda, \mu = (2 \text{hr}^{-1})(2 \text{min}) = \frac{12 \text{min}}{60 \text{min}} = \frac{1}{15}$.

So $\frac{1}{15}U = 34.59 \rightarrow U = 519 \text{ users}$.
\( D = j \sqrt{3} R \)

\[ D^2 = j^2 3R^2 + k^2 3R^2 - 2jk 3R^2 \cos 120^\circ \]

\[ \cos 120^\circ = -\frac{1}{2} \rightarrow D^2 = 3R^2 (j^2 + k^2 + jk) \]

but (see 13.16) \( j^2 + k^2 + jk = N \)

So \( D^2 = 3N \rightarrow \frac{D}{R} = \sqrt{3N} \)
5.3

The sides of the square cell have length $\sqrt{2}R$, which is also the spacing between the basestations. So the square cell’s area is $A_{sc} = 2R^2$. (This is the largest square cell such that every point in the plane is at radius R from at least one basestation). The efficiency percentage is therefore

$100 \times \frac{2R^2}{\pi R^2} = 63.7\%$. This is substantially less than the hexagonal cell efficiency, which is

$100 \times \frac{3\sqrt{3}R^2/2}{\pi R^2} = 82.7\%$. 
5.4 \[
\frac{600}{4} = 150 \text{ channels per cell.} \text{ With PB = 2\% these will support 136.8 Erlangs. The carried traffic will be (136.8) \cdot 0.98 = 134.1 \text{ Erlangs.}}
\]
At $0.05/\text{min} \text{ this will provide } 134.1 \cdot 0.05 = 6.70/\text{min per cell in revenue.}

If we are able to use \( N = 3 \) then 
\[ C = \frac{600}{3} = 200 \text{ channels per cell.} \text{ At PB = 2\% this supports } p = 186.15 \text{ Erlangs or } 9.31 \text{ cell}^{-1} \text{ mm}^{-1}.
\]
9.31 - 6.70 = $2.61 \text{ mm}^{-1} \text{ cell}^{-1} \text{ in additional $}. \text{ You would want to consider if the cost of the smart antenna technology amortized over the life of the equipment is substantially less than $2.61 \text{ mm}^{-1} \text{ cell}^{-1}$. If it is, then the new technology will make money for your company. You would also need to consider if the smart antenna technology is fully compatible with your existing system. Will it cause new problems? Who will support it, and so on.
need $S/I = 10 \text{dB} = 10$. With $n = 3$

\[ \frac{\sqrt{3N}}{L} = 10 \rightarrow N = 5.1 \rightarrow N = 7 \text{ reuse} \]

(there is no $N = 6$ pattern)

\[ \frac{600 \text{ total channels}}{7 \text{ cells}} = 85.7 \rightarrow 85 \text{ channels per cell (round down)} \]

C = 85 @ PB = 2% \rightarrow \rho = 73.49 \text{ Erlangs per cell}

a customer will use 1000 minutes per day

\[ \frac{8 \text{ hr} \cdot 20 \text{ day} \cdot 60 \text{ min}}{\text{hr}} = 9,600 \text{ min} \]

So $\lambda_H = \frac{1000}{9600}$ is the carried traffic per user

$\rightarrow \lambda_H = \frac{10}{96} = 0.1042$ Erlangs/user

$0.1042 \cdot U = 73.49 \rightarrow U = \frac{73.49}{0.1042} = 705 \text{ users per cell}$

there are 500 users $\text{km}^{-2}$ so cell area is

\[ 2.6R^2 = \frac{705}{500} = 1.410 \rightarrow R = 0.7364 \text{ km} \]
12500 = ρ(1 – PB) × billing rate / month of calling time

Billing rate / month of calling time = 12500/(73.49 × (1 – 0.02)) = $173.56 per user per month of calling time.

But users are only making calls a fraction 1000/9600 of the month.

Monthly rate = (Billing rate / month of calling time) × 1000/9600 = 173.56 × (10/96) = $18.08 per user per month
erlangs.c (problem 6 of homework 5)
Scott Hudson
29-Sep-2002

This is a straight-forward implementation of the erlangs Matlab function given in Lecture 15 of EE432.

/*first we need to implement the PB function of the same notes*/
double PB(int c, double rho)
{
    double sum=1.0, x=1.0;
    int i;

    /*this is a direct coding of equation (15.10)*/
    for (i=c; i>=1; i--) {
        x *= i/rho;
        sum += x;
    }
    return 1/sum;
}

/*now we can implement the erlangs function*/
double erlangs(int c, double pb)
{
    double rho;

    rho = (double) c;
    /*for large enough PB, we can have rho>c, in which case
    PB(c,rho) will be < the desired pb. so, we need to start out
    with a larger rho.*/
    while (PB(c, rho)<=pb)
        rho += 1.0;
    /*at this point we know that the current value of rho is too
    large, so we will start shaving it down until we find the
    right value. first we shave off the integer part.*/
    while (PB(c, rho)>pb)
    {
        rho -= 1.0;
        rho += 1.0;
    /*now the first decimal place.*/
    while (PB(c, rho)>pb)
    {
        rho -= 1.0;
        rho += 1.0;
        /*now the second decimal place. and that is good enough.*/
    }
    }
    return rho;
}

main()
{
double pb=0.01, rho;
int c;

for (c=5; c<=100; c++) {
    /*the if statement starts a new line every 5th value*/
    if (c==5*(c/5))
        printf("\n");
    printf("%3d %6.2f  ", c, erlangs(c, pb));
}
}
5.7

% hexagon(x,y,R) plots a hexagon on the current figure
% centered at x,y with radius R
function hexagon(x,y,R)
    hexagonx(1) = x+R; %there are 6 vertices at a distance R from the
    hexagony(1) = y; %center, the first is to the right of the center
    for i=2:7 %the others are rotated 60 degrees (pi/3), vertex 7 is the same
        hexagonx(i) = x+R*cos(pi*(i-1)/3); %as vertex 1 so we close the
        hexagony(i) = y+R*sin(pi*(i-1)/3); %figure
    end
    plot(hexagonx, hexagony);
    clear hexagonx, hexagony;

Save this in a file named hexagon.m. Then you can do things like:

>>
>> hexagon(0,0,1)
>> hold on
>> axis equal
>> hexagon(3,0,1)
>> hexagon(1.5,sqrt(3)/2,1)
>> hexagon(1.5,-sqrt(3)/2,1)
>>

to get the following figure

![Hexagon Plot](image-url)