# CORRESPONDENCE BETWEEN THE SOLUTIONS OF AN EQUATION AND THE DIVISORS OF ODD NUMBERS

Edoardo GUEGLIO

egueglio@gmail.com

*There is a correspondence between the positive solutions of a diophantine equation and the divisors of odd numbers*

**Premise**

All this research has been done using Mathematica©, a symbolic computation environment that uses a programming language called Wolfram Language.

I investigated the positive solutions of Diophantine equation x+y+2*x*y=n where n is a natural number greater 2. Because this is a symmetric equation I added x<=y to remove redundant solutions. In terms of Mathematica all this can be expressed as:

```
Reduce[x+y+2*x*y==n&&x>0&&y>0&&x≤y,{x,y},Integers]
```

Reduce is the function used to solve diophantine equations and has three arguments:

- the set of conditions that must be satisfied by the variables joined together by the symbol && which stands for 'and'
- the set of variable names (in this case x and y)
- the domain of the values (in this case the domain of integers)

To complete the informations needed to understand the following we have another functions:

```
Divisors[n]
```

This function given a natural n returns the list expressed as a comma separated sequence of values between braces that starts from 1 and goes to n. Example:

```
In[ ]:= Divisors[30]
Out[ ]= {1, 2, 3, 5, 6, 10, 15, 30}
```

Now lets try to calculate the solutions of the above equation replacing n with 10,12,22,30 for example and use in Divisors function the mapping n->2*n+1:

```
Reduce[x+y+2*x*y==10&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*10+1]
```

Out[•]= x == 1 && y == 3

Out[•]= {1, 3, 7, 21}

```
Reduce[x+y+2*x*y==12&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*12+1]
```

Out[•]= x == 2 && y == 2

Out[•]= {1, 5, 25}

```
Reduce[x+y+2*x*y==22&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*22+1]
```

Out[•]= (x == 1 && y == 7) || (x == 2 && y == 4)

Out[•]= {1, 3, 5, 9, 15, 45}

```
Reduce[x+y+2*x*y==40&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*40+1]
```

Out[•]= (x == 1 && y == 13) || (x == 4 && y == 4)

Out[•]= {1, 3, 9, 27, 81}

What we get is a list of solutions, each enclosed in round brackets and separated by symbol || which stands for 'or' followed by the list of divisors. **The interesting thing is that if you calculate in each solution 2y+1 you get the same values in the second half of divisors list except the last number if the number of divisors is even or the second half plus the medium value except the last number if the number of divisors is odd.**

Following this concept what is happen when we use for n a prime number? A prime number p has the divisors list like {1,p} and hence the second half except last corresponds to empty solutions list:

```
Reduce[x+y+2*x*y==15&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*15+1]
```

Out[•]= False

Out[•]= {1, 31}

Output False means that there are no solutions.

Now we start with 2n+1=0 to check what we have verified. 1 in this case is considered as a prime.

In[•]:=
```
Reduce[x+y+2*x*y==0&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*0+1]
```

Out[•]= False

Out[•]= {1}

In[ ]:=
```
Reduce[x+y+2*x*y==1&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*1+1]
```

Out[ ]= False

Out[ ]= {1, 3}

In[ ]:=
```
Reduce[x+y+2*x*y==2&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*2+1]
```

Out[ ]= False

Out[ ]= {1, 5}

In[ ]:=
```
Reduce[x+y+2*x*y==3&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*3+1]
```

Out[ ]= False

Out[ ]= {1, 7}

In[ ]:=
```
Reduce[x+y+2*x*y==4&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*4+1]
```

Out[ ]= x == 1 && y == 1

Out[ ]= {1, 3, 9}

In[ ]:=
```
Reduce[x+y+2*x*y==5&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*5+1]
```

Out[ ]= False

Out[ ]= {1, 11}

In[ ]:=
```
Reduce[x+y+2*x*y==6&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*6+1]
```

Out[ ]= False

Out[ ]= {1, 13}

In[ ]:=
```
Reduce[x+y+2*x*y==7&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*7+1]
```

Out[ ]= x == 1 && y == 2

Out[ ]= {1, 3, 5, 15}

In[ ]:=
```
Reduce[x+y+2*x*y==8&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*8+1]
```

Out[ ]= False

Out[ ]= {1, 17}

In[ ]:=
```
Reduce[x+y+2*x*y==9&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*9+1]
```

Out[ ]= False

Out[ ]= {1, 19}

In[ ]:=
```
Reduce[x+y+2*x*y==10&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*10+1]
```

Out[ ]= x == 1 && y == 3

Out[ ]= {1, 3, 7, 21}

In[ ]:=
```
Reduce[x+y+2*x*y==11&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*11+1]
```

Out[ ]= False

Out[ ]= {1, 23}

In[ ]:=
```
Reduce[x+y+2*x*y==12&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*12+1]
```

Out[ ]= x == 2 && y == 2

Out[ ]= {1, 5, 25}

In[ ]:=
```
Reduce[x+y+2*x*y==13&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*13+1]
```

Out[ ]= x == 1 && y == 4

Out[ ]= {1, 3, 9, 27}

In[ ]:=
```
Reduce[x+y+2*x*y==14&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*14+1]
```

Out[ ]= False

Out[ ]= {1, 29}

In[ ]:=
```
Reduce[x+y+2*x*y==15&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*15+1]
```

Out[ ]= False

Out[ ]= {1, 31}

In[ ]:=
```
Reduce[x+y+2*x*y==16&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*16+1]
```

Out[ ]= x == 1 && y == 5

Out[ ]= {1, 3, 11, 33}

In[ ]:=
```
Reduce[x+y+2*x*y==17&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*17+1]
```

Out[ ]= x == 2 && y == 3

Out[ ]= {1, 5, 7, 35}

In[ ]:=
```
Reduce[x+y+2*x*y==18&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*18+1]
```

Out[ ]= False

Out[ ]= {1, 37}

In[ ]:=
```
Reduce[x+y+2*x*y==19&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*19+1]
```

Out[ ]= x == 1 && y == 6

Out[ ]= {1, 3, 13, 39}

In[ ]:=
```
Reduce[x+y+2*x*y==20&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*20+1]
```

Out[ ]= False

Out[ ]= {1, 41}

In[ ]:=
```
Reduce[x+y+2*x*y==21&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*21+1]
```

Out[ ]= False

Out[ ]= {1, 43}

In[ ]:=
```
Reduce[x+y+2*x*y==22&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*22+1]
```

Out[ ]= (x == 1 && y == 7) || (x == 2 && y == 4)

Out[ ]= {1, 3, 5, 9, 15, 45}

In[ ]:=
```
Reduce[x+y+2*x*y==23&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*23+1]
```

Out[ ]= False

Out[ ]= {1, 47}

In[ ]:=
```
Reduce[x+y+2*x*y==24&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*24+1]
```

Out[ ]= x == 3 && y == 3

Out[ ]= {1, 7, 49}

In[ ]:=
```
Reduce[x+y+2*x*y==25&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*25+1]
```

Out[ ]= x == 1 && y == 8

Out[ ]= {1, 3, 17, 51}

In[ ]:=
```
Reduce[x+y+2*x*y==26&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*26+1]
```

Out[ ]= False

Out[ ]= {1, 53}

In[ ]:=
```
Reduce[x+y+2*x*y==27&&x>0&&y>0&&x≤y,{x,y},Integers]
Divisors[2*27+1]
```

Out[ ]= x == 2 && y == 5

Out[ ]= {1, 5, 11, 55}

```
In[•]:= Reduce[x+y+2*x*y==28&&x>0&&y>0&&x≤y,{x,y},Integers]
        Divisors[2*28+1]
```

```
Out[•]= x == 1 && y == 9

Out[•]= {1, 3, 19, 57}
```

```
In[•]:= Reduce[x+y+2*x*y==29&&x>0&&y>0,{x,y},Integers]
        Divisors[2*29+1]
```

```
Out[•]= False

Out[•]= {1, 59}
```

## Main Results

We have found a way to identify odd prime numbers by an equation. **A number n is prime iff there are no positive solutions to diophantine equation x+y+2*x*y=(n-1)/2.**

For the non prime numbers all the proper divisors are identified; let (x,y) a solution of diophantine equation x+y+2*x*y=n then 2*y+1 is a divisor of 2*n+1.