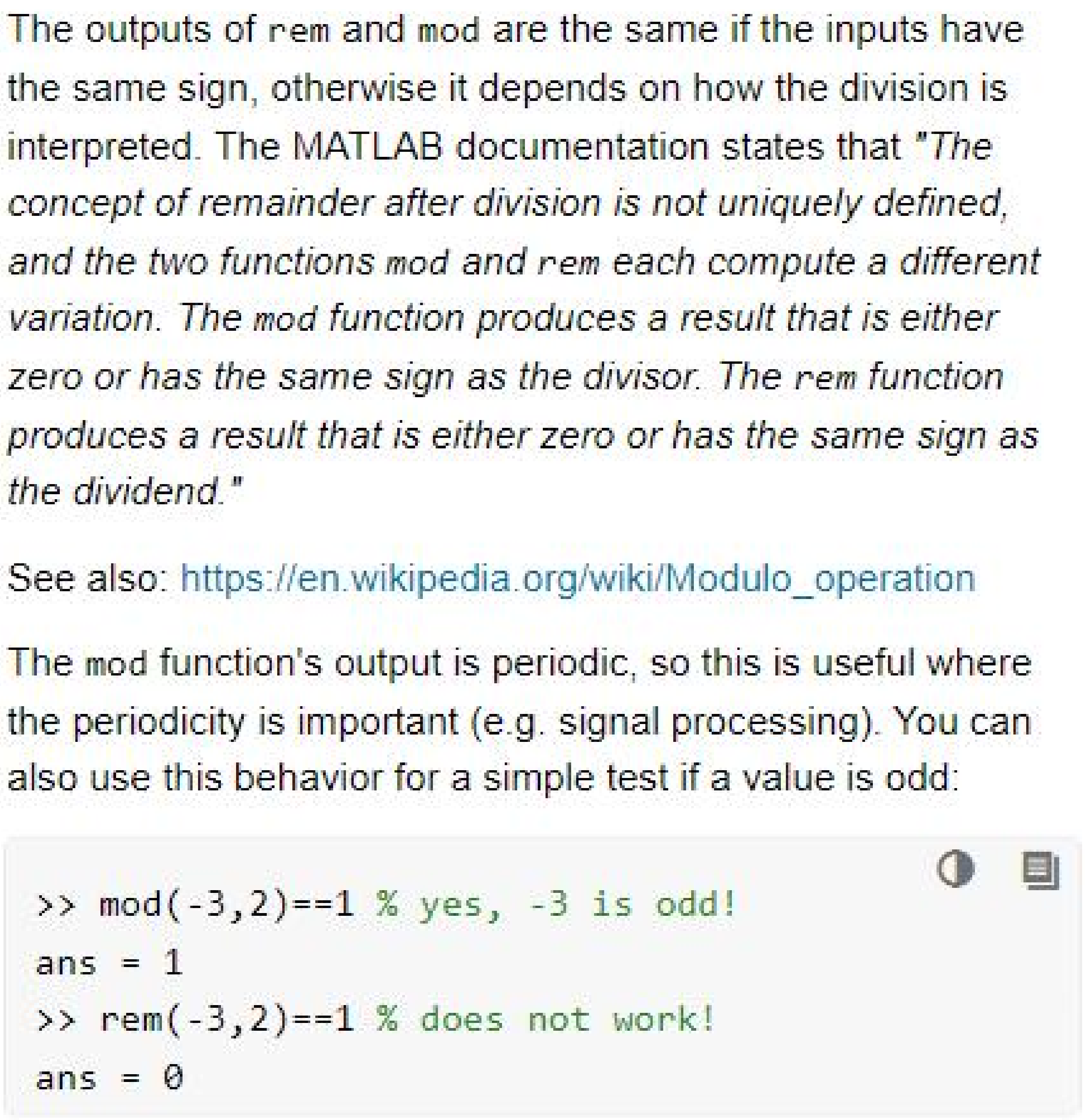
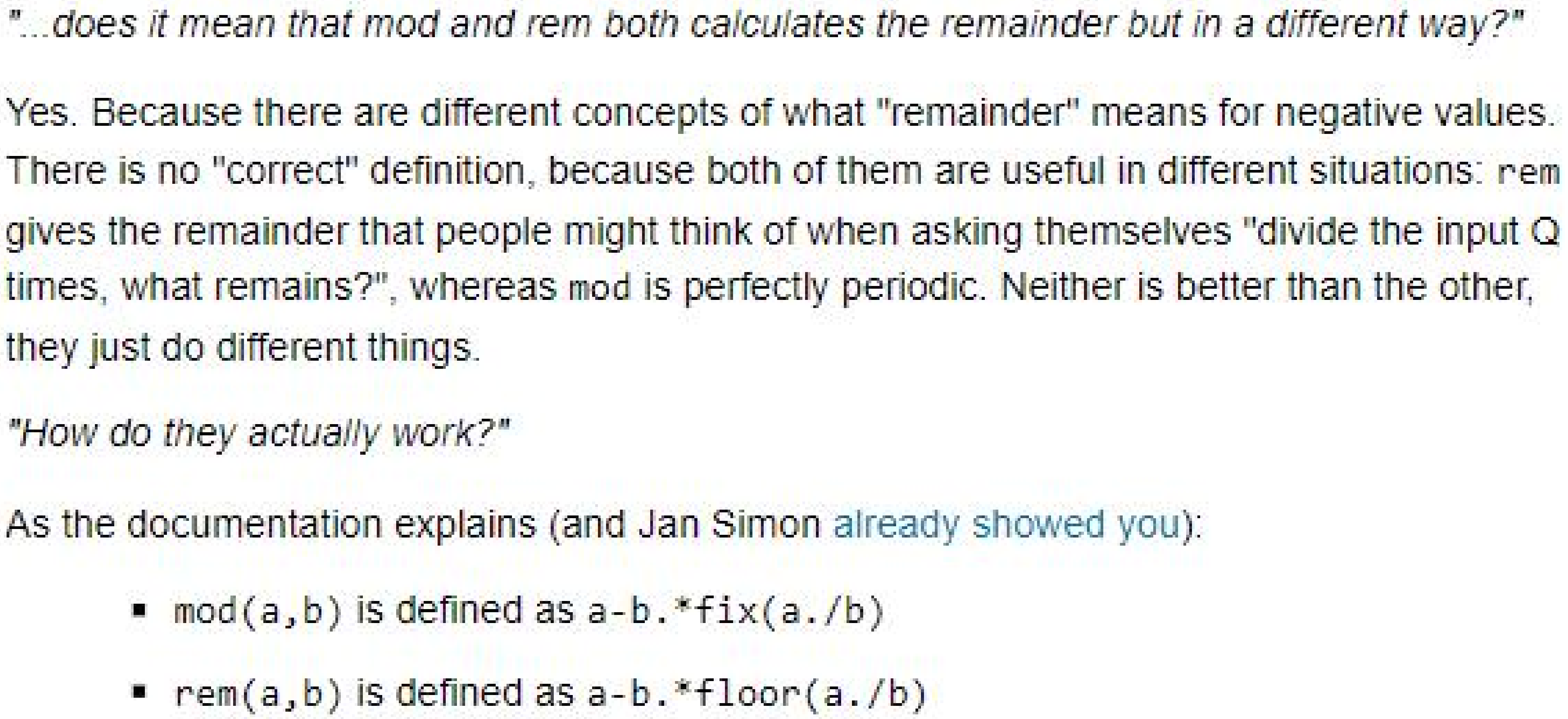


The outputs of `rem` and `mod` are the same if the inputs have the same sign, otherwise it depends on how the division is interpreted. The MATLAB documentation states that "*The concept of remainder after division is not uniquely defined, and the two functions `mod` and `rem` each compute a different variation. The `mod` function produces a result that is either zero or has the same sign as the divisor. The `rem` function produces a result that is either zero or has the same sign as the dividend.*"

See also: https://en.wikipedia.org/wiki/Modulo_operation

The `mod` function's output is periodic, so this is useful where the periodicity is important (e.g. signal processing). You can also use this behavior for a simple test if a value is odd:

```
>> mod(-3,2)==1 % yes, -3 is odd!
ans = 1
>> rem(-3,2)==1 % does not work!
ans = 0
```



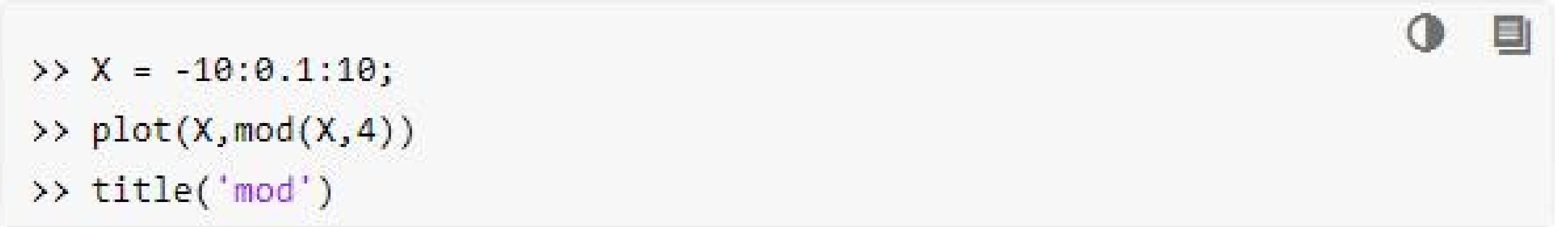
"...does it mean that mod and rem both calculates the remainder but in a different way?"

Yes. Because there are different concepts of what "remainder" means for negative values. There is no "correct" definition, because both of them are useful in different situations: `rem` gives the remainder that people might think of when asking themselves "divide the input Q times, what remains?", whereas `mod` is perfectly periodic. Neither is better than the other, they just do different things.

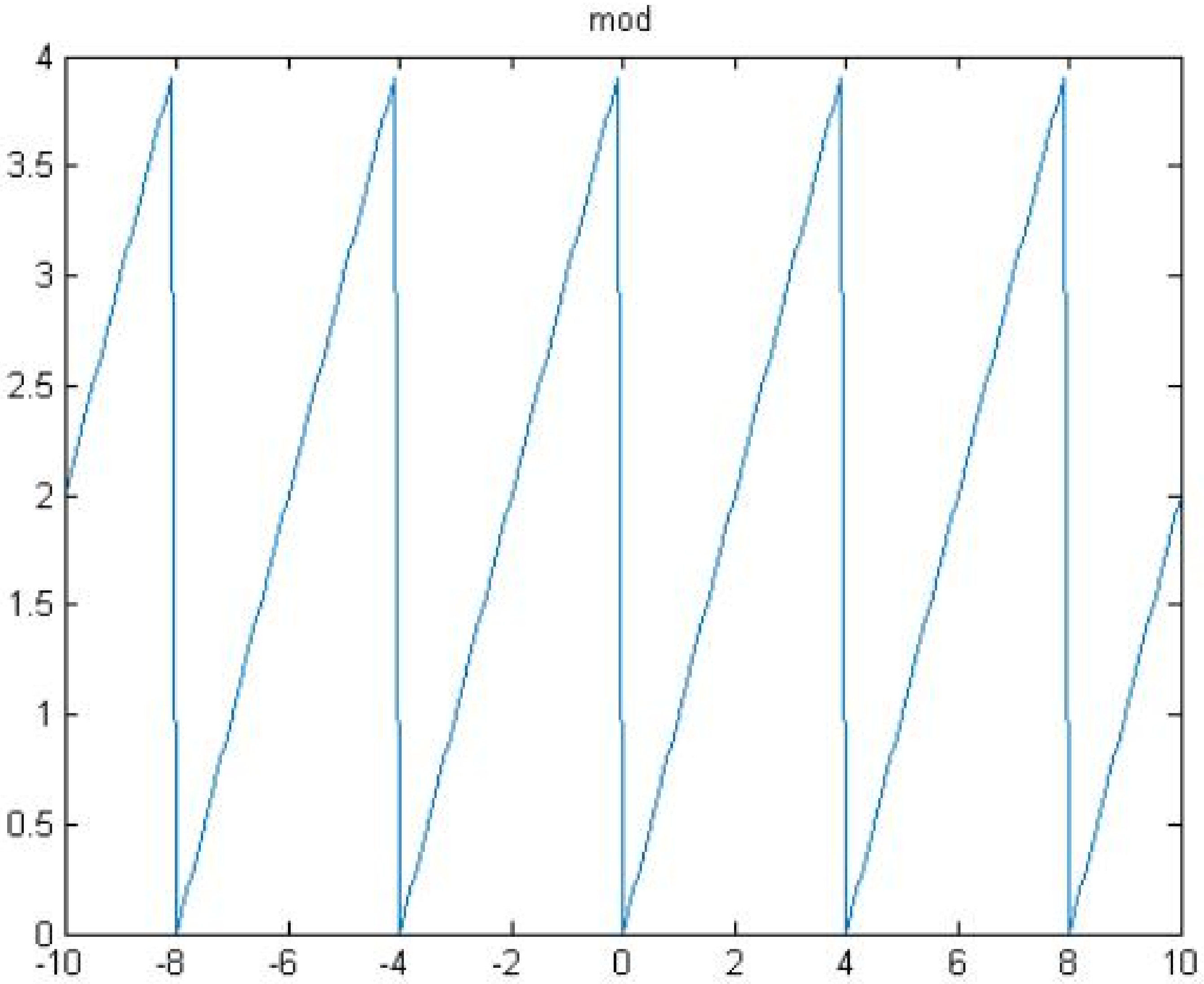
"How do they actually work?"

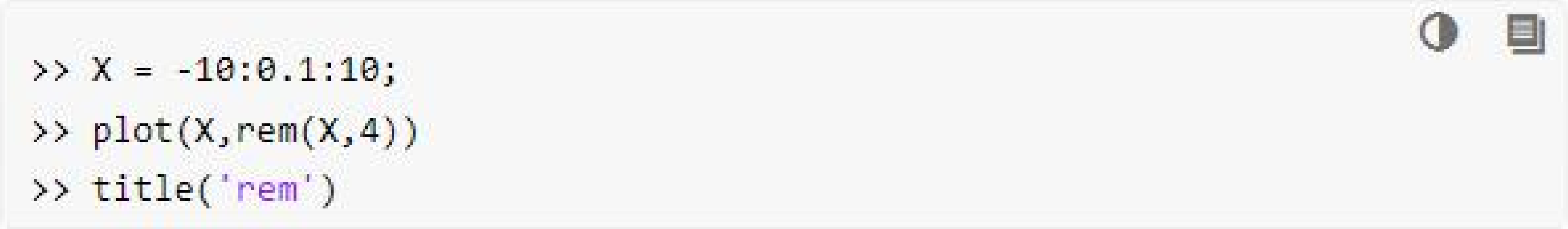
As the documentation explains (and Jan Simon already showed you):

- `mod(a,b)` is defined as `a-b.*fix(a./b)`
- `rem(a,b)` is defined as `a-b.*floor(a./b)`

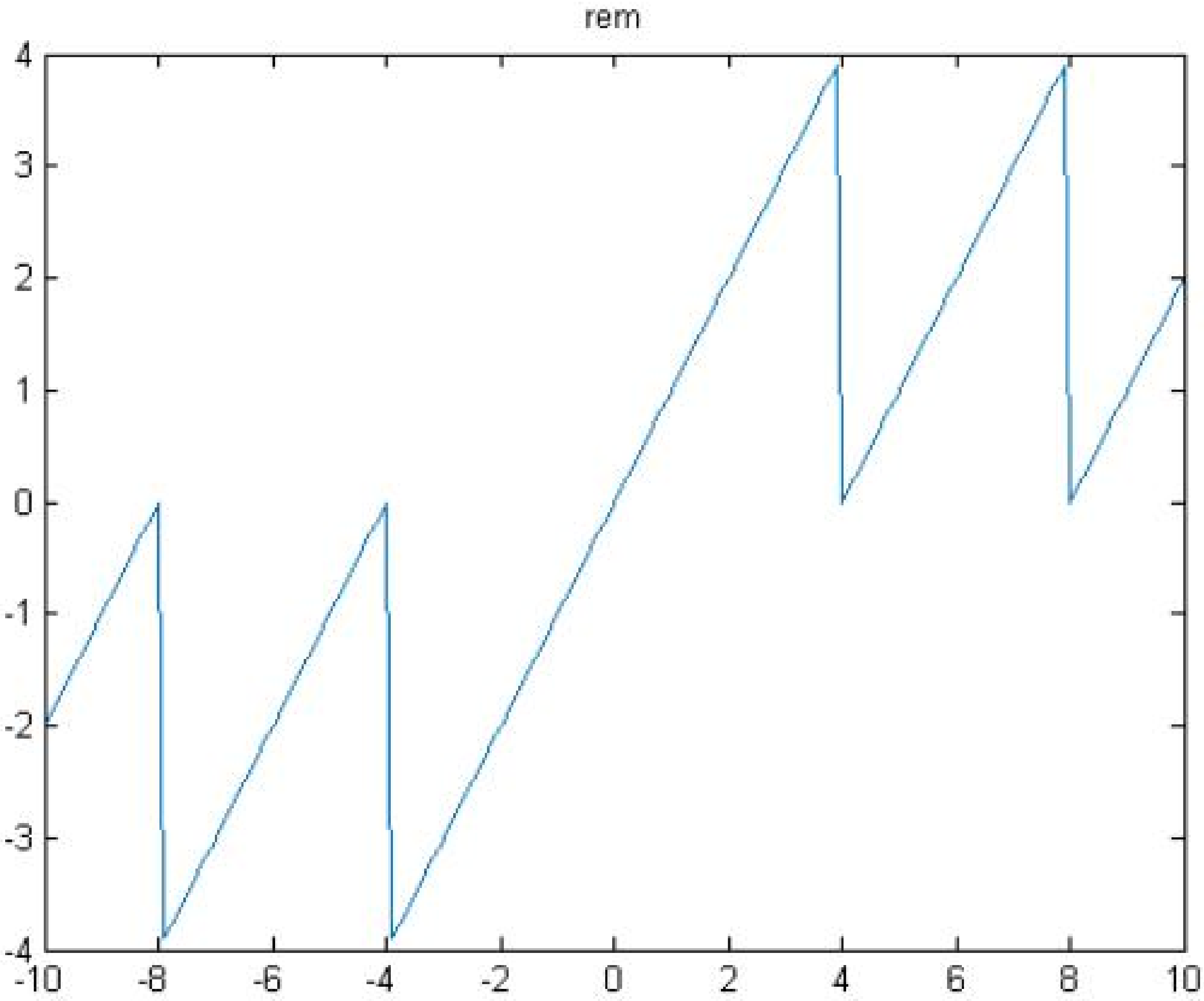


```
>> X = -10:0.1:10;
>> plot(X,mod(X,4))
>> title('mod')
```





```
>> X = -10:0.1:10;
>> plot(X,rem(X,4))
>> title('rem')
```


rem