#### RATE

Syntax:

RATE ( nper , pmt , pv [ , [ [ fv ] [ , [ [ type ] [ , [ guess ] ] ] ] ] ] )

Description: Computes the interest rate per period of an annuity, using iteration, which can result in zero or more solutions.

Arguments:

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| nper | number | The total number of payment periods. |
| pmt | number | The payment made each period and cannot change over the life of the annuity. (Typically, pmt includes principal and interest but no other fees or taxes.) If omitted, fv shall be present. |
| pv | number | The present value. |
| fv | number | The future value, or a cash balance to be attained after the last payment is made. If omitted, it is assumed to be 0 (i.e., the future value of a loan, for example, is 0). |
| type | number | The timing of the payment, truncated to integer, as follows:   |  |  | | --- | --- | | Value | Timing | | 0 | Payment at the end of the period | | 1 | Payment at the beginning of the period | |
| guess | number | A guess for what the rate will be. If omitted, it is assumed to be 10 percent. |

Return Type and Value: number – The interest rate per period of an annuity.

However, if

* type is any number other than 0 or 1, #NUM! is returned.
* The result has not converged after an implementation-defined number of iterations, #NUM! is returned.

[Example:  
  
RATE(4\*12,-200,8000) results in 0.7701%  
RATE(4\*12,-200,8000)\*12 results in 9.2418%  
  
end example]