Numerical Integral

Write a function called num_int that calculates the numerical integral of sin(x)/x using trapezoidal rule.

- The function should have three parameters: a, b and x
- The endpoints of the interval to integrate over: 0 < a < b
- The past parameter will be either a positive integer or a numpy array. This will represent the partition of the interval [a, b].
- Return the numerical approximation of the integral (a real number).

If the last parameter is a number, then divide the interval to that many trapezoids with equal width (use linspace).

If it is a numpy array, that will contain the points in the partition. The first number will be a and the last one b and the points will be increasing.

Example

```
Suppose that f(x) = sin(x)/x.
num_int(1, 2, 1)
Then the result should be: (f(1) + f(2))/2
num_int(1, 2, 2)
Then the result should be: (f(1) + f(1.5))/4 + (f(1.5) + f(2))/4
Since the partition is: [1, 1.5, 2]
In fact, the latter should be the same as:
num_int(1, 2, numpy.array([1,1.5,2]))
```

Hint: There are one less trapezoids in the partition than points. "http://wiki.math.bme.hu

Numerical Differential

Write a function called num_diff that calculates the numerical derivative of a function using central difference.

- The function should have 2 parameters: y and x
- y will be a numpy array containing the values of the function (y-coordinates).
- x This will determine the x-coordinate values.
- Its default value should be None in which case the x values should be (1, 2, ..., n) (n is the length of y). Otherwise, this parameter will be a numpy array with the same length as y and a (xi, yi) pair represents a point on the graph of the function.
- Return a numpy array that contains the central differences and is two elements shorter than y.

Hint:

The formula for central difference:

```
f '(xi) ? (yi+1-yi-1) / (xi+1-xi-1)
If x=None then the denominator is constant 2.
This formula is invalid for the first and last i indexes, therefore the result is 2 elements short
(f '(x2), f '(x3), ... f '(xn-1))
Use numpy!
```

Example

```
num_diff([1,4,9,16]) -> [4,6]
```

```
Which is the same as:
num_diff([1,4,9,16], [1,2,3,4]) -> [4,6]
In formula: [(9-1)/(3-1), (16-4)/(4-2)]
```

Note:

Each problem counts 1 point