

ΠΕΠΕΡΑΣΜΕΝΕΣ ΔΙΑΦΟΡΕΣ - MATLAB

Τρίγωνο Pascal

```
function pt = pascal_triangle(n)
% The first two rows are constant
pt(1, 1) = 1;
pt(2, 1 : 2) = [1 1];
% If only two rows are requested, then exit
if n < 3
    return
end
for r = 3 : n
    % The first element of every row is always 1
    pt(r, 1) = 1;
    % Every element is the addition of the two elements
    % on top of it. That means the previous row.
    for c = 2 : r-1
        pt(r, c) = pt(r-1, c-1) + pt(r-1, c);
    end
    % The last element of every row is always 1
    pt(r, r) = 1;
end
```

Κλήση συνάρτησης

```
>> >> pascal_triangle(5)  
ans =
```

```
1  0  0  0  0  
1  1  0  0  0  
1  2  1  0  0  
1  3  3  1  0  
1  4  6  4  1
```

Προς τα εμπρός διαφορές

```
function TFD = Fdiff(X, Y)
%% Input::
%      X = [ x0 x1 .. xn ] - object vector
%      Y = [ y0 y1 .. yn ] - image vector
%% Output:
%  TFD - table of forward differences
[ p, m ] = size(X); % m points, polynomial order <= m-1
if p ~= 1 || p ~= size(Y, 1) || m ~= size(Y, 2)
    error('divdiff: input vectors must have the same dimension');
end
TFD = zeros(m, m);
TFD(:, 1) = Y';
for j = 2 : m
    for i = 1 : (m - j + 1)
        TFD(i,j) = (TFD(i + 1, j - 1) - TFD(i, j - 1));
    end
end
end
```

Κλήση συνάρτησης

```
>> x= [10 15 20 25 30 35]
```

```
x =
```

```
10 15 20 25 30 35
```

```
>> y= [19.97 21.51 22.47 23.52 24.65 25.89]
```

```
y =
```

```
19.970 21.510 22.470 23.520 24.650 25.890
```

```
>> TFD = Fdiff(x, y)
```

```
TFD =
```

19.97000	1.54000	-0.58000	0.67000	-0.68000	0.72000
21.51000	0.96000	0.09000	-0.01000	0.04000	0.00000
22.47000	1.05000	0.08000	0.03000	0.00000	0.00000
23.52000	1.13000	0.11000	0.00000	0.00000	0.00000
24.65000	1.24000	0.00000	0.00000	0.00000	0.00000
25.89000	0.00000	0.00000	0.00000	0.00000	0.00000

Προς τα πίσω διαφορές

```
function TBD = Bdiff(X, Y)
% Input::
%      X = [ x0 x1 .. xn ] - object vector
%      Y = [ y0 y1 .. yn ] - image vector
% Output:
% TBD - table of backward differences
[ p, m ] = size(X); % m points, polynomial order <= m-1
if p ~= 1 || p ~= size(Y, 1) || m ~= size(Y, 2)
    error('divdiff: input vectors must have the same dimension');
end
TBD = zeros(m, m);
TBD(:, 1) = Y';
for j = 2 : m
    for i=j : m
        TBD(i,j) = (TBD(i, j - 1) - TBD(i-1, j - 1));
    end
end
end
```

Κλήση συνάρτησης

```
>> x= [10 15 20 25 30 35]
```

```
x =
```

```
10 15 20 25 30 35
```

```
>> y= [19.97 21.51 22.47 23.52 24.65 25.89]
```

```
y =
```

```
19.970 21.510 22.470 23.520 24.650 25.890
```

```
>> Bdiff(x, y)
```

```
ans =
```

19.97000	0.00000	0.00000	0.00000	0.00000	0.00000
21.51000	1.54000	0.00000	0.00000	0.00000	0.00000
22.47000	0.96000	-0.58000	0.00000	0.00000	0.00000
23.52000	1.05000	0.09000	0.67000	0.00000	0.00000
24.65000	1.13000	0.08000	-0.01000	-0.68000	0.00000
25.89000	1.24000	0.11000	0.03000	0.04000	0.72000

Διαιρεμένες διαφορές

```
function TDD = divdiff(X, Y)
% Input::
%      X = [ x0 x1 .. xn ] - object vector
%      Y = [ y0 y1 .. yn ] - image vector
% Output:
%  TDD - table of divided differences
    [ p, m ] = size(X); % m points, polynomial order <= m-1
    if p ~= 1 || p ~= size(Y, 1) || m ~= size(Y, 2)
        error('divdiff: input vectors must have the same dimension');
    end
    TDD = zeros(m, m);
    TDD(:, 1) = Y';
    for j = 2 : m
        for i = 1 : (m - j + 1)
            TDD(i,j) = (TDD(i + 1, j - 1) - TDD(i, j - 1)) / (X(i + j - 1) - X(i));
        end
    end
end
```


Κλήση συνάρτησης

```
>> x=[1:8]
```

$$X =$$

1 2 3 4 5 6 7 8

>> $y = \log(x)$

$$y =$$

0.00000 0.69315 1.09861 1.38629 1.60944 1.79176 1.94591 2.07944

```
>> divdiff(x, y)
```

ans =

[illegible]