Description

Interpolation refers to the process of creating new data points given within the given set of data.The above MATLAB code computes the desired data point within the given range of discrete data sets using the formula given by Gauss.
The gaussian interpolation comes under the Central Difference Interpolation Formulae which differs from Newton's Forward interpolation formula formula.
Suppose we are given the following value of $y=f(x)$ for a set values of x:

X: x0 x1 x2 ………. xn
Y: y0 y1 y2 ………… yn
Then the process of finding the value of y corresponding to any value of x=xi between x0 and xn is called interpolation. Thus interpolation is the technique of estimating the value of a function for any intermediate value of the independent variable while the process of computing the value of the function outside the given range is called extrapolation. However, extrapolation is beyond the scope of this code.

Suppose that the function $y=f(x)$ is tabulated for the equally spaced values x = x0, x = x0+h , x = x0+2h ,……… x = x0+nh, giving y= y0 , y1, y2,……, yn. . To determine the value of f(x) or f'(x) for some intermediate values of x, the following three types of differences are found useful.
1. Forward Difference
2. Backward difference
3. Central Difference
The common Newton's forward formula belongs to the Forward difference category. However , the Gaussian forward formula formulated in the attached code belongs to the central difference method.
Gauss forward formula is derived from Newton's forward formula which is:
Newton's forward interpretation formula:

$Yp=y0+p.\Delta y0+ p(p-1)\Delta 2y0/(1.2) + p(p-1)(p-2)\Delta 3y0/(1.2.3)+\ldots.$
We have ,
$\Delta 2y0 – \Delta 2y-1=\Delta 3y-1$
i.e. $\Delta 2y0=\Delta 2y-1 + \Delta 3y-1$
Similarly,
$\Delta 3y0=\Delta 3y-1 + \Delta 4y-1$
$\Delta 4y0=\Delta 4y-1+\Delta 5y-1$ etc..
Also, $\Delta 3y-1-\Delta 3y-2=\Delta 4y-2$
I.e. $\Delta 3y-1=\Delta 3y-2+\Delta 4y-2$
Similarly $\Delta 4y-1=\Delta 4y-2+\Delta 5y-2$ etc..

Substituting for $\Delta 2y0$, $\Delta 3y0$, $\Delta 4y0$….. from (2),(3),(4)….. in (1), we get
$Yp=y0+ p\Delta y0 + (p(p-1).(\Delta 2y-1+\Delta 3y-1))/(1.2) + (p.(p-1).(p-2)(\Delta 3y-1+ \Delta 4y-1))/(1.2.3)$
$+ (p.(p-1)(p-2)(p-3)(\Delta 4y-1+\Delta 5y-1))/(1.2.3.4)+\ldots$…..
Hence $yp=y0 + p\Delta y0 + (p(p-1)\Delta 2y-1)/2!+ ((p+1)p(p-1)\Delta 3y-1)/3! + ((p+1)p(p-1)(p-2)\Delta 4y-2)/4!+\ldots$……… using (5)……….

Which is then called gauss's forward interpolation formula.

One more representation of this formula is in the central difference notation , this formula will be

$yp=y0+ p\delta y1/2 + (p(p-1)\delta 2y0)/2! + ((p+1)p(p-1)\delta 3y1/2)/3! + ((p+1)p(p-1)(p-2)\delta 4y0)/4!+\ldots$……

These interpolation formulae are applicable for interpretation near the beginning and end of tabulated values. However, the gaussian forward formula are best suited for interpolation near the middle of the table.
The coefficients in the central difference formula such as that of gauss are smaller and converge faster than those in Newton's formulae. Therefore, as much, whenever possible, central difference formulae should be used in preference to Newton's formulae.
The above code starts with asking the no. of data. The input data is denoted by x.
The difference between the two consecutive values of x is denoted by h.
One more parameter used in this formula is p, which is computed by the following formula
$p=(x-x0)/h$
The corresponding value of y is also asked from the user.
After entering this values, the code computes the parameters h and p automatically and displays the result on calling.
The final desired answer is stored in a variable named ans and is displayed on calling in MATLAB window.
An example :-

```
No. of variables =5
The x values are :- 21 , 25 , 29 , 33 ,37 and the corresponding values of y are given by 18.478 , 17.8144 ,
17.8144 , 16.3432 and 15.5154.
        x p yp Δyp Δ2yp Δ3yp Δ4yp
    21 -2 18.478 -0.6564 -0.0510 -0.0074 -0.0022
25 -1 17.8144 -0.7074 -0.0564 -0.0076
29 0 17.8144 -0.7638 -0.0640
33 1 16.3432 --0.8278
37 2 15.5154
On running the code for the desired value of y30, we get the following values :-
h=4
p=0.25
ans=16.9216
```

```
N=input('Enter the number of variables ');

for i=1:N
    x(i)=0;
end

for i=1:N
    x(i)= input('Enter the values of x ');
end

x_des=input(' Enter the desired value of y ');

fin=N;
n=fin;

h=x(2)-x(1);



if rem(N,2)==0
    N=N/2+1;
else
    N=N/2+0.5;
end

p=(x_des-x(N))/h;

for i=1:fin
    yp(i)=0;
end


for i=1:fin
    y(i)= input('Enter the corresponding values of y
with respect to x ');
end


for i= 1:n
    for j=1:(n-1)
        y_arr(i,j)=0;
    end
end

for i= 1:n
    for j=1:(n)
```

```
        k_arr(i,j)=0;
    end
end


for i=1:n
    k_arr(i,1)=y(i);
end


i=1;
b=2;
while i<(fin+1)            %pending

j=2;
c=1;

while j~=(n+1)                %y1

    k_arr(c,b)=(k_arr(j,i)-k_arr(j-1,i));
    j=j+1;
    c=c+1;


end
i=i+1;
n=n-1;
b=b+1;

end
fact(1)=1;
fact(2)=p;
fact(3)=(p*(p-1));
fact(4)=((p+1)*p*(p-1));
init=fact(4);
er=p-2;
if(fin>4)
    inc=0;
for i=5:fin
    fact(i)=init*er;
    init=fact(i);
    er=er-1;
end
end
```

```
for i=1:fin
    perm(i)=factorial(i);
end

ans=0;
c=1;
bck=N;
fig=N;
for i=1:(fin-1)
    ans=ans+(fact(i+1)*k_arr(fig,(i+1)))/perm(i);
    c=c+1;
 if c==2
     fig=fig-1;
     c=0;
 end
end
ans=ans+k_arr(bck,1)
```