3. SIMULATIONS

Implementation of design approach is simple, i.e., the theory so far discussed in the last chapter has been tested through MATLAB codes, and their complete listing is given in the list at the end of the report.

Initially for simulation No.1 we have to run the function FindC.m, which calculates the average and detail coefficients. It is advisable to use MATLAB 5.2 and above, at the matlab command prompt write [CX,CY,DX,DY] = FindC(j); where CX,CY,DX,DY are average coefficients along "x" and "y" axes and detailed coefficients along "x" and "y" axes respectively while "j" is the level upto which the decomposition is required. So at "j" write 1,2,3,..., etc., suppose we enter [CX,CY,DX,DY] = FindC(4) then the number of control points for the curve will be: $2^j+3 = 2^4+3 = 19$. Matlab will ask to enter the "x" and "y" axes ranges by the user, lets say, "x" and "y" axes ranges are 10, then it will prompt the user that 19 points have to be entered. Pressing the space bar will open an input screen having "x" and "y" axes with the specified ranges and a mouse pointer waiting for control points to be entered. The user could enter 19 points any where within the xy-plane. Pressing a key after entering 19 control points will make matlab to compute the CX,CY,DX,DY coefficients. It calculates the coefficients for all levels from the highest to lowest, i.e., 4,3,2,1,0 (five levels) and arrange them in CX,CY,DX,DY four separate matrices and displays them separately. In each CX and CYmatrices the first row has level "0" coefficients and the last row has the highest level coefficients. For DX and DY matrices the first row has level "0" coefficients and the last row has one-less than the highest level coefficients.

Simulation No.2 is for observing the smoothing process. Since we have already calculated the CX,CY,DX,DY matrices so they are in the memory. We have to run the Cplot.m function to see the smoothing process. At the MATLAB prompt write Cplot(CX,CY,j), for our example replace "j" with "4" and enter. This will make five separate subplot screens four at a time, starting from the highest resolution first then towards the lowest zero level resolution through lower levels. Description of level and number of control points are shown for each level. A slider control is provided for each subplot level except for the highest level. Moving this slider will allow fractional level smoothing between two adjacent integer levels. The control points are shown with red circles and the overall run of the curve is in blue color. Pressing the space bar will open another window if there are further levels to be displayed, having the same properties as described above. A fast computer may produce morphing effect when sliders are made to move continuously.

Simulation No.3 is the editing of the sweep of the curve. We can use the esweep.m function. At the MATLAB command prompt for our example enter esweep(CX,CY,4), where "4" is the level and CX and CY are average coefficients already in the memory. After entering the above information it opens a windows having four subplot screens each having adjacent edit buttons except the highest level subplot screen. On pressing an edit button the relevant subplot screen turns in edit mode and the mouse pointer waiting for control point selection. Select any red circled average coefficient by pressing the right mouse button after getting closure to the control point, this will change the red circle into high intensity greenish white which confirms the selection. Now any where within the subplot screen, select a new control point by clicking the right mouse button this will replace the old control point with this new point and will reconstruct all the higher levels again with the run of the curve as red.

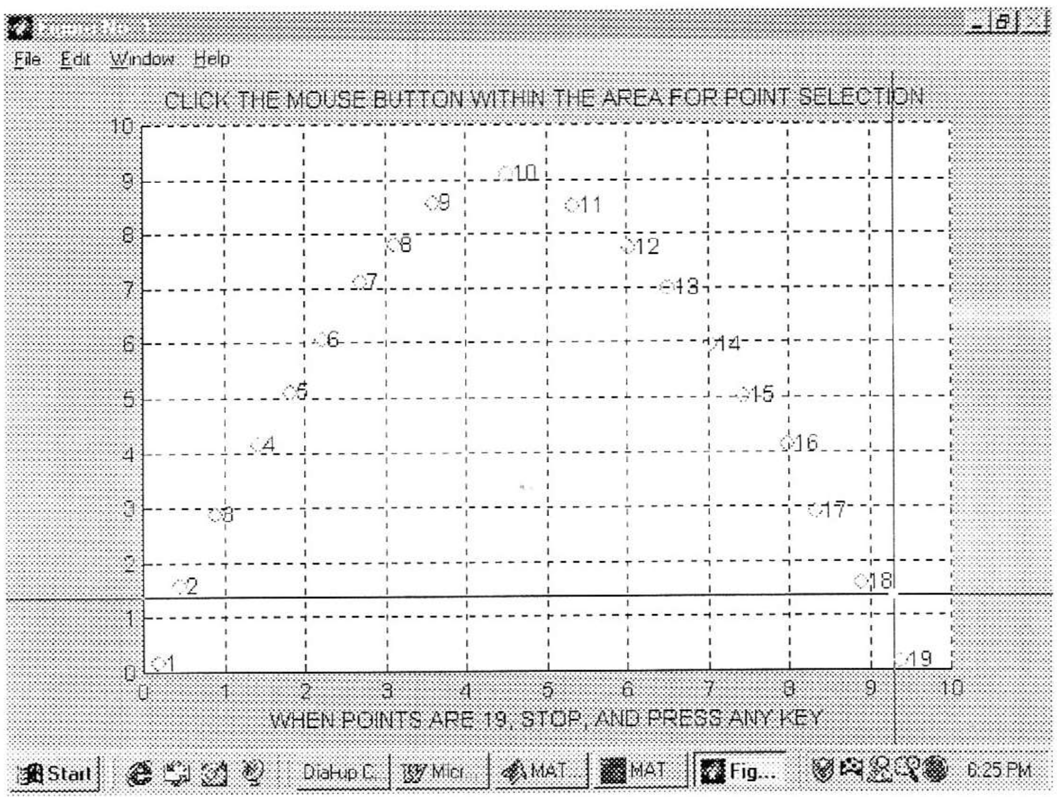
Simulation No.4 edits the detailed coefficients at any level except the highest and then it reconstructs for the one next higher level. To carry out this task use the echaracter.m function and as input arguments write at the MATLAB command prompt echaracter(CX,CY,DX,DY,4) for our example. This will open a window having four subplot screens at a time on the console with adjacent edit button with all the levels except the highest level. On pressing an edit button it turns that subplot screen into editing mode and a message informs that editing will be carried out in the matlab command/text window. So switch in the command/text window and edit the DX and DY matrices for a level. The software will prompt the value of each existng detailed coefficients and ask the new coefficients to be entered one by one. On entering all the new detailed coefficients the reconstruction process is done for the one next higher level and is displayed.

The following observations have been made from the results of "curve editing": When a curve is continuously made smooth through fractional level editing, i.e., via from slider control between two adjacent integer levels, then an animation effect can be obtained. If the slider control steps are made smaller and smoothing is carried out at higher processing rates then morphing effect may also be obtained, see Simulation No.2.

Editing the sweep of a curve at lower levels of smoothing "j" affects larger portions of the high-resolution curve, see Simulation No.3a, and a small effect is obtained while edited at higher levels. It is due to the fact that at lower levels the coefficients are related to more coefficients in the higher levels as compared to coefficients being at higher levels and vice versa. At the lowest level, when $j = 0$, the entire curve is affected. At the highest level, when $j = J$, only the narrow portion is affected, see Simulation No.3b.

Editing the character of a curve requires some precautions, firstly, the magnitudes of the detail coefficients, they must be comparable with the existing magnitudes. If the existing magnitude is in the order 10^{-4} then the new coefficients must not be of an order like 10^{2}, this may suppress the average coefficient information in the curve; secondly, the sign of the detail coefficients also contribute a lot as the sign is in accordance with the x and y axes coordinate system. See Simulation No.4, all detail coefficients are entered with same magnitudes but with opposite signs.

The simulations in MATLAB are illustrated from next page.



Simulation No:1. GUI allows a user to enter control points of a curve on an XY coordinate system, MATLAB function FindC.m.

YOU HAVE TO ENTER "19" POINTS

P R E S S   A N Y   K E Y   T O   C O N T I N U E
End of data entry

CX =

Columns 1 through 7

0.7232	2.2447	6.4862	8.1121	0	0	0
0.7708	1.3889	4.5081	7.2040	8.1596	0	0
0.5298	1.4136	1.9388	4.4832	6.6314	7.4673	8.3298
0.3959	1.2161	1.2579	2.3901	3.0780	4.3742	5.7919
0.3917	0.8295	1.1751	1.4516	1.7742	2.3502	2.7189

Columns 8 through 14

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
6.2322	7.5440	7.7105	8.4101	0	0	0
3.2258	3.6406	4.3779	5.1843	5.5991	6.0138	6.3364

Columns 15 through 19

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
6.9355	7.3502	7.7189	8.0415	8.4101

Simulation No.1. Function FindC.m calculates out average CX coefficients along x-direction.

CY =

Columns 1 through 7

0.3778	10.9151	10.5235	0.0792	0	0	0
1.4893	3.4234	14.0538	3.0783	1.1907	0	0
0.6406	3.8527	4.6351	11.1642	4.5281	3.2795	0.5305
0.4984	2.5722	3.5545	5.8339	7.3204	10.0591	7.4066
0.5246	1.5082	2.8197	3.6721	4.6885	5.7049	6.5902

Columns 8 through 14

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
5.4726	3.2633	2.1410	0.4156	0	0	0
7.5738	8.4918	9.5738	8.6230	7.5410	6.4262	5.4426

Columns 15 through 19

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
4.3607	3.3443	2.3934	1.3115	0.3934

DX =

Columns 1 through 7

0.0476	0	0	0	0	0	0
-0.0423	0.0299	0	0	0	0	0
-0.0000	-0.0028	0.0040	0.0000	0	0	0
-0.0000	-0.0000	0.0000	-0.0000	0.0000	-0.0000	0.0000

Column 8

0
0
0
0.0000

Simulation No.1. Function FindC.m calculates out average coefficients CY along y-direction, detail coefficients DX along x-direction.

DY =

Columns 1 through 7

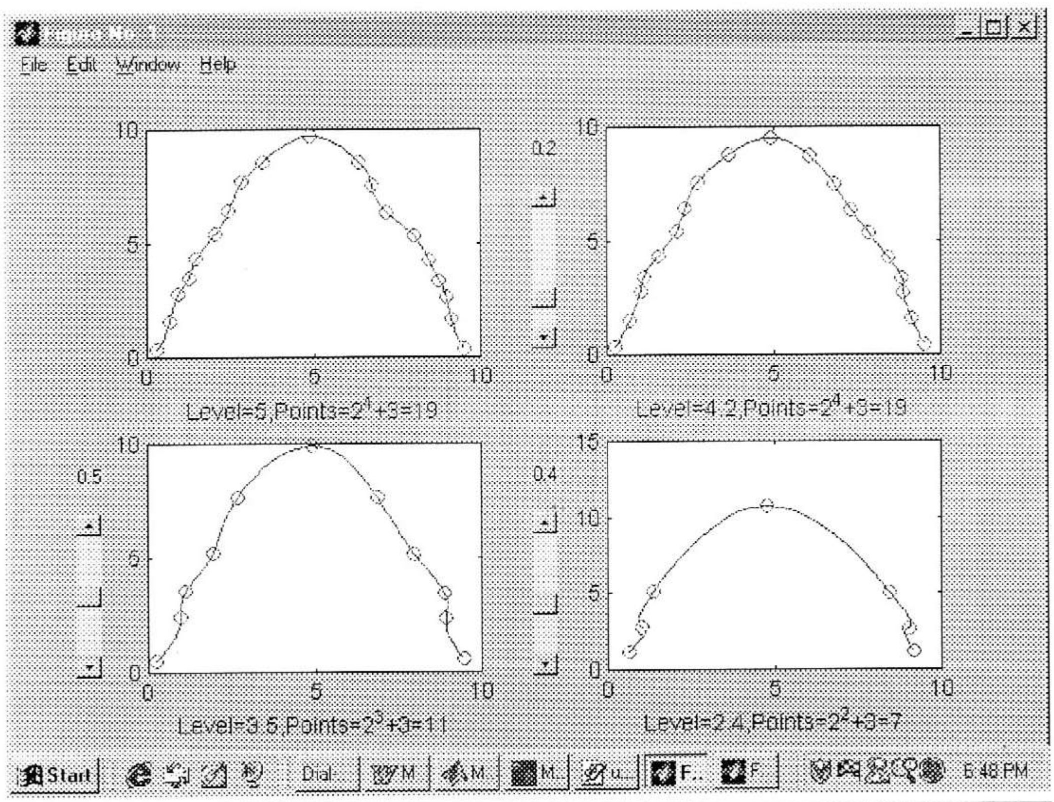
1.1115	0	0	0	0	0	0
-0.1489	-0.1158	0	0	0	0	0
-0.0000	-0.0067	-0.0039	-0.0000	0	0	0
0.0000	-0.0000	0.0000	-0.0000	-0.0000	0.0000	-0.0000

Column 8

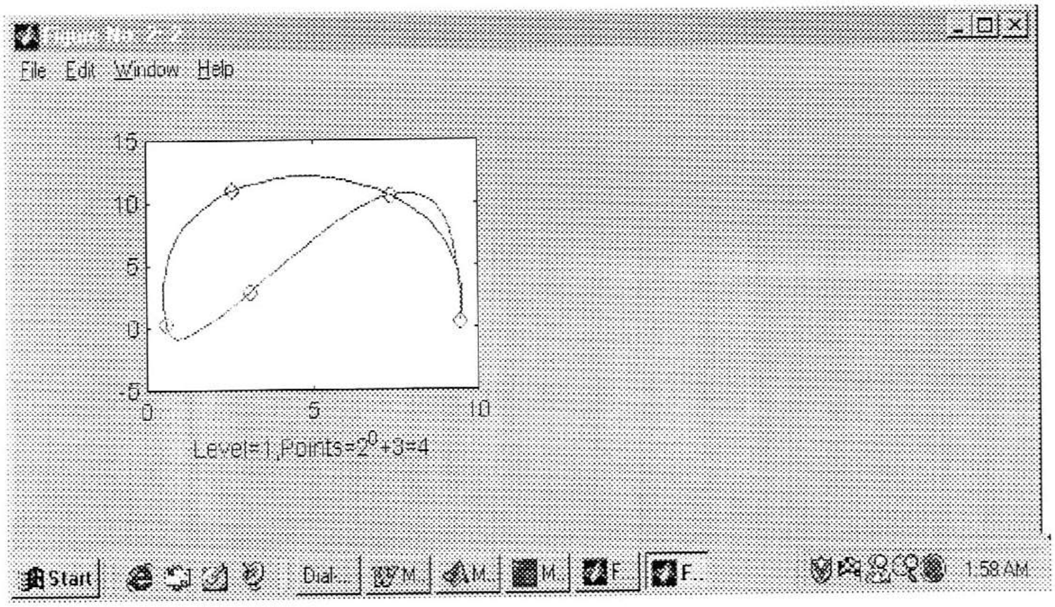
0
0
0
-0.0000

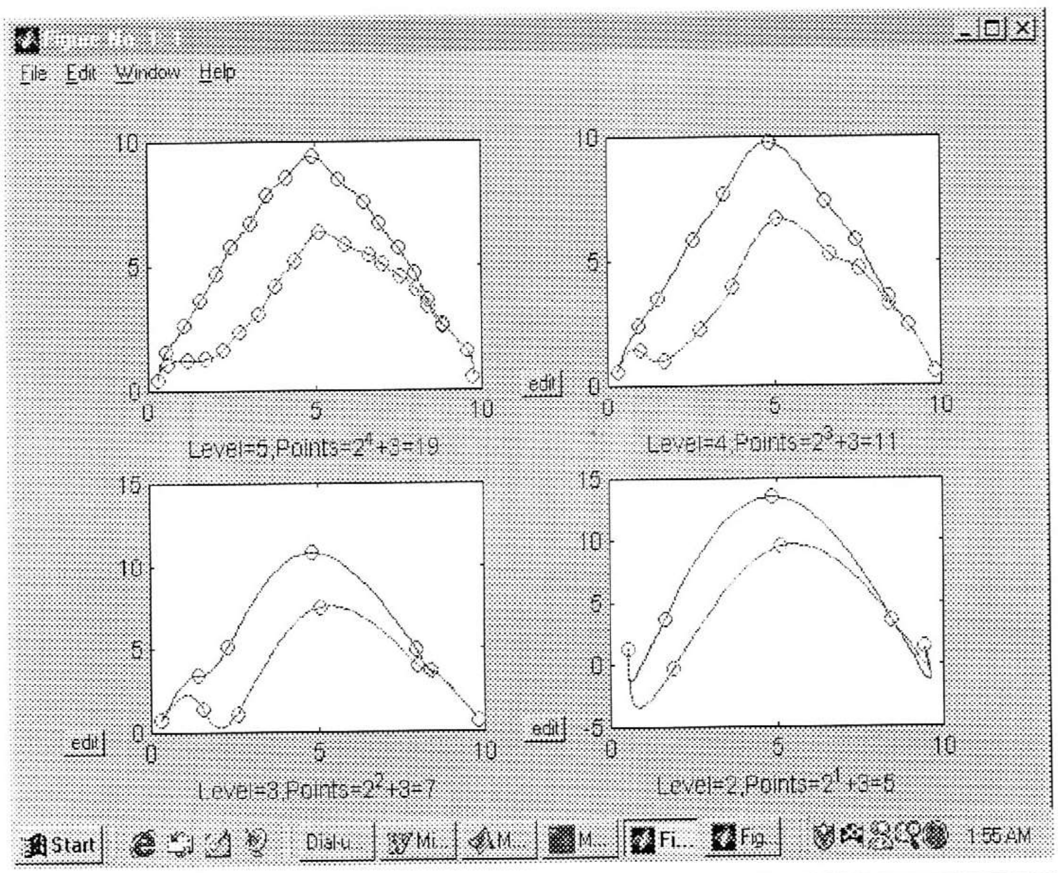
»

Simulation No.1. Function FindC.m calculated out detail coefficients DY Along y-direction.

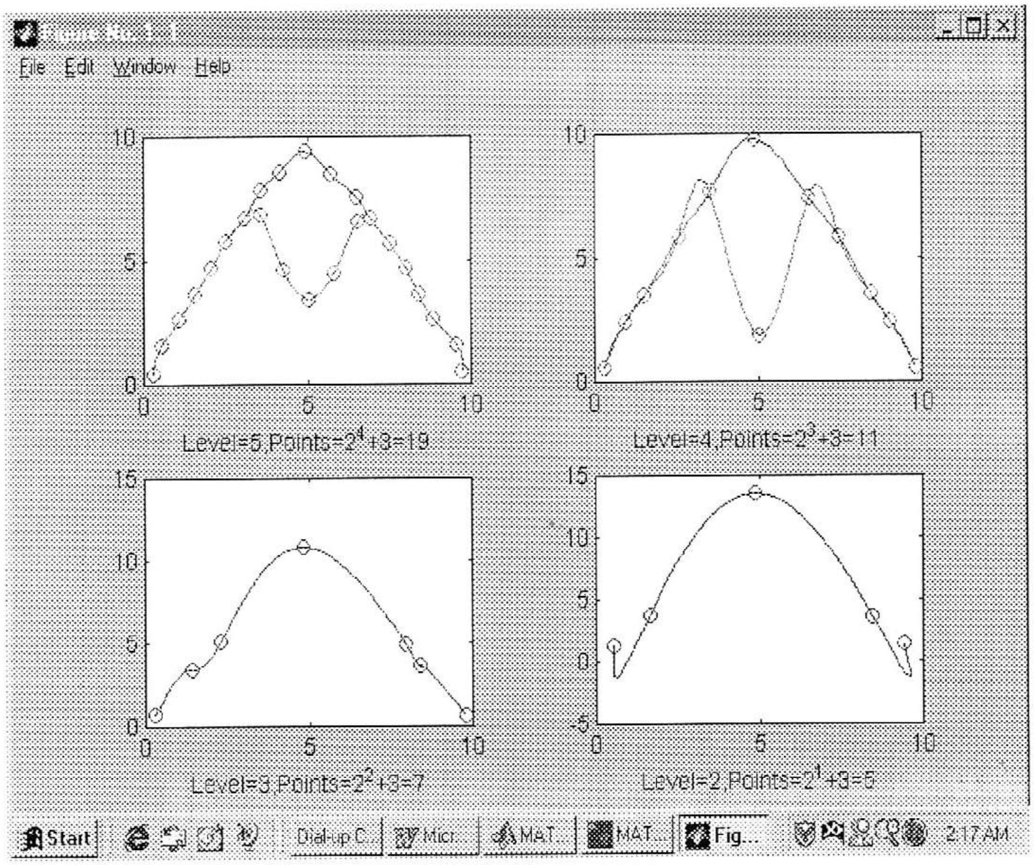


Simulation No.2. Function Cplot.m shows the given curve decomposed through various levels. Note that slider controls allow both integer and fractional level smoothing.

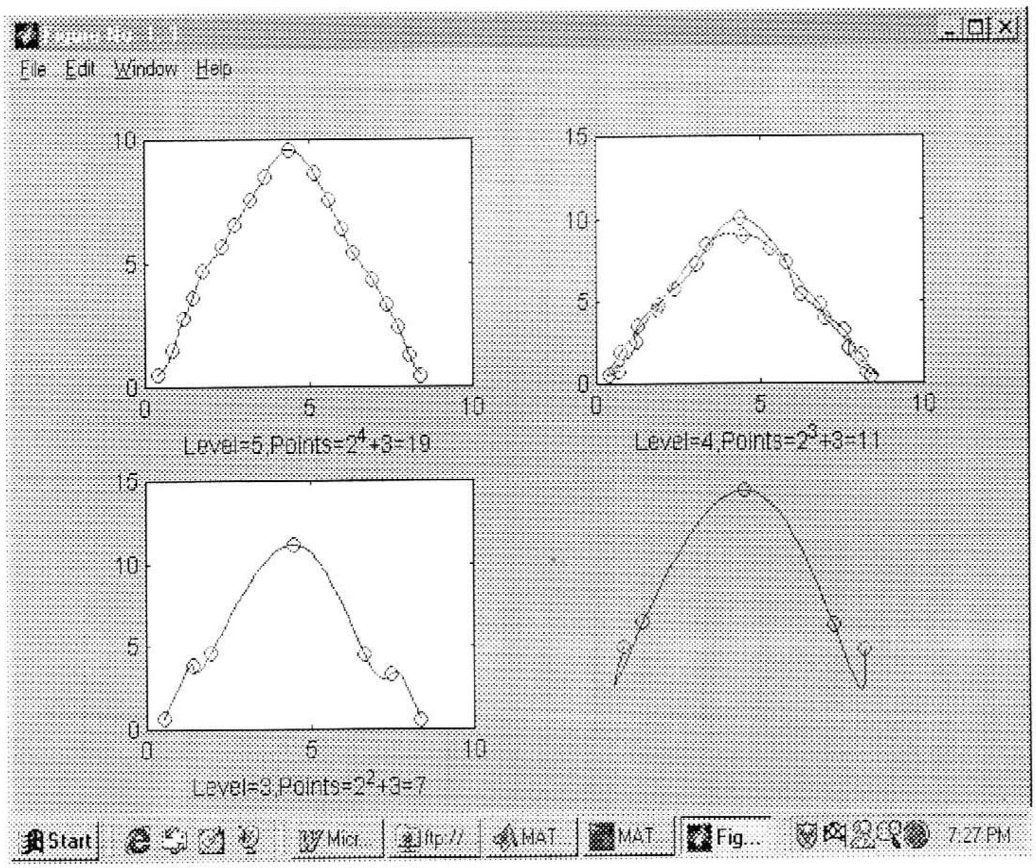




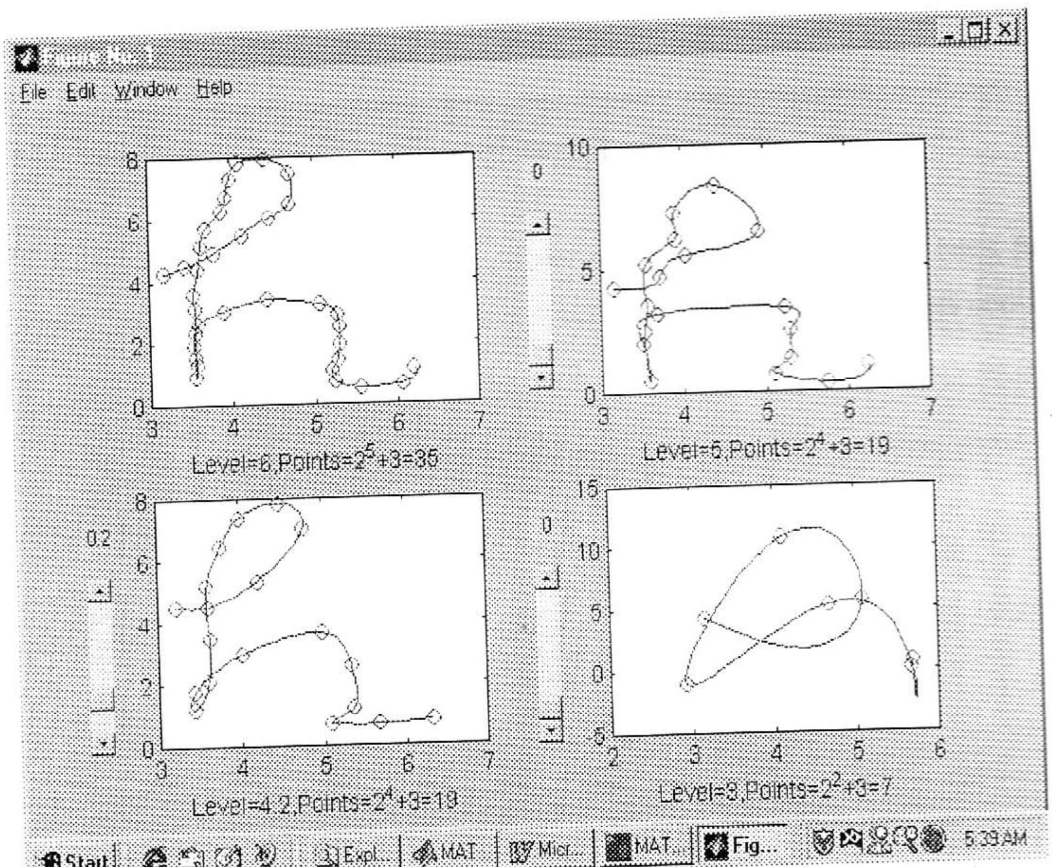
Simulation No.3a. Function esweep.m smoothens the given curve. Note that editing at a lower level makes broader changes at higher levels.



Simulation No.3b. Function esweep.m permits to edit the CX and CY coefficients dynamically. Note that editing a coefficient at higher level makes narrow changes in the higher level.



Simulation No.4. Function echaracter.m allows changes in the detail coefficients. Note how curve path changes when there is only difference of sign in the new detail coefficients.



Simulation No.5. A constant zero thickness font is made to decompose through various levels and the derivatives are quite meaningful.

CONCLUSIONS AND FUTURE WORK

The following conclusions are drawn during design, implementation and simulation of curve editing:

- For all levels, detail coefficients provides a better control for the whole run of the curve.
- A detailed coefficient library may be more useful to visualise the effect of changes in the character of a curve.
- The loss of information in the reconstruction process is found almost insignificant in computer graphics applications.
- A low level polynomial, i.e., linear or quadratic selected as a basis function would not be a good illustrative function as at their control points non-smooth edges are formed.

Due to the flexibility provided by the multiresolution analysis there can be numerous areas for future work, including:

- Multiresolution editing can be extended to surfaces [4,13,15] and volumes [3] by using tensor products of scaling functions and wavelets.
- This extension would allow the techniques to be applied more readily to font design [2], among other applications. In order to design different fonts of zero thickness throughout we can observe various font styles in each level through the process of smooth. We tested different English alphabets and their derivatives are found giving meaningful appearance. If constant/varying thickness and orientation properties are incorporated in coefficients of curve's control points then a comprehensive font-designing tool may be constructed, see simulation No.5.

- For illustrations, it is useful to associate other properties with curves, such as color, thickness, texture and transparency [2]. These quantities may be considered extra dimensions in the data associated with each control point.