```
PRINT T
2.60417E-03
```

Program breaks at the following line after an execute. Print value of third term. Both incorrect when checked by hand.

```
42 LET T = X
44 LET N = 2
UNBREAK
```

This leads us to question initial values of T and N. New lines 42 and 44 compiled and replace old versions.

```
READY
LIST
```

Call for a listing of the program as it now stands.

```
EXAMPLE
10 READ E
40 FOR X = 0 TO 6.5 STEP 0.5
42 LET T = X
44 LET N = 2
50 LET S = X
60 LET T = -T*X*X/(N*(N + 1))
70 LET N = N + 2
80 IF ABS(T) > E THEN 110
90 PRINT X, S
100 GOTO 130
110 LET S = S + T
120 GOTO 60
130 NEXT X
140 DATA 0.00005
150 END
READY

CONTINUE 40

0       0
.5      .479427
1       .841468
1.5     .997497
2       .909296
2.5     .598449
3       .141131
3.5     -.350788
4       -.756849
4.5     -.97751
5       -.958933
5.5     -.705536
6       -.279387
6.5     .215107
END OF PROGRAM
```

Conclusions

In the above example, some, but by no means all of the potential of a dynamic debugging system is exhibited. Most arguments in favour of time-sharing place emphasis on quick turn-around for testing, access to a filing system, and interaction with the running program by supplying data dynamically to control its action. However, in the author's opinion, the greatest advantages will be realised when systems provide debugging aids similar to those described here for all languages available at the terminal.

Acknowledgements

The author is indebted to his colleague W. Freeman for his suggestions and comments on the draft of this paper.

References

BASIC Reference Manual. Dartmouth College, Hanover, NH, USA.

Bull, G. M., and Freeman, W. (1971). BASIC—A preliminary specification. Hatfield Polytechnic Comp. Sc. Tech. Memo No. 1.

Bull, G. M. (1971). BASIC—its growth and development, *J. Instn. Comp. Sc.*, Vol. 2, No. 3, p. 51.

Barron, D. W. (1969). A note on program debugging in an on-line environment, *The Computer Journal*, Vol. 12, No. 1, p. 104.

Barron, D. W. (1971). Approaches to conversational FORTRAN, *The Computer Journal*, Vol. 14, No. 2, p. 123.

Barron, D. W. (1971). Programming in Wonderland, *The Computer Bulletin*, Vol. 15, No. 4, p. 153.

Correspondence

To the Editor
The Computer Journal

Sir,

With reference to your article 'Step size adjustment at discontinuities for fourth order Runge-Kutta Methods', by P. G. O'Regan, published in this *Journal*, Volume 13, Number 4, November 1970, I should like to point out that equation (21) appears to be incorrect in the fifth term of the right-hand side which should, I think, be $A^5(14B^4 - 21B^2C + 3C^2)$ instead of $A^5(14B^4 - 21B^2C - 3C^2)$. The last column of Table 1 of the article was evidently computed using the erroneous equation.

Table 1 has been recomputed to a precision of 20 significant digits using the correct equation and the results are reproduced below. The answers for Newton's iteration formula are quoted to 9 decimal places, for which two iterations were necessary. The revised table shows even more clearly than the original that Newton's iteration formula (23) is more accurate than (21).

Yours faithfully,
E. Whiteley (Miss)

Table 1 $h = 0{\cdot}1$

α_t	$10^6(\alpha_t - \alpha_n)$	$10^6(\alpha_t - \alpha_{iii})$	$10^6(\alpha_t - \alpha_{iv})$	$10^6(\alpha_t - \alpha_v)$
0	0	0	0	0
0·1	−0·202	− 0·301	−0·201	− 0·202
0·2	−0·655	− 2·249	−0·609	− 0·657
0·3	−1·170	− 9·336	−0·808	− 1·187
0·4	−1·606	− 27·730	−0·055	− 1·703
0·5	−1·871	− 66·426	2·948	− 2·252
0·6	−1·915	− 137·403	10·292	− 3·081
0·7	−1·729	− 255·782	25·130	− 4·741
0·8	−1·340	− 439·996	51·967	− 8·214
0·9	−0·809	− 711·955	96·976	−15·080
1·0	−0·228	−1097·231	168·339	−27·727

Structures Department
Royal Aircraft Establishment
Farnborough
Hampshire

13 August 1971