APIC Studies in Data Processing No. 19

INTRODUCTION TO THE GRAPHICAL KERNEL SYSTEM -G.K.S.-

F.R.A.Hopgood D.A.Duce

J.R.Gallop D.C.Sutcliffe

A.P.I.C. Studies in Data Processing
No. 19

Introduction to the Graphical Kernel System (GKS)

F. R. A. HOPGOOD
D. A. DUCE
J. R. GALLOP
D. C. SUTCLIFFE

Computing Division, Rutherford Appleton Laboratory, Didcot, UK

1983

ACADEMIC PRESS, INC.
Harcourt Brace Jovanovich, Publishers
London Orlando San Diego
New York Austin Montreal Sydney
Tokyo Toronto

ACADEMIC PRESS INC. (LONDON) LTD.
24/28 Oval Road
London NW1

United States Edition published by
ACADEMIC PRESS, INC.
Orlando, Florida 32887

British Library Cataloguing in Publication Data
Introduction to the Graphical Kernel System (GKS) —
(Automatic Programming Information Centre studies
in data processing ISSN 0067-2483; 19)
1. Computer graphics
I. Hopgood, F. R. A. II. Series
001.64'43 T385

ISBN 0-12-355570-1

LCCCN 82-074141

PRINTED IN THE UNITED STATES OF AMERICA

85 86 87 88 9 8 7 6 5

Introduction to the Graphical Kernel System (GKS)

Preface

Standards in computer graphics are long overdue. Whereas *de facto* standards in programming languages were common very early on (FORTRAN and ALGOL 60) and international standards soon followed, there has been a long period of graphics history where, at best, regional de facto standards have existed and no international standards have evolved.

Now, after some 6 years work by a highly dedicated international group of graphics experts, this trend has been broken, with the acceptance of the Graphical Kernel System (GKS) for processing as a Draft International Standard by ISO in 1982.

GKS is a graphics system which allows programs to support a wide variety of graphics devices. It is defined independently of programming languages.

This book aims to provide the application programmer with a good understanding of the principles behind GKS and to serve subsequently as an informal manual for GKS. No knowledge of GKS is assumed, but the reader is expected to have a good understanding of programming and at least a rudimentary knowledge of computer graphics. The book is arranged in two main parts. The background of GKS is described and its essential concepts are introduced in the first part whereas those features more likely to be required by the specialist graphics programmer are described in the second. The chapters in Part I, which introduce the essential concepts, are intended to be read in sequence. However, the chapters in Part II may be read more or less independently of each other, although they assume knowledge of the whole of Part I.

Examples are expressed in a dialect of FORTRAN 77 described in the introductory section on Notation. As an aid to understanding, names greater than 6 characters are allowed. The correspondence between the names actually used and those in the FORTRAN 77 language binding for GKS is given in an appendix.

Part I begins with a chapter which outlines the historical background of graphics standards and the emergence of GKS. In Chapters 2 and 3, output primitives of GKS and their attributes and the coordinate systems, in which they are specified, are described. Chapter 4 describes how pictures may be divided into segments and how these segments may be manipulated. The segment attributes are introduced. GKS is not only concerned with graphical output but also with graphical input. Its powerful input facilities are covered in Chapters 5 and 6. Chapter 7 concludes Part I with a description of the workstation concept which is the central concept in GKS promoting program portability.

The GKS environment and the more advanced facilities are discussed in Part II. Chapter 8 describes the GKS environment, which includes initialisation of GKS, the GKS data structures, the level structure and error handling. Chapter 9 describes the control of input devices. In Chapter 10, more advanced segmentation facilities (system wide segment storage) are considered and Chapter 11 describes the GKS metafile, a means of transporting graphical information between graphics installations. Chapters 12 and 13 describe the output primitives not covered in Chapter 2 and say more about attribute handling.

The authors of the book were the editors of the GKS document. Although that document is quite readable, we felt there was a need for a more descriptive introduction to GKS illustrated with many examples. This book is our answer to that need. We believe it will be useful to all application programmers with an interest in computer graphics.

Finally, we would like to thank all those who have participated, in any way, in the standardization process, which has lasted a number of years at national and international levels. In particular, we would like to thank Howard Watkins for his detailed comments on the book. Our thanks must also go to wives, families and friends, who have borne with us during both the standardization process and the writing of this book.

Maundy Thursday 1983

In no time at all, this book needs to be reprinted. Since the book went to press, GKS has moved forward although its story is not yet complete. GKS 7.2 (registered as ISO/DIS 7942) is now out for letter ballot to be approved as a Draft International Standard. The ballot closes on 23 December 1983 and allows for editorial comments (including clarifications) to accompany positive votes. The document incorporating the necessary changes will be forwarded to become the International Standard.

The ISO graphics working group (see Chapter 1) met in Gananoque, Canada in September 1983 and agreed a single language binding for FORTRAN 77. The agreed language binding differs in one or two respects from the one used in this book (listed in Appendix B). The differences are outlined in the revised introductory section on Notation. It has now been decided to publish a separate standard to cover language bindings for GKS and although, formally, this is at an early stage, the language binding for FORTRAN 77 is not expected to change.

All Saints' Day 1983

David A. Duce
Julian R. Gallop
Dale C. Sutcliffe
F. Robert A. Hopgood

Notation

It is inevitable, in a technical work such as this, that a number of abbreviations are used. The more important ones are defined as they are introduced but, for completeness, all the abbreviations are defined in Appendix A.

GKS, itself, is defined independently of a programming language. Before it can be used from a particular language, a *language binding* must be defined for that language. In this book, GKS is presented in terms of a FORTRAN 77 language binding, listed in Appendix B. Since the first printing, a revised language binding for FORTRAN 77 has been agreed. The few differences are summarized at the end of this section.

Since a language binding must obey the conventions of a language, all the subroutine names in the FORTRAN 77 language binding are restricted to be at most six characters. To make this book more readable, each subroutine name has been replaced by the full function name from the GKS document (ISO/DIS 7942). Both names are listed in Appendix B. For example, the name of the subroutine for line drawing is GPL but in this book we use the GKS name POLYLINE.

The examples in this book are given in a dialect of FORTRAN 77. In the interests of illustrating the subject matter, some liberties have been taken. In particular, no restrictions are placed on the length of identifiers and the word 'CALL' is omitted from CALL statements. Thus, using the example above, in FORTRAN 77 we would use:

```
CALL GPL(N, X, Y)
```

to draw a line but here this is written as:

```
POLYLINE(N, X, Y)
```

INTEGER and REAL values are mixed indiscriminately (for example, where a subroutine requires a REAL parameter, an INTEGER or REAL actual parameter may be used). Parameters specifying one of a number of options, which in PASCAL would be of enumeration type, are written as variable names, with the assumption that the appropriate settings have been made elsewhere. For example, the GKS function to specify the CLIP (rather than NOCLIP) option for the clipping indicator will be written as:

```
SET CLIPPING INDICATOR(CLIP)
```

The FORTRAN 77 language binding internationally agreed at Gananoque in September 1983 differs from the one used in this book as follows. Some of the integer values corresponding to the enumeration type parameters referred to above have been changed. Table 3 of Appendix B has been updated to reflect the new values. Names have been defined for these values and are listed in Table 4 at the end of Appendix B. These should be made available, by means of PARAMETER or DATA statements, for inclusion in application programs (in an implementation dependent manner). Now, to set the clipping indicator to CLIP requires:

```
CALL GSCLIP(GCLIP)
```

The set of subroutine names has been improved resulting in the following changes (in the order of Table 2 of Appendix B):

Book name	New name	Book name	New name	Book name	New name
GAWE	GWAIT	GSDFS	GSDS	GSWKVW	GSWKVP
GFLDE	GFLUSH	GSMPC	GSMPK	GSWKW	GSWKWN
GGTPC	GGTPK	GSPCID	GSPKID	GUPDRC	GPREC
GINPC	GINPK	GSPCM	GSPKM	GUPDWK	GUWK
GMESS	GMSG	GSVW	GSVP	GUUDRC	GUREC
GRQPC	GRQPK	GSVWIP	GSVPIP		
GSCLIN	GSCLIP	GSW	GSWN		

The specification of two dimensional arrays as parameters has been generalized, to allow a part of a two dimensional array to be passed to a subroutine, by adding the size of the first dimension of the array to the parameter list. For example, the subroutine for CELL ARRAY is now:

```
SUBROUTINE GCA(PX, PY, QX, QY, DX, DY, DIMX, CLA)
```

In the input INITIALISE functions that pass data records with compulsory parameters (those for STROKE, VALUATOR and STRING), the compulsory parameters are now explicit parameters to the FORTRAN subroutines. For example, the subroutine for INITIALISE VALUATOR is:

```
SUBROUTINE GINVL(WKID, VLDNR, IVAL, PET,
  XMIN, XMAX, YMIN, YMAX, LOVAL, HIVAL, IL, DREC)
```

Data records in GKS functions are now mapped to arrays of CHARACTER*80 elements but the parameters defining the size of the arrays are unchanged. The parameter lists of the two language binding utilities PACK DATA RECORD and UNPACK DATA RECORD have been changed: the function identifier has been removed, only one REAL array is provided and there is now an error indicator to return errors. Thus, the subroutine for PACK DATA RECORD is now:

```
SUBROUTINE GPREC(IL, IA, RL, RA, CC, CA, IDIL, ERRIND, IDOL, DREC)
```

In order to simplify the parameter lists of inquiry subroutines, items which contain four or more components (for example, window limits) are returned in an array rather than individually. Thus, to inquire the state of a valuator device requires:

```
CALL GQVLS(WS, DV, MMX, ERRIND, OPMODE, EC, INITVAL, PE,
  EAREA, LOVAL, HIVAL, IMX, DREC)
```

Other changes affect details that are beyond the scope of this book.

Contents

1 Introduction

The tiny Dutch village of Steensel was the setting for the announcement in June 1982 that the Graphical Kernel System (GKS) would be adopted by the International Organizaton for Standardization (ISO) as a Draft International Standard. Why has it taken until 1982 for a Draft International Standard for computer graphics to emerge? After all, *de facto* standards in programming languages have existed for some twenty years and international standards soon followed. At first sight it might be thought that this was due to the relative infancy of computer graphics, but the origins of computer graphics can be traced back almost to the advent of digital computers.

On the MIT Whirlwind, dual 16 inch displays were available as early as 1951; plotters were in use by 1953 and at least one high speed microfilm recorder was available in 1958. Input devices emerged later, yet lightpens can be traced back to 1958 and the RAND tablet made its debut in 1964. Colour displays appeared in 1962 and by 1965 most of the hardware facilities which we now take for granted had appeared.

However, despite the existence of the technology, the number of display systems installed worldwide by 1964 is believed to be only about 100. Predictably these came from a number of different manufacturers and different graphics packages were written to capitalize on the features of each. There is a parallel here with the early history of programming languages. In that field, however, the availability of FORTRAN on a widely marketed range of IBM machines led to the emergence of a *de facto* standard.

Some standard approaches began to appear in computer graphics, accepted techniques in the use of stand-alone and satellite refresh display systems were emerging in the late 1960's and it is possible that had developments continued in this manner, a standard would have emerged. Likewise, patterns were apparent in the use of plotters and the GINO and GHOST packages began to be more widely used in the UK in the early 1970's.

1.1 THE CHANGING SCENE

Until the late 1960's, interactive graphics required an expensive refresh display and dedicated host computer. A total cost of $400,000 was not uncommon and thus such systems were only available to a few.

The advent of timesharing systems at about the same time that the storage tube display emerged (at a fraction of the cost, say $4,000 against $80,000 for a refresh display) had a dramatic impact. Interactive graphics was now possible for a large number of people. The storage tube did not allow changes to be made to the picture without completely redrawing it (which was unattractive because of the low bandwidth typically available to the mainframe) but at the same time allowed a large amount of information to be displayed flicker free. Consequently, users developed new and different techniques for graphics with this type of display.

The changes did not stop there. In recent years, the cost of raster displays has plummetted and so low cost raster displays have appeared as competitors to the storage tube. These displays, like storage tubes, do not flicker when storing a large amount of information but in addition provide selective erasure, and other attractive features such as colour and area fill.

One might ask whether these changes will continue and whether there will ever be a time for standardization in computer graphics. The cautious prophet would argue that such rapid changes in both hardware and patterns of usage are unlikely to be repeated in the near future. There are now extensive investments in software (and hardware) mitigating against rapid change. We are, though, seeing the advent of high powered, low cost, single user systems, which is ironic when one recalls that it all started with dedicated systems! New input devices (for example, voice) are also on the horizon. As more and more graphics devices contain embedded microprocessors, more facilities are being put into hardware. It is possible, though obviously undesirable, for widely differing facilities to be put into different devices. Standards will provide guidance on what facilities should be performed in devices and will help to guard against this unproductive diversity.

1.2 SEILLAC I

In August 1974, at an IFIP WG5.2 meeting in Malmo, Sweden, Richard Guedj (France) was asked to initiate an active programme directed towards establishing standards for computer graphics. At a meeting in Bellinglise, it was agreed to organize what later turned out to be a seminal event in computer graphics standardization, the Seillac I workshop at Château de Seillac in the Loire Valley, France. Held in May 1976, the workshop was attended by a number of experts from the computer graphics field (Bob Hopgood was one of the UK delegates) and aimed to reveal the underlying concepts of computer graphics which earlier discussions had shown to be ill-understood.

Topics studied at Seillac ranged over the reasons for standardization as well as the scope and requirements of a standard. There was agreement that both output and input should be included. Standardization of the former was considered easy. However, as subsequent events showed, even the standardization of the 'easy' can generate much debate!

The requirements for a standard received much attention. To agree that the standard should serve the areas of, for example, cartography, schematics, engineering drawing and animation was easy, but it was more difficult to decide whether image processing and high quality typesetting should be included. A standard should be in line with current practice and should answer the needs of the user community. It should not be in conflict with other standards in, say, character sets, and programming languages. A standard is only likely to gain acceptance if its design reflects a high level of expertise.

The graphics system must be at the right level. It should not include features specific to single applications, but at the same time must not be so low as to be device dependent. By this time, it was generally agreed that any standard would specify a set of virtual input and output functions which would be realized in terms of the functions of actual devices.

One of the major debates revolved around both the concept of current position and also its behaviour with respect to transformations. One of the problems was that existing packages did not distinguish between transformations for viewing the picture and those used for constructing the picture out of smaller items (referred to as modelling). Seillac I resolved that there should be a clear distinction between these two types of transformation and that an initial core, or kernel, graphics system should be designed which would only use transformations for viewing a previously constructed picture.

Originally there had been no intention to publish the proceedings of Seillac I, but at the request of IFIP, the working papers were edited some two years later and subsequently published [1]. The Seillac I volume is, therefore, not a polished document, but is invaluable in revealing the seeds from which future activities and ideas grew.

1.3 DEVELOPMENTS

The graphics experts at Seillac I included representatives of the USA and West Germany. Some of those from the USA were members of the Graphic Standards Planning Committee (GSPC), formed two years earlier under the auspices of the ACM Special Interest Group on Computer Graphics (SIGGRAPH). Previously, progress had been slow but Seillac I generated enthusiasm which had led GSPC to work towards the specification of a core graphics system to fulfill one of the Seillac goals. After a large amount of work by GSPC, a first public draft of a core graphics system (often referred to as the Core) was published in 1977 [2]. A whole issue of ACM Computing Surveys in 1978 [3] was devoted to describing the GSPC Core, the major issues that had to be resolved, and examples of use of the GSPC Core. A further version of the Core was published in 1979 [4] which included some raster extensions. Whilst raster graphics had been dismissed in the early version, as being different from vector graphics and only available to a few, the dramatic drop in cost meant that raster graphics had to be considered in the

1979 proposal. The Core is a full 3D graphics system and a number of implementations have been produced in the USA.

Inspired by Seillac I, members of the West German standards organization (DIN) had also been active in defining a core graphics system. The most obvious difference between the German Graphical Kernel System (GKS) and the GSPC Core was that the former was only a 2D system, and, consequently, was significantly smaller.

Meanwhile, a proposal had been made in 1976 to ISO by the Standards Committee of the British Computer Society that the British GINO-F graphics package should become an international standard. At the time there was not a specific working group to deal with computer graphics, and so the programming language subcommittee ISO/TC97/SC5, within whose remit computer graphics lay, organized a working party in London in February 1977. The meeting resolved that no existing graphics package would be a suitable candidate for a graphics standard. It also recommended that a working group of SC5 should be established to deal with standardization of computer graphics and that the specification of a core graphics system should be an early target.

What was intended to be the first meeting of the new working group (ISO/TC97/SC5/WG2 - Graphics) was held in Toronto in August 1977. However, a procedural point actually prevented it being a formal meeting. Whilst the focal point of the discussion was the GSPC Core report of 1977, it became clear that other core systems were under development. In particular, the graphics working group of DIN (UA5.9), chaired by Jose Encarnacao, were working on the specification of a core graphics system to become a German Standard. The meeting resolved that the Germans and Americans should work towards a common specification of a core graphics system.

1.4 THE ISO GRAPHICS WORKING GROUP

The first formal meeting of WG2 (the graphics working group) was held in Bologna in September 1978. A report of GSPC activities was received and the DIN group reported on GKS, detailing timescales for a DIN standard to be approved in 1981. Norway indicated their intention to propose IDIGS as a Norwegian standard; IDIGS was to be a successor to GPGS, a graphics system in widespread use in Norway and the Netherlands. In order that a single standard proposal might be possible, an Editorial Board was set up to compare the various proposals and recommend changes so that the three proposals might converge or at least be compatible.

The Editorial Board, chaired by Paul ten Hagen (WG2 convenor) and Bob Hopgood, met in Amsterdam in February 1979 and was presented with GKS Version 3 (a document of 46 pages) and the 1977 GSPC Core report (a document of 117 pages), whilst the IDIGS proposal was not available. Two major differences between the proposals were the inclusion of a current position concept in the GSPC Core (GKS did not have one) and the pen concept for attribute handling in GKS (GSPC Core adopted the more conventional approach of individual attribute setting) [5]. The fact that GSPC Core was a 3D system and GKS was a 2D system did not itself cause problems since GSPC Core was also capable of 2D graphics. The Editorial Board recommended changes to both proposals to bring them closer together [6]. Both DIN and GSPC discussed the Editorial Board's recommendations and a joint meeting was held in Boulder, Colorado.

By June 1979, it was recommended that the GSPC work should be passed to the formal American standards body, ANSI. The ANSI graphics working group, X3H3, had its first meeting in September 1979. At its second meeting in December of the same year, X3H3 adopted the 1979 GSPC Core as the starting point for its work.

At the following meeting of the ISO graphics working group in Budapest in October 1979, GKS 5.0, incorporating many of the Editorial Board recommendations, was presented. In addition, the input facilities had been enhanced and the ability to use multiple output devices simultaneously had been introduced. The 1979 GSPC Core was presented by ANSI including a pen concept, related to that of GKS, and enhanced text output. IDIGS was also presented.

GKS was the most technically refined of the three and the DIN members were keen that GKS should be submitted to ISO as a standard proposal. After discussions as to whether the working group could evaluate two proposals at the same time, it was eventually decided that only GKS would be put to the parent body (the programming languages subcommittee ISO/TC97/SC5) for registration as a Work Item with the aim of it becoming a Draft Proposal in a year.

1.5 THE GKS REVIEW

A technical meeting was arranged in Tiefenbach, Germany in June 1980 at which national bodies raised issues resulting from a thorough review of GKS 5.2. About 300 issues were put before the meeting of which over 200 were raised by ANSI. The issues were of varying types including clarification of the document and suggested changes to increase the functionality or reduce the complexity. The meeting was complicated by the fact that DIN presented GKS 6.0 just prior to the meeting. However, it had resolved a large number of issues particularly in the area of clarification and so it was considered the most appropriate basis for discussion. More issues were resolved during the meeting but there was no consensus on many of the main substantive issues. Some 50 issues remained unresolved. GKS 6.2, now 132 pages, was produced as a result of the meeting. It was agreed that this should be the basis of the first of two further rounds of technical discussion.

For these two rounds the issue submission and documentation procedure was formalized. Based on the ideas of GSPC and ANSI, each issue was presented as a question, followed by a description and a set of alternative answers. Arguments in favour of and against the alternatives were listed. The complete list of unresolved issues was referred to as an Active Issues List. An editorial round to improve the language and to produce the document in the correct ISO format proceeded in parallel.

The British Standards Institution (BSI) delegation made its presence felt at the technical experts' meeting in Melbourne, Florida in January