

IL VIDEO TEKTRONIX 4025

61

UN'APPLICAZIONE PER IL GRAFICO IN MODO INTERATTIVO  
DI MAPPE RADIO BIDIMENSIONALI

A. Ficarra

LRA 40/80

LABORATORIO INTERNO

CONSIGLIO NAZIONALE DELLE RICERCHE

LABORATORIO DI RADIOASTRONOMIA

ISTITUTO DI FISICA "A. EINSTEIN"

VIA P. ABATE, 12 - 00187 ROMA (ITALIA)

## INTRODUZIONE AL TEKTRONIX 4025

Il video Tektronix 4025, recentemente acquistato in questo Istituto, promette di essere un mezzo molto utile per l'esecuzione di lavori in modo interattivo. La programmabilità della tastiera, per esempio, permette di associare a ogni tasto un'intera stringa di istruzioni, anche dinamicamente, cioè nell'ambito della esecuzione di un programma. Inoltre la facilità d'uso del sistema e la sua grande versabilità, per cui è possibile comporre insieme istruzioni CMS e comandi locali, oppure svolgere "editing" locale su un intero testo e poi spedire tutto al calcolatore con un solo comando, offrono la possibilità di operare in modo rapido ed efficiente su una vasta gamma di applicazioni.

La memoria locale è suddivisa in due parti completamente distinte: la "workspace area" e la "monitor area". Entrambe queste porzioni possono contenere la lista dei comandi o dei dati spediti al calcolatore da tastiera e le risposte o i risultati inviati dal calcolatore. I grafici sono sempre eseguiti nella "workspace". I comandi locali sono visualizzati sempre nella "monitor" se inviati da tastiera, mentre sono ugualmente eseguiti ma non visualizzati se inviati dal calcolatore.

Il sistema riconosce un comando locale se il primo carattere di una stringa è un particolare carattere, il "command character" (di solito il punto esclamativo), a sua volta ridefinibile tramite un particolare comando.

Facendo uso di alcuni comandi è possibile istruire il sistema su come suddividere le 34 righe dello schermo fra "workspace" e "monitor" e dove devono essere visualizzati i dati immessi da tastiera e le risposte inviate da calcolatore (+). E' così possibile, fra l'altro, separare l'input dall'output e ciò è molto utile nel caso che l'output sia costituito dai risultati di un programma e non si voglia mescolare tali risultati con i dati input o con la lista dei comandi CMS.

---

(+) In ogni caso è utile aggiungere che sullo schermo è visualizzata solo una piccola parte della memoria: in entrambe le aree i dati entrano sullo schermo dal basso ed escono dall'alto, come se lo schermo fosse una finestra e i dati fossero scritti su un foglio che si arrotola sopra e sotto lo schermo. Esistono due tasti premendo i quali si può far scorrere verso l'alto o verso il basso questo ipotetico foglio ("scrolling"), facendo così riapparire quella parte di display che è già uscita dallo schermo.

I dati, infine, sono spediti al calcolatore riga per riga se provengono dall'area "monitor", mentre tutta la "workspace" è trasmessa con un solo comando (utile per la trasmissione di testi "editati" localmente).

Per quanto riguarda i grafici è stato detto che questi vanno a occupare sempre una porzione della "workspace". Uno speciale comando ( !GRA ) definisce quante righe (max. 52) e quante colonne (max. 80) della "workspace" devono essere dedicate all'area grafica. A seguito di questo comando il sistema pone lo zero di riferimento nell'estremo punto in basso a sinistra. Ciascuna cella alfanumerica dello schermo è reinterpretabile così come una matrice grafica di punti su 14 righe per 8 colonne, per un massimo quindi di 640 punti in ascisse (da 0 a 639) e 728 punti in ordinate. Tramite un successivo comando ( !SHRINK ), il sistema è istruito a ridurre di  $\frac{5}{8}$  il valore numerico delle coordinate prima di tradurle in un punto grafico. In questo modo i limiti delle coordinate di schermo che si possono fornire sono da 0 a 1023 in ascisse e da 0 a 1100 circa in ordinate (consentite solo fino a 1023). Questa strana procedura è obbligatoria perché non esiste un "software" specifico per il modello 4025. Le "routines" usate sono quelle della serie 4010 in cui lo schermo ha una capacità di 1024 per 1024 punti. Con il comando !SHRINK il programmatore simula quindi l'esistenza di un 4010 e il sistema sa che per graficare i punti ne deve prima ridurre le coordinate alla capacità dello schermo del 4025.

In locale esistono pochi comandi grafici, che permettono sostanzialmente di disegnare segmenti di cui vengono fornite le coordinate degli estremi. A programma, invece, è possibile utilizzare una libreria di "subroutines", fornita dalla Tektronix e residente sui dischi dell'IBM 370/168 del CNUCE. Lavorando in CMS e aggranciando con il comando GLOBAL il file che contiene tale libreria, che si chiama TCS10 (istruzione automaticamente eseguita perché inserita nel PROFILE di ogni macchina virtuale), l'utente può costruire dei grafici a programma, inserendo nel testo le chiamate alle subroutines che gli interessano.

Il principio base su cui lavorano tali subroutines è il seguente: anzitutto è necessario definire l'area dello schermo entro cui deve essere contenuto il grafico. I limiti di tale area devono essere dati in coordinate di schermo, cioè devono essere dei numeri interi e non devono eccedere l'intervallo 0 - 1023, che è il massimo consentito in entrambe le direzioni. Se la subroutine che definisce tale "finestra di schermo" non è richiamata, i limiti di "default" fissati dal sistema sono 0 - 1023 in ascisse e 0 - 780 in ordinate. A questo punto l'utente può se vuole costruire già dei grafici, specificando le coordinate dei suoi punti in unità di schermo, cioè mediante numeri interi compresi entro i limiti definiti dalla finestra di schermo. Esiste a questo proposito una serie di subroutines che

permettono di tracciare linee a tratto intero, o tratteggiate in diversi modi, oppure di spostare semplicemente il pennello luminoso ( l'equivalente della "penna alzata" del plotter) ecc..; gli argomenti di queste subroutines, cioè le coordinate del punto su cui si deve spostare il pennello, sono sempre numeri interi e, a secondo della subroutine che si usa, possono corrispondere alle coordinate assolute del punto da raggiungere o alle coordinate relative all'ultimo punto raggiunto.

Il sistema offre un'altra possibilità, che permette un utilizzo del video più comodo e funzionale: quella di riferire le coordinate a una scala arbitrariamente scelta dall'utente. Per ottenere ciò bisogna definire, richiamando una particolare subroutine, una "finestra virtuale" che indica a quali limiti in coordinate di utente (fornite questa volta come numeri "floating") corrispondono gli estremi della finestra di schermo. Alcune subroutines svolgono funzioni del tutto analoghe a quelle già descritte, con l'unica differenza che gli argomenti sono numeri non più interi ma "floating" e hanno il significato di coordinate dei punti in questa nuova scala. Se qualche punto del grafico è localizzato fuori dalla "finestra virtuale", si possono ugualmente tracciare linee dirette verso tale punto, salvo il fatto che appare visibile solo quella parte di grafico che giace all'interno dei limiti fissati, proprio come se esistesse una "finestra" che permetta la visione solo di una parte del disegno. Cambiando opportunamente i limiti della finestra di schermo e della finestra virtuale si possono ottenere dei risultati interessanti, come quello di deformare in una o entrambe le direzioni un grafico, oppure fare degli "zoom" per esaminare in dettaglio certe regioni ecc....

In questa breve introduzione si è voluto illustrare i principi generali del funzionamento del terminale video 4025. Per una sua conoscenza più approfondita è necessario consultare i seguenti manuali della Tektronix :

- 1) 4025 Computer Display Terminal/ Programmer's Reference Manual
- 2) 4025 Computer Display Terminal/ Operator's Manual
- 3) Tektronix Plot10 Terminal Control System/ User Manual

I primi due manuali insegnano l'utilizzo del terminale e dei comandi locali, il terzo descrive il "package" di subroutines necessarie per ottenere grafici a programma.

## INTRODUZIONE AL PROGRAMMA

Il programma, di cui descriveremo l'uso nel prossimo paragrafo, è stato scritto con lo scopo di visualizzare sullo schermo mappe bidimensionali di intensità radio, con l'opportunità anche di scegliere interattivamente zone particolari, "zoommare" su di esse e avere una prima rudimentale misura del flusso e della posizione.

Il metodo di graficazione è quello ben noto di costruire tante strisce, una per declinazione, spaziate in direzione dell'asse y; ciascuna striscia è così un grafico unidimensionale dell'intensità in funzione dell'ascensione retta (asse x).

Il programma si aspetta di trovare i dati in un file di una macchina virtuale, a cui l'utente deve accedere, prima dell'esecuzione, con un'istruzione del tipo: "FI 09 DISK nome tipo modo". Tale file contiene alcuni records iniziali con informazioni generali e successivamente la matrice dei dati a gruppi di declinazioni; all'interno di ogni gruppo i punti sono dati come numeri interi in unità di 5 mJy in funzione dell'ascensione retta, con spaziatura di 4 sec.. Questo è il formato tipico dei dati radio come risultano dal procedimento standard di elaborazione delle registrazioni fatte con la Croce di Medicina. C'è da dire, tuttavia, che con leggere modifiche, riguardanti soprattutto l'organizzazione dei dati di input e il significato da attribuire alle coordinate dei punti, il programma può essere reso di interesse ben più generale e applicarsi a tutti i casi in cui serve visualizzare per strisce unidimensionali i punti di una matrice.

Tornando al file di input, una sua descrizione dettagliata sarà fornita nel prossimo paragrafo. L'utente deve infatti conoscere esattamente come sono organizzati i dati, in quanto è compito suo scrivere un semplice programma "batch" che legga i dati da disco o da nastro e li perfori sulla propria macchina virtuale, riorganizzandoli nel modo voluto dal programma interattivo. Nel caso, particolarissimo, che i dati provengano dalla convoluzione in Nord-Sud di registrazioni originariamente effettuate a intervalli di puntamento di 15' (Perseo, Coma, seconda parte della Survey B3 ecc...), il programma "batch" esiste già: risiede nella libreria di moduli load CO63MDL2 e si chiama M149; legge i dati dall'unità 9 e li trasferisce sul perforatore virtuale; usa una sola scheda di input (unità 5) con i seguenti dati:

IOI,MII,ISI,IOF,MIF,ISF,LSC,LF1,LF2      FORMAT(9I3)      dove:

IOI,MII,ISI = ora, minuti e secondi del primo punto da trasferire

IOF,MIF,ISF = " " " dell'ultimo " "

LSC = 0 : dati coseno; 1 : dati seno

LF1 = numero d'ordine della prima declinazione da trasferire

LF2 = " " dell'ultima " "

Il programma interattivo è stato scritto in linguaggio FORTRAN. Il testo e il modulo compilato sono accessibili nel disco C di RADIOLAB con il nome, rispettivamente, di TEK1 FORTRAN e TEK1 TEXT.

Il programma usa le subroutines della libreria TCS10 della Tēktronix, che sono automaticamente "linkate" in quanto il relativo comando GLOBAL è contenuto nel PROFILE di ogni macchina virtuale; cosicché, per eseguire il programma senza modifiche, è sufficiente dare il comando RUN TEK1.

Prima dell'esecuzione, tuttavia, è necessario formattare il video in modo che una parte dello schermo sia dedicata al grafico. Per fare questo bisogna anzitutto definire quante righe dello schermo sono da riservare alla "workspace" (normalmente infatti tutto il video è lasciato a disposizione dell'area "monitor"); successivamente, con il comando !GRA 1,52 si deve inizializzare l'area grafica e infine si deve fornire il comando !SHRINK. Risulta molto comodo, a questo proposito, l'uso di un file contenente una serie di comandi locali già preparati. Questo file si chiama TEK TEK e risiede anch'esso nel disco C di RADIOLAB. I comandi locali contenuti nel file, eseguiti immediatamente con un'istruzione del tipo TY TEK TEK, sono stati costruiti con lo scopo di riprogrammare la tastiera del video, attribuendo una particolare funzione (utile in molti casi di "editing" e di definizione di aree) ad alcuni tasti che del resto non vengono quasi mai usati nel loro significato originario. Dopo l'esecuzione di questi comandi la tastiera del video risulta così riprogrammata:

Tasto	f u n z i o n e
PAD KEY 0	!WOR 0 : Ripristina la "monitor area" su tutto lo schermo.
PAD KEY 3	Esegue 3 spazi (utile per il sottocomando OVERLAY del TED)
PAD KEY 5	!MON : i dati provenienti da tastiera sono diretti sulla "monitor".
PAD KEY 9	!WOR : i dati da tastiera sono diretti sulla "workspace".
PAD KEY .	!WOR 30 ; !GRA 1,52 ; !SHRINK : Riserva 30 righe dello schermo alla "workspace"; crea un'area grafica di 52 righe; comanda la riduzione della scala di 5/8 per simulare un sistema 4010 .
PAD TERMINATOR	CARRIAGE RETURN ; !BUF NO (utile per uscire dal "modo bufferato" alla fine di un'istruzione)
FUNCTION KEY 8 (SEND)	!BUF YES ; !SEND (utile per spedire un intero testo dalla "workspace area" al computer)
↖	# (utile per andare a capo in CMS senza premere il tasto delle maiuscole)
;	: (utile per andare a colonna 7 nella stesura di programmi FORTRAN senza premere il tasto delle maiuscole)
ESC	!BUF YES (per passare in "modo bufferato")

Come si può vedere, le funzioni eseguite dai tasti così riprogrammati sono utili in una vasta serie di casi. Nel nostro specifico tutte le operazioni necessarie per preparare il video all'esecuzione del programma sono eseguite semplicemente premendo il tasto "PAD KEY ." .

#### SUBROUTINE CURSOR

E' stato detto che il programma accede alle subroutines della libreria Tektronix. Oltre a queste il programma usa una particolare subroutine che è stata scritta allo scopo di sopperire al più grave difetto (a parer nostro) del sistema Tektronix 4025, cioè la mancanza di input grafico. Infatti un sistema, per essere realmente interattivo, deve non solo produrre grafici, ma anche prevedere la possibilità di operare sui grafici stessi, rinviando al calcolatore i risultati delle misure e attendendo le decisioni che verranno prese di conseguenza. In altre parole, la mancanza di un cursore grafico posizionabile da tastiera e in grado di leggere e trasmettere al calcolatore le coordinate del punto su cui si trova, costituisce una fortissima limitazione alle possibilità interattive del sistema.

A disposizione dell'utente c'è solo il cursore alfanumerico, gestibile tramite alcuni comandi locali che permettono di posizionarlo o di spostarlo di un numero voluto di righe o di colonne. Sfruttando queste limitatissime risorse, si è cercato di attrezzare il sistema di un rudimentale input grafico, tramite la stesura di una subroutine, di nome CURSOR. Tale subroutine è stata scritta seguendo criteri del tutto generali ed è quindi applicabile da qualunque programma necessiti di input grafico. Ovviamente, ed è questa una limitazione inevitabile, la precisione nel posizionamento del cursore alfanumerico non può essere quella del grafico, ma solo a step di numero di righe e colonne. Tuttavia può essere ugualmente sufficiente per la localizzazione di zone del grafico di particolare interesse. Inoltre la subroutine offre un interessantissimo esempio di applicazione della possibilità di riprogrammare dinamicamente la tastiera, cioè di definire a programma, a secondo delle necessità, particolari funzioni a certi tasti, ripristinando alla fine le condizioni iniziali.

La subroutine CURSOR si trova insieme al programma principale nel file TEK1 FORTRAN e, nella versione già compilata, nel file TEK1 TEXT. Nessuna operazione di "link" è pertanto necessaria da parte dell'utente che voglia eseguire il programma.

Tuttavia vogliamo insistere sul fatto che la Subroutine ha anche un utilizzo generale e riteniamo quindi utile metterla in evidenza in questo paragrafo. Il testo della sola Subroutine si trova anche nel file CURSORE FORTRAN.

L'ordine di richiamo é : CALL CURSOR(L,XC,YC), dove :

L é un dato input costante che indica su quante righe é stata definita l'area grafica con il comando !GRA (comunemente L = 52 ).

XC,YC sono le coordinate,in unità grafiche di utente, del punto su cui il cursore si trova posizionato all'uscita dalla Subroutine.

L'uso della Subroutine é immediato: non appena appare nell'area "monitor" la scritta: "CURSORE" viene evidenziata la posizione del cursore nella "workspace" e l'elaboratore resta in attesa che l'utente trasferisca manualmente il cursore nella posizione voluta; infine, premendo il tasto "RETURN" (qualunque altro tasto avrebbe effetto distruttivo sul grafico), l'utente provoca l'uscita dalla subroutine con il trasferimento in XC e YC delle coordinate del punto raggiunto dal cursore. Tali coordinate, s'è detto, sono in unità grafiche di utente, ma la precisione del posizionamento é limitata a uno step di riga o di colonna.

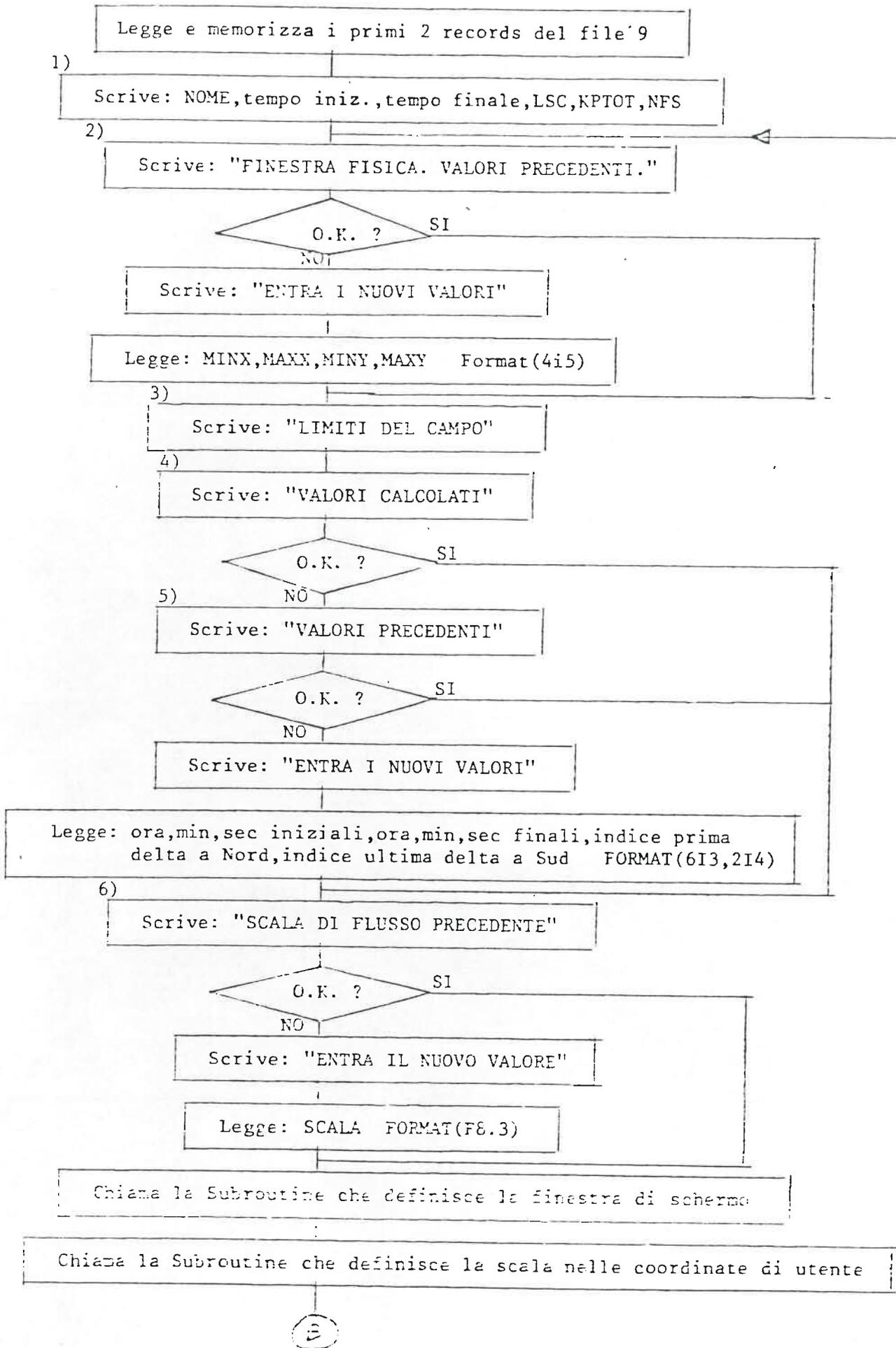
#### USO DEL PROGRAMMA

S'è detto che il programma legge i dati in un file della macchina virtuale su cui sta lavorando. Il file é costituito da records logici contenenti nell'ordine le seguenti variabili:

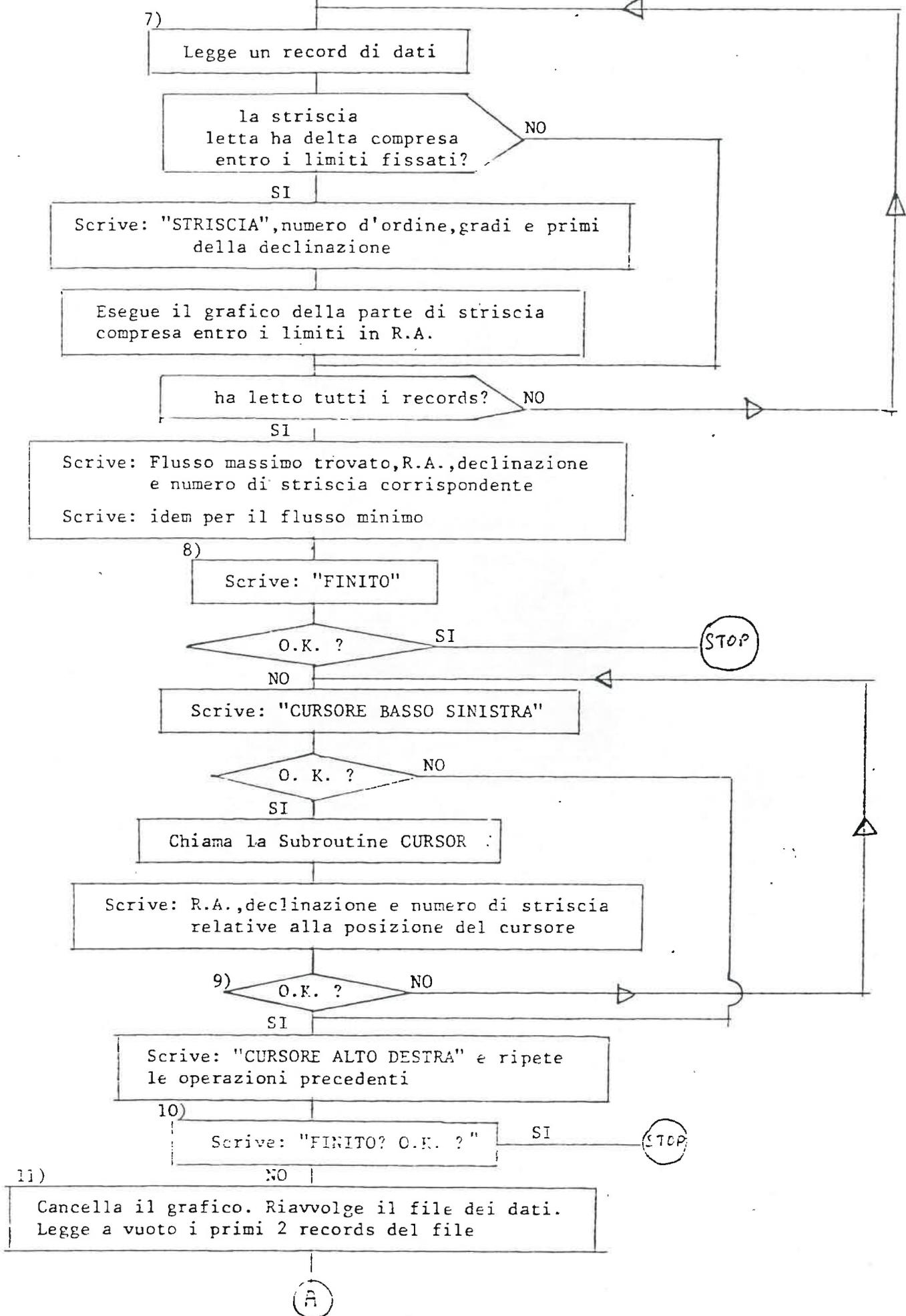
- I) NOME,IOI,MII,ISI,IOF,MIF,ISF,LSC,KPTOT,NFS      FORMAT(A8,7I3,I5,I3)      dove:  
NOME (REAL\*8) = una stringa alfanumerica qualsiasi associata come titolo ai dati.  
IOI,MII,ISI = ora,min,sec del primo punto in R.A.  
IOF,MIF,ISF = ora,min,sec dell'ultimo punto in R.A.  
LSC = 0 : dati coseno ; 1 : dati seno.  
KPTOT = Numero totale dei punti in R.A.  
NFS = Numero totale dei punti in declinazione.
- II) (IGC(K),PRC(K),K=1,NFS)      FORMAT(20A4)      dove :  
IRC(K),PRC(K) = gradi.e primi della declinazione k-esima. Le declinazioni sono memorizzate in ordine da Nord a Sud.
- III) Per ogni declinazione: (IDAT(J),J=1,KPTOT)      FORMAT(40A2)      dove:  
IDAT(J) (INTEGER\*2) = Il flusso nel punto di j-esima R.A. Per ogni punto in declinazione si ha un record logico di dati in R.A. Le R.A. sono date in ordine crescente. Il flusso é dato sotto forma di multiplo intero di 5 mJy.

Il programma lavora utilizzando l'area "monitor" per le scritte alfanumeriche e per i dati di input e lasciando all'area "workspace" la parte puramente grafica. E' orientato a colloquiare con l'utente in modo tale che per ogni nuova scelta sia l'utente stesso sul momento a prendere una decisione, generalmente rispondendo a una domanda in termini di un SI o un NO (qualunque altra risposta é interpretata come un NO).

Per comprendere bene il funzionamento del programma (che l'interattività rende peraltro di uso molto semplice e immediato) riteniamo utile descriverlo tramite uno schema a blocchi cui seguiranno note esplicative.



3



NOTE

- 1) All'inizio il programma trascrive su video le informazioni generali sui dati lette nel primo record del file di input (file 9)
- 2) Da qui in poi si parla di "VALORI PRECEDENTI" in quanto il programma lavora in un "loop" che gli permette, ritornando a questo punto, di ricreare ogni volta un nuovo grafico, eventualmente modificando alcuni parametri secondo scelte opportune.  
La finestra fisica, detta anche area di schermo, definisce quale parte dell'area di video disponibile deve essere utilizzata per il grafico. I limiti iniziali sono fissati in modo da prevedere l'utilizzo di tutto lo schermo disponibile, cioè hanno valore 0 e 1023 in entrambe le direzioni.
- 3) Per "limiti del campo" si intendono i valori in R.A. e declinazione (questa ultima in termini di numero d'ordine di record di dati, o "striscia", come si dirà d'ora in poi) che dovranno corrispondere sul grafico ai limiti della finestra fisica. Tramite questa corrispondenza sarà possibile definire la scala in unità grafiche di utente (ascensione retta e declinazione).
- 4) I "valori calcolati" dei limiti del campo sono quelli ottenuti dal posizionamento del cursore sul grafico fatto precedentemente. La prima volta questi valori coincidono con quelli dell'intera registrazione, letti nel primo record del file dei dati.
- 5) Anche i "valori precedenti" coincidono la prima volta con i limiti dell'intera registrazione.
- 6) E' noto che usando questo metodo di graficazione, cioè scomponendo una funzione bidimensionale in "strisce" di funzioni di una sola variabile, l'altra variabile e la funzione stessa si estendono lungo la stessa coordinata. Noi definiamo la scala di flusso fornendo al programma il valore dell'intensità in  $J_y$  che corrisponde a uno step in declinazione. Il valore "precedente" iniziale è 1 .
- 7) Notare che non è necessario trasferire l'intera matrice dei dati in memoria in quanto ogni striscia viene immediatamente "plottata" dopo la lettura. D'altronde non sarebbe possibile una scelta diversa, a causa della limitatissima memoria a disposizione di un programma che "giri" sotto macchina virtuale.
- 8) Ottenuto il grafico, con in aggiunta l'informazione sulla posizione e flusso dei punti di massima e minima deflessione, l'utente può ritenersi soddisfatto e provocare l'uscita del programma, oppure può voler riesaminare con maggiore

dettaglio un campo all'interno di quello disegnato. Per definire tale campo l'utente può conoscere già o calcolare i limiti estremi fornendoli poi al programma durante la preparazione del nuovo grafico ( in questo caso deve rispondere NO alle prossime richieste di uso del cursore); oppure l'utente vuole definire visivamente sul grafico attuale i limiti del nuovo campo e allora ha la possibilità di posizionare due volte il cursore fissando i vertici, la prima volta in basso a sinistra e la seconda volta in alto a destra, del rettangolo che delimita il campo. La subroutine CURSOR restituirà al programma le due coppie di coordinate così definite e il programma determinerà da esse i "valori calcolati" dei limiti del campo che l'utente potrà assumere come nuovi limiti per il grafico successivo.

- 9) Esiste la possibilità che l'utente voglia, per un errore precedente o un ripensamento, ripetere l'operazione di posizionamento del cursore.
- 10) E' previsto uno stop di programma a questo punto, nel caso in cui l'utente si sia servito del cursore non per definire i limiti di un nuovo campo, ma semplicemente per ottenere le coordinate di alcuni punti del grafico.
- 11) Nel caso che l'utente voglia costruire un nuovo grafico, il programma ritorna al punto di cui alla nota 2), dopo avere operato un "rewind" del file dei dati e dopo aver saltato la lettura dei primi due records che contengono informazioni immagazzinate già in memoria la prima volta.

## CONCLUSIONI

Nell'introduzione di questo rapporto si è cercato di illustrare i vantaggi e i pregi offerti dal sistema video Tektronix 4025. Non bisogna però nascondere che, nell'uso, si sono riscontrati anche certi difetti, alcuni dei quali intrinseci al sistema, altri legati alla particolare configurazione in cui attualmente si trova il sistema stesso.

Alla prima categoria appartiene la mancanza, di cui si è già detto, di un vero e proprio input grafico ad hardware, indispensabile per fare del lavoro realmente interattivo.

Inoltre la memoria grafica del sistema è alquanto limitata, anche nella sua configurazione massima ( 32 KByte ): per fare un esempio che chiarisca questo concetto, basta osservare che il programma TEK1 può "plottare" al massimo circa 300x30 punti; superato questo limite, il "pointer" continua a percorrere lo schermo, ma non appare più nessun segno grafico sul video.

Un altro limite piuttosto scomodo è l'inesistenza di routines che permettano di accompagnare al grafico scritte alfanumeriche di dimensioni prefissate (l'analogo della SYMBOL e della NUMBER nel "package" CALCOMP): le dimensioni della cella alfanumerica sono inalterabili, a meno che non si ridefinisca, carattere per carattere, i segni grafici che costituiscono i simboli, procedura certo estremamente lunga e laboriosa.

Infine, l'ultimo difetto "intrinseco" del sistema consiste nella scarsa velocità con cui procede un grafico mentre si forma sullo schermo: corrisponde, per fare un esempio, alla velocità di un terminale che lavori a 300 baud. E' chiaro che questa lentezza si traduce in un'ulteriore limitazione delle possibilità interattive del sistema.

Alla seconda categoria appartiene l'impossibilità attuale di memorizzare permanentemente il grafico su qualche supporto. Infatti il metodo che normalmente si segue in lavori interattivi di questo genere è quello di operare sui grafici esaminandoli visivamente e modificandoli in base a scelte derivate dall'esame visivo e dalle misure effettuate sui grafici stessi; spesso però si vuole conservare la versione finale che risulta da tutto questo lavoro. Il problema potrebbe essere risolto semplicemente attrezzando il sistema con un hard-copy o un plotter. L'hard-copy non è stata prevista quando si è decisa la configurazione del sistema, perché ad essa è stato preferito il plotter, il quale peraltro non è ancora in grado di funzionare per motivi tecnici dovuti a gravi inadempienze da parte della

casa fornitrice ( questa situazione può essere messa nel conto dei difetti di funzionamento del sistema) .

Per finire, un certo disagio nasce dal tipo di collegamento attuale del sistema che, come è noto, funziona come terminale interattivo dell'elaboratore IBM 370/168 del CNUCE. Lavorando in ambiente CMS, si è costretti a operare, salvo speciali concessioni, nell'angusto ambito della propria macchina virtuale, con la conseguenza che i dati da elaborare devono essere prima trasferiti dal disco o nastro dove risiedono in un file CMS, che non può essere troppo grande per non esaurire lo spazio a disposizione. Inoltre il programma stesso non può occupare troppa memoria, per le stesse ragioni; tutto ciò limita fortemente le possibilità d'uso del sistema. Per ovviare a queste difficoltà si è progettato di disconnettere il video dall'IBM e collegarlo all'elaboratore VAX della Digital, di prossima installazione presso il nostro Istituto, il cui sistema interattivo non è soggetto alle stesse pesanti limitazioni.