IL SISTEMA GRAFICO "APLCORE"

Giorgio Faconti
Patrizia Palamidese
Antonino Santoro

Rapporto C83-16

Istituto CNUCE
PISA
Settembre 1983

INTRODUZIONE

L'"APLCORE" e' un sistema grafico interattivo che nasce dalla utilizzazione del sistema APL e del CORE e dei due ne possiede le principali caratteristiche e potenzialita'.

Il risultato di questa fusione e' la possibilita' di scrivere programmi applicativi in linguaggio APL. E' noto come questo linguaggio consente di operare su strutture dati quali vettori e matrici e di eseguire facilmente operazioni anche complesse con l'ausilio di semplici operatori. L'uso dell'APL semplifica quindi certamente la stesura di programmi applicativi per la rappresentazione di oggetti in due e tre dimensioni.

In questo sistema il ruolo dell'APL e' quello di preparare i dati e i parametri necessari al CORE per effettuare il disegno di linee e caratteri.

Questa funzione e' resa possibile dal fatto che l'APL non e' soltanto e semplicemente un linguaggio di programmazione ma costituisce un vero e proprio sistema capace di accedere alle risorse dell'elaboratore e di permettere lo scambio di informazioni tra differenti programmi attivi in memoria.

Nel seguito il sistema APL preso in considerazione e' il VSAPL Release 4 nell'implementazione sotto CMS installato in un calcolatore IBM 3033

L'APL come linguaggio

L'APL e' un linguaggio "general purpose" che viene largamente usato in molte applicazioni che vanno dal calcolo matematico-scientifico al trattamento di dati commerciali, dalla progettazione di sistemi al Business Planning, dal 'modelling' alla simulazione.

La caratteristica principale dell'APL e' quella di essere un linguaggio interpretato e quindi fortemente orientato alla programmazione di tipo conversazionale e interattivo. L'utente dialoga col calcolatore e introducendo un'istruzione ottiene immediatamente la risposta al proprio quesito. Inoltre in fase di stesura dei programmi, quest'ultimi possono essere fermati a ogni singola espressione, corretti e fatti ripartire dal punto in cui sono stati interrotti. Questo porta all'eliminazione dei tempi morti dovuti alla compilazione del programma e all'attesa della esecuzione dell'intero programma prima di poterlo correggere. Infine l'interattivita' permette all'utente di intervenire durante l'esecuzione stessa per compiere delle scelte che per la loro complessita' non potevano essere previste in anticipo.

Altre caratteristiche del linguaggio APL sono:

- regole di sintassi semplici ed uniformi

- uso di simboli per le usuali operazioni aritmetiche e per operazioni piu' complesse.

  Esempi:

  X←3×4 (assegnazione)

  X←⌹M (inversione di matrice)

- ingresso di numeri decimali in formato libero

- un vasto insieme di funzioni primitive (+,-,⍴,⌈,↑,⍳..)

- funzioni definite dall'utente che si possono utilizzare con la stessa facilita' di quelle primitive.

L'unita' organizzativa di base nel sistema APL e' lo spazio di lavoro (Workspace), un'area di memoria dove risiedono i programmi e i dati di input/output scritti dall'utente, rispettivamente classificati come funzioni e variabili.

Quando uno spazio di lavoro e' caricato in memoria diventa attivo e le informazioni in esso contenute possono essere modificate e utilizzate per l'esecuzione di funzioni specifiche. Alla fine di ogni sessione lo spazio di lavoro viene memorizzato su un supporto di memoria ausiliaria.

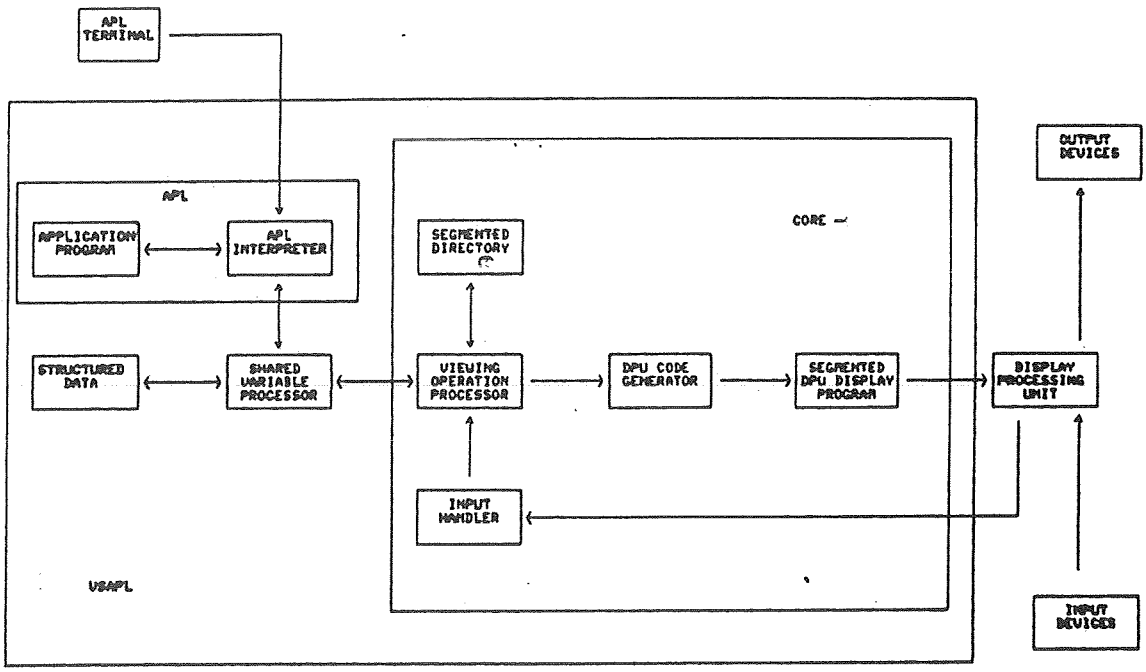
Nell'ambito dell'APLCORE la funzione svolta dall'APL e' quella di gestire i programmi applicativi che permettono di modellare l'oggetto che si vuole raffigurare e di descrivere al CORE il modo in cui deve essere rappresentato sull'unita' di output voluta.

Questo implica la costruzione di un modello, cioe' di un insieme di dati che rappresentano un oggetto e le relazioni che intercorrono tra di essi. Una volta fatto cio' il programma applicativo deve descrivere al CORE quale parte dell'oggetto deve essere vista, da quale punto di osservazione, l'unita' fisica su cui deve essere visualizzato e quale porzione di tale superficie deve occupare il disegno stesso.

Il sistema VSAPL.

In molte applicazioni risulta necessario uscire dal limitato ambiente dello spazio di lavoro APL per accedere ai servizi del sistema, come creare e utilizzare files non APL o scambiare informazioni con altri programmi. Questo e' reso possibile dall'esistenza di programmi (Auxiliary Processor), normalmente non residenti in memoria, ma che possono essere attivati e utilizzati in parallelo dall'utente VSAPL.

L'ambiente VSAPL e' costituito da uno SHARED VARIABLE PROCESSOR (SVP) che controlla lo scambio di informazioni tra gli Auxiliary Processors mediante delle variabili di scambio (Shared Variables). In questo contesto l'interprete APL e il CORE Graphic System sono due Auxiliary Processors. Viene riportato di seguito un diagramma funzionale del sistema grafico APLCORE.



APLCORE FUNCTIONAL BLOCK DIAGRAM

La procedura per lo scambio di informazioni tra due processors viene attivata sempre dall'utente VSAPL, il quale notifica al sistema l'intenzione di comunicare con un'altro programma e il nome della o delle variabili scelte per lo scambio: se il programma esiste viene caricato in memoria e da questo momento la variabile e' accessibile ad entrambi i processors: quando alla variabile viene assegnato un valore, lo Shared Storage Manager dell'SVP alloca uno spazio di memoria in cui viene trasferito tale valore; il successivo riferimento dell'altro processor al nome della variabile gliene fa acquisire il contenuto.

Una variabile di scambio puo' anche servire per accedere e modificare i valori utilizzati dal sistema come codici di controllo. In effetti le variabili di scambio usate nell'APLCORE sono due: una variabile di controllo e una variabile dati. La prima viene utilizzata per passare al Core Graphics System i codici numeri corrispondenti alle funzioni da eseguire e per esaminare eventuali codici di errore relativi all'ultima operazione svolta. La variabile dati e' utilizzata per passare al Core gli argomenti di input richiesti dalla funzione attualmente in esecuzione e per accedere ai valori da essa restituiti.

Vengono riportati di seguito alcuni esempi di funzioni APL per lo scambio di informazioni con il processor 213 (CORE):

```
      ∇INITIALIZE[□]∇
      ∇ INITIALIZE
[1]   INITCORE 2
[2]   INITSURF 2
[3]   SELSURF 2
[4]   SYSTEMTYPE 1
      ∇
```

```
      ∇INITCORE[□]∇
      ∇ INITCORE LEVEL;X
[1]   ⍝ INITIALIZE_CORE (LEVEL)
[2]   X←213 ⎕SVO 2 3 ⍴'DATCTL'
[3]   DAT←LEVEL
[4]   →FCIL 0
      ∇
```

```
      ∇FCIL[□]∇
      ∇ Z←FCIL C
[1]   CTL←C
[2]   →(0=RC←CTL)/⍴Z←10
[3]   'E(',(1↓ 6 0 ⍕RC+100000),')',DAT,';'
[4]   Z←0
      ∇
```

```
      ∇INITSURF[□]∇
      ∇ INITSURF SURFNAME
[1]   ⍝ INITIALIZE_VIEW_SURFACE
[2]   DAT←SURFNAME
[3]   →FCIL-2
[4]   →0
[5]   ⍝ 1 = 32FD
[6]   ⍝ 2 = 59FD
      ∇
```

```
      ∇SELSURF[□]∇
      ∇ SELSURF SURFNAME
[1]   ⍝ SELECT_VIEW_SURFACE
[2]   DAT←SURFNAME
[3]   →FCIL 4
      ∇
```

```
      ∇SYSTEMTYPE[□]∇
      ∇ SYSTEMTYPE N
[1]   ⍝N = 0 LEFT SYSTEM
[2]   ⍝N = 1 RIGHT SYSTEM
[3]   DAT←N
[4]   →FCIL 19
      ∇
```

Il CORE

Il CORE e' un sistema grafico di base composto da un insieme di soubroutines, scritte in Assembler, per la rappresentazione grafica di oggetti in 3D. La caratteristica essenziale di questo package grafico e' la portabilita' cioe l'indipendenza delle funzioni dalle particolari unita' di output usate. L'adattabilita' di quest'ultime viene assicurata da specifici drivers d'interfaccia.

Le possibilita' che il CORE offre all'utente per la generazione di immagini puo' essere divisa in sei classi:

1) Primitive di output: sono le funzioni che creano il disegno sotto forma di linee e testi, passati al Display Processing Unit (DPU) come lista di comandi.

2) Attributi delle primitive: queste funzioni controllano le caratteristiche delle primitive, come colore, intensita', tipo di linee.

3) Controllo dei segmenti: mediante queste funzioni e' possibile raggruppare le primitive di output legate da relazioni logiche in unita' (segmenti) modificabili negli attributi o ricreabili con nomi e caratteristiche differenti.

4) Operazioni di vista: servono a specificare al CORE quale parte dell'oggetto rappresentare e la superficie dell'unita' di output da occupare.

5) Funzioni di input: controllano l'interazione dell'utente con il programma applicativo.

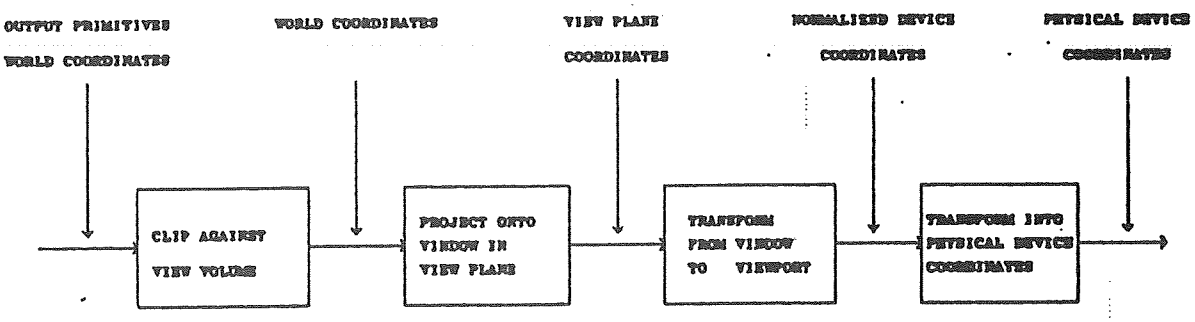
6) Funzioni di controllo: inizializzano a valori default lo stato del sistema grafico come la window o il viewport o forniscono informazioni sullo stato del sistema.

Il primo passo nel processo della generazione delle immagini e' costituito dall'acquisizione dei parametri che descrivono lo spazio entro cui e' confinato l'oggetto da rappresentare. Tali parametri, a seconda che si sia scelta una proiezione prospettica o parallela, definiscono un volume a forma di piramide o di parallelepipedo. Tutte le parti dell'oggetto che restano al di fuori di questo volume vengono scartate, rendendo cosi' possibile la visualizzazione di parti dell'oggetto considerate piu' interessanti dall'operatore.

I parametri di vista sono costituiti essenzialmente da:

- Window
- Viewport
- View Reference Point
- Center of Projection
- View Plane Normal
- View Plane Distance
- Front Clipping Plane
- Back Clipping Plane

La disponibilita' di questi elementi rende possibile le successive trasformazioni operate dal CORE e che possono essere schematizzate nel modo seguente:



PROCESSO DI GENERAZIONE DELL'IMMAGINE

Alla fine di questo processo l'oggetto, descritto inizialmente nel sistema di coordinate reali, dopo esser passato attraverso una trasformazione intermedia in un sistema di coordinate virtuali e "device independent" detto Normalized Device Coordinates (NDC), giunge allo stato finale in cui e' descritto in unita' del device fisico su cui verra' disegnato.

Questo stadio finale e' costituito dalla definizione di una lista di comandi che il DPU tradurra' in segnali analogici per controllare, per esempio, la deflessione in x e y, l'intensita' e la velocita' del pennello elettronico che traccera' fisicamente il disegno in un tubo a raggi catodici.

I codici cosi' generati dal DPU vengono organizzati in segmenti per poter permettere all'utente di gestire secondo le proprie esigenze le varie parti logiche in cui e' stato suddiviso l'oggetto da rappresentare.

Parallelamente al flusso di dati uscenti verso le unita' di output, il CORE puo' gestire interattivamente dati di input che possono essergli comunicati da vari dispositivi. Sebbene molti di questi ultimi possano permettere la generazione di eventi asincroni, il CORE presenta al programma applicativo una interfaccia sincrona per la loro gestione. Al livello 2 di implementazione il CORE non e' abilitato alla gestione asincrona degli eventi, mentre al livello 3 questo e' possibile con la creazione di una coda di eventi, che viene man mano svuotata dalle richieste del programma applicativo.

I dispositivi di input fisico tramite i quali si stabilisce una interazione tra il programma e l'operatore sono svariati e la loro gestione potrebbe costituire un problema per un sistema grafico. Questa difficolta' viene superata dal CORE ricorrendo alle proprieta' che accomunano i vari dispositivi di input rendendoli raggruppabili in cinque classi logiche:

- Pick: identifica un segmento e una primitiva all'interno di un segmento.
- Keyboard: fornisce informazioni alfanumeriche.
- Button: da' la possibilita' di scegliere tra diverse possibilita' predefinite.

Questi tre tipi di devices logici causano eventi, nel senso che notificano al programma informazioni dietro attivazione di una procedura di Wait, che fa mettere il programma in attesa di input da parte di una specifica unita'.

Gli altri due tipi sono:

- Locator: fornisce una coppia di coordinate in unita' di schermo (NDC).
- Valuator: fornisce un valore scalare.

Questi ultimi non creano eventi in quanto sono interrogati dal programma in seguito all'attivazione di una funzione di READ.

Segue un esempio di funzioni di un programma applicativo per la generazione di viste della Terra da punti di osservazione arbitrari con la sovrapposizione di tracce relative a un satellite in orbita terrestre. Questo programma e' stato effettivamente usato per seguire il decadimento orbitale del Cosmos 1402.

```
      ∇PLOTSFERA[□]∇
      ∇ PLOTSFERA POINT
[1]   QQ←LOLATOXYZ1 LOLA[;1 2]
[2]   SETUP POINT
[3]   DISEGNA 1
[4]   CREATESEG 3
[5]   MOVE QQ[1;]
[6]   POLYLINE QQ
[7]   CLOSESEG
[8]   WRITEHEAD 7
      ∇

      ∇SETUP[□]∇
      ∇ SETUP D
[1]   CLIPWINDOW 1
[2]   CLIPPLANES 1
[3]   VIEWPORT 0.06 0.7 0.06 0.7
[4]   VIEWUP 0 0 1
[5]   SETSEGTYPE 1
[6]   VIEWPDIST ¯1
[7]   VIEWREF 0 0 0
[8]   SETPARMS D
      ∇

      ∇SETPARMS[□]∇
      ∇ SETPARMS D;POSAT;PLN;W;R
[1]   POS←2↑D
[2]   DISI←D[3]
[3]   VIEWPNORM-POSAT←LOLATOXYZ POS
[4]   R←6378.14
[5]   PERSPECTIVE POSAT×(DISI÷R)÷R
[6]   PLN←R÷DISI÷R
[7]   OMEGA←(÷0÷180)×¯10÷PLN
[8]   W←(DISI÷R)×30¯10÷PLN
[9]   WINDOW 4↑(-W),W
[10]  VIEWDEPTH ¯1,-PLN
      ∇

      ∇DISEGNA[□]∇
      ∇ DISEGNA N
[1]   DWORLD N
[2]   DBOUNDRY N+1
      ∇

      ∇DWORLD[□]∇
      ∇ DWORLD N
[1]   CREATESEG N
[2]   DMAP
[3]   SETLNSTYLE 1
[4]   DMER -
[5]   DPAR
[6]   51 MARKER LOLATOXYZ POS
[7]   CLOSESEG
      ∇

      ∇DMAP[□]∇
      ∇ DMAP
[1]   USE 'XYZMAP'
[2]   DRAW
      ∇

      ∇DRAW[□]∇
      ∇ DRAW;M
[1]   L:M←AREAD
[2]   →(0=1↑⍴M)/0
[3]   MOVE M[1;]
[4]   POLYLINE M
[5]   →L
      ∇
```

BIBLIOGRAFIA

[1] ACM/SIGGRAPH GSPC:
" Status Report of the GSPC "
Computer Graphics Vol. 13, n.1, Mar.1979

[2] P.O. Bergeron, P.R. Bono, j.d. Foley
" Graphics Programming using the CORE System"
ACM Computing Surveys, Vol. 10, n.4, Dec 1978

[3] L. Gilman & A.J. Rose:
" APL: An Interactive Approach"
John Wiley Sons, Inc.

[4] G. Faconti, P. Palamidese, A. Santoro:
" Utilizzo di un Sistema Grafico Standard in Applicazioni per l'Analisi della Missione di Satelliti "
Aicographics, Ottobre 1982

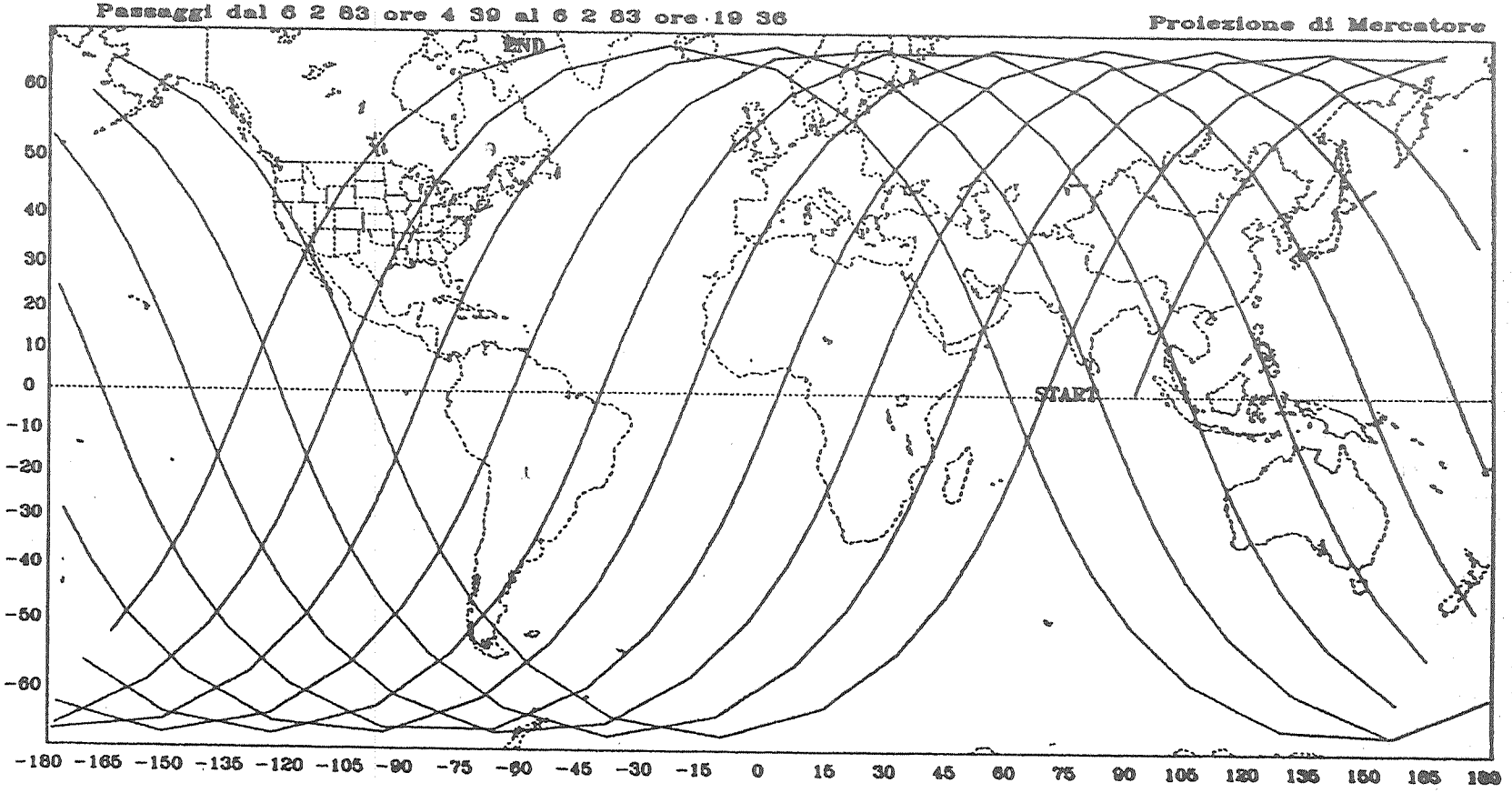
PROIEZIONE CONICA CON PUNTO DI VISTA IN :

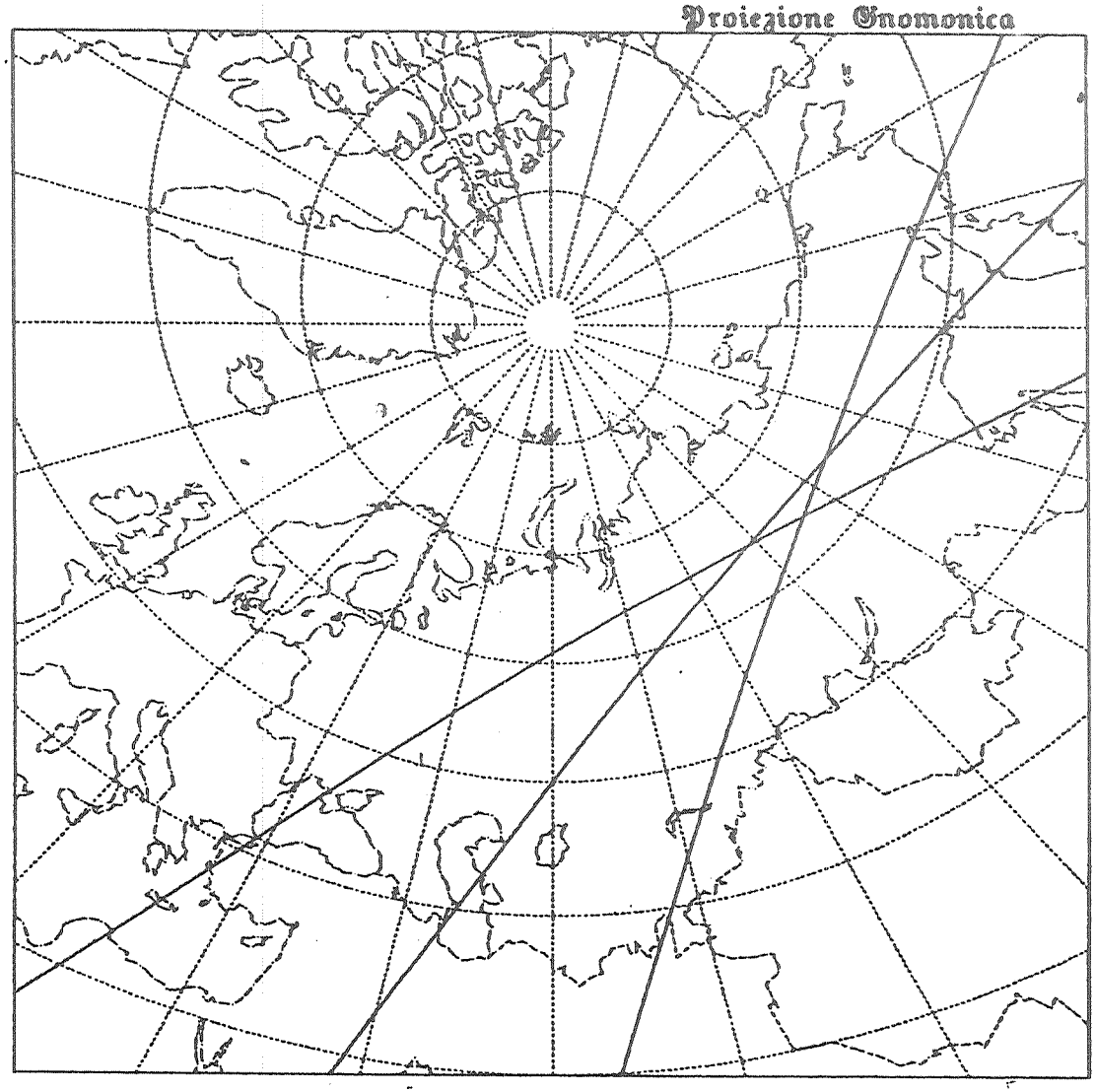
LON.	LAT.	ALT. (KM.)
20.00	40.00	36000

CONSIGLIO NAZIONALE DELLE RICERCHE
ISTITUTO CNUCE
per conto del PIANO SPAZIALE NAZIONALE

RUN ID.: ORBITA 5/2
S/C ID.: COSMOS 1402/C

EPOCH:	1983 2 5 9 38 6.68
SEMIMAJOR AXIS:	6546.49351
ECCENTRICITY:	0.0003862
INCLINATION:	64.984
ARG. PERIGEE:	305.7937
RIGHT ASC. NODE:	299.2011
MEAN ANOMALY:	54.3298





CNR – CNUCE

S/C ID : COSMOS 1402/C

DIREZ. DI VISTA : 60 70