

LOGICIEL D'AMÉLIORATION DE L'ORBITE D'UNE ÉTOILE DOUBLE VISUELLE

Méthode et Programmes

(Seconde Partie)

Edgar SOULIÉ

Commission des Étoiles Doubles

L'AMÉLIORATION D'UNE ORBITE

Lorsqu'une première orbite a été obtenue par la méthode exposée dans la première partie (OBSERVATIONS ET TRAVAUX n° 35, pp. 7-19), elle ne doit être regardée que comme préliminaire, parce qu'on n'a pas tenu compte de la loi des aires. Cette orbite doit être améliorée en faisant varier les éléments de Campbell de telle sorte que les éphémérides reproduisent au mieux les observations. Pour apprécier cette amélioration, on doit choisir une fonction des écarts ayant un minimum nul si tous les écarts venaient à s'annuler. La fonction la plus simple ayant cette propriété est la somme des carrés des écarts, et l'on retient le critère usuel des moindres carrés. Mais l'angle de position et la distance angulaire, et les écarts correspondants, ne sont pas des grandeurs de même nature. Aussi introduisons-nous les écarts réduits obtenus en divisant les écarts par les incertitudes correspondantes.

Soient $\theta(t; \mathbf{x})$ et $\rho(t; \mathbf{x})$ les formules d'éphémérides où \mathbf{x} représente le vecteur des éléments de Campbell dans l'espace \mathbb{R}^7 . Aux observations (t_i, θ_i, ρ_i) et aux incertitudes $\Delta\theta_i$ et $\Delta\rho_i$ correspond la fonction que l'on rendra minimale :

$$\mathcal{F}(\mathbf{x}) = \sum_i \left[\frac{\theta_i^{obs} - \theta(t_i; \mathbf{x})}{\Delta\theta_i} \right]^2 + \left[\frac{\rho_i^{obs} - \rho(t_i; \mathbf{x})}{\Delta\rho_i} \right]^2$$

Les éléments ont la probabilité maximale d'être la solution si les erreurs sont gaussiennes.

L'amélioration d'orbite est réalisée en plusieurs itérations; à chaque itération, on opère une modification sur les éléments, puis on calcule l'éphéméride pour les époques des observations et la fonction $\mathcal{F}(\mathbf{x})$.

Les méthodes d'amélioration empiriques ou semi-systématiques utilisées naguère impliquaient un assez grand nombre d'itérations pour parvenir à une représentation correcte des observations. Nous avons proposé (8) l'utilisation d'un algorithme général de minimisation d'une somme de carrés de fonctions non linéaires, dû à LEVENBERG et MARQUARDT (9).

Cet algorithme utilise les dérivées partielles premières des fonctions (les écarts réduits dans le cas présent) par rapport aux paramètres. Le calcul de ces dérivées à partir des formules d'éphéméride ne présente pas de difficulté, mais doit être soigneusement vérifié; cette vérification a été faite en comparant les dérivées analytiques avec les dérivées numériques. L'algorithme de LEVENBERG-MARQUARDT est applicable aux problèmes sans contraintes : les paramètres ajustables doivent pouvoir prendre chacun n'importe quelle valeur réelle. Ce n'est pas le cas des éléments de Campbell puisque pour une orbite elliptique, l'excentricité doit vérifier la double inégalité :

$$0 \leq e < 1$$

Pour transformer le problème de l'amélioration d'orbite avec contrainte de bornes en un problème sans contrainte, on substitue à l'excentricité la variables $e = \sqrt{\frac{e}{1-e}}$; en même temps, pour alléger les calculs de dérivées, on substitue à la période P le moyen mouvement $\mu = 2\pi/P$.

A la fin de la minimisation de la fonction $\chi^2(\mathbf{x})$ où \mathbf{x} désigne maintenant l'ensemble des variables internes ($\mu, T, s, a, i, \omega, \Omega$) où les angles sont exprimés en radians, on repasse aux éléments de Campbell, et l'on ramène les angles i, ω, Ω exprimés en degrés dans les intervalles standards respectifs. On juge de la qualité d'une orbite d'après les écarts résiduels sur les angles de position $(O-C)_\theta$ et les distances angulaires $(O-C)_\rho$.

Cette orbite résultant de la minimisation est enfin utilisée pour un calcul d'éphémérides à des époques futures. Au bout de plusieurs années ou décennies, des écarts systématiques pourraient apparaître, et il faudrait alors réviser l'orbite en tenant compte des nouvelles observations.

Notre programme d'amélioration d'orbite présenté plus loin met en oeuvre la simulation et la minimisation esquissées ci-dessus, et a été appliqué à de nombreux couples.

L'expérience de son utilisation a montré qu'une vingtaine de simulations suffisait largement pour parvenir à la bonne orbite - s'il en existe une.

Deux exemples d'application relatifs aux couples ADS 8635 = A 1851 et ADS 8862 = Hussey 644 sont décrits en détail dans la réf.(8).

Ci-dessous, reprenons l'exemple du couple ADS 8119 = STRUVE 1523. Les écarts initiaux sur les angles de position vont de -54 à +26 degrés; la plupart des écarts initiaux sur les distances angulaires sont inférieurs à la seconde de degré. Après une vingtaine d'itérations, on obtient un accord assez satisfaisant, comme le montre le tableau joint. Les paramètres finaux sont assez différents des paramètres initiaux :

$$P = 181,059 \text{ ans} \quad T = 1923,62 \text{ après J.-C.}$$

$$e = 0,49134 \quad a = 5,5681'' \quad i = 107,93^\circ \quad \omega = 29,96^\circ \quad \Omega = 106,51^\circ$$

Les éphémérides calculées pour 1986 à 1999 montrent l'intérêt que présenterait une mesure récente de ce couple. La difficulté traditionnelle de calcul des orbites est donc levée, et se trouve reportée sur les observateurs qui ne peuvent accumuler les mesures que lentement. Par ailleurs, le satellite d'astrométrie "Hipparcos" lancé en août 1989 fournit une assez bonne moisson de mesures d'étoiles doubles malgré l'échec de l'allumage du moteur d'apogée.

REMERCIEMENTS

L'auteur remercie le Docteur Paul Baize et Messieurs Pierre Bacchus, Paul Couteau et Pierre Dommanget pour leurs suggestions.

BIBLIOGRAPHIE (Suite)

- (8) Edgar SOULIÉ : *L'amélioration de l'orbite des étoiles doubles visuelles*. *Astronomy and Astrophysics*, vol.164, pp.408-414, année 1986.
- (9) Donald W. MARQUARDT : *An algorithm for least-squares estimation of nonlinear parameters*. *Journal of the Society of industrial and applied mathematics*, vol.11, n°2, p. 431-441, Juin 1963.
- (10) K.LEVENBERG : *A method for the solution of certain non-linear problems in least squares*. *Quarterly Journal of Applied Mathematics*, vol.2, p.164-168, année 1944.

IMPLANTATION

Langage de programmation : FORTRAN 77 (norme NF/Z 65-110 publiée par l'AFNOR en juin 1983, équivalente à la norme ISO 1539); la seule extension est l'emploi du caractère & comme caractère de continuation en colonne 6.

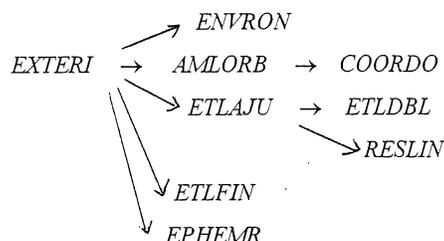
Périphérique nécessaire : imprimante.

Fichiers utilisés : fichier de lecture des données (unité logique IENTRE), fichier d'impression des résultats (unité logique ISOR).

Limitation du programme : le programme est limité à 136 observations. Pour permettre son utilisation avec un plus grand nombre d'observations, il faut remplacer dans toutes les instructions COMMON le nombre 136 par le nombre d'observations à utiliser, et le nombre 272 par le double du nombre d'observations à utiliser.

Outils de développement : Compilateur CFT avec options FOXZ sous le système d'exploitation COS sur ordinateur CRAY. Analyseur statique et dynamique "RXVP80". Analyseur syntaxique "FTAx" de Daniel Taupin. Analyseur statique et dynamique "LOGISCOPE" de la société VERILOG. Analyseur "TESTBED". Les outils "RXVP80", "LOGISCOPE" et "TESTBED" n'ont été essayés qu'occasionnellement.

ORGANIGRAMME



LE RÔLE DES UNITÉS DE PROGRAMME

EXTERI programme principal. Lecture interactive des noms externes des fichiers de lecture des données et d'écriture des résultats. Mise en correspondance de ces noms avec les unités logiques (ou noms internes) désignant ces fichiers à l'intérieur du programme.

AMLORB sous-programme dont dépendent tous les autres. Lecture du titre et des paramètres de départ. Conversion des angles en radians, et passage aux paramètres de minimisation. Appel au sous-programme de minimisation ETLAJU.

ENVRON détermination de la précision de la machine, utilisée dans les critères d'arrêt des itérations, pour la minimisation d'une part, et pour la résolution de l'équation de Képler d'autre part.

COORDO lecture d'une incertitude sur les angles de position, et d'une incertitude sur les distances angulaires. Lecture des observations qui peuvent être en nombre quelconque inférieur ou égal à 136. Stockage des observations "réduites" et lecture de la première et de la dernière année de calcul des éphémérides.

ETLAJU minimisation de la somme des carrés des écarts réduits suivant l'algorithme de Levenberg-Marquardt.

RESLIN résolution d'un système linéaire, nécessaire au calcul des nouveaux paramètres dans l'algorithme de Levenberg-Marquardt.

ETLDBL nom du point d'entrée principal du sous-programme ETLDBL. Calcul de l'angle de position et d'un écart ramené entre -180 et + 180 degrés, de la distance angulaire puis des valeurs calculées "réduites", ainsi que des dérivées partielles premières de l'angle de position et de la distance angulaire "réduites" par rapport aux sept paramètres internes de la minimisation.

ETLFIN point d'entrée secondaire de ETLDBL. Après ajustement, impression des éléments de Campbell ainsi que du tableau des angles de position et distances angulaires observés et calculés et des écarts correspondants.

EPHEMR point d'entrée secondaire de ETLDBL. Calcul et impression des éphémérides.

UTILISATION DU PROGRAMME

Au lancement du programme, la question suivante apparaît à l'écran : "Quel est le nom du fichier de lectures de données ?".

On tape au clavier le nom d'un fichier contenant des données adéquates (voir ci-dessous), puis on valide ce nom en tapant sur "Retour". Une seconde question apparaît alors à l'écran : "Quel est le nom de fichier d'écriture des résultats ?" On tape le nom choisi puis "Retour". L'exécution du programme proprement dit commence alors.

Le fichier de données du programme doit contenir les données suivantes, dans l'ordre :

- TITRE format (A80) une ligne de texte de 80 caractères.

- valeurs de départ des éléments de Campbell, en format libre, dans l'ordre suivant :

PERIOD = période de révolution en années décimales;

PASSAG = époque du passage au périastre de l'orbite réelle, en années décimales après J.C.;

EXCENT = excentricité de l'orbite réelle;

DEMAXE = demi-grand axe de l'orbite réelle, en secondes de degré;

AINCLI = inclinaison du plan de l'orbite réelle sur le plan perpendiculaire à la ligne de visée, exprimée en degrés décimaux, et comprise entre 0 et 180 degrés;

PERIAS = argument du périastre de l'orbite réelle, exprimé en degrés décimaux, et compris entre 0 et 360 degrés;

ANOEUUD = angle de position de la ligne des noeuds compté depuis le nord vers l'est, exprimé en degrés décimaux, et compris entre 0 et 180 degrés;

- EA, ER au format libre :

EA = incertitude affectant toutes les mesures d'angle de position faites une seule nuit, en degrés décimaux;

ER = incertitude affectant toutes les mesures de distance angulaire faites une seule nuit, en secondes de degré

- pour chacune des observations :

IRANG, T(I), B, DIST(I), NUIT, OBSERV(I)

FORMAT(I3, 2X, F8.3, 1X, F5.1, 2X, F4.2, 2X, I2, 3X, A40)

colonne

1 - 3 IRANG = rang d'observation

6 - 13 T(I) = instant de l'observation, exprimé en années décimales

15 - 19 B = angle de position de l'étoile-compagnon, exprimé en degrés décimaux

22-25 DIST(I) = distance angulaire de l'étoile-compagnon, exprimée en secondes d'arc

28 - 29 NUIT = nombre de nuits d'observation

33- 72 OBSERV(I) = texte donnant le nom de l'observateur, l'instrument utilisé, etc.

- IDEB, IFIN, IPAS au format (I5, I6, I4)

IDEB = première année de l'éphéméride

IFIN = dernière année de l'éphéméride

IPAS = périodicité du calcul d'éphéméride, en années

LE PROGRAMME

Le programme comprend 814 lignes dont 516 lignes d'instructions et 298 lignes de commentaires.

*
* *

```

C          AMELIORATION DE L'ORBITE D'UNE ETOILE DOUBLE VISUELLE
C
C          PAR MINIMISATION DE LA SOMME DES CARRÉS DES ECARTS REDUITS
C
C          ENTRE LES GRANDEURS OBSERVEES ET CALCULEES :
C
C          ANGLES DE POSITION ET DISTANCES ANGULAIRES
C
C          ALGORITHME DE LEVENBERG-MARQUARDT
C
C          par Edgar Soulié
C
C Réfrence : "L'amélioration de l'orbite d'une étoile double visuelle"
C
C          ASTRONOMY & ASTROPHYSICS, volume 164 (1986) pages 408 à 414
C
C          logiciel en FORTRAN 77 portable
C
C          noms des unités de programme :
C
C          EXTERI   AMLORB   ENVRDN   COORDD   ETLAJU   RESLIN   ETLDBL
C
C          vendredi 03 septembre 1993
C-----
C          PROGRAM EXTERI
C          IENTRE = numéro de l'unité logique de lecture des données
C          ISOR = numéro de l'unité logique d'écriture des résultats
C          doit être différent de 6 sur "I.B.M."
C          .. Scalaires locaux ..
C          CHARACTER*30 FICENT , FICSOR
C          .. Scalaires en "Common" ..
C          INTEGER IENTRE , ISOR
C          ..
C          .. Sous-programmes appelés ..
C          EXTERNAL AMLORB
C          ..
C          .. Blocs "Common" ..
C          COMMON / FICLEC / IENTRE
C          COMMON / FICSOR / ISOR
C-----
C          IENTRE = 5
C          ISOR = 6
C          WRITE( * , * ) ' quel est le nom du fichier de données ? '
C          READ( * , '(A)' ) FICENT
C          OPEN( UNIT = IENTRE , FILE = FICENT )
C          WRITE( * , * ) ' quel est le nom du fichier de résultats ? '
C          READ( * , '(A)' ) FICSOR
C          OPEN( UNIT = ISOR , FILE = FICSOR )
C          CALL AMLORB
C--- CLOSE( UNIT = ISOR )
C          STOP
C          END
C
C          SUBROUTINE AMLORB
C-----
C          Sous-programme pilotant l'ensemble des calculs
C
C          Les données nécessaires à l'optimiseur ETLAJU et au simulateur ETLDBL
C          sont transmises par des instructions COMMON
C
C          Inverses des incertitudes :
C          DA : sur l'angle de position , exprimée en radians
C          DD : sur la distance angulaire , exprimée en secondes de degré
C
C          DRE = distance angulaire réduite, c'est-à-dire divisée par
C          l'incertitude sur la distance angulaire
C          MAXOBS = nombre maximum d'observations
C          NIMP = périodicité de l'impression de la somme des carrés des écarts
C          NMAX = nombre maximum d'appels au sous-programme "ETLDBL"
C-----
C          INTEGER IDEB, IENTRE, IFIN, IPAS, ISOR, K, NB, NIMP, NMAX ,
C          & IBETA , IT , MAXOBS
C          PARAMETER( MAXOBS = 136 )
C          .. Scalaires présents dans des instructions "Common" ..
C          REAL DEMAXE , DEUPI , EPSILL , OMEGA , PASSAG , PETOME ,
C          & PIS180 , PRECIS , S , XINCLI , XMU
C          ..
C          .. Tableaux présents dans des instructions "Common" ..
C          REAL ANG(MAXOBS) , DA(MAXOBS) , DD(MAXOBS) , DIST(MAXOBS) ,
C          & DRE(MAXOBS) , Q(9.2*MAXOBS) , T(MAXOBS)

```

```

C CHARACTER OBSERV(MAXOBS)*40 26
C .. Scalaires Locaux ..
REAL AINCLI , ANOEUD , EXCENT , PERIAS , PERIOD 27
CHARACTER TITRE*80 28
C ..
C .. Sous-programmes appelés ..
EXTERNAL COORDO , ENVRON , EPHEMR , ETLAJU , ETLFIN 29
C ..
C .. Fonctions Intrinsèques ..
INTRINSIC DATAN , SNGL , SQRT 30
C ..
C .. Blocs "Common" ..
COMMON / COBSER / OBSERV 31
COMMON / COOBL / PIS180 , IDEB , IFIN , IPAS 32
COMMON / CO / Q 33
COMMON / CX / XMU , PASSAG , S , DEMAXE , XINCLI , PETOME , 34
& OMEGA , NB , K 35
COMMON / FICLEC / IENTRE 36
COMMON / FICSOR / ISOR 37
COMMON / PARAJU / DRE , EPSILL , NIMP , NMAX 38
COMMON / TRSFRT / T , ANG , DA , DIST , DD , PRECIS , DEUPI 39
C-----
C--- constantes mathématiques utilisables sur toutes machines
PIS180 = SNGL( DATAN( 1.DO ) / 45.DO ) 40
DEUPI = SNGL( DATAN( 1.DO ) * 8.DO ) 41
WRITE( ISOR , FMT=9000 ) PIS180 , DEUPI 42
9000 FORMAT( /,/, PIS180 = ' , F22.20 , ' DEUPI = ' , F22.20 ) 43
C-----
C EPSILL = plus petit réel, à un facteur 2 près, tel qu'en machine
C 1. + EPSILL > 1.
C PRECIS = critère d'arrêt des itérations pour
C la résolution de l'équation de Képler
C Pour une valeur trop petite de PRECIS, le critère de convergence
C vers la solution n'est jamais satisfait.
C Le coefficient 20 a été déterminé empiriquement :
C Un coefficient égal à 16 est trop petit. Pour un coefficient égal
C à 20 nous n'avons pas rencontré d'arrêt par le contrôle du temps
C dans la boucle implicite du sous-programme "ETLDBL" utilisée pour
C résoudre l'équation de Képler
C-----
CALL ENVRON( IBETA , IT , EPSILL ) 44
WRITE( ISOR , FMT= 900 ) IBETA , IT , EPSILL 45
900 FORMAT( /, ' base de numération interne = ' , I2 , 5X , ' nombre de ' , 46
& ' chiffres de la mantisse = ' , I2,/,/, ' plus petit réel ' , 47
& ' EPSILL tel qu'en machine : 1 + EPSILL > 1 ' , 48
& E12.6 , /, / ) 49
PRECIS = 20. * EPSILL 50
C-----
NIMP = 1 51
NMAX = 19 52
READ( IENTRE , FMT= 910 ) TITRE 53
910 FORMAT( A80 ) 54
WRITE( ISOR , FMT= 920 ) TITRE 55
920 FORMAT( '1' , A80 ,/,/) 56
C valeurs estimées des paramètres
READ( IENTRE , FMT= * ) PERIOD , PASSAG , EXCENT , DEMAXE , 57
& AINCLI , PERIAS , ANOEUD 58
WRITE( ISOR , FMT= 930 ) PERIOD , PASSAG , EXCENT , DEMAXE , 59
& AINCLI , PERIAS , ANOEUD 60
930 FORMAT( 'PERIODE = ' ,F7.2,' INSTANT DU PASSAGE AU PERIASTRE = ' 61
& ,F7.2,' EXCENTRICITE = ' ,F7.5,' DEMI GRAND AXE = ' , 62
& F8.5,/,/, ' INCLINAISON = ' ,F9.4,' ARGUMENT DU PERIASTRE', 63
& ' ,F9.4,' ANGLE DE POSITION DE LA LIGNE DES NOEUDS = ' , 64
& F9.4 ) 65
XMU = DEUPI / PERIOD 66
S = SQRT( EXCENT / ( 1. - EXCENT ) ) 67
C----- conversion des angles en radians
XINCLI = PIS180 * AINCLI 68
PETOME = PIS180 * PERIAS 69
OMEGA = PIS180 * ANOEUD 70
C-----lecture des observations
CALL COORDO(T, ANG, DA, DIST, DRE, DD, OBSERV, MAXOBS, NB ) 71
C--en sortie de COORDO, les angles de position sont exprimés en radians 72
CLOSE( UNIT = IENTRE )
C
CALL ETLAJU 73
CALL ETLFIN 74
CALL EPHEMR 75
RETURN 76

```

```

C      END 77
C
C      SUBROUTINE ENVRON( IBETA , IT , EPSILL ) 78
C-----
C
C          d'après William J. CODY
C
C      "ALGORITHM 665 MACHAR : A Subroutine to dynamically determine machine
C          parameters"
C
C      ACM Trans. Math. Soft. Vol.14, n 4, décembre 1988, pp.303-311
C
C      IBETA = base de la représentation interne des nombres en machine
C
C      IT = nombre de chiffres dans la base IBETA de représentation
C
C      EPSILL = plus petit réel tel qu'en machine : 1. + EPSILL > 1.
C          à UN FACTEUR 2 PRÈS
C-----
C
C      .. Arguments Scalaires ..
C      INTEGER IBETA , IT 79
C      REAL EPSILL 80
C
C      .. Fonctions Intrinsèques ..
C      INTRINSIC INT , REAL 81
C
C      .. Scalaires Locaux ..
C      INTEGER K 82
C      REAL A , B , BETA , C , D , UN , ZERO 83
C-----
C      UN = REAL(1) 84
C      ZERO = UN - UN 85
C-----
C      détermination de IBETA et de BETA par l'algorithme de Malcolm
C-----
C
C      A = UN 86
C      10 A = A + A 87
C      C = A + UN 88
C      D = C - A 89
C      IF( D - UN .EQ. ZERO ) GO TO 10 90
C      B = UN 91
C      20 B = B + B 92
C      C = A + B 93
C      K = INT( C - A ) 94
C      IF( K .EQ. 0 ) GO TO 20 95
C      IBETA = K 96
C      BETA = REAL( IBETA ) 97
C-----
C      détermination de IT et de EPSILL
C-----
C
C      IT = 0 98
C      B = UN 99
C      30 IT = IT + 1 100
C      B = B * BETA 101
C      C = B + UN 102
C      D = C - B 103
C      IF( D - UN .EQ. ZERO ) GO TO 30 104
C      EPSILL = BETA ** ( 1 - IT ) 105
C      RETURN 106
C      END 107
C
C      SUBROUTINE COORDO( T , ANG , DA , DIST , DRE , DD , OBSERV , N , NB ) 108
C-----
C      lecture des observations d'étoiles doubles visuelles
C      T = époque , exprimée en années décimales
C      B = angle de position du compagnon, lu en degrés décimaux
C      ANG = angle de position du compagnon, stocké en radians
C      DIST = distance angulaire du compagnon, en secondes de degré
C      N = nombre maximum d'observations
C      NB = nombre d'observations lues dans le fichier de données
C      EA = incertitude sur l'angle de position, lue en degrés décimaux
C      ER = incertitude sur la distance angulaire, lue en secondes de degré
C      DA = inverse de l'incertitude sur l'angle de position, exprimée en radians
C      DD = inverse de l'incertitude sur la distance angulaire,
C          en secondes de degré
C      DRE = distance angulaire réduite, c'est-à-dire divisée par l'incertitude
C      J = numéro d'ordre d'une observation
C-----

```



```

C
C NIMP = périodicité de l'impression de la somme des carrés des écarts
C-----
C .. Constantes locales ..
INTEGER MAXOBS                                156
REAL PEU , U , ZERO                            157
PARAMETER( MAXOBS = 136 , PEU = 1.E-5 , U = 5. , ZERO = 0. ) 158
C .. Scalaires en "Common" ..
INTEGER ISOR , K , NB , NIMP , NMAX           159
REAL EPSILL                                   160
C
C .. Tableaux en "Common" ..
REAL DRE(MAXOBS) , Q(9.2*MAXOBS) , X(7)      161
C
C .. Scalaires Locaux ..
INTEGER I , ITN , J , L , N , NBPI , NBPN , NY 162
REAL AUX , PHI , S , SOMCAR                  163
C
C .. Tableaux Locaux ..
REAL G(7)                                     164
C
C .. Sous-programmes appelés ..
EXTERNAL ETLDBL , RESLIN                      165
C
C .. Fonctions intrinsèques ..
INTRINSIC ABS , MOD , SQRT                   166
C
C .. Blocs en "Common" ..
COMMON / CQ / Q                                167
COMMON / CX / X , NB , K                      168
COMMON / FICSOR / ISOR                        169
COMMON / PARAJU / DRE , EPSILL , NIMP , NMAX 170
C-----
K = 0                                         171
NY = 2 * NB                                  172
S = 0.01                                     173
ITN = 0                                       174
DO 20 I = 1 , NY                              175
  DO 10 J = 1 , 9                              176
    Q(J,I) = 0.                                177
10 CONTINUE                                   178
20 CONTINUE                                   179
30 CALL ETLDBL                                 180
C-----
C calcul de la somme des carrés des écarts réduits
C Pour chaque observation, on ajoute l'écart réduit relatif à
C la distance angulaire et l'écart réduit relatif à l'angle de positio
C-----
SOMCAR = 0.                                   181
DO 40 I = 1 , NB                              182
  NBPI = NB + I                                183
  SOMCAR = SOMCAR + (DRE(I) - Q(9,I))**2 + Q(9,NBPI)**2 184
40 CONTINUE                                   185
IF( MOD( K , NIMP ) .EQ. 0 ) WRITE( ISOR , FMT = 900 ) K , SOMCAR 186
900 FORMAT( ' NUMERO ' , I4 , ' SOMME DES CARRÉS DES ECARTS = ' , 187
  & E21.15 , '/./')
IF( SOMCAR .LE. ZERO ) GO TO 230              189
ITN = ITN + 1                                 190
IF( ITN .GT. 1 ) GO TO 210                   191
C----- premier passage
PHI = SOMCAR                                  192
C----- calcul du gradient 'G'
50 DO 70 I = 1 , 7                            193
  AUX = 0.                                    194
  DO 60 N = 1 , NB                            195
    NBPN = NB + N                             196
    AUX = AUX + (DRE(N)-Q(9,N))*Q(I,N) - Q(9,NBPN)*Q(I,NBPN) 197
60 CONTINUE                                   198
  Q(8,I) = AUX                                 199
70 CONTINUE                                   200
Construction de la matrice symétrique A
DO 120 I = 1 , 7                              201
  DO 100 J = 1 , 7                             202
    Q(9,J) = 0.                                203
    IF( J .LT. I ) GO TO 90                   204
    DO 80 L = 1 , NY                          205
      Q(9,J) = Q(9,J) + Q(I,L) * Q(J,L)      206
80 CONTINUE                                   207
GO TO 100                                     208
90 Q(9,J) = Q(J,I)                            209

```

```

100 CONTINUE
    DD 110 L = 1 , 7
        Q(I,L) = Q(9,L)
110 CONTINUE
120 CONTINUE
C----- Calcul de 'G*'
    DD 140 I = 1 , 7
        IF( Q(I,I) .NE. ZERO ) GO TO 130
        WRITE( ISOR , FMT= 910 ) I
910 FORMAT ( /, / LA FONCTION EST INDEPENDANTE DU PARAMETRE NO',I4)
        RETURN
130 Q(9,I) = SORT(Q(I,I))
    Q(8,I) = Q(8,I) / Q(9,I)
140 CONTINUE
C----- Construction de la matrice A* + S.I
    DD 190 I = 1 , 7
        DD 180 J = 1 , 7
            IF (I-J) 160 , 150 , 170
150 Q(I,J) = 1. + S
        GO TO 180
160 Q(J,I) = Q(J,I) / ( Q(9,I) * Q(9,J) )
        GO TO 180
170 Q(J,I) = Q(I,J)
180 CONTINUE
190 CONTINUE
    CALL RESLIN
C----- Stockage anciens paramètres et calcul nouveaux paramètres
    DD 200 I = 1 , 7
        Q(8,I) = Q(8,I) / Q(9,I)
        G(I) = X(I)
        X(I) = X(I) + Q(8,I)
200 CONTINUE
    GO TO 30
C----- Test de convergence
210 DD 220 I = 1 , 7
        IF( ABS(Q(8,I)) / ( PEU + ABS(X(I)) ) .GE. EPSILL ) GO TO 240
220 CONTINUE
230 RETURN
240 IF( SOMCAR .GE. PHI ) GO TO 250
C---- PHI diminue
    S = S / U
    PHI = SOMCAR
    GO TO 50
C---- PHI augmente
250 S = S * U
    ITN = ITN - 1
    DD 260 I = 1 , 7
        X(I) = G(I)
260 CONTINUE
    CALL ETLDBL
    IF( K .GT. NMAX ) GO TO 230
    GO TO 50
    END
C
SUBROUTINE RESLIN
C-----
C Résolution de systèmes linéaires à éléments réels
C A matrice et seconds membres (remplacés par les solutions)
C K bloc de travail
C-----
C .. Constante locale ..
INTEGER MAXOBS
PARAMETER( MAXOBS = 136 )
C .. Scalars en "Common" ..
INTEGER ISOR
C .. Tableaux en "Common" ..
REAL A(9,2*MAXOBS)
C ..
C .. Scalars Locaux ..
INTEGER I, I1, I2, I3, IN, IT, J, J2, J3, JMAX, KC, MP, N, NAB, NDEB, NM
REAL AMAX, AUX, ERA, P, S, T
C ..
C .. Tableaux Locaux ..
INTEGER K(7)
C ..
C .. Fonctions Intrinsèques ..
INTRINSIC ABS
C ..
C .. Blocs en "Common" ..

```

```

COMMON / CQ / A
COMMON / FICSOR / ISOR
C
N = 7
DO 10 I = 1 , N
  K(I) = I
10 CONTINUE
NDEB = N + 1
NM = N + 1
DO 130 I = 1 , N
C----- recherche du pivot maximum
  AMAX = ABS(A(I,I))
  JMAX = I
  I1 = I + 1
  IF (I1.GE.N) GO TO 30
  DO 20 J = I1 , N
    IF (AMAX .GT. ABS(A(J,I))) GO TO 20
    AMAX = ABS(A(J,I))
    JMAX = J
  20 CONTINUE
C----- matrice singulière
  30 IF (AMAX .GT. 0.) GO TO 40
  WRITE( ISOR , FMT= 900 )
  900 FORMAT ( ' MATRICE SINGULIERE ' )
  STOP
C----- transport de la colonne
  40 IF ( JMAX .LE. I ) GO TO 60
  DO 50 I2 = 1 , N
    AUX = A(I,I2)
    A(I,I2) = A(JMAX,I2)
    A(JMAX,I2) = AUX
  50 CONTINUE
  60 IF ( I.LE.1 ) GO TO 80
C----- test sur la singularité de la matrice
  S = 0.
  T = 0.0
  IN = I - 1
  DO 70 IT = 1 , IN
    P = A(I,IT) * A(IT,I)
    S = S + P
    T = T + ABS(P)
  70 CONTINUE
  ERA = 1.E-6 * ( T + ABS( A(I,I) - S ) )
  IF ( AMAX .GT. ERA ) GO TO 80
  WRITE( ISOR , FMT= 910 )
  910 FORMAT ( ' MATRICE QUASI SINGULIERE ' )
  RETURN
C----- division par le pivot
  80 DO 90 J2 = I1 , NM
    A(J2,I) = A(J2,I) / A(I,I)
  90 CONTINUE
C----- Substitution des lignes
  IF ( I .GE. N ) GO TO 120
  DO 110 I3 = I1 , N
    DO 100 J3 = I1 , NM
      A(J3,I3) = A(J3,I3) - A(I,I3) * A(J3,I)
    100 CONTINUE
  110 CONTINUE
C----- sortie des indices
  120 IF ( JMAX .LE. I ) GO TO 130
  NAB = K(JMAX)
  K(JMAX) = K(I)
  K(I) = NAB
  130 CONTINUE
  IF ( N .EQ. 1 ) RETURN
C----- Calcul des solutions
  DO 160 KC = NDEB , NM
    J = N
    140 I = J - 1
    150 A(KC,I) = A(KC,I) - A(KC,J) * A(J,I)
    I = I - 1
    IF ( I .GT. 0 ) GO TO 150
    J = J - 1
    IF ( J .GT. 1 ) GO TO 140
  160 CONTINUE
C----- Classement des solutions
  DO 170 I = 1 , N
    J = K(I)
    170 IF ( J .LE. I ) GO TO 190
    K(I) = K(J)

```

```

      K(J) = J                                336
      DD 180 MP = NDEB , NM                    337
      AUX = A(MP,J)                           338
      A(MP,J) = A(MP,I)                       339
      A(MP,I) = AUX                            340
180   CONTINUE                               341
      GO TO 170                               342
190   CONTINUE                               343
      RETURN                                  344
      END                                     345
C
C   SUBROUTINE ETLDBL                          346
C-----
C   calcul d'éphéméride d'étoile double ( "THETA" et "RO" en fonction
C   de l'époque) et calcul des dérivées partielles de "THETA" et de "RO"
C   par rapport à sept paramètres internes. De ces paramètres internes,
C   on déduit les sept éléments de Campbell
C
C   tous les angles sont exprimés en radians dans le point d'entrée
C   principal ETLDBL
C   inverses des incertitudes :
C   DA : sur l'angle de position , exprimée en radians
C   DD : sur la distance angulaire , en secondes de degré
C-----
C   INTEGER MAXOBS                             347
C   PARAMETER( MAXOBS = 136 )                  348
C
C   .. Scalaires en "Common" ..
C   INTEGER IDEB , IFIN , IPAS , ISOR , K , NB   349
C   REAL   DEMAXE , DEUPI , OMEGA , PASSAG , PETOME , PIS180 ,
C   &      PRECIS , S , XINCLI , XMU           351
C
C   .. Tableaux en "Common" ..
C   REAL   ANG(MAXDES) , DA(MAXOBS) , DD(MAXOBS) , DIST(MAXOBS) ,
C   &      O(9,2*MAXOBS) , T(MAXOBS)         352
C   CHARACTER OBSERV(MAXOBS)*40              353
C
C   .. Scalaires Locaux ..
C   REAL A , AINCLI , AKAP , AKHI , ALAMB , ALPHA , AMU , ANDEUD , ANU , API ,
C   &      ARGUM , B , BETA , C , COSINC , COSV , DELTA , DESDS , DFSDE , DPSDA ,
C   &      DPSDE , DRMDMU , DRMDT , DVSDE , ECANG , ECDIS , EPOQUE , EPSIL ,
C   &      ETA , EXCENT , FACT , G , GAMMA , D , PARAME , PERIAS ,
C   &      PERIOD , PHI , R , RM , RO , SIGMA , SININC ,
C   &      SINKSI , TAU , THETA , TRCSP2 , U , UNSIGM , V , W , XKSI , Y , Z , ZZ
C   INTEGER I , J , NBPI                      361
C
C   .. Fonctions Intrinsèques ..
C   INTRINSIC ABS , ATAN , ATAN2 , COS , DATAN , MOD , NINT , REAL ,
C   &      SIN , SNGL , SQRT , TAN            362
C
C   .. Blocs en "Common" ..
C   COMMON / COBSER / OBSERV                  364
C   COMMON / COOBDL / PIS180 , IDEB , IFIN , IPAS 365
C   COMMON / CO / Q                            366
C   COMMON / CX / XMU , PASSAG , S , DEMAXE , XINCLI , PETOME ,
C   &      OMEGA , NB , K                      367
C   COMMON / FICSOR / ISOR                    368
C   COMMON / TRSFRT / T , ANG , DA , DIST , DD , PRECIS , DEUPI 369
C   COMMON / TRSFRT / T , ANG , DA , DIST , DD , PRECIS , DEUPI 370
C
C   .. Fonctions Instructions ..
C   REAL RESTE                                371
C
C   .. Instruction "Save" ..
C   SAVE COSINC , DPSDA , EXCENT , FACT , PARAME , PERIOD
C   ..                                         372
C-----
C   Définition d'une fonction-instruction :
C   reste de la division de A par B compris entre
C   B *( C - 0.5 ) et B *( C + 0.5 )
C   RESTE(A,B,C) = A - B * REAL( NINT( A / B - C ) )
C-----
C   K = K + 1                                374
C   IF( K .EQ. 2 ) WRITE( ISOR , FMT= 900 )    375
C   900 FORMAT( 'ONUMERO',3X,'PERIODE',3X,'PASSAGE', 6X , 'EXCENTRICITE'
C   &      6X , 'DEMI GRAND AXE' , 3X , 'INCLINAISON' , 3X ,
C   &      'ANGLE PERIASTRE' , 3X , 'ANGLE LIGNE DES NOEUDS' )
C   PERIOD = DEUPI / XMU                      376
C   Y = S * S                                 377
C   Z = 1. + Y                                378
C
C   ..                                         379
C   ..                                         380
C   ..                                         381

```

```

EXCENT = Y / Z
WRITE(ISOR , FMT= 910 ) K , PERIOD , PASSAG , EXCENT , DEMAXE ,
& XINCLI , PETOME , OMEGA
910 FORMAT( / , 14 , 4X , F7.2 , 3X , F7.2 , 6X , F7.2 , 10X ,
& F6.3 , 11X , F6.4 , 10X , F8.4 , 16X , F6.4 )
ZZ = Z * Z
DESDS = 2. * S / ZZ
FACT = SQRT( Y + Z )
DFSDE = ZZ / FACT
DPSDA = 1. - EXCENT * EXCENT
COSINC = COS( XINCLI )
SININC = SIN( XINCLI )
PARAME = DEMAXE * DPSDA
DPSDE = -2. * DEMAXE * EXCENT
DRMDT = -XMU
DO 30 I = 1 , NB
NBPI = NB + I
C----- calculs d'éphéméride
DRMDMU = T(I) - PASSAG
RM = XMU * DRMDMU
U = RM
C----- Résolution de l'équation de Képler : RM = U - E.SIN(U)
C Methode : calcul des termes successifs de la suite récurrente convergente
C U(N+1) = (F(U(N)) - U(N) * FPRIME(U(N))) / ( 1 - FPRIME(U(N)) )
C avec : F(X) = RM + EXCENT * SIN( X )
10 W = U
U = W + ( RM + EXCENT * SIN(W) - W ) / ( 1. - EXCENT * COS(W) )
IF( ABS(U-W) .GT. PRECIS ) GO TO 10
C-----
D = TAN( 0.5 * U )
G = FACT * D
V = 2. * ATAN(G)
PHI = 1. / ( 1. + G * G )
ANU = PHI * FACT * ( 1. + D * D ) / ( 1. - EXCENT * COS(U) )
DVSDE = ANU * SIN( U ) + 2. * PHI * D * DFSDE
C----- calculs relatifs à l'angle de position
XKSI = PETOME + V
ALPHA = COS( XKSI )
SINKSI = SIN( XKSI )
BETA = SINKSI * COSINC
C--- conversion des coordonnées cartésiennes en coordonnées polaires
C--- reste de la division par DEUPI compris entre -PI et +PI
C--- THETA = RESTE( OMEGA + ATAN2( BETA , ALPHA ) , DEUPI , 0. )
ECANG = RESTE( OMEGA + ATAN2(BETA,ALPHA)-ANG(I) , DEUPI , 0. )
SIGMA = ALPHA * ALPHA + BETA * BETA
TAU = SQRT( SIGMA )
AMU = SININC * SINKSI
UNSIGM = 1. / SIGMA
ETA = COSINC * UNSIGM
API = ETA * ANU
C----- calcul des dérivées partielles de l'angle de position
Q(1,NBPI) = API * DRMDMU
Q(2,NBPI) = API * DRMDT
Q(3,NBPI) = ETA * DVSDE * DESDS
Q(4,NBPI) = 0.
Q(5,NBPI) = -ALPHA * AMU * UNSIGM
Q(6,NBPI) = ETA
Q(7,NBPI) = 1.
C--- division de l'écart angulaire par l'incertitude correspondante
Q(9,NBPI) = ECANG * DA(I)
C-----
C----- Calculs relatifs à la distance angulaire
COSV = COS( V )
AKHI = 1. / ( 1. + EXCENT * COSV )
R = PARAME * AKHI
RD = R * TAU
GAMMA = -R * AMU / TAU
DELTA = GAMMA * SININC * ALPHA
EPSIL = TAU * AKHI
AKAP = R * EPSIL * EXCENT * SIN( V ) + DELTA
ALAMB = AKAP * ANU
C----- Calcul des dérivées partielles de la distance angulaire
Q(1,I) = ALAMB * DRMDMU
Q(2,I) = ALAMB * DRMDT
Q(3,I) = DESDS * ( AKAP * DVSDE + EPSIL * ( DPSDE - R * COSV ) )
Q(4,I) = EPSIL * DPSDA
Q(5,I) = GAMMA * BETA
Q(6,I) = DELTA
Q(7,I) = 0.
C----- Division de l'écart en distance angulaire par l'incertitude

```

```

          Q(9,I) = RO * DD(I)
-----
C-----multiplication des dérivées partielles par les inverses des incertitudes
      DD 20 J = 1 . 7
          Q(J,I) = Q(J,I) * DD(I)
          Q(J,NBPI) = Q(J,NBPI) * DA(I)
      20 CONTINUE
      30 CONTINUE
C-----Impression des observations, des valeurs calculées et des résidus
C--- pour les valeurs initiales des paramètres
      IF( K .EQ. 1 ) GO TO 40
      RETURN
-----
C
      Calculs finaux après ajustement
-----
      ENTRY ETLFIN
      40 CONTINUE
      WRITE( ISDR , FMT= 920 ) PERIOD , PASSAG , EXCENT , DEMAXE
      920 FORMAT( / , / , / , / , PERIODE = ' , F8.3 , ' INSTANT DU PASSAGE AU ' ,
      & ' PÉRIASTRE = ' , F8.3 , ' EXCENTRICITÉ = ' , F7.5 ,
      & ' DEMI GRAND AXE = ' , F7.4 )
C----- TRCSP2 = 3.C/P**2 : paramètre d'orbite parabolique
      TRCSP2 = SNGL(24.DO * DATAN(1.DO) ) / ( DPSDA**1.5 * PERIOD )
      WRITE( ISDR , FMT= 930 ) TRCSP2 , PARAME
      930 FDMAT( '03.C/P**2 = ' , F10.7 , 5X , 'PARAMÈTRE = ' , F8.5 )
C----- Conversion des angles en degrés
      AINCLI = RESTE( XINCLI / PIS180 , 180. , 0.5 )
      PERIAS = PETOME / PIS180
      ANDEUD = RESTE( OMEGA / PIS180 , 360. , 0.5 )
-----
C Si l'angle du noeud est compris entre 180 et 360 degrés, on lui
C retranche 180 degrés, et on ajoute simultanément 180 degrés à
C l'argument du périastre
-----
      IF( ANDEUD .LT. 180. ) GO TO 50
      ANDEUD = ANDEUD - 180.
      PERIAS = PERIAS + 180.
      50 CONTINUE
C On peut alors ramener l'argument du périastre entre 0 et 360 degrés
      PERIAS = RESTE( PERIAS , 360. , 0.5 )
      WRITE( ISDR , FMT= 940 ) AINCLI , PERIAS , ANDEUD
      940 FORMAT( '0ANGLES EN DEGRES' , / , / , ' INCLINAISON = ' , F9.5 , 3X , ' ARGUM' ,
      & ' ENT DU PÉRIASTRE ' , F9.5 , 3X , ' POSITION DE LA LIGNE' ,
      & ' DES NOEUDS = ' , F10.5 )
      WRITE( ISDR , FMT= 950 )
      950 FORMAT( '1' , / , / , / , / , 8X , 'EPOQUE' , 3X , 'ANGLE DE POSITION' , 6X , 'DISTA' ,
      & ' NCE ANGULAIRE' , 5X , 'OBSERVATEURS' , / , / , '0' , 16X , 'OBSERVE CALC' ,
      & ' ULE ECART' , 2X , 'OBSERVE CALCULE ECART' , / )
      DD 60 I = 1 , NB
      NBPI = NB + I
      ECANG = -Q(9,NBPI) / ( DA(I) * PIS180 )
      ARGUM = ANG(I) / PIS180
      THETA = RESTE( ARGUM - ECANG , 360. , 0.5 )
      RO = Q(9,I) / DD(I)
      ECDIS = DIST(I) - RO
      WRITE( ISDR , FMT= 960 ) I , T(I) , ARGUM , THETA , ECANG ,
      & DIST(I) , RO , ECDIS , OBSERV(I)
      960 FDMAT( ' ' , I3 , 3X , F8.3 , 2X , F5.1 , 3X , F5.1 , F7.1 , 2X ,
      & F5.2 , 3X , F5.2 , 4X , F5.2 , 3X , A40 )
      IF( MOD(I,5) .EQ. 0 ) WRITE( ISDR , FMT= 970 )
      970 FORMAT( '0' )
      60 CONTINUE
      RETURN
-----
C
      Ephéméride
-----
      ENTRY EPHEMR
      WRITE( ISDR , FMT= 980 )
      980 FORMAT( '1EPHEMERIDE' , / , / , ' EPOQUE' , ' ANGLE DE POSITION' ,
      & 2X , 'DISTANCE ANGULAIRE' )
      DO 80 I = IDEB , IFIN , IPAS
      EPOQUE = REAL(I)
      RM = XMU * ( EPOQUE - PASSAG )
      IF( I .EQ. 1 ) U = RM
C----- Résolution de l'équation de Képler : RM = U - E.SIN(U)
C Méthode : calcul des termes successifs de la suite récurrente convergent
C U(N+1) = ( F(U(N)) - U(N) * FPRIME(U(N)) ) / ( 1 - FPRIME(U(N)) )
C avec : F(X) = RM + EXCENT * SIN( X )
      70 W = U
      U = W + ( RM + EXCENT * SIN(W) - W ) / ( 1. - EXCENT * COS(W) )

```

```

          IF( ABS(U-W) .GT. PRECIS ) GO TO 70          504
C-----
      V = 2. * ATAN( FACT * TAN( 0.5 * U ) )          505
      XKSI = PETDME + V                               506
      ALPHA = COS( XKSI )                             507
      BETA = SIN( XKSI ) * COSINC                     508
      THETA = RESTE( OMEGA + ATAN2(BETA,ALPHA) , DEUPI , 0.5 )/ PIS180 509
      R = PARAME./ ( 1. + EXCENT * COS( V ) )         510
      RD = R * SQRT( ALPHA **2 + BETA **2 )           511
      WRITE( ISOR , FMT = 990 ) EPOQUE , THETA , RD   512
990 FORMAT( 1X , F8.3 , 6X , F6.2 , 12X , F6.3 )     513
      80 CONTINUE                                     514
      RETURN                                          515
      END                                             516

```