Адаптация алгоритмов решения задач небесной механики для GPU.

Mikhail Zakhvatkin

19 февраля 2013 г.

Содержание

1 Интегрирование уравнений движения

В общем виде система уравнений записывается в виде:

$$\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t).$$

Интегрирование ведется от начальный условий $\mathbf{x}(t_0) = \mathbf{x_0}$. В общем случае функция правых частей может зависеть не только от вектора состояния $\mathbf{x}$ и времени t, но также от характеристик объекта, обуславливающих световое давление, сопротивление атмосферы и пр.

1.1 Алгоритм интегрирования системы ОДУ.

Оригинальный файл с описание алгоритма: rk8cc.pdf

1.2 Уравнения правых частей.

Центральным членом функции правых частей уравнений движения является гравитационное возмущение материальной точки. В рассматриваемых задачах исследуется движение вокруг Земли, следовательно основной составляющей является притяжение вызванное центральным гравитационным полем Земли

$$\mathbf{f_0}(\mathbf{x}, t) = -\frac{\mu}{|\mathbf{x}|^3}\mathbf{x}.$$

Уравнения, ограничивающиеся только центральным полем единственного притягивающего тела, описывают кеплеровское движение. Более точные модели движения вводят так называемые возмущения. Возмущения могут вызываться несферичностью гравитационного поля Земли (неоднородным распределением массы), воздействием других гравитирующих тел, например, Солнца и Луны, световым давлением, сопротивлением атмосферы, твердые и океанические приливы Земли, релятивистские поправки к классическим уравнениям движения и пр. При этом функция в правых частях уравнений движения имеет вид

$$\mathbf{f}(\mathbf{x}, t) = -\frac{\mu}{|\mathbf{x}|^3}\mathbf{x} + \mathbf{f_{harm}} + \mathbf{f_{sun}} + \mathbf{f_{moon}} + \mathbf{f_{sp}} + \mathbf{f_{atm}} + \ldots$$

Каждое из возмущений имеет ряд актуальных моделей и различную сложность реализации.

1.2.1 Возмущение от Луны, Солнца и планет солнечной системы.

В подавляющем большинстве случаев подобные возмущения адекватно описываются в предположении точечности возмущающих тел. Действие тела B на объект с радиус-вектором $\mathbf{x}$ опиывается уравнением

$$\mathbf{f_B} = \mu_B \left(\frac{\mathbf{r_B} - \mathbf{r}}{|\mathbf{r_B} - \mathbf{r}|^3} - \frac{\mathbf{r_B}}{|\mathbf{r_B}|^3} \right),$$

где $\mathbf{r_B}$ – радиус-вектор притягивающего тела относительно центра Земли. Второе слагаемое обусловлено действием притягивающего тела на Землю. Система отсчета, в которой проводится интегрирование, хоть и не меняет направления своих осей в пространстве, однако движется вместе с Землей и не является инерциальной, как, например, та, что была бы помещена в барицентр солнечной системы. Если просуммировать возмущения от Луны, Солнца и планет, то все “вторые слагаемые” сложатся в величину, очень близкую к реальному ускорению центра масс Земли. В данном случае предлагается учитывать только Солнце и Луну, поэтому ускорение Земли будет рассчитываться чуть менее точно. Возможно, ускорение Земли удобнее будет получать непосредственно из эфемерид планет.

Фрагмент модуля, отвечающий за расчет возмущения от Луны, Солнца и Планет

```
subroutine planets_grav(x, f_gr)
  implicit none
  real*8, intent(in) :: x(3)
  real*8, dimension(3), intent(out) :: f_gr(3)
  real*8 BC(3), BO(3), BC_r, BO_r
  integer i
  f_gr = 0.d0
  do i = 1,11
     if (pln_flag(i)) then
        BC = pln_coords(i,:) ! body - center
        BC_r = dot_product(BC,BC)
        if (BC_r.LT.1.d-12) then
           ! Начало координат в текущем притягивающем центре
           BO = BC-x ! body - object
           BO_r = 1.d0/dot_product(BO,BO)
           BO_r   = BO_r*sqrt(BO_r)
           f_gr = f_gr+mu_plan(i)*BO*BO_r
        else
           ! Накало координат не в притягивающем центре
           BO = BC-x ! body - object
           BC_r = 1.d0/BC_r
           BO_r = 1.d0/dot_product(BO,BO)
           BC_r   = BC_r*sqrt(BC_r)
           BO_r   = BO_r*sqrt(BO_r)
           f_gr = f_gr+mu_plan(i)*(BO*BO_r-BC*BC_r)
        end if
     end if
  end do
end subroutine planets_grav
```

Естественно, для расчета этого возмущения необходимо знать положение возмущающих тел на момент времени t. Аналитические теории в данном случае неприменимы. Результатом численных теорий движения планет являются либо непосредственно координаты нужной планеты, либо побочные величины, позволяющие эти координаты получить, с определенным шагом по времени. Нужное значение на момент времени t получается интерполяцией по соседним точкам. Наиболее точные эфемериды (данные о движении планет) на сегодня получены лабораторией JPL в Калифорнии.

Они называются DE4** (DE421, DE405, DE403 и т.д.). Для околоземных аппаратов мы используем DE403.

DE403

Бинарный файл эфемерид представляет набор блоков по 5968 байт (746 double'ов). Каждый блок покрывает интервал длительностью 32 суток (сутки = 86400 сек. атомного времени). Первое значение массива блока - номер блока блока, в целях унификации он записан как переменная double, т.е. необходимо взять ближайшее целое. Второе значение – юлианская дата, соответствующая началу интервала, который покрывает блок. Начиная с 3-его элемента идут коэффициенты разложения интерполяционных полиномов по полиномам Чебышева.

Каждой планете отведен определенный кусок массива. Например, Меркурию отведены элементы с 3 по 146 (см. таблицу). Значение Δt показывает длину подынтервала, для Меркурия это 8 суток. Т.к. в блоке содержится информация по 32 суткам, то блок Меркурия содержит 4 мини-блока по 8 суток. Координаты планеты на мини-интервале описываются полиномами, разложенными по базису полиномов Чебышева вплоть до степени Degree, к примеру для Меркурия это полиномы до степени 11. Имеется в виду подобное разложение

$$x(t) = \sum_{i=0}^{Degree} \alpha_i T_i(t),$$

где $T_i(t)$ – полином Чебышева степени i

Итого, для Меркурия каждый из четырёх мини-итервалов длиной 8 суток описывается полиномами степени 11 (разложением по полиномам Чебышева от 0 до 11 степени, всего 12 коэффициентов). Интерполируется три координаты, таким образом для этого необходимо 3*(11+1)*4 = 144 значений. Внутри мини-блока сначала записано %Degree коэффициентов для полинома X-координаты, потом столько же для полинома Y-координаты, Z-координаты.

Planet	Degree	Offset	Δ t
Меркурий	11	3	8.0
Венера	11	147	32.0
Зелмя+Луна	14	183	16.0
Марс	9	273	32.0
Юпитер	8	303	32.0
Сатурн	7	330	32.0
Уран	7	354	32.0
Нептун	5	378	32.0
Плутон	5	396	32.0
Луна*	11	414	4.0
Солнце	14	702	32.0

Пример фортрановской процедуры, которая считывает коэффициенты полиномов из файла.

```
subroutine DE403(t, n_pl, mode, x_pl)
  use time_reference, only : ajd0, delt0
  use io_utils, only : get_free_unit
  implicit none
  real*8, intent(in) :: t
  integer, intent(in) :: n_pl, mode
  real*8, intent(out) :: x_pl(*)
  real*8, parameter :: jd50      = 2433282.5d0
  real*8, parameter :: day_begin =  -54770.d0
  real*8, parameter :: day_end   =   59726.d0
  real*8, parameter :: span      =      32.d0
  real*8, parameter :: daysec    = 2.314814814815d-5
  real*8, save :: buffer(746)
  integer clu, rec_num, c_pl, i, j, knot, degree, ierr, un
  real*8 dif, t_span, arg, junk1, junk2, day
  real*8, dimension(11), parameter :: intp = [ &
       8.d0, 32.d0, 16.d0, 32.d0, 32.d0, 32.d0, &
       32.d0, 32.d0, 32.d0, 4.d0, 32.d0]
  integer, dimension(11), parameter :: p = [ &
       3, 147, 183, 273, 303, 330, 354, 378, 396, 414, 702]
  integer, dimension(11), parameter :: c = [ &
       12, 12, 15, 10, 9, 8, 8, 6, 6, 12, 15]
  integer, parameter :: first_rec = 1
  day = dnint(2.d0*(ajd0-jd50))/2.d0+(t+delt0)/86.4d0
  rec_num = int((day-day_begin)/span)
  clu = rec_num+first_rec
  call get_free_unit(un)
  if (clu.ne.nint(buffer(1))) then
     open (un, file=trim(DE_path), access='direct', recl=5968, &
          iostat=ierr, form='binary')
     if (ierr.ne.0) then
        print '(x,a34,x,i4)' , "Cannot open DE403 file. IOSTAT is", ierr
        return
     end if
     read (un, rec=(rec_num+first_rec), iostat=ierr) buffer
     if (ierr.ne.0) then
        print '(x,a39,x,i4)' , "Cannot read from DE403 file. IOSTAT is", ierr
        return
     end if
     close (un)
  end if
  dif = nint(2.d0*(ajd0-jd50-buffer(2))/2.d0)+(t+delt0)/86.4d0
  c_pl   = c(n_pl)
  degree = c_pl-1
  t_span = intp(n_pl)
  knot   = int(dif/t_span)
```

```
  arg    = 2.d0*(dif/t_span-knot)-1.d0
  j      = p(n_pl)+(3*knot-1)*c_pl
  t_span = daysec/t_span
  select case (mode)
  case (1)
     do i = 1,3
        call cheb2(1, arg, degree, buffer(j+i*c_pl), x_pl(i), junk1, junk2)
        x_pl(i)   = x_pl(i)*1.d-3
     end do
  case (2)
     do i = 1,3
        call cheb2(2, arg, degree, buffer(j+i*c_pl), x_pl(i), x_pl(i+3), junk2)
        x_pl(i)   = x_pl(i)*1.d-3
        x_pl(i+3) = x_pl(i+3)*t_span
     end do
  case (3)
     do i = 1,3
        call cheb2(3, arg, degree, buffer(j+i*c_pl), x_pl(i), x_pl(i+3), x_pl(i+6))
        x_pl(i)   = x_pl(i)*1.d-3
        x_pl(i+3) = x_pl(i+3)*t_span
        x_pl(i+6) = x_pl(i+6)*t_span*t_span*1.d3
     end do
  end select
end subroutine DE403
```

В процедуре интервал интерполяции приводится к интервалу $[-1, 1]$, на котором применимы полиномы Чебышева. На этом интервале определяется точка, соответствующая времени интерполяции. После того как известны коэффициенты разложения по полиномам Чебышева и значение аргумента идет обращение к процедуре cheb2. Эта процедура рассчитывает значение разложенного полинома в данной точке, а также его первую и вторую производную (скорость и ускорение) при необходимости. Для определения значений и производных полинома нет необходимости напрямую суммировать полиномы Чебышева со значениями в конкретной точке. По рекуррентной формуле высчитываются вспомогательные значения

$$b_k = \alpha_i - b_{k+2} + 2b_{k+1}t, \quad k = n, 1, \quad b_{n+2} = b_{n+1} = 0,$$

где α_i – коэффициенты разложения, а n - степень полинома. Тогда значение полинома равно

$$x = \alpha_0 - b_2 + b_1 t.$$

Аналогичные рекуррентные формулы есть для производных в данной точке.

Пример процедуры cheb2

```
subroutine cheb2 (mode, x, degree, coeff, w, dw, ddw)
  implicit none
  integer, intent(in) :: mode, degree
  real*8, intent(in) :: x
  real*8, dimension(degree+1) :: coeff(0:degree)
  real*8, intent(out) :: w, dw, ddw
  integer i
  real*8 bk, bk1, bk2, dbk, dbk1, dbk2, ddbk, ddbk1, ddbk2
  select case (mode)
  case (1)
     bk1 = 0d0
     bk2 = 0d0
     do i = degree,1,-1
        bk  = coeff(i)-bk2+2.d0*bk1*x
        bk2 = bk1
        bk1 = bk
     end do
     w   = coeff(0)-bk2+bk1*x
     dw  = 0.d0
     ddw = 0.d0
  case (2)
     bk1  = 0d0
     bk2  = 0d0
     dbk1 = 0d0
     dbk2 = 0d0
     do i = degree,1,-1
        bk   = coeff(i)-bk2+2.d0*bk1*x
        bk2  = bk1
        bk1  = bk
        dbk  = 2.d0*(dbk1*x+bk2)-dbk2
        dbk2 = dbk1
        dbk1 = dbk
     end do
     w   = coeff(0)-bk2+bk1*x
     dw  = bk1+dbk1*x-dbk2
     ddw = 0.d0
  case (3)
     bk1   = 0d0
     bk2   = 0d0
     dbk1  = 0d0
     dbk2  = 0d0
     ddbk1 = 0d0
     ddbk2 = 0d0
     do i = degree,1,-1
        bk = coeff(i) - bk2 + 2.d0*bk1*x
        bk2   = bk1
```

```
      bk1   = bk
      dbk   = 2.d0*(dbk1*x+bk2)-dbk2
      dbk2  = dbk1
      dbk1  = dbk
      ddbk  = 2.d0*(ddbk1*x+2.d0*dbk2 )-ddbk2
      ddbk2 = ddbk1
      ddbk1 = ddbk
    end do
    w   = coeff(0)-bk2+bk1*x
    dw  = bk1+dbk1*x-dbk2
    ddw = 2d0*dbk1+ddbk1*x-ddbk2
  end select
end subroutine cheb2
```

Гравитационные параметры планет

```
real*8,  parameter :: mu_plan(11) = [ &
     22.03208047245d0, & ! Меркурий
     324.8587656142d0, & ! Венера
     398.6004415d0, &    ! Земля
     42.828287d0, &      ! Марс
     126712.59708d0, &   ! Юпитер
     37939.51971d0, &    ! Сатурн
     5780.158533417d0, & ! Уран
     6871.307771094d0, & ! Нептун
     1.02086d0, &        ! Плутон
     4.90279914d0, &     ! Луна
     132712439.935d0 &   ! Солнце
     ]
```

Прогрмма `planet_test` На вход получает дату и время. Печатает значения координат планет на заданное время. Перед использованием собрать make'ом.

1.2.2 Возмущение от сферических гармоник Земли.

Поскольку масса распределена внтури планет неравномерно, то потенциал поля отличается от того, что создает точечный источник. Потенциал гравитационного поля от телы произвольной формы

$$U = G\int_{\Sigma}\frac{\rho(\mathbf{s})d^{\mathbf{s}}}{|\mathbf{r}-\mathbf{s}|}$$

$\mathbf{s}$ – радиус-вектор малого элемента притягивающего тела, $\mathbf{r}$ – радиус-вектор объекта. Обратное расстояние раскладывается по полиномом Лежандра:

$$\frac{1}{|\mathbf{r}-\mathbf{s}|} = \frac{1}{r}\sum_{n=0}^{\infty}\left(\frac{s}{r}\right)^n P_n(\cos\gamma)$$

$P_n(u)$ – полином Лежандра степени n, γ – угол между $\mathbf{r}$ и $\mathbf{s}$. Сам полином раскладывается на сумму присоединенных полиномов Лежандра порядка от 0 до n (модифицированные производные соответствующего порядка). Полученное выражение подставляется в изначальный интеграл, суммирование выносится за интеграл, и все, что связано с интегрированием по $d\mathbf{s}$ загоняется в коэффициенты. В итоге потенциал выражается так

$$U = \frac{\mu}{r} \sum_{n=0}^{\infty} \sum_{m=0}^{n} \left(\frac{R_e}{r} \right)^n P_{nm}(\sin\varphi) \left(C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda) \right)$$

μ – гравитационная постоянная Земли, R_e – экваториальный радиус Земли, P_{nm} – присоединенный полином Лежандра степени n порядка m, φ – широта направления на объект, λ – долгота направления на объект, r – расстояние до объекта. $\mathbf{r} = (r\cos\varphi\cos\lambda, r\cos\varphi\sin\lambda, r\sin\varphi)^T$, C_{nm}, S_{nm} – коэффициенты, содержащие в себе информацию о распределении вещества внутри притягивающего тела.

К сожалению, это только выражение для потенциала. Сила, возникающая в результате неоднородности поля

$$\mathbf{f_{harm}} = -\nabla U$$

Дифференцирование значения потенциала по переменной x (в данном случае не важно, по какой именно из трех переменных):

$$\frac{\partial U}{\partial x} = \frac{\mu}{R_e} \sum_{n=0}^{\infty} \sum_{m=0}^{n} \left(\frac{R_e}{r} \right)^{n+1} P_{nm}(\sin\varphi) \Big[-\frac{n+1}{r} \frac{\partial r}{\partial x} (C_{nm} \cos m\lambda + S_{nm} \sin m\lambda) -$$

$$-m \frac{\sin\varphi}{\cos^2\varphi} \frac{\partial \sin\varphi}{\partial x} (C_{nm} \cos m\lambda + S_{nm} \sin m\lambda) +$$

$$+\frac{1}{\cos\varphi} \frac{\partial \sin\varphi}{\partial x} (C_{nm-1} \cos(m-1)\lambda + S_{nm-1} \sin(m-1)\lambda) +$$

$$+m \frac{\partial \lambda}{\partial x} (S_{nm} \cos m\lambda - C_{nm} \sin m\lambda) \Big]$$

Первое слагемое в сумме возникает из-за дифференцирования $\left(\frac{R_e}{r}\right)^{n+1}$, второе и третье – из-за дифференцирования $P_{nm}(\sin\varphi)$:

$$\frac{dP_{nm}(u)}{du} = -\frac{mu}{1-u^2} P_{nm}(u) + \frac{1}{\sqrt{1-u^2}} P_{nm+1}(u)$$

Так как группировка сумм происходит по P_{nm}, в нее попадают члены из дифференцирования P_{nm-1}. Четвертое слагаемое – результат дифференцирования $C_{nm} \cos m\lambda + S_{nm} \sin m\lambda$.

В программе расчета возмущений от сферических гармоник суммирование ведется от сначала по n а уже затем по m. Внешний цикл проходит значения $m = 1, 2, \ldots, N$, а внутренний цикл проходит значения $n = m, m+1, \ldots, N$, т.е. внутреннему циклу соответствуют столбцы таблицы. Элемент 00 опускается, поскольку соответствует центральному полю.

0,0			
1,0	1,1		
2,0	2,1	2,2	
3,0	3,1	3,2	3,3
...	...	...	...

Столбец с элементами $n0$ суммируется отдельно, т.к. все $S_{n0} = 0$. В цикле вычисляются четыре вспомогательные суммы:

$$W_1 = \sum_{n=1}^{N} \sum_{m=1}^{n} (n+1) \left(\frac{R_e}{r}\right)^n P_{nm}(\sin\varphi)(C_{nm}\cos m\lambda + S_{nm}\sin m\lambda),$$

$$W_2 = \sum_{n=1}^{N} \sum_{m=1}^{n} m \left(\frac{R_e}{r}\right)^n P_{nm}(\sin\varphi)(C_{nm}\cos m\lambda + S_{nm}\sin m\lambda),$$

$$W_3 = \sum_{n=1}^{N} \sum_{m=1}^{n} m \left(\frac{R_e}{r}\right)^n P_{nm}(\sin\varphi)(-C_{nm}\sin m\lambda + S_{nm}\cos m\lambda),$$

$$W_4 = \sum_{n=1}^{N} \sum_{m=1}^{n} \left(\frac{R_e}{r}\right)^n P_{nm}(\sin\varphi)(C_{nm-1}\cos(m-1)\lambda + S_{nm-1}\sin(m-1)\lambda),$$

В действительности, вычисленные внутри цикла значения W_i необходимо помножить на $\cos\varphi$, т.к. для удобства использование рекуррентных формул значение присоединенного полинома Лежандра P_{11} перед циклом было инициализировано как 1 вместо своего истинного значения $P_{11} = \cos\varphi$.

После работы основного цикла к сумме W_1 добавляются значения, обусловленные наличием коэффициентов C_{n0}:

$$U_0 = \frac{\mu}{R_e} \sum_{n=1}^{N} \left(\frac{R_e}{r}\right)^{n+1} P_n(\sin\varphi) C_{n0}$$

Результирующее ускорение после подстановки частных производных выражается следующим образом:

$$f_x = \frac{\mu}{r^2} \left(W_1 \frac{x}{r} - W_2 \frac{x}{r} \frac{z^2}{x^2+y^2} + W_3 \frac{yr}{x^2+y^2} + W_4 \frac{x}{r} \frac{z}{\sqrt{x^2+y^2}} \right)$$

$$f_y = \frac{\mu}{r^2} \left(W_1 \frac{y}{r} - W_2 \frac{y}{r} \frac{z^2}{x^2+y^2} - W_3 \frac{xr}{x^2+y^2} + W_4 \frac{y}{r} \frac{z}{\sqrt{x^2+y^2}} \right)$$

$$f_z = \frac{\mu}{r^2} \left(W_1 \frac{z}{r} + W_2 \frac{z}{r} - W_4 \frac{\sqrt{x^2+y^2}}{r} \right)$$

1.2.3 Возмущение от сопротивления атмосферы

Ускорение вызванное сопротивлением атмосферы определяется формулой

$$f_{atm} = -C\frac{S}{m}\rho v_{rel}^2 \mathbf{e_v},$$

$$f_{atm} = -\sigma\rho v_{rel}^2 \mathbf{e_v}.$$

S – площадь сечения объекта, m – масса объекта, C – коэффициент сопротивления, три параметра сводятся к одному ma – баллистическому коэффициенту. Обычном C находится в интервале 1.5 - 3.0, и зная массу и форму аппарата баллистический коэффициент можно восстановить, однако чаще он вносится в число неизвестных параметров и уточняется по измерениям. Следует отметить, то сила зависит от скорости аппарата относительно вращающейся Земли. Существует множество моделей, описывающих плотность верхних слоев земной атмосферы. Все более или менее адекватные модели плотности привязаны к положению Солнца относительно Земли, т.е. суточному времени (нагретая Солнцем атмосфера "раздувается"), а также к индексам солнечной и геомагнитной активности. В комплексе используется модель плотности верхней атмосферы, соответствующая ГОСТу 2004 года. Входными параметрами для нее являются:

- индекс $F_{10.7}$ – плотность потока солнечного излучения на длине волны 10.7 см, выраженная в 10^4 Jy (Янских) $= 10^{-22}$ Вт/м^2/с.
- индекс F_{81} – средневзвешанное значение $F_{10.7}$ на интервале трех оборотов Солнца или 81 суток. Если $i = \overline{-80, 0}$ – индекс суток, где 0 соответствует текущим суткам, а -80 последним суткам в интервале, вес $W_i = 1 + \frac{0.5 \cdot i}{80}$ индекс высчитывается по формуле

$$F_{81} = \frac{\sum_{i=-80}^{0} W_i \cdot F_{10.7}}{\sum_{i=-80}^{0} W_i}.$$

- индекс F_0 – постоянный уровень излучения, получаемый из F_{81} округлением до ближайшего значения кратного 25.

- Индекс K_p геомагнитной активности, который варьируется от 0 до 9 и обычно отмечается дискретно с шагом 1/3, т.е. 3.0, 3.3, 3.7 и т.д., иногда попадаются обозначения 3_0, $3_+, 4_-$..

1.3 Шкалы времени

Если ограничиться только рамками задачи, которой мы занимаемся, то можно не заморачиваться различными шкалами времени. Во время интегрирования используется одно и то же время – TDB, оно же эфемеридное время. Однкао все земные события привязываются как правило к UTC. Также вектора состояния аппарата и начальные условия принято давать в

UTC. В СССР и позже в России сложилось так, что баллистики и военные вместо UTC часто используют МДВ (Московское Декретное Время) совпадающее с UTC+3:00 - московским зимним временем.

К примеру, в задаче `planet_test` на вход берется дата и время t в шкале МДВ, переводится в эфемеридное время $t_{ET} = t + 32.184 + 35 - 3 \cdot 3600$, по эфемеридному времени вычисляются координаты планет.

1.3.1 TDB (Barycentric Dynamical Time)

Временная шкала пространства-времени в системе отсчета, находящейся в барицетре солнечной системы. Наиболее близкий современный аналог эфемеридного времени (Ephemeris Time - ET). Время данной шкалы является аргументом для получения эфемерид DE4** лаборатории JPL. Время TDB в среднем отличается от шкалы атомного времени TAI ровно на 32.183 секунды. Термин "в среднем" в предыдущем приложении обусловлен тем, что шкала TAI зависит от места, в которою находятся атомные часы. TDB отличается от TAI + 32.183 лишь периодическими вариациями.

Именно эта шкала используется в качестве шкалы внутреннего времени комплекса, а также в качестве времени в дифференциальных уравнениях движения. В этой же шкале времени задаются эфемериды планет JPL, используемые при интегрировании уравнений движения.

1.3.2 UTC (Coordinated Universal Time)

Основная шкала времени земных обитателей. К этой же шкале привязываются основные события на Земле, в то числе траекторные измерения, а также пуски, включение двигателей, ориентация и пр.

Время UTC подстраивается таким образом, чтобы соответствовать вращению Земли. Базой для него является все то же атомное время TAI, от которого оно отстает на целое (с 1968) число секунд. На момент конца 2012 года TAI - UTC = 35.0 сек. Таким образом, UTC отстает от эфемеридного времени 32.183+35.0 секунд.

1.3.3 TAI (International Atomic Time)

Международное атомное время – это измеренное время. Фактически, отличие времени, измеренного атомными часами в конкретной точки, от координатного времени системы зависит от движения этой точки: $d\tau = ds/c$. Однако TAI является взвешенной совокупностью времен, измеренных 200 атомными часами по всему миру. Подобное усреднение используется для устранения эффектов, связанных с относительным движением часов. TAI базируется не на вращении Земли, а на более фундаментальных законах. Основной атомного времени обычно является цезиевый стандарт частоты.

1.3.4 Юлианская дата (Julian Date, JD)

Юлианская дата в общем случае величина не целая. Она может считаться аналогом абсолютного времени (чем когда-то и являлась). Ее полезно использовать на крупномасштабных задачах, где события отделены друг от друга годами или даже столетиями. К примеру, юлианская дата, соответствующая 12:00 TDB 1 января 2000 года, равна 2451545.0, эта эпоха (момент времени) также называется J2000. Современные системы отсчета определены через различные геометрические характеристики солнечной системы, соответствующие именно этому моменту времени.

Целая юлианская дата соответствует полудню всемирного времени (по эфемеридной шкале). Одни сутки юлианская даты содержат 86400 секунд. Соответствие между юлианской и обычной датой $(year, month, day)$ устанавливается следующим образом (**примечание: здесь и далее под делением подразумевается целочисленное деление**)

$$JD = (146097 \cdot C_{100})/4 + (1461 \cdot Y_{100})/4 + (153 \cdot M + 2)/5 + D + 1721119$$

M - месяц, но в нестандартной нумеровке, нулевой месяц соответствует марту, февралю соответствует 11 месяц **предыдущего** года;
$C_{100} = year/100$ – юлианское столетие;
$Y_{100} = year - 100 \cdot C_{100}$ – год в столетии;
D – день в месяце.

Пример Fortran-функции, вычисляющей юлианскую дату. На вход подается дата в виде YYYYMMDD.000000.

```
double precision function dt_ajd (dt)
  implicit none
  real*8 dt
  integer year, month, day, century, year_c
  year = floor(dt/10000)
  month = floor((dt-year*10000.d0)/100.d0)
  day = nint(dt)-year*10000-month*100
  if (month.gt.2) then
     month = month - 3
  else
     month = month + 9
     year = year - 1
  endif
  century = year/100
  year_c = year - 100*century
  dt_ajd = (146097*century)/4+(1461*year_c)/4+ &
       (153*month+2)/5+day+1721119
  return
end function dt_ajd
```

Для представления времени чаще удобнее использовать юлианскую дату начала суток $JD_0 = JD - 0.5$, такая дата соответствует полуночи.

1.4 Системы координат

Наиболее распространенными системами координат в прикладных задачах небесной механики являются ICRF (International Celestial Reference Frame) и ITRF (International Terrestrial Reference Frame). Первая система координат фиксирована относительно пространства, в частности по большей части фиксирована относительно удаленных космических источников, вторая с небольшими оговорками фиксирована относительно Земли. Оси ICRF принято направлять согласно определенной геометрии солнечной системы в эпоху J2000 ($JD = 2451545.0$, 12 часов, 1 января 2000 года по эфемеридному времени). Ось Z ICRF направлена перпендикулярно среднему экватору Земли эпохи J2000 в сторону северного небесного полюса, в среднюю точку весеннего равноденствия эпохи J2000 (точку пересечения среднего экватора Земли со эклиптикой), ось Y дополняет систему до правой тройки. Оси ITRF связаны с Землей, X и Y лежат в плоскости экватора, X и Z лежат в плоскости нулевого меридиана. Фиксированная относительно пространства ICRF имеет начало координат в барицентре солнечной системы и может считаться инерциальной. При рассмотрении околоземных аппаратов часто прибегают к системе, с теми же осями, что и ICRF, на началом координат в центре масс Земли. Такую систему часто называют EME2000 (mean equator and equinox) или просто по названию эпохи – J2000.

Матрица перехода от EME2000 к ITRF записывается в следующем виде:

$$\mathbf{T} = \mathbf{P} \cdot \mathbf{B} \cdot \mathbf{A_{nut}} \cdot \mathbf{A_{prc}},$$

где составляющие матрицы это:

- $\mathbf{A_{prc}}(t)$ – матрица прецессии, которая обеспечивает переход от среднего экватора и равноденствия эпохи J2000 к среднему экватору и равноденствию эпохи t.
- $\mathbf{A_{nut}}(t)$ – матрица нутации, обеспечивающая переход от среднего экватора и равноденствия эпохи к истинному экватору и равноденствию эпохи.
- $\mathbf{B}(t)$ – матрица суточного вращения земли, обеспечивающая переход от истинного экватора и равноденствия эпохи к истинному экватору и нулевому меридиану эпохи.
- $\mathbf{P}(t)$ – матрица движения полюсов, которая корректирует связанную с землей систему на движение полюсов (близка к единичной).

1.4.1 Матрица прецессии

Матрицу можно получить при помощи трех последовательных поворотов

$$\mathbf{A_{prc}} = \mathbf{R_z}(-z_A) \cdot \mathbf{R_y}(\theta_A) \cdot \mathbf{R_z}(-\zeta_A).$$

Матрица $\mathbf{R_u}(\alpha)$ – матрица поворота вокруг оси u на угол α. Величины z_A, θ_A и ζ_A – функции юлианских столетий с момента эпохи J2000.

$$\zeta_A = 2306''.2181T + 0''.30188T^2 + 0''.017998T^3,$$

$$z_A = 2306''.2181T + 1''.09468T^2 + 0''.018203T^3,$$

$$\theta_A = 2004''.3109T + 0''.42665T^2 + 0''.041833T^3.$$

$$T = \frac{JD - J2000}{100}$$

Расчет матрицы прецессии реализован в PM2000.F [1]. Однако там рассмотрен более общий случай поворота от одной эпохи к другой.

1.4.2 Матрица нутации

Матрица эквивалентна трем поворотам:

$$\mathbf{A_{nut}} = \mathbf{R_x}(-\bar{\epsilon} - \Delta\epsilon) \cdot \mathbf{R_z}(-\Delta\psi) \cdot \mathbf{R_x}(\bar{\epsilon}),$$

где $\bar{\epsilon}$ – среднее наклонение эклиптики к экватору в данную эпоху, $\Delta\epsilon$ – нутация в наклонении, $\Delta\psi$ – нутация в долготе. Средний наклон эклиптики к экватору является функцией юлианских столетий с момента J2000

$$\bar{\epsilon} = 84381''.448 - 46''.8150T - 0''.00059T^2 + 0''.001813T^3$$

Расчет среднего наклона эклиптики реализован в процедуре E2000.F. Поправки к наклону эклиптики и нутации в долготе либо интерполируются из файла параметров ориентации Земли EOP (Earth Orientation Parameters), либо высчитываются непосредственно из теории нутации. В программе N2000.F реализована теория нутации IAU 1980 (Seidelmann, 1982), поправки разложены на 106 гармоник. Итоговый расчет матрицы нутации реализован в NM2000.F.

[1] здесь и далее файлы *.F соответствуют legacy code, написанному на FORTRAN77, одной из особенностей таких программ является представление возвращаемых матриц в виде линейного массива, элементы матрицы записываются в массив последовательно по строкам.

1.4.3 Вращение Земли

Вращение Земли определяется так называемым здвездным временем (sidereal time) θ. Соответствующий поворот - поворот на угол θ вокруг оси z. Истинное звездное время:

$$\theta = \theta_M + \Delta\theta,$$

где θ_M – среднее звездное время, а $\Delta\theta$ – поправка, определяющаяся из уравненя равноденствий. Среднее звездное время является функцией времени UT1.

$$\theta_M = \theta_M(0^h UT1) + UT1_d,$$

$UT1_d$ – время UT1 прошедшее с полуночи UT1, или попросту UT mod 86400s, $\theta_M(0^h UT1)$ – поправка на начало суток UT1, вычисляющаяся как функция от юлианских столетий с эпохи J2000

$$\theta_M(0^h UT1) = 24110^s.54841 + 8640184^s.812866 T_U + 0^s.093104 T_U^2 - 6^s.2 \times 10^{-6} T_U^3.$$

T_U – число юлианских столетий с эпохи J2000 **в шкале UT1**. Поправка из уравнений равноденствий с достаточной точностью получается по формуле

$$\Delta\theta = \Delta\psi \cos\bar{\epsilon} - \Delta\psi(\sin\bar{\epsilon})\Delta\epsilon,$$

где $\Delta\psi$ – нутация в долготе, $\bar{\epsilon}$ – средний наклон эклиптики, $\Delta\epsilon$ – нутация в наклоне эклиптики. Поскольку сами поправки достаточно маленькие, вторым членом в предыдущем выражении можно пренебречь. Итоговая матрица вращения Земли

$$\mathbf{B} = \mathbf{R_z}(\theta).$$

1.4.4 Матрица движения полюсов

Пусть x_p и y_p – малые угловые смещения небесного полюса. Матрица, нейтрализующая эти смещения имеет вид:

$$\mathbf{P} = \mathbf{R_y}(-x_p) \cdot \mathbf{R_x}(-y_p),$$

Что в силу малости аргументов примерно равно

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & +x_p \\ 0 & 1 & -y_p \\ -x_p & +y_p & 1 \end{pmatrix},$$

В процедуре, `PM_mat` модуля `iers` матрица вычисляется более точно, тем не менее, приведенной формы тоже достаточно. Таким образом, для расчета матрицы перехода от инерциальной системы координат к связанной с Землей необходимо три внешних параметра:

- угол x_p смещения истинного полюса
- угол y_p смещения истинного полюса
- значение $UT1 - UTC$ для расчета времени UT1

2 Уточнение орбитальных параметров при помощи измерений

Предположим, что движение объекта описывается в рамках заданной модели описывается постоянным вектором $\mathbf{q}$ и временем t. Простейшим примером такого вектора может служить вектор состояния объекта или вектор из элементов орбиты, соответствующий определенному моменту времени. В общем случае вектор $\mathbf{q}$ может быть расширен за счет других параметров, например коэффициента светового давления и/или баллистического коэффициента. Предположим также, что мы имеем набор измерений $\mathbf{\Psi_i}$ по заданному объекту и хотим подобрать такой набор параметров $\mathbf{q}$, чтобы измерения согласовывались с орбитой в терминах наименьших квадратов рассогласований. Другими словами, достичь минимума функционала

$$\mathbf{\Phi} = \sum_i \left(\mathbf{\Psi_i^{obs}} - \mathbf{\Psi_i^{calc}}\right)^T \mathbf{P_i} \left(\mathbf{\Psi_i^{obs}} - \mathbf{\Psi_i^{calc}}\right)$$

где $\mathbf{\Psi_i^{obs}}$ — и есть имеющиеся измерения, $\mathbf{\Psi_i^{calc}}$ — расчетные значения этих измерений, $\mathbf{P_i} = \mathbf{K_i^{-1}}$ — весовая матрица, обратная к ковариационной.

$$\mathbf{\Psi_i^{calc}} = \mathbf{\Psi_i}(\mathbf{q}, t_i),$$

$$\mathbf{\Phi} = \sum_i \xi_\mathbf{i}^\mathbf{T} \mathbf{P_i} \xi_\mathbf{i}.$$

Наиболее простым и наиболее часто используемым методом нахождения минимума функционала является метод обобщенных касательных Ньютона, в котором желаемый вектор $\mathbf{q}$ ищется путем последовательных приближений. Предположим, на определенном шаге мы имеем вектор $\mathbf{q^{(i-1)}}$. Рассмотрим случай одного измерения (любой случай можно свести к одному измерению большой размерности), выразим вектор уточняемых параметров

$$\mathbf{q^{(i)}} = \mathbf{q^{(i-1)}} + \mathbf{\Delta q}.$$

Выберем поправки такими образом, чтобы выполнялось необходимое условие минимума функционала.

$$\left(\frac{\partial \xi}{\partial q_j}\right)^T \mathbf{P}\xi + \xi^T \mathbf{P} \frac{\partial \xi}{\partial q_j} = 0, \quad j = 1, 2, \dots m$$

В предыдущем выражении в силу симметричности матрицы $\mathbf{P}$ слагаемые равны, и достаточно рассмотреть только одно из них. Рассогласования ξ приближенно выражаются через изменение вектора уточняемых параметров

$$\xi(q^{(i-1)}) = \xi(q^{(i)}) + \frac{\partial \xi}{\partial \mathbf{q}} \mathbf{\Delta q} + \dots$$

После подстановки этого выражения в условие минимума и пренебрежения членами степени два и выше по $\mathbf{\Delta q}$, а также пренебрегая изменением в частных производных на шаге, получаем уравнение для поправок к уточняемым параметрам

$$\mathbf{A\Delta q} = \mathbf{B},$$

$$\mathbf{A} = \left(\frac{\partial\xi}{\partial\mathbf{q}}\right)^{\mathbf{T}} \mathbf{P} \left(\frac{\partial\xi}{\partial\mathbf{q}}\right), \quad \mathbf{B} = \left(\frac{\partial\xi}{\partial\mathbf{q}}\right)^{\mathbf{T}} \mathbf{P}\xi.$$

Система уравнений для поправок называется системой нормальных уравнений, легко показать, что матрица системы $\mathbf{A}$ и вектор правых частей $\mathbf{B}$ аддитивны по измерениям.

$$\mathbf{A} = \sum_{\mathbf{k}} \mathbf{A_k}, \quad \mathbf{B} = \sum_{\mathbf{k}} \mathbf{B_k}.$$

Все расчетные значения и частные производные, необходимые для получения системы нормальных уравнений, рассчитываются из параметров $(i-1)$ шага. Изучение сходимости такого процесса — сложная и глубоко нелинейная задача. В некоторых случаях, когда сходимость затрудняется плохим начальным приближением, неточным расчетом частных производных или формой функционала, можно использовать более мелкий шаг, давая приращение параметрам $\alpha\mathbf{\Delta q}$, где $\alpha < 1$. Чаще же всего никаких манипуляций с шагом не требуется, и задача уточнения орбиты сходится нормально.

Основой для формирования системы нормальных уравнений служат частные производных от измерений по уточняемым параметрам $\frac{\partial\xi}{\partial\mathbf{q}}$. Для абсолютного большинства измерений, однако, гораздо проще выразить зависимость от текущего вектора состояния, чем от набора уточняемых параметров

$$\frac{\partial\xi}{\partial\mathbf{q}} = \frac{\partial\xi}{\partial\mathbf{X}}\frac{\partial\mathbf{X}}{\partial\mathbf{q}}.$$

При этом $\frac{\partial\xi}{\partial\mathbf{X}}$ вычисляется довольно просто и хорошо известны для традиционных типов измерений. Второй множитель, в свою очередь, не зависит от измерений и может быть рассчитан либо интегрированием дополнительных $6m$ (m — размерность $\mathbf{q}$), либо посчитаны разностно при варьировании элементов $\mathbf{q}$. Например для оптических измерений $\xi = (RA, Dec)$

$$\frac{\partial RA}{\partial\mathbf{r}} = \begin{pmatrix} -\frac{X_2}{X_1^2+X_2^2} \\ \frac{X_1}{X_1^2+X_2^2} \\ 0 \end{pmatrix}, \quad \frac{\partial RA}{\partial\mathbf{v}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

$$\frac{\partial Dec}{\partial\mathbf{r}} = \begin{pmatrix} -\frac{X_3X_1}{r^2\sqrt{X_1^2+X_2^2}} \\ -\frac{X_3X_2}{r^2\sqrt{X_1^2+X_2^2}} \\ \frac{\sqrt{X_1^2+X_2^2}}{r^2} \end{pmatrix}, \quad \frac{\partial Dec}{\partial\mathbf{v}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$