



SPICE toolkit

N. Hirata (Univ. of Aizu)

Ancillary Data



- Science (Instrument) Data

- ある観測の結果として、直接取得されたデータ

- Ancillary Data

- ancillary: 付随, 補助
- Science Dataに付随し, これを補助するためのデータ
- データに意味付けを行う

Ancillary Data (cont.)

- 月惑星探査において，補助データは探査機の運用記録として保存される
- 補助データの管理，利用のため必要な
 - データ保存フォーマット
 - データの読み出し方法
 - データ利用のためのルーチン
- を定義する枠組みが必要

SPICE

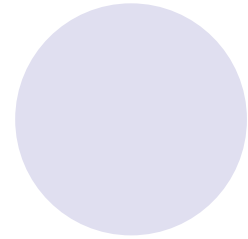
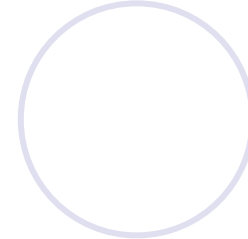
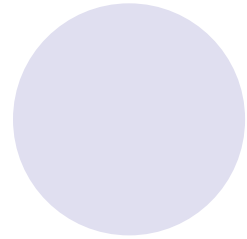
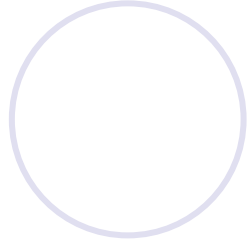


- NASAジェット推進研究所(JPL)が開発した，補助データに関する枠組み
- 補助データを保存するファイル
 - カーネル(kernel)
- 補助データを扱うツール
 - SPICE toolkit (or NAIF toolkit)
- ミッションに依存しない，共通の枠組み
 - 日本でも「はやぶさ」以降使われている
 - 「かぐや」， Planet-C...

運用計画と補助データ

- 運用計画
 - 未来の運用「記録」
- 探査機の未来の状態を知って，データ取得の計画を立てる
- 将来のことを扱うが，考え方は運用記録と同じ
 - 補助データの枠組みで取り扱う

語源



S

Spacecraft

P

Planet

I

Instrument

C

C-matrix

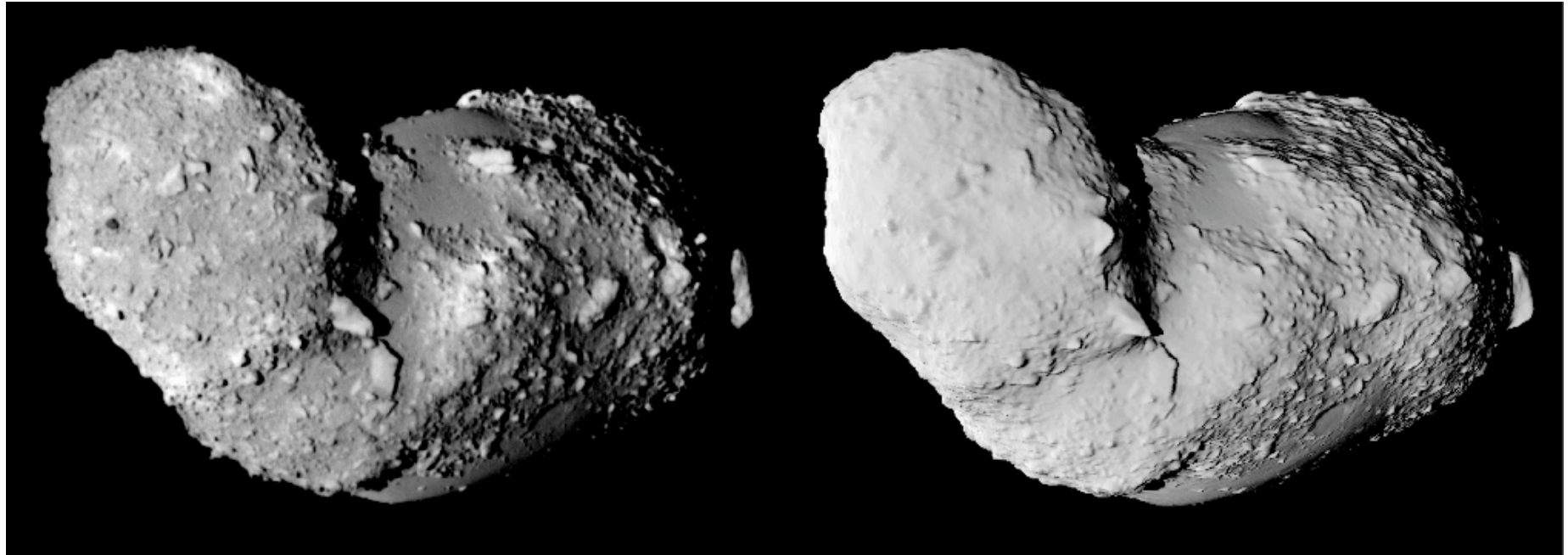
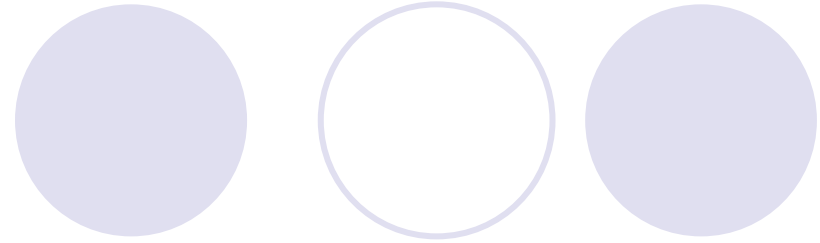
E

Events

観測データと補助データ

- 観測データ (science data)
 - 実際に研究の源泉になるデータ
- 補助データ (ancillary data)
 - 観測データに付随するデータ
 - 観測データに「意味付け」を行う
 - 観測データそのものと同じぐらい重要 (素性のわからないデータは意味がない)
 - 5W1Hを与える

SPICEの活用例



はやぶさが
撮影した
実際の画像

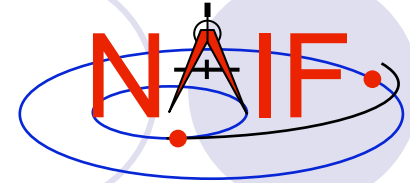


時刻
探査機位置・姿勢
天体, 光源位置, 形状
機器情報



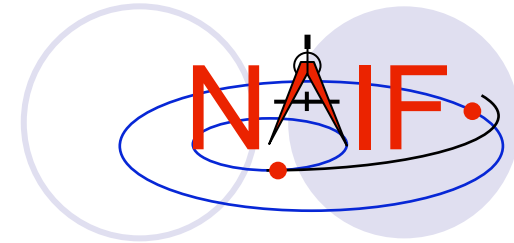
SPICEを使って
作成した
模擬画像

カーネル



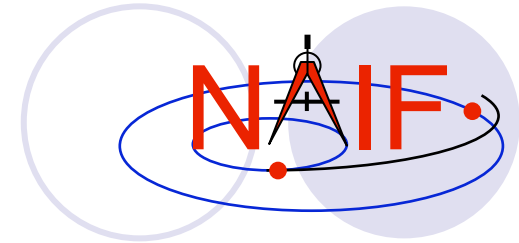
- 補助データの実態であるデータファイル
 - OSのkernelと語源は同じ（果物の芯，核心）
- 情報の種類ごとに個別のファイルで保存
 - ミッション共通のカーネルとミッション固有のカーネルがある
 - 共通
 - 天体の位置（軌道）情報，うるう秒情報など
 - 固有
 - 探査機の位置情報，カメラの視野情報など

SPICE Kernels Family



- **SPK**
 - Spacecraft and Planet Ephemeris
- **PcK**
 - Planetary Constants, for natural bodies
 - Orientation
 - Size and shape
- **IK**
 - Instrument
- **CK**
 - Pointing (“C-matrix”)
- **EK**
 - Events
 - *The Events kernel idea has not taken hold. After Cassini it may disappear*
- **FK**
 - Reference frame specifications
- **SCLK**
 - Spacecraft clock correlation data
- **LSK**
 - Leapseconds
- **Meta-Kernel (a.k.a. “FURNISH kernel”)**
 - Mechanism for aggregating and easily loading a collection of kernel files
- **Plate kernel**
 - Shape model

Text and Binary Kernels



SPICE **text** kernels are:

- text PCK (the most common type of PCK)
- IK
- FK
- LSK
- SCLK

- MK (“Furnsh” meta-kernel)

SPICE **binary** kernels are:

Architecture Independent

- SPK
- binary PCK (exists only for Earth and moon)
- CK

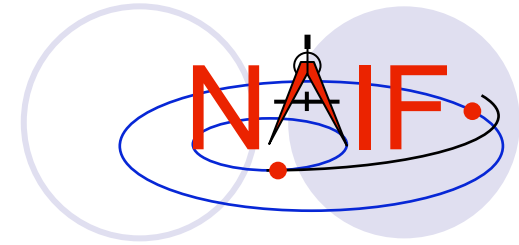
Architecture Dependent

- ESQ (part of the E-kernel)
- DBK (database kernel)
- Plate kernel

SPICE Toolkitでできること

- 一言で言えば以下の二点
 - kernelからのデータ読み出し
 - データの変換
 - 座標系の変換
 - 時間系の変換
- データも，データ変換のための情報も kernelに含まれている
 - 必要な全てのカーネルを用意し，読み込むことが重要

SPICE Toolkit



- 入手元

- <http://naif.jpl.nasa.gov/naif/toolkit.html>
- <ftp://naif.jpl.nasa.gov/>

- 動作環境

- FORTRANで開発後, C, IDL, MATLABへ移植
- UNIX系OS (linux, solaris, MacOSX, Cygwin) と Windows上で動作

- 今回はC版 (CSPICE) を使用

- 当初の開発言語はFORTRANのため, tutorialなどではFORTRAN版での関数用例が書いてある
- 引数などの使い方はC/FORTRANで同じ
- 関数名: SPKEZR → `spkezr_c`

ドキュメント



- チュートリアルがtoolkit本体とともに配布
- doc/ディレクトリにusers guideと詳説文書 (required reading)
 - HTML版もavailable
- “Hands-on”レッスンで自習も可能

SPICE IDとSPICE Name

- SPICE toolkitにおいて識別すべき要素
 - 天体
 - 惑星と衛星
 - 探査機
 - 搭載機器
 - 座標系
- SPICEではこれらにIDと名称を付与し、識別できるようにしている
 - 数字のIDコードは主に古い関数で用いる
 - 名称は比較的新しい関数で用いる
 - 関数によってはどちらでも認識する

SPICE IDとSPICE Nameの例

- 人工天体のID (S/C ID) は負の整数
 - -130 'HAYABUSA', 'MUSES-C'
 - -131 'SELENE'
 - -500 'RSAT', 'SELENE Relay Satellite'
 - -502 'VSAT', 'SELENE VRAD Satellite'
- 人工天体の搭載機器, 内部座標系はS/C IDx1000-機器番号
 - -130000 'HAYABUSA_SC_BUS_PRIME'
 - -130102 'HAYABUSA_AMICA'
 - -130200 'HAYABUSA_NIRS'

SPICE IDとSPICE Nameの例

- 自然天体と系の重心は正の整数
 - 0 'SOLAR SYSTEM BARYCENTER'
 - 10 'SUN'
 - 1 'MERCURY BARYCENTER'
 - 199 'MERCURY'
 - 3 'EARTH MOON BARYCENTER'
 - 399 'EARTH'
 - 301 'MOON'
 - 2000001 'CERES'
 - 2025143 'ITOKAWA'

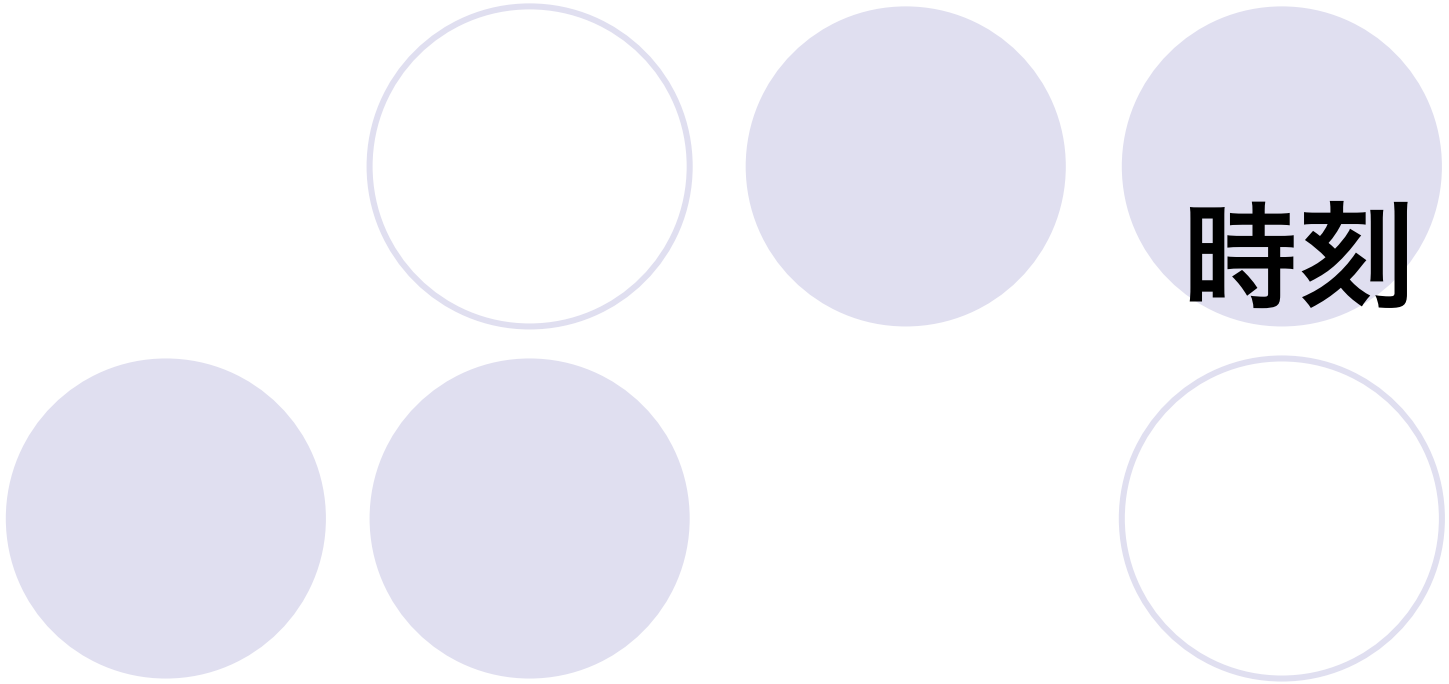
SPICE IDとSPICE Nameの例

- 慣性座標系や自然天体に関する座標系は基本的にはSPICE Nameを用いる
 - 'J2000'
 - 'ECLIPJ2000'
 - 'EARTH_FIXED'
 - 'IAU_MOON'
 - 'ITOKAWA_FIXED'
- 多くはtoolkit内で定義されているが、FK, IKなどでローカル定義を行うこともできる
 - -130900 'HAYABUSA_HP_FRAME'
 - -130910 'ITOKAWA_-Z_SUN_+X'
 - これらはHAYABUSA_HP.TFで定義

実習の内容



- 時刻
 - 探査機の時計と地上の時計との相互変換
- 位置
 - 探査機と他の天体との位置関係の算出
- 方位
 - 探査機とその観測機器と他の天体との方位関係の算出
- はやぶさを例に取り， SPICE toolkitを用いて様々な計算を試みる

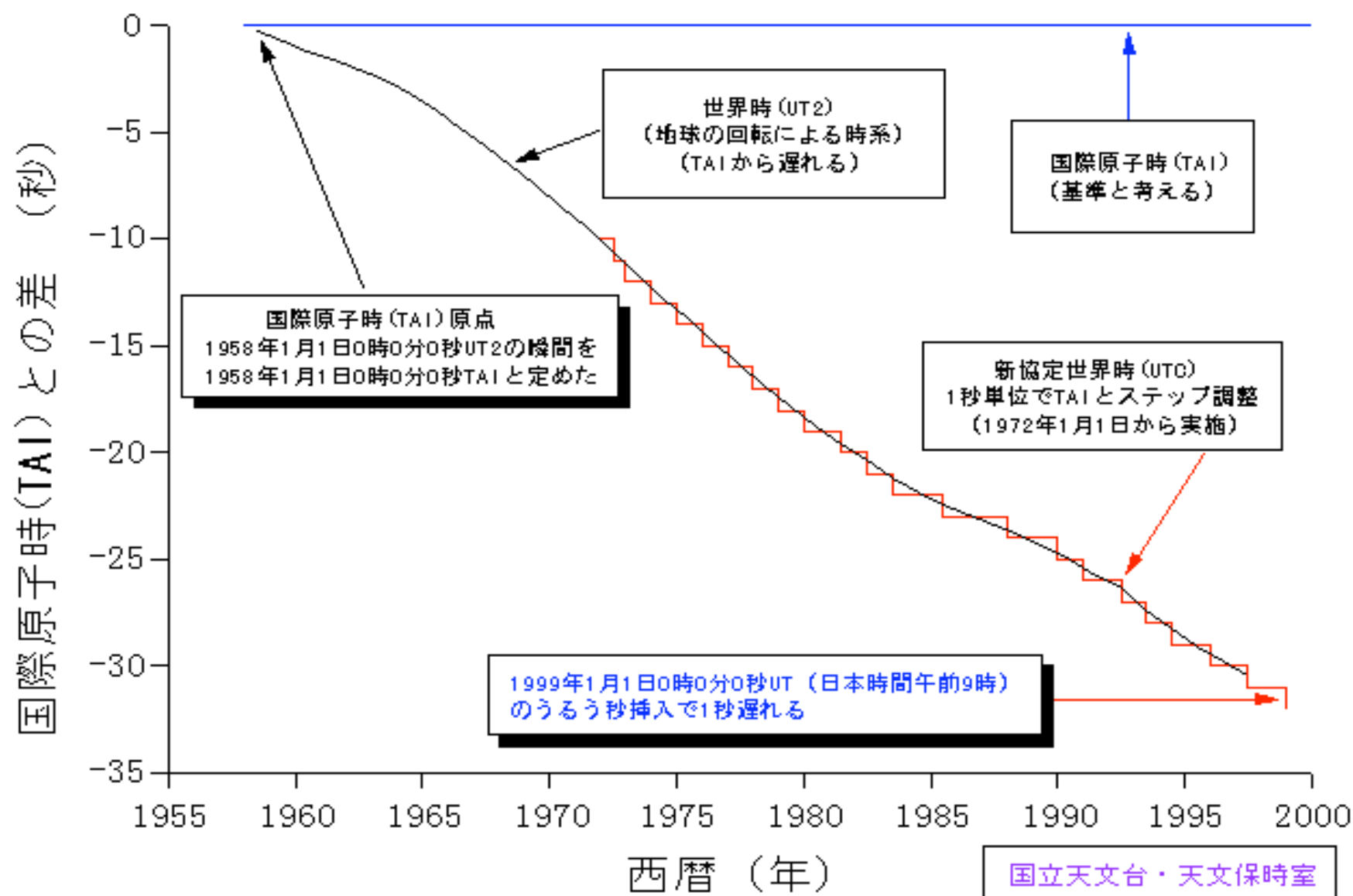


時刻

いろいろな時系 (time system)

- 国際原子時 (TAI:International Atomic Time)
 - 一番正確な, 一定刻みの時計
 - セシウム原子の振動数が基準
- UT (世界時, Universal Time)
 - 地球の自転 (昼/夜) を基準とする
 - 自転速度の変化に合わせて, 刻みの伸び縮みがある
- UTC(協定世界時, Coordinated Universal Time)
 - 「普通の」時計
 - 世界時(UT1)とTAIの兼ね合わせで決められる
 - TAIと同じく一定刻みだが, うるう秒 (leap second) の挿入によってUT2と合わせている

国際原子時(TAI)と協定世界時(UTC)・うるう秒の関係



いろいろな時系 (time system)

- ET (暦表時, Ephemeris Time)
 - 軌道計算などに用いられていた
 - 地球の公転を基準とする
- DT (力学時, Dynamical Time)
 - ETに替わって用いられる (1984年以降)
 - 刻みは原子時による
 - 相対性理論に基づくので、基準位置によって異なる2種類の時系がある
 - 地球重心基準 (TDT: Terrestrial Dynamical Time)
 - 太陽系重心基準 (TDB: Barycentric Dynamical Time)
 - TDT/TDBとETは連続していると見なされる (SPICE上ではET=TDBとして扱う)

探査機の時計 (TI)

- ただのストップウォッチ
- 刻みは様々 (tickと呼ぶ)
 - はやぶさでは1/8192sec
 - ただし, 表示などは1/32sec
 - ST_2420777432_v.fitsの2420777432は1/32sec刻みのTI (機上処理の時刻を表す)
- 進んだり, 遅れたりする
- “0”に戻ったりすることもある
 - 地球上での時刻との時刻合わせが必要
- 時刻合わせ情報: sclk
- うるう秒情報: lsk

サンプル 1

- ti2utcはTIをUTCに変換
- utc2tiはUTCをTIに変換
- 最も基本的なユーティリティの例
- コンパイル
 - \$ gcc ti2utc.c /usr/local/lib/cspice.a -o ti2utc
 - \$ gcc utc2ti.c /usr/local/lib/cspice.a -o utc2ti
 - 環境によってはstrcpyで警告メッセージが出るかも
 - cspice.aがSPICE toolkitのライブラリ

サンプル1 (続き)

- 実行例

 - \$./ti2utc

- エラーメッセージが出るはず

 - 必要なカーネルが見つからない

 - hayabusa.tsc HAYABUSAのSCLK

 - naif0009.tls うるう秒情報

 - ソースコードにファイル名が直書きになっている

 - ファイルをutc2tiと同じ場所にコピーするか、ソースコードにカーネルへのフルパスを記入して再度コンパイル

サンプル 1 (続き)

- 実行例

```
$ ./ti2utc 2420777432  
2005-09-30T12:27:38
```

```
$ ./utc2ti 2005-09-30T12:27:38  
2420777432
```

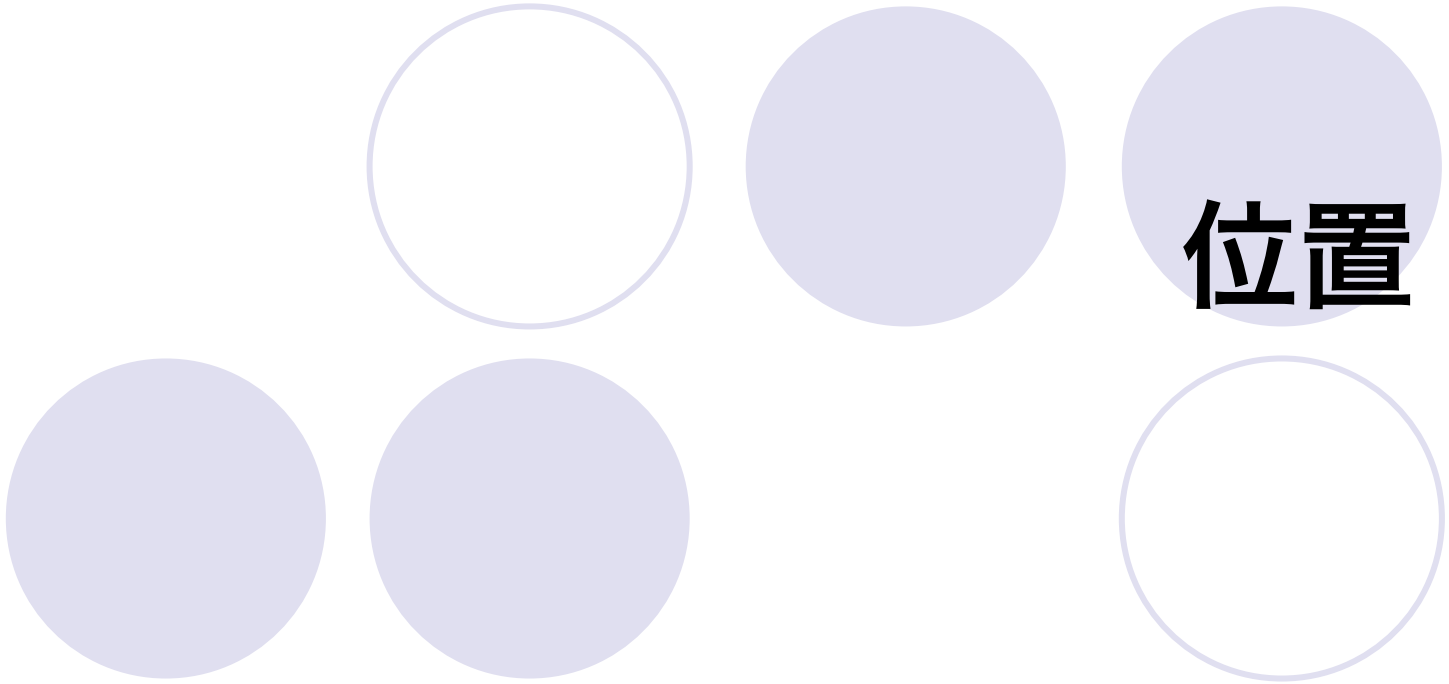
- TIとUTCの相互変換ができる

サンプル1 (続き)

- ti2utcとutc2tiで使ったtoolkitの関数
 - furnsh_c: カーネル読み込み
 - scs2e_c: SCLK string to ET
 - TI (文字列, 表示用) をetに変換
 - etはSpiceDouble型, 単位は秒
 - tick単位 (SpiceDouble) が入力ならsct2e_cを使う
 - et2utc_c: etをUTC文字列に変換
 - UTCの出力フォーマットを選べる
 - utc2et_c: UTC文字列をetに変換
 - sce2t_c: etをTI (tick単位) に変換
 - 最終的には256で割って1/32sec刻みの値を得る

カーネル読み込みの実際的な対処

- **カーネル名の直書きはよくない**
 - 通常カーネルは特定の場所にひとまとめにしておいた方がよい
 - 実行ファイルの場所も不定
 - カレントディレクトリも不定
 - 必要なカーネルは多岐にわたり，数も不定
- **実際的な対処**
 - 引数で指定
 - 環境変数を利用
 - メタカーネル（カーネルのリスト）を利用



位置

ケプラーの軌道6要素

- 軌道の形状

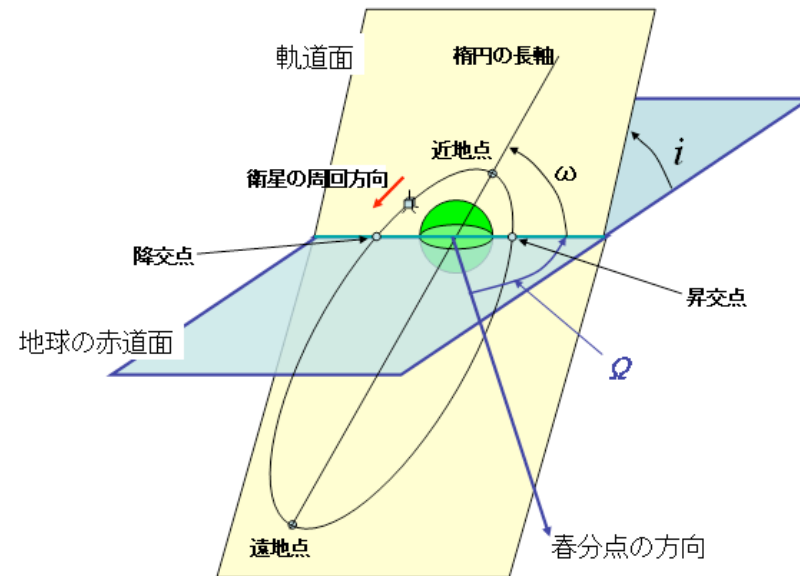
- 軌道長半径: a
- 離心率: e

- 軌道の配置

- 軌道傾斜角: i
- 昇交点赤経: Ω
- 近点引数: ω

- 軌道上の位置

- 平均近点角: M
- 元期: epoch



カルテシアンのの6要素

- ある時点における
 - 探査機の位置 (X,Y,Z)
 - 探査機の速度 (Vx,Vy,Vz)
- ケプラーの要素と同じく, 6つの変数で探査機の軌道を表現できる
- SPICEでは, SpiceDoubleの配列で表現
 - 3要素で位置
SpiceDouble position[3];
 - 6要素で状態 (位置+速度)
SpiceDouble state[6];

座標系



- 中心

- 地球, 太陽, 他の天体, 探査機
- Geocentric, Heliocentric, Planetocentric

- 座標軸の向き

- 慣性系に固定, 自転する天体に固定 : 緯経度
- Non-rotating, Body-fixed

- 緯経度の地理座標系, 地心座標系

- Geocentric, Geographic

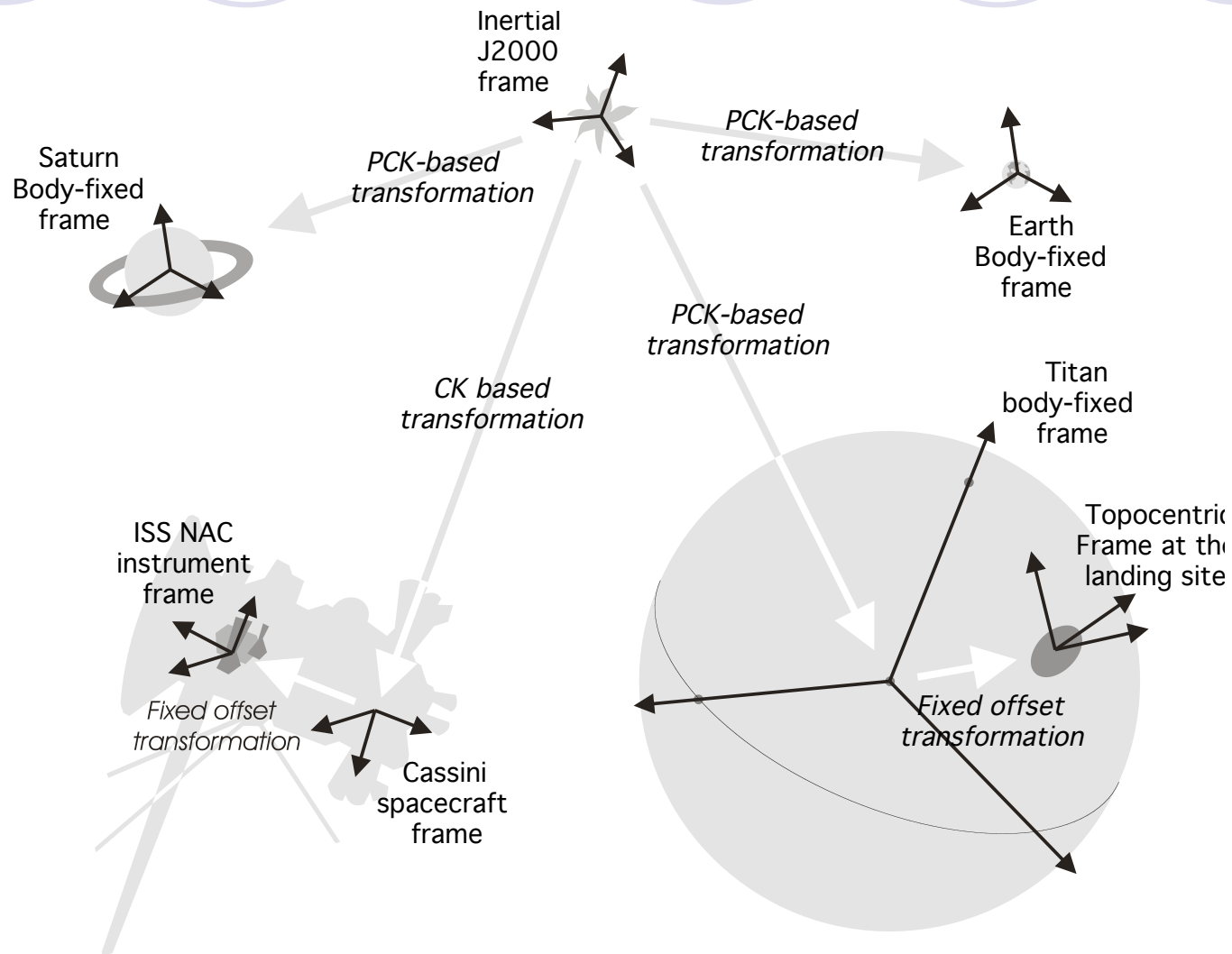
J2000 (J2000.0)

- 2000年1月1.5日（基準元期，元来はこの元期そのものをJ2000と呼ぶ）における，
 - 春分点方向をX軸
 - 赤道面をXY平面
 - 北極方向をZ軸にした右手系の慣性座標系
- SPICEの座標系のrootになっている

SPICEにおける座標系

- 座標系はJ2000に対して静的または動的に変化する
 - 座標変換（回転，並進）が必要
- SPICEでは動的に変化する座標系間の変換をユーザから隠蔽した形で処理する
 - 座標系を指定すれば内部で変換してから出力
 - 軌道情報 (spk), 自転情報 (pck), 姿勢情報 (ck), 指向情報 (ik), 座標系管理情報 (fk)
 - 明示的な座標変換のため，回転行列を用いてもよい

Sample Frame Tree and Chains

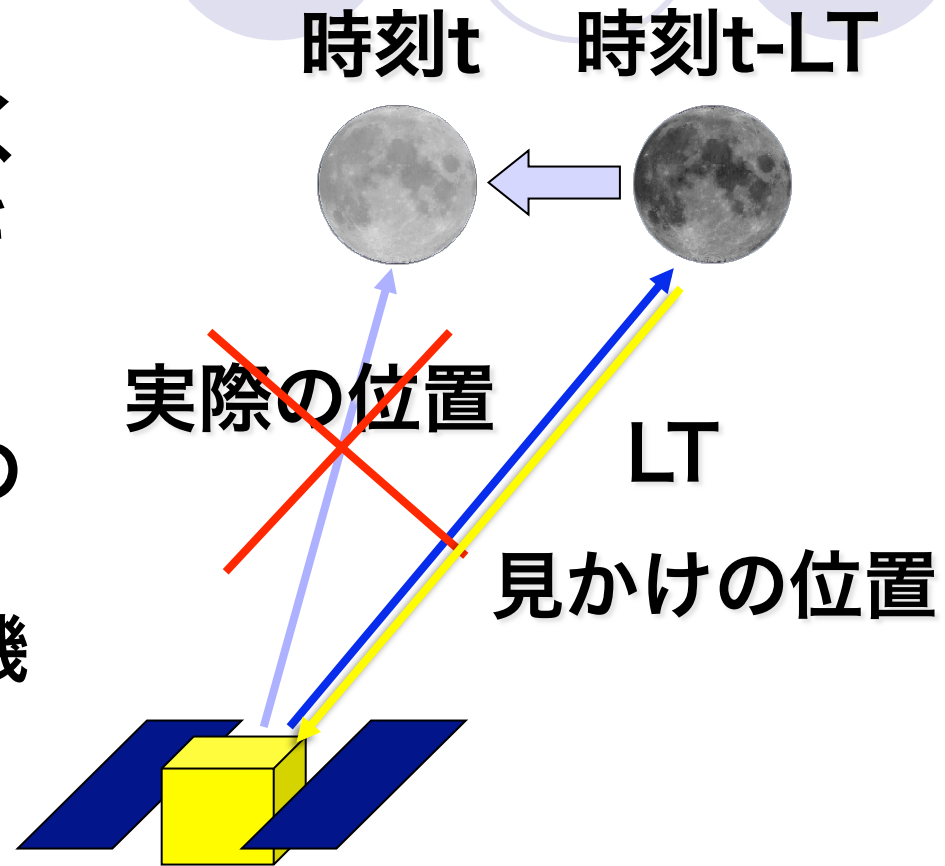


光時補正と光行差補正

- 光速が有限であるため，見かけの位置が実際の位置と異なる現象
- 光時 (light time)
 - 対象の運動と，対象までの距離に依存
- 光行差 (aberration of light)
 - 観測者の運動と，対象の方向に依存
- SPICEではこれらを簡単に補正できる
 - 補正手順はユーザから隠蔽されている

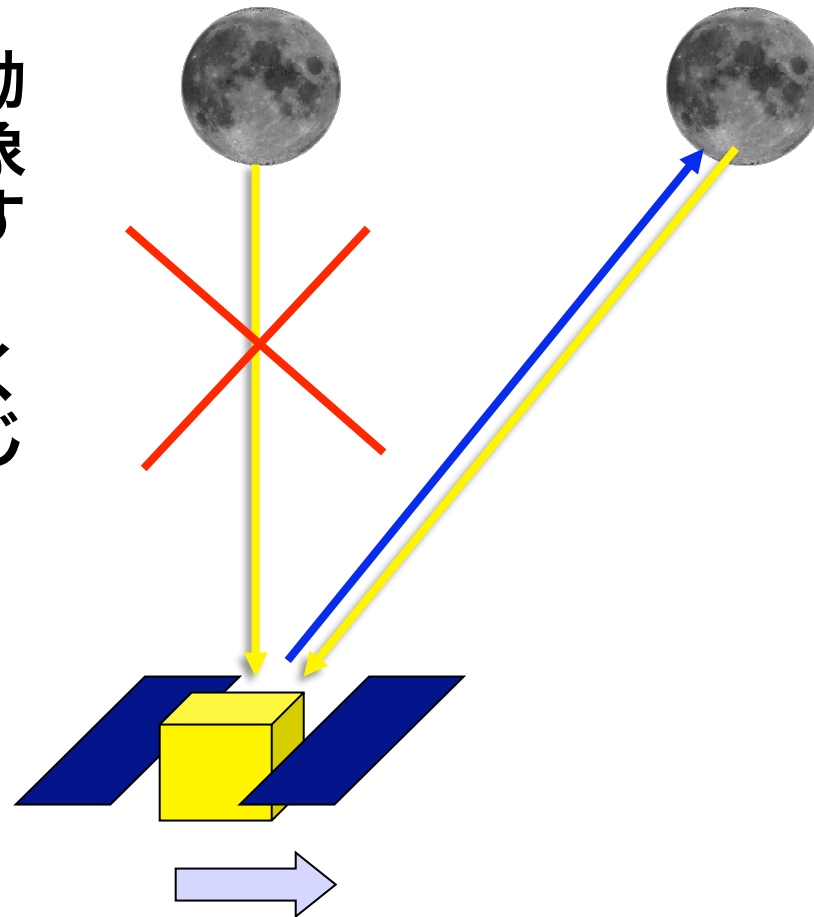
光時補正 (light time: LT)

- 時刻 t に探査機に届く光は、 $t-LT$ に対象を出発している
- 対象がある速度で移動していると、 LT の間に位置が変わる
- 時刻 t における探査機から見た対象の見かけの位置は、実際には $t-LT$ での対象の位置



光行差補正 (stellar aberration)

- 探査機がある速度で運動しているとき、観測対象の見かけの位置が移動する
- 雨が「前」から降ってくるように見えるのと同じ原理
- 地球の場合の光行差
 - 自転による日周光行差
 - 公転による年周光行差



位置情報の取得

- 何種類か関数がある

- `spkpos_c`, `spkezr_c`, `spkezp_c`...

- 違いは引数の与え方や、読み出し内容（位置/速度）など

- 基本的な引数

- 時刻 (ET)

- 対象物

- 観測者 (原点)

- 座標系

- 光時補正, 光行差補正の有無, 手法

位置情報の取得例

```
SpiceDouble      lt, pos[3];  
spkpos_c ("MUSES-C", et, "IAU_EARTH",  
         "LT+S", "EARTH", pos, &lt;);
```

- 入力用引数
 - "MUSES-C": 対象
 - et: 時刻
 - "IAU_EARTH": 座標系 (地球固定座標系)
 - "LT+S": 光時補正と光行差補正を実施 (不要なら "NONE")
 - "EARTH": 観測者 (原点)
- 出力用引数 (ポインタ渡し)
 - pos: 位置
 - lt: 距離 (光時補正の副産物. 光速での所要時間の形で得る)

サンプル2

- 「はやぶさ」は2004年5月に地球をスイングバイした
- swingby.cは以下を計算し，出力する
 - 地球を原点とした探査機および月の三次元的な位置関係
 - 座標系は以下の二種類
 - J2000 (慣性系) (光時, 光行差補正なし)
 - IAU_EARTH (地球固定) (光時, 光行差補正あり)
 - 地球 (中心) -探査機間距離
 - 計算期間は2004/05/17 00:00 (UTC) から2週間, 1時間刻み
 - 計算結果はテキストファイルにリダイレクトして, gnuplotなどでプロットすることができる

サンプル2

● 実行例

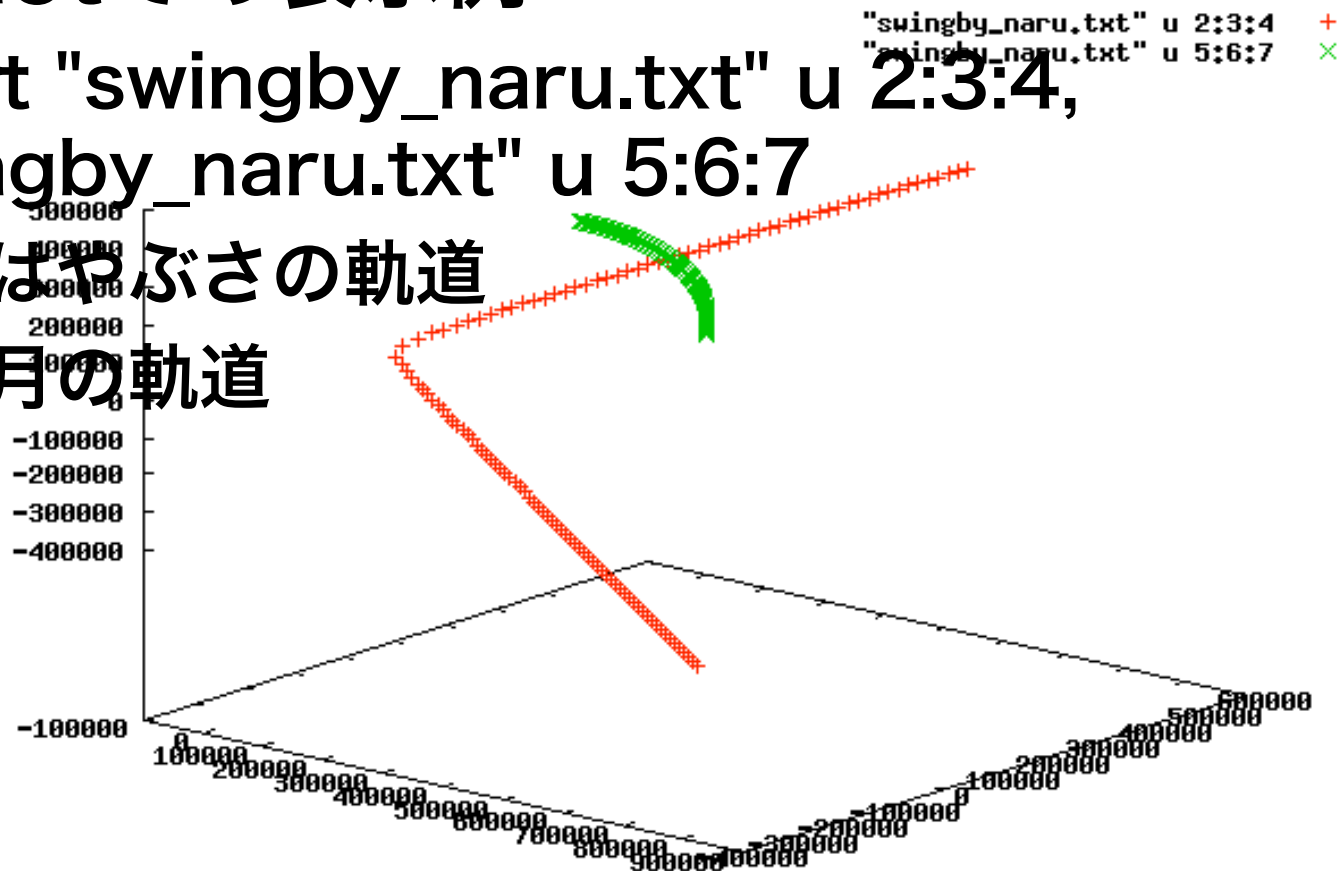
```
$ ./swingby $( cat kernel_list_swingby )  
../kernels/generic_kernels/spk/de403s.bsp is loaded.  
(中略, カーネルが多数読み込まれている)  
2004-05-17T00:00:00  
(時刻)  
-734524.5782          41762.3969          400351.9557  
(J2000系でのはやぶさの位置)  
-355598.9079          164270.4869          78935.3132  
(J2000系での月の位置)  
456478.3420           577093.7550          400152.2902  
(地球固定座標系でのはやぶさの位置)  
338844.4167           196607.7645          78794.9630  
(地球固定座標系での月の位置)  
837587.0953  
(地球-はやぶさ間の距離)  
座標系の原点は地球, 単位はすべてkm
```

サンプル2

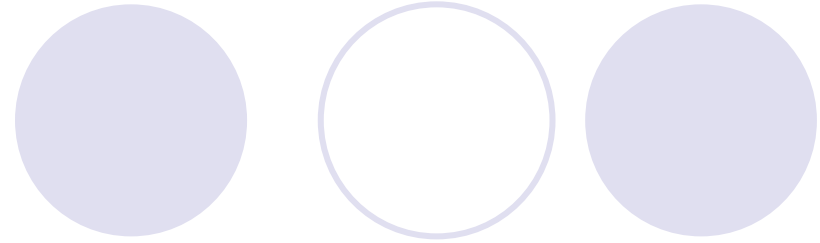
- gnuplotでの表示例

```
> splot "swingby_naru.txt" u 2:3:4,  
"swingby_naru.txt" u 5:6:7
```

- 赤がはやぶさの軌道
- 緑が月の軌道



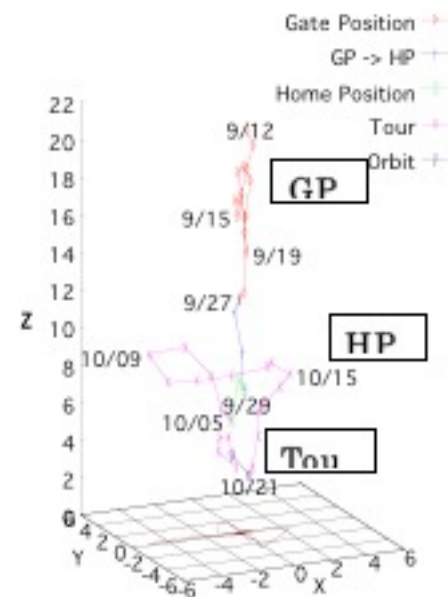
使用したカーネル

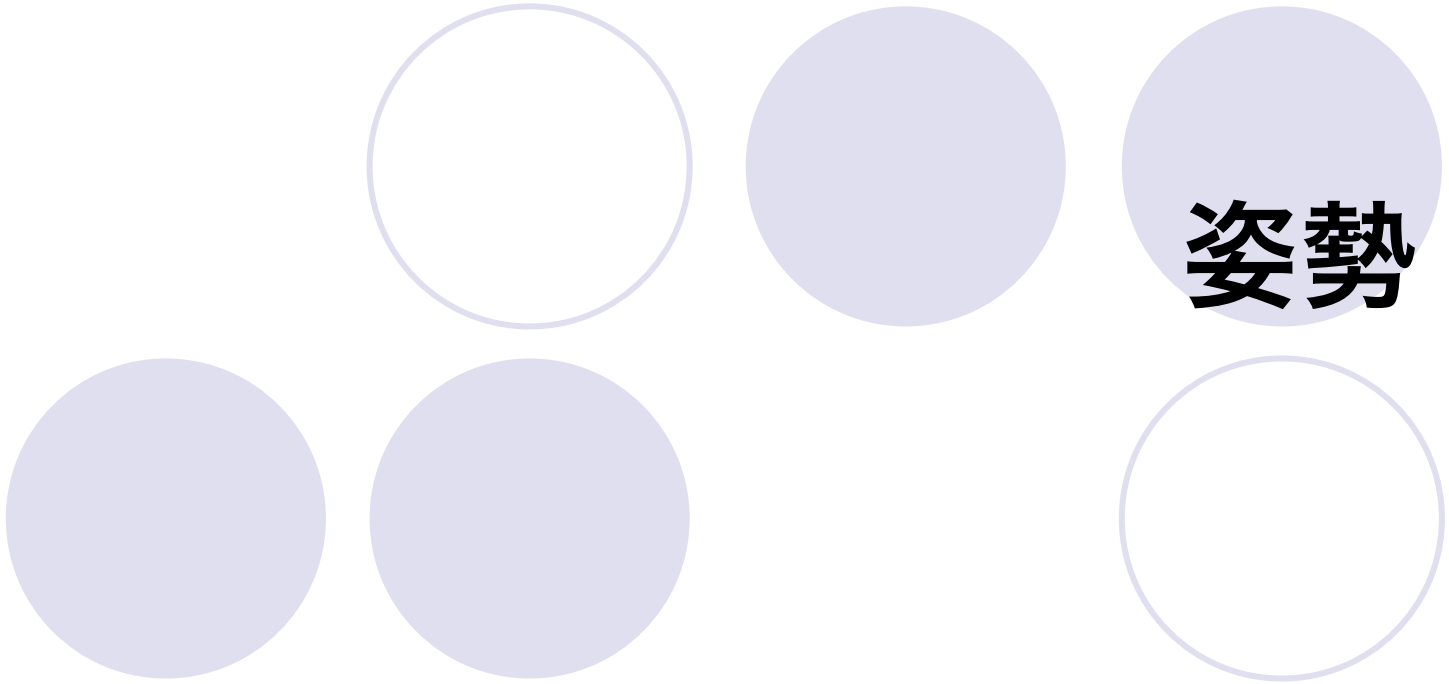


- **de403s.bsp**
 - 惑星と月の軌道情報 (spk)
- **pck00008.tpc**
 - 惑星と月の自転, 形状情報 (pck)
- **hayabusa_20040516-20040531.bsp**
 - はやぶさの軌道情報 (spk)

応用

- ランデブー期間中
(2005年9月12日以降) のはやぶさの
イトカワに対する相
対位置履歴の取得
 - 座標系とその原点を交
更
 - “HAYABUSA_HP”
 - 必要なspkも交
わる





姿勢

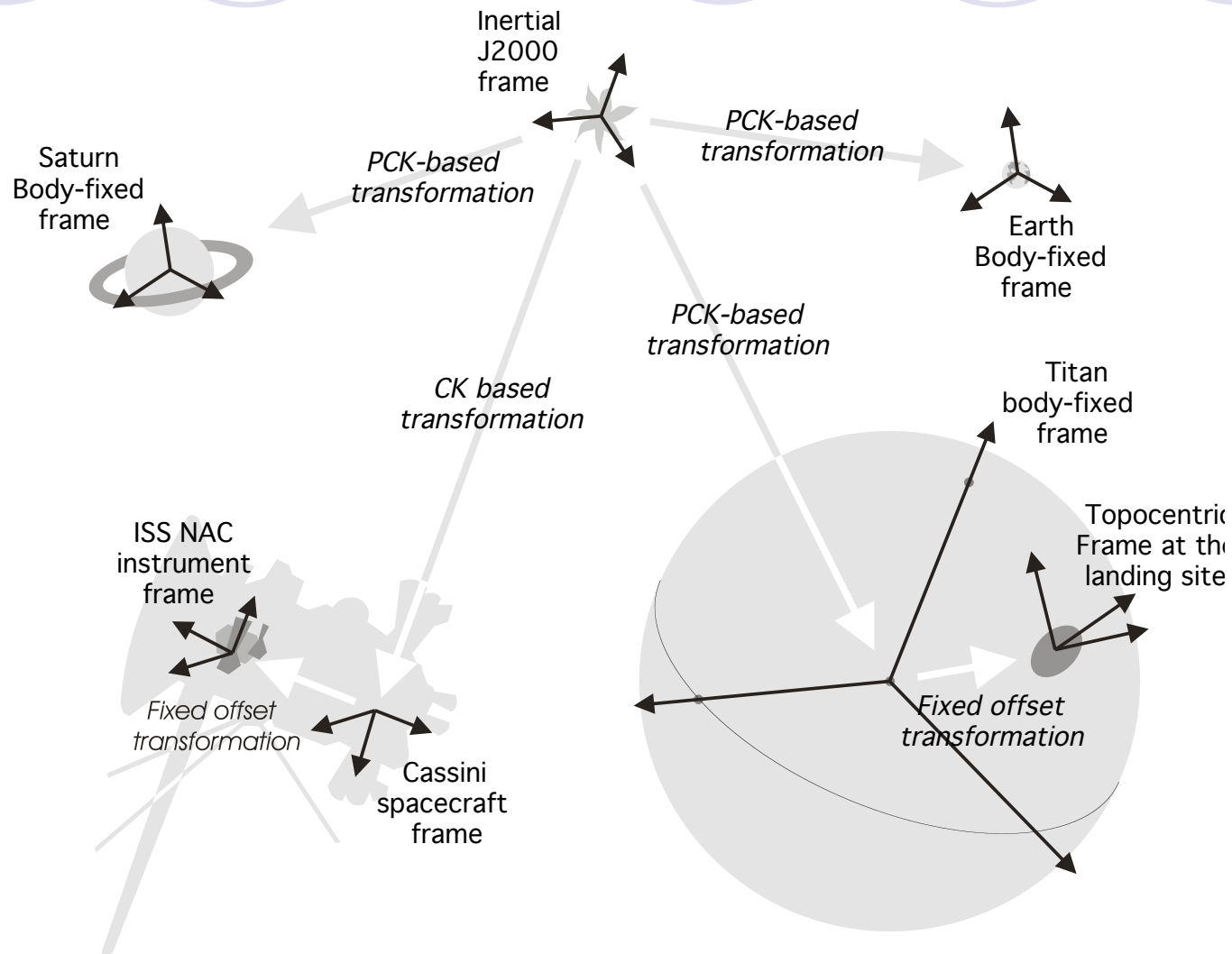
探査機の姿勢決定

- 太陽センサでおおまかな姿勢を検出
- 星センサ (ST: Star Tracker) で恒星を観測して、精密な姿勢を検出
- ジャイロで短時間の姿勢変動を測定
- 定期的に自らの姿勢情報を地上へ送信
 - 姿勢情報カーネル(ck)の源泉
 - 探査機構体座標系が、基準座標系 (例えば J2000) に対してどこを向いているか、という形式の情報

観測機器の向き

- 探査機構体に対して、どのように取り付けられているか
- 打ち上げ前に精密な測定
- 打ち上げ後にも確認可能（恒星観測など）
- 取り付け情報カーネル (ik, fk) の源泉
- ある時点でどこを向いているかは,
 - 探査機の向き + 観測機器の向き
で決定される
- ただし、「何に対して」の向きかを定義しなければならない

Sample Frame Tree and Chains



回転変換の表現方法



- オイラー角
- 方向余弦行列
- 四元数 (クォータニオン)

SPICE toolkitでの回転変換

- 探査機の姿勢など，変化するものはクォータニオンで情報を保存 (ck)
- 姿勢の変化レートはroll, pitch, yawの角度レートとする
- 座標変換など，回転の計算では回転行列を用いる
- 行列同士のかけ算，行列-ベクトルのかけ算などの関数が整備されている

サンプル3



- **lunar_footprint.c** はやぶさの地球スイングバイ時に実施した月観測の状況を計算する
 - 月面上における衛星直下点の経緯度
 - カメラ視線ベクトルと月面の交点（footprint, すなわち視野中心）の経緯度
 - footprint位置における太陽光の入射角, 出射角, 位相角
- 軌道情報に加え, 探査機の姿勢と観測機器の視野方向情報が必要

サンプル3

- 実行例

```
$ ./lunar_footprint $(cat kernel_list_swingby )  
  ../kernels/generic_kernels/spk/de403s.bsp is  
loaded.
```

(中略, カーネルが多数読み込まれている)

```
2004-05-17T09:30:00      20.15      -132.82
```

(視線が月から外れている場合, 直下点の緯度と経度のみ出力)

```
2004-05-17T09:30:10      20.16      -132.82
```

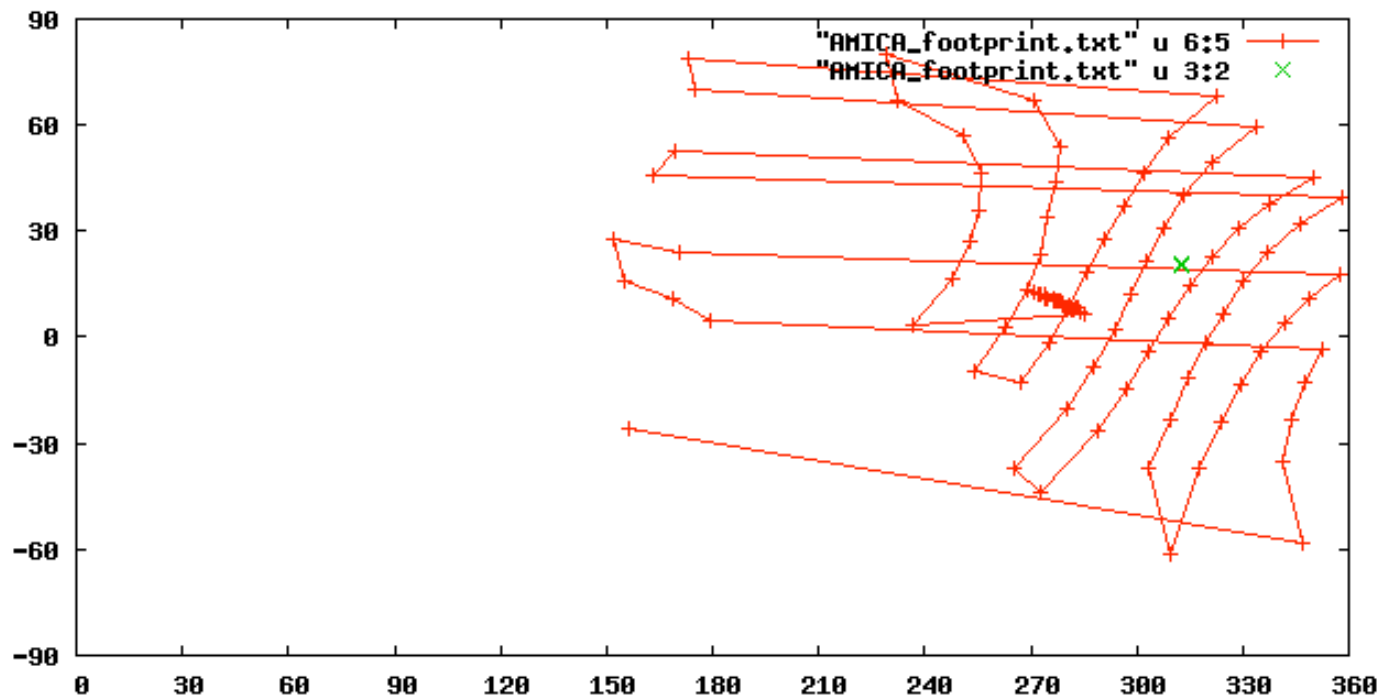
```
2004-05-17T09:30:09      -25.61      156.36
```

```
30.30    52.84      82.85
```

(視線が月に当たっている場合, 直下点に加えて視線位置
(footprint) の緯度, 経度とその地点における位相角, 入
射角, 出射角を出力)

サンプル3

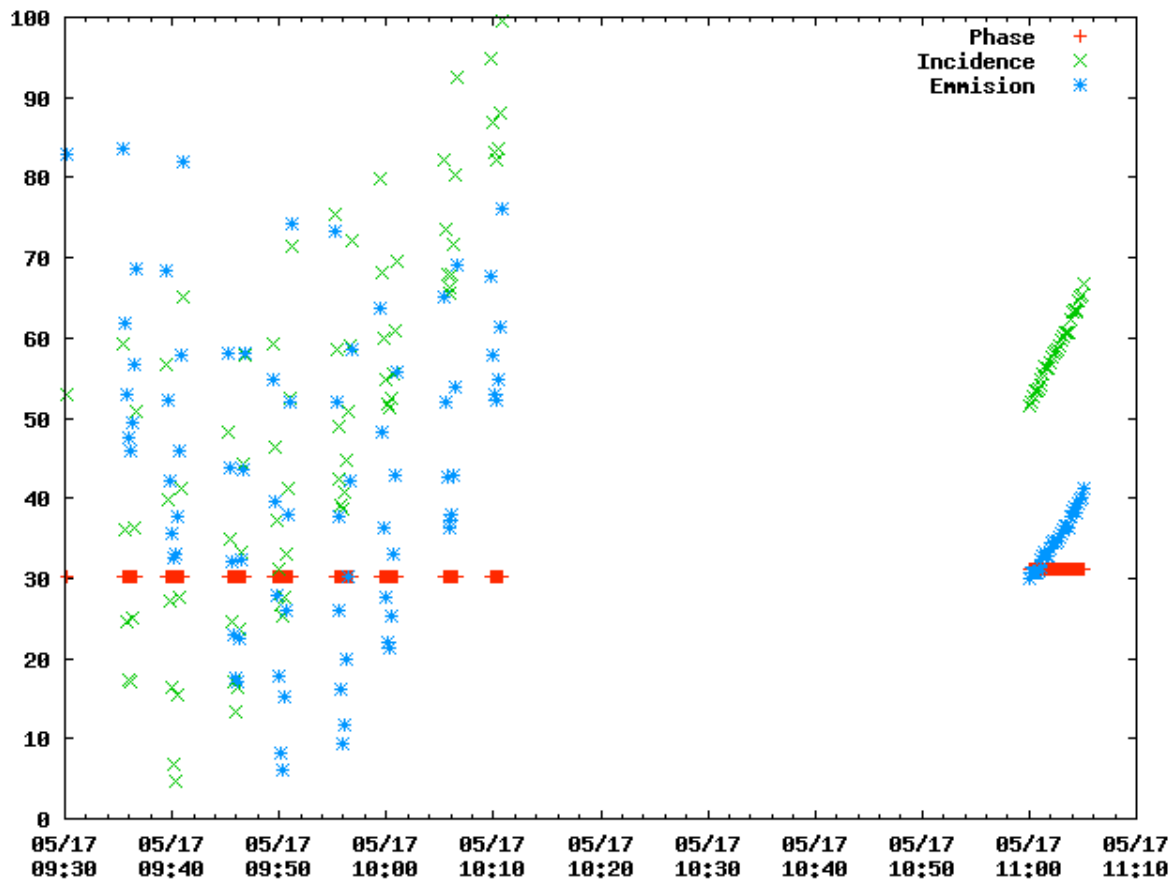
- gnuplotによるプロット例
- footprint位置の履歴



サンプル3

- gnuplotによるプロット例

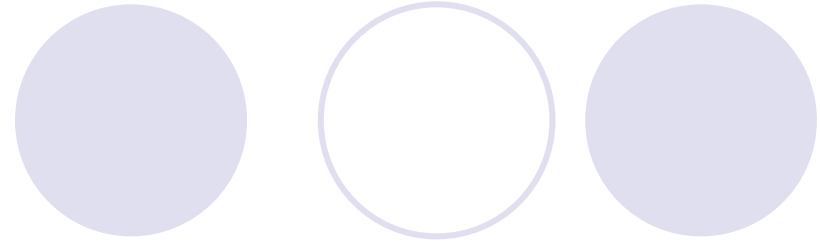
- 日照条件の履歴



gnuplot TIPS

- 2004-05-17T11:29:50という書式の時間情報をX軸に示す
 - > set xdata time
 - > set timefmt “%Y-%m-%dT%H:%M:%S”
 - > set format x “%m/%d\n%H:%M”

使用したカーネル



- サンプル2に加えて
- 20040517-20040517_att.bc
 - はやぶさの姿勢情報 (ck)
- amica31.ti
 - カメラの視線ベクトル情報 (ik)
- hayabusa.tsc, naif0009.tls
 - はやぶさの時刻情報 (sclk) とうるう秒情報 (lsk)
 - ckははやぶさの時計で情報が記録されているため

使用した関数

```
getfov_c ( instid, BCVLEN, STRLEN, STRLEN,  
          shape, frame, bsight, &n_bcv,  
          bounds );
```

- カメラの視線ベクトル情報を取得
- 入力用引数
 - instid: 機器のID (-130102 = AMICA)
 - BCVLEN, STRLEN, STRLEN: おまじない
- 出力用引数
 - shape: 機器視野形状 (AMICAは矩形)
 - frame: 視線ベクトル情報の座標系
 - bsight: 視線ベクトル
 - &n_bcv, bounds pos: 視野境界を定義するベクトルを得るための変数群

使用した関数

```
subpnt_c ( "Intercept: ellipsoid",  
          "MOON", et, "IAU_MOON", "LT+S",  
          "MUSES-C", subpoint, &trgepc, subsrfvec );
```

- 探査機の直下点を得る
- 入力用引数
 - “Intercept: ellipsoid”: 直下点の定義方法
 - “MOON”: 対象天体
 - et: 時刻
 - “IAU_MOON”: 座標系 (通常は対象天体の固定座標系)
 - “LT+S”: 光時, 光行差補正
 - “MUSES-C”: 観測者 (探査機)
- 出力用引数
 - subpoint: 直下点の三次元座標
 - &trgepc: 光時遅延を考慮した観測時刻
 - subsrfvec: 直下点-観測者のベクトル

使用した関数

```
sincpt_c ( "Ellipsoid", "MOON", et, "IAU_MOON",  
          "LT+S", "MUSES-C", frame, bsight,  
          xpoint, &trgepc, xsrfvec, &found );
```

- 視線ベクトルbsightと天体との交点を得る
- 入力用引数 (略)
- 出力用引数
 - found : 視線ベクトルと天体が交点を持つ場合真, ない場合偽となる. SpiceBoolean型の変数

使用した関数

```
reclat_c ( subpoint, &radius, &lon, &lat );
```

- 三次元座標を経緯度に変換する

```
dpr_c();
```

- degree per radian

```
ilumin_c ( "Ellipsoid",  
           "MOON", et, "IAU_MOON", "LT+S",  
           "MUSES-C", xpoint, &trgepc, xsrfvec,  
           &phase, &incidence, &emission);
```

- 指定時刻における天体上の一点での観測者から見たときの見かけの日照条件を得る