For JAXA R&D

PPP - Models, Algorithms and Implementations (2)

Tokyo Univ. of Marine Science and Technology (TUMSAT)

Tomoji TAKASU

2019-10-18 @Tokyo, Japan

PPP - Models, Algorithms and Implementation

1. 2019-10-04 **PPP models**
   geometric range, ionosphere, troposphere, antenna PCV, earth tides, wind-up, relativity, biases, coordinates
2. 2019-10-18 **PPP algorithms**
   SPP, LSQ, GN, EKF, noise-model, RAIM/QC, LAPACK/BLAS
3. 2019-11-01 **PPP data handling**
   LC, interpolation, slip detection, RINEX, SP3, ANTEX, RTCM, CSSR
4. 2019-11-22 **PPP-AR**
   UPD/FCB, EWL/WL/NL, ILS, LAMBDA, TCAR, PAR, validation
5. 2019-12-06 **INS integration**
   INS sensors, Inertial navigation, INS integration
6. 2019-12-20 **POD of satellites**
   orbit element, orbit model, reduced-dynamic, ECI-ECEF transformation, precession/nutation, EOP

(1.5 h / session)

PPP Algorithms

Notations

c	: Speed of light (m/s)
P_{r,L_i}^s	: L_i Peudorange measurement (m)
φ_{r,L_i}^s	: L_i Carrier phase measurement (cyc)
Φ_{r,L_i}^s	: L_i Phase-range measurement (m)
t_r	: Signal reception time (s)
t^s	: Signal transmission time (s)
ρ_r^s	: Geometric range (m)
$\mathbf{r}^s(t)$	: Satellite position in ECEF (m)
$\mathbf{v}^s(t)$	: Satellite velocity in ECEF (m)
$\mathbf{r}_r$	: Receiver position in ECEF (m)
$\mathbf{e}_r^s$	: LOS vector in ECEF
$\mathbf{e}_{r,enu}^s$	: LOS vector in local coordinates
ϕ_r	: Latitude of receiver position (rad)
λ_r	: Longitude of receiver position (rad)
h_r	: Ellipsoidal height of receiver (m)
H_r	: Orthometric height of receiver (m)
Az_r^s	: Azimuth angle of satellite (rad)
El_r^s	: Elevation angle of satellite (rad)
dt_r	: Receiver clock bias (s)
$dT^s(t)$	: Satellite clock bias (s)
I_r^s	: Ionospheric delay (m)
T_r^s	: Tropospheric delay (m)
f_i	: L_i carrier frequency (Hz)
λ_i	: L_i carrier wavelength (m)
B_{r,L_i}^s	: L_i Carrier phase bias (m)
N_{r,L_i}^s	: L_i Carrier phase ambiguity (cyc)
ε_P	: Code measurement error (m)
ε_Φ	: Phase measurement error (m)
ω_e	: Earth rotation velocity (rad/s)
Z_t	: Zenith total delay (m)
Z_h	: Zenith hydrostatic delay (m)
Z_w	: Zenith wet delay (m)
$m_h(El)$	: Hydrostatic mapping function
$m_w(El)$	: Wet mapping function
$\mathbf{U}(t)$	: ECEF to ECI transformation matrix
$\mathbf{E}_r$	: ECEF to local coordinates rotation matrix
$\mathbf{E}^s$	: ECEF to satellite body rotation matrix
$\mathbf{R}_x(\theta)$	: Coordinates rotation matrix around X
$\mathbf{R}_y(\theta)$	: Coordinates rotation matrix around Y
$\mathbf{R}_z(\theta)$	: Coordinates rotation matrix around Z

LSE (Least Square Estimation)

Measurement Equations

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\varepsilon}$$

$\mathbf{x}$: Parameters (n x 1) $\mathbf{H}$: Design matrix (m x n)

$\mathbf{y}$: Measurements (m x 1) $\boldsymbol{\varepsilon}$: Measurement errors (m x 1)

(n < m)

Normal Equation (NEQ)

$$J_{LS} = \frac{1}{2}\left(v_1^{\,2} + v_2^{\,2} + \ldots + v_m^{\,2}\right) = \frac{1}{2}\mathbf{v}^T\mathbf{v} = \frac{1}{2}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}})^T(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}})$$

$\mathbf{v}$: Residuals (m x 1)

$$= \frac{1}{2}\left(\mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{H}\hat{\mathbf{x}} - \hat{\mathbf{x}}^T\mathbf{H}^T\mathbf{y} + \hat{\mathbf{x}}^T\mathbf{H}^T\mathbf{H}\hat{\mathbf{x}}\right) = \min$$

Least squares cost function

$$\frac{\partial J_{LS}}{\partial \hat{\mathbf{x}}} = \frac{1}{2}\left(\mathbf{0}^T - \mathbf{y}^T\mathbf{H} - (\mathbf{H}^T\mathbf{y})^T + (\mathbf{H}^T\mathbf{H}\mathbf{x})^T + \hat{\mathbf{x}}^T\mathbf{H}^T\mathbf{H}\right)$$

$$= -\mathbf{y}^T\mathbf{H} + \hat{\mathbf{x}}^T\mathbf{H}^T\mathbf{H} = \mathbf{0}^T$$

$$\mathbf{H}^T\mathbf{H}\hat{\mathbf{x}} = \mathbf{H}^T\mathbf{y}$$

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}$$

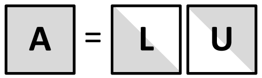
Normal equation (NEQ)

$\hat{\mathbf{x}}$: Estimated parameters (n x 1)

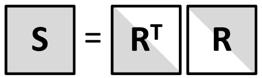
(BLUE: best linear unbiased estimate)

Numerical Solution of LSE

LU-decomposition

[L,U] = lu(A) 

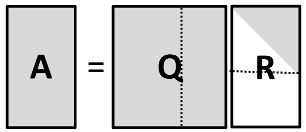
Cholesky-decomposition

R = chol(S) 

$S=S^T$

QR-decomposition

[Q,R] = qr(A,0)

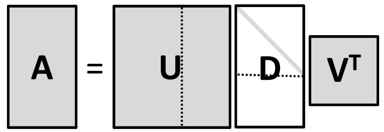


$QQ^T=I$

(Householder, Givens rot, MGS)

SVD (Singular value decomposition)

[U,D,V] = svd(A,0)



$UU^T=I$, $VV^T=I$

CPU Time to Solve LSE

Solution	MATLAB Notation	CPU Time
NEQ + Auto	x = (H'*H)¥(H'*y);	13.3 s
NEQ + LU	[L,U] = lu(H'*H); x = U¥(L¥(H'*y));	12.6 s
NEQ + Cholesky	R = chol(H'*H); x = R¥(R'¥(H'*y));	12.4 s
QR	[Q,R] = qr(H,0); x = R¥(Q'*y);	66.0 s
SVD	[U,D,V] = svd(H,0); x = V*(D¥(U'*y));	126.7 s
Pseudo-Inverse	x = pinv(H)*y;	151.9 s

size(H)=[50000,5000], Core i7-3930K (6C/12T, 3.2 GHz), MATLAB R2011b, Windows 10

Weighted LSE

Weighted LSE

$$\mathbf{W} = \mathbf{R}^{-1} \qquad \left(E\{\boldsymbol{\varepsilon}\} = \mathbf{0},\ E\{\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\} = \mathbf{R} \right)$$

$$J_{WLS} = \frac{1}{2}\mathbf{v}^T\mathbf{W}\mathbf{v} = \frac{1}{2}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}})^T\mathbf{W}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}) = \min$$

$$\mathbf{H}^T\mathbf{W}\mathbf{H}\hat{\mathbf{x}} = \mathbf{H}^T\mathbf{W}\mathbf{y}$$

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\mathbf{y}$$

$\mathbf{R}$: Covariance matrix of measurements (m x m)

$\mathbf{W}$: Weighting matrix (m x m)

NEQ for weighted LSE

w/o correlation between Measurements

$$\mathbf{W} = \mathbf{R}^{-1} = \mathrm{diag}(\sigma_1^{-2}, \sigma_2^{-2}, \ldots, \sigma_m^{-2}) \qquad \left(E\{\varepsilon_i\varepsilon_j\} = \begin{cases} \sigma_i^2 & (i = j) \\ 0 & (i \neq j) \end{cases} \right)$$

$$\mathbf{H}^{*T}\mathbf{H}^*\hat{\mathbf{x}} = \mathbf{H}^{*T}\mathbf{y}^*$$

$$\hat{\mathbf{x}} = (\mathbf{H}^{*T}\mathbf{H}^*)^{-1}\mathbf{H}^{*T}\mathbf{y}^*$$

$$\mathbf{y}^* = \mathbf{W}^{1/2}\mathbf{y} = \left(\frac{y_1}{\sigma_1}, \frac{y_2}{\sigma_2}, \ldots, \frac{y_m}{\sigma_m} \right)^T \qquad \mathbf{H}^* = \mathbf{W}^{1/2}\mathbf{H} = \begin{pmatrix} H_{11}/\sigma_1 & H_{12}/\sigma_1 & \cdots & H_{1n}/\sigma_1 \\ H_{21}/\sigma_2 & H_{22}/\sigma_2 & \cdots & H_{2n}/\sigma_2 \\ \vdots & \vdots & \ddots & \vdots \\ H_{m1}/\sigma_m & H_{2n}/\sigma_m & \cdots & H_{mn}/\sigma_m \end{pmatrix}$$

Estimated Parameter Errors

Estimated Parameter Errors

$$
\begin{aligned}
\delta\mathbf{x} &= \mathbf{x} - \hat{\mathbf{x}} \\
&= \mathbf{x} - (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\mathbf{y} \\
&= \mathbf{x} - (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}(\mathbf{H}\mathbf{x} + \boldsymbol{\varepsilon}) \\
&= -(\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\boldsymbol{\varepsilon}
\end{aligned}
$$

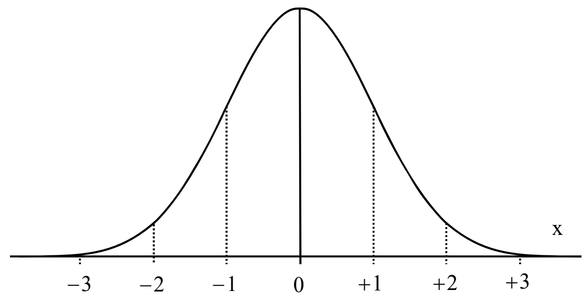
Covariance of Estimated Parameter Errors

$$
\begin{aligned}
\mathbf{P} &= \mathrm{E}\left\{\delta\mathbf{x}\delta\mathbf{x}^T\right\} \\
&= (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\,\mathrm{E}\left\{\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\right\}\mathbf{W}\mathbf{H}(\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1} \\
&= (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\mathbf{W}^{-1}\mathbf{W}\mathbf{H}(\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1} \\
&= (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1} \\
&= (\mathbf{H}^{*T}\mathbf{H}^*)^{-1}
\end{aligned}
$$

$$\mathrm{E}\{\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\} = \mathbf{R} = \mathbf{W}^{-1}$$

$\sigma_{x_i}^2 = P_{ii}$: Variance of i^{th} estimated parameter error

Standard Normal Distribution



$$f(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)$$

Cumulative Distribution Function (CDF)

$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}\exp\left(-\frac{t^2}{2}\right)dt = \frac{1}{2}\left(1 - \mathrm{erf}\left(\frac{1}{\sqrt{2}}\right)\right)$$

$$
\begin{aligned}
P(-1 \le x \le 1) &= \Phi(1) - \Phi(-1) = 0.682689 \\
P(-2 \le x \le 2) &= \Phi(2) - \Phi(-2) = 0.954500 \\
P(-3 \le x \le 3) &= \Phi(2) - \Phi(-2) = 0.997300 \\
P(-4 \le x \le 4) &= \Phi(3) - \Phi(-3) = 0.999937
\end{aligned}
$$

$\mathrm{erf}(x) = \frac{1}{\sqrt{\pi}}\int_{-x}^{x} e^{-t^2}dt$: Error function

Nonlinear LSE

Nonlinear Measurement Equations

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\varepsilon}$$

$\mathbf{h}(\mathbf{x})$: Measurement model functions (m x 1) (n < m)

$\mathbf{x}_0$: Initial parameters (n x 1)

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0} = \begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \\ \frac{\partial h_m}{\partial x_1} & \frac{\partial h_m}{\partial x_2} & \cdots & \frac{\partial h_m}{\partial x_n} \end{pmatrix}$$

: Partial derivatives (Jacobian matrix) (m x n)

Normal Equation (NEQ)

$$\mathbf{h}(\mathbf{x}) = \mathbf{h}(\mathbf{x}_0) + \mathbf{H}(\mathbf{x} - \mathbf{x}_0) + \ldots$$

$$\mathbf{y} \approx \mathbf{h}(\mathbf{x}_0) + \mathbf{H}(\mathbf{x} - \mathbf{x}_0) + \boldsymbol{\varepsilon}$$

$$\mathbf{y} - \mathbf{h}(\mathbf{x}_0) = \mathbf{H}(\mathbf{x} - \mathbf{x}_0) + \boldsymbol{\varepsilon}$$

$$\mathbf{H}^T\mathbf{W}\mathbf{H}(\hat{\mathbf{x}} - \mathbf{x}_0) = \mathbf{H}^T\mathbf{W}(\mathbf{y} - \mathbf{h}(\mathbf{x}_0))$$

$$\hat{\mathbf{x}} = \mathbf{x}_0 + (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}(\mathbf{y} - \mathbf{h}(\mathbf{x}_0))$$

GN (Gauss-Newton) Method

$$\hat{\mathbf{x}}_0 = \mathbf{x}_0$$

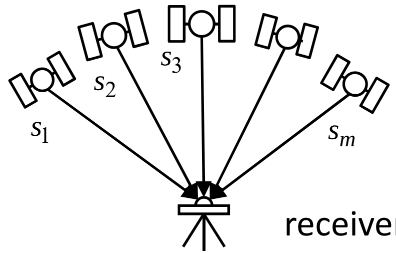
$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i + (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}(\mathbf{y} - \mathbf{h}(\hat{\mathbf{x}}_i))$$

$$\hat{\mathbf{x}} = \lim_{i\to\infty} \hat{\mathbf{x}}_i$$

Note: GN does not always converge. In ill conditions, it often diverges. Consider other non-linear LSE methods like LM (Levenberg-Marquardt).

SPP (Single Point Positioning)

Parameters and Measurements



$$\mathbf{x} = (\mathbf{r}_r^T, cdt_r)^T, \mathbf{y} = (P_r^{s1}, P_r^{s2}, P_r^{s3}, \ldots, P_r^{sm})^T$$

Measurement Equations and Partial Derivatives

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\varepsilon}$$

$$\mathbf{h}(\mathbf{x}) = \begin{pmatrix} \rho_r^{s1} + c(dt_r - dT^{s1}(t^{s1})) + I_r^{s1} + T_r^{s1} + d_r^{s1} \\ \rho_r^{s2} + c(dt_r - dT^{s2}(t^{s2})) + I_r^{s2} + T_r^{s2} + d_r^{s2} \\ \rho_r^{s3} + c(dt_r - dT^{s3}(t^{s3})) + I_r^{s3} + T_r^{s3} + d_r^{s3} \\ \vdots \\ \rho_r^{sm} + c(dt_r - dT^{sm}(t^{sm})) + I_r^{sm} + T_r^{sm} + d_r^{sm} \end{pmatrix} \quad \mathbf{H} = \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}} = \begin{pmatrix} -\mathbf{e}_r^{s1T} & 1 \\ -\mathbf{e}_r^{s2T} & 1 \\ -\mathbf{e}_r^{s3T} & 1 \\ \vdots & \vdots \\ -\mathbf{e}_r^{smT} & 1 \end{pmatrix}$$

Nonlinear LSE by GN

$$\hat{\mathbf{x}}_0 = \mathbf{x}_0$$

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i + (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}(\mathbf{y} - \mathbf{h}(\hat{\mathbf{x}}_i)) \qquad \mathbf{P} = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1} = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & \sigma_{xt} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} & \sigma_{yt} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 & \sigma_{zt} \\ \sigma_{xt} & \sigma_{yt} & \sigma_{zt} & \sigma_t^2 \end{pmatrix}$$

$$\hat{\mathbf{x}} = \lim_{i \to \infty} \hat{\mathbf{x}}_i = (\hat{\mathbf{r}}_r^T, c\hat{d}t_r)^T$$

Partial Derivatives of Range

Peudorange and Phase-range Measurement Models

$$P_r^s = \rho_r^s + c(dt_r - dT^s(t^s)) + I_r^s + T_r^s + d_r^s + \varepsilon_P$$

$$\Phi_r^s = \rho_r^s + c(dt_r - dT^s(t^s)) - I_r^s + T_r^s + d_r^s + \lambda d_{pw} + B_r^s + \varepsilon_\Phi$$

Partial Derivatives of Range by Receiver Position

$$\frac{\partial \rho_r^s}{\partial x_r} \approx \frac{\partial \sqrt{(x^s - x_r)^2 + (y^s - y_r)^2 + (y^s - y_r)^2}}{\partial x_r} = \frac{1}{2}\left\{(x^s - x_r)^2 + (y^s - y_r)^2 + (y^s - y_r)^2\right\}^{-1/2} \frac{\partial (x^s - x_r)^2}{\partial x_r}$$

$$= \frac{-2(x^s - x_r)}{2\sqrt{(x^s - x_r)^2 + (y^s - y_r)^2 + (y^s - y_r)^2}} = \frac{-(x^s - x_r)}{\rho_r^s}$$

$$\frac{\partial \rho_r^s}{\partial y_r} \approx \frac{-(y^s - y_r)}{\rho_r^s}, \quad \frac{\partial \rho_r^s}{\partial z_r} \approx \frac{-(z^s - z_r)}{\rho_r^s}$$

$$\frac{\partial \rho_r^s}{\partial \mathbf{r}_r} = \left(\frac{\partial \rho_r^s}{\partial x_r}, \frac{\partial \rho_r^s}{\partial y_r}, \frac{\partial \rho_r^s}{\partial z_r}\right) \approx -\left(\frac{x^s - x_r}{\rho_r^s}, \frac{y^s - y_r}{\rho_r^s}, \frac{z^s - z_r}{\rho_r^s}\right) = -\frac{(\mathbf{r}^s(t^s) - \mathbf{r}_r)^T}{\rho_r^s} = -\mathbf{e}_r^{sT}$$

$$\frac{\partial P_r^s}{\partial \mathbf{r}_r} \approx -\mathbf{e}_r^{sT}, \quad \frac{\partial \Phi_r^s}{\partial \mathbf{r}_r} \approx -\mathbf{e}_r^{sT}$$

$$\mathbf{r}_r = (x_r, y_r, z_r)^T, \quad \mathbf{r}^s(t^s) = (x^s, y^s, z^s)^T$$

Noise Model

Measurement Errors

$$\varepsilon_P \sim N(0, \sigma_P^{\ 2})$$

$$\varepsilon_\Phi \sim N(0, \sigma_\Phi^{\ 2})$$

$N(0, \sigma^2)$: Normal distribution with mean 0 and variance σ^2

Noise Model (RTKLIB)

$$\sigma_{\Phi,}^{\ 2} = F_s^{\ 2}\left\{a^2 + \frac{b^2}{\sin^2(El)}\right\} \quad (El > 0)$$

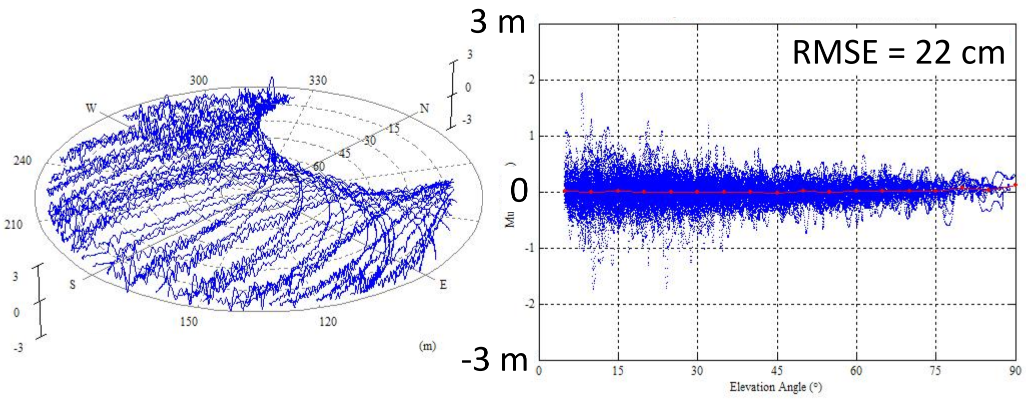
$$\sigma_P^{\ 2} = R^2 \sigma_\Phi^{\ 2}$$

a, b : Noise factor of phase (m) (default: 3 mm, 3 mm)

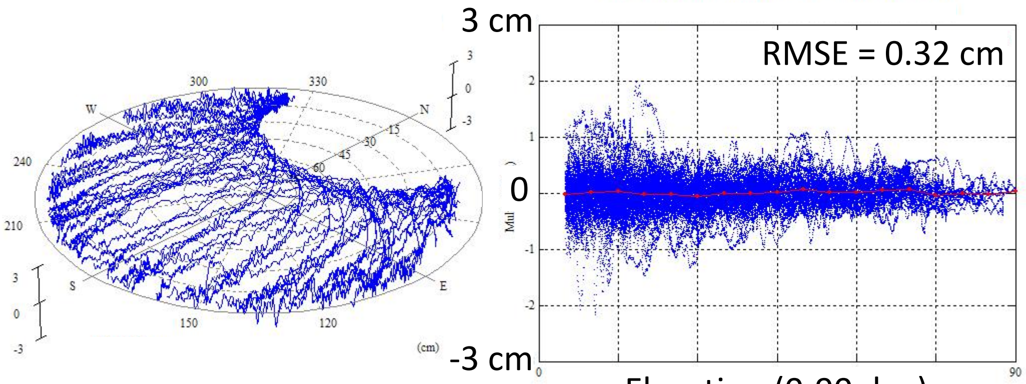
F_s : Noise factor of system (1.5: GLONASS, 1: others)

R : Noise ratio of peudorange wrt phase (default: 100)

Peudorange



Phase-range



Receiver Noise (NovAtel OEMV-3 + GPS-702-GG, GPS L1C/A)

SPP by GN

Initial parameters for GN

$$\mathbf{x}_0(t_k) = \begin{cases} \mathbf{0} & (k=1) \\ \hat{\mathbf{x}}(t_{k-1}) & (k>1) \end{cases}$$

Condition of convergence

$$\left\| \hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{i-1} \right\| < \delta_{thres}$$

Condition of divergence

$$i > i_{MAX}$$

δ_{thres} : Threshold to exit iteration loop (RTKLIB: 10^{-4} m)

i_{MAX} : Max number of iterations (RTKLIB: 10)

Example of Convergence of Solutions ($X_0 = 0$)

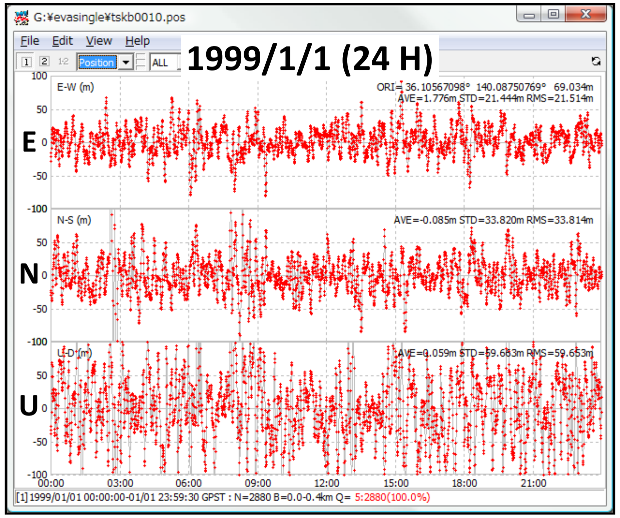
i	x (m)	y (m)	z (m)	cdt_r (m)
0	0.0000000	0.0000000	0.0000000	0.0000000
1	-4739338.8790644	3968053.3426383	4470195.0681293	1290751.6350707
2	-3990084.5939062	3334559.7805777	3763444.6383541	50195.3310677
3	-3957255.7455862	3310242.1098583	3737755.6233736	510.7878812
4	-3957205.2229884	3310203.7001970	3737718.0508664	432.5789153
5	-3957205.1820501	3310203.6651692	3737718.0078941	432.4910365
6	-3957205.1820116	3310203.6651363	3737718.0078537	432.4909539
7	-3957205.1820116	3310203.6651363	3737718.0078536	432.4909538
8	-3957205.1820116	3310203.6651363	3737718.0078536	432.4909538
9	-3957205.1820116	3310203.6651363	3737718.0078536	432.4909538

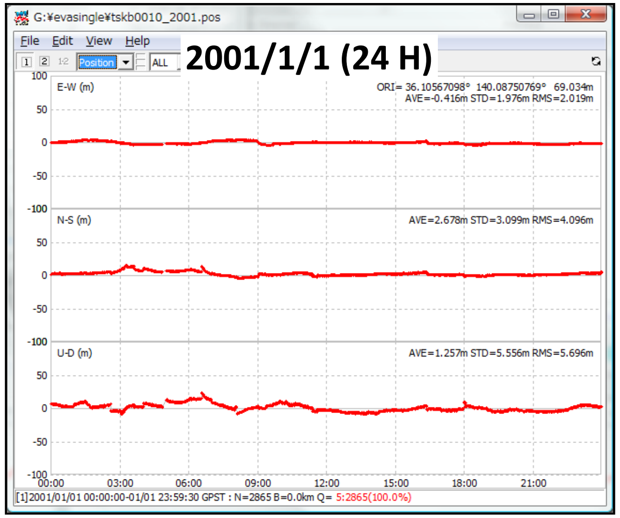
2001/1/1 0:00:00, TKSB, processed by RTKLIB 2.2.1 (n = 8)

Examples of SPP Solutions

RMSE (m)
E: 21.51
N: 33.81
U: 59.65

100 m



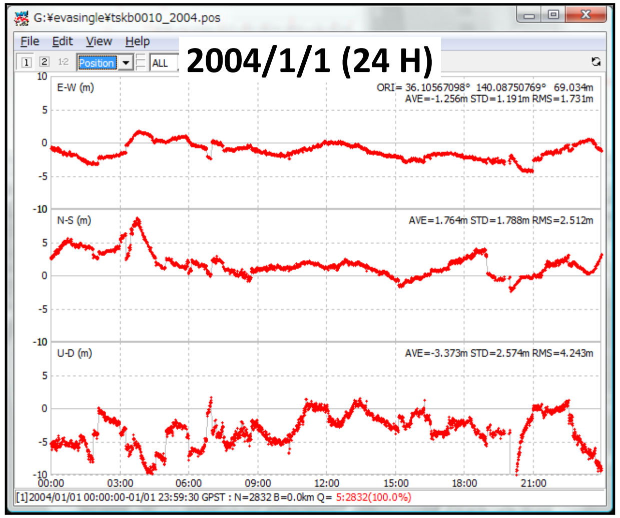


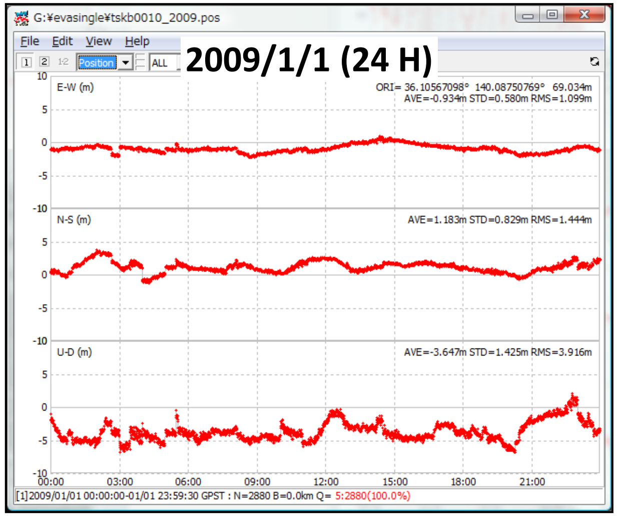
RMSE (m)
E: 2.02
N: 4.10
U: 5.70

100 m

RMSE (m)
E: 1.73
N: 2.51
U: 4.24

10 m





RMSE (m)
E: 1.10
N: 1.44
U: 3.92

10 m

IGS Station TSKB, by RTKLIB 2.3.0

Validation and RAIM

Chi-Square and DOP Test

$$S(\hat{\mathbf{x}}) = (\mathbf{y}-\mathbf{h}(\hat{\mathbf{x}}))^T \mathbf{W}(\mathbf{y}-\mathbf{h}(\hat{\mathbf{x}})) = \sum_{i=1}^{m} \frac{(y_i - h_i(\hat{\mathbf{x}}))^2}{\sigma_i^2} \sim \chi^2_{m-n} \quad (m > n)$$

$$GDOP = \sqrt{\mathrm{trace}\left\{(\mathbf{H}^T\mathbf{H})^{-1}\right\}}$$

if $S(\hat{\mathbf{x}}) > \chi^2_{m-n,\alpha}$ **or** $GDOP > GDOP_{max}$ **then**
  reject solution $\hat{\mathbf{x}}$
end

χ^2_p : Chi-square distribution of with p degree freedom

α : Significance level of chi-square test

$GDOP_{max}$: Max GDOP

RAIM FDE

if $S(\hat{\mathbf{x}}) > \chi^2_{m-n,\alpha}$ **then** $(m > n+1)$
  for i **in** 1 ... m
    exclude y_i
    solve LSE $\rightarrow \hat{\mathbf{x}}_{e,i}$
  end
  select i **with** $\min(S(\hat{\mathbf{x}}_{e,i}))$
  if $S(\hat{\mathbf{x}}_{e,i}) > \chi^2_{m-n-1,\alpha}$ **then**
    exit as error
  else
    $\hat{\mathbf{x}} = \hat{\mathbf{x}}_{e,i}$
  end
end

RAIM: Receiver autonomous integrity monitoring
FDE : Fault detection and exclusion)

Kalman Filter (KF)

Linear system/measurement models

System Model : $\mathbf{x}_k = \mathbf{\Phi}_k \mathbf{x}_{k-1} + \mathbf{q}_k \quad (\mathbf{q}_k \sim N(\mathbf{0}, \mathbf{Q}_k))$

Measurement Model : $\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \boldsymbol{\varepsilon}_k \quad (\boldsymbol{\varepsilon}_k \sim N(\mathbf{0}, \mathbf{R}_k))$

Time Update

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_{k-1} \hat{\mathbf{x}}_{k-1}^+$$

$$\mathbf{P}_k^- = \mathbf{\Phi}_k \mathbf{P}_{k-1}^+ \mathbf{\Phi}_k^T + \mathbf{Q}_k$$

$\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-$

$\hat{\mathbf{x}}_k^+, \mathbf{P}_k^+$

Measurement Update

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$$

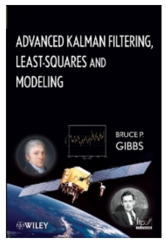
$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \ldots$

$\hat{\mathbf{x}}_1^+, \hat{\mathbf{x}}_2^+, \hat{\mathbf{x}}_3^+, \ldots$

Reference

B. P. Gibbs, Advanced Kalman filtering, least-squares and modeling, A John Willey & Sons, 2011



Extended KF (EKF)

Nonlinear system/measurement models

System Model : $\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}) + \mathbf{q}_k \quad (\mathbf{q}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{Q}_k))$

Measurement Model : $\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{\varepsilon}_k \quad (\boldsymbol{\varepsilon}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{R}_k))$

$$\dot{\mathbf{x}} = \mathbf{g}(t, \mathbf{x}) + \mathbf{r}(t) \rightarrow$$

$$\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_{k-1}^+ + \int_{t-1}^{t} \mathbf{g}(\tau, \mathbf{x})\, d\tau$$

$$\boldsymbol{\Phi}(t_{k-1}) = \mathbf{I},\ \dot{\boldsymbol{\Phi}} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \boldsymbol{\Phi}$$

Time Update

$$\hat{\mathbf{x}}_k^- = \mathbf{f}_k(\hat{\mathbf{x}}_{k-1}^+)$$

$$\mathbf{P}_k^- = \boldsymbol{\Phi}_k \mathbf{P}_{k-1}^+ \boldsymbol{\Phi}_k^T + \mathbf{Q}_k$$

$\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-$

$\hat{\mathbf{x}}_k^+, \mathbf{P}_k^+$

Measurement Update

$$\left.\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)) \end{aligned}\right\} (1)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \ldots$

$\hat{\mathbf{x}}_1^+, \hat{\mathbf{x}}_2^+, \hat{\mathbf{x}}_3^+, \ldots$

$$\boldsymbol{\Phi}_k = \left.\frac{\partial \mathbf{f}_k(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x} = \hat{\mathbf{x}}_{k-1}^+},\quad \mathbf{H}_k = \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x} = \hat{\mathbf{x}}_k^-}$$

Iterated EKF (IEKF)

Repeat (1) until it converges. Refer 9.1.2 of Gibbs (2011) for details.

Outlier Detection and Exclusion

Objectives

Detect erroneous data
Reject and exclude outliers
Avoid filter divergence

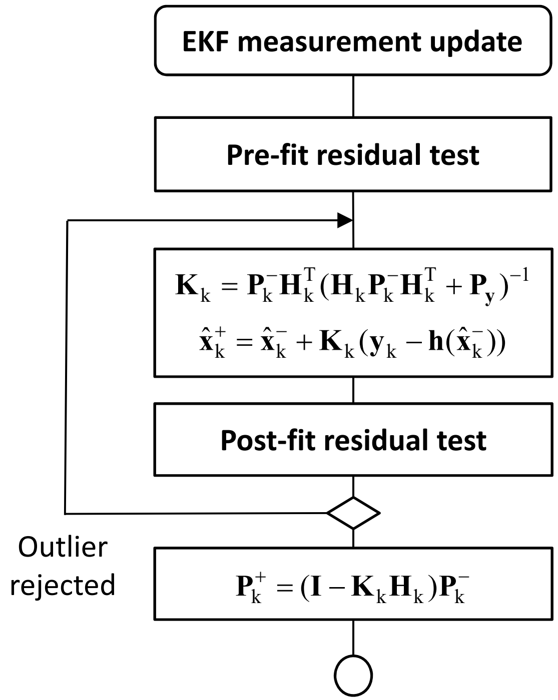
Pre-fit Residuals Test

$\mathbf{P}_y = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k$

for i **in** 1 … m
  $\nu_i = y_i - h_i(\hat{\mathbf{x}}_k^-)$
  if $\nu_i^2 > n_{pre-fit}^2 P_{y,ii}$ **then**
    reject y_i **and flag** i as potential cycle-slip (phase)
  end
end

Post-fit Residuals Test

for i **in** 1 … m
  $\nu_i = y_i - h_i(\hat{\mathbf{x}}_k^+)$
  if $\nu_i^2 > n_{post-fit}^2 R_{k,ii}$ **then**
    reject y_i **and flag** i as potential cycle-slip (phase)
    goto start of EKF measurement update
  end
end



$n_{pre-fit}$: Threshold of pre-fit residual test (sigma)
$n_{post-fit}$: Threshold of prost-fit residual test (sigma)

PPP by EKF (1)

States and Measurements

$$\mathbf{x} = (\mathbf{r}_r^T, cdt_r, Z_t, B_{r,LC}^{s1}, B_{r,LC}^{s2}, ..., B_{r,LC}^{sm})^T$$

$$\mathbf{y}_k = (\mathbf{y}_P{}^T, \mathbf{y}_\Phi{}^T)^T \quad (t = t_k)$$

$$\mathbf{y}_p = (P_{r,LC}^{s1}, P_{r,LC}^{s2}, ..., P_{r,LC}^{sm})^T \quad (P_{r,LC}^{si} = C_1 P_{r,L1}^{si} + C_2 P_{r,L2}^{si})$$

$$\mathbf{y}_\Phi = (\Phi_{r,LC}^{s1}, \Phi_{r,LC}^{s2}, ..., \Phi_{r,LC}^{sm})^T \quad (\Phi_{r,LC}^{si} = C_1 \lambda_1 \varphi_{r,L1}^{si} + C_2 \lambda_2 \varphi_{r,L2}^{si})$$

(Iono-free code and phase)

Measurement Equations and Partial Derivatives

$$\mathbf{h}_k(\mathbf{x}) = (\mathbf{h}_p(\mathbf{x})^T, \mathbf{h}_\Phi(\mathbf{x})^T)^T \quad \mathbf{H}_k = (\mathbf{H}_p{}^T, \mathbf{H}_\Phi{}^T)^T \quad (t = t_k)$$

$$\mathbf{h}_P(\mathbf{x}) = \begin{pmatrix} \rho_r^{s1} + c(dt_r - dT^{s1}(t^{s1})) + T_r^{s1} + d_r^{s1} \\ \rho_r^{s2} + c(dt_r - dT^{s2}(t^{s2})) + T_r^{s2} + d_r^{s2} \\ \vdots \\ \rho_r^{sm} + c(dt_r - dT^{sm}(t^{sm})) + T_r^{sm} + d_r^{sm} \end{pmatrix} \quad \mathbf{h}_\Phi(\mathbf{x}) = \begin{pmatrix} \rho_r^{s1} + c(dt_r - dT^{s1}(t^{s1})) + T_r^{s1} + d_r^{s1} + \lambda_{NL} d_{pw}^{s1} + B_r^{s1} \\ \rho_r^{s2} + c(dt_r - dT^{s2}(t^{s2})) + T_r^{s2} + d_r^{s2} + \lambda_{NL} d_{pw}^{s2} + B_r^{s2} \\ \vdots \\ \rho_r^{sm} + c(dt_r - dT^{sm}(t^{sm})) + T_r^{sm} + d_r^{sm} + \lambda_{NL} d_{pw}^{sm} + B_r^{sm} \end{pmatrix}$$

$$\mathbf{H}_P = \begin{pmatrix} -\mathbf{e}_r^{s1T} & 1 & m_w(El_r^{s1}) & \mathbf{0}_{1\times m} \\ -\mathbf{e}_r^{s2T} & 1 & m_w(El_r^{s2}) & \mathbf{0}_{1\times m} \\ \vdots & \vdots & \vdots & \vdots \\ -\mathbf{e}_r^{smT} & 1 & m_w(El_r^{sm}) & \mathbf{0}_{1\times m} \end{pmatrix} \quad \mathbf{H}_\Phi = \begin{pmatrix} -\mathbf{e}_r^{s1T} & 1 & m_w(El_r^{s1}) & 1 & 0 & \cdots & 0 \\ -\mathbf{e}_r^{s2T} & 1 & m_w(El_r^{s2}) & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mathbf{e}_r^{smT} & 1 & m_w(El_r^{sm}) & 0 & 0 & \cdots & 1 \end{pmatrix}$$

PPP by EKF (2)

Initial States and Covariances

$$\hat{\mathbf{x}}_1^- = (\hat{\mathbf{r}}_r{}^T, c\hat{d}t_r, Z(\hat{\mathbf{r}}_r), B_0^{s1}, B_0^{s2}, ..., B_0^{sm})^T$$

$$\mathbf{P}_1^- = \mathrm{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_t^2, \sigma_Z^2, \sigma_B^2, \sigma_B^2, ..., \sigma_B^2)^T$$

$\hat{\mathbf{r}}_r, c\hat{d}t_r, \sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_t^2$: SPP solutions

$Z(\hat{\mathbf{r}}_r)$: ZTD by empirical tropos. model (m)

$B_0^{si} = \Phi_{r,LC}^{si} - P_{r,LC}^{si}$: Guess phase bias (m)

σ_Z^2, σ_B^2 : Variance of $Z(\hat{\mathbf{r}}_r), B_o^{si}$

Time Update of EKF

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_k \hat{\mathbf{x}}_{k-1}^+ ,\ \mathbf{P}_k^- = \mathbf{\Phi}_k \mathbf{P}_{k-1}^+ \mathbf{\Phi}_k^T + \mathbf{Q}_k$$

$$\mathbf{\Phi}_k = \begin{pmatrix} \mathbf{I}_{3\times3} & & & \\ & 1 & & \\ & & 1 & \\ & & & \mathbf{I}_{m\times m} \end{pmatrix} \quad \mathbf{Q}_k = \begin{pmatrix} \sigma_{q_r}^2 \mathbf{I}_{3\times3} & & & \\ & \sigma_{q_t}^2 & & \\ & & \sigma_{q_Z}^2 & \\ & & & \sigma_{q_B}^2 \mathbf{I}_{m\times m} \end{pmatrix} (t_k - t_{k-1})$$

$\sigma_{q_r}^2, \sigma_{q_t}^2, \sigma_{q_Z}^2 \sigma_{q_B}^2$: System noise variance of receiver position, receiver clock, ZTD and phase bias (m^2/s)

Typical System Noises

$$\sigma_{q_r}^2 = \begin{cases} 0 & \text{(Static PPP)} \\ \infty & \text{(Kinematic PPP)} \end{cases} \quad \sigma_{q_t}^2 = \infty \quad \sigma_{q_Z}^2 = \left(10^{-4}\right)^2 \quad \sigma_{q_B}^2 = \begin{cases} \left(10^{-4}\right)^2 & \text{(cycle slip)} \\ \infty & \text{(no cycle slip)} \end{cases}$$

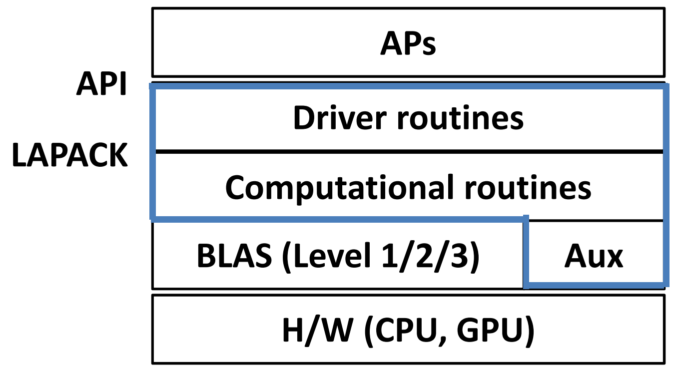
(stochastic models: receiver clock = white-noise, ZTD = random-walk)

LAPACK/BLAS

LAPACK (Linear Algebra Package)

Linear equations
Linear least squares (LLS) problems
Eigenproblems
Singular value decomposition (SVD)
...

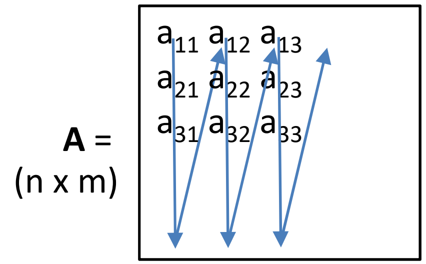
xyyyzzz() **x** = (S: SP, D: DP C: SP-CPX, Z: DP-CPX)
yy = matrix type, **zzz** = function



BLAS (Basic Linear Algebra Subprograms)

Level 1 $\mathbf{y} \leftarrow \alpha\mathbf{x} + \mathbf{y}$
Level 2 $\mathbf{y} \leftarrow \alpha\mathbf{A}^{\gamma}\mathbf{x} + \beta\mathbf{y}$
Level 3 $\mathbf{C} \leftarrow \alpha\mathbf{A}^{\gamma}\mathbf{B}^{\delta} + \beta\mathbf{C}$

xzzzz() **x** = (S: SP, D: DP C: SP-CPX, Z: DP-CPX)
zzzz = function



column-major order in memory
(a_{ij} = A[i+n*j]=A[j][i])

Many Compatible Libraries to LAPACK/BLAS

ATLAS, GotoBLAS, OpenBLAS, Intel MKL, cuBLAS, ...

(http://netlib.org/lapack, http://netlib.org/blas)

Performance of Matrix Handling

Function	Size (n =)	CPU Time				
		RTKLIB	LAPACK/BLAS	ATLAS	OpenBLAS *	Intel MKL *
matmul() C = A * B (A,B,C: n x n)	100	0.002 s (1.00 Gflops)	0.002 s (1.00 Gflops)	0.001 s (1.99 Gflops)	0.019 s (0.10 Gflops)	0.013 s (0.15 Gflops)
	300	0.056 s (0.96 Gflops)	0.057 s (0.95 Gflops)	0.012 s (4.49 Gflops)	0.003 s (17.97 Gflops)	0.001 s (53.91 Gflops)
	1000	1.527 s (1.31 Gflops)	0.660 s (3.03 Gflops)	0.200 s (10.00 Gflops)	0.033 s (60.58 Gflops)	0.032 s (62.47 Gflops)
	3000	287.319 s (0.19 Gflops)	22.178 s (2.43 Gflops)	4.031 s (13.40 Gflops)	0.262 s (206.07 Gflops)	0.513 s (105.25 Gflops)
	10000	> 1800 s	815.128 s (2.45 Gflops)	148.005 s (13.51 Gflops)	5.674 s (352.47 Gflops)	10.716 s (186.63 Gflops)
solve() A * x = y (A: n x n, x,y: n x 1)	100	0.005 s	0.004 s	0.000 s	0.001 s	0.001 s
	300	0.084 s	0.056 s	0.007 s	0.002 s	0.016 s
	1000	2.389 s	0.939 s	0.224 s	0.025 s	0.054 s
	3000	493.587 s	28.674 s	5.649 s	0.389 s	0.697 s
	10000	> 1800 s	1010.426 s	202.090 s	9.342 s	12.122 s

CPU: Core i7-5960X (8C/16T, 3.0-3.5 GHz, HT=ON, peak 448 Gflops), RAM: 32 GB,
Ubuntu 16.04 LTS (64 bit), gcc 5.4.0, RTKLIB 2.4.2, MKL 11.1.2.244, * multi-threaded using all cores

matmul() and solve()

DGEMM : $C \leftarrow \alpha \mathrm{op}(A)\mathrm{op}(B) + \beta C,\ \mathrm{op}(X) = X, X^T, X^H,\ C - m \times n$

DGETRF : Compute an LU factorization if a general matrix, using partial pivoting with row interchange

DGETRS : Solve a general system of linear equations AX = B, A^TX = B or A^HX = B, using the LU factorization computed by DGETRF

matmul()

```
extern void matmul(const char *tr, int n, int k, int m, double alpha,
                   const double *A, const double *B, double beta, double *C)
{
    int lda=tr[0]=='T'?m:n,ldb=tr[1]=='T'?k:m;

    dgemm_((char *)tr,(char *)tr+1,&n,&k,&m,&alpha,(double *)A,&lda,(double *)B,
           &ldb,&beta,C,&n);
}
```

solve()

```
extern int solve(const char *tr, const double *A, const double *Y, int n,
                 int m, double *X)
{
    double *B=mat(n,n);
    int info,*ipiv=imat(n,1);

    matcpy(B,A,n,n);
    matcpy(X,Y,n,m);
    dgetrf_(&n,&n,B,&n,ipiv,&info);
    if (!info) dgetrs_((char *)tr,&n,&m,B,&n,ipiv,X,&n,&info);
    free(ipiv); free(B);
    return info;
}
```

RTKLIB
2.4.2
rtkcmn.c

Extension of PPP Algorithms

Support Multi-Constellation GNSS
- More satellites improve performance: GPS, GLONASS, Galileo, QZSS, BDS, ...
- Consider time and coordinate system difference
- Manage receiver biases (ISB and IFB) between systems

No Explicit LC to Eliminate Ionosphere Effects
- With less measurement noise
- Reduce solution re-convergence time after loss-of-lock of signals
- Support local STEC correction generation

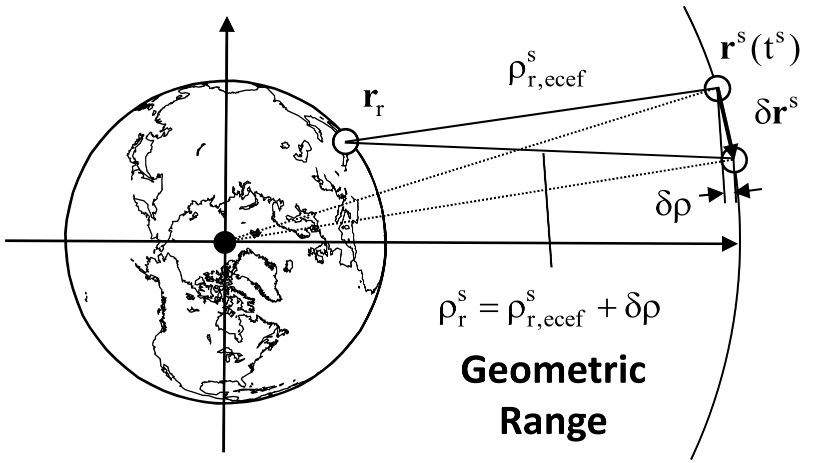
Involve Adaptive Filter Algorithm
- Enable auto tuning of stochastic parameters
- Improve performance in dynamic environments

Reduce CPU Time
- Employ sparse matrix handling
- Dynamic assignment of phase bias parameters to reduce matrix size

PPP-AR and INS Integration
- Refer advanced topics

Notes
for PPP Models

Sagnac Effect in Geometric Range



$$\rho^s_{r,ecef} = \left\| \mathbf{r}^s(t^s) - \mathbf{r}_r \right\|$$

$$\delta\mathbf{r}^s \approx \left\{ (0,0,-\omega_e)^T \times \mathbf{r}^s(t^s) \right\} (t_r - t^s)$$

$$\approx \left\{ (0,0,-\omega_e)^T \times \mathbf{r}^s(t^s) \right\} \frac{\rho^s_{r,ecef}}{c}$$

$$\delta\rho \approx \left(\frac{\mathbf{r}^s(t^s) - \mathbf{r}_r}{\rho^s_{r,ecef}} \right)^T \delta\mathbf{r}^s$$

$$= \frac{1}{c} \begin{pmatrix} x^s - x_r \\ y^s - y_r \\ z^s - z_r \end{pmatrix}^T \begin{pmatrix} \omega_e y^s \\ -\omega_e x^s \\ 0 \end{pmatrix}$$

$$= \frac{(x^s - x_r)\omega_e y^s - (y^s - y_r)\omega_e x^s}{c}$$

$$= \frac{\omega_e (x^s y_r - y^s x_r)}{c}$$

$$\mathbf{r}_r = (x_r, y_r, z_r)^T, \; \mathbf{r}^s(t^s) = (x^s, y^s, z^s)^T$$

Relativistic Correction for Satellite Clock

J. Kouba, A guide to using International GNSS Service (IGS) products, 2009

The GPS System already has some well developed modeling conventions, e.g., that only the periodic relativity correction

(5.3.4)

$$\Delta t_{rel} = -2\vec{X}s \cdot \vec{V}s / c^2 \qquad (27)$$

is to be applied by all GPS users (ION, 1980; ICD-GPS-200, 1991). Here $\vec{X}s$, $\vec{V}s$ are the satellite position and velocity vectors and c is the speed of light. The same convention has also been adopted by IGS, i.e., all the IGS satellite clock solutions are consistent with and require this correction. Approximation errors of

IERS Conventions 2010

In this framework, the proper time of a clock A located at the GCRS coordinate position $\mathbf{x}_A(t)$, and moving with the coordinate velocity $\mathbf{v}_A = d\mathbf{x}_A/dt$, where t is TCG,

An alternative expression for the relativistic periodic correction is

(10.2)

$$\Delta\tau_A^{per} = -\frac{2}{c^2}\mathbf{v}_A \cdot \mathbf{x}_A, \qquad (10.11)$$

which is exactly equivalent to the preceding Keplerian orbit formulation, provided that the osculating Keplerian orbit elements are used. This formulation is used *e.g.* by the IGS (International GNSS Service) for its official GPS and GLONASS clock solution products.

GCRS: Geocentric Celestial Reference System,
TCG : Geocentric Coordinate Time

IS-GPS-200E, GPS Space segment/navigation user interface specifications

$$\Delta t_r = -\frac{2\vec{R} \bullet \vec{V}}{c^2}$$

where

(20.3.3.3.3.1)

$\vec{R}$ is the instantaneous position vector of the SV,

$\vec{V}$ is the instantaneous velocity vector of the SV, and

c is the speed of light. (Reference paragraph 20.3.4.3).

It is immaterial whether the vectors $\vec{R}$ and $\vec{V}$ are expressed in earth-fixed, rotating coordinates or in earth-centered, inertial coordinates.