

Geophysical Data Analysis Quiz #2

05-242628 三田村彰大

November 28, 2025

(1)

支配方程式が線形性を持つことを示せば良い*1。今、歪みテンソル ϵ_{ij} 、応力テンソル σ_{ij} 、変位 u_i に対し、等方弾性体を考えれば、

$$\epsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (1)$$

$$\sigma_{ij} = \lambda \delta_{ij} \sum_k \epsilon_{kk} + 2\mu \epsilon_{ij} \quad (2)$$

が成立する。特に今、変位が y 方向のみに存在し、空間が y 方向に均一な場合を考える。すなわち $u_x = u_z = 0$ 、 $u_y = u(x, z)$ が成り立つとする。この時、(1) 式より

$$\epsilon_{xy} = \epsilon_{yx} = \frac{1}{2} \frac{\partial u}{\partial x}, \quad \epsilon_{yz} = \epsilon_{zy} = \frac{1}{2} \frac{\partial u}{\partial z} \quad (3)$$

となり、(2) 式に代入すれば

$$\sigma_{xy} = \mu \frac{\partial u}{\partial x}, \quad \sigma_{zy} = \mu \frac{\partial u}{\partial z}, \quad \sigma_{yy} = 0 \quad (4)$$

を得ることができる。ここで、今回の断層のモデルでは力は y 方向にしか働かないと考えて良いから、力の釣り合いは y 方向のみについて考えればよい。よって y 向きの体積力を $\rho(x, z)$ とすれば、

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{zy}}{\partial z} = \rho(x, z) \quad (5)$$

$$\rightarrow \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2} \right) = \rho(x, z) \quad (6)$$

となる。(ただし (4) 式を用いた。) (6) 式を見るとこれは u, ρ に対して明らかに線形で、 $\rho_1(x, z)$ に対する解 $u_1(x, z)$ と $\rho_2(x, z)$ に対する解 $u_2(x, z)$ を考えた時 $u_1 + u_2$ は $\rho_1 + \rho_2$ に対する解となっている。

よって今 magnitude s_1 の right-lateral faulting が深さ $d_1^1 \sim d_1^2$ にあり、magnitude s_2 の right-lateral faulting が深さ $d_2^1 \sim d_2^2$ に存在している時、変位 u はそれぞれの right-lateral faulting が個別に存在した場合の変位を重ね合わせとなる。

特に、問題で問われている「magnitude s の right-lateral faulting が深さ $d_1 \sim d_2$ にある状態」は、「magnitude

*1参考：https://topex.ucsd.edu/pub/sandwell/geodynamics_notes/14_1_strike_slip_fault_1.pdf

− s の right-lateral faulting が深さ $0 \sim d_1$ にある状態と magnitude s の right-lateral faulting が深さ $0 \sim d_2$ にある状態の重ね合わせ」とみなすことができる。よって、求める変位は

$$u(x; d_1, d_2) = -\frac{(-s)}{\pi} \tan^{-1} \frac{d_1}{x} - \frac{s}{\pi} \tan^{-1} \frac{d_2}{x} \quad (7)$$

$$\therefore u(x; d_1, d_2) = -\frac{s}{\pi} \left(\tan^{-1} \frac{d_2}{x} - \tan^{-1} \frac{d_1}{x} \right) \quad (8)$$

となる。

(2)

簡単のため、「magnitude s の right-lateral faulting が深さ $d_1 \sim d_2$ にある時の $z = 0$ での y 方向への変位」を $u(x; d_1, d_2, s)$ と書くことにする。^{*2}また、観測点 $x_i = -50, -45, \dots, 50$ km で観測された y 方向への変位を $u(x_i)$ と表すことにする。

さて今、 $0 - 1$ km, $1 - 2$ km, \dots , $19 - 20$ km と discretize した fault の各深さにおける magnitude を s_1, s_2, \dots, s_{20} と書き記すことにすると、データ \mathbf{d} 及びモデルパラメタ \mathbf{m} は $1 \leq i \leq 20$ に対して

$$d_i = u(x_i) \quad (10)$$

$$m_i = s_i \quad (11)$$

である。また、(1) で示した線形性から、discretize した fault による表層での変位は線形重ね合わせで求めることができる。よって深さを $z_j = 0, 1, 2, \dots, 20$ km で表して先に導入した表記を用いれば、

$$u(x_i) = u(x_i; z_0, z_1, s_1) + u(x_i; z_1, z_2, s_2) + \dots + u(x_i; z_{19}, z_{20}, s_{20}) = \sum_{j=1}^{20} u(x_i; z_{j-1}, z_j, s_j) \quad (12)$$

となる。そこで、 $1 \leq i \leq 20, 1 \leq j \leq 20$ に対しデータカーネル G を

$$G_{ij} = \frac{1}{\pi} \left(\tan^{-1} \frac{z_{j-1}}{x_i} - \tan^{-1} \frac{z_j}{x_i} \right) = \frac{1}{s_j} u(x_i; z_{j-1}, z_j, s_j) \quad (13)$$

のようにとれば

$$d_i = u(x_i) = \sum_{j=1}^{20} u(x_i; z_{j-1}, z_j, s_j) = \sum_{j=1}^{20} \frac{1}{s_j} u(x_i; z_{j-1}, z_j, s_j) \cdot s_j = \sum_{j=1}^{20} G_{ij} m_j \quad (14)$$

となるので、 G は観測方程式 $\mathbf{d} = G\mathbf{m}$ を満たすことができる。今、離散化した深さ z_j と観測位置 x_i は既知であるから G_{ij} は既知であり、観測値 d_i もまた既知の値であるのでこれらを用いてモデルパラメタ \mathbf{m} を推定することができる。

^{*2} (8) 式の s をパラメタとして明示的に書き記しただけである。すなわち、

$$u(x; d_1, d_2, s) = -\frac{s}{\pi} \left(\tan^{-1} \frac{d_2}{x} - \tan^{-1} \frac{d_1}{x} \right) \quad (9)$$

a priori 情報を用いない場合：

手始めに、*a priori* 情報を用いずに m を求めることを考えてみる。これは授業で扱った通りで、

$$\hat{\mathbf{m}} = G^{-g} \mathbf{d} = (G^T G)^{-1} G^T \mathbf{d} \quad (15)$$

のように求めることができる。よって、実際に \mathbf{m} を求めて $s_i = m_i$ を図示すると以下のよう。

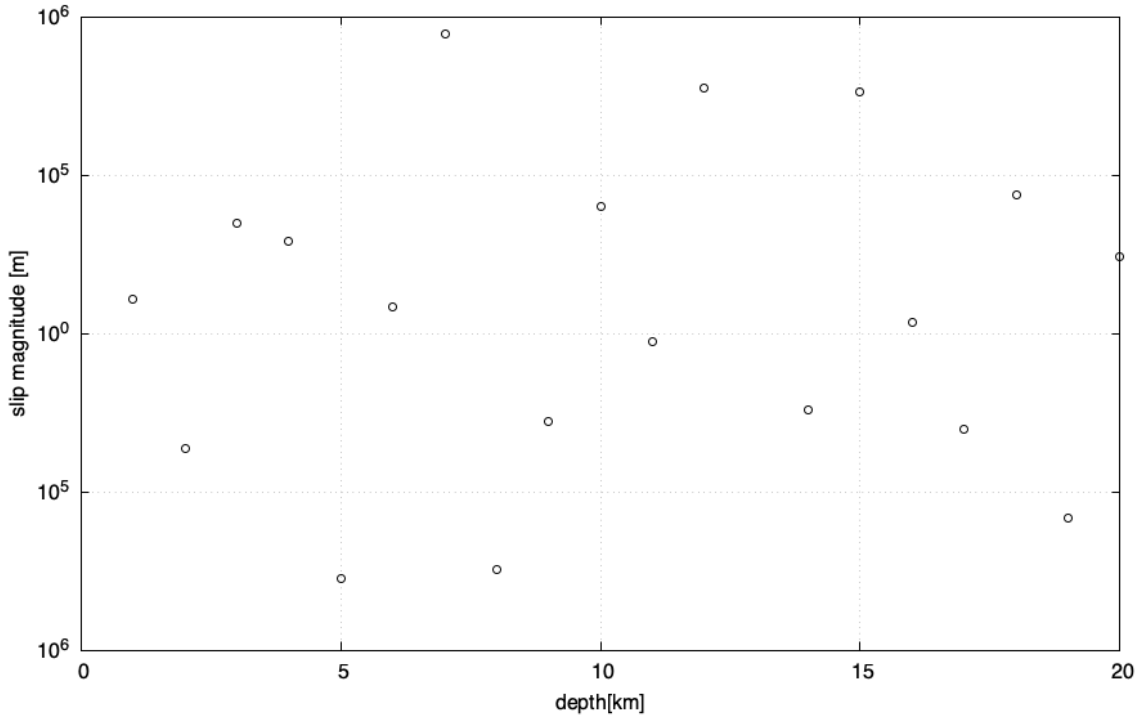


Figure1: Slip estimation with no *a priori* information. Be aware that the y axis has a very wide range, showing the numerical instability of this estimation.

上図は深さごとの slip magnitude s_i をプロットしたものであるが、 $s_i = m_i$ の値が非常に暴れていることがわかる。これは、*a priori* information を入れなかったことによって残差 $E = \|\mathbf{d} - G\mathbf{m}\|$ を小さくすることだけが優先された結果、滑り分布の連続性を考慮できなかったためである。よって、滑り分布に連続性を課す *a priori* information を導入する必要がある。

a priori information (1): continuous slip condition

先の考察に従い、滑り分布に連続性を課す *a priori* information、すなわち *continuous slip condition* を課すことにする。これは、上下の層で slip magnitude の差を小さくすることで実現されるので、 $s_{i+1} - s_i$ の絶対値を小さくするような条件を課せば良い。よって、 $m_i = s_i$ であるから、 \mathbf{m} に対し

$$\left[L_{\text{CSC}} = \sum_{i=1}^{19} (m_{i+1} - m_i)^2 \right] \text{ is small} \quad (16)$$

という *a priori* information を課すことにする。^{*3}

ここで、サイズ 19×20 の差分行列

$$D = \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{pmatrix} \quad (17)$$

を定義すると、(16) 式の L は $L = (D\mathbf{m})^T D\mathbf{m}$ と書き換えることができる。よって、 L is small という条件のもとで \mathbf{m} を推定するということは、damping parameter を λ^2 として、

$$E = (\mathbf{d} - G\mathbf{m})^T (\mathbf{d} - G\mathbf{m}) + \lambda^2 (D\mathbf{m})^T D\mathbf{m} \quad (18)$$

を最小にするような \mathbf{m} を求めるという問題に帰着される。よって、講義で与えられた通りに、

$$\mathbf{m} = (G^T G + \lambda^2 D^T D)^{-1} G^T \mathbf{d} \quad (19)$$

で \mathbf{m} を求めることができる。

さて、damping parameter λ^2 の決定方法は様々にあるが、ここでは Akaike's Bayesian Information Criterion (ABIC) を用いることにする^{*4}。今、適当に決定した λ^2 に対し (19) 式で \mathbf{m} を求めることができれば、以下の式によって λ^2 の関数としての ABIC が求まる。

$$\begin{aligned} \text{ABIC}(\lambda^2) &= (N - M + r) \log \left((\mathbf{d} - G\mathbf{m})^T (\mathbf{d} - G\mathbf{m}) + \lambda^2 (D\mathbf{m})^T D\mathbf{m} \right) \\ &\quad - r \log \lambda^2 + \log |G^T G + \lambda^2 D^T D| + \text{const.} \end{aligned} \quad (20)$$

よって、 λ^2 に関する *iteration* を回しながら、

$$\text{「(19) 式で } \mathbf{m} \text{ を求める」} \rightarrow \text{「(20) 式を用いて } \text{ABIC}(\lambda^2) \text{ を求める」} \quad (21)$$

という計算を行い、最小の $\text{ABIC}(\lambda^2)$ を与えた λ^2 に対して、 \mathbf{m} を推定値として出力するというを行う。^{*5}これにより求めた結果が下図である。

^{*3}CSC は continuous slip condition の略

^{*4}なんだかんだ言って式一本で求まる ABIC が一番やりやすいのではないかと、というのが ABIC を選んだ正直な理由である。

^{*5}特に深い意味はないが、せっかくなので FORTRAN のコードをレポートの最後に書き加えておく。

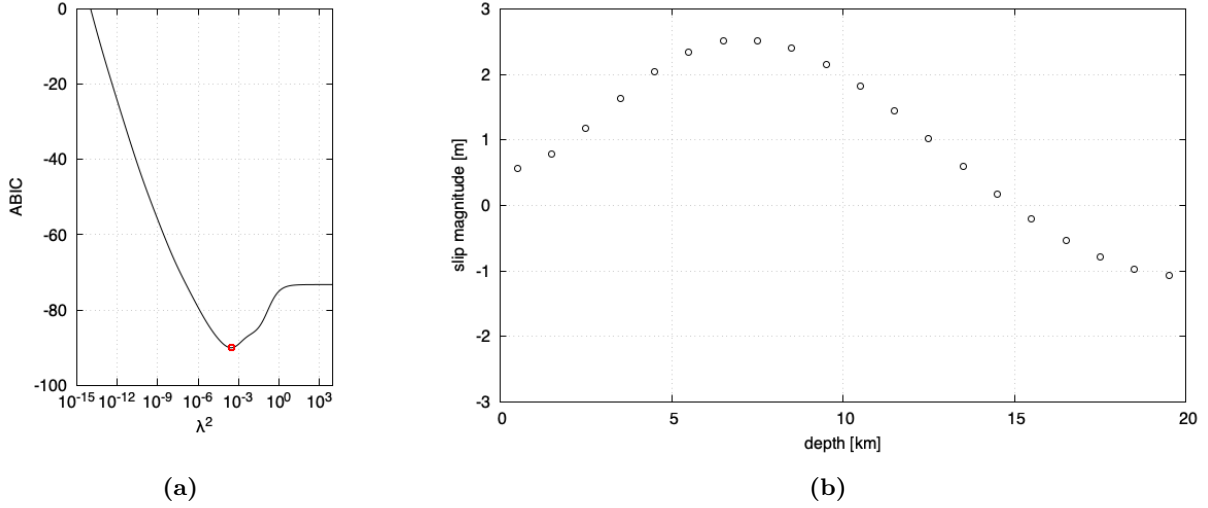


Figure2: Slip estimation using *continuous slip condition* as a *a priori* information. (a): ABIC as function of the damping parameter. (b): Estimation of slip magnitude s [m]

Figure2は、ABIC の結果と得られた滑り分布 $s_i = m_i$ である。Figure2aの赤丸が ABIC を最小にする λ^2 であり、その時の式 (19) で求まる \mathbf{m} が Figure2bである。この図を見ると、5 – 10 km の間に滑りのピークがあることがわかる。また、*continuous slip condition* は満たされているが、20 km の深さで slip magnitude s は負の値を持ってしまっており、問題文中の”no fault slips occur below 20 km” という条件と矛盾する。そこで次は、深部で slip がなくなるといふ *a priori* information を導入する必要がある。

a priori information (2): boundary no-slip condition

考察に従い、”no fault slips occur below 20 km” という *a priori* information を追加する。今回はこれを「深度 20 km での滑り s_{20} の絶対値は小さくしなければならない」といふ *boundary no-slip condition* によって実現させることにする。すなわち今、

$$\left[L_{BNSC} = (m_{20})^2 \right] \text{ is small} \quad (22)$$

という *a priori* information が追加で課されることとなる。^{*6} よって、条件(16)と組み合わせれば *a priori* information は次のようになる。

$$\left[(L_{CSC} + L_{BNSC}) = \sum_{i=1}^{19} (m_{i+1} - m_i)^2 + (m_{20})^2 \right] \text{ is small} \quad (23)$$

ここで、先ほど用いた差分行列 D の下に m_{20} に効く行を合体させるということをする。すなわち

$$D_{\text{new}} = \begin{bmatrix} \boxed{D} \\ 0 \cdots 0 \ 1 \end{bmatrix} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & 1 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \quad (24)$$

^{*6}BNSC は boundary no-slip condition の略

として D を書き直す。すると、 $(L_{CSC} + L_{BNSC}) = (D\mathbf{m})^T D\mathbf{m}$ となるから、これが小さいという条件の下で \mathbf{m} を推定するという事は、damping parameter を λ^2 として、

$$E = (\mathbf{d} - G\mathbf{m})^T (\mathbf{d} - G\mathbf{m}) + \lambda^2 (D\mathbf{m})^T D\mathbf{m} \quad (25)$$

を最小にするような \mathbf{m} を求めるという問題に帰着される。よって先ほどと全く同じ問題に帰着し、先と同様の手続きによって (すなわち ABIC を用いて) \mathbf{m} を推定することができる。

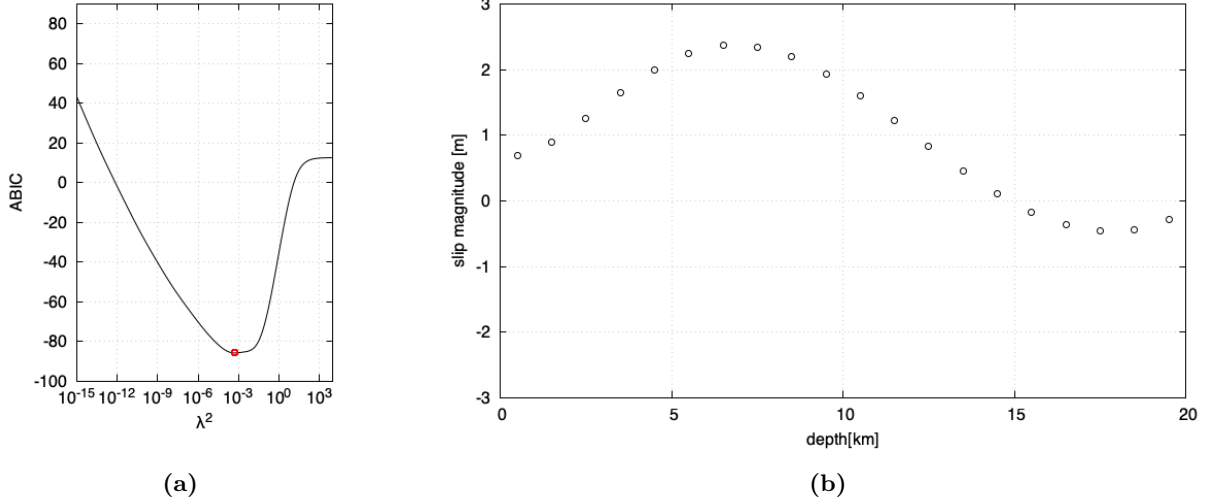


Figure3: Slip estimation using *continuous slip condition* and *boundary no-slip condition* as *a priori* information. *Boundary no-slip condition* here constrains slip magnitude to 0 m at depth of 20 km. (a): ABIC as function of the damping parameter. (b): Estimation of slip magnitude s [m]

Figure3は、*continuous slip condition* と *boundary no-slip condition* の下で ABIC を用いて \mathbf{m} を推定した結果である。Figure3bを見ると、滑りは連続性を持っており、尚且つ深度 20 km で滑りが無くなるという条件を満たしている。しかし、18 – 20 km の滑り分布には不自然な感じがある。まるで、「 $s_{20} = 0$ となるように無理やり合わせた」かのように、深度 18 km 付近で slip magnitude は急に上昇する。よって、次はこの問題を解決するために *a priori* information に若干の修正を加えることにする。

a priori information (3): weighted no-slip condition

先ほど述べた不自然な滑り分布の問題を解決するためには、 L_{BNSC} を修正しなければならない。すなわち、*no-slip condition* を s_{20} のみに与えたから問題になったのであり、深くなるに従ってだんだんと *no-slip condition* が効いてくるような *a priori* information を与えなければならない。そこで、次のような *a priori* information を与えることにする。

$$\left[L_{WNSC} = \sum_{i=1}^{20} (w_i m_i)^2 \right] \text{ is small} \quad (26)$$

w_i は重みであり、*no-slip condition* の深さごとの効きの強さを決めるパラメタである。今回は簡単のため、*boundary no-slip* の場合で滑りが負となった 15 km 以降の深さのみに *no-slip condition* がかかるようにする。すなわち、 $i \leq 14$ では $w_i = 0$ である。また、 $i \geq 15$ については深部ほど *no-slip condition* が強く効くように線形で重み付けしていき、

$w_{14} = 0$ と $w_{20} = 1$ を満たすように取る。すなわち、 w_i は以下のように取る。

$$w_i = \begin{cases} 0 & (i \leq 14) \\ \frac{i - 14}{20 - 14} & (i \geq 15) \end{cases} \quad (27)$$

また、条件 (26) と条件 (16) とを合わせれば、*a priori* information は以下のようになる。

$$\left[(L_{CSC} + L_{WNSC}) = \sum_{i=1}^{19} (m_{i+1} - m_i)^2 + \sum_{i=1}^{20} (w_i m_i)^2 \right] \text{ is small} \quad (28)$$

よって今回は、行列 D の変形は次のように行えば良いことがわかる。^{*7}

$$D_{\text{new}} = \begin{bmatrix} \boxed{D} \\ \hline w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{20} \end{bmatrix} \quad (29)$$

実際、このように取ると $(L_{CSC} + L_{WNSC}) = (D\mathbf{m})^T D\mathbf{m}$ となっていることがすぐにわかる。よってやはり、問題は

$$E = (\mathbf{d} - G\mathbf{m})^T (\mathbf{d} - G\mathbf{m}) + \lambda^2 (D\mathbf{m})^T D\mathbf{m} \quad (30)$$

を最小にする \mathbf{m} を求めるという問題に帰着され、(21) で示した手続きにより \mathbf{m} を推定することができる。

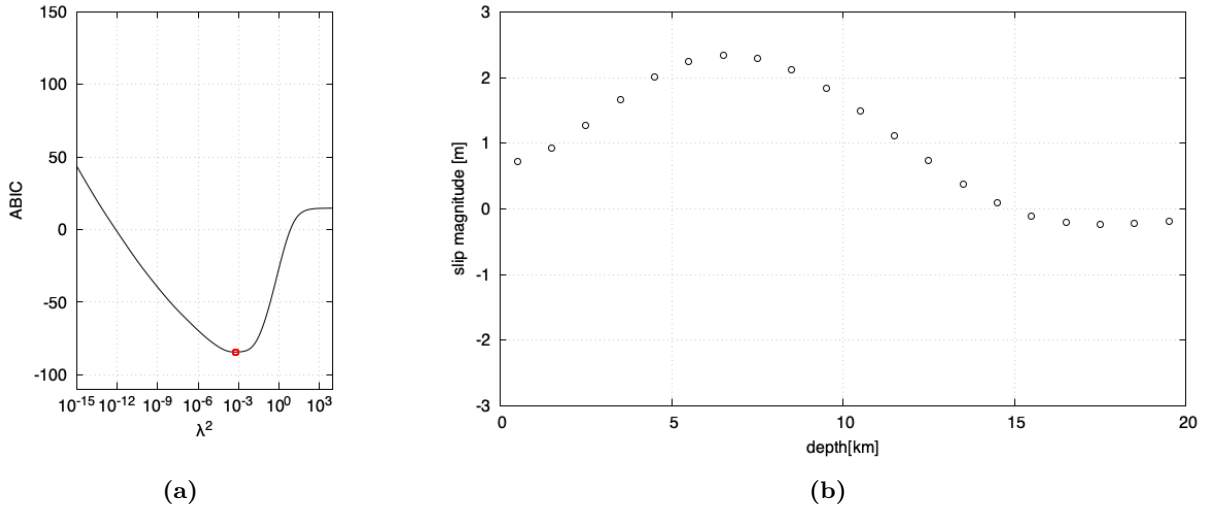


Figure4: Slip estimation using *continuous slip condition* and *weighted no-slip condition* as *a priori* information. No-slip constraint gets stronger as depth deepens, starting from 15 km and reaching its maximum at 20 km. (a): ABIC as function of the damping parameter. (b): Estimation of slip magnitude s [m]

Figure4は *weighted no-slip condition* を用いた場合の slip estimation である。 *boundary no-slip condition* を用

^{*7} D は式 (17) のものを参照。

いた場合に比べると、深さ 15 km 以下でのマイナス方向へのモデルパラメタの振り切れは緩和されている。^{*8}

a priori information (4): *smooth slip condition*

最後に、*smooth slip condition* という *a priori* information を与えることを考える。すなわち、これまでは滑りが連続だという条件 (*continuous slip condition*) を課してきたが、そこへさらに、滑りは滑らかに変化するという *a priori* information を課するのである。これは、物理的には非常に自然な考えに思われる。しかし、特定の深度にだけ鋭い滑りが見られるといった状況を排除してしまうことにも繋がりがねない。^{*9} よって、今回はあくまで実験的にこの *a priori* information を導入する。

滑らかであるという主張は、差分の差分が小さくなることと同義であるから、*a priori* information は

$$\left[L_{SSC} = \sum_{i=1}^{18} \{(m_i - m_{i+1}) - (m_{i+1} - m_{i+2})\}^2 \right] \text{ is small} \quad (31)$$

である。よって、 D を次のような 18×20 の二階差分行列に取る。

$$D = \begin{pmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -2 & 1 \end{pmatrix} \quad (32)$$

すると、 $L_{SSC} = (D\mathbf{m})^T D\mathbf{m}$ と書き換えることができる。ここに、*weighted no-slip condition* を付け加えたいので、*a priori* information をあらためて

$$\left[(L_{SSC} + L_{WNSC}) = \sum_{i=1}^{18} \{(m_i - m_{i+1}) - (m_{i+1} - m_{i+2})\}^2 + \sum_{i=1}^{20} (w_i m_i)^2 \right] \text{ is small} \quad (33)$$

と書き直す。この時、おなじように D も

$$D_{\text{new}} = \begin{bmatrix} \boxed{D} \\ \hline w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{20} \end{bmatrix} \quad (34)$$

のように書き直す。するとやはり、 $(L_{SSC} + L_{WNSC}) = (D\mathbf{m})^T D\mathbf{m}$ となっている。よってやはり、問題は

$$E = (\mathbf{d} - G\mathbf{m})^T (\mathbf{d} - G\mathbf{m}) + \lambda^2 (D\mathbf{m})^T D\mathbf{m} \quad (35)$$

を最小にする \mathbf{m} を求めるという問題に帰着され、(21) で示した手続きにより \mathbf{m} を推定することができる。下の Figure5 はこれにより得られる slip magnitude = \mathbf{m} の解である。

^{*8}詳しい比較は Figure6 を参照。

^{*9}地震学においてはアスペリティーと言った概念があると聞き齧っているので、正直この *a priori* information を導入するかどうかはひどく迷った。が、比較のため *smooth slip* も *continuous slip* も両方やってみることにする。

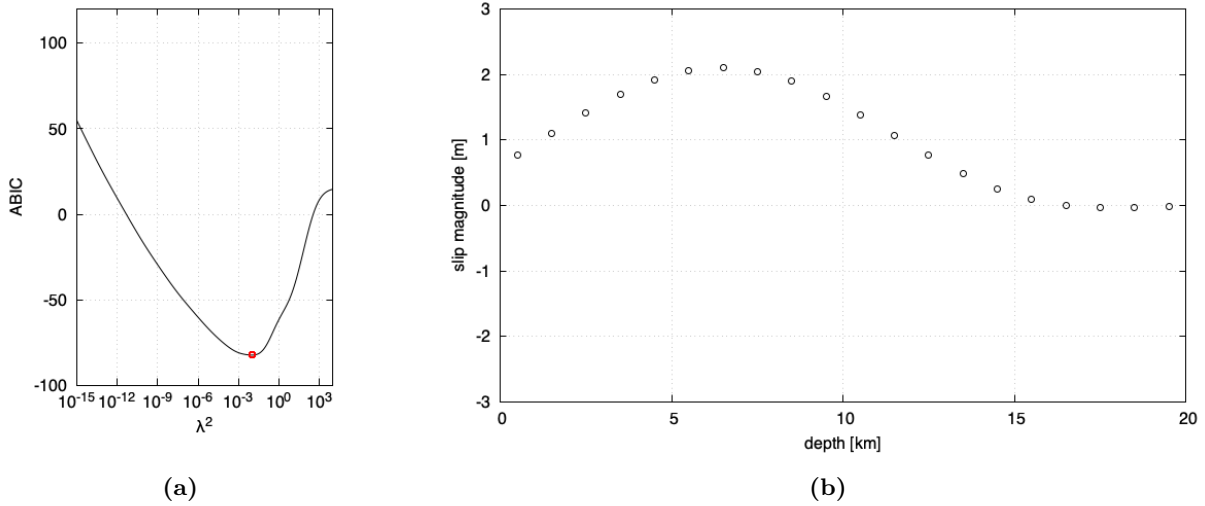


Figure5: Slip estimation using *smooth slip condition* and *weighted no-slip condition* as a *priori* information. No-slip constraint gets stronger as depth deepens, starting from 15 km and reaching its maximum at 20 km. (a): ABIC as function of the damping parameter. (b): Estimation of slip magnitude s [m]

解の比較

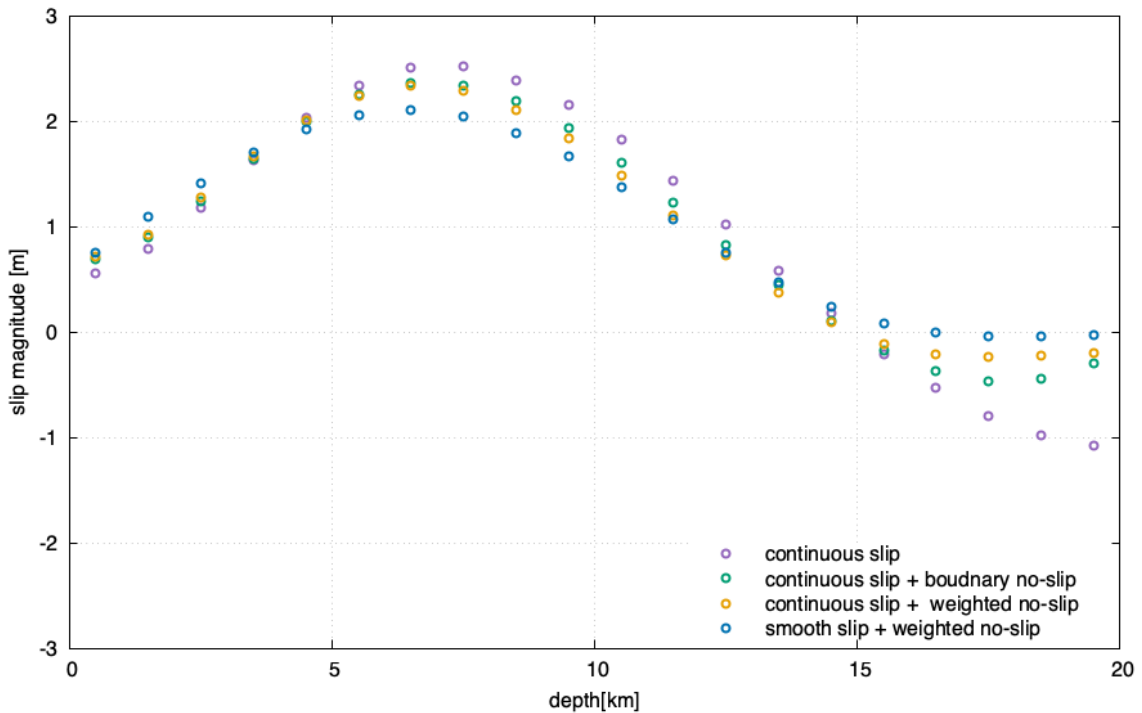


Figure6: Comparison of slip estimation under various *a priori* information. The correspondence is as follows; Purple points: data from Figure2. Green points: data from Figure3. Orange points: data from Figure4. Blue points: data from Figure5.

さて、ここまで様々な形の *a priori* information を与え \mathbf{m} を推定してきたが、ここでこれらの方法を比較することを考える。今、Figure6はここまで行ってきた推定により得られた \mathbf{m} を同一平面に並べたものである。

紫の点は *continuous slip condition* のみを与えたときの解であり、*no-slip condition* を与えて深部での滑りを取

束させた他の解と比べると明らかに 20 km 時点での 0 への収束性が悪いことがわかる。また 0 への収束という観点で言うと、*weighted no-slip condition* は *boundary no-slip condition* よりもスムーズに滑りが 0 へ収束している。*boundary no-slip condition* では s_{20} だけに slip magnitude が 0 になると言う拘束を与えたので、収束が鈍くなるのは当然のことである。また、今回は *weighted no-slip condition* における重み w_i をいささか恣意的に選んだことは否めない。^{*10} その上で、黄色の *boundary no-slip* と緑色の *weighted no-slip* を見比べてみると、深さ 14km 以上では両者はよく一致している。これは、「14km 以下で重みを 0 としたことによって深部境界条件が浅い部分の解の挙動の邪魔をしていない」と言う良好な結果と捉えることができる。

最後に、青の *smooth slip condition* について述べる。これは二階差分に関する *a priori information* であるが、今青のプロットを見ると全体的に非常になだらかになっていることがわかる。特に、深部で滑りが 0 へ収束する際の挙動が他の解に比べて非常に安定している（マイナスに触れていない・急な傾きの変化が起きていない）。これは狙い通りの挙動であり、今回用いた *a priori information* の定式化によって実際に解に滑らかさを要求できていることがわかる。ただし、滑りのピーク付近 (5 – 10km) の挙動を見ると、他の *a priori information* を用いた場合に比べてピークが潰れていることがわかる。これは、二階差分に関する *a priori information* が曲率を小さくさせる方向に働くからであると考えられる。ピークの形は実際の地層構造の分析において非常に重要な問題であると予想でき、*a priori information* をどのようにとるかまでここまで明瞭に変わってしまう以上、その選び方には相当な慎重さが要求されると思われる。^{*11}

今回、問 (2) の回答としては Figure4 と Figure5 の両方を採用することにする。Figure6 で言う青と黄色のプロットである。二つの採用となったのは、*continuous* (一階差分) か *smooth* (二階差分) のどちらが *a priori information* としてふさわしいのか、自分では判断ができなためである。

(3)

授業と同様の Bayesian 的なものの見方を用いるのが良いと思われる。今、観測で得られた surface offset: $s_1 = 1$ を標準偏差 $\sigma_1 = 0.02$ の観測データとして \mathbf{d} に加えれば、求めるべき \mathbf{m} は $p(\mathbf{m}|\mathbf{d}, s_1)$ を最大にする \mathbf{m} である。特に今、ベイズの定理より

$$p(\mathbf{m}|\mathbf{d}, s_1) = \frac{p(\mathbf{d}, s_1|\mathbf{m})p(\mathbf{m})}{p(\mathbf{d}, s_1)} \propto p(\mathbf{d}|\mathbf{m})p(s_1|\mathbf{m})p(\mathbf{m}) \quad (36)$$

となる。ただし、上の式変形をするにあたって s_1 の観測と \mathbf{d} の観測は独立と見做した。この時、それぞれの確率は

$$p(\mathbf{d}|\mathbf{m}) \propto \exp\left(-\frac{1}{2\sigma_d^2} (\mathbf{d} - G\mathbf{m})^T (\mathbf{d} - G\mathbf{m})\right) \quad (37)$$

$$p(s_1|\mathbf{m}) \propto \exp\left(-\frac{1}{2\sigma_1^2} (s_1 - m_1)^T (s_1 - m_1)\right) \quad (38)$$

$$p(\mathbf{m}) \propto \exp\left(-\frac{1}{2}\lambda^2 (D\mathbf{m})^T D\mathbf{m}\right) \quad (39)$$

^{*10}今回は $i \leq 14$ において $w_i = 0$ としたこと深さ 14 km 以下では「深部で slip が 0 になる」と言う条件が直接効かないようにした。しかし、深さ 0km から (少しずつ大きくしながら) *no-slip condition* を効かせていくような重みの付け方も考えられる。また、今回は重みは深度に対して線形に増加するとしたが、これも、2次や3次の関数で重みを増加させていくことで、深い場所でより激しく、浅い場所ではより穏やかに「深部で slip が 0」と言う条件を効かせるやり方も考えられる。今回はあくまで一例として、恣意的な選び方を行ったが、より定量的な重みの決定方法についてはよくわかっていない。

^{*11}実際、*continuous slip* か *smooth slip* のどちらを採用するのが良いのだろうか。個人的には、*continuous slip* を用いて一階差分までしか拘束しない場合、ノイズに弱くなってしまうような気はしている。

のようにかける。^{*12}ただし、*a priori* information に対応する行列 D については、問 (2) で用いたものと同様のものを用いることにする。よって、(36)~(39) を用いれば、

$$p(\mathbf{m}|d, s_1) \propto \exp\left(-\frac{1}{2}\left(\frac{1}{\sigma_d^2}(\mathbf{d}-G\mathbf{m})^T(\mathbf{d}-G\mathbf{m}) + \frac{1}{\sigma_1^2}(s_1-m_1)^T(s_1-m_1) + \lambda^2(D\mathbf{m})^T D\mathbf{m}\right)\right) \quad (40)$$

となる。よって、求めるべき \mathbf{m} は、以下の E を最小にする \mathbf{m} である。^{*13}

$$E = \frac{1}{\sigma_d^2}(\mathbf{d}-G\mathbf{m})^T(\mathbf{d}-G\mathbf{m}) + \lambda^2(D\mathbf{m})^T D\mathbf{m} + \frac{1}{\sigma_1^2}(s_1-m_1)^T(s_1-m_1) \quad (41)$$

よって講義と同様に、極小では E の m_p による偏微分が 0 となると予想されることを用いれば、

$$\frac{\partial E}{\partial m_p} = \frac{2}{\sigma_d^2} \sum_k m_k \sum_i G_{ip} G_{ik} - \frac{2}{\sigma_d^2} \sum_i G_{ip} d_i + 2\lambda^2 \sum_k m_k \sum_i D_{ip} D_{ik} + f_p = 0 \quad (42)$$

$$\text{ただし； } f_p = \begin{cases} -\frac{2}{\sigma_1^2}(s_1-m_1) & (\text{iff } p=1) \\ 0 & (\text{otherwise}) \end{cases} \quad (43)$$

となる。よって、総和表記から行列表記等を復元すれば、

$$\frac{1}{\sigma_d^2} G^T G \mathbf{m} - \frac{1}{\sigma_d^2} G^T \mathbf{d} + \lambda^2 D^T D \mathbf{m} - \frac{1}{\sigma_1^2} \mathbf{e}_1 (s_1 - m_1) = 0 \quad (44)$$

となっており、これを \mathbf{m} について解くと

$$\mathbf{m} = \left(G^T G + \lambda^2 D^T D + \left(\frac{\sigma_d}{\sigma_1} \right)^2 \mathbf{e}_1 \mathbf{e}_1^T \right)^{-1} \left(G^T \mathbf{d} + \left(\frac{\sigma_d}{\sigma_1} \right)^2 s_1 \mathbf{e}_1 \right) \quad (45)$$

となる。^{*14*15*16}

\mathbf{m} を与える式が求まったので、次は damping parameter λ^2 を求める方法を考える。今回は特に、CVSS を用いて λ^2 を求める。^{*17}中でも、*leave-one-out cross validation* を行う。具体的には、 \mathbf{d} から j 番目の要素を除いたデータ $\mathbf{d}^{(-j)}$ と、 G から j 行目を取り除いた行列 $G^{(-j)}$ を用いて

$$\mathbf{m}^{(-j)} = \left(\left(G^{(-j)} \right)^T G^{(-j)} + \lambda^2 D^T D + \left(\frac{\sigma_d}{\sigma_1} \right)^2 \mathbf{e}_1 \mathbf{e}_1^T \right)^{-1} \left(\left(G^{(-j)} \right)^T \mathbf{d}^{(-j)} + \left(\frac{\sigma_d}{\sigma_1} \right)^2 s_1 \mathbf{e}_1 \right) \quad (46)$$

のように $\mathbf{m}^{(-j)}$ を求める。その後、始め取り除いたデータを得られた $\mathbf{m}^{(-j)}$ でどの程度精度よく再現できるかを、

$$E_j = \frac{1}{N\sigma_d^2} \left(d_j - G_j \mathbf{m}^{(-j)} \right)^2 \quad (47)$$

^{*12}分かりやすさのため $(s_1 - m_1)^2 = (s_1 - m_1)^T (s_1 - m_1)$ と書き換えている。

^{*13}式 (41) を見るとわかるが、今回の問題設定では σ_d^2 を λ^2 に吸収させることができない。よって、前回の問題とは異なり、 σ_d^2 を明示的に用いて問題を解かなければいけないことがわかる。

^{*14}逆行列が取れることは行列の対称性から保証される。

^{*15}途中で $m_1 = \mathbf{e}_1^T \mathbf{m}$ を用いた

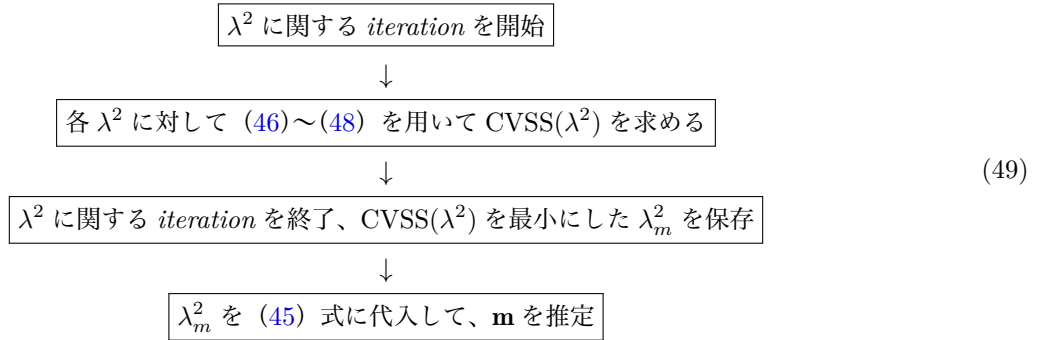
^{*16}途中で $\lambda^2 \sigma_d^2 = \lambda^2$ と置き直している。

^{*17}そもそも、(45) 式に対して ABIC がどういう形になるのかわからないので、CVSS しか選択肢がない。

を求めることで計算する。ただし、 G_j は G の j 行からなる行ベクトルであり、 $N = 20$ はデータの次元である。これを、 $j = 1 \sim 20$ に対して行えば、 E_j の和をとることによって CVSS を以下のように計算することができる。

$$CVSS = \sum_j E_j = \frac{1}{N} \sum_j \frac{1}{\sigma_d^2} \left(d_j - G_j \mathbf{m}^{(-j)} \right)^2 \quad (48)$$

ここまでの流れからわかるように、CVSS は λ^2 のみに依存した値である。よって、



というプロセスによって \mathbf{m} を求めることができる。^{*18}このようにして求めた \mathbf{m} が以下の図のよう。

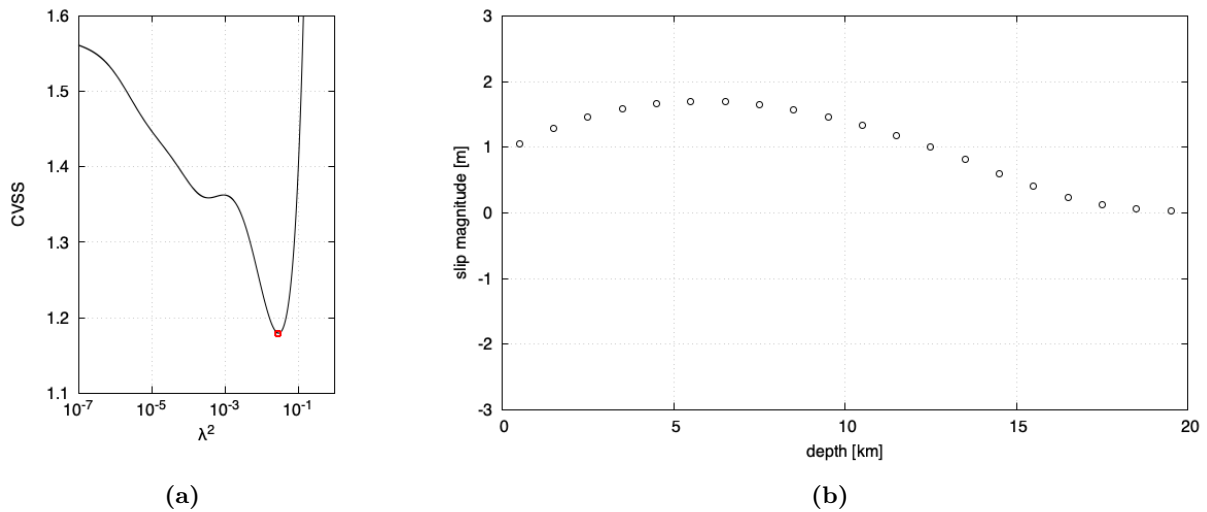


Figure7: Slip estimation using *continuous slip condition* and *weighted no-slip condition* and surface offset as a *priori* information. (a): CVSS as function of the damping parameter. (b): Estimation of slip magnitude s [m]

^{*18}例に倣って、特に深い意味はないが CVSS についても FORTRAN のコードをレポートの最後に貼っておく。

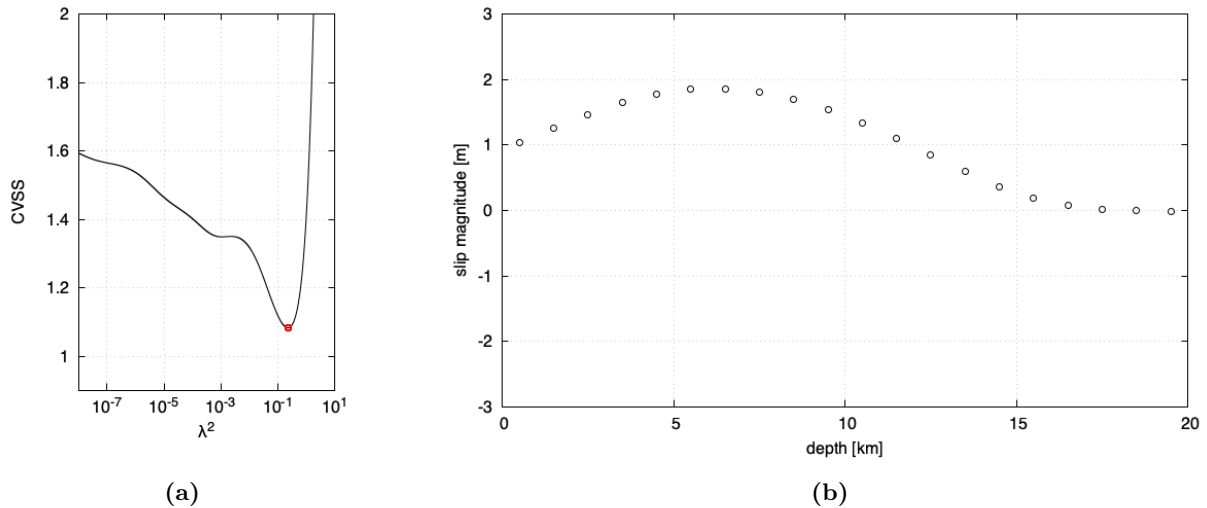


Figure8: Slip estimation using *smooth slip condition* and *weighted no-slip condition* and surface offset as *a priori* information. (a): CVSS as function of the damping parameter. (b): Estimation of slip magnitude s [m]

Figure7および Figure8はそれぞれ、*continuous slip condition* と *smooth slip condition* に対して CVSS method を用いて slip magnitude を推定したものである。左側の図はいずれも、CVSS(λ^2) を表しており、赤丸が λ_m^2 、すなわち \mathbf{m} の推定に用いた damping parameter である。また、深層での *no-slip* 条件については、いずれの場合においても *weighted no-slip* を用いている。

Figure7および Figure8のいずれにおいても、深度 0[km] での値が 1[m] となっていることは非常に重要である。これは、surface offset の情報が今回の推定にしっかり反映されていることを表している。

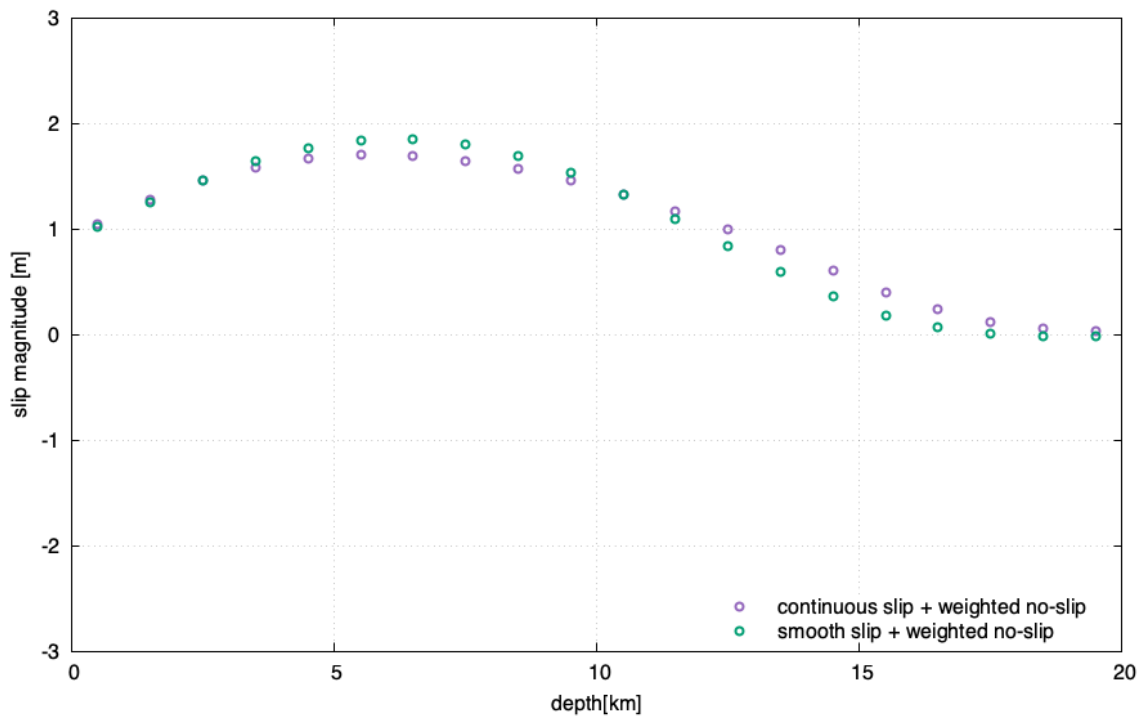


Figure9: Comparison of slip estimation under different *a priori* information. The correspondence is as follows; Purple points: data from Figure8. Green points: data from Figure7.

Figure9は今回の surface offset ありの推定で *continuous slip condition* と *smooth slip condition* を用いた場合の結果の違いを表している。両者に特段大きな違いがあるとは言えないが、曲率や深部での 0 への取束の滑らかさが (2) の時と逆転しているのは注目に値する。すなわち、今回の場合においては *continuous slip condition* の方がピークが潰れ、また全体的に滑らかになっている。ただ、ここら辺は λ_m^2 の値の違いなどによる誤差など、考えられる要因は様々にあるため、今回このようになっている原因の特定は難しい。

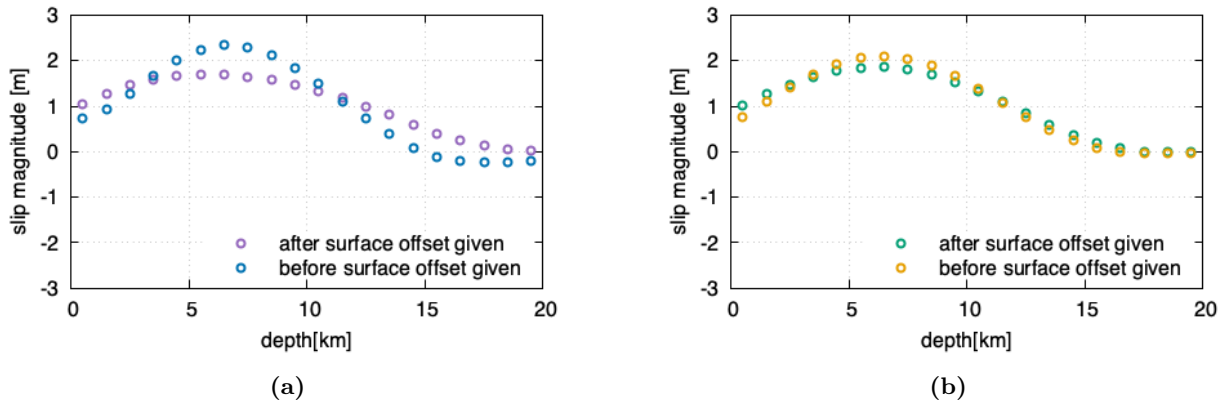


Figure10: Comparison of Slip estimation before/after surface offset was given. (a): *a priori* : *continuous slip condition* and *weighted no-slip condition* (b): *a priori* : *smooth slip condition* and *weighted no-slip condition*

最後に、Figure10では surface offset が与えられた前と後でどのように slip が変化したかを図示している。10a,10bがそれぞれ *continuous slip* と *smooth slip* に対応しているが、両者においても注目すべきなのは、深度 0 での値が 1[m] に吸い寄せられるにつれ、全体の slip の分布が平均して浅い方へ引き寄せられていることである。surface offset が観測で与えられる前はいずれの場合においても surface offset を 1[m] より小さく見積もっていたため、surface offset=1[m] が与えられたことで浅部での slip magnitude が全体として増大するに至ったと考えられる。

最後に、問 (3) の回答としては Figure7 と Figure8 の両方を採用することにする。やはり、*continuous* (一階差分) か *smooth* (二階差分) のどちらが *a priori information* としてふさわしいのか、自分では判断ができないため、二つを持って答えとさせていただきます。

おまけ：今回用いた FORTRAN コード

FORTRAN: ABIC method under (continuous slip / weighted no-slip)

```
1  program main
2  !1回差分
3  !深くなるほど重みつけて0に近づける
4  implicit none
5
6  real(8) :: data_vector(20)
7  real(8) :: G(20,20),depth(0:20),x(20)
8  real(8) :: G_transpose(20,20),mat_inverse(20,20)
9  real(8) :: G_g(20,20)
10 real(8) :: model_vector(20),model_vector_best(20)
11 real(8) :: D_mat(39,20),D_T_D(20,20),weight(20),alpha
12
13 real(8),parameter :: pi = 3.14159265359
14 integer :: i,j,k
15 real(8),allocatable :: WORK(:)
16 integer :: IPIV(20),INFO,LWORK=64
17
18 real(8) :: lambda_2,lambda_2min,lambda_2max,ratio,lambda_2_best
19 integer :: lambdamesh
20 real(8) :: temp(20,20),temp_vect(20)
21
22 integer :: N,M,rankDTD,MN
23 real(8) , allocatable :: s(:)
24 real(8) :: U(1,1),VT(1,1)
25 integer :: ldu,ldvt
26 real(8) :: WORK_query(1)
27 real(8) :: eps, tol
28 real(8) :: ABIC,temp_val_1,temp_val_2,temp_val_3,determinant
29 real(8) :: ABIC_best
30 integer :: sign
31
32 !初期数値決定-----
33 data_vector= (/0.142,0.138,0.174,0.180,0.208,&
34 & 0.238,0.290,0.348,0.491,0.529,-0.590,-0.459,-0.367,&
35 & -0.290,-0.239,-0.206,-0.166,-0.149,-0.125,-0.091/)
36
37 do i = 0,20
38     depth(i) = real(i,8)
39 end do
40
41 x = (/ -50.0, -45.0, -40.0, -35.0, -30.0, -25.0, -20.0, &
42 & -15.0, -10.0, -5.0, 5.0, 10.0, 15.0, 20.0, 25.0, 30.0, 35.0, 40.0, 45.0, 50.0/)
43
44 do i = 1,20
45     do j = 1,20
46         G(i,j)=(-1.0/pi)*( atan(depth(j)/x(i)) - atan(depth(j-1)/x(i)) )
47     end do
48 end do
49
50 D_mat = 0.0d0
51 ! 1~19行：今まで通りの1次差分
52 do i = 1,19
53     do j = 1,20
54         if (i == j) then
55             D_mat(i,j) = -1.0d0
56         else if (j == i+1) then
57             D_mat(i,j) = 1.0d0
```

```

58         else
59             D_mat(i,j) = 0.0d0
60         end if
61     end do
62 end do
63 ! 深さごとの重み w_i を決める (例: 10km 以深を強く 0 に寄せる)
64 alpha = 1.0 ! 全体のスケール (あとで調整)
65 do i = 1,20
66     if (i <= 14) then
67         weight(i) = 0.0d0 ! 0~10 km は自由
68     else
69         weight(i) = ((i-14.0)/6.0)**1.0 ! 10~20 km で 徐々に強く
70     end if
71 end do
72
73 ! 行 20~39 : sqrt(alpha*w_i)*e_i^T
74 do i = 1,20
75     if (weight(i) > 0.0d0) then
76         D_mat(19+i, i) = alpha*weight(i)
77     end if
78 end do
79 D_T_D = matmul(transpose(D_mat),D_mat)
80 !初期数値決定-----
81
82 !lambda_2をlogスケールで変化させてm(^2)を求め、ABICを算出する。
83 lambdamesh = 10000
84 lambda_2min = 1.0e-15
85 lambda_2max = 1.0e4
86 lambda_2_best = lambda_2min
87 ratio = (lambda_2max/lambda_2min)**(1.0/real(lambdamesh,8))
88
89 lambda_2=lambda_2min
90 LWORK = 64
91 allocate(WORK(LWORK))
92 open(10,file='ABIC_5.dat',status='replace',action='write')
93 do k = 1,lambdamesh
94
95     G_transpose = transpose(G)
96     temp = lambda_2*D_T_D
97     mat_inverse = matmul(G_transpose,G) + temp
98     call dgetrf(20,20,mat_inverse,20,IPIV,INFO)
99     call dgetri(20,mat_inverse,20,IPIV,WORK,LWORK,INFO)
100     G_g = matmul(mat_inverse,G_transpose)
101
102     model_vector= matmul(G_g,data_vector)
103     !write(*,*) 'model_parameter'
104     model_vector = matmul(G_g,data_vector)
105     do i = 1,20
106         !write(*,*) i,model_vector(i)
107     end do
108     !write(*,*)
109     !write(*,*) 'd-Gm'
110     temp_vect = matmul(G,model_vector)
111     do i = 1,20
112         !write(*,*) i,data_vector(i)-temp_vect(i)
113     end do
114     !write(*,*)
115
116     !calculating ABIC
117     N=20
118     M=20
119     MN = min(M,N)
120

```

```

121      !rank D^T Dの計算
122      temp = D_T_D
123      if (allocated(s)) then
124          deallocate(s)
125      end if
126      allocate(S(MN))
127      ldu = 1
128      ldvt = 1
129      call dgesvd('N','N',20,20,temp,20,S,U,ldu,VT,ldvt,WORK_query,-1,INFO)
130      LWORK = int(WORK_query(1))
131      deallocate(WORK)
132      allocate(WORK(LWORK))
133      temp = D_T_D
134      call dgesvd('N','N',20,20,temp,20,S,U,ldu,VT,ldvt,WORK,LWORK,INFO)
135      eps = epsilon(1.0d0)
136      tol = max(M,N) * eps * S(1)
137      rankDTD=0
138      do i = 1, MN
139          if (S(i) > tol) rankDTD = rankDTD + 1
140      end do
141
142      !ABIC
143      temp_val_1 = sum((data_vector-matmul(G,model_vector))*(data_vector-matmul(G,model_vector)))
144      temp_val_2 = lambda_2*sum(matmul(D_mat,model_vector)*matmul(D_mat,model_vector))
145      temp_val_1 = real(20-20+rankDTD,8)*log(temp_val_1+temp_val_2)
146      temp_val_2 = -rankDTD*log(lambda_2)
147      temp = matmul(G_transpose,G)+lambda_2*D_T_D
148      call dgetrf(20,20,temp,20,IPIV,INFO)
149      sign = 1
150      do i = 1,20
151          if(IPIV(i)/=i)then
152              sign = -sign
153          end if
154      end do
155      determinant = sign
156      do i = 1, n
157          determinant = determinant* temp(i,i)    ! U の対角成分の積
158      end do
159      temp_val_3 = log(abs(determinant))
160      ABIC = temp_val_1+temp_val_2+temp_val_3
161      if(k==1)then
162          ABIC_best=ABIC
163          model_vector_best = model_vector
164      end if
165      if (ABIC_best > ABIC) then
166          ABIC_best = ABIC
167          lambda_2_best = lambda_2
168          model_vector_best = model_vector
169      end if
170      write(10,*) lambda_2,ABIC
171
172      lambda_2 = lambda_2*ratio
173  end do
174
175  write(*,*) 'best_lambda_2=',lambda_2_best
176  close(10)
177  open(10,file='./model_parameter_5.dat',status='replace',action='write')
178  do i = 1,20
179      write(10,*) i-0.5,model_vector_best(i)
180  end do
181  close(10)
182  end program

```

FORTRAN: CVSS method under (smooth slip / weighted no-slip)

```

1 program main
2   !surface拘束あり
3   !さらにweighted,
4   !さらに2階差分正規化
5   implicit none
6
7   real(8) :: data_vector(20)
8   real(8) :: G(20,20),depth(0:20),x(20)
9   real(8) :: G_transpose(20,20),mat_inverse(20,20)
10  real(8) :: G_g(20,20)
11  real(8) :: model_vector(20),model_vector_best(20)
12  real(8) :: D_mat(38,20),D_T_D(20,20),weight(20),alpha
13
14  real(8),parameter :: pi = 3.14159265359
15  integer :: i,j,k
16  real(8),allocatable :: WORK(:)
17  integer :: IPIV(20),INFO,LWORK=64
18
19  real(8) :: lambda_2,lambda_2min,lambda_2max,ratio,lambda_2_best
20  integer :: lambdamesh
21  real(8) :: temp(20,20),temp_vect(20)
22
23  integer :: N,M,rankDTD,MN
24  real(8) , allocatable :: s(:)
25  real(8) :: U(1,1),VT(1,1)
26  integer :: ldu,ldvt
27  real(8) :: WORK_query(1)
28  real(8) :: eps, tol
29  real(8) :: ABIC,temp_val_1,temp_val_2,temp_val_3,determinant
30  real(8) :: ABIC_best
31  integer :: sign
32
33  real(8) :: e_1(20),sigd_sig1_pow2,G_j(19,20),d_j(19),CVSS,e1_e1_T(20,20),sig_d_2,&
34  & CVSS_best
35
36  !初期数値決定-----
37  data_vector= (/0.142,0.138,0.174,0.180,0.208,&
38  & 0.238,0.290,0.348,0.491,0.529,-0.590,-0.459,-0.367,&
39  &-0.290,-0.239,-0.206,-0.166,-0.149,-0.125,-0.091/)
40
41  do i = 0,20
42    depth(i) = real(i,8)
43  end do
44
45  x = (/ -50.0,-45.0,-40.0,-35.0,-30.0,-25.0,-20.0,&
46  & -15.0,-10.0,-5.0,5.0,10.0,15.0,20.0,25.0,30.0,35.0,40.0,45.0,50.0/)
47
48  do i = 1,20
49    do j = 1,20
50      G(i,j)=(-1.0/pi)*( atan(depth(j)/x(i)) - atan(depth(j-1)/x(i)) )
51    end do
52  end do
53
54  D_mat = 0.0d0
55  ! === 1~18行: 二階差分 (i = 2..19) ===
56  do i = 2, 19      ! 物理インデックス i
57    j = i - 1      ! 行番号 (1..18)
58    D_mat(j, i-1) = 1.0d0 ! m_{i-1}
59    D_mat(j, i ) = -2.0d0 ! m_i
60    D_mat(j, i+1) = 1.0d0 ! m_{i+1}
61  end do
62  ! 深さごとの重み w_i を決める (例: 10km 以深を強く 0 に寄せる)

```

```

63 alpha = 1.0 ! 全体のスケール (あとで調整)
64 do i = 1,20
65     if (i <= 14) then
66         weight(i) = 0.0d0 ! 0~10 km は自由
67     else
68         weight(i) = ((i-14.0)/6.0)**1.0 ! 10~20 km で 徐々に強く
69     end if
70 end do
71
72 ! 行 19~38 : sqrt(alpha*w_i)*e_i^T
73 do i = 1,20
74     if (weight(i) > 0.0d0) then
75         D_mat(18+i, i) = alpha*weight(i)
76     end if
77 end do
78 D_T_D = matmul(transpose(D_mat),D_mat)
79 ! 初期数値決定 -----
80
81 D_T_D = matmul(transpose(D_mat),D_mat)
82
83 e_1 = 0.0
84 e_1(1) = 1.0
85 sigd_sig1_pow2 = (0.02/0.05)**2.0
86 sig_d_2 = 0.02**2.0
87 ! 初期数値決定 -----
88
89 ! lambda_2をlogスケールで変化させてm(^2)を求め、ABICを算出する。
90 lambdamesh = 100000
91 lambda_2min = 1.0e-8
92 lambda_2max = 1.0e3
93 lambda_2_best = lambda_2min
94 ratio = (lambda_2max/lambda_2min)**(1.0/real(lambdamesh,8))
95
96 lambda_2=lambda_2min
97 LWORK = 64
98 allocate(WORK(LWORK))
99 open(10,file='CVSS_2.dat',status='replace',action='write')
100 do k = 1,lambdamesh
101     !write(*,*) real(k,8)/real(lambdamesh,8)
102     CVSS = 0.0
103     do j = 1,20
104         do i = 1,19
105             if(i<j)then
106                 d_j(i) = data_vector(i)
107             else if (i>=j) then
108                 d_j(i) = data_vector(i+1)
109             end if
110
111             if(i<j)then
112                 G_j(i,:) = G(i,:)
113             else if (i>=j) then
114                 G_j(i,:) = G(i+1,:)
115             end if
116         end do
117
118
119         e1_e1_T = 0.0
120         e1_e1_T(1,1) = 1.0
121         temp = lambda_2*D_T_D+sigd_sig1_pow2*e1_e1_T
122         mat_inverse = matmul(transpose(G_j),G_j) + temp
123         call dgetrf(20,20,mat_inverse,20,IPIV,INFO)
124         call dgetri(20,mat_inverse,20,IPIV,WORK,LWORK,INFO)
125         G_g = mat_inverse

```

```

126
127     temp_vect = matmul(transpose(G_j),d_j) + sigd_sig1_pow2*e_1
128     model_vector= matmul(G_g,temp_vect)
129
130
131     CVSS = CVSS + (1.0/sig_d_2)*((data_vector(j)-dot_product(G(j,:),model_vector))**2.0)
132
133 end do
134 CVSS = CVSS/20.0
135
136 temp = lambda_2*D_T_D+sigd_sig1_pow2*e1_e1_T
137 mat_inverse = matmul(transpose(G),G) + temp
138 call dgetrf(20,20,mat_inverse,20,IPIV,INFO)
139 call dgetri(20,mat_inverse,20,IPIV,WORK,LWORK,INFO)
140 G_g = mat_inverse
141
142 temp_vect = matmul(transpose(G),data_vector) + sigd_sig1_pow2*e_1
143 model_vector= matmul(G_g,temp_vect)
144
145 if(k==1)then
146     CVSS_best=CVSS
147     model_vector_best = model_vector
148 end if
149 if (CVSS_best > CVSS) then
150     CVSS_best = CVSS
151     lambda_2_best = lambda_2
152     model_vector_best = model_vector
153 end if
154
155 write(10,*) lambda_2,CVSS
156
157 lambda_2 = lambda_2*ratio
158 end do
159
160 write(*,*) 'best_lambda_2=',lambda_2_best
161 close(10)
162 open(10,file='./model_parameter_surface_2.dat',status='replace',action='write')
163 do i = 1,20
164     write(10,*) i-0.5,model_vector_best(i)
165 end do
166 close(10)
167
168 end program

```