Supplemental data to *Efficient hyper-parameter determination for regularised linear BRDF parameter retrieval*

1. Solution of the Tikhonov Smoothing Equation

To solve the Tikhonov smoothing equation $(\mathbf{K}^{\mathrm{T}}\mathbf{K}+\lambda^2\mathbf{B}^{\mathrm{T}}\mathbf{B})\boldsymbol{f}=\mathbf{K}^{\mathrm{T}}\boldsymbol{\rho}$, subject to $\|\boldsymbol{\epsilon}\|=n\delta^2$, we apply the Generalised Singular Value Decomposition (GSVD) to the matrix pair $(\mathbf{K},\mathbf{B})$, where it is assumed $\mathbf{K}\in\mathbb{R}^{m\times n}$ and $\mathbf{B}\in\mathbb{R}^{p\times n}$. The Generalised Singular Value Decomposition has been defined elsewhere (Bai 1992; Hansen 1992; Hansen and O'Leary 1993; Hansen 1996; Golub and Van Loan 2014) and implemented in LAPACK (Anderson et al. 1999). The GSVD decompositions are used to calculate $\boldsymbol{f}$, $\|\mathbf{B}\boldsymbol{f}\|$ and $\|\boldsymbol{\epsilon}\|$.

The GSVD computes the simultaneous decompositions $\boldsymbol{K}=\mathbf{U}\boldsymbol{\Sigma}[\mathbf{0}\ \ \mathbf{R}]\mathbf{Q}^{\mathrm{T}}$ and $\mathbf{B}=\mathbf{V}\mathbf{M}[\mathbf{0}\ \ \mathbf{R}]\mathbf{Q}^{\mathrm{T}}$, where $\mathbf{R}\in\mathbf{R}^{r\times r}$ is upper triangular and non-singular and $\mathbf{U},\mathbf{Q},\mathbf{V}$ are all orthogonal matrices of appropriate dimensions, $\boldsymbol{\Sigma}$ is a $m\times r$ matrix with the property that $\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{\Sigma}=\mathrm{diag}(\alpha_1^2,...,\alpha_r^2)$ with α_i between zero and unity and $\mathbf{M}$ is a $p\times r$ matrix with the property the $\mathbf{M}^{\mathrm{T}}\mathbf{M}=\mathrm{diag}(\beta_1^2,...,\beta_r^2)$ with β_i between zero and unity. The ratios α_i/β_i are the *generalised singular values* of the matrix pair $(\mathbf{K},\mathbf{B})$. Some of these singular values may equal zero or infinity. The GSVD has the added property that $\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{\Sigma}+\mathbf{M}^{\mathrm{T}}\mathbf{M}=\mathbf{I}_r$, the $r\times r$ identity matrix.

For all cases studied here $\mathbf{K}$ has full row rank equal to m with $m<<n$, $\mathbf{B}$ is a singular square matrix ($p=n$) with $\mathrm{rank}(\mathbf{B})=n-3$ and the composite matrix $\begin{pmatrix}\mathbf{K}\\ \mathbf{B}\end{pmatrix}$ has full column rank $r=n$. The matrix $\mathbf{R}$ is then $n\times n$ and nonsingular. We may therefore define the nonsingular square matrix $\mathbf{X}=\mathbf{Q}\mathbf{R}^{-1}$ with the properties that $\mathbf{K}\mathbf{X}=\mathbf{U}\boldsymbol{\Sigma}$, $\mathbf{B}\mathbf{X}=\mathbf{V}\mathbf{M}$, and $\mathbf{X}^{\mathrm{T}}\mathbf{K}^{\mathrm{T}}\mathbf{K}\mathbf{X}=\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{\Sigma}$, $\mathbf{X}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{B}\mathbf{X}=\mathbf{M}^{\mathrm{T}}\mathbf{M}$. The matrices $\boldsymbol{\Sigma}$ and $\mathbf{M}$ may be written

$$\boldsymbol{\Sigma}=\begin{pmatrix}\mathbf{I}_3 & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{C} & \mathbf{0}\end{pmatrix},\qquad \mathbf{M}=\begin{pmatrix}\mathbf{0} & \mathbf{S} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{n-m}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{pmatrix},$$

where $\mathbf{C}=\mathrm{diag}(c_{ii})$ and $\mathbf{S}=\mathrm{diag}(s_{ii})$ with $\mathbf{C}^2+\mathbf{S}^2=\mathbf{I}_{m-3}$ and $c_{ii}=\alpha_{3+i}$ and $s_{ii}=\beta_{3+i}$ for $i=1,...,m-3$.

1.1. *Determining the solution $\boldsymbol{f}$*

To solve the equation $(\mathbf{K}^{\mathrm{T}}\mathbf{K}+\lambda^2\mathbf{B}^{\mathrm{T}}\mathbf{B})\boldsymbol{f}=\mathbf{K}^{\mathrm{T}}\boldsymbol{\rho}$ for a given value of λ, we apply the transformation $\mathbf{X}$ to our equation and define a change of basis $\boldsymbol{f}=\mathrm{X}\boldsymbol{g}$, to give

$$\mathbf{X}^{\mathrm{T}}(\mathbf{K}^{\mathrm{T}}\mathbf{K}+\lambda^2\mathbf{B}^{\mathrm{T}}\mathbf{B})\mathbf{X}\boldsymbol{g}=\mathbf{X}^{\mathrm{T}}\mathbf{K}^{\mathrm{T}}\boldsymbol{\rho}.$$

With the identities defined previously we then have the diagonalised equation

$$\mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma} + \lambda^2\mathbf{M}^{\mathrm{T}}\mathbf{M}\boldsymbol{g} = \mathbf{\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\boldsymbol{\rho}\,.$$

Equating components on the right and left hand sides gives

$$\begin{aligned} g_i &= \mathbf{U}_i^{\mathrm{T}}\boldsymbol{\rho}, && i = 1, 2, 3,\\ g_i &= (\alpha_i/(\alpha_i^2 + \lambda^2\beta_i^2))(\mathbf{U}_i^{\mathrm{T}}\boldsymbol{\rho}), && i = 4, \ldots, m,\\ g_i &= 0, && i = m+1, \ldots, n, \end{aligned}$$

where $\mathbf{U}_i$ denotes the ith column of $\mathbf{U}$. The solution is then given by $\boldsymbol{f} = \mathbf{X}\boldsymbol{g}$.

1.2. *Calculating* $\|\mathbf{B}\boldsymbol{f}\|$

Using the change of basis $\boldsymbol{f} = \mathbf{X}\boldsymbol{g}$ we find that $\|\mathbf{B}\boldsymbol{f}\| = \|\mathbf{B}\mathbf{X}\boldsymbol{g}\| = \|\mathbf{M}\boldsymbol{g}\|$. The components of this vector are then given by

$$\begin{aligned} (\mathbf{M}\boldsymbol{g})_i &= (\alpha_i\beta_i/(\alpha_i^2 + \lambda^2\beta_i^2))(\mathbf{U}_i^{\mathrm{T}}\boldsymbol{\rho}), && i = 1, \ldots, m-3,\\ (\mathbf{M}\boldsymbol{g})_i &= 0, && i = m-2, \ldots, n. \end{aligned}$$

and the norm follows directly.

1.3. *Calculating the residual norm*

By the properties of the GSVD and with the change of basis $\boldsymbol{f} = \mathbf{X}\boldsymbol{g}$, we have $\boldsymbol{\epsilon} = \boldsymbol{\rho} - \mathbf{K}\boldsymbol{f} = \boldsymbol{\rho} - \mathbf{U}\mathbf{\Sigma}\boldsymbol{g}$, and it follows that $\|\boldsymbol{\epsilon}\| = \|\mathbf{U}^{\mathrm{T}}\boldsymbol{\epsilon}\| = \|\mathbf{U}^{\mathrm{T}}\boldsymbol{\rho} - \mathbf{\Sigma}\boldsymbol{g}\|$ by the orthogonality of $\mathbf{U}$. The non-zero components of $\mathbf{U}^{\mathrm{T}}\boldsymbol{\rho} - \mathbf{\Sigma}\boldsymbol{g}$ are given by $(1 - (\alpha_i^2/(\alpha_i^2 + \lambda^2\beta_i^2)))(\mathbf{U}_i^{\mathrm{T}}\boldsymbol{\rho}),\ i = 3, \ldots, m$, and the norm again follows directly.

Appendix A. Pseudocode

This pseudocode outlines 4 key procedures for this study:

- Computation of the solution norm $\|\mathbf{B}\boldsymbol{f}\|$.
  Inputs need to include the matrices $\mathbf{K}$, $\mathbf{B}$, $\boldsymbol{\rho}$, *and* λ.
- Computation of the residual norm $\|\boldsymbol{\epsilon}\|$.
  Same inputs as above.
- Computation of the solution $\boldsymbol{f}$.
  Same inputs as above.
- Determination of λ that satisfies $\|\boldsymbol{\epsilon}\| = \sqrt{n}\delta$.
  Inputs need to include $\mathbf{K}$, $\mathbf{B}$, $\boldsymbol{\rho}$, *the length* v *of an initial* $\boldsymbol{\Lambda}$ *vector with lower and upper bounds, delta, a tolerance (tol) and maximum number of iterations (maxIter).*

For all of these algorithms we assume the outputs of the GSVD as defined by its implementation in LAPACK (Anderson et al. 1999). For the benefit of completion we also include the code for the case $m \geq r$ (not detailed above, but can also be derived from GSVD properties as defined in Anderson et al. (1999)).

Algorithm 1 Compute $\|\mathbf{B}\boldsymbol{f}\|$

procedure COMPUTESOLUTIONNORM($\mathbf{K}$, $\mathbf{B}$, $\boldsymbol{\rho}$, λ)
  compute GSVD$(\mathbf{K}, \mathbf{B})$
  filter $\leftarrow \dfrac{\alpha}{\alpha^2 + \lambda^2\beta^2}$
  if $m \geq r$ **then**
    for i **from** 1 **to** l **do**
      $(\mathbf{B}\boldsymbol{f})_i \leftarrow (\text{filter} \cdot \beta_i) \cdot (\mathbf{U}_i^{\mathrm{T}}\boldsymbol{\rho})$
  else
    for i **from** 1 **to** m **do**
      $(\mathbf{B}\boldsymbol{f})_i \leftarrow (\text{filter} \cdot \beta_i) \cdot (\mathbf{U}_i^{\mathrm{T}}\boldsymbol{\rho})$
  return $\|\mathbf{B}\boldsymbol{f}\|$

Algorithm 2 Compute $\|\epsilon\|$

procedure COMPUTERESIDUALNORM($\mathbf{K}$, $\mathbf{B}$, $\boldsymbol{\rho}$, λ)
  compute GSVD$(\mathbf{K}, \mathbf{B})$
  filter $\leftarrow \dfrac{\alpha}{\alpha^2 + \lambda^2\beta^2}$
  if $m \geq r$ **then**
    for i **from** 1 **to** r **do**
      $\epsilon_i \leftarrow (1 - \text{filter} \cdot \alpha_i) \cdot (\mathbf{U}_i^{\mathrm{T}}\boldsymbol{\rho})$
    for i **from** r+1 **to** m **do**
      $\epsilon_i \leftarrow \cdot(\mathbf{U}_i^{\mathrm{T}}\boldsymbol{\rho})$
  else
    for i **from** 1 **to** m **do**
      $\epsilon_i \leftarrow (1 - \text{filter} \cdot \alpha) \cdot (\mathbf{U}_i^{\mathrm{T}}\boldsymbol{\rho})$
  return $\|\boldsymbol{\epsilon}\|$

Algorithm 3 Compute BRDF kernel weights $\boldsymbol{f}$ given λ

procedure COMPUTEBRDF($\mathbf{K}$, $\mathbf{B}$, $\boldsymbol{\rho}$, λ)
    compute GSVD($\mathbf{K}, \mathbf{B}$)
    filter $\leftarrow \dfrac{\alpha}{\alpha^2 + \lambda^2\beta^2}$
    if $n > r$ **then**
        $R \leftarrow \begin{pmatrix} \mathbf{I}_{n-r} & \mathbf{0}_{n-r,r} \\ \mathbf{0}_{r,n-r} & \mathbf{R} \end{pmatrix}$
    $\mathbf{X} \leftarrow \mathbf{Q}\mathbf{R}^{-1}$
    if $m \geq r$ **then**
        for i **from** $(n-r+1)$ **to** n **do**
            idx $\leftarrow i - (n-r)$
            $g_i \leftarrow \text{filter}_{idx} \cdot (\mathbf{U}_{idx}^{\mathrm{T}}\boldsymbol{\rho})$
    else
        for i **from** $(n-r+1)$ **to** $(n-r+m)$ **do**
            idx $\leftarrow i - (n-r)$
            $g_i \leftarrow \text{filter}_{idx} \cdot (\mathbf{U}_{idx}^{\mathrm{T}}\boldsymbol{\rho})$
    $\boldsymbol{f} \leftarrow \mathbf{X}\boldsymbol{g}$
    return $\boldsymbol{f}$

Algorithm 4 Compute λ that matches $\|\epsilon\| = \sqrt{n}\delta$

procedure BISECTION($\mathbf{K}$, $\mathbf{B}$, $\boldsymbol{\rho}$, v, lower, upper, δ, tol, maxIter)
    $\boldsymbol{\Lambda} \leftarrow$ **create logarithmic sequence**(lower,upper) **of length** v
    $\boldsymbol{\Lambda} \leftarrow$ **pad** 0 **to** $\boldsymbol{\Lambda}$
    for i **from** 1 **to** $v+1$ **do**
        $\text{lambdaNorm}_i \leftarrow$ COMPUTERESIDUALNORM($\mathbf{K}$, $\mathbf{B}$, $\boldsymbol{\rho}$, $\boldsymbol{\Lambda}_i$)
    $n \leftarrow$ **length of** $\boldsymbol{\rho}$
    negIndex $\leftarrow$ **which indices** lambdaNorm $< \sigma$
    posIndex $\leftarrow$ **which indices** lambdaNorm $> \sigma$
    if is empty negIndex **or** posIndex **then return** -1
    else
        $l \leftarrow$ **length of** negIndex
        $a \leftarrow \text{negIndex}_l$
        $b \leftarrow \text{posIndex}_1$
        iter $\leftarrow 1$
        while iter $\leq$ maxIter **do**
            $c \leftarrow (a+b)/2$
            aNorm $\leftarrow$ COMPUTERESIDUALNORM($\mathbf{K}$, $\mathbf{B}$, $\boldsymbol{\rho}$, a)
            bNorm $\leftarrow$ COMPUTERESIDUALNORM($\mathbf{K}$, $\mathbf{B}$, $\boldsymbol{\rho}$, b)
            cNorm $\leftarrow$ COMPUTERESIDUALNORM($\mathbf{K}$, $\mathbf{B}$, $\boldsymbol{\rho}$, c)
            if (cNorm- $\sqrt{n}\ \delta) = 0$ **or** $(b-a)/2 <$ tol **then return** c
            iter $\leftarrow$ iter+1
            if sign(cNorm)=**sign**(aNorm) **then**
                $a \leftarrow c$
            else $b \leftarrow c$

Algorithm 5 Define a BRDF solution $\boldsymbol{f}$

1: **procedure** SOLVEBRDF($\mathbf{K}$, $\mathbf{B}$, $\boldsymbol{\rho}$, v, lower, upper, δ, tol, maxIter)
2: $\lambda \leftarrow$BISECTION($\mathbf{K}$, $\mathbf{B}$, $\boldsymbol{\rho}$, v, lower, upper, δ, tol, maxIter)
3: $\boldsymbol{f} \leftarrow$ COMPUTESOLUTION($\mathbf{K}$, $\mathbf{B}$, $\boldsymbol{\rho}$, λ)

References

Anderson, E., Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, et al. 1999. *LAPACK Users' Guide*. 3rd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Bai, Zhaojun. 1992. "The CSD, GSVD, Their Applications and Computations." In *IMA Preprints Series*, Vol. 958, 3–5. University of Minnesota.

Golub, Gene H., and C. Van Loan. 2014. *Matrix Computations*. 4th ed. Johns Hopkins University Press.

Hansen, P. 1992. "Analysis of Discrete Ill-Posed Problems by Means of the L-Curve." *SIAM Review* 34 (4): 561–580.

Hansen, P., and D. O'Leary. 1993. "The Use of the L-Curve in the Regularization of Discrete Ill-Posed Problems." *SIAM Journal on Scientific Computing* 14 (6): 1487–1503.

Hansen, Per Christian. 1996. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM Books, Philadelphia, PA.