GEOSDAS SCM STUDY

Samson Cheung
cheung@halcyonsystems.com
August-September, 2002

Abstract:

The current SCM (Successive Correction Method) routine in GEOSDAS seems to leave room for improvement. This report first shows the problem with the current SCM smoothing solution, and then introduces an iterative algorithm and a Sigmoid function to improve the solution. Parameters in the iterative algorithm are studied in this report. The overall performance and accuracy of the iterative method exceeds the current SCM routine.

The Current Result

The precipitation simulation in the GEOSDAS employs a single column model as a simulation base. After the analysis is done, the analysis increment is in the form of vertical profile (column); it requires a mathematical routine to smooth the increment solution on the horizontal grid. This is currently done by a routine called SCM (Successive Correction Method). In this report, an off-line SCM routine with a given data set is studied. Figure 1 shows the input increment data and the SCM results at the 55th computational level, which is near the ground level (1000mb). The computational resolution is 1x1.25 degrees.

![Input Increment Data](a)

![SCM Results](b)

It is clear that there are four bands of increment data (input of SCM) in Figure 1a. Between the four bands of data there is a gap in which there are no increment data. The SCM result is shown in Figure 1b. Here, four distinct bands are seen, as well, after the smoothing. This
indicates that the current SCM scheme may have difficulty in the region where no input data are available.

Zooming into the region near the gap (see Figure 2) it is obvious that the current method may not give accurate results as it should be. Figure 2 overlays the increment data and the SCM solution within a smaller region. The isolated “red” data (see the arrow) has a value of 0.006, but the SCM has a value of 0.002 in that location. In addition, the cliff-like feature indicates that there may be problems in the SCM routine when input data are insufficient.

Figure 2. Current SCM has accuracy issue at isolated input data.

**SCM Introduction**

The algorithm of the current SCM routine is as follows: Given a set of input data for each computational grid point, $\vec{r}_i$, define a maximum radius of influence ($R$). Within the radius of $R$, there are $K_i$ numbers of input data. Let $f_o(\vec{r}_k)$ be the input data at location $\vec{r}_k$. Then, the smoothing solution at $\vec{r}_i$ is given by

$$f_A(\vec{r}_i) = \frac{\sum_{k_i} w(r_{ik}) f_o(\vec{r}_k)}{\sum_{k_i} w(r_{ik})}$$

(1)

where $w(r_{ik})$ is a weighting function and $r_{ik}$ is the distance between $\vec{r}_i$ and $\vec{r}_k$. There are weighting function choices in the current SCM routine -- the Cressman function [1],

$$w(r) = \frac{R^2 - r^2}{R^2 + r^2}$$

(2)
and the Sasaki function (1960)\(^1\) [2],

\[
w(r) = \exp(-\frac{r^2}{2R^2})
\]  

(3)

The choice of the maximum radius of influence \(R\) is hard-coded to 500km in the SCM routine.

### Barnes Algorithm

The general analysis formulation given by Bergthorsson and Doos (1955) [3] is

\[
f_A(\vec{r}_i) = f_B(\vec{r}_i) + \frac{\sum_{k_i} w(r_{ik}) \{f_o(\vec{r}_k) - f_B(\vec{r}_k)\}}{\sum_{k_i} w(r_{ik}) + \epsilon^2}
\]  

(4)

where \(f_B\) is the background value and \(\epsilon^2 = E_0^2 / E_B^2\), the expected observation error variance normalized by the expected background error variance.

In the current SCM application there is no observation, only smooth. Thus, there are no background data, and it is expected that the increment data are accurate; so \(f_B(\vec{r}_i)\) and \(\epsilon^2\) are zero. Thus, equation (4) reduces to (1).

A way to improve the accuracy of the analysis is by utilizing an iterative scheme, Barnes algorithm [4]. Equation (4) would be

\[
f_A^{(j+1)}(\vec{r}_i) = f_A^{(j)}(\vec{r}_i) + \frac{\sum_{k_i} w^{(j)}(r_{ik}) \{f_o(\vec{r}_k) - f_A^{(j)}(\vec{r}_k)\}}{\sum_{k_i} w^{(j)}(r_{ik})}
\]  

(5)

where \(f_A^{(j)}\) is the \(j\)th iteration of \(f_A\). The initial guess of this algorithm is

\[
f_A^0(\vec{r}_i) = \frac{\sum_{k_i} w^{(0)}(r_{ik}) f_0(\vec{r}_k)}{\sum_{k_i} w^{(0)}(r_{ik})}
\]  

(6)

Note that the weighting function changes after each iteration. Take the Sasaki function, equation (3), for example. We will have

\(^1\) The code names the weighting function “Gaussian”; actually, that function was first used by Sasaki (1960).
$$w^{(j)}(r) = \exp\left(-\frac{r^2}{2R_j^2}\right), \quad R_j^2 = \gamma^j R_0^2$$

(7)

where $\gamma$ is factored less than or equal to 1. It has been proven that when $\gamma$ is equal to or less than 1, the algorithm converges. We will discuss the effect of $\gamma$ on the SCM solution later.

Figures 3a and 3b compare the original SCM calculation and the 10 iterations of the Barnes algorithm with the Sasaki weighting function and $\gamma = 0.6$.  

Figure 3a. The original SCM solution.

Figure 3b. The 10-iteration Barnes’ result with Sasaki function and $\gamma = 0.6$. 
It is shown that the new method (Figure 3b) has improved the accuracy of the solution (see arrow A). The red point is observation data with a value of 0.006. One would expect the smoothing routine gives a closed value of 0.006 in the vicinity region at the observation point. The original SCM gives a value of 0.002 in the vicinity; whereas, the new method gives a value of 0.005 in the vicinity region. So, the new method seems to provide more accurate result.

However, some unexpected features appear in the regions where there are no increment data (see arrow B). These concentrated features lead one to suspect that the SCM method could be improved. There are two possibilities that come to mind -- namely, the hard-coded radius of influence, 500km, may not be enough; and the weight function is not a good one to use.

It is noted that increasing the number of iterations does not improve the big picture in the smoothing. For $\gamma = 0.6$, 10 iterations are sufficient to reveal the important features of the smoothing solution. We will discuss this later in another section.

### Weighting Function

For a given computational grid point, the SCM utilizes the increment (input) data in the vicinity area to approximate the SCM solution (output) at the grid point. A weighting function is used to give more weight for those input data close to the grid point (small $r$) and less weight for those data away from the grid point (big $r$).

Figure 4 shows the plots of the Sasaki and Cressman weighting functions, equations (2) and (3), with $R$ being 500km. The weighting function is zero when $r$ is larger than $R$. In a relative sense, the Cressman function allows too much weight for the big value of $r$; whereas the Sasaki function performs as a straight line with weights evenly distributed.

In this report, a Sigmoid function is introduced as a weighting function. The plot of the Sigmoid function is also shown in Figure 4.

The formulation of the Sigmoid function is

$$w(r) = \frac{1}{1 + \exp\left\{\alpha \cdot (r - R/2)/R\right\}}$$  

(8)
where $\alpha$ is a constant. The Sigmoid function in Figure 4 has $\alpha=13$. A series of Sigmoid functions, with a different $\alpha$, is shown in Appendix A. It is interesting to see the effect of it on the SCM routine. Appendix B shows the original SCM routine (no iteration) with these 3 functions.

The solution of 10 iterations of the Barnes algorithm with the Sigmoid weighting function and $\gamma=0.6$ is shown in Figure 5.

![Figure 5](image_url)

**Figure 5.** The 10-iteration Barnes' calculation with Sigmoid function and $\gamma=0.6$.

In comparing Figure 3b and Figure 5, the differences between the Sasaki function and the Sigmoid function on the smoothing solution are revealed. The cliff-like concentrated features in Figure 3b are gone; and the smoothing solution seems to match the input data.

---

### Note on Iterations and value of $\gamma$

How many iterations should the Barnes algorithm take? In the current application, since there is no accounting for an observation error (we assumed $\varepsilon^2=0$), the error could add up as we iterate.

![Figure 6a and b](image_url)

**Figure 6a and b.** SCM smoothing with 10 iterations and 5 iterations, respectively. $\gamma=0.6$. 

In addition, the more it iterates, the more computational resources it requires. So, we want to stop at some point. Figure 6 allows us to compare a 10-iteration solution and a 5-iteration solution. It shows that the two solutions have very comparable features and values, but one is half the computation than the other.

The functionality of $\gamma$ in the weighting function is to decrease the radius of interference ($R$) for every iteration. As Daley’s book indicates, if $\gamma$ is small, it will increase the rate of convergence and reveal more details. However, in our case, we have a large area with no increment (input) data and occasional, isolated input data (see arrow A in Figure 3). Figure 7a shows that a small $\gamma$ will make these isolated input data stand out – perhaps even out of proportion. Thus, we don’t want $\gamma$ to be too small. If $\gamma$ is large (close to 1), the convergence is relatively slow. And, in the regions where input data are ample, you lose the details that a smaller $\gamma$ would provide (see Figure 7b). A recommended number of iterations is ~ 4-6 and $\gamma$ ~ 0.4-0.6.
**Timing**

The smoothing solution is improved by the iterative method; but a price is paid in computational time. The table below shows the timing (in seconds) of the original routine and the modified algorithm on Kalnay (400MHz Origin O2K).

<table>
<thead>
<tr>
<th>Number of CPUs</th>
<th>Original Routine</th>
<th>Iterative Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.4097</td>
<td>11.845</td>
</tr>
<tr>
<td>4</td>
<td>0.7678</td>
<td>5.9607</td>
</tr>
<tr>
<td>6</td>
<td>0.5152</td>
<td>3.9963</td>
</tr>
<tr>
<td>8</td>
<td>0.5494</td>
<td>3.9867</td>
</tr>
</tbody>
</table>

Note: Parallelization only from Level 44 to 55, therefore no speedup after 6 CPUs. More information and improvement about the timing is in the next section.

**Further Improvement**

Although the Barnes’ method has significantly improved the accuracy and taken care the cliff-like features in the original code; there are still have room for improvement in terms of timing and accuracy. In Figure 6, there is a large negative value (-0.015), near the region 6°S longitude and latitude 173°W, that seems to be undesirable.

Currently, $R$ is hard-coded to be 500km. The following experiment shows the speed and the Min/Max values of the Barnes’ method as $R$ changes. The results are obtained using 6 CPUs on Kalnay by 6-iteration Barnes’ method with Sigmoid function and $\gamma=0.6$.

<table>
<thead>
<tr>
<th>Input Satellite Data</th>
<th>Maximum</th>
<th>Minimum</th>
<th>Timing (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0114896</td>
<td>-0.0115246</td>
<td>N/A</td>
</tr>
<tr>
<td>$R=500$km</td>
<td>0.0114896</td>
<td>-0.00276401</td>
<td>3.97732615</td>
</tr>
<tr>
<td>$R=400$km</td>
<td>0.0114896</td>
<td>-0.00237385</td>
<td>2.72645807</td>
</tr>
<tr>
<td>$R=300$km</td>
<td>0.0114896</td>
<td>-0.00192094</td>
<td>1.84244204</td>
</tr>
<tr>
<td>$R=200$km</td>
<td>0.0114896</td>
<td>-0.00143323</td>
<td>0.947566986</td>
</tr>
</tbody>
</table>

The bigger the $R$, the more area to search for, thus, it will increase the computational time. This experiment indicates that if $R$ is reduced, we decrease the searching time. In addition, a big $R$ near the data boundary will make the SCM routine to consider too many undefined data in the “gap”; thus accordingly, increase the smoothing error. In this study, the minimum value is happened to be at the data boundary. Therefore, this experiment indicates that a smaller $R$ will decrease the smoothing error near the data boundary.
Figure 8, on the right, is the results from the $R=200\text{km}$ case. It can be compared with Fig. 6 in this study.

Thus, by reducing to the maximum radius of influence $R$ to the appropriate value (250km is recommended in this case), we can increase the computational speed and obtain more accurate result.

**Figure 8.** The 6 iteration Barnes’ method with Sigmoid function and $\gamma =0.6$. The maximum radius of influence $R$ is 200km.

## Conclusion

This report reveals the shortcomings of the current SCM routine and introduces an iterative scheme and a Sigmoid function to improve the accuracy of the routine. It has been shown that the iterative method produces a more reasonable/accurate output result than the original SCM routine. This report also discusses the result of the parametric study of the iterative scheme and the radius of inference ($R$). The recommended number of iterations is 4-6, the factor $\gamma$ should be 0.4-0.6, and the radius of inference ($R$) is 250km is recommended.

## Future Work

We notice that there is a big gap between the input data band (about an hour from one band to another). The current DAS system does analysis every six hours, but satellite data come in every hour. Thus, the gap will not be filled until the analysis is done 6 hours later. It would be desirable to do analysis for some components in the DAS system every hour. The proposed method for improving the analysis at all levels is to:  (1) get the time stamp of the data, (2) use some schemes to calculate values at the gap, thereby narrowing or eliminating it, and (3) do an SCM smoothing.
Reference:


Appendix A

A series of Sigmoid functions, with differing values of $\alpha$ (5, 10, 15, and 20), is shown in the following figure. When $\alpha$ is big, it likes a step-function; when $\alpha$ is very small, it likes a straight line. Thirteen was chosen as $\alpha$ in the current study because it appears to fall between the two extremes.
The plots are the original SCM solution (no iteration) with different weighting functions -- Cressman, Sasaki, and Sigmoid. We concentrate on the region where an isolated data point is located. The rest of the map is quite similar for the three functions. Both the Cressman and Sasaki smoothing miss the isolated data point value of 0.006; only the Sigmoid function gives 0.006 at that region. It is because the Sigmoid function pulls more weight in the local area (small $r$).