New method for solving high accuracy surface modeling

CHEN Chuanfa¹, YUE Tianxiang¹, LIU Hongtao²

1. Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China
2. Jinan Institute of Engineering and surveying, Shandong Jinan 250013, China

Abstract: High accuracy surface modelling (HASM) constructed based on the fundamental theorem of surface is more accurate than the classical methods, but the computational speed of HASM is proportional to the third power of the total number of grid cells in the computational domain. In order to decrease the computational cost and improve the accuracy of HASM, this paper employed a modified Gauss-Seidel (MGS) to solve HASM. The fact that MGS is more accurate and faster than GS is proved in terms of theorem. Gauss synthetic surface was employed to comparatively analyze the simulation errors and the computing time of MGS and GS. The numerical tests showed that under the same simulation accuracy, MGS is faster than GS, and the time difference between MGS and GS is approximately proportional to the second power of the total number of grid cells. Under the same outer or inner iterative cycles, MGS is more accurate than GS. The computing time of MGS is proportional to the first power of the total number of grid cells. Compared with the direct methods for solving HASM, MGS greatly shortens the computing time of HASM. SRTM3 (36°—37°N, 107°—108°E) of Dongzhi tableland located in Gansu province was employed as a real word example to validate the accuracy of HASM based on MGS. The example indicated that RMSE of HASM based on MGS is about 2.4, 1.8, 1.3, 2.7 times less than those of KRIGING, IDW, TIN and NEAREST.

Key words: GS iteration, surface simulation, accuracy, test analysis, interpolation

CLC number: TP751.1 Document code: A

1 INTRODUCTION

The computational process of high accuracy surface modelling (HASM) has been mature in terms of theory (Yue & Du, 2006). It can be divided into three processes including the coefficient matrix formulating of Gaussian equation, sampling equation formulating and solving the linear system of HASM. When the computational domain is regular and the sampling equation has one order truncation error, the solving method of HASM equations has an effect on its computational efficiency (Al-Kurdi & Kincaid, 2006, Yue et al., 2007). Former researches indicated that the computational cost of the direct method for solving HASM is proportional to the third power of the total number of grid cells, which seriously influences its wide application (Yue et al., 2007).

The iterative method has been accepted as an efficient method for solving huge linear system (Saad, 2003). One of those methods is Gauss-Seidel (GS), which only needs few save volume, especially for solving huge sparse one (Bramble & Pasciak, 1992; Ujević, 2006). But the convergence speed of GS is very low. So we used a modified GS method (MGS) for solving HASM. Gaussian synthetic surface was employed to compare the efficiency of MGS with GS. In the real world example, the SRTM3 of Dongzhi tableland was used to compare the performance of HASM based on MGS with those of the classical interpolation methods including IDW, TIN, KRIGING and NEAREST with the default parameters performed in ARCGIS 9.3.

2 MGS ITERATION

2.1 MGS formulation

Suppose HASM can be eventually transformed to solve a linear system, which is expressed as $Sx^{k+1}=b^{k}$, where, $S$ is a symmetric definitive matrix, $S \in R^{n \times n}$, $b \in R^{n}$, $k$ is iterative times. The process of solving the linear system of HASM is named outer iteration, while updating the vector $b$ is termed inner iteration. The detailed information about the formulation of MGS can be found in the paper (Yue & Du, 2005).

The component process of solving HASM based on GS is expressed as,
for $k = 0, 1, 2, \cdots$ // iterative times
$y_1 = x_k$ // the kth iteration
for $i = 1, 2, \cdots, n$
\[ y_{i+1} = y_i + \alpha_i e_i \]
end for $i$
\[ x_{k+1} = y_{n+1} \]
end for

where,
\[ e_1 = (1, 0, 0, \cdots, 0)^T, e_2 = (0, 1, 0, \cdots, 0)^T, \cdots, e_n = (0, 0, 0, \cdots, 1)^T \] (2)
\[ \alpha_i = -\frac{p_i}{s_{ij}} p_i = (S_i, y_i) - h_i \] (3)

If $f$ has a second derivative at $y_i$, in terms of Taylor expansion, we have
\[ f(y_{i+1}) = f(y_i + \alpha_i e_i) = f(y_i) + \alpha_i(S_i - b_i, e_i) + \frac{1}{2} \alpha_i^2(s_{ij}, e_i) \] (4)

Based on Eq. (3) and Eq. (4), we can get
\[ f(y_i + \alpha_i e_i) - f(y_i) = -\frac{1}{2} \frac{p_i^2}{s_{ii}} \leq 0 \] (5)
i.e.
\[ f(y_{i+1}) - f(y_i) = -\frac{1}{2} \frac{p_i^2}{s_{ii}} \leq 0 \] (6)

Eq. (6) indicates that GS can reach its convergence.

Let us consider the element
\[ z_{i+1} = y_i + h_i + \gamma_i q_i \] (7)
where, $h_i = \frac{p_i}{s_{ij}} e_i$, $y_i \in R, q_i \in R^n$. In terms of Taylor expansion, we have,
\[ f(y_i + h_i + \gamma_i q_i) = f(y_i) + \frac{1}{2} S(h_i + \gamma_i q_i), h_i + \gamma_i q_i) \]
\[ = f(y_i) + (S_i - b_i, h_i + \gamma_i q_i) + \frac{1}{2} \gamma_i^2(s_{ij}, q_i) \]
\[ = f(y_i) + (S_i - b_i, q_i) + \gamma_i (S_i - b_i, q_i) + \frac{1}{2} \gamma_i^2(s_{ij}, q_i) \]
\[ = f(y_{i+1}) + \gamma_i (S_i - b_i, q_i) + \gamma_i (S_i, h_i) + \frac{1}{2} \gamma_i^2(s_{ij}, q_i) \] (8)

We now define the function,
\[ g(\gamma) = \gamma [(S_i - b_i, q_i) + (S_i, h_i)] + \frac{1}{2} \gamma_i^2(s_{ij}, q_i) \] (9)

Such that, $g'(\gamma) = (S_i - b_i, q_i) + (S_i, h_i)$, $g''(\gamma) = (S_i, q_i) \geq 0$ indicating that $g(\gamma)$ has its minimum. From the equation $g'(\gamma) = 0$, we get
\[ \gamma_i = -\frac{(S_i - b_i, q_i) + (S_i, h_i)}{s_{ij}, q_i} \] (10)

From Eq. (9) and Eq. (10), we have
\[ g(\gamma_i) = \frac{1}{2} \frac{[(S_i - b_i, q_i) + (S_i, h_i)]^2}{(S_i, q_i)} \]
\[ \frac{1}{2} \frac{[(S_i - b_i, q_i) + (S_i, h_i)]^2}{(S_i, q_i)} \] (11)

From Eq. (8) and Eq. (11), we get
\[ f(z_{i+1}) - f(y_i) = f(y_{i+1}) - f(y_i) \]
\[ = 1 - \frac{1}{2} \frac{[(S_i - b_i, q_i) + (S_i, h_i)]^2}{(S_i, q_i)} \] (12)

Eq. (12) indicates that the element $z_{i+1}$ gives better reduction of $f$ than the element $y_{i+1}$. The pre-code of MGS for solving HASM can be expressed as,

for $k = 1, 2, \cdots$ // iterative times
\[ z_i = x_k \] // the kth iteration
for $i = 1, 2, \cdots, n$
\[ z_{i+1} = z_i + h_i + \gamma_i q_i \]
end for $i$
\[ x_{k+1} = z_{n+1} \]
end for $k$

where,
\[ h_i = \frac{p_i}{s_{ij}} e_i, p_i = (S_i, z_i) - b_i \] (13)
\[ \gamma_i = -\frac{(S_i - b_i, q_i) + (S_i, h_i)}{s_{ij}, q_i} \] (14)

From Eq. (1) and Eq. (13), we can get that MGS updates two components of the approximate solution of GS at each iterative cycle.

2.2 $q$ determination

From the theory of MGS, we can see that $q$ is an $n$-order vector. In this paper,
\[ q = e_j, \text{ (i.e.)} \]
\[ s_{ij} = z_{i+1} = h_i + \gamma_i e_j \] (15)

where,
\[ h_i = \frac{p_i}{s_{ij}} e_i, p_i = (S_i, z_i) - b_i \] (16)
\[ \gamma_i = -\frac{(S_i - b_i, q_i) + (S_i, h_i)}{s_{ij}, q_i} \] (17)

where,
\[ \gamma_i = (S_j, z_j) - b_j \] (18)

We can choose $j$ in different way. In this paper, we choose
\[ j = i-1, (i=2, 3, \ldots, n), \]
\[ n = 1, 2, \cdots, n \]
\[ z_i = z_{i-1} + h_{i-1} + \gamma_{i-1} e_{i-2} \]
\[ h_{i-1} = -\frac{p_{i-1}}{s_{i-1,j-1}} e_{i-1}, (j = 1, 2, \cdots, n) \]
(19)

where
\[ \gamma_0 = \gamma_n, e_0 = e_n, e_{-1} = e_{n-1} + h_b = h_b, p_0 = p_n, s_{00} = s_{nn} \] .

From Eq. (18) and Eq. (19), we get that
\[ p_j = (S_j, z_j) - b_j = (S_{j-1}, z_{j-1} + h_{j-1} + \gamma_{j-1} e_{j-2}) - b_{j-1} \]
\[ = (S_{j-1}, z_{j-1} - h_{j-1} - \frac{p_{j-1}}{s_{j-1,j-1}} (S_{j-1}, e_{j-1}) + \gamma_{j-1} (S_{j-1}, e_{j-2}) \]
\[ = p_{j-1} - \frac{p_{j-1}}{s_{j-1,j-1}} (S_{j-1}, e_{j-1}) + \gamma_{j-1} (S_{j-1}, e_{j-2}) \]
\[ = p_{j-1} - \frac{p_{j-1}}{s_{j-1,j-1}} (S_{j-1}, e_{j-1}) + \gamma_{j-1} (S_{j-1}, e_{j-2}) \]
\[ = \gamma_{j-1} (S_{j-1}, e_{j-2}) \]
\[ = \gamma_{j-1} (S_{j-1}, e_{j-2}) \] (20)
From Eq. (17) and Eq. (20), we get,
\[ y_i = -y_{i-1} \frac{S_{i-1,j-2}}{S_{i-1,j-1}} + \bar{p}_{i} \frac{S_{i,j-1}}{S_{i-1,j-2}S_{i,j}} \]  \hspace{1cm} (21)

3 GAUSSIAN SYNTHETIC SURFACE SIMULATION

In this paper, we employed Gaussian synthetic surface (Fig.1) to validate the efficiency of MGS and GS for solving HASM. The formulation of Gaussian synthetic surface is expressed as,
\[ f(x, y) = 3(1-x)^2e^{-x^2-(y+1)^2} - 10(x/5-x^2-y^2)e^{-x^2-y^2} - e^{-(x+1)^2-y^2}/3 \]
The computational domain is \([-3,3] \times [-3,3] , < 8.1062.\]

Fig. 1 Gaussian synthetic surface

The first test is that we fix sampling interval \((m=4)\), inner accuracy tolerance \((\max(f^n_{i,j} - f^{n+1}_{i,j}) < 10^{-7})\) and outer iterative times \((1 \text{ times})\), and change the number of sampling points in the computational domain to compare the computational efficiency of MGS with GS. From Table 1 and Fig.2, we can get that the computational time difference of MGS and GS is proportional to the second power of the total number of grid cells, which indicates that the bigger the computational domain, the more efficiency of MGS is. The relationship between the time difference of MGS and GS and the number of grid points is expressed as,
\[ t = 2.2336 - 3.5292 \times 10^{-7} gn + 6.472 \times 10^{-13} gn^2 \]  \hspace{1cm} (22)
where, \(t\) is the time difference of the two methods, \(gn\) is the number of grid cells. From Table 1, we can get that the inner iteration of MGS is always smaller than that of GS. Although the iterative times difference of the two methods becomes smaller with the number of grid cells increasing, the time difference is still increasing, which is due to the increasing time of each iterative cycle.

The second test is that we fix the sampling interval \((m=4)\), number of sampling points \((1001 \times 1001)\) and inner iterations \((5 \text{ times})\), and change the outer iterations to compare the RMSEs of the two methods. The test results are shown in Table 2. From Table 2, we get that when the inner iteration is small, MGS can obtain higher accuracy than GS. In this test, when the inner iterative times are 320, GS can also reach its convergence.

The third test is that we fix the sampling interval \((m=4)\), the number of sampling points \((1001 \times 1001)\) and inner iterations \((50 \text{ times})\), and change the outer iterations to compare the RMSEs of the two methods. The results are shown in Table 3. From Table 3 we can get that the lesser the outer iterations, the more accuracy of MGS is. With the increasing of outer iterations, both GS and MGS can reach the same convergence.

The fourth test, we fix the sampling interval \((m=4)\), inner and outer iterations \((10 \text{ times} \text{ and} 5 \text{ times})\), and change the number of grid cells to validate the computing time of MGS. The results are shown in Table 4. From Table 4 and Fig.3, we can get that the computing time of MGS is proportional to the first power of the total number of grid cells in the computational domain. The regression relationship between the number of grid cells and the computing time can be expressed as,

<table>
<thead>
<tr>
<th>Inner iterations</th>
<th>GS×10^{-3}</th>
<th>MGS×10^{-3}</th>
<th>GS-MGS×10^{-3}</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6.4517</td>
<td>3.9587</td>
<td>2.4930</td>
</tr>
<tr>
<td>10</td>
<td>3.4696</td>
<td>2.0536</td>
<td>1.4160</td>
</tr>
<tr>
<td>20</td>
<td>1.8455</td>
<td>1.3144</td>
<td>0.5311</td>
</tr>
<tr>
<td>40</td>
<td>1.2548</td>
<td>1.1298</td>
<td>0.1250</td>
</tr>
<tr>
<td>80</td>
<td>1.1217</td>
<td>1.1122</td>
<td>0.0090</td>
</tr>
<tr>
<td>160</td>
<td>1.1120</td>
<td>1.1119</td>
<td>0.0001</td>
</tr>
<tr>
<td>320</td>
<td>1.1119</td>
<td>1.1119</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outer iteration</th>
<th>GS×10^{-5}</th>
<th>MGS×10^{-5}</th>
<th>GS-MGS×10^{-5}</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.5254</td>
<td>1.5178</td>
<td>0.0340</td>
</tr>
<tr>
<td>4</td>
<td>6.6014</td>
<td>6.4578</td>
<td>0.1436</td>
</tr>
<tr>
<td>8</td>
<td>3.4646</td>
<td>3.4426</td>
<td>0.0220</td>
</tr>
<tr>
<td>10</td>
<td>3.0331</td>
<td>3.0203</td>
<td>0.0128</td>
</tr>
<tr>
<td>20</td>
<td>2.1085</td>
<td>2.1065</td>
<td>0.0020</td>
</tr>
<tr>
<td>25</td>
<td>1.8870</td>
<td>1.8865</td>
<td>0.0005</td>
</tr>
<tr>
<td>28</td>
<td>1.7848</td>
<td>1.7848</td>
<td>0</td>
</tr>
</tbody>
</table>
simulate 3201 (Song & Yue, 2009). In this paper, we only used 208.63 seconds to as that of HASM.

In this study area, the number of grid cells is about 1601×101, which is sufficient to adequately resolve the resolution, which can save much space and computational cost (Piquet & Vasseur, 1998; Liu, 1995). So the future effort is toward developing an adaptive method of HASM.

Firstly, we proved the efficiency of MGS in terms of theory; secondly we compared the computational accuracy of MGS with that of GS based on Gaussian synthetic surface. The results indicated MGS is more accurate than GS. Dongzhi tableland in Gansu province as a real world example showed that HASM based on MGS is more accurate than the classical interpolation methods including IDW, TIN, KRIGING and NEAREST.

The pre-smoother and post-soother of classical Multi-grid method is GS. In this paper, we proved MGS is more efficient than GS, so MGS can take place of GS as a smoother. HASM has much potential in computing speed improvement. An adaptive method adapts the finite-difference mesh to place more grid points in regions where high resolution is needed (Berger, 1989), while using fewer grid points in regions where a coarser mesh is sufficient to adequately resolve the resolution, which can save much space and computational cost (Piquet & Vasseur, 1998; Liu, 1995). So the future effort is toward developing an adaptive method of HASM.

REFERENCES


4 REAL WORLD EXAMPLE

Dongzhi tableland (36°—37°N, 107°—108°E) was employed as a real world example to compare the performance of HASM based on MGS with those of the classical interpolation methods including IDW, TIN, KRIGING and NEAREST with the default parameters performed in ARCGIS 9.3. Dongzhi tableland was located in Gansu province, in the middle of loess plateau, north of jin river, south of malian river. The Dongzhi tableland was incised to be a fragmented landform. The complex landform is very suitable for DEM construction test. The source of the data is from the SRTM3 with the resolution of 90 m. In this study area, the number of grid cells is about 1201×1201, half of which were randomly selected for DEM construction, the others for DEM accuracy validation.

The simulation results of the DEM are shown in Table 5. From Table 5, we can obtain that the RMSEs of KRIGING, IDW, TIN, NEAREST are 2.4, 1.8, 1.3 and 2.7 times as much as that of HASM.

Table 4 MGS CPU time under different number of grid cells

<table>
<thead>
<tr>
<th>Number of grid cells</th>
<th>CPU time/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>101×101</td>
<td>0.289764</td>
</tr>
<tr>
<td>201×201</td>
<td>0.805835</td>
</tr>
<tr>
<td>401×401</td>
<td>3.253388</td>
</tr>
<tr>
<td>801×801</td>
<td>13.038350</td>
</tr>
<tr>
<td>1601×1601</td>
<td>52.220574</td>
</tr>
<tr>
<td>3201×3201</td>
<td>208.628482</td>
</tr>
</tbody>
</table>

\[ t = 2.036 \times 10^{-2} g_n + 0.010 (R^2 = 1) \]  (23)

Where, \( t \) is the computing time, \( g_n \) is the number of grid cells.

In former researches, the sources of computing time of the classical HASM were mainly from the differential equation simulation, inverse matrix computation, matrix multiplication, and linear system solution (Yue et al., 2004). The DEM construction in Dafosi Shaanxi province indicated that the computing time is about 10 hours to simulate 1500×1500 grid cells with the classical HASM (Song & Yue, 2009). In this paper, we only used 208.63 seconds to simulate 3201×3201 grid cells with HASM based on MGS. The computing time of MGS decreases two order of magnitude compared with the direct methods for solving HASM.

Fig. 3 MGS regression curve of computation time against total number of grid cells

Table 5 RMSE comparison among different methods

<table>
<thead>
<tr>
<th>Method</th>
<th>HASM</th>
<th>KRIGING</th>
<th>IDW</th>
<th>TIN</th>
<th>NEAREST</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE m</td>
<td>9.8</td>
<td>24.0</td>
<td>17.2</td>
<td>12.9</td>
<td>26.8</td>
</tr>
<tr>
<td>Ratio</td>
<td>1.0</td>
<td>2.4</td>
<td>1.8</td>
<td>1.3</td>
<td>2.7</td>
</tr>
</tbody>
</table>

5 CONCLUSIONS

Former researches indicated that the computational cost of the classical HASM seriously influences its widespread application. In this paper, MGS was employed to solve HASM. Firstly, we proved the efficiency of MGS in terms of theory; secondly we compared the computational accuracy of MGS with that of GS based on Gaussian synthetic surface. The results indicated MGS is more accurate than GS. Dongzhi tableland in Gansu province as a real world example showed that HASM based on MGS is more accurate than the classical interpolation methods including IDW, TIN, KRIGING and NEAREST.

The pre-smoother and post-soother of classical Multi-grid method is GS. In this paper, we proved MGS is more efficient than GS, so MGS can take place of GS as a smoother. HASM has much potential in computing speed improvement. An adaptive method adapts the finite-difference mesh to place more grid points in regions where high resolution is needed (Berger, 1989), while using fewer grid points in regions where a coarser mesh is sufficient to adequately resolve the resolution, which can save much space and computational cost (Piquet & Vasseur, 1998; Liu, 1995). So the future effort is toward developing an adaptive method of HASM.
Gauss-Seidel

1

(1) . (HASM) . . . (2006); (2010a) . (Al-Kurdi & Kincaid, 2006; , 2007)
, (Saad, 2003), , Gauss-Seidel(GS) . . .
. GS (Modified Gauss-Seidel, MGS) . . . HASM .

2

2.1

HASM . . . (, 2010b) . . . . . (, 2010b), . . . .
. . . . . . . . . . . . . . .

for $k = 0,1,2,\ldots$ // 表示迭代次数

$y_1 = x_k$ // 第k次迭代

for $i = 1,2,\ldots,n$

$y_{i+1} = y_i + \alpha_i e_i$
end for $i$
\[ x_{k+1} = y_{n+1} \]

end for \( k \)

\[ e_1 = (1, 0, 0, \ldots, 0)^T, e_2 = (0, 1, 0, \ldots, 0)^T, \]
\[ \ldots, e_n = (0, 0, 0, \ldots, 1)^T \]

\[ \alpha_i = -\frac{p_i}{s_{ii}}, p_i = (S_i, y_i) - b_i \]

\[ f(y_{i+1}) = f(y_i + \alpha_i e_i) = f(y_i) + \alpha_i (S y_i - b, e_i) + \frac{1}{2} \alpha_i^2 (S e_i, e_i) = f(y_i) + \alpha_i p_i + \frac{1}{2} \alpha_i^2 s_{ii} \]

\[ f(y_{i+1}) - f(y_i) = -\frac{1}{2} \frac{p_i^2}{s_{ii}} \leq 0 \]

\[ z_{i+1} = y_i + h_i + \gamma_i g_i \]

\[ h_i = -\frac{p_i}{s_{ii}}, \gamma_i \in R, g_i \in R^n \]

\[ g(y_i) = \gamma_i [(S y_i - b, q_i) + (S q_i, h_i)] + \frac{1}{2} \gamma_i^2 (S q_i, q_i) \]

\[ g'(y_i) = (S y_i - b, q_i) + (S q_i, h_i) + \gamma_i (S q_i, q_i), \quad g''(y_i) = (S q_i, q_i) \geq 0 \]

\[ \gamma_i = \frac{(S y_i - b, q_i) + (S q_i, h_i)}{(S q_i, q_i)} \]

\[ f(z_{i+1}) - f(y_i) = -\frac{1}{2} \frac{p_i^2}{s_{ii}} \leq 0 \]

\[ z_{i+1} = y_i + h_i + \gamma_i g_i \]

\[ \gamma_i = \frac{(S y_i - b, q_i) + (S q_i, h_i)}{(S q_i, q_i)} \]

\[ f(z_{i+1}) - f(y_i) = -\frac{1}{2} \frac{p_i^2}{s_{ii}} \leq 0 \]

\[ z_{i+1} = y_i + h_i + \gamma_i g_i \]

\[ h_i = -\frac{p_i}{s_{ii}}, \quad \overline{p}_i = (S_i, z_i) - b_i, \]

\[ \gamma_i = \frac{(S y_i - b, q_i) + (S q_i, h_i)}{(S q_i, q_i)} \]

\[ z_{i+1} = y_i + h_i + \gamma_i g_i \]

\[ h_i = -\frac{p_i}{s_{ii}}, \quad \overline{s}_j = (S_j, z_j) - b_j \]

\[ \gamma_j = -\frac{p_j}{s_{jj}}, \quad \overline{p}_j = (S_j, z_j) - b_j \]

\[ \gamma_j = -\frac{p_j}{s_{jj}}, \quad \overline{p}_j = (S_j, z_j) - b_j \]
3, ..., n; j = n, (i = 1)

\[ z_i = z_{i-1} + h_{i-1} + \gamma_i e_{i-1}, \]

\[ h_{i-1} = -\frac{\overline{p}_{i-1} - e_{i-1}}{S_{i-1,i-1}}, (i = 1, 2, ..., n) \]  (19)

\[ \gamma_0 = \gamma_{a} e_0 = e_0, e_{-1} = e_{n-1}, h_0 = h_n, \overline{p}_0 = \overline{p}_n, S_{00} = S_{nn} \]

\[ \overline{p}_j = (S_j, z_i - h_j = (S_{i-1}, z_{i-1} + h_{i-1} + \gamma_{i-1} e_{i-2}) - h_{i-1} \]

\[ = (S_{i-1}, z_{i-1}) - h_{i-1} - \frac{\overline{p}_{i-1}}{S_{i-1,i-1}} (S_{i-1}, e_{i-1}) + \gamma_{i-1} (S_{i-1}, e_{i-2}) \]

\[ = \frac{\overline{p}_{i-1}}{S_{i-1,i-1}} S_{i-1,i-1} - \gamma_{i-1} S_{i-1,i-1} = \gamma_{i-1} S_{i-1,i-1} \]  (20)

\[ \gamma_i = -\gamma_{i-1} \frac{S_{i-1,i-2}}{S_{i-1,i-1}} + \frac{\overline{p}_i}{S_{i-1,i-1}} S_{i-1,i-1} \]  (21)

\[ f(x, y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10(x/5-x^2-y^2) e^{-x^2-y^2} - e^{-(x+1)^2-(y+3)^2}/3 \]

\[ [-3.3, 3.3]^2 \]

\[ f(x, y) < 8.1062 \]  (m = 4)

\[ \max(f_{i,j}^m - f_{i,j}^{m+1}) < 10^{-7} \]

<table>
<thead>
<tr>
<th>( g )</th>
<th>( \text{CPU} )</th>
<th>( \text{GS} )</th>
<th>( \text{MGS} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>101 x 101</td>
<td>0.3997</td>
<td>0.3952</td>
<td>0.0045</td>
</tr>
<tr>
<td>301 x 301</td>
<td>3.4381</td>
<td>3.1122</td>
<td>0.3259</td>
</tr>
<tr>
<td>501 x 501</td>
<td>9.5758</td>
<td>8.5352</td>
<td>1.0406</td>
</tr>
<tr>
<td>1001 x 1001</td>
<td>38.5613</td>
<td>34.0659</td>
<td>4.4955</td>
</tr>
<tr>
<td>2001 x 2001</td>
<td>157.3256</td>
<td>136.6111</td>
<td>20.7145</td>
</tr>
<tr>
<td>3001 x 3001</td>
<td>355.0508</td>
<td>310.8424</td>
<td>44.2083</td>
</tr>
<tr>
<td>4001 x 4001</td>
<td>984.5033</td>
<td>820.0605</td>
<td>164.4428</td>
</tr>
</tbody>
</table>

Table 1: MGS 和 GS 计算效率比较
4

不同外迭代次数, MGS 和 GS 中误差比较

<table>
<thead>
<tr>
<th>外迭代次数</th>
<th>GS×10⁻⁵</th>
<th>MGS×10⁻⁵</th>
<th>GS-MGS×10⁻⁵</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>15.524</td>
<td>15.178</td>
<td>0.3460</td>
</tr>
<tr>
<td>4</td>
<td>6.6014</td>
<td>6.4578</td>
<td>0.1436</td>
</tr>
<tr>
<td>8</td>
<td>3.4646</td>
<td>3.4246</td>
<td>0.0200</td>
</tr>
<tr>
<td>10</td>
<td>3.0331</td>
<td>3.0203</td>
<td>0.0128</td>
</tr>
<tr>
<td>20</td>
<td>2.1085</td>
<td>2.1065</td>
<td>0.0020</td>
</tr>
<tr>
<td>25</td>
<td>1.8870</td>
<td>1.8865</td>
<td>0.0005</td>
</tr>
<tr>
<td>28</td>
<td>1.7848</td>
<td>1.7848</td>
<td>0</td>
</tr>
</tbody>
</table>

表 4 MGS 计算时间与模拟区域网格数比较

<table>
<thead>
<tr>
<th>网格数</th>
<th>CPU  (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>101×101</td>
<td>0.289764</td>
</tr>
<tr>
<td>201×201</td>
<td>0.805835</td>
</tr>
<tr>
<td>401×401</td>
<td>3.253838</td>
</tr>
<tr>
<td>801×801</td>
<td>13.038501</td>
</tr>
<tr>
<td>1601×1601</td>
<td>52.220574</td>
</tr>
<tr>
<td>3201×3201</td>
<td>208.628482</td>
</tr>
</tbody>
</table>

5
REFERENCES


Yue T X and Du Z P. 2006. Comparison analysis between HASM and the classical methods. Progress in Natural Science, 16(8): 986—991


附中文参考文献

SRTM (Brandt, 1977) 遥感学报, 2010, 14(4)