ADAPTIVE CROSS APPROXIMATION FOR COMPRESSING THE JACOBIAN MATRIX IN THE GAUSS-NEWTON INVERSION

Maokun Li, Aria Abubakar and Tarek M. Habashy

Schlumberger-Doll Research, Cambridge, MA, USA

1. INTRODUCTION

Among offshore hydrocarbon exploration technologies, the controlled source electromagnetic (CSEM) method have gained a lot of interest in both academia and industry because of its ability to detect hydrocarbon reservoirs [1]. In order to maximally extract information from the data, a full nonlinear inversion approach is employed [2]. In such an approach, the investigation domain is subdivided into pixels and by using an optimization process, the conductivity distribution in the domain can be generated and hence the location, the shape and the conductivity of the reservoir can be inferred. Most of these nonlinear methods use iterative schemes where the conductivity distribution is updated in each iteration based on a search direction computed from a gradient of the cost function. Therefore, in these derivative-based approaches the Jacobian matrix plays a key role. The elements of this matrix are the derivative of the simulated data with respect to the pixel conductivities. Its size is equal to the number of measurement data times the number of unknown pixels. In the CSEM data inversion, the size of the data set and the inversion region can be very large. Hence, the storage of the Jacobian matrix requires a huge amount of memory. This is one of the bottlenecks of using gradient-type nonlinear inversion approaches. Moreover, because the Jacobian matrix is a dense matrix, the arithmetic operation of a matrix-vector multiplication can be very expensive as the size of the Jacobian matrix increases. To reduce the size of the Jacobian matrix, we usually invert a subset of the data at the risk of missing important data points. An alternative way is to compress the Jacobian matrix based on the fact that the electromagnetic field has a limited spatial bandwidth. We can either use physics-based techniques that rely on the field kernel or use pure numerical methods that tend to be very expensive to compute. In this work, we use the adaptive cross approximation (ACA) technique introduced by Bebendorf [3] to compress the Jacobian matrix. The ACA technique converts the Jacobian matrix into two smaller rectangular matrices. This approach reduces both memory usage and CPU time of the Gauss-Newton inversion approach as well as stabilizes the inversion process. To demonstrate the Gauss-Newton inversion using the compressed Jacobian matrix, we employ inversion examples in two-and-half dimensional (2.5D) geometries.

2. FORMULATION

Following the same notation as in [2], we write the cost function with multiplicative regularization as follows:

$$\Phi_n(\mathbf{m}) = \phi^d(\mathbf{m}) \times \phi_n^m(\mathbf{m}) \,, \tag{1}$$

where ϕ^d is the data misfit cost function given by

$$\phi^d(\mathbf{m}) = \frac{1}{2K}\sum_{k=1}^{K}\frac{\sum_{i=1}^{I_k}\sum_{j=1}^{J_{i,k}}|W_{i,j,k}[d_{i,j,k}-s_{i,j,k}(\mathbf{m})]|^2}{\sum_{i=1}^{I_k}\sum_{j=1}^{J_{i,k}}|W_{i,j,k}d_{i,j,k}|^2} = \frac{1}{2}\left\|\mathbf{W}_d\cdot[\mathbf{d}-\mathbf{s}(\mathbf{m})]\right\|^2 , \tag{2}$$

where K is the number of frequencies, I_k is the number of transmitters at frequency k and $J_{i,k}$ is the number of receivers for transmitter i and at frequency k. The diagonal matrix $W_{i,j,k}$ is a data weighting matrix. $\phi_n^m(\mathbf{m})$ is the regularization cost function at the n-th iteration as given in [2]. We employ a Gauss-Newton minimization approach to solve Eq. 1 for the pixel conductivity values represented by a vector of the model parameter $\mathbf{m}$. In each iteration n, a linearized equation is solved to obtain the step vector $\mathbf{p}_n$:

$$\mathbf{H}_n\cdot\mathbf{p}_n = -\mathbf{g}_n, \tag{3}$$

where $\mathbf{H}_n$ is the Hessian matrix approximated as follows:

$$\mathbf{H}_n \approx \mathbf{J}_n^T\cdot\mathbf{W}_d^T\cdot\mathbf{W}_d\cdot\mathbf{J}_n + \phi^d(\mathbf{m}_n)\mathbf{L}(\mathbf{m}_n)\,. \tag{4}$$

In Eq. 4, $\mathbf{J}_n$ is the Jacobian matrix and $\mathbf{L}$ is the second derivative of ϕ_n^m with respect to the model parameter $\mathbf{m}$. The gradient vector $\mathbf{g}_n$ is given by

$$\mathbf{g}_n = -\mathbf{J}_n^T\cdot\mathbf{W}_d^T\cdot\mathbf{W}_d\cdot[\mathbf{d}-\mathbf{s}(\mathbf{m}_n)] + \phi_n^d(\mathbf{m}_n)\,\mathbf{L}(\mathbf{m}_n)\cdot\mathbf{m}_n. \tag{5}$$

In each iteration, the unknown model parameter $\mathbf{m}$ is updated using $\mathbf{p}_n$. The iterative process stops after some error criteria are satisfied. The Hessian matrix is a dense square matrix. Its dimension is equal to the number of pixels N_p that can easily exceed thousands in 2D geometries and millions in 3D geometries. Therefore, Eq. 3 is usually solved using an iterative solver. Since the Hessian matrix in Eq. 4 is positive definite, we employ a conjugate gradient least-square (CGLS) iterative scheme. The computational complexity of this CGLS solver is $O(MN_p^2)$ where M is the number of CGLS iterations. Since the number of measurement data N_m is usually much less than the number of unknown parameters N_p, we do not explicitly construct the Hessian matrix $\mathbf{H}$. Instead we store the Jacobian matrix $\mathbf{J}$ and construct routines to compute $\mathbf{J}\cdot\mathbf{v}$ and $\mathbf{J}^T\cdot\mathbf{u}$. The memory usage to store the Jacobian matrix is $O(N_mN_p)$ and the computational complexity of the CGLS solver using the Jacobian matrix-vector multiplication is $O(M(2N_mN_p))$. Note that the operations in the CGLS routine that explicitly uses the Jacobian matrix is much smaller than the one using the Hessian matrix when $N_m \ll N_p$. Though it is usually more efficient to use $\mathbf{J}$ instead of $\mathbf{H}$, the size of $\mathbf{J}$ can still be large as the number of measurement data points and/or the number of unknowns increases. For example in a CSEM survey, there could be 100 receivers and 1000 transmitters, and each receiver world record all six components of the electromagnetic fields at 5 frequencies. The total number of measurement data is around 3 million. Gridding the domain into 1000×100 cells, and for double precision computation (8 bytes per real number), the memory usage for $\mathbf{J}$ is about 9 TB. This is still beyond the capacity of most current computers. Therefore, we apply the ACA to reduce the memory usage and the cost of the matrix-vector multiplication in the CGLS iterative process.

The adaptive cross approximation was first introduced by Bebendorf [3]. It can be used to convert a dense

Jacobian matrix $\mathbf{J}$ into the product of two rectangular matrices, i.e., $\mathbf{J} = \mathbf{U}^T \cdot \mathbf{V}$. If $\mathbf{J}$ is rank-deficient or if we only need to approximate $\mathbf{J}$ to a certain error tolerance ϵ, $\mathbf{U}$ and $\mathbf{V}$ can be much smaller than $\mathbf{J}$. The detailed description of this algorithm can be found in [3]. In the inversion algorithm, we apply ACA compression to the weighted Jacobian matrix at every frequency. Therefore, we can re-write Eqs. 4 and 5 as follows:

$$\mathbf{H}_n = \mathbf{V}^T \cdot \mathbf{U} \cdot \mathbf{U}^T \cdot \mathbf{V} + \phi^d(\mathbf{m}_n)\, \mathbf{L}(\mathbf{m}_n), \tag{6}$$

$$\mathbf{g}_n = -\mathbf{V}^T \cdot \mathbf{U} \cdot \mathbf{W}_d \cdot [\mathbf{d} - \mathbf{s}(\mathbf{m}_n)] + \phi_n^d(\mathbf{m}_n)\mathbf{L}(\mathbf{m}_n) \cdot \mathbf{m}_n. \tag{7}$$

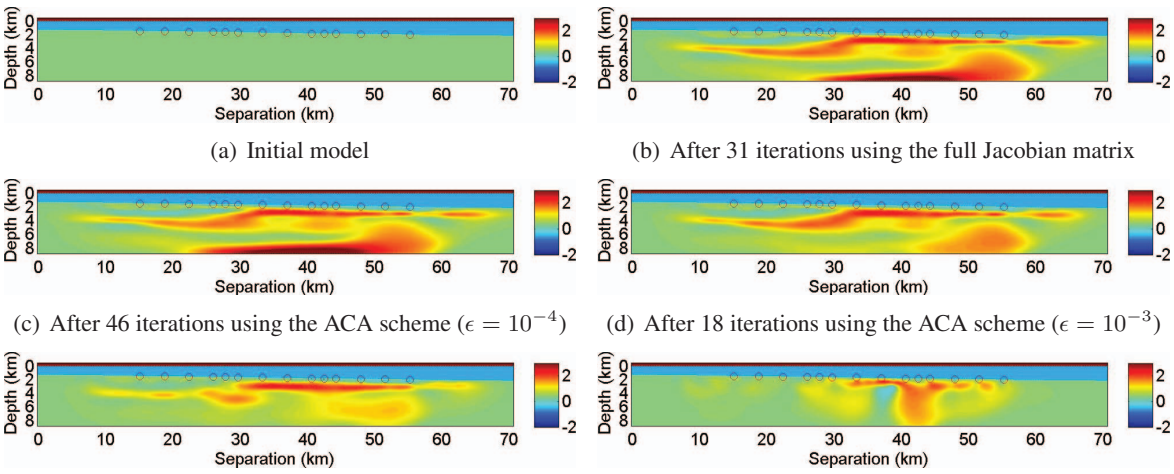
If the compressed matrix $\mathbf{U}$ has N_c rows, the computational complexity of the Jacobian matrix vector multiplication in Eqs. 6 and 7 is $O(N_c(N_m + N_p))$, which can be smaller than the one in Eqs. 4 and 5 if $N_c < N_m N_p/(N_m + N_p)$. Compared with other compression methods, ACA has several advantages: It relies on pure algebraic operations and does not depend on the generating kernel of the matrix. Moreover, in its construction it does not require the full uncompressed matrix (the Jacobian matrix). Therefore it is not necessary to store the full uncompressed matrix in memory. In addition, the compressed Jacobian matrix only contains the dominant singular values of the original Jacobian matrix and omits the smaller ones. This also helps to stabilize the CGLS iterative process.

3. NUMERICAL EXAMPLES

As an example, we invert the FORCE data set acquired in the Norwegian sea [4]. The survey region extends 70 km horizontally and 8.8 km in depth. It is discretized into cells of 250 meters in the x-direction and 50 meters in the z-direction. Hence, the number of conductivity unknowns to be inverted is 42450. There are 324 transmitters and 14 receivers in the survey. The frequencies of operation are 0.125 Hz, 0.25 Hz and 0.375 Hz. The total number of data points is 2928. We use an L_2-norm regularization as described in [2]. The initial model for the inversion is shown in Figure 1(a). The inversion result after 31 iterations without using the ACA scheme is shown in Figure 1(b). The size of the Jacobian matrix $\mathbf{J}$ is 5856×42450 and its memory usage is 1.9 GB.

compression criterion	memory	ratio	$\mathbf{J} \cdot \mathbf{v}$ or $\mathbf{J}^T \cdot \mathbf{v}$-CPU time
0	2 GB	100%	10.38 sec
1×10^{-4}	1.14 GB	57%	5.02 sec
1×10^{-3}	0.822 GB	43%	1.80 sec
1×10^{-2}	0.298 GB	16%	0.87 sec
1×10^{-1}	0.173 GB	9%	0.26 sec

Next, we apply the ACA scheme to compress the Jacobian matrix $\mathbf{J}$. Figure 1(c)-(f) show the reconstructed conductivity distribution using different compression error tolerance ($\epsilon = 10^{-4}, 10^{-3}, 10^{-2}, 0.1$). We observe that the inversion results using the ACA scheme converge to the uncompressed one as the error tolerance is made smaller. The resistive body is well constructed if the compression error is smaller than 10^{-2}. There are some differences in the region deeper than 4 km. This is due to the lack of sensitivity of the CSEM measurement in these deep areas. In addition, we observe that the false high resistive body appearing at depth 8 km is not showing up when using the ACA compressed Jacobian matrix with a compression error larger than 10^{-3}. This is because small singular values in the Jacobian matrix was suppressed at large error tolerance in the ACA compression. Moreover, the inversion using the compressed Jacobian matrix with a compression error 10^{-3} converges faster than using the full Jacobian matrix. Therefore, the ACA scheme may help to further regularize the Gauss-Newton inversion. The table on the left shows the memory usage and CPU time for



(a) Initial model

(b) After 31 iterations using the full Jacobian matrix

(c) After 46 iterations using the ACA scheme ($\epsilon = 10^{-4}$)

(d) After 18 iterations using the ACA scheme ($\epsilon = 10^{-3}$)

(e) After 22 iterations using the ACA scheme ($\epsilon = 10^{-2}$)

(f) After 24 iterations using the ACA scheme ($\epsilon = 10^{-1}$)

Fig. 1. Inversion results.

one matrix-vector multiplication in CGLS. At compression error 10^{-3}, only 822 MB memory is used for storing the Jacobian matrix (43% of the memory of the full Jacobian matrix). Furthermore, the CPU-time for one matrix-vector multiplication reduces significantly from 10.38 seconds to 1.80 seconds.

4. CONCLUSIONS

We improved the efficiency of the regularized Gauss-Newton inversion by using an adaptive cross approximation (ACA) scheme. With the application of the ACA scheme to decompose the Jacobian matrix into smaller matrices, we can reduce both the memory usage and CPU time of the inversion algorithm. In addition, ACA may also help to further regularize the Gauss-Newton inversion process by supressing smaller singular values in the Jacobian matrix. The memory reduction ratio in this approach will increase as more data points are used. This compressed Gauss-Newton inversion approach shows a great potential for handling large data sets.

5. REFERENCES

[1] S. C. Constable, C. S. Cox, and A. D. Chave, "Offshore electromagnetic surveying techniques," in *SEG Technical Program Expanded Abstracts*, pp. 81–82. SEG Expanded Abstracts, 1986.

[2] A. Abubakar, T. M. Habashy, V. L. Druskin, L. Knizhnerman, and D. Alumbaugh, "2.5D forward and inverse modeling for interpreting low-frequency electromagnetic measurements," *Geophysics*, vol. 73, no. 4, pp. F165–F177, 2008.

[3] M. Bebendorf, "Approximation of boundary element matrices," *Numerische Mathematik*, vol. 86, no. 4, pp. 565–589, 2000.

[4] M. Virgilio, L. Masnaghetti, M. Clementi, and M. D. Watts, "Simultaneous joint inversion of MMT and seismic data for sub-basalt exploration on the Atlantic Margin, Norway," in *71th EAGE Conference and Exhibition*, Amsterdam, The Netherlands, June 2009.