DATA-INTENSIVE COMPUTING IN RADIATIVE TRANSFER MODELLING

Efremenko D.S., Loyola D., Doicu A., Trautmann T.

Remote Sensing Technology Institute, German Aerospace Center (DLR),
Oberpfaffenhofen, Wessling, Germany

ABSTRACT

The operational processing of remote sensing data requires high-performance radiative transfer model (RTM) simulations. To date, considerable success has been achieved in dimensionality reduction techniques as well as in heterogeneous multi-CPU/GPU computing for highly intensive parallel computations. We have developed several techniques for accelerating the radiative transfer solver. They include (1) analytical methods which allow to compute set of atmospheric scenarios in one RTM call; (2) dimensionality reduction of the datasets, and (3) GPU-computing using CUDA framework. These techniques provide almost 300x cumulative speed-up for the RTM with respect to the original single-threaded CPU code. In this paper, we analyze the applicability of the proposed methods to a practical problem of total ozone column retrieval from UV-backscatter measurements.

Index Terms— Radiative transfer models, discrete ordinate method, CUDA, heterogeneous computing, dimensionality reduction

1. INTRODUCTION

Massive amounts of spectral information are expected from the new generation of European atmospheric sensors (Sentinel 5 Precursor, Sentinel 4 and Sentinel 5). They impose new challenges to data driven algorithms. In this regard, a fast processing of the data in the UVNS spectral domain is required.

The radiative transfer modelling (RTM) is a critical part in the processing chain from the raw instrumental data (level 0) to the geophysical products (level 2) and is the major performance bottle-neck for the retrieval algorithms. Furthermore, the processing of satellite-measured atmospheric composition data involves many computational loops. These are shown in the following serial-CPU pseudo-code:

```
for each pixel:
  for each wavelength:
    for each cloud_fraction:
      for each geometry:
        call RTE_solver();
```

We have developed several techniques for RTM performance enhancement with particular application to trace gas retrievals. Some of them are used to accelerate the radiative transfer solver itself [1] while others are designed to optimize the loops containing the radiative transfer solver [2, 3]. They are described in the following sections. In this study we also investigate the cumulative the performance enhancement obtained by using all these methods together.

2. ACCELERATION TECHNIQUES FOR THE DISCRETE ORDINATE METHOD

The radiative transfer equation (RTE) in a one-dimensional medium is well-known [4]. The discrete ordinate method (DOM) for solving the RTE is numerically stable for arbitrary optical thicknesses in a multi-layer stratified medium. An important parameter controlling the computational time and the accuracy of computations is the number of streams in the polar hemisphere N_{do}. RTMs are called multi-stream if $N_{\text{do}} \geq 2$, and two-stream if $N_{\text{do}} = 1$. Two-stream RTMs are based on closed-form solutions for the radiance, and are therefore considerably faster than multi-stream models. However, two-stream accuracy is not high enough for practical applications in remote sensing. In contrast, multi-stream models are computationally expensive.

To speed-up the RTM, we implemented several acceleration techniques for the discrete ordinate method including:

1. The computation of the inverse of the eigenvector matrix by first scaling the original matrix to yield a symmetric matrix, and then by calculating the inverse of the symmetric matrix by means of the left eigenvector matrix.

2. The use of the telescoping technique, which consists in the reduction of the linear algebra system to the active layers of the clouds and for azimuthal modes $m \geq 3$.

3. The use of an additional discrete ordinate with zero weight in the direction of the line of sight in order to avoid the post-processing step of the conventional discrete ordinate method (source integration along the line of sight).

The most time consuming part of the radiative transfer solver is the eigenvalue problem which is related to the scattering properties of the atmosphere. The code is designed in such a way, that it aggregates the scenarios with common scattering properties and then computes spectral radiances for a set of solar zenith angles, viewing zenith angles and relative azimuth angles at no additional computational costs.

3. LOOP OPTIMIZATION

3.1. Dimensionality reduction of the input data

To optimize performance over spectral loops, we extended the RTM with principal component analysis (PCA) of optical parameters [5]. This approach has the following features: (a) a two-stream model is used to compute the approximate spectrum; (b) differences (or "correction factors") between the approximate and exact solutions are expressed through a second-degree polynomial in the optical parameters; (c) PCA is used to map the initial data set of optical properties to a lower-dimensional subspace, in which the computation of the correction factors is performed.

Assume, the optical parameters, representing the input parameters of the radiative transfer code, are encapsulated in the vector $\mathbf{x}$. High-dimensional real data often lies on or near a lower-dimensional manifold. The fundamental issues in dimensionality reduction are the modeling of the geometry structure of the manifold, and the design of an appropriate embedding for data projection. For the N-dimensional data set $\{\mathbf{x}_w\}_{w=1}^{W}$, where $\mathbf{x}_w \in R^N$ and W is the number of wavelengths, let $\bar{\mathbf{x}} = (1/W)\sum_{w=1}^{W}\mathbf{x}_w$ be the sample mean of the data. The goal of a linear embedding method is to find an M-dimensional subspace ($M < N < W$) spanned by a set of linear independent vectors $\{\mathbf{a}_k\}_{k=1}^{M}$, such that the centered (mean-removed) data $\mathbf{x}_w - \bar{\mathbf{x}}$ lie mainly on this subspace (manifold),

$$\mathbf{x}_w \approx \bar{\mathbf{x}} + \sum_{k=1}^{M} y_{wk}\mathbf{a}_k = \bar{\mathbf{x}} + \mathbf{A}\mathbf{y}_w, \quad w = 1, \ldots, W, \qquad (1)$$

here $\mathbf{A} = [\mathbf{a}_k]_{k=1}^{M}$ is an $N \times M$ matrix comprising the column vectors $\mathbf{a}_k$, and y_{wk} is the kth component of the vector of parameters $\mathbf{y}_w \in R^M$.

In the operational processor, the radiance I is computed from

$$\ln \frac{I(\mathbf{x}_w)}{I^{\mathrm{TS}}(\mathbf{x}_w)} = f_{\mathrm{I}}(\mathbf{x}_w), \qquad (2)$$

where I^{TS} is the radiance computed by the two-stream model, and f_{I} is a correction factor. Setting

$$\Delta\mathbf{x}_w = \sum_{k=1}^{M} y_{wk}\mathbf{a}_k, \qquad (3)$$

we approximate $f(\mathbf{x}_w)$ by a second-order Taylor expansion around $\bar{\mathbf{x}}$, that is,

$$f(\mathbf{x}_w) \approx f(\bar{\mathbf{x}} + \Delta\mathbf{x}_w) \approx f(\bar{\mathbf{x}}) + \Delta\mathbf{x}_w^T \nabla f(\bar{\mathbf{x}}) + \frac{1}{2}\Delta\mathbf{x}_w^T \nabla^2 f(\bar{\mathbf{x}})\,\Delta\mathbf{x}_w, \qquad (4)$$

where ∇f and $\nabla^2 f$ are the gradient and the Hessian of f, respectively. Using central differences to approximate the first and the second-order directional derivatives gives

$$f(\mathbf{x}_w) \approx f(\bar{\mathbf{x}}) + \frac{1}{2}\sum_{k=1}^{M}\left[f(\bar{\mathbf{x}} + \mathbf{a}_k) - f(\bar{\mathbf{x}} - \mathbf{a}_k)\right] y_{wk} + \frac{1}{2}\sum_{k=1}^{M}\left[f(\bar{\mathbf{x}} + \mathbf{a}_k) - 2f(\bar{\mathbf{x}}) + f(\bar{\mathbf{x}} - \mathbf{a}_k)\right] y_{wk}^2. \qquad (5)$$

From Eq. (5) it is apparent that the computation of the correction factor requires $2M + 1$ calls of the full- and two-stream models. As a result and taking into account that $M \ll W$, we are led to a substantial reduction of the computational time.

A similar approach is used to compute derivatives of the radiance (Jacobians) with respect to atmospheric parameters. Forward-model RTM simulations for total ozone retrieval in the wavelength domain 325–335 nm (Huggins bands) containing 88 spectral points were obtained by calling the multi-stream model with 8 streams per hemisphere only 5 times and the faster two-stream model 93 times. The speed improvement was about 8, with the maximum radiance error smaller than 0.2%.

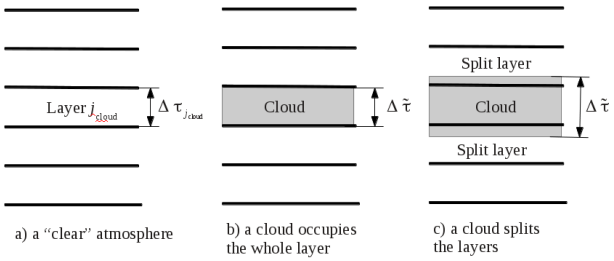

Fig. 1. Cloud position with respect to atmosphere layers.

3.2. Computations under cloudy conditions

In the independent-pixel approximation for cloud-contaminated scenes, radiances are computed as a linear superposition of two solutions for the clear-sky and fully-cloudy scenarios, requiring two RTM calls. We developed two methods based on the re-use of results from clear-sky RTM calculations to speed up corresponding calculations for the cloud-filled scenario [6]. The first approach is numerically exact, in that results from the clear sky computation can be saved in memory and reused for all non-cloudy layers in the second computation involving clouds. The insertion of a cloud layer in a clear sky atmosphere will affect the atmospheric layering scheme. This depends on the cloud-top height and the cloud geometrical thickness, and the possible options are illustrated in Figure 1.

The simplest case involves a cloud with optical thickness $\Delta\tau$ introduced into the layer j_0 which has clear-sky optical depth $\Delta\tau_0$ (Figure 1b). In this case, when solving the clear-sky problem we store the temporary matrices for all layers

$j \neq j_0$, as well as, the source vectors for all layers $j \neq j_0$. When solving the cloudy-sky problem, we use the clear-sky layer equations for all layers $j < j_0$; for the layers $j > j_0$ we take account the change in attenuation of the direct solar beam. If the boundary of the cloud splits a layer, as shown in Figure 1c, then these corrections are applied to clear-sky layers situated above and below the split layers. For the split layers themselves, we must store the solutions to the homogeneous RTE obtained for the original clear-sky calculation.

The second approach is (for the cloudy scenario) to generate a spectral correction applied to the radiation field from a fast two-stream RTM. We propose the following computational formula for the multi-stream solution in a cloudy atmosphere

$$I_{\text{cloud}}(\lambda) \approx I_{\text{clear}}(\lambda) \frac{I^{\text{TS}}_{\text{cloud}}(\lambda)}{I^{\text{TS}}_{\text{clear}}(\lambda)} K(\lambda). \tag{6}$$

Here, $I^{\text{TS}}_{\text{cloud}}(\lambda)$ is the two-stream solution for the cloudy scenario, $I^{\text{TS}}_{\text{clear}}(\lambda)$ is the two-stream solution for clear-sky scenario and $K(\lambda)$ is the correction factor to be determined. Second, applying the dimensionality reduction techniques for computing the multi-stream solution for a clear sky, i.e., $I_{\text{clear}}(\lambda) \approx I^{\text{TS}}_{\text{clear}}(\lambda) f(\lambda)$, Eq. (6) becomes

$$I_{\text{cloud}}(\lambda) \approx I^{\text{TS}}_{\text{clear}} f(\lambda) \frac{I^{\text{TS}}_{\text{cloud}}(\lambda)}{I^{\text{TS}}_{\text{clear}}(\lambda)} K(\lambda). \tag{7}$$

With this approach, the computation of the multi-stream solution for a cloudy sky I_{cloud} requires only one additional call to of the two-stream solution $I^{\text{TS}}_{\text{cloud}}$ for each spectral point.

Correction factors $K(\lambda)$ are pre-computed for various values of the cloud parameters using the following inverted form of Eq. (6)

$$K(\lambda) = \frac{I_{\text{cloud}}(\lambda)}{I_{\text{clear}}(\lambda)} \frac{I^{\text{TS}}_{\text{clear}}(\lambda)}{I^{\text{TS}}_{\text{cloudy}}(\lambda)}. \tag{8}$$

Next , $K(\lambda)$ is interpolated in the spectral domain as

$$K(\lambda) \approx k\left(\lambda - \bar{\lambda}\right) + b + v\sigma_{\mathtt{O}_3}(\lambda) \tag{9}$$

where k, b and v are constants, the values of which are stored in look-up tables, $\bar{\lambda}$ is the mean wavelength for the spectral window (in this case $\bar{\lambda} = 330$ nm), $\sigma_{\mathtt{O}_3}$ is the $\mathtt{O}_3$ absorption cross section at temperature 270 K convolved with the GOME slit function to the 88-point wavelength grid in our 325-335 nm window. Although this method involves some approximations, it still provides radiance accuracy better than 0.2%, with a speed-up factor of approximately 2 compared with time taken for two separate RTM calls.

3.3. GPU-accelerated radiative transfer model

To optimize the loop over ground pixels, we designed a RTM code using the GPU architecture of modern graphical cards. To implement GPUs, the original CPU code has been redesigned using the C-oriented Compute Unified Device Architecture (CUDA) developed by NVIDIA. To reduce the CPU/GPU communication overhead, we exploited the asynchronous data transfer between host and device. To obtain optimal performance, we also used overlapping of CPU and GPU computations by distributing the workload between them.

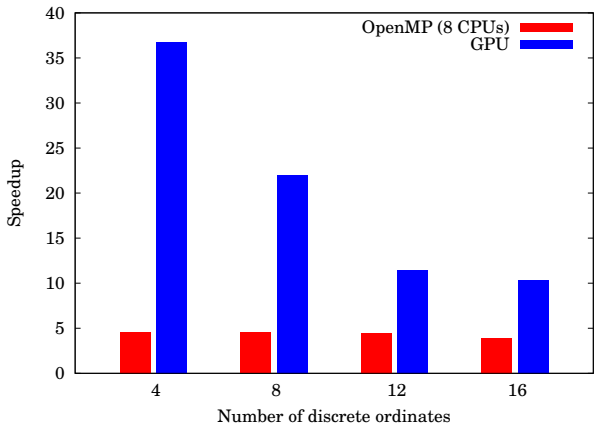

Fig. 2. Speed-up of the GPU-implemented DOM with the matrix operator technique compared to sequential CPU code execution.

Typically, values of $N_{\mathtt{do}} = 4 \div 8$ are chosen for simulations of scattered sunlight in the UV spectral range. The dimensions of matrices involved in the computations are mostly $N_{\mathtt{do}} \times N_{\mathtt{do}}$. Our numerical simulations regarding basic matrix operations evidence that for matrix sizes 8×8 the highest performance is achieved when all arrays required for the RTM solver are placed into registers. Speed-up factors are plotted as functions of the number of discrete ordinates in Figure 2. The speed-up for $N_{\mathtt{do}} = 16$ is less than that for $N_{\mathtt{do}} = 4$. With a low number of discrete ordinates, the kernel consumes a small number of registers, and so a large number of kernels can run simultaneously and the occupancy of GPU is high.

For the algorithm consisting of n parts with corresponding workloads W_i and speedups S_i, the total speedup for the whole algorithm reads as

$$S_{\mathtt{total}} = \left(\sum_{j=1}^{n} \frac{W_j}{S_j} \right)^{-1}. \tag{10}$$

Let's also introduce the "reduced workload" $\bar{W}_i$ as

$$\bar{W}_i = \frac{W_i/S_i}{\sum\limits_{j=1}^{n} \frac{W_j}{S_j}}. \tag{11}$$

Values of workloads, corresponding speedups as well as reduce workloads are given in Table 1 for $N_{\mathtt{do}} = 8$. For presented numbers, the theoretical speedup is $S_{\mathtt{total}} \approx 15$. In

Table 1. Workload of the PCA-based RTM and the corresponding speedup for $N_{\mathtt{do}} = 8$.

	Workload	Speedup	Reduced workload
Multi-stream RTM	50%	22	34.8%
Two-stream RTM	25%	53	7.2%
PCA	20%	6	51%
Rest	5%	$\sim$10	6.12%

our computations, we obtain the speedup factor of $S_{\mathtt{total}} \approx 12$. Note that, PCA has the largest reduced workload. The main part of the PCA is the eigenvalue problem. According to Amdahl's law, the eigenvalue solver is a main limitation factor of the performance. PCA could be implemented on GPU. However, the standard eigenvalue solver from CULA library [7] shows poor performance for small matrices (see benchmarks at www.culatools.com). Moreover, it cannot support the computations in the batched mode.

With GPUs (Tesla K20 with 2496 cores), we achieved a 20x-40x speed-up for the multi-stream RTM, and 50x speed-up for the two-stream RTM, these figures with respect to performance with the original single-threaded CPU codes run on Intel Xeon CPU E5-1620 3.60GHz. The speed-up of the PCA-based RTM is of about 12 times.

4. CUMULATIVE PERFORMANCE ENHANCEMENT

Above mentioned acceleration techniques have been implemented in a common framework. The resulting code has been validated against the codes DISORT [4] and LIDORT [8]. The error imposed by the acceleration techniques is less than 0.1% in the spectral radiances. The code has been applied to the problem of ozone retrieval. The performance enhancement for considered techniques is given in Table 2. The obtained cumulative performance enhancement is of about 300 times which is 85% of a theoretical maximum estimated as a product of speed-up rates of all methods. The performance enhancement excluding GPU computations is of about 25 times. Our analysis shows that the considerable speed-up can be achieved by tuning and optimizing the RTM code to a specific remote sensing problem. It is required to make a lot of substantial changes to the underlying codebase to make it efficient. These changes affect the memory organization, data flows, the generation of appropriate look-up tables, the data compression algorithms and sparse matrix computations.

5. ACKNOWLEDGMENT

Part of this work was supported by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology grant 07 03/893 73/ 5 /2013.

Table 2. The performance enhancement for the acceleration techniques

Acceleration technique	**Speed-up rate**
Dimensionality reduction	8
Parallel computing on GPU	12
Optimization of cloudy scenarios	1.9
Telescoping	1.5
Left eigenvectors	1.3
Cumulative speed-up	320

6. REFERENCES

[1] D. Efremenko, A. Doicu, D. Loyola, and T. Trautmann, "Acceleration techniques for the discrete ordinate method," *J Quant Spectrosc Radiat Transfer*, vol. 114, pp. 73–81, January 2013.

[2] D.S. Efremenko, A. Doicu, D. Loyola, and T. Trautmann, "Optical property dimensionality reduction techniques for accelerated radiative transfer performance: Application to remote sensing total ozone retrievals," *J Quant Spectrosc Radiat Transfer*, vol. 133, pp. 128–135, 2014.

[3] D.S. Efremenko, D.G. Loyola, A. Doicu, and R.J.D. Spurr, "Multi-core-CPU and GPU-accelerated radiative transfer models based on the discrete ordinate method," *Computer Physics Communications*, vol. 185, no. 12, pp. 3079 – 3089, 2014.

[4] K. Stamnes, S.C. Tsay, W. Wiscombe, and K. Jayaweera, "Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media," *Appl Opt*, vol. 12, pp. 2502–2509, 1988.

[5] V. Natraj, R.L. Shia, and Y.L. Yung, "On the use of principal component analysis to speed up radiative transfer calculations," *J Quant Spectrosc Radiat Transfer*, vol. 111, no. 5, pp. 810–816, 2010.

[6] D.S. Efremenko, D. Loyola, R.J.D. Spurr, and A. Doicu, "Acceleration of radiative transfer model calculations for the retrieval of trace gases under cloudy conditions," *J Quant Spectrosc Radiat Transfer*, vol. 135, pp. 58–65, 2014.

[7] J.R. Humphrey, D.K. Price, K.E. Spagnoli, A.L. Paolini, and E.J. Kelmelis, "CULA: hybrid GPU accelerated linear algebra routines," *Proc. SPIE*, vol. 7705, pp. 770502–770502–7, 2010.

[8] R.J.D. Spurr, "LIDORT and VLIDORT. Linearized pseudo-spherical scalar and vector discrete ordinate radiative transfer models for use in remote sensing retrieval problems," in *Light scattering reviews*, A.A. Kokhanovsky, Ed., vol. 3, pp. 229–275. 2008.