

기술보고

## 고성능 파이썬 라이브러리를 이용한 3차원 시간 영역 파동 전파 모델링

조상훈<sup>1)</sup> · 하완수<sup>1)\*</sup>

### 3D Time-Domain Wave Propagation Modeling Using High-Performance Python Libraries

Sang Hoon Jo and Wansoo Ha\*

(Received 3 May 2018; Final version Received 22 June 2018; Accepted 22 June 2018)

**Abstract :** In this study, we compared performance and productivity of C and Python with high-performance libraries using 3D time-domain wave propagation modeling. As a result of numerical modeling, Numba, Cython and C attained speed-ups of 1884, 2049, and 2305 times, respectively, compared with a reference Python implementation. Python programs using Numba and Cython were slower than the C program by 23.3% and 16.2%, respectively. However, the numbers of source code lines for Python programs using Numba and Cython were 45.1% and 40.7% shorter than that of C, respectively. Modeling experiments using Numba and Cython for the SEG/EAGE 3D salt dome model were 22.9% and 13.3% slower than that of the C program. Therefore, if performance difference of 20% is acceptable, it will be more productive to use Python with high-performance libraries instead of static-typing languages such as C.

**Key words :** 3D, Wave propagation modeling, High-performance Python, Productivity

**요약 :** 본 연구에서는 3차원 시간 영역 파동 전파 모델링을 이용하여 고성능 라이브러리를 이용한 파이썬과 C언어의 성능 및 생산성을 비교하였다. 수치 모델링 결과 고성능 파이썬 라이브러리인 Numba 또는 Cython을 사용하는 경우와 C언어를 사용하는 경우 순수 파이썬에 비해 각각 1844배, 2049배, 2305배 빨라졌다. Numba와 Cython 라이브러리를 사용한 모델링의 경우 C언어보다 각각 23.3%, 16.2% 느렸다. 그러나 소스 코드 줄 수는 C언어보다 각각 45.1%, 40.7% 적었다. SEG/EAGE 3차원 암염돔 모델에 적용한 결과, Numba와 Cython을 사용한 경우 C언어에 비해 계산 속도가 각각 22.9%, 13.3% 느렸다. 따라서 20%의 성능 차이를 받아들일 수 있는 경우 C와 같은 정적 자료형 언어 대신 고성능 파이썬 라이브러리를 이용하는 것이 더욱 생산적일 것이다.

**주요어 :** 3차원, 파동 전파 모델링, 고성능 파이썬, 생산성

## 서 론

탄성과 탐사는 고분해능의 지하 영상을 얻는 것이 목적이며, 최근 석유 및 가스 탐사 분야에서는 3차원 탄성파 탐사를 이용하여 복잡한 지질구조를 규명하고 있다. 3차원 완전 파형 역산이나 역시간 구조보정과 같은 탄성파 자료 처리는 2차원 탄성파 자료 처리에 비해 많은 계산 자원을 필요로 하며 그에 따른 시간 소모가 크다.

일반적으로 수치해석 분야에서는 계산 속도가 빠른 C/C++, 포트란과 같은 정적 자료형 프로그래밍 언어를 선호한다. 정적 자료형 언어에서는 컴파일 과정을 통해 실행

파일을 생성하며, 컴파일 시에 자료형을 검사하게 된다. 자료형 불일치 문제가 있을 경우 컴파일 시 이를 발견할 수 있어 프로그램의 안전성을 높일 수 있다. 또한 자료형 검사를 프로그램 실행 시에 반복할 필요가 없으므로 프로그램 실행 속도를 높일 수 있다. 그러나 정적 자료형 프로그래밍 언어의 경우 일반적으로 동적 자료형 프로그래밍 언어에 비해 코드가 길어진다는 단점이 있다.

파이썬, 루비, 펄과 같은 동적 자료형 프로그래밍 언어는 일반적으로 정적 자료형 프로그래밍 언어에 비해 코드 작성이 수월하고, 표현력이 좋아 짧은 코드로 더 많은 작업을 진행할 수 있다는 장점을 지니고 있다(Pyhton, 2018). 많은 경우 별도의 컴파일 과정이 없기 때문에 프로그램 실행 시에 자료형을 검사한다. 특정 변수의 자료형은 변수에 할당된 값으로부터 결정되므로 별도의 변수 선언이 필요 없다. 그로 인해 동일한 기능의 함수를 자료형마다 선언할 필요가 없다. 반면에 예상치 못한 변수로 인한 실행 오류가 발생

1) 부경대학교 에너지자원공학과

\*Corresponding Author(하완수)

E-mail; wansooaha@pknu.ac.kr

Address; Department of Energy Resources Engineering,  
Pukyong National University

할 수 있으며, 프로그램 실행 시 변수의 자료형을 검사하여 자료형에 맞는 연산을 수행하기 때문에 정적 자료형 언어 보다 프로그램 실행 속도가 느리다는 단점이 있다(Beazley, 2009). 이러한 단점에도 불구하고 파이썬의 경우 간단한 사용법, 다양하고 사용이 편리한 고성능 라이브러리 등으로 인해 과학, 공학, 데이터 과학 등 여러 분야에서 사용량이 증가하고 있다(McKinney, 2012; Lanaro, 2013; Ha, 2014; Ha, 2015). 일반적인 프로그래밍에서도 언어의 설명서를 검색한 양으로 순위를 평가하는 PyPL 프로그래밍 언어 순위에서 파이썬 언어는 1위를 기록하고 있다(PyPI, 2018). 소스 코드를 배포한 양으로 순위를 평가하는 Redmonk 프로그래밍 언어 순위에서는 3위를 기록하는 등 파이썬은 현재 가장 주목받는 프로그래밍 언어 중 하나이다(Redmonk, 2018).

본 연구에서는 고성능 파이썬 수치 연산 라이브러리와 C 언어를 이용하여 3차원 시간 영역 파동 전파 모델링 프로그램을 작성하고 성능과 생산성을 비교하였다. 성능은 3번의 반복 실험 후 평균 실행 시간을 이용하여 비교하였다. 이때 컴파일 시간은 제외하였다. 생산성의 비교는 일반적으로 프로그래밍 언어 비교에 사용하는 코드 줄 수 비교 방법을 이용하였다(Prechelt, 2000). 본 연구를 수행하기 위해 대표적인 파이썬 수치 연산 라이브러리인 Numpy, Numba 및 Cython을 이용하여 고성능 파이썬 프로그램을 구현하였다. 파이썬은 버전 3.5.3, Cython은 버전 0.24, Numba는 버전 0.26, Numpy는 버전 1.11.1을 사용하였다. C 프로그램 컴파일을 위해서는 GNU C 컴파일러 버전 4.4.7을 사용하였다. 본 논문에 사용한 프로그램 소스 코드는 결론에 제시한 인터넷 사이트에 공개하였다.

## 고성능 파이썬 라이브러리

파이썬의 계산 효율을 증가시키는 대표적인 고성능 라이브러리로 Numpy, Numba와 Cython이 있다. Numpy는 파이썬에서 배열 연산을 위해 기본적으로 사용되는 라이브러리로, 내부적으로는 C언어로 작성되어 있다. 본 연구에서는 파동 방정식 모델링을 위해 Numba 또는 Cython과 함께 Numpy 라이브러리를 사용하였다.

Numba는 LLVM(LLVM, 2018) 기술을 사용하여 프로그램 실행 시 파이썬 함수를 머신 코드로 실시간 컴파일하고 빠르게 실행하는 라이브러리이다. Numba를 이용하면 함수에 간단한 파이썬 장식자만 추가하여 속도 향상을 얻을 수 있다(Numba, 2018). 본 연구에서는 Numpy 배열을 이용하여 작성한 함수에 “@jit” 장식자 한 줄만 추가하여 코드를 수정하였다.

Cython은 파이썬 코드로부터 C 코드를 생성하여 파이썬

에서 사용할 수 있도록 해주는 라이브러리이다(Cython, 2018). 함수 내에서 사용하는 변수에 자료형 정보를 추가하여 컴파일하는 방식으로, 동적 자료형 언어의 단점을 극복하기 위해 정적 자료형 언어의 기능을 도입한 것이라 볼 수 있다. 본 연구에서는 Cython을 사용하기 위해 파이썬 함수에 자료형 정보를 추가하고 파이썬에서 메모리 오류를 찾기 위해 자동으로 수행하는 배열 경계 검사를 생략하여 속도를 향상시켰다.

## 시간 영역 파동 전파 모델링 알고리즘

유한 차분식을 이용하여 3차원 시간 영역 음향파 파동 방정식을 계산하였다(Choi, 2010; Kreyszig, 2011; Min et al., 2016). 3차원 시간 영역 음향파 파동 방정식은 식 (1)과 같다.

$$\frac{1}{v(\mathbf{x})^2} \frac{\partial^2 u(\mathbf{x}, t)}{\partial t^2} = \nabla^2 u(\mathbf{x}, t) + f(\mathbf{x}, t) \quad (1)$$

위 식에서  $u(\mathbf{x}, t)$ 는 압력 파동장,  $v(\mathbf{x})$ 는  $P$ 파 전파 속도,  $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ 는 라플라시안 연산자,  $f(\mathbf{x}, t)$ 는 송신원이다. 식 (1)에 다음과 같이 시간에 대해 2차, 공간에 대해 2차 또는 8차 유한 차분식을 적용하였다.

$$\frac{1}{v(\mathbf{x})^2} \frac{u(\mathbf{x}, t + \Delta t) - 2u(\mathbf{x}, t) + u(\mathbf{x}, t - \Delta t))}{\Delta t^2} = L\{u(\mathbf{x}, t)\} + f(\mathbf{x}, t) \quad (2)$$

위 식에서  $\Delta t$ 는 시간 차분 간격,  $L$ 은 유한 차분 라플라시안 연산자로, 세부 식은 유한 차분식 차수에 따라 달라진다. 식 (2)는 양함수법을 이용한 편미분 방정식 풀이를 위해 다음과 같이 정리할 수 있다.

$$u(\mathbf{x}, t + \Delta t) = v(\mathbf{x})^2 \Delta t^2 [L\{u(\mathbf{x}, t)\} + f(\mathbf{x}, t)] + 2u(\mathbf{x}, t) - u(\mathbf{x}, t - \Delta t) \quad (3)$$

시간 영역 파동 전파 모델링에서는 일반적으로 초기화를 실시한 후, 시간에 따라 식 (3)의 유한 차분식을 이용하여 공간 차분식을 계산하고 경계 조건을 적용한 후 파동장을 갱신해 나간다(Fig. 1). 본 연구에서는 위와 같은 알고리즘을 사용하여 3차원 시간 영역 파동 전파 모델링을 구현하였다. 우선 Numpy 라이브러리만 이용하여 파이썬 함수를 구현한 후 이 함수를 바탕으로 Numba와 Cython 라이브러리를 적용한 함수를 생성하였다.

```

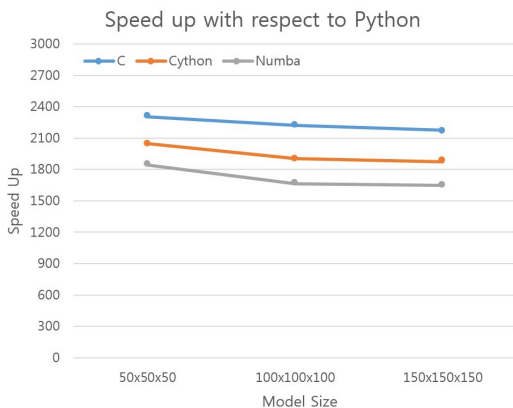
Initialize
For each time
    Solve the finite-difference equation in eq. (3)
    Apply boundary conditions
    Update wavefields
    Write output
End For
    
```

**Fig. 1.** An explicit wave propagation modeling algorithm.

**수치예제**

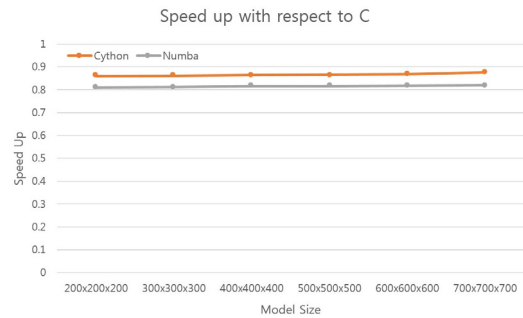
먼저 3차원 시간 영역 파동 전파 모델링 프로그램 구현 시 파이썬과 Numpy 배열만 사용한 경우, Numba를 사용한 경우, Cython을 사용한 경우, C언어를 사용한 경우로 나누어 각 경우의 성능을 비교하였다. P파 속도 모델은 반무한 매질이라고 가정하고 2 km/s의 상속도 모델을 적용하였다. 경계 조건으로는 투과 경계 조건을 적용하였다(Keys, 1985). 모델링은 1 ms 간격으로 총 3초간 수행하였다. 먼저 Numpy만 사용한 파이썬 함수와 비교하여 얼마나 속도가 향상되었는지 비교하였다. 이 예제에서는 공간에 대한 2차 유한 차분식을 사용하였으며, 모델 크기를 50x50x50부터 50간격으로 150 × 150 × 150까지 크기를 늘려가며 계산하였다. 계산 시간은 3번의 반복 실행 평균 시간으로 정의하였다. 이에 따른 결과는 Fig. 2에 나타내었다. 고성능 라이브러리인 Numba와 Cython을 사용하면 각각 최대 1,844배, 2,049배의 속도 향상을 얻을 수 있었다. C언어를 사용한 경우에는 최대 2,305배 빨랐다.

다음으로 Numpy만 사용한 경우를 제외하고, 모델 크기를 200 × 200 × 200부터 100 간격으로 700 × 700 × 700까지 변경시켜가며 1 ms 간격으로 총 3 초간 모델링을 수행하

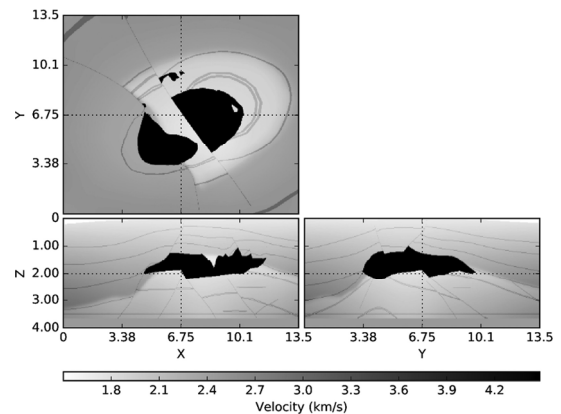


**Fig. 2.** Speed-ups of Python high performance libraries and C with respect to a reference Python program.

여 성능 및 생산성을 비교하였다. 성능은 계산 시간으로 비교하였으며 생산성은 코드 줄 수를 통해 비교하였다. P파 속도 모델은 반무한 매질이라고 가정하고 2 km/s 상속도 모델을 적용하였으며, 공간에 대해 8차 유한 차분식을 사용하였다. 유한 차분식의 차수가 증가하게 되면 계산해야 되는 격자점이 증가하게 되므로 모델링 시간 및 정확도가 증가하게 된다. 3차원 시간 영역 파동 전파 모델링을 이용하여 비교한 결과를 Fig. 3에 C 언어 대비 속도 향상 정도로 나타내었다. 고성능 파이썬 라이브러리인 Numba와 Cython을 사용한 경우에는 각각 C 프로그램에 비해 계산 속도가 약 23.3%, 16.2% 느렸다. 코드 줄 수를 비교한 결과, Numpy만 사용한 파이썬의 경우 모델링 프로그램을 작성하는데 총 92줄을 소모하였으며, Numba는 import문과 장식자만 추가하면 되므로 총 100줄, Cython은 총 103줄을 소모하였다. C언어의 경우 동일한 모델링 코드를 작성하는데 총 145줄을 소모하였다. 단순히 코드 줄 수를 이용하여 생산성을 비교해본 결과 Numba와 Cython을 사용하면 C언어에 비해



**Fig. 3.** Speed-ups of Python high performance libraries with respect to the C program.



**Fig. 4.** SEG/EAGE 3D salt-dome velocity model (Aminzadeh et al., 1994).

각각 생산성이 약 45.1%, 40.7% 높았다. 이를 통해 고성능 파이썬 라이브러리를 사용하면 C언어보다 계산 속도가 느리지만 비교할만한 성능을 보여주며 생산성은 더 높은 것을 확인할 수 있었다. 따라서 약 20%의 성능 차이가 중요하지 않은 경우라면 생산성이 높고 코드 작성 및 유지 보수가 수월한 파이썬을 사용하는 것이 유리하다고 할 수 있다.

다음으로 실제적 모델에서의 적용성을 알아보기 위해 SEG/EAGE 3차원 압연 모델(Aminzadeh *et al.*, 1994)을 이용하여 모델링 성능을 비교하였다(Fig. 4). 3차원 압연 속도 모델의 격자 크기는  $676 \times 676 \times 201$ 이고, 2 ms 간격

으로 총 6 초간 모델링을 수행하였다. C 프로그램을 이용하여 모델링하는데 6,543초, Cython은 7,417초, Numba는 8,044초가 소모되었다. Numba와 Cython을 이용한 경우 C에 비해 각각 22.9%, 13.3% 계산 속도가 느렸다. Fig. 5는 C언어, Numba, Cython 프로그램을 이용하여 시간에 따라 3차원 압연 모델을 모델링한 후 추출된 표면 탄성파기록을 보여준다. Fig. 6과 Fig. 7은 Numba, Cython, C언어를 통해 얻어진 모델링 결과(Fig. 5)에서 각각 4 km, 10 km 지점에서 추출한 트레이스를 이용하여 정확도를 비교한 결과를 보여준다. 트레이스를 비교한 결과 파동장이 거의 일치

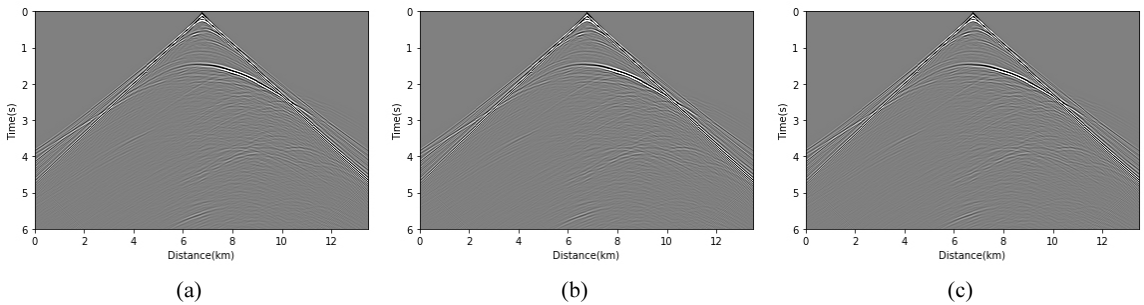


Fig. 5. Surface seismograms ( $y = 6$  km) from the SEG/EAGE 3D salt-dome velocity model using (a) C, (b) Numba and (c) Cython.

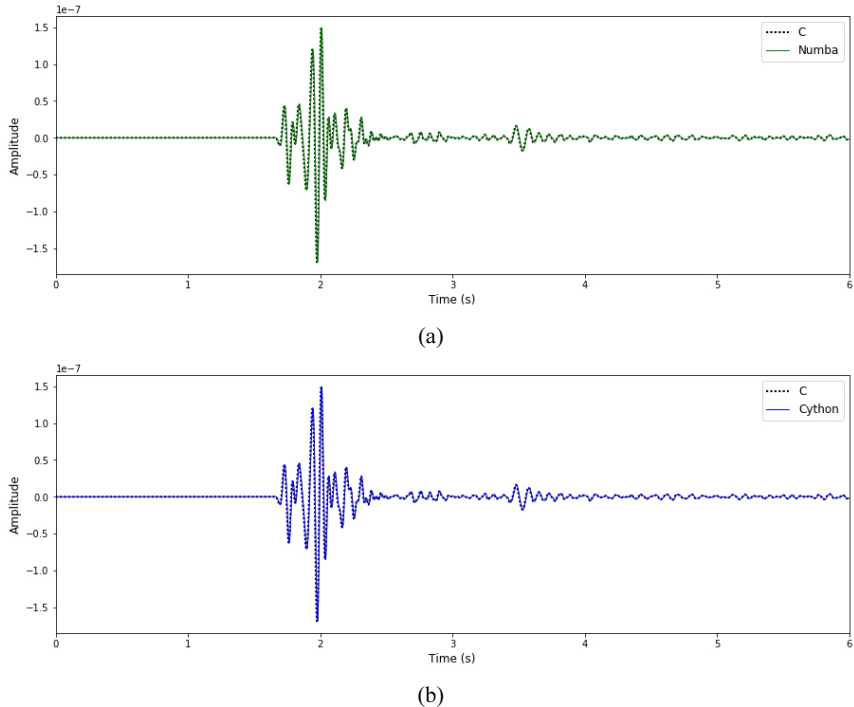
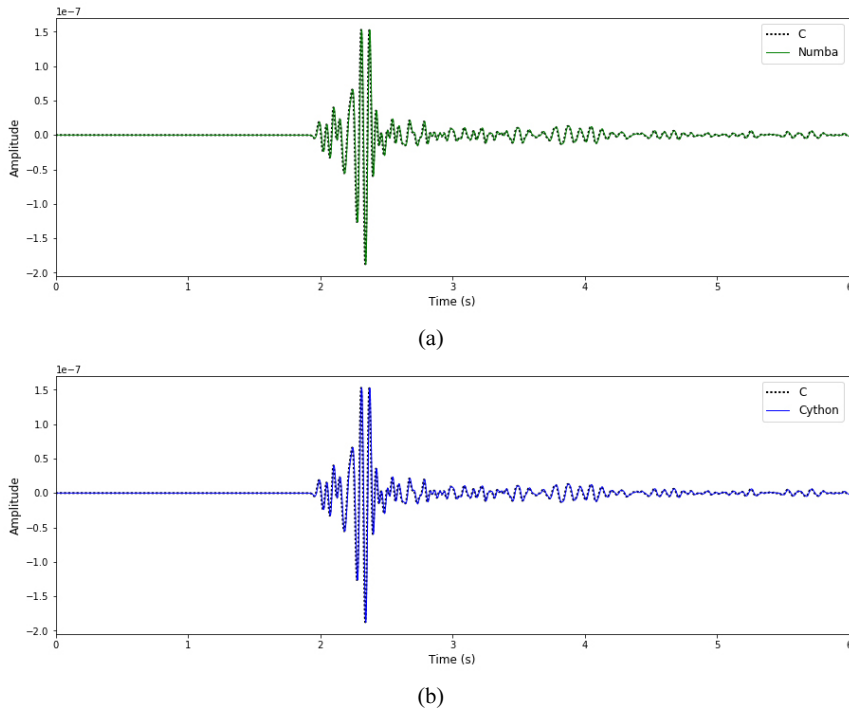


Fig. 6. Traces from the wavefields shown in Fig. 5. Traces are extracted at 4 km from the left. Results from (a) C and Numba, (b) C and Cython.



**Fig. 7.** Traces from the wavefields shown in Fig. 5. Traces are extracted at 10 km from the left. Results from (a) C and Numba, (b) C and Cython.

하며, 무시할 수 있을 만큼의 차이만 존재함을 확인할 수 있었다.

## 결론

본 연구에서는 고성능 파이썬 라이브러리 및 C언어를 사용하여 3차원 시간 영역 파동 전파 모델링을 수행한 후, 성능 및 생산성을 비교하였다. 성능은 계산 시간, 생산성은 코드 줄 수를 통해 비교하였다. 고성능 파이썬 라이브러리인 Numba와 Cython을 이용한 경우, 그리고 C언어를 이용한 경우 Numpy 배열만 이용한 순수 파이썬보다 각각 최대 1,844배, 2,049배, 2,305배 속도 향상을 얻을 수 있었다. Numba와 Cython을 이용한 파이썬 프로그램은 C 프로그램에 비해 계산 속도가 23.3%, 16.2% 느렸지만 코드 줄 수를 통해 비교한 생산성은 각각 약 45.1%, 40.7% 높았다. SEG/EAGE 3차원 암염동 모델을 이용해 비교해 본 결과, Numba와 Cython은 C언어에 비해 계산 속도가 각각 22.9%, 13.3% 느렸으며, 트레이스를 추출하여 정확도를 비교해본 결과 거의 일치 하였다. 따라서 연구나 프로토타이핑의 경우와 같이 약 20% 정도의 속도 차이가 크게 중요하지 않는

경우에는 생산성이 뛰어난 파이썬과 고성능 라이브러리들을 사용하는 것이 더 유리할 것이다. 기존의 C 또는 포트란 함수가 있다면 파이썬에서 불러 쓰는 것 또한 가능하다. 앞으로 탄성파 자료 처리뿐 아니라 다양한 고성능 수치해석 분야에서 파이썬 언어의 사용량이 더욱 늘어날 것으로 기대된다. 본 논문에 사용한 프로그램은 <https://github.com/rbgus8045/High-performance-Python>에서 확인할 수 있다.

## 사사

이 논문은 한국 지질자원연구원 주요사업인 ‘유가스 부존특성 및 해저천부구조 정밀 규명을 위한 융복합 탐사기술 개발’ 사업 및 산업통상자원부 자원개발특성화대학사업의 지원을 받아 수행되었음.

## References

- Aminzadeh, F., Burkhard, N., Nicoletis, L., Rocca, F., and Wyatt, K., 1994. SEG/EAGE 3-D modeling project: 2nd update. *The Leading Edge*, 13(9), 949-952.

- Beazley, D.M., 2009. *Python essential reference (4th Ed.)*, Addison-Weasley Professional, Boston, United States, p.5-24.
- Choi, J., 2010. *Numerical Analysis*, Textbooks, Seoul, Korea, p.237-277.
- Cython, 2018.04.30., <http://www.cython.org>.
- Ha, W., 2014. Productive small-scale data processing using Python. *Journal of Korean Society of Mineral and Energy Resources Engineers*, 51(5), 705-714.
- Ha, W., 2015. Development of a Python package for interactive seismic data processing classes. *Journal of Korean Society of Mineral and Energy Resources Engineers*, 52(4), 414-421.
- Keys, R.G., 1985. Absorbing boundary conditions for acoustic media. *Geophysics*, 50(6), 892-902.
- Kreyszig, E., 2011. *Advanced engineering mathematics, (10th Ed.)*, John Wiley & Sons, Inc., New York, United States, p.919-941.
- Lanaro, G., 2013. *Python high performance programming*, Packt Publishing, Birmingham, England, p.49-70.
- LLVM, 2018.04.30., <http://llvm.org>.
- McKinney, W., 2012. *Python for data analysis*, O'Reilly Media, California, United States, p.3-6.
- Min. D.-J., Pyun, S., Ha, W., Kwak, S., Chung, W., and Shin, C., 2016. *Numerical Analysis for Geophysics*, CIR, Seoul, Korea, p.37-52.
- Numba, 2018.04.30., <http://numba.pydata.org>.
- Prechelt, L., 2000. An empirical comparison of seven programming languages. *IEEE Computer*, 33(10), 23-29.
- PyPI, 2018.6.21., <http://pypi.github.io/PYPL.html>.
- Python, 2018.6.21., <http://www.python.org>.
- Redmonk, 2018.6.21., <http://redmonk.com/sograde/2018/03/07/language-rankings-1-18>.



### 조 상 훈

2016년 부경대학교 환경해양대학 에너지  
자원공학과 공학사  
2018년 부경대학교 대학원 에너지자원공  
학과 공학석사

현재 부경대학교 에너지자원연구소 전임연구원  
(E-mail; rbgus8045@naver.com)



### 하 원 수

2006년 서울대학교 공과대학 지구환경시  
스템공학부 공학사  
2008년 서울대학교 대학원 에너지시스템  
공학부 공학석사  
2011년 서울대학교 대학원 에너지시스템  
공학부 공학박사

현재 부경대학교 에너지자원공학과 부교수  
(E-mail; wansooaha@pknu.ac.kr)