Study of Algorithms for Fast Computation of Crack Expansion Problem

Farid Smaï[1] and Hideo Aochi[1]

Bureau de Recherches Géologieques et Miniére, Orléans, France
f.smai@brgm.fr; h.aochi@brgm.fr

Abstract
A problem of quasi-static growth of an arbitrary shaped-crack along an interface requires many times of iterations not only for finding a spatial distribution of discontinuity but also for determining the crack tip. This is crucial when refining model resolution and also when the phenomena progresses quickly from one step to another. We propose a mathematical reformulation of the problem as a nonlinear equation and adopt different numerical methods to solve it efficiently. Compared to a previous work of the authors, the resulting code shows a great improvement of performance. This gain is important for further application of aseismic slip process along the fault interface, in the context of plate convergence as well as the reactivation of fault systems in reservoirs.

Keywords: Arbitrary shaped-crack, Aseismic slip, Boundary integral equation, Optimization.

1 Introduction

Earthquakes occur to release the strain accumulated by tectonic movement and also can be induced due to stress change of pore pressure or other factors. The earthquake generation process is a boundary problem of continuum medium with or without inertia term. The equations of motion with inertia solve the dynamic process of an earthquake and the wave propagation, usually occurring within a few to tenths seconds. For example, in an elastic, isotropic medium characterized by medium density ρ and elastic coefficients (Lamé constants λ and μ), it is written as

$$\rho\ddot{\boldsymbol{u}} = (\lambda + 2\mu)\nabla(\nabla \cdot \boldsymbol{u}) - \mu\nabla \times (\nabla \times \boldsymbol{u}) + \boldsymbol{f}, \tag{1}$$

where $\boldsymbol{u}$ is displacement field and $\boldsymbol{f}$ is body force (e.g. [2]). On the other hand, the equilibrium theorem of the continuum medium without inertia solves the quasi-static process of fault sliding often called as aseismic slip, slow earthquake or slow slip, which may last from hours to centuries;

$$\boldsymbol{0} = (\lambda + 2\mu)\nabla(\nabla \cdot \boldsymbol{u}) - \mu\nabla \times (\nabla \times \boldsymbol{u}) + \boldsymbol{f}. \tag{2}$$

In Equation (2), there is no more notion of time, but visco-elasticity or plasticity might be considered as well according to the time scale and material rheology.

From the numerical point of view, the dynamic process, Equation (1), can be solved explicitly using boundary integral equation[7, 8], finite difference[19] or spectral element methods [5]. It can be also solved implicitly using finite element or finite volume methods, for example, but the mechanical causality (distance where waves propagate) within a short time span allows limiting the non-zero components in matrix of the equations. In the contrast, the quasi-static process, Equation (2), needs to solve a full matrix due to the equilibrium equations (response in the medium arrives infinitely) regardless of the methods.

Let us consider a quasi-static evolution of an arbitrary-shaped crack along an inteface (Σ) in a volume V. This is a typical problem of the Earth science, called aseismic slip on a fault. Slip should be properly distributed along the fault, so as to satisfy the stress field of the medium. Here, "slip" is defined as an amount of discontinuity along the both sides of the interface; $\boldsymbol{w} = \boldsymbol{u}\big|_{\Sigma+} - \boldsymbol{u}\big|_{\Sigma-}$ [2]. In the context of earthquakes, only the shear component, parallel to the fault interface, is generally considered. From the mechanical point of view, slip ($\boldsymbol{w}$) at any point starts when applied stress traction τ exceeds some critical point τ_p. Once it begins, stress-slip relation is still controlled by any constitutive relation (friction law), $\boldsymbol{\tau} = g(\boldsymbol{w}, \dot{\boldsymbol{w}})$ because shear slip continues having contact along the interface. During this process, slip $\boldsymbol{w}$ and shear traction $\boldsymbol{\tau}$ have the same vector direction. The boundary condition along the interface is then classified into two categories, namely inside and outside of crack:

- For inside of crack (domain S), the slip $\boldsymbol{w}(\neq 0)$ appears such that the stress-slip constitutive relation is satisfied on it.
- For outside of crack (domain Σ_∞, the slip does not appear ($\boldsymbol{w} = 0$), because the stress applied on this point does not yet the rupture criteria.

For the purpose of solving Equation (2), these conditions are sufficient, as the boundary condition surrounding the medium (speaking exactly, the boundary of interest is surrounded by the medium) is given by either of stress or slip condition. One can write the boundary condition:

$$\begin{cases} \left|\boldsymbol{\tau}\right| < \tau_p & \text{for } \left|\boldsymbol{w}\right| = 0 \text{ on } \Sigma_\infty \\ \boldsymbol{\tau} = g(\boldsymbol{w}, \dot{\boldsymbol{w}}) & \text{for } \left|\boldsymbol{w}\right| \neq \boldsymbol{0} \text{ on } S \end{cases}. \tag{3}$$

As Equation (3) is conditioned, one has to solve not only the distribution of $\boldsymbol{w}$ and $\boldsymbol{\tau}$, but also the spatial limit of the ruptured area S which is variable during the simulation. This is the reason why the calculation often requires much more iterations just for finding adequate domain S than finding the corresponding solutions $\boldsymbol{w}$ and $\boldsymbol{\tau}$ on S. If the whole fault plane is always sliding ($\Sigma = S$), the problem reduces to find just a pair of $\boldsymbol{w}$ and $\boldsymbol{\tau}$ simultaneously. Therefore the dimension of problem is fixed, even if it is huge.

This study aims to apply the efficient ways to find quickly the correct domain size of S and the solution $(\boldsymbol{w}, \boldsymbol{\tau})$. We reformulate mathematically the problem, describe the possible algorithms to be implemented and show the performance for a simple test case at the end.

2 Problem and Reformulation

2.1 Boundary Integral Equation

Let us consider an infinite planar fault in a 3D infinite, isotropic homogeneous elastic medium. A crack may grow from the model center, initially driven by the fluid migration along the

fault [3]. This is a well-known situation as a matured fault interface at depth (ex. plate boundary or subduction) or a fault reactivation due to fluid injection (ex. enhanced geothermal system, geological storage, and so on). For demonstrating the procedure, we simplify the model configuration by prefixing one direction of $\boldsymbol{w}$ and $\boldsymbol{\tau}$. Hereafter we refer slip and shear traction just as scalars. In the medium of interest, analytical solutions exist for Equation (2) and therefore one can rewrite it in the form of integral eqation. The stress field at any position ($\boldsymbol{x}$) in the medium (V) including the interface (Σ) is [18]

$$\begin{cases} \tau(\boldsymbol{x}) = \tau_0(\boldsymbol{x}) + \Delta\tau \\ \Delta\tau(\boldsymbol{x}) = -\dfrac{\mu}{4\pi}\displaystyle\int_\Sigma \frac{\boldsymbol{x}-\boldsymbol{\xi}}{|\boldsymbol{x}-\boldsymbol{\xi}|^3}\cdot A\nabla w d\boldsymbol{\xi} \end{cases}, \tag{4}$$

where τ_0 is initial stress applied on the system before the phenomena and $A = \begin{bmatrix} 2(1-p^2) & 0 \\ 0 & 1 \end{bmatrix}$. $p = |v_S|/|v_P|$ is the ratio of the S-wave over P-wave velocity. Using integration by parts and assuming ∇w vanishes on boundaries, one has

$$\Delta\tau(\boldsymbol{x}) = \frac{\mu}{4\pi}\int_\Sigma \frac{1}{|\boldsymbol{x}-\boldsymbol{\xi}|}\nabla\cdot(A\nabla w)\, d\boldsymbol{\xi} = \gamma * \nabla\cdot(A\nabla w) \tag{5}$$

with $\gamma(\boldsymbol{r}) = \frac{\mu}{4\pi}|\boldsymbol{r}|^{-1}$.

2.2 Friction Law

Next, let us specify Equation (3) for a problem of interest. A type of Coulomb criteria is often introduced for τ_p, being proportional to the effective normal stress on Σ:

$$\tau_p = \mu_f \sigma_{eff} = \mu_f(\sigma_n - Pr). \tag{6}$$

where μ_f is static frictional coefficient, and the effective normal stress σ_{eff} is defined as the applied normal stress σ_n reduced by a pore pressure Pr. After the onset of rupture, the function $g(w)$ still controls the process. Many relations have been proposed after laboratory experiments: A rate- and state-dependent friction [16] is mostly used for simulating earthquake cycles on plate boundaries. We adopt a slip-dependent friction consisting of static (τ_p) and dynamic ($\tau_r < \tau_p$) frictions with a gradual weakening transition [13]. This is generally used for modeling rupture process of an earthquake [9] and also suggested for slow slip events found in Mexico [12]. The following form of the expression is introduced:

$$g(w) = \tau_p - (\tau_p - \tau_r)(1 - \frac{w}{D_c})H(1 - \frac{w}{D_c}), \tag{7}$$

where τ_p is defined by Equation (6) and τ_r is similarly given as $\tau_r = \mu_d(\sigma_n - Pr)$ with a dynamic frictional coefficient μ_d ($\mu_d < \mu_f$). D_c is a constant called as critical slip displacement and $H(\cdots)$ is Heaviside step function. For earthquake rupture on natural faults, scaling in D_c should be important [9]. The frictional parameters (μ_f and μ_r) and D_c may be spatially heterogeneous along the interface Σ.

Noting that $g(0) = \tau_p$, Equation (3) can be stated as complementarity constraint:

$$\left| \begin{array}{lll} w > 0 & \text{and} & g(w) - \tau = 0 \\ \text{or} & & \\ w = 0 & \text{and} & g(w) - \tau > 0 \end{array} \right. .$$

Complementarity constraints can be transformed in equivalent nonlinear constraints (*e.g.* [11]) by using a function $\phi : \mathbb{R}^2 \to \mathbb{R}$ such that $\phi(a, b) = 0$ if and only if $a, b \geq 0$ and $ab \leq 0$. A simple example of such a function is the min function, $\phi_{\min}(a, b) = \min(a, b)$. Writing Equation (3) as a nonlinear constraint gives

$$\phi(w, g(w) - \tau) = 0. \tag{8}$$

2.3 Final Formulation

Combining Equations (5) and (8), the problem can finally be formulated as a nonlinear equation on w:

$$\phi\left(w, g(w) - \tau_0 - \gamma * \nabla \cdot (A \nabla w)\right) = 0. \tag{9}$$

Knowing w, τ is given by equation (5):

$$\tau = \tau_0 + \gamma * \nabla \cdot (A \nabla w).$$

3 Numerical computation

We discretize Σ with a regular grid and index the grid centers by $\boldsymbol{x}_{i,j} = (ih_1 + x_{0,0,1}, jh_2 + x_{0,0,2})$. The whole domain Σ is approximated by cell fields with constant on each cell. For any field v, $\tilde{v}$ denotes the vector of its values on the grid cells.

3.1 Discretization of the stress redistribution

Equation (5) uses space convolution and differentiation to express $\Delta\tau$ depending on w. Considering that w is discretized on a regular grid of Σ, the term $\nabla \cdot (A \nabla w)$ can be estimated by a two points flux approximation (TPFA, see [4]). By denoting the sparse matrix associated to the differential operator $\nabla \cdot A \nabla$ with L, the vector $L\tilde{w}$ approximates the values of $\nabla \cdot (A \nabla w)$ on each grid cell.

Considering any constant by cell field v, the value of the space convolution $\gamma * v$ on each cell center $\boldsymbol{x}_{i,j}$ can be written as a discrete convolution:

$$\begin{aligned}
(\gamma * v)(\boldsymbol{x}_{i,j}) &= \frac{\mu}{4\pi} \int_\Sigma \frac{v(\boldsymbol{\xi})}{|\boldsymbol{x}_{i,j} - \boldsymbol{\xi}|} d\boldsymbol{\xi} \\
&= \frac{\mu}{4\pi} \sum_{k,l} v(\boldsymbol{x}_{i,j}) \int_{\boldsymbol{x}_{k,l,1} - h_1/2}^{\boldsymbol{x}_{k,l,1} + h_1/2} \int_{\boldsymbol{x}_{k,l,2} - h_2/2}^{\boldsymbol{x}_{k,l,2} + h_2/2} \frac{d\boldsymbol{\xi}}{|\boldsymbol{x}_{i,j} - \boldsymbol{\xi}|} \\
&= \frac{\mu}{4\pi} \sum_{k,l} v(\boldsymbol{x}_{i,j}) \int_{(k-i-1/2)h_1}^{(k-i+1/2)h_1} \int_{(l-j-1/2)h_2}^{(l-j+1/2)h_2} \frac{d\boldsymbol{\xi}}{|\boldsymbol{\xi}|} \\
&= \frac{\mu}{4\pi} \sum_{k,l} v(\boldsymbol{x}_{i,j}) K_{k-i,l-j}.
\end{aligned}$$

Using the following function $\Gamma(x, y)$:

$$\begin{aligned}
\Gamma(x, y) &= \int_0^x \int_0^y \frac{dx\, dy}{\sqrt{x^2 + y^2}} \\
&= \frac{x}{2} \log\left(\frac{\sqrt{x^2 + y^2} + y}{\sqrt{x^2 + y^2} - y}\right) + \frac{y}{2} \log\left(\frac{\sqrt{x^2 + y^2} + x}{\sqrt{x^2 + y^2} - x}\right),
\end{aligned} \tag{10}$$

the convolution kernel K is given by

$$\begin{aligned} K_{k,l} =& \Gamma((k+1/2)h_1,(l+1/2)h_2) + \Gamma((k-1/2)h_1,(l-1/2)h_2) \\ & - \Gamma((k+1/2)h_1,(l-1/2)h_2) - \Gamma((k-1/2)h_1,(l+1/2)h_2). \end{aligned}$$

One should note that computing a discrete convolution as a sum over the N cells of the grid presents a computational complexity of $O(N^2)$. This complexity is reduced to $O(N \log N)$ with the help of the fast Fourier transform.

Finally, the stress redistribution equation (5) is discretized by

$$\Delta\tilde{\tau} = \frac{\mu}{4\pi} K * L\tilde{w}$$

3.2 Solving the Discrete Problem

The continuous problem (9) on w can now be discretized and consists in finding $\tilde{w}$ such that

$$F(\tilde{w}) = \phi(\tilde{w}, g(\tilde{w}) - \tilde{\tau}_0 - \frac{\mu}{4\pi} K * L\tilde{w}) = 0. \tag{11}$$

The Newton method is used to solve $F(\tilde{w}) = 0$, and defines the sequence $(\tilde{w}^n)_n$ by

$$\tilde{w}^{n+1} = \tilde{w}^n + \Delta\tilde{w}^n \quad \text{with} \quad J_F(\tilde{w}^n)\Delta\tilde{w}^n = -F(\tilde{w}^n).$$

Here J_F denote the Jacobian matrix of F. The sequence is computed from a given $\tilde{w}^0$ until the residual's norm goes below a given tolerance: $||F(\tilde{w}^n)|| \leq tol$.

By defining the diagonal matrices $D_{\phi 1}$, $D_{\phi 2}$ and D_g as

$$(D_{\phi 1}(a,b))_{i,i} = \frac{\partial \phi}{\partial a}(a_i,b_i) \quad , \quad (D_{\phi 2}(a,b))_{i,i} = \frac{\partial \phi}{\partial b}(a_i,b_i) \quad \text{and} \quad (D_g(\tilde{w}))_{i,i} = g'(\tilde{w}_i)$$

and by denoting the full matrix $M = \frac{\mu}{4\pi} K * L$, the Jacobian matrix is given by

$$J_F(\tilde{w}) = D_{\phi 1}(\tilde{w}, g(\tilde{w}) - \tilde{\tau}_0 - M\tilde{w}) + D_{\phi 2}(\tilde{w}, g(\tilde{w}) - \tilde{\tau}_0 - M\tilde{w})(D_g(\tilde{w}) - M).$$

Due to the convolution with K involved in M, the matrix J_F is full and we never assemble or store it. To perform the inversion of J_F required by the Newton steps, we used a Krylov iterative method (*e.g.* GMRES, see [17]). These kind of method does not strictly need an assembled matrix but only to perform matrix-vector products when needed. This allows us to use the fast Fourier transform to efficiently compute the convolution. Being iterative, the Krylov methods produce a sequence $(\Delta\tilde{w}^{n,k})_k$ to approximate $\Delta\tilde{w}^n$. We use a simple adaptive criterion to control the termination of the method: $||J_F(\tilde{w}^n)\Delta\tilde{w}^{n,k} + F(\tilde{w}^n)|| \leq 10^{-3}||F(\tilde{w}^n)||$. The idea here is that the Newton correction at a given step does not really need a high precision if this step is still far from convergence. This adaptive criterion reduces the total number of Krylov iterations over the whole computation. The choice of the preconditioner has an important impact of the number of Krylov iterations. The incomplete LU factorization (ILU, see [17]) is a robust and common choice for the preconditioner, but it requires an assembled matrix. Here, we use the ILU factorization but on a sparse approximation, $\bar{J}_F$, of J_F. The approximation is done by using a sparse approximation of the kernel K, denoted $\bar{K}$, such that

$$\bar{K}_{k,l} = \begin{cases} K_{k,l} & \text{if } |k-l| \leq 2 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \bar{J}_F = D_{\phi 1} + D_{\phi 2}(D_g - \frac{\mu}{4\pi}\bar{K} * L).$$

Parameter	Quantity
Rigidity μ	30 [GPa]
Ratio of shear and compressive wave velocity p	0.3
Initial normal stress $\sigma_n(t=0)$	100 [MPa]
Frictional coefficients μ_f and μ_d	0.65 and 0.55
Initial shear and normal stresses $\tau(t=0)$ and $\sigma_n(t=0)$	30 and 100 [MPa]
Critical slip displacement D_c	0 m

Table 1: Model parameters used in this study.

4 Test Case and Result

4.1 Test Case Description

The process of pore pressure in Equation (6) Pr [3] can be solved as flow in porous medium but now is neglected. We *a priori* provide a spatial distribution of Pr on Σ:

$$Pr(\boldsymbol{x}) = P_0 + \Delta P \exp\Big(-|\boldsymbol{x}|^2/R^2\Big), \tag{12}$$

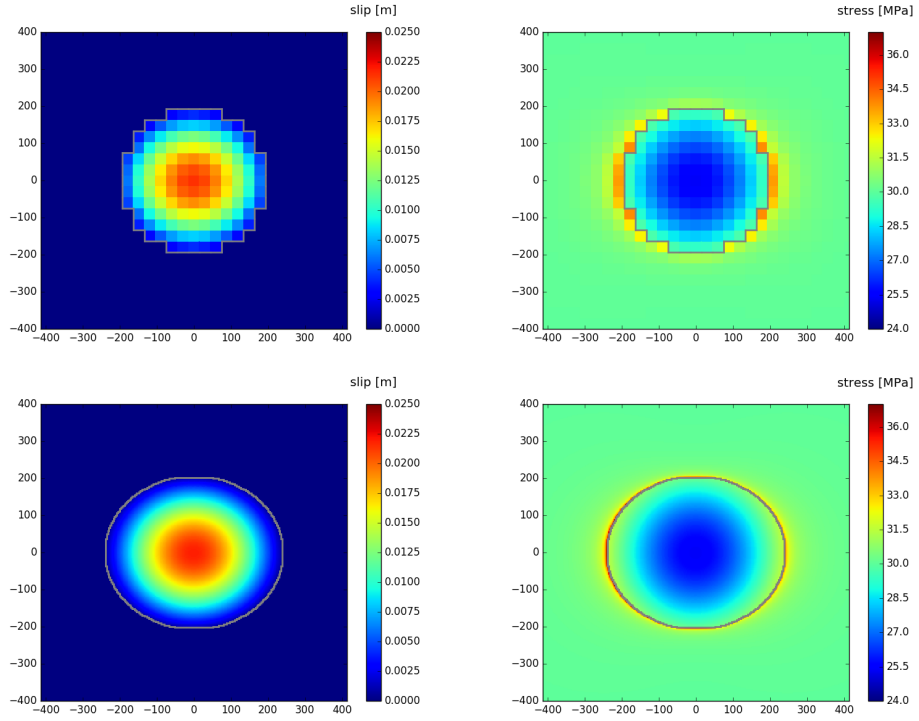

Figure 1: Slip (left) and shear stress (right) computed on 101×101 grid (above) and 1001×1001 grid (below). The whole domain is $(-1500m, 1500m) \times (-1500m, 1500m)$ and results are zoomed in $(-400m, 400m) \times (-400m, 400m)$. The grey line shows the limit between the ruptured S $(w > 0)$ and unruptured Σ_∞ $(w = 0)$ regions.

where P_0, R and ΔP are constant, being fixed as $P_0 = 30$ MPa, $R = 300$ m and $\Delta P \equiv \sigma_n(t = 0) - P_0 - (\tau(t = 0) - \epsilon)/\mu_f$ with $\epsilon = 10^{-3}MPa$, such that the applied stress τ exceeds the rupture criteria τ_p only around the origin (radius ~ 1 m), namely on a single element located at the origin, initially. The model parameters are summarized in Table 1. The problem is now to find the dimension of S and the corresponding solutions $\boldsymbol{w}$ and $\boldsymbol{\tau}$.

However, one element ruptured is not the final solution of this problem. Onset of the rupture (sliding) on an element increases the shear stress in the surrounding, and some other elements may reach the rupture criteria. Let us suppose a fault of 3000 m $\times$ 3000 m around the origin where the peak of pore pressure Pr is set. As a reference, we discretize the fault with an square element of $\Delta s \approx 30m$ length, namely 101 $\times$ 101 elements. We choose an odd number of cells in each direction to ensure that the initial state contains exactly one cell (the origin) out of equilibrium. We compare our original 'previous' code [3] with our 'new' one reformulated in the previous section. Then we refine the spatial discretization upto 10 times, namely $\Delta s \approx 3m$.

Each version of the codes is characterized in the following.

- Previous code [3]: Equations (3) and (4) are programmed straightforward by Fortran90. The convolution is directly calculated in Equation (4). In order to find the dimension of S, iterations are held to check Equation (3). The distribution of $\boldsymbol{w}$ is solved by Levenberg-Marquardt method [15].

- New code (this study): The strategy described above is implemented in Python with the numerical libraries Numpy [14] and Scipy [10]. The FFTW library [6, 1] is used to perform efficient fast Fourier transforms. Thanks to these libraries, the code is capable of shared memory parallelism and thus exploiting multi-core architectures.

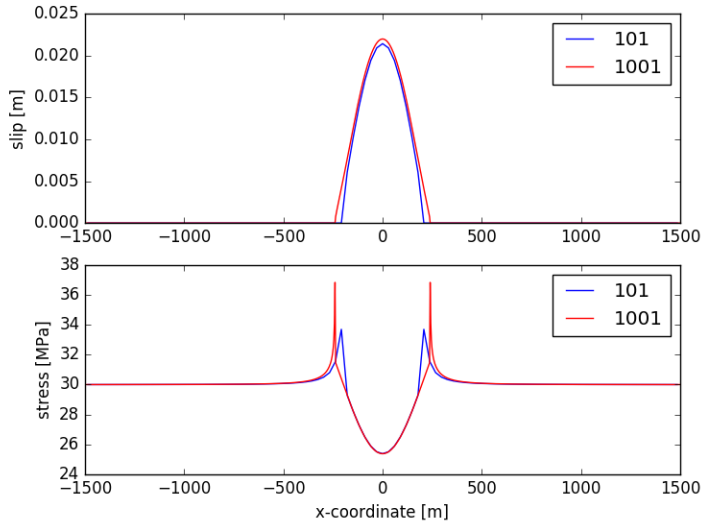

Figure 2: Cross section along the line $x_2 = 0$ (dislocation direction at the center of the model) of the slip and shear stress computed on 101 $\times$ 101 grid (blue) and 1001 $\times$ 1001 grid (red).

4.2 Results and Performance

Simulations on the test case have been performed for two grids with ten times difference in resolution: 101×101 and 1001×1001. Figure 1 presents the final slip and shear stress for the two grids. The observed anisotropy results from the wave velocity ratio $p = 0.3$. Slip and shear stress is supposed toward x_1-direction. When $p^2 < 0.5$, the pattern follows the dislocation direction whereas when $p^2 > 0.5$, the pattern is in the orthogonal direction. This anisotropy is the best captured on the fine grid but a higher stress accumulation in the dislocation direction is still observed on the coarse grid. Figure 2 presents the cross section of the results along the line $x_2 = 0$ (the dislocation direction). It shows how thin ($\approx 100\,m$) is the stress accumulation region around the ruptured area. This means that only 3 coarse cells can cover this length while the fine grid can be much more accurate with about 30 cells.

In order to discuss the performance of our new implementation, other simulations for the same test case have been performed with several $n \times n$ grids, by changing the grid size. Figure 3 shows how the CPU time increases according to the grid size. The performance of our previous code (see [3]) is also shown for comparison. Note that both codes are ran on different machines: our previous code is ran on a single core on Linux cluster (Bi-processor AMD Abu Dhabi at 2.6 GHz, 12 cores par processor) and new code is ran on a Intel Core i5-4590 CPU (3.3 GHz) under Windows environment. Thus the absolute time of the performance is dependent on the used system. However, the improvement of the performance is significant, not only in the absolute time but also in the computational complexity.

The measured complexities are around $O(n^4)$ for our previous code and $O(n^3)$ for our new code. The improvement of complexity is mainly explained by the computation of the convolution. Our previous code used a direct computation of complexity $O(N_{cell}^2) = O(n^4)$ whereas the new code uses the fast Fourier transform to achieve a complexity of $O(N_{cell} \log N_{cell}) =$

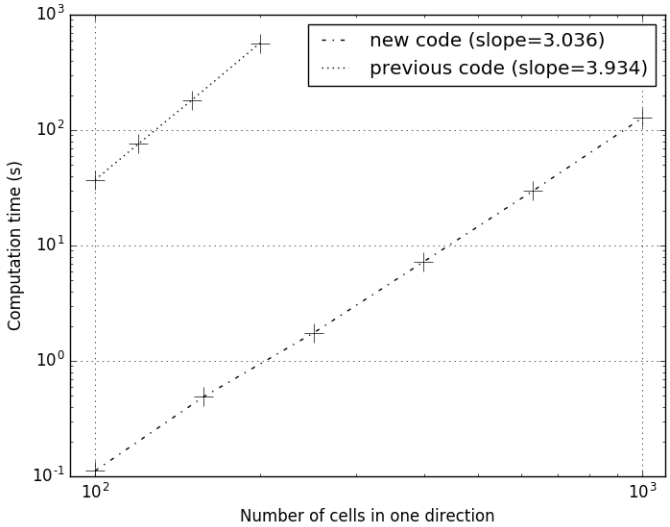

Figure 3: Computation time for different grid sizes and for the new code (dashed line) and our previous code (dotted line). All grids are $n \times n$, n varying from 10^2 to 10^3 ; the total number of cells, n^2, varies from 10^4 to 10^6. The slope of each curve is estimated through linear regression.

$O(n^2 \log n)$. The global complexity of the new code is nonetheless higher than $O(n^2 \log n)$ because the number of Newton iterations increases when the grid is finer. As the theoretical quadratic convergence of the Newton method relies on the problem regularity, the non optimal behavior of our Newton loop may be due to the stiffness of the shear stress (see Figure 2).

5 Discussion and Conclusion

In this study, we have shown the reformulation of a nonlinear crack expanding problem in order to improve the numerical performance. The reformulation has allowed a better understanding of the mathematical challenges of the problem and the use of proven numerical methods (Newton, Krylov and FFT methods) to tackle them. This leads us to a higher performance code: faster on small grid and more efficient on larger grid, the code also takes advantage of the multi-core architectures of modern computers.

With our previous code [3], as shown in Figure 1, we could model a fault dimension of 3000 m $\times$ 3000 m with a grid of 30 m. Under the evolution of Pr due to injection and intrinsic heterogeneity of a fault plane, complex seismicity was able to be simulated, with earthquake magnitude from 1 to 4.5. Here magnitude is defined as $M_w = (\log_{10}(M_0) - 9.1)/1.5$ with seismic moment magnitude $M_0 = \int_\Sigma (\mu \cdot w) d\boldsymbol{\xi}$. Thus magnitude which can be simulated is related to the grid spacing and model dimension. For the purpose of the induced seismicity in a reservoir, it is expected to simulate micro-seismicity of magnitude much smaller than 1. It is numerically challenging to produce a large band of magnitude in a single simulation scheme so as to understand the behavior of the reservoir and also to assess the induced risk. With our new implementation, it becomes possible to simulate with ten times finer grid of $\Delta s = 3$ m. It is thus expected to produce micro-seismicity of around M_w = -0.5 to -1.0, which can be generally several tenths to a hundred more than earthquakes of $M_w = 1$. The heterogeneity of the Earth should be treated stochastically due to the limit of our knowledge. This requires many moderate-size simulations with probabilistic distribution of model parameters rather than a single huge simulation with deterministic parameter choice. Therefore, our new code is expected to simulate more micro-seismicity on a larger magnitude band in the realistic context of geoscience application.

Acknowledgments

This work has been supported by internal Research and Development fund of BRGM.

References

[1] pyFFTW. Python wrapper for FFTW.

[2] Keiiti Aki and Paul G. Richards. *Quantitative seismology (second edition).* University Science Books, New York, 2002.

[3] Hideo Aochi, Blanche Poisson, Renaud Toussaint, Xavier Rachez, and Jean Schmittbuhl. Self-induced seismicity due to fluid circulation along faults. *Geophys. J. Int.*, 196(3):1544–1563, 2014.

[4] Robert Eymard, Thierry Gallouët, and Raphaèle Herbin. Finite volume methods. *Handbook of numerical analysis*, 7:713–1018, 2000.

[5] Gaetano Festa and Jean-Pierre Volotte. Influence of rupture initiation on the intersonic transition: Crack-like versus pulse-like modes. *Geophys. Res. Lett.*, 33:L15320, 2006.

[6] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[7] Eiichi Fukuyama and Raul Madariaga. Integral equation method for plane crack with arbitrary shape in 3d elastic medium. *Bull. Seismol. Soc. Am.*, 85:614, 1995.

[8] Phlippe H. Geubelle and James R. Rice. A spectral method for three-dimensional elastodynamic fracture problems. *J. Mech. Phys. Solids.*, 43:1791–1824, 1995.

[9] Satoshi Ide and Hideo Aochi. Historical seismicity and dynamic rupture process of the 2011 tohoku-oki earthquake. *Tectonophys.*, 600:1–13, 2013.

[10] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2017-01-31].

[11] Sven Leyffer. Complementarity constraints as nonlinear equations: Theory and numerical experience. pages 169–208, 2006.

[12] Julie Maury, Hideo Aochi, and Mathilde Radiguet. Fault constitutive relations inferred from the 2009-2010 slow slip event in guerrero, mexico. *Geophys. Res. Lett.*, 41, 2014.

[13] Mitiyasu Ohnaka. A constitutive scaling law and a unified comprehension for frictional slip failure, shear fracture of intact rock, and earthquake rupture. *J. Geophys. Res.*, 108:2080, 2003.

[14] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

[15] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in Fortran: The art of Scientific Computing*. Cambridge University Press, 1986.

[16] Andy Ruina. Slip instability and state variable friction laws. *J. Geophys. Res.*, 88:10359–10370, 1983.

[17] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[18] Taku Tada, Eiichi Fukuyama, and Raul Madariaga. Non-hypersingular boundary integral equations for 3-d non-planar crack dynamics. *Comput. Mech.*, 25:613–626, 2000.

[19] Jean Virieux and Raul Madariaga. Dynamic faulting studied by a finite difference method. *Bull. Seismol. Soc. Am.*, 72:345–369, 1982.