Elastostatic computations on aggregates of grains with sharp interfaces, corners, and triple-junctions

Helsing, Johan; Ojala, Rikard

Published in:
International Journal of Solids and Structures

DOI:
10.1016/j.ijsolstr.2009.09.008

2009

Elastostatic computations on aggregates of grains with sharp interfaces, corners, and triple-junctions

Johan Helsing*, Rikard Ojala

Numerical Analysis, Centre for Mathematical Sciences,
Lund University, Box 118, SE-221 00 LUND, Sweden

Abstract

We present a fast algorithm for the calculation of elastostatic fields in two-dimensional assemblies of elastic grains, separated by sharp grain boundaries. The algorithm uses an integral equation approach, combined with the fast multipole method and recursive compression to resolve stress concentrations also very close to grain boundary junctions. Singular basis functions on analytic form are not required. Accurate results can be obtained at a cost roughly proportional to the number of grains in the assembly. Large-scale problems, with thousands of grains, are solved using modest computational resources.

Key words: Corner singularities, Elasticity, Fast multipole method, Granular media, Integral equation, Multi-wedge points

1. Introduction

Large-scale elastostatic computations are crucial parts in the simulation of many important material science processes. Often the computations have to be performed repeatedly, rapidly and also accurately as to resolve stress fields everywhere in the material and to prevent accumulated error from corrupting the final result. Examples include dislocation dynamics (Jonsson, 2003), grain boundary diffusion (Sethian and Wilkening, 2004; Wei et al., 2008), quasi-static crack growth (Englund, 2007), and more general damage evolution in composite materials (Kushch et al., 2008).

*Corresponding author

Email addresses: helsing@maths.lth.se (Johan Helsing), rikardo@maths.lth.se (Rikard Ojala)

URL: http://www.maths.lth.se/na/staff/helsing/ (Johan Helsing)

In terms of algorithmic development, the last decades have seen great progress. Particularly so for problems with simple boundary conditions and smooth well-separated boundaries. Here the introduction of matrix-free matrix-vector multiplication via fast multipole techniques (Greenbaum et al., 1992; Greengard and Rokhlin, 1987) has revived the use of integral equation methods (Mikhlin, 1964). Two-dimensional domains with up to several thousand separated circular inclusions or straight cracks in an elastic matrix pose no major difficulty in achieving high accuracy on ordinary computers (Greengard and Helsing, 1998; Helsing and Jonsson, 2002; Kushch et al., 2008; Liu, 2008; Mogilevskaya et al., 2007; Wang et al., 2005). Storage requirements and computational speed are also acceptable. Even thousands of smooth cylindrical non-aligned rigid fibers in three dimensions can be readily handled on multi-processor machines (Liu et al., 2005). Note that smooth boundaries and loads often result in smooth elastic fields which can be well resolved by polynomials or Fourier series. This is favorable for numerical solvers.

Unfortunately, many material geometries of engineering interest do not exhibit smooth boundaries. This has severe effects on numerics. Non-smooth boundaries generally give rise to solutions with complicated asymptotics close to singular points. These are hard to resolve irrespective of what numerical method is used (Jin and Wu, 2004). One technique to resolve singular fields is local mesh refinement, also called mesh grading (Atkinson, 1997). Alternatively, one can use singular basis functions depending on geometry and material parameters and whose exponents are constructed via variable separation and the solution of non-linear eigenvalue problems (Carpinteri and Paggi, 2007; Linkov and Koshelev, 2006; Mantič et al., 2003). Not only are both these techniques complicated in terms of analysis and programming – they also lead to performance degradation. Mesh grading makes spectra of system matrices grow, which is bad for convergence in iterative solvers. Singular basis functions may lead to reduced order accuracy in quadrature rules. A natural consequence of these difficulties is that elastostatic computations on domains modeling, for example, branching cracks, notches, polygonal inclusions, multilevel thin film packages, and aggregates of grains often stand in stark contrast to computations on domains with smooth boundaries. They are much smaller, less accurate, or slower (Dong et al., 2002; Englund, 2007; Jeon et al., 2008; Koshelev and Ghassemi, 2008; Noda et al., 2003; Sethian and Wilkening, 2004).

This paper takes a multi-level approach to resolving multiple junction singularities that arise in linear elasticity. We shall work in an integral equation environment. Our proposed technique, *recursive compression*, uses

both singular basis functions and mesh refinement. But unlike eigenfunction expansion techniques, the construction of our basis functions is purely numerical. There is no asymptotic analysis involved. And unlike ordinary mesh grading, our fine mesh is not visible on the coarse level where large linear systems are solved. The fine mesh is only used to construct the basis functions. More precisely, our basis functions are chosen as a set of functions that accurately solve local problems which, in addition to taking geometry and material parameters into account, also involve the nature of the applied load. This makes the basis set efficient. Our technique can be seen as an extension of a recent scheme for electrostatics (Helsing and Ojala, 2008b). The major difference between elastostatics and electrostatics is the type of integral equations required for modeling. In electrostatics they are of Fredholm's second kind with compact operators away from the singular boundary points. In elastostatics they are singular everywhere. The absence of compactness leads to more involved numerical constructions. But, as we shall see, the final scheme of this paper performs on par with that for electrostatics. There is just extra need for storage.

The paper is organized as follows: Section 2 presents integral equations for linear elasticity in multiphase granular materials. These contain a Cauchy-singular operator which is discretized in Sections 3 and 4. Section 5 is about the resolution of integral operators near boundary singularities on fine grids and the compression back onto coarse grids. This gives block diagonal matrices whose fast and accurate construction is the topic of Sections 6, 7, and 8. We aim at solving large-scale problems. Our compressed equations are stable under increased resolution. Still, as more grains are added or as stronger inhomogeneity is introduced, spectra of system matrices may grow. Then the preconditioning of Section 9 is useful. Well documented small-scale numerical examples are given in Section 10. The ability to retain high accuracy and speed as the computational domain grows in complexity is illustrated by the large-scale examples of Section 11. Some details, needed for reproducibility, are collected in Appendix A and B.

2. An elastostatic multiphase inclusion problem

This section summarizes and extends results from Helsing and Peters (1999); Sherman (1959); Theocaris and Ioakimidis (1977). We make no distinction between points or vectors in a real plane $\mathbb{R}^2$ and points in a complex plane $\mathbb{C}$. All points will be denoted z or τ.

Let an inclusion made out of $N_{\rm gr}$ linearly elastic grains with two-dimensional bulk and shear moduli κ_k and μ_k, $k = 1, \ldots, N_{\rm gr}$, be embedded in an infinite

elastic plane D. The remainder of D has bulk and shear modulus κ_0 and μ_0. The local moduli $\kappa(z)$ and $\mu(z)$ are then piecewise constant functions on D. Let the boundary of all grains be denoted Γ and be given orientation. Γ will have branching-points in the form of triple-junctions for $N_{\rm gr} > 1$. Γ may also have corners. Let $n(z)$ be the outward unit normal of Γ at z. Let the stress

$$\lim_{z\to\infty} (\sigma_{xx}(z), \sigma_{yy}(z), \sigma_{xy}(z)) = (\sigma_{xx}^{\infty}, \sigma_{yy}^{\infty}, \sigma_{xy}^{\infty}) \tag{1}$$

be applied at infinity. The stress field in D can be computed in several ways. For example, via a system of second-order elliptic partial differential equations for the displacement field or via a variety of singular and hypersingular integral equations for quantities related to the Airy stress function. The stress state at infinity may be described by two constants α and β as $\sigma_{xx}^{\infty} + \sigma_{yy}^{\infty} = 2\alpha$ and $\sigma_{yy}^{\infty} - \sigma_{xx}^{\infty} + 2\mathrm{i}\sigma_{xy}^{\infty} = 2\bar{\beta}$, where the bar means complex conjugation.

In terms of a complex layer density $\Omega(z)$ on Γ and via the analytic functions

$$\Phi(z) = \frac{1}{2\pi\mathrm{i}} \int_{\Gamma} \frac{\Omega(\tau)\,\mathrm{d}\tau}{\tau - z} + \frac{\alpha}{2}\,, \quad z \in D\backslash\Gamma\,, \tag{2}$$

and

$$\Psi(z) = -\frac{1}{2\pi\mathrm{i}} \left[\int_{\Gamma} \frac{\overline{\Omega(\tau)}\,\mathrm{d}\bar{\tau}}{\tau - z} + \int_{\Gamma} \frac{\bar{\tau}\Omega(\tau)\,\mathrm{d}\tau}{(\tau - z)^2} \right] + \beta\,, \quad z \in D\backslash\Gamma\,, \tag{3}$$

one can derive the singular integral equation for the inclusion problem

$$(I + \lambda_1 M_1 + \lambda_2 M_3)\,\Omega(z) = -\alpha\lambda_1 - \bar{\beta}\lambda_2 \frac{\bar{n}}{n}\,, \quad z \in \Gamma\,. \tag{4}$$

Here the two piecewise constant functions $\lambda_1(z), \lambda_2(z) \in [-1, 1]$ are

$$\lambda_1(z) = \frac{a_\kappa(z)}{b(z)}\,, \quad \lambda_2(z) = \frac{a_\mu(z)}{b(z)}\,, \quad z \in \Gamma\,, \tag{5}$$

with

$$a_\kappa(z) = \frac{1}{\kappa_+(z)} - \frac{1}{\kappa_-(z)}\,, \quad a_\mu(z) = \frac{1}{\mu_+(z)} - \frac{1}{\mu_-(z)}\,, \tag{6}$$

$$b(z) = \frac{1}{\kappa_+(z)} + \frac{1}{\kappa_-(z)} + \frac{1}{\mu_+(z)} + \frac{1}{\mu_-(z)}\,, \tag{7}$$

where subscripts '+' and '−' indicate values on the positive and on the negative side of Γ. Further, I is the identity and M_1 and M_3 are integral operators whose action on a function $f(z)$ are

$$M_1 f(z) = \frac{1}{\pi\mathrm{i}} \int_{\Gamma} \frac{f(\tau)\,\mathrm{d}\tau}{\tau - z}\,, \quad z \in \Gamma\,, \tag{8}$$

and

$$M_3 f(z) = \frac{1}{2\pi \mathrm{i}} \left[\int_\Gamma \frac{f(\tau)\,\mathrm{d}\tau}{\tau - z} + \frac{\overline{n(z)}}{n(z)} \int_\Gamma \frac{f(\tau)\,\mathrm{d}\tau}{\bar{\tau} - \bar{z}} \right.$$

$$\left. + \int_\Gamma \frac{\overline{f(\tau)}\,\mathrm{d}\bar{\tau}}{\bar{\tau} - \bar{z}} + \frac{\overline{n(z)}}{n(z)} \int_\Gamma \frac{(\tau - z)\overline{f(\tau)}\,\mathrm{d}\bar{\tau}}{(\bar{\tau} - \bar{z})^2} \right], \quad z \in \Gamma\,. \tag{9}$$

The operator M_3 is compact on smooth Γ and computable limits exist for its kernel when $\tau \to z$. The potentials $\Phi(z)$ and $\Psi(z)$, which can be evaluated once $\Omega(z)$ is solved from (4), are related to the stress field in D via

$$\sigma_{xx} + \sigma_{yy} = 4\Re\mathrm{e}\{\Phi(z)\}\,, \tag{10}$$

$$\sigma_{yy} - \sigma_{xx} - 2\mathrm{i}\sigma_{xy} = 2\left(z\overline{\Phi'(z)} + \overline{\Psi(z)}\right)\,. \tag{11}$$

One can show, by multiplying both sides of (4) with $b(z)$ and integrating over Γ with respect to z, that for non-zero moduli

$$\int_\Gamma \Omega(z)\,\mathrm{d}z = 0\,. \tag{12}$$

This corresponds to single-valued displacements and enables the construction of yet another integral equation for the inclusion problem in terms of a layer density $\omega(z)$, which is a primitive function of $\Omega(z)$,

$$(I + \lambda_1 M_1 + \lambda_2 M_2)\,\omega(z) = -\alpha\lambda_1 z + \bar{\beta}\lambda_2 \bar{z}\,, \quad z \in \Gamma\,, \tag{13}$$

where the action of the integral operator M_2 is

$$M_2 f(z) = \frac{1}{\pi} \int_\Gamma f(\tau) \Im\mathrm{m}\left\{\frac{\mathrm{d}\tau}{\tau - z}\right\} + \frac{1}{2\pi\mathrm{i}} \int_\Gamma \overline{f(\tau)}\,\mathrm{d}\left[\frac{\tau - z}{\bar{\tau} - \bar{z}}\right], \quad z \in \Gamma\,. \tag{14}$$

In addition to being more regular than $\Omega(z)$, the density $\omega(z)$ is useful when displacements are to be computed. See Helsing and Peters (1999) for details.

Remark 1: Planar elasticity can be viewed as either a special case of three-dimensional linear elasticity with planar loads known as 'plane strain' or an approximation of three-dimensional linear elasticity with planar loads known as 'plane stress'. Under plane strain conditions one has $\lambda_1(z) + \lambda_2(z) \in [-1, 1]$ and $\lambda_2 - \frac{5}{3}\lambda_1(z) \in [-1, 1]$. Under plane stress conditions one has $\lambda_1(z) + \lambda_2(z) \in [-1, 1]$ and $\frac{5}{3}\lambda_2(z) - \lambda_1(z) \in [-1, 1]$.

3. Regularization and integrals of the solution

The action of M_1 of (8) should be interpreted in the Cauchy principal value sense. Let

$$c_\kappa(z) = \frac{1}{\kappa_+(z)} + \frac{1}{\kappa_-(z)} - \frac{2}{\kappa_0}\,, \quad z \in \Gamma\,. \tag{15}$$

One can then show

$$\frac{1}{\pi \mathrm{i}} \int_\Gamma \frac{a_\kappa(\tau)\,\mathrm{d}\tau}{\tau - z} = c_\kappa(z)\,, \quad z \in \Gamma\,, \tag{16}$$

and write the action of $\lambda_1 M_1$, occurring in (4) and (13), in *globally regularized* form as

$$\lambda_1 M_1 f(z) = \frac{c_\kappa}{b} f(z) + \frac{1}{b(z)\pi \mathrm{i}} \int_\Gamma \frac{(a_\kappa(z) f(\tau) - a_\kappa(\tau) f(z))\,\mathrm{d}\tau}{\tau - z}\,, \quad z \in \Gamma\,. \tag{17}$$

Note that $a_\kappa(z)$, $b(z)$, $c_\kappa(z)$, and also $\lambda_1(z)$ and $\lambda_2(z)$ all are piecewise constant functions on Γ which only change values at triple-junctions.

After having solved (4) or (13) for a given problem and obtained the density $\Omega(z)$ or $\omega(z)$, we would like to compute some functionals on the solution for convergence studies. We settle for the integrals

$$q_1 = -\left(\frac{1}{\kappa_0} + \frac{1}{\mu_0}\right) \int_\Gamma \Im\mathrm{m}\{z\Omega(z)\,\mathrm{d}z\} = \left(\frac{1}{\kappa_0} + \frac{1}{\mu_0}\right) \int_\Gamma \Im\mathrm{m}\{\omega(z)\,\mathrm{d}z\} \tag{18}$$

and

$$\begin{aligned} q_2 &= -\int_\Gamma \Im\mathrm{m}\{(b + (a_\kappa + a_\mu) M_3)\,\Omega(z)\bar{z}\,\mathrm{d}z\} \\ &= \int_\Gamma \Im\mathrm{m}\{(b + (a_\kappa + a_\mu) M_2)\,\omega(z)\,\mathrm{d}\bar{z}\}\,, \end{aligned} \tag{19}$$

which are of importance in homogenization theory when the effective compliance tensor is to be computed. See Helsing and Peters (1999) for similar expressions in the context of the stiffness tensor.

4. Basic quadrature and the Cauchy integral

We shall use Nyström discretization for the integral equations and composite 16-point Gauss-Legendre quadrature as our basic quadrature tool. Let $\tau(t)$ be a parameterization of Γ and let there be n_Γ quadrature panels

placed on Γ. Then one can easily compute $N = 16n_\Gamma$ nodes t_k and weights w_k, $k = 1, \ldots, N$, associated with integration in t. Let f be a function on Γ. The parameterization allows us to view f both as function of position $f(\tau)$ and of parameter $f(t)$. We let the argument indicate which view is taken in a particular situation. The discretization of $\lambda_1 M_1$ of (17) leads to a matrix $\lambda_1 \mathbf{M}_1$ with entries

$$(\lambda_1 M_1)_{jk} = \delta_{jk} \left(\frac{c_\kappa(\tau_j)}{b(\tau_j)} - d_\kappa(\tau_j) \right) + \frac{\lambda_1(\tau_j)}{\pi \mathrm{i}} \frac{(1-\delta_{jk})\tau_k' w_k}{(\tau_k - \tau_j)} + \frac{2\lambda_1(\tau_j) w_j}{\pi \mathrm{i} \Delta_j} B_{jk}\,, \quad j,k = 1, \ldots, N\,. \tag{20}$$

Here $\tau_j = \tau(t_j)$, δ_{jk} is the Kronecker delta and the second term on the right hand side is zero for $j = k$,

$$d_\kappa(\tau_j) = \frac{1}{\pi \mathrm{i} b(\tau_j)} \sum_{\substack{k=1 \\ k \neq j}}^{N} \frac{a_\kappa(\tau_k)\tau_k' w_k}{\tau_k - \tau_j}, \tag{21}$$

the prime means differentiation so that $\tau_k' = \tau'(t_k)$, Δ_j is the length in parameter of the panel on which τ_j is situated, and B_{jk} are entries of a block diagonal matrix with identical 16×16 blocks $\mathbf{B}^I$ performing numerical differentiation based on polynomial interpolation at the Legendre nodes x_k on the canonical panel $[-1, 1]$. The entries of $\mathbf{B}^I$ are given by

$$\sum_{k=1}^{16} B^I_{jk} x_k^n = n x_j^{n-1}\,, \quad j = 1, \ldots, 16\,, \quad n = 0, \ldots, 15\,.$$

The choice of 16-point quadrature is a compromise between conflicting considerations such as the order of convergence and the size and conditioning of various auxiliary matrices.

The accuracy to which we can compute and interpolate quantities related to discretized Cauchy integrals will be of great importance in later sections. Consider the one-panel Cauchy integral

$$C_1(z) = \frac{1}{\pi \mathrm{i}} \int_{\Gamma_p} \frac{f(\tau)\,\mathrm{d}\tau}{\tau - z} \approx \frac{1}{\pi \mathrm{i}} \sum_{k=1}^{16} \frac{f_k \tau_k' w_k}{\tau_k - z}\,, \quad \tau_k \in \Gamma_p, \quad z \in D\,, \tag{22}$$

where Γ_p is a quadrature panel of arc length l_p, $f(\tau)$ is a smooth function, and z is a target point, perhaps on another panel Γ_q with arc length l_q. The discretization for $C_1(z)$ is often accurate to machine epsilon (ϵ_{mach}) when z

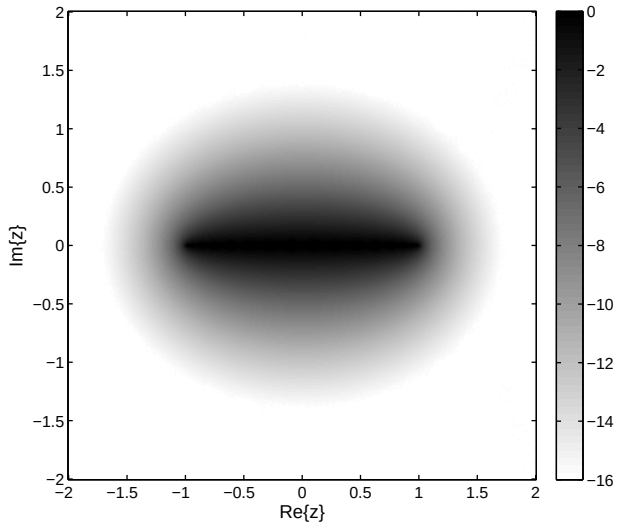

Figure 1: The discretization for $C_1(z)$ of (22) with $f(\tau) = 1$ and Γ_p being the line segment $[-1, 1]$. The figure depicts the base-10 logarithm of the pointwise absolute error for various $z \in D$.

is at least a distance l_p away from Γ_p. Fig. 1 shows that for $f(\tau) = 1$ and Γ_p being a line segment, the discretization is accurate for z approximately collinear with Γ_p already at a distance $0.5l_p$ away from the tips of Γ_p.

As for interpolation, consider the field due to a source at the origin

$$C_0(z) = \frac{l_q}{\pi \mathrm{i} z}\,, \quad z \in \Gamma_q\,. \tag{23}$$

Polynomial interpolation in z on Γ_q, based on $C_0(z_j)$ with $z_j \in \Gamma_q$ and $j = 1, \ldots, 16$, is illustrated in Fig. 2. The maximum interpolation error is shown as a function of the distance between the origin and Γ_q. When Γ_q is separated a distance $2l_q$, the interpolation is accurate to ϵ_{mach}. The result is similar for $C_1(z)$. It can typically be interpolated accurate to ϵ_{mach} when $l_q = l_p$ and Γ_q is at least a distance $2l_p$ away from Γ_p. For $f(\tau) = 1$, $l_q = l_p$, and Γ_p and Γ_q being collinear line segments separated a distance l_p, the maximum pointwise interpolation error in $C_1(z)$ is on the order of $10^3\epsilon_{\text{mach}}$.

5. Compressed equations

Assume that the boundary Γ has s corners or triple-junctions at points γ_i, $i = 1, \ldots, s$. A coarse mesh, that is, a division into quadrature panels

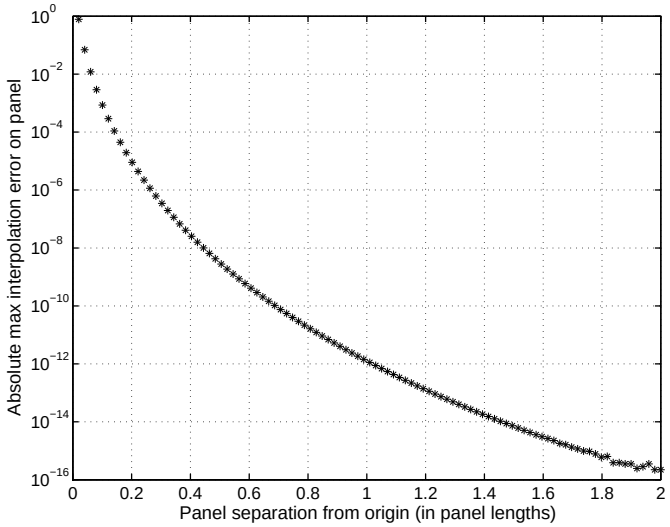

Figure 2: Maximum interpolation error of $C_0(z)$ of (23), $z \in \Gamma_q$, as a function of how far the origin is separated from the closest tip of Γ_q. A 15th-degree interpolating polynomial is used. Γ_q is a line segment collinear with the origin.

of approximately equal length, is constructed on Γ. The mesh is arranged so that no panel has γ_i as an interior point. Let $\Gamma_i^{1\star}$ refer to the part of Γ covered by coarse panels that are nearest neighbors to γ_i. The union of all $\Gamma_i^{1\star}$ is $\Gamma^{1\star}$. Now let there also be a fine mesh on Γ, constructed from the coarse mesh by repeated subdivision of the coarse panels on each $\Gamma_i^{1\star}$ in direction towards γ_i. See the left image of Fig. 3 for an example with one corner. Grids are constructed by placing quadrature points corresponding to composite 16-point Gauss-Legendre quadrature on the two meshes. Nyström discretization of (13) on the fine grid gives

$$(\mathbf{I}_{\text{fin}} + \lambda_1 \mathbf{M}_{1\text{fin}} + \lambda_2 \mathbf{M}_{2\text{fin}})\, \boldsymbol{\omega}_{\text{fin}} = \mathbf{g}_{2\text{fin}}\,, \tag{24}$$

where $\mathbf{I}_{\text{fin}}$, $\lambda_1 \mathbf{M}_{1\text{fin}}$, and $\lambda_2 \mathbf{M}_{2\text{fin}}$ are matrices and $\boldsymbol{\omega}_{\text{fin}}$ and $\mathbf{g}_{2\text{fin}}$ are column vectors, the latter corresponding to the right hand side. A similar discretization is done on the coarse grid with subscripts 'coa'. Obviously, the solution to the coarse grid equation will be less accurate than the solution to the fine grid equation. The purpose of this section is to compress (24) into an equation on the coarse grid without loss of point-wise accuracy (under some mild conditions on the relation between discretized operators and functions on the two grids). We shall also, in an analogous manner, compress a dis-

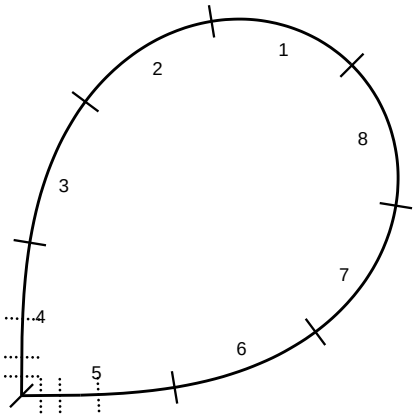

Figure 3: A coarse mesh with eight quadrature panels on a closed contour Γ with one corner. $\Gamma_1^{1\star}$ is panels 4 and 5, $\Gamma_1^{2\star}$ is panels 3 and 6, and $\Gamma_1^{\star}$ is panels 3, 4, 5, and 6. A fine mesh of 14 panels is created by subdividing $\Gamma_1^{1\star}$ three times towards the corner.

cretization of (4)

$$(\mathbf{I}_{\text{fin}} + \lambda_1 \mathbf{M}_{1\text{fin}} + \lambda_2 \mathbf{M}_{3\text{fin}})\, \boldsymbol{\Omega}_{\text{fin}} = \mathbf{g}_{3\text{fin}}\,. \tag{25}$$

5.1. Prolongation and restriction

We introduce two discrete operators, $\mathbf{P}$ and $\mathbf{Q}$ on sparse block-matrix form, intended to act on discretizations $\mathbf{f}_{\text{coa}}$ and $\mathbf{f}_{\text{fin}}$ of functions $f(t)$ on the two grids. $\mathbf{P}$ is the prolongation operator that performs panelwise 15th-degree polynomial interpolation in t from the coarse grid to the fine grid so that $\mathbf{P}\mathbf{f}_{\text{coa}} = \mathbf{f}_{\text{fin}}$ provided that $f(t)$ is resolved on the coarse grid on $\Gamma^{1\star}$. $\mathbf{Q}$ is the restriction operator that performs panelwise 15th-degree polynomial interpolation in the other direction. $\mathbf{P}$ and $\mathbf{Q}$ differ from identity matrices only for blocks corresponding to points on $\Gamma_i^{1\star}$ and they obey

$$\mathbf{Q}\mathbf{P} = \mathbf{I}_{\text{coa}}\,. \tag{26}$$

With $\mathbf{W}_{\text{coa}}$ and $\mathbf{W}_{\text{fin}}$ being diagonal matrices containing the quadrature weights on the two grids one can also show

$$\mathbf{P}^T \mathbf{W}_{\text{fin}} \mathbf{P} = \mathbf{W}_{\text{coa}}\,, \tag{27}$$

where superscript 'T' denotes the transpose.

5.2. Splitting

Let $\Gamma_i^{2\star}$ refer to the part of Γ covered by coarse panels that are next-nearest neighbors to γ_i. Let $\Gamma_i^\star = \Gamma_i^{1\star} \cup \Gamma_i^{2\star}$, let $\Gamma_i^\circ = \Gamma \backslash \Gamma_i^\star$, and let the union of all $\Gamma_i^\star$ be $\Gamma^\star$. Thus, $\Gamma^\star$ covers four coarse panels per corner, see Fig. 3, and six coarse panels per triple-junction. Let the matrices $\mathbf{M}_{2\text{coa}}$ and $\mathbf{M}_{2\text{fin}}$ be split into two parts each

$$\mathbf{M}_{2j} = \mathbf{M}_{2j}^\star + \mathbf{M}_{2j}^\circ \,, \quad j = \text{coa}, \text{fin} \,. \tag{28}$$

Here matrices with superscript '$\star$' contain 16×16 blocks of the unsplit matrices describing interaction and self-interaction of all panels on $\Gamma^\star$ which are close to the same γ_i. Matrices with superscript '$\circ$' contain the remaining elements. With *interaction* of two boundary parts Γ_p and Γ_q we mean matrix elements whose column indices belong to points on Γ_p and whose row indices belong to points on Γ_q, and vice versa. We also split $\lambda_1 \mathbf{M}_{1\text{coa}}$ and $\lambda_1 \mathbf{M}_{1\text{fin}}$, discretized according to (20), into two parts each, in an analogous way.

The coarse grid on Γ cannot be arbitrarily coarse. It should be sufficiently fine for

$$\lambda_1 \mathbf{M}_{1\text{fin}}^\circ \mathbf{W}_{\text{fin}}^{-1} = \mathbf{P} \lambda_1 \mathbf{M}_{1\text{coa}}^\circ \mathbf{W}_{\text{coa}}^{-1} \mathbf{P}^T \tag{29}$$

and

$$\mathbf{M}_{2\text{fin}}^\circ \mathbf{W}_{\text{fin}}^{-1} = \mathbf{P} \mathbf{M}_{2\text{coa}}^\circ \mathbf{W}_{\text{coa}}^{-1} \mathbf{P}^T \tag{30}$$

to hold to high accuracy. These requirements can be interpreted as that the restriction of the kernels whose discretizations are $\lambda_1 \mathbf{M}_{1\text{coa}}^\circ$ and $\mathbf{M}_{2\text{coa}}^\circ$ are resolved in both their variables on $\Gamma^{1\star}$.

Note that the diagonal elements of $\lambda_1 \mathbf{M}_{1\text{coa}}$ or $\lambda_1 \mathbf{M}_{1\text{fin}}$, accounting for self-interaction on a panel Γ_p, do not just contain contributions from discretization points on Γ_p but contain contributions from points on all of Γ via $d_\kappa(\tau_j)$ of (21). Consider, in particular, the diagonal elements of the square block of $\lambda_1 \mathbf{M}_{1\text{fin}}^\star$ accounting for self-interaction of a panel on $\Gamma_i^\star$. These elements will contain contributions $d_\kappa(\tau_j)$ of (21) which in turn could be split into two parts,

$$d_\kappa(\tau_j) = d_\kappa^\star(\tau_j) + d_\kappa^\circ(\tau_j) \,, \tag{31}$$

where $d_\kappa^\star(\tau_j)$ denotes the partial sum of (21) with $\tau_k \in \Gamma_i^\star$ and $d_\kappa^\circ(\tau_j)$ denotes the partial sum of (21) with $\tau_k \in \Gamma_i^\circ$. According to the discussion about interpolation of $C_1(z)$ in Section 4, $d_\kappa^\circ(z)$ is smooth for $z \in \Gamma_i^{1\star}$.

5.3. Compression

Now (24) will be compressed into an equation on the coarse grid. In addition to (29) and (30) we also assume that the coarse grid is sufficiently fine for

$$\mathbf{g}_{2\text{fin}} = \mathbf{P}\mathbf{g}_{2\text{coa}} \tag{32}$$

to hold to high accuracy. Matrix splitting (28) and use of the transformed density

$$\tilde{\boldsymbol{\omega}}_{\text{fin}} = (\mathbf{I}_{\text{fin}} + \lambda_1 \mathbf{M}^{\star}_{1\text{fin}} + \lambda_2 \mathbf{M}^{\star}_{2\text{fin}})\, \boldsymbol{\omega}_{\text{fin}} \tag{33}$$

in (24), together with (29,30,32), give

$$\tilde{\boldsymbol{\omega}}_{\text{fin}} + \mathbf{P}\,(\lambda_1 \mathbf{M}^{\circ}_{1\text{coa}} + \lambda_2 \mathbf{M}^{\circ}_{2\text{coa}})\, \mathbf{W}^{-1}_{\text{coa}} \mathbf{P}^T \mathbf{W}_{\text{fin}}\, (\mathbf{I}_{\text{fin}} + \lambda_1 \mathbf{M}^{\star}_{1\text{fin}} + \lambda_2 \mathbf{M}^{\star}_{2\text{fin}})^{-1}\, \tilde{\boldsymbol{\omega}}_{\text{fin}}$$

$$= \mathbf{P}\mathbf{g}_{2\text{coa}}\,. \tag{34}$$

Multiplication of (34) with $\mathbf{PQ}$ from the left and (26) imply that $\tilde{\boldsymbol{\omega}}_{\text{fin}}$ can be restricted and prolonged without loss of information

$$\tilde{\boldsymbol{\omega}}_{\text{fin}} = \mathbf{P}\mathbf{Q}\tilde{\boldsymbol{\omega}}_{\text{fin}}\,. \tag{35}$$

Defining

$$\tilde{\boldsymbol{\omega}}_{\text{coa}} = \mathbf{Q}\tilde{\boldsymbol{\omega}}_{\text{fin}} \quad \text{and} \quad \boldsymbol{\omega}_{\text{coa}} = \mathbf{Q}\boldsymbol{\omega}_{\text{fin}}\,, \tag{36}$$

one can, via (33,35,36), write

$$\boldsymbol{\omega}_{\text{coa}} = \mathbf{S}\tilde{\boldsymbol{\omega}}_{\text{coa}}\,, \tag{37}$$

where the compressed un-weighted inverse

$$\mathbf{S} = \mathbf{Q}\,(\mathbf{I}_{\text{fin}} + \lambda_1 \mathbf{M}^{\star}_{1\text{fin}} + \lambda_2 \mathbf{M}^{\star}_{2\text{fin}})^{-1}\, \mathbf{P} \tag{38}$$

has to be computed on the fine grid. The columns of $\mathbf{S}$ can, from (37), be interpreted as discrete basis functions for $\omega(z)$ on the coarse grid. Multiplication of (34) with $\mathbf{Q}$ from the left and use of (26) and (35–37), lead to the equation for $\boldsymbol{\omega}_{\text{coa}}$ on the coarse grid

$$\left(\mathbf{I}_{\text{coa}} + (\mathbf{S}^{-1} - \mathbf{I}_{\text{coa}}) + (\lambda_1 \mathbf{M}^{\circ}_{1\text{coa}} + \lambda_2 \mathbf{M}^{\circ}_{2\text{coa}})\, \mathbf{R}\mathbf{S}^{-1}\right) \boldsymbol{\omega}_{\text{coa}} = \mathbf{g}_{2\text{coa}}\,, \tag{39}$$

where also the compressed weighted inverse

$$\mathbf{R} = \mathbf{W}^{-1}_{\text{coa}} \mathbf{P}^T \mathbf{W}_{\text{fin}}\, (\mathbf{I}_{\text{fin}} + \lambda_1 \mathbf{M}^{\star}_{1\text{fin}} + \lambda_2 \mathbf{M}^{\star}_{2\text{fin}})^{-1}\, \mathbf{P} \tag{40}$$

has to be computed on the fine grid. Note that the block diagonal matrices $\mathbf{S}$ and $\mathbf{R}$ differ from $\mathbf{I}_{\text{coa}}$ only for blocks describing interaction and self-interaction between panels on the same $\Gamma_i^\star$. The term $(\mathbf{S}^{-1} - \mathbf{I}_{\text{coa}})$ in (39) can be interpreted as an accurate implementation of $\lambda_1 \mathbf{M}^\star_{1\text{coa}} + \lambda_2 \mathbf{M}^\star_{2\text{coa}}$.

Equation (39) has a counterpart in a compressed version of (25)

$$\left(\mathbf{I}_{\text{coa}} + (\mathbf{S}^{-1} - \mathbf{I}_{\text{coa}}) + (\lambda_1 \mathbf{M}^\circ_{1\text{coa}} + \lambda_2 \mathbf{M}^\circ_{3\text{coa}})\, \mathbf{R}\mathbf{S}^{-1}\right) \boldsymbol{\Omega}_{\text{coa}} = \mathbf{g}_{3\text{coa}}\,, \tag{41}$$

where $\mathbf{S}$ and $\mathbf{R}$ are as in (38) and (40) but with $\mathbf{M}^\star_{2\text{fin}}$ replaced by $\mathbf{M}^\star_{3\text{fin}}$.

5.4. Alternative formulations

The columns of $\mathbf{R}$ can be be interpreted as the columns of $\mathbf{S}$ multiplied with quadrature weight corrections. The action of $\mathbf{R}\mathbf{S}^{-1}$ on $\boldsymbol{\omega}_{\text{coa}}$ in (39), for example, produces a discrete density on which $\lambda_1 \mathbf{M}^\circ_{1\text{coa}}$ and $\lambda_2 \mathbf{M}^\circ_{2\text{coa}}$ can act accurately. Introducing the weight-corrected densities

$$\hat{\boldsymbol{\omega}}_{\text{coa}} = \mathbf{R}\mathbf{S}^{-1}\boldsymbol{\omega}_{\text{coa}} \quad \text{and} \quad \hat{\boldsymbol{\Omega}}_{\text{coa}} = \mathbf{R}\mathbf{S}^{-1}\boldsymbol{\Omega}_{\text{coa}} \tag{42}$$

on the coarse grid, one can rewrite (39) and (41) on the simpler form

$$\left(\mathbf{I}_{\text{coa}} + (\mathbf{R}^{-1} - \mathbf{I}_{\text{coa}}) + \lambda_1 \mathbf{M}^\circ_{1\text{coa}} + \lambda_2 \mathbf{M}^\circ_{2\text{coa}}\right) \hat{\boldsymbol{\omega}}_{\text{coa}} = \mathbf{g}_{2\text{coa}}\,, \tag{43}$$

$$\left(\mathbf{I}_{\text{coa}} + (\mathbf{R}^{-1} - \mathbf{I}_{\text{coa}}) + \lambda_1 \mathbf{M}^\circ_{1\text{coa}} + \lambda_2 \mathbf{M}^\circ_{3\text{coa}}\right) \hat{\boldsymbol{\Omega}}_{\text{coa}} = \mathbf{g}_{3\text{coa}}\,, \tag{44}$$

which is free of $\mathbf{S}$. One can also construct compressed right inverse preconditioned equations for the transformed densities $\tilde{\boldsymbol{\omega}}_{\text{coa}}$ and $\tilde{\boldsymbol{\Omega}}_{\text{coa}}$:

$$\left(\mathbf{I}_{\text{coa}} + (\lambda_1 \mathbf{M}^\circ_{1\text{coa}} + \lambda_2 \mathbf{M}^\circ_{2\text{coa}})\mathbf{R}\right) \tilde{\boldsymbol{\omega}}_{\text{coa}} = \mathbf{g}_{2\text{coa}}\,, \tag{45}$$

$$\left(\mathbf{I}_{\text{coa}} + (\lambda_1 \mathbf{M}^\circ_{1\text{coa}} + \lambda_2 \mathbf{M}^\circ_{3\text{coa}})\mathbf{R}\right) \tilde{\boldsymbol{\Omega}}_{\text{coa}} = \mathbf{g}_{3\text{coa}}\,. \tag{46}$$

5.5. Computing q_1 and q_2

The weight-corrected vector $\hat{\boldsymbol{\omega}}_{\text{coa}}$ is useful for numerical integration of $\omega(t)$ against functions that are well resolved by polynomials on the coarse grid. Therefore, once (40) is constructed and (43) is solved, the quantity q_1 of (18) can be obtained as

$$q_1 = \left(\frac{1}{\kappa_0} + \frac{1}{\mu_0}\right) \sum_{k=1}^{N} \Im\mathrm{m}\left\{(\hat{\boldsymbol{\omega}}_{\text{coa}})_k \tau'_k w_k\right\}\,. \tag{47}$$

The quantity q_2 of (19) is more involved. This is so since the kernel of the integral operator whose discretization is $\mathbf{M}^\star_{i\text{coa}}$ is not resolved in both

its variables on $\Gamma^{1\star}$ on the coarse grid. To accurately compute q_2 we also need the compressed block-diagonal matrix

$$\mathbf{X} = \mathbf{W}_{\rm coa}^{-1}\mathbf{P}^T\mathbf{W}_{\rm fin}\mathbf{M}^{\star}_{\rm 2fin}\left(\mathbf{I}_{\rm fin} + \lambda_1\mathbf{M}^{\star}_{\rm 1fin} + \lambda_2\mathbf{M}^{\star}_{\rm 2fin}\right)^{-1}\mathbf{P}\,. \tag{48}$$

Then

$$q_2 = \sum_{k=1}^{N} \Im{\rm m}\left\{b(\tau_k)(\hat{\boldsymbol{\omega}}_{\rm coa})_k\bar{\tau}'_k w_k\right\}$$

$$+\sum_{k=1}^{N} \Im{\rm m}\left\{(a_\kappa(\tau_k) + a_\mu(\tau_k))((\mathbf{M}^{\circ}_{\rm 2coa} + \mathbf{X}\mathbf{R}^{-1})\hat{\boldsymbol{\omega}}_{\rm coa})_k\bar{\tau}'_k w_k\right\}\,. \tag{49}$$

The formulas (47) and (49) can easily be adjusted as to allow for computations based on the densities $\boldsymbol{\omega}_{\rm coa}$, $\tilde{\boldsymbol{\omega}}_{\rm coa}$, $\hat{\boldsymbol{\Omega}}_{\rm coa}$, $\boldsymbol{\Omega}_{\rm coa}$, and $\tilde{\boldsymbol{\Omega}}_{\rm coa}$.

6. Intermediary results for one corner

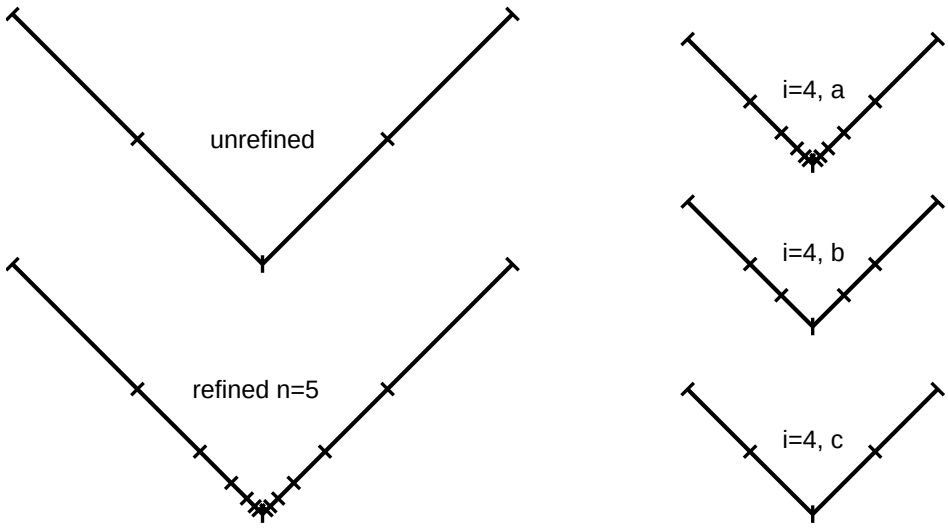

Figure 4: Placement of panels on $\Gamma^{\star}_p$ of a corner in the shape of a wedge. Upper left: Panels of the coarse mesh on $\Gamma^{\star}_p$. Lower left: Panels of an n-ply refined mesh on $\Gamma^{\star}_p$, $n=5$. Right: Panels on $\Gamma^{\star}_{pi}$ upon which the grids $\mathcal{G}_{i\rm a}$, $\mathcal{G}_{i\rm b}$, and $\mathcal{G}_{ic}$ are constructed for $i=4$ and $n=5$. Note that $\mathcal{G}_{i\rm a}$ and $\mathcal{G}_{i\rm b}$ coincide for $i=1$.

This section defines some auxiliary constructions that will help us to compute $\mathbf{R}$ of (40) in an efficient way. For simplicity we shall consider

a single corner. Triple-junctions can be treated analogously. Consider a boundary Γ with a singularity in the shape of a corner with vertex γ_p at the origin and $\Gamma_p^\star$ being a part of Γ surrounding γ_p. There are n_Γ coarse panels on Γ, four of which are located on $\Gamma_p^\star$ and the rest on Γ_p°. See Fig. 4, upper left image for an illustration of the coarse mesh on $\Gamma_p^\star$. The mesh on $\Gamma_p^{1\star}$ is now refined n times, using binary panel subdivision, so that there are $2(n+2)$ panels on $\Gamma_p^\star$. We call this is an n-ply refined mesh. The placement of panels on a 5-ply refined mesh on $\Gamma_p^\star$ is shown in the lower left image of Fig. 4.

Different grids on various subsets of $\Gamma_p^\star$ are introduced. Assume an n-ply refined mesh on $\Gamma_p^\star$ and let $\Gamma_{pi}^\star$ denote the part of $\Gamma_p^\star$ covering a mesh made up of the $2(i+2)$ innermost panels, $1 \le i \le n$. Let $\mathcal{G}_{i\mathrm{a}}$ denote a grid of $32(i+2)$ quadrature points τ_k on this latter mesh. The right images of Fig. 4 show $\Gamma_{pi}^\star$ for $i=4$ and $n=5$ and the upper right image is the mesh upon which $\mathcal{G}_{i\mathrm{a}}$ is constructed. Let $\mathcal{G}_{i\mathrm{b}}$ denote a grid of 96 quadrature points on a simply refined mesh on $\Gamma_{pi}^\star$, see the middle right image. Let $\mathcal{G}_{i\mathrm{c}}$ denote a grid of 64 quadrature points placed on a unrefined mesh on $\Gamma_{pi}^\star$, see the lower right image. On Γ_p° we just need a single grid $\mathcal{G}_p^\circ$ with $16(n_\Gamma - 4)$ points.

For each quadrature point in $\mathcal{G}_{i\mathrm{a}}$, $\mathcal{G}_{i\mathrm{b}}$, and $\mathcal{G}_{i\mathrm{c}}$ there is a corresponding Gauss-Legendre weight. Let $\mathbf{W}_{i\mathrm{a}}$, $\mathbf{W}_{i\mathrm{b}}$, and $\mathbf{W}_{i\mathrm{c}}$ be diagonal matrices containing these weights on the diagonal. We also need discrete prolongation operators, in the style of Section 5.1, on $\mathcal{G}_{i\mathrm{a}}$, $\mathcal{G}_{i\mathrm{b}}$, and $\mathcal{G}_{i\mathrm{c}}$. Let $\mathbf{P}_{i\mathrm{ab}}$ be the prolongation operator that performs polynomial interpolation from $\mathcal{G}_{i\mathrm{b}}$ to $\mathcal{G}_{i\mathrm{a}}$. Define $\mathbf{P}_{i\mathrm{ac}}$ and $\mathbf{P}_{\mathrm{bc}} = \mathbf{P}_{i\mathrm{bc}}$ in a similar fashion.

Introduce $\mathbf{T}$ as the discretized operator

$$\mathbf{T} = \mathbf{I}_{\mathrm{fin}} + \lambda_1 \mathbf{M}_{1\mathrm{fin}}^\star + \lambda_2 \mathbf{M}_{2\mathrm{fin}}^\star \,, \tag{50}$$

from (40). Let the fine grid on $\Gamma_p^\star$ be $\mathcal{G}_{n\mathrm{a}}$ and consider the square block of $\mathbf{T}$ describing interaction on $\Gamma_p^\star$. Denote this block $\mathbf{T}_{n\mathrm{a}}$ and assume that the τ_k' associated with $\tau_k \in \mathcal{G}_{n\mathrm{a}}$ and $\tau_k \in \mathcal{G}_p^\circ$ and the corresponding weights w_k are available. Then $\mathbf{T}_{n\mathrm{a}}$ can be written as a sum of several $32(n+2)\times 32(n+2)$ matrices

$$\mathbf{T}_{n\mathrm{a}} = \mathbf{I}_{n\mathrm{a}} + \mathbf{D}_{n\mathrm{a}} + \mathbf{K}_{n\mathrm{a}} \tag{51}$$

$$\mathbf{K}_{n\mathrm{a}} = \mathbf{K}_{n\mathrm{a}}^{(1)} + \mathbf{K}_{n\mathrm{a}}^{(2)} + \mathbf{K}_{n\mathrm{a}}^{(3)} + \mathbf{K}_{n\mathrm{a}}^{(4)} \,, \tag{52}$$

where for $j,k = 1,\ldots,32(n+2)$

$$(I_{n\mathrm{a}})_{jk} = \delta_{jk} \,, \quad (D_{n\mathrm{a}})_{jk} = \delta_{jk}\left(\frac{c_\kappa(\tau_j)}{b(\tau_j)} - d_\kappa(\tau_j, \mathcal{G}_{n\mathrm{a}}) - d_\kappa(\tau_j, \mathcal{G}_p^\circ)\right) \,,$$

$$(K^{(1)}_{\rm na})_{jk} = \frac{\lambda_1(\tau_j)}{\pi{\rm i}}\frac{(1-\delta_{jk})\tau'_k w_k}{(\tau_k-\tau_j)}\,, \quad (K^{(2)}_{\rm na})_{jk} = \frac{2\lambda_1(\tau_j)w_j}{\pi{\rm i}\Delta_j}B_{jk}\,,$$
$$(K^{(3)}_{\rm na})_{jk} = \frac{\lambda_2(\tau_j)}{\pi}\Im{\rm m}\left\{\frac{\tau'_k w_k}{\tau_k-\tau_j}\right\}\,, \quad (K^{(4)}_{\rm na})_{jk} = \frac{\lambda_2(\tau_j)}{\pi}\frac{\Im{\rm m}\left\{(\bar\tau_k-\bar\tau_j)\tau'_k w_k\right\}}{(\bar\tau_k-\bar\tau_j)^2}\vec{c}\,. \tag{53}$$

Here $\vec{c}$ is an operator that conjugates to the right, compare the second integral of (14) where $f(\tau)$ is conjugated, and

$$d_\kappa(\tau_j,\mathcal{G}) = \frac{1}{\pi{\rm i}b(\tau_j)}\sum_{\substack{\tau_k\in\mathcal{G}\\ \tau_k\neq\tau_j}}\frac{a_\kappa(\tau_k)\tau'_k w_k}{\tau_k-\tau_j}\,, \tag{54}$$

where w_k are weights corresponding to points on $\mathcal{G}$. The $(K^{(1)}_{\rm na})_{jk}$ and $(K^{(2)}_{\rm na})_{jk}$ elements are discussed in connection with (20). The diagonal elements $(K^{(3)}_{\rm na})_{jj}$ and $(K^{(4)}_{\rm na})_{jj}$ can be computed taking limits $\tau\to z$ in (14).

Let $\mathbf{I}_{i\rm a}$, $\mathbf{D}_{i\rm a}$, $\mathbf{K}_{i\rm a}$, $\mathbf{K}^{(1)}_{i\rm a}$, $\mathbf{K}^{(2)}_{i\rm a}$, $\mathbf{K}^{(3)}_{i\rm a}$, and $\mathbf{K}^{(4)}_{i\rm a}$ be $32(i+2)\times 32(i+2)$ submatrices extracted from the centers of the corresponding matrices with subscript 'na'. Let the 64×64 matrices $\mathbf{R}_i$ be defined as

$$\mathbf{R}_i = 2^{(n-i)}\mathbf{P}^T_{i\rm ac}\mathbf{W}_{i\rm a}(\mathbf{I}_{i\rm a}+\mathbf{D}_{i\rm a}+\mathbf{K}_{i\rm a})^{-1}\mathbf{P}_{i\rm ac}\,, \quad i=1,\ldots,n\,. \tag{55}$$

The matrix $\mathbf{W}^{-1}_{n\rm c}\mathbf{R}_n$ corresponds to the square block in $\mathbf{R}$ of (40) describing interaction on $\Gamma^\star_p$. The next section is devoted to its rapid construction.

Remark 2: The complex matrix $\mathbf{K}^{(4)}_{i\rm a}$ contains the $\vec{c}$ operator. Linear systems with system matrices of this type are, in fact, not linear. But if expanded into real systems the linear property is retained. Inverses of matrices containing the $\vec{c}$ operator should be interpreted with this in mind.

7. Recursive construction of $\mathbf{R}_n$

We seek a relation between $\mathbf{R}_i$ and $\mathbf{R}_{i-1}$ of (55). For this, we split $\mathbf{K}_{i\rm a}$ into two $32(i+2)\times 32(i+2)$ matrices

$$\mathbf{K}_{i\rm a} = \mathbf{K}^\star_{i\rm a} + \mathbf{K}^\circ_{i\rm a}\,, \quad i=1,\ldots,n\,. \tag{56}$$

Here $\mathbf{K}^\star_{i\rm a}$ contains the elements of $\mathbf{K}_{i\rm a}$ which have both indices in the set $\{17:32i{+}48\}$, the remaining elements being zero. $\mathbf{K}^\circ_{i\rm a}$ contains the elements of $\mathbf{K}_{i\rm a}$ which has at least one index in the set $\{1:16\}$ or $\{32i{+}49:32i{+}64\}$, the remaining elements being zero. One can view the non-zero elements of

$\mathbf{K}^{\circ}_{i\mathrm{a}}$ as a frame of width 16 around the non-zero elements of $\mathbf{K}^{\star}_{i\mathrm{a}}$. Let $\mathbb{F}\{\cdot\}$ denote an operator which creates a frame of width 16 of zeros around its argument. Then

$$\mathbf{K}^{\star}_{i\mathrm{a}} = \mathbb{F}\{\mathbf{K}_{(i-1)\mathrm{a}}\}\,, \quad i = 2, \ldots, n\,. \tag{57}$$

Similar splits are allowed for $\mathbf{D}_{i\mathrm{a}}$ and $\mathbf{W}_{i\mathrm{a}}$.

Now let the fine grid be constructed on an $(n-i+1)$-ply refined mesh on $\Gamma^{\star}_{p}$, so that it coincides with $\mathcal{G}_{i\mathrm{b}}$ on $\Gamma^{\star}_{pi}$, and consider the square block of $\mathbf{T}$ of (50) describing interaction on $\Gamma^{\star}_{pi}$. Denote this block $\mathbf{T}_{i\mathrm{b}}$. The elements of $\mathbf{T}_{i\mathrm{b}}$ are similar to those of $\mathbf{T}_{n\mathrm{a}}$ of (53,54), the difference being that τ_k, τ'_k, and w_k now refer to 96 points on $\mathcal{G}_{i\mathrm{b}}$ rather than to $32(n+2)$ points on $\mathcal{G}_{n\mathrm{a}}$ and that, for $i < n$, the term $d_{\kappa}(\tau_j, \mathcal{G}_{n\mathrm{a}})$ is replaced with $d_{\kappa}(\tau_j, \mathcal{G}_{i\mathrm{b}} \cup (\mathcal{G}_{n\mathrm{a}} \setminus \mathcal{G}_{i\mathrm{a}}))$. As in (51,52) we group the elements of $\mathbf{T}_{i\mathrm{b}}$ into 96×96 matrices

$$\mathbf{T}_{i\mathrm{b}} = \mathbf{I}_{i\mathrm{b}} + \mathbf{D}_{i\mathrm{b}} + \mathbf{K}_{i\mathrm{b}}\,, \quad i = 1, \ldots, n\,. \tag{58}$$

$$\mathbf{K}_{i\mathrm{b}} = \mathbf{K}^{(1)}_{i\mathrm{b}} + \mathbf{K}^{(2)}_{i\mathrm{b}} + \mathbf{K}^{(3)}_{i\mathrm{b}} + \mathbf{K}^{(4)}_{i\mathrm{b}}\,. \tag{59}$$

We split $\mathbf{K}_{i\mathrm{b}}$, $\mathbf{D}_{i\mathrm{b}}$, and $\mathbf{W}_{i\mathrm{b}}$ as in (56). Note that $\mathbf{D}^{\circ}_{i\mathrm{a}}$ and $\mathbf{D}^{\circ}_{i\mathrm{b}}$ up to rounding error have identical non-zero elements corresponding to self-interaction on the outermost panels on $\Gamma^{\star}_{pi}$. This is so since the meshes for $\mathcal{G}_{i\mathrm{a}}$ and $\mathcal{G}_{i\mathrm{b}}$ only differ for panels which are more than one panel length away from the outermost panels, see Fig. 4, and since discretization of non-regularized Cauchy integrals such as $d_{\kappa}(\tau_j, \mathcal{G})$ in (54) are accurate a panel length away from the panel on which the source points $\tau_k \in \mathcal{G}$ are located, compare the discussion about $C_1(z)$ in Section 4.

Using matrix splitting and elementary relations between the $\mathbf{P}$ operators, equation (55) can be rewritten

$$\mathbf{R}_i = 2^{(n-i)}\mathbf{P}^T_{\mathrm{bc}}\mathbf{P}^T_{i\mathrm{ab}}\Big[\big(\mathbf{W}_{i\mathrm{a}}(\mathbf{I}_{i\mathrm{a}} + \mathbf{D}^{\star}_{i\mathrm{a}} + \mathbf{K}^{\star}_{i\mathrm{a}})^{-1}\big)^{-1} + (\mathbf{D}^{\circ}_{i\mathrm{a}} + \mathbf{K}^{\circ}_{i\mathrm{a}})\mathbf{W}^{-1}_{i\mathrm{a}}\Big]^{-1}\mathbf{P}_{i\mathrm{ab}}\mathbf{P}_{\mathrm{bc}}\,, \quad i = 1, \ldots, n\,. \tag{60}$$

Now the following approximation, corresponding to prolongation of some low-rank matrix blocks, holds to high accuracy irrespective of the corner opening angle and the precise values of n, i, λ_1, and λ_2

$$(\mathbf{D}^{\circ}_{i\mathrm{a}} + \mathbf{K}^{\circ}_{i\mathrm{a}})\mathbf{W}^{-1}_{i\mathrm{a}} \approx \mathbf{P}_{i\mathrm{ab}}(\mathbf{D}^{\circ}_{i\mathrm{b}} + \mathbf{K}^{\circ}_{i\mathrm{b}})\mathbf{W}^{-1}_{i\mathrm{b}}\mathbf{P}^T_{i\mathrm{ab}}\,. \tag{61}$$

Actually, the relative error in (61) is on the order of $10^2\epsilon_{\mathrm{mach}}$ in 2-norm. This is so since the interpolating action of $\mathbf{P}_{i\mathrm{ab}}$ and $\mathbf{P}^T_{i\mathrm{ab}}$ in (61) differs from

identity only for blocks of $(\mathbf{D}^{\circ}_{i\mathrm{b}} + \mathbf{K}^{\circ}_{i\mathrm{b}})\mathbf{W}^{-1}_{i\mathrm{b}}$ describing interaction between innermost and outermost panels of the mesh upon which $\mathcal{G}_{i\mathrm{b}}$ is constructed. The row- and column elements of these blocks can be seen as expressions similar to $C_0(z)$ of (23), where z are distances between points on innermost and outermost panels and l_q is the length of the innermost panel. The separation distance is at least l_q irrespective of corner opening angle, see the middle right image of Fig. 4. The interpolation error is then on the order of $10^4\epsilon_{\mathrm{mach}}$, see Fig. 2. Since the largest singular value of $(\mathbf{D}^{\circ}_{i\mathrm{b}} + \mathbf{K}^{\circ}_{i\mathrm{b}})\mathbf{W}^{-1}_{i\mathrm{b}}$ typically is on the order of 10^2 the relative error becomes $10^2\epsilon_{\mathrm{mach}}$.

The spectral radius of the matrix

$$(\mathbf{I}_{i\mathrm{a}} + \mathbf{D}^{\star}_{i\mathrm{a}} + \mathbf{K}^{\star}_{i\mathrm{a}})^{-1}(\mathbf{D}^{\circ}_{i\mathrm{a}} + \mathbf{K}^{\circ}_{i\mathrm{a}}) \tag{62}$$

is well below unity for most corner opening angles and permissible values of λ_1 and λ_2. Use of a Neumann series argument in (60) together with (61) gives

$$\mathbf{R}_i = 2^{(n-i)}\mathbf{P}^T_{\mathrm{bc}}\Big[\big(\mathbf{P}^T_{i\mathrm{ab}}\mathbf{W}_{i\mathrm{a}}(\mathbf{I}_{i\mathrm{a}} + \mathbf{D}^{\star}_{i\mathrm{a}} + \mathbf{K}^{\star}_{i\mathrm{a}})^{-1}\mathbf{P}_{i\mathrm{ab}}\big)^{-1} + (\mathbf{D}^{\circ}_{i\mathrm{b}} + \mathbf{K}^{\circ}_{i\mathrm{b}})\mathbf{W}^{-1}_{i\mathrm{b}}\Big]^{-1}\mathbf{P}_{\mathrm{bc}}\,, \quad i = 1, \ldots, n\,. \tag{63}$$

Using

$$2^{(n-i)}\mathbf{W}_{i\mathrm{b}} = \mathbf{W}_{n\mathrm{b}}\,, \tag{64}$$

$$\mathbf{K}_{i\mathrm{b}} = \mathbf{K}_{n\mathrm{b}}\,, \tag{65}$$

where (65) only holds for a corner in the shape of a wedge, and (55,57) we can rewrite (63) on the simple recursive form

$$\mathbf{R}_i = \mathbf{P}^T_{\mathrm{bc}}\left[2\mathbb{F}\{\mathbf{R}^{-1}_{i-1}\} + (\mathbf{I}^{\circ}_{n\mathrm{b}} + \mathbf{D}^{\circ}_{i\mathrm{b}} + \mathbf{K}^{\circ}_{n\mathrm{b}})\mathbf{W}^{-1}_{n\mathrm{b}}\right]^{-1}\mathbf{P}_{\mathrm{bc}}\,, \quad i = 1, \ldots, n\,, \tag{66}$$

and take

$$2\mathbb{F}\{\mathbf{R}^{-1}_0\} = (\mathbf{I}^{\star}_{n\mathrm{b}} + \mathbf{D}^{\star}_{1\mathrm{b}} + \mathbf{K}^{\star}_{n\mathrm{b}})\,\mathbf{W}^{-1}_{n\mathrm{b}}\,, \tag{67}$$

to start the recursion.

8. Recursive construction of X

Let $\mathbf{L} = \mathbf{M}^{\star}_{2\mathrm{fin}}$. In analogy with (55) we define

$$\mathbf{X}_i = 2^{(n-i)}\mathbf{P}^T_{i\mathrm{ac}}\mathbf{W}_{i\mathrm{a}}\mathbf{L}_{i\mathrm{a}}(\mathbf{I}_{i\mathrm{a}} + \mathbf{D}_{i\mathrm{a}} + \mathbf{K}_{i\mathrm{a}})^{-1}\mathbf{P}_{i\mathrm{ac}}\,, \quad i = 1, \ldots, n\,, \tag{68}$$

where $\mathbf{W}_{nc}^{-1}\mathbf{X}_n$ corresponds to the square block in $\mathbf{X}$ of (48) describing interaction on $\Gamma_p^\star$. Matrix splitting in (68), an approximation in the style of (61)

$$\mathbf{L}_{i\mathrm{a}} \approx \mathbf{L}_{i\mathrm{a}}^\star + \mathbf{P}_{i\mathrm{ab}}\mathbf{L}_{i\mathrm{b}}^\circ\mathbf{W}_{i\mathrm{b}}^{-1}\mathbf{P}_{i\mathrm{ab}}^T\mathbf{W}_{i\mathrm{a}}\,, \tag{69}$$

use of $\mathbf{P}_{i\mathrm{ac}} = \mathbf{P}_{i\mathrm{ab}}\mathbf{P}_{\mathrm{bc}}$, a Neumann series argument, equations (55,57,61,64,68), the assumption of a corner in the shape of a wedge, and reasoning similar to that in Section 7 allow us to rewrite (68) as a recursion in tandem with (66,67)

$$\mathbf{X}_i = \mathbf{P}_{\mathrm{bc}}^T\left[\mathbb{F}\{\mathbf{X}_{i-1}\}\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\} + \mathbf{W}_{n\mathrm{b}}\mathbf{L}_{n\mathrm{b}}^\circ\mathbf{W}_{n\mathrm{b}}^{-1}\right]$$

$$\left[2\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\} + (\mathbf{I}_{n\mathrm{b}}^\circ + \mathbf{D}_{i\mathrm{b}}^\circ + \mathbf{K}_{n\mathrm{b}}^\circ)\mathbf{W}_{n\mathrm{b}}^{-1}\right]^{-1}\mathbf{P}_{\mathrm{bc}}\,, \quad i = 1,\ldots,n\,, \tag{70}$$

where

$$\mathbb{F}\{\mathbf{X}_0\} = 2\mathbf{W}_{n\mathrm{b}}^\star\mathbf{L}_{n\mathrm{b}}^\star\left(\mathbf{I}_{n\mathrm{b}} + \mathbf{D}_{1\mathrm{b}}^\star + \mathbf{K}_{n\mathrm{b}}^\star\right)^{-1}\,, \tag{71}$$

is used to start the recursion.

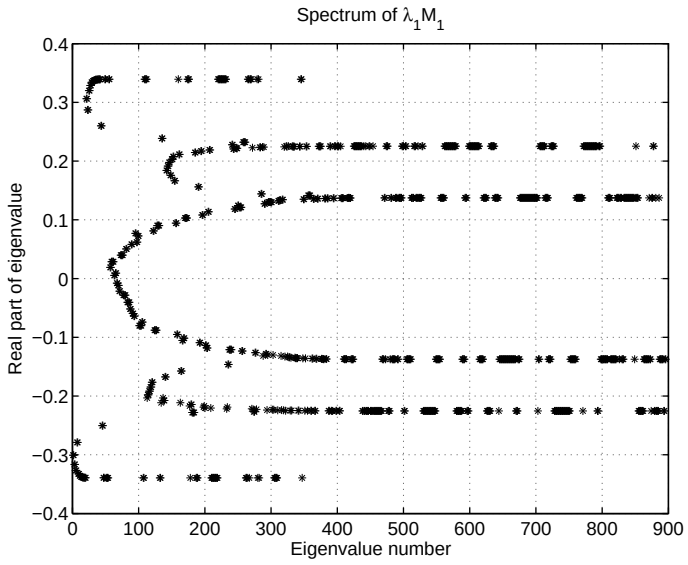

Figure 5: The operator $\lambda_1 M_1$ is discretized on a coarse mesh with 448 discretization points and expanded into a real 896×896 matrix. The real parts of its eigenvalues are shown. The geometry is that of Section 10.2 with $\lambda_1 \approx -0.225$, -0.137, and 0.339 on different parts of the boundary.

9. Spectral properties and preconditioning

System matrices arising from the discretization of Fredholm second kind integral equations with compact operators have eigenvalues clustered at unity – a property which is good for iterative solvers. The integral equations (4) and (13) do not have compact operators. Not even for smooth disjoint interfaces each with its own constant λ_1 and λ_2. The integral operators $\lambda_2 M_2$ and $\lambda_2 M_3$ are then compact, but $\lambda_1 M_1$ has a spectrum consisting of all distinct values of $\pm\lambda_1$. The situation is similar for aggregates of grains. Fig. 5 shows an example with two grains in contact and where λ_1 assumes three different values on the interfaces. One can clearly see six points of accumulation corresponding to $\pm\lambda_1$.

The presence of multiple points of eigenvalue accumulation for $\lambda_1 M_1$ may have a negative influence on the convergence rate of iterative solvers used for our discretized equations. Especially so when the range of values of λ_1 is wide. To better cluster the spectra we suggest the use of $I + \lambda_1 M_1$, or something similar, as preconditioner. A simple option is the Neumann series polynomial inverse preconditioner

$$(I + \lambda_1 M_1)^{-1} \approx I - \lambda_1 M_1 \,. \tag{72}$$

A more elaborate option is the exact inverse, given in closed form by Muskhelishvili (1953), see eq. (107.12) in his book. The inverse can be expressed in globally regularized form as

$$(I + \lambda_1 M_1)^{-1} f(z) =$$

$$\frac{Z(z)}{1-\lambda_1(z)^2}\left(f(z) - \frac{1}{\pi \mathrm{i}}\int_\Gamma \frac{(\lambda_1(z)f(\tau) - \lambda_1(\tau)f(z))\,\mathrm{d}\tau}{Z(\tau)(\tau - z)}\right), \quad z \in \Gamma\,, \tag{73}$$

where $Z(z)$ is the fundamental function (107.9) of Muskhelishvili (1953). For its efficient computation we choose

$$Z(z) = (1+\lambda_1(z))\exp\left(\frac{1}{2\pi\mathrm{i}}\sum_{k=1}^{N}\log G(\lambda_1(\tau_k))\log H(z,\tau_k^+,\tau_k^-)\right), \quad z \in \Gamma\,, \tag{74}$$

where

$$G(\lambda) = \frac{1-\lambda}{1+\lambda}\,, \tag{75}$$

and

$$H(z,\tau_k^+,\tau_k^-) = \frac{z-\tau_k^+}{z-\tau_k^-} \tag{76}$$

is a Möbius transformation whose logarithm has no branch cuts outside a circle of radius $|\tau_k^+ - \tau_k^-|/2$ centered at $(\tau_k^+ + \tau_k^-)/2$ and where $\tau_k^+ = \tau_{k+1}^-$ are points on Γ in between the discretizations points τ_k and τ_{k+1} (assuming that they lie next to each other). For straight panels and $z = \tau_k$, that is when z is inside the circle where $\log H(z, \tau_k^+, \tau_k^-)$ has branch cuts, one should take $\log \Re\mathrm{e}\{H(z, \tau_k^+, \tau_k^-)\}$ as to ensure the limit from the right in the function called $X(z)^+$ by Muskhelishvili. Note that, despite the discretization, the formula (74) is exact. The preconditioners (72) and (73) can be applied numerically to the left in (39), (41), (43), and (44). Finding an accurate quadrature for the integral in (73) requires yet more matrix compression, but for the purpose of preconditioning one can discretize (73) along the same lines as (17) is discretized in (20), possibly modifying the quadrature weights as to account for the behavior of $Z(\tau)$ close to corners and triple-junctions. We will do so in the examples of Section 11.

For future reference we loosely say that λ_2 *dominates* λ_1 when $|\lambda_2(z)| > |\lambda_1(z)|$ on most of Γ. The relevance of this concept is that if λ_2 dominates λ_1, the operators $\lambda_2 M_3$ and $\lambda_2 M_2$ control (4) and (13), and the benefit of (72) and (73), targeting the operator $\lambda_1 M_1$, becomes less pronounced. Then (45) and (46) are the most efficient equations, see Sections 10 and 11.

Remark 3: The inverse (73) assumes a much simpler form if the grains are separated. Should a grain be a hole, then $\lambda_1 + \lambda_2 = \pm 1$ on the boundary of that grain and (4) and (13) have nontrivial homogeneous solutions. See Helsing and Jonsson (2002) for techniques to remove this non-uniqueness.

10. Small-scale numerical examples

The performance of our techniques is now demonstrated for three small geometries. In the first two experiments the goal is to compute (q_1, q_2) of (18) and (19) accurately and cheaply using the GMRES iterative solver (Saad and Schultz, 1986) with a low-threshold stagnation avoiding technique (Helsing and Ojala, 2008a) for the main linear system. In the third experiment we investigate the convergence of the von Mises effective stress. The code is implemented in MATLAB version 7.6 and executed on an ordinary workstation equipped with an Intel Core2 Duo E8400 CPU at 3.00 GHz. As for reference values of (q_1, q_2) we take estimates based on a variant of the un-compressed Eq. (24), involving heavy use of local coordinates and special interpolatory quadrature, see the last paragraph of Appendix A.

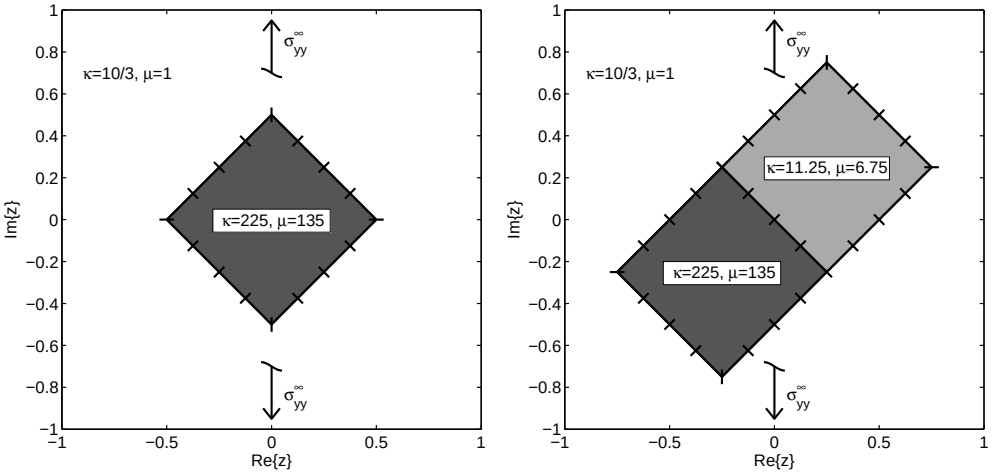

Figure 6: The single square problem and the double square problem.

10.1. The single square problem

The left image of Fig. 6 depicts a square inclusion with boundary Γ of length $2\sqrt{2}$ centered at the origin and embedded in an infinite plane. Unit stress, $\sigma_{yy}^{\infty} = 1$, is applied. The elastic moduli are chosen as in earlier work (Eischen and Torquato, 1993; Greengard and Helsing, 1998) for boron-epoxy composites under plane strain conditions and give $\lambda_1 \approx -0.2253$ and $\lambda_2 \approx -0.7566$. Four equi-sized coarse quadrature panels are placed on each side of the square.

First fine meshes are created by repeated subdivision of panels adjacent to corners and the un-compressed equations (24) and (25), discretized according to Section 4, are solved. The quantities (q_1, q_2) are computed via discretizations of (18) and (19). The compressed equations (43) and (44) and their right preconditioned variants (45) and (46) are also implemented along with the discretizations (47,49) for (q_1, q_2), and the recursions (66,67) for $\mathbf{R}$, and (70,71) for $\mathbf{X}$. Here special interpolatory quadrature is used.

Fig. 7, left image, shows convergence for (q_1, q_2). Convergence for the un-compressed equations is plotted against the number of subdivisions of the innermost coarse panels of each corner. Convergence for the compressed equations is plotted against the number of recursion steps in $\mathbf{R}$ and $\mathbf{X}$. One can see that all equations give convergence to the same values of (q_1, q_2) and that the accuracy produced by the compressed equations is very high, thereby illustrating the consistency and the validity of various equations, discretizations, approximations, and recursions of previous sections. The

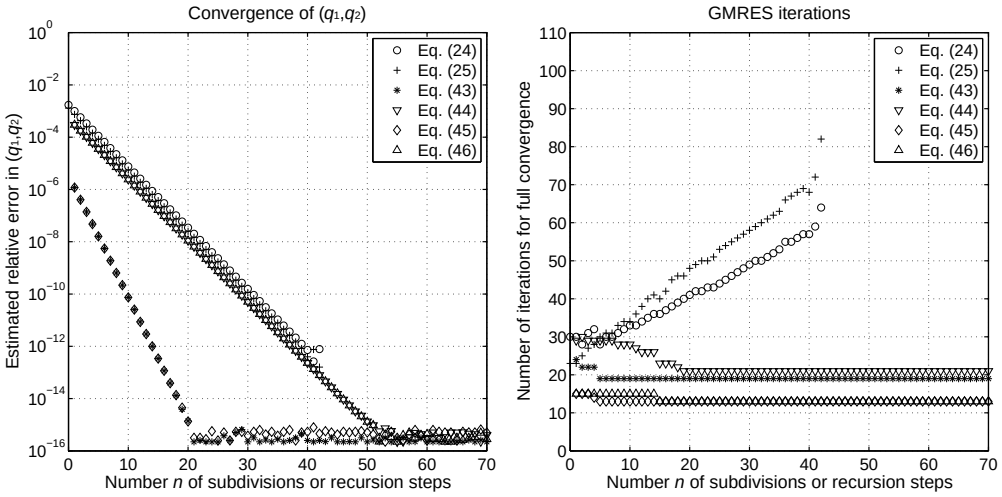

Figure 7: The single square problem. Left: Convergence of (q_1, q_2). Reference values are $q_1 = -0.3949603789008513$ and $q_2 = -0.1974624767099451$. Eqs. (43) and (45) use special interpolatory quadrature and therefore converge faster than Eq. (24). Right: the number of GMRES iterations needed to reach a stopping criterion threshold of ϵ_{mach} in the estimated relative residual.

high final accuracy also shows that there is nothing ill-conditioned about computing (q_1, q_2) on a domain with non-smooth interfaces.

The un-compressed equations (24) and (25), despite the different regularities of $\Omega(z)$ and $\omega(z)$, behave similarly. With 40 subdivisions the relative error in (q_1, q_2) is about 10^{-12}. By then, the number of GMRES iterations needed is in the range 55–70, see right image of Fig. 7. The compressed equations (43) and (44) behave much better. They are computationally cheaper than their un-compressed counterparts. The un-compressed equations at n subdivisions result in linear systems with $256 + 64n$ unknowns while the compressed equations have 256 unknowns irrespective of the number of recursion steps. Furthermore, the compressed equations exhibit lower relative errors and fewer GMRES iterations. The right preconditioned variants (45) and (46) are even more efficient than (43) and (44) for this problem. In terms of achievable accuracy they are on par with (43) and (44), but in terms of GMRES iterations they are almost twice as fast. This is in accordance with the discussion in Section 9 about (45) and (46) being preferable when λ_2 dominates λ_1. The total wall-clock time for obtaining the solution at $n = 40$ subdivisions or recursion steps, including the setup of various matrices (all system matrices in this experiment are formed explicitly), is

around 40 seconds for (24), around 55 seconds for (25), and between 0.6 and 0.8 seconds each for (43), (44), (45) and (46).

Further experiments, not illustrated, show that beyond 76 recursion steps $\mathbf{R}$ of (43) has converged to within ϵ_{mach} in Frobenius norm and nothing more happens. The same holds for $\mathbf{R}$ of (44) beyond 84 recursion steps.

10.2. The double square problem

The right image of Fig. 6 depicts an extension of the single square problem. The original square is translated by $-0.25 - 0.25$i. A new square is added. The values of (λ_1, λ_2) are approximately $(-0.2253, -0.7566)$, $(-0.1373, -0.5542)$ and $(0.3393, 0.5655)$ on the different interfaces, so λ_2 again dominates λ_1. In addition to testing the equations used in Section 10.1, we shall also apply the numerical preconditioner (73) to (43) and to (44).

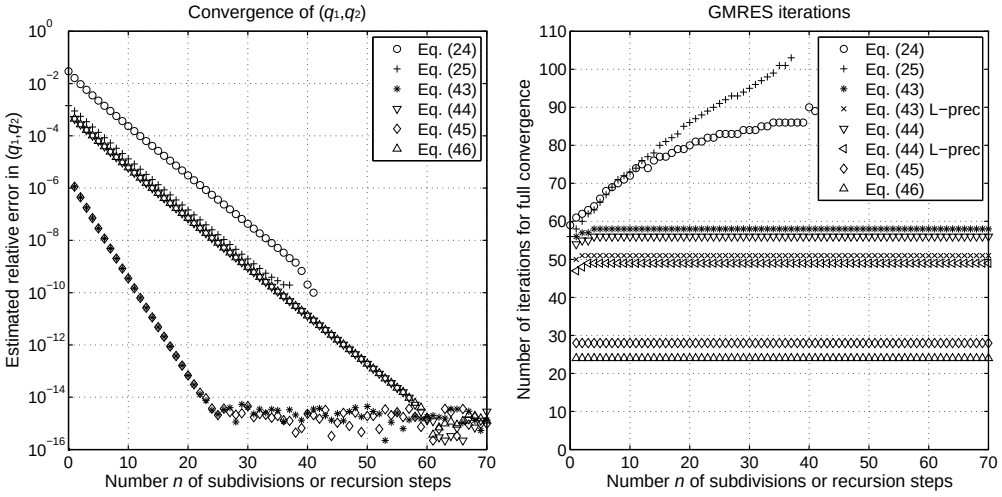

Figure 8: The double square problem. Left: convergence of (q_1, q_2). Reference values are $q_1 = -0.675066104519704$ and $q_2 = -0.336029323458141$. Right: the number of GMRES iterations needed to reach a stopping criterion threshold of ϵ_{mach} in the estimated relative residual. 'L-prec' refers to application of the preconditioner (73) from the left.

Fig. 8, analogous to Fig. 7, shows that (24), (25), and (43-46) all give convergence to the same values of (q_1, q_2). Again, the convergence of (44) and (46) are almost identical, as is the convergence of (43) and (45). Again, the compressed equations exhibit a far more stable behavior in terms of GMRES iterations than do the un-compressed equations. The application of the preconditioner (73) to (43) and to (44), only shown in the right image of Fig. 8, gives a 10% improvement in terms of GMRES iterations.

The differences between (43-46) become more pronounced as we change from the single- to the double square problem. Eqs. (43) and (45), which have more regular solutions, need fewer recursion steps for convergence than do (44) and (46), which have less regular solutions. The right preconditioned variants (45) and (46) need far fewer GMRES iterations than do the merely compressed equations (43) and (44). Still, the most important conclusion may be that using any of the equations (43-46), the less symmetric double square problem with corners, triple-junctions, and discontinuous $\lambda_1(z)$ and $\lambda_2(z)$ is only slightly more difficult to solve than the highly symmetric single square problem with only corners and constant λ_1 and λ_2.

10.3. The three-grain problem

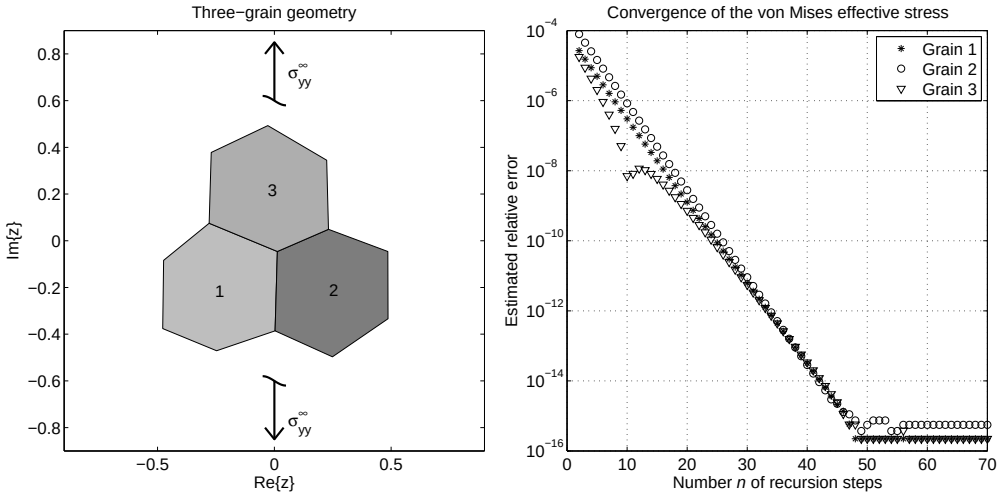

Figure 9: Convergence of the von Mises effective stress at the centroids of three hexagonal grains, as a function of the number of recursion steps in (66,67). The geometric and material properties are given in Tab. 1 and the reference solutions are $(\sigma_v)_{\mathrm{ref}} = [1.143007309293997, 1.200686919310857, 1.167025937518209]$.

As a last small-scale experiment we investigate the convergence of the von Mises effective stress computed on a geometry consisting of three hexagonal grains. Plane strain conditions are assumed, and the geometric and material properties of the grains are given in Tab. 1. The surrounding material has Young's modulus $E_0 = 30$ GPa and Poission's ratio $\nu_0 = 0.2$. Unit stress, $\sigma^{\infty}_{yy} = 1$, is applied. Under plane strain conditions, the von Mises effective

	Vertices			E_k (GPa)	ν_k
1	-0.478-0.376i	-0.247-0.471i	0.003-0.386i	78	0.42
	0.013-0.046i	-0.279+0.075i	-0.474-0.084i		
2	0.003-0.386i	0.249-0.497i	0.487-0.334i	200	0.3
	0.486-0.046i	0.232+0.049i	0.013-0.046i		
3	-0.279+0.075i	0.013-0.046i	0.232+0.049i	100	0.33
	0.224+0.345i	-0.028+0.493i	-0.270+0.378i		

Table 1: Geometric and material properties for the three grains. For the surrounding material we use $E_0 = 30$ GPa and $\nu_0 = 0.2$.

stress is given by

$$\sigma_v = \sqrt{(1-\nu(1-\nu))\left(\sigma_{xx}^2+\sigma_{yy}^2\right) - 3\left(\sigma_{xx}\sigma_{yy}-\sigma_{xy}^2\right)}\,. \tag{77}$$

In order to evaluate this quantity, we first compute the weight-corrected density $\hat{\boldsymbol{\Omega}}_{\text{coa}}$ from (44), using 4 quadrature panels per grain edge and n steps in the recursion (66,67) with n varying between 2 and 70. Having obtained $\hat{\boldsymbol{\Omega}}_{\text{coa}}$, we then compute the stress components σ_{xx}, σ_{yy}, and σ_{xy} via (2,3) and (10,11) at the centroids of the three grains, which in turn give the von Mises effective stress via (77). As reference solutions we use effective stresses computed using 8 panels per grain edge and 80 recursion steps. They are

$$(\sigma_v)_{\text{ref}} = [1.143007309293997, 1.200686919310857, 1.167025937518209]$$

at the three centroids, respectively. The geometry and the relative error as a function of the number of recursion steps is shown in Fig. 9. As can be seen, the values have converged to 15 digits at 50 recursion steps.

11. Large-scale numerical examples

This section demonstrates the performance of different combinations of equations and preconditioners, introduced in preceding sections, on a larger scale. We also discuss aspects of applicability, that is, under what circumstances to use what equations and preconditioners in order to achieve the best results. The goal is to compute the quantity (q_1, q_2), given in discretized form by (47) and (49), for inclusions consisting of perturbed hexagonal grains. The experiments are performed in the MATLAB environment,

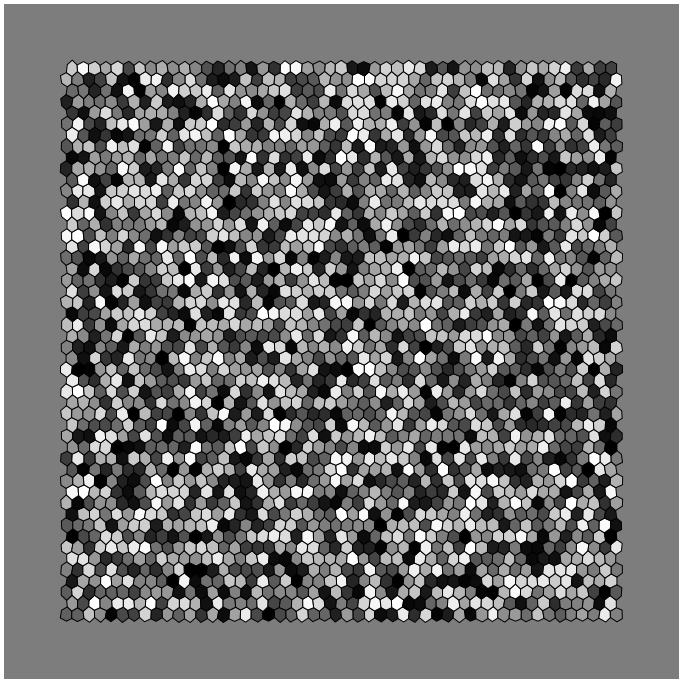

Figure 10: A perturbed honeycomb structure with 2475 grains. The coloring is based on the Young's modulus. Darker colors indicate stiffer materials.

with some time-critical functions written in C. Linear systems are solved using GMRES as in the small-scale examples, but matrix-vector multiplication is now accelerated with fast multipole techniques (Greenbaum et al., 1992). The computer is an ordinary workstation equipped with an Intel Core i7 CPU running at 2.66 GHz.

A number N_{gr} of hexagonal grains are placed in a honeycomb-like pattern, within the unit square. The structure is randomized slightly by moving each vertex a distance of one tenth of the length of a grain edge in a random direction. This is done to avoid symmetries that may allow for a simpler solution. The largest structure used, with $N_{\text{gr}} = 2475$ is displayed in Fig. 10. The material properties of the grains are assigned as follows:

- The Young's modulus E_k for grain k is given by $E_k = 10^{\gamma}$ where γ is a uniformly distributed random variable in the interval $[4, 6]$. The geometric mean of the E_k is then approximately 10^5.

- The Poisson's ratio ν_k for grain k is uniformly distributed in the in-

terval $[-0.99, 0.49]$, encompassing a wide range of materials, while avoiding the most pathological ones.

The surrounding material has $E_0 = 10^5$ and $\nu_0 = 0$. The bulk and shear moduli κ_k and μ_k of each grain is computed according to different formulas depending on whether we simulate plane strain or plane stress conditions. Unit stress $\sigma_{yy}^{\infty} = 1$ is applied in all examples.

The coarse mesh over Γ is constructed by distributing n_{e} panels of equal length in parameter as well as arc length on the shortest grain edge in the structure. Panels on the other edges are distributed so that all the panels in the mesh have approximately equal length. That is, each edge in the structure consists of at least n_{e} panels. Each panel, in turn, has 16 Gauss-Legendre nodes with accompanying suitably scaled weights.

After the mesh is constructed the blocks of $\mathbf{R}$ and $\mathbf{X}$ are computed. Let $\mathbf{R}_p$ and $\mathbf{X}_p$ denote the block of $\mathbf{R}$ and $\mathbf{X}$ describing interaction on $\Gamma_p^\star$. For each corner and triple-junction these matrix blocks are constructed using the recursion (66,67) for $\mathbf{R}_p$ and (70,71) for $\mathbf{X}_p$. The techniques in Appendix A and B are applied for efficiency. The recursion for $\mathbf{R}_p$ is run until

$$\frac{\|(\mathbf{R}_p)_i - (\mathbf{R}_p)_{i-1}\|_{\mathrm{F}}}{\|(\mathbf{R}_p)_{i-1}\|_{\mathrm{F}}} \leq 10\epsilon_{\mathrm{mach}}\,, \tag{78}$$

where $(\mathbf{R}_p)_i$ is the ith step in the recursion of $\mathbf{R}_p$, and 'F' indicates the Frobenius norm. An analogous condition is used for $\mathbf{X}_p$. Since the fast multipole method is used for the system matrices, blocks corresponding to $\mathbf{M}_{1\mathrm{coa},p}^\star, \mathbf{M}_{2\mathrm{coa},p}^\star$ and $\mathbf{M}_{3\mathrm{coa},p}^\star$ need to be subtracted. Furthermore, in the post-processor (49), $\mathbf{X}$ does not appear by itself but is multiplied by $\mathbf{R}^{-1}$. To save time and memory, compositions of these matrix blocks are precomputed and stored. For example, in the setting of (43), the compositions

$$\mathbf{B}_{\mathrm{R},p} = \mathbf{R}_p^{-1} - \mathbf{I} - \lambda_1 \mathbf{M}_{1\mathrm{coa},p}^\star - \lambda_2 \mathbf{M}_{2\mathrm{coa},p}^\star \tag{79}$$

$$\mathbf{B}_{\mathrm{X},p} = \mathbf{X}_p \mathbf{R}_p^{-1} - \mathbf{M}_{2\mathrm{coa},p}^\star \tag{80}$$

are computed for each corner and triple-junction. The former is stored in RAM and the latter on disk, since it is needed in the post-processor only. Analogous expressions are used in the setting (44).

We take Eqs. (43) and (44) as our primary choice of equations. In addition, we apply the preconditioners (72) and (73) to the left in (43) and (44) for a total of six setups. These are tested under both plane strain and plane stress conditions. It should be mentioned that the right inverse preconditioned equations (45) and (46) are competitive for certain materials.

As a rule of thumb, λ_2 dominates λ_1 when $\nu_0 > 0$, $\nu_k > 0$ for all k, which is true for most common materials. In this case, Eqs. (45) and (46) are favorable in that they require about as many GMRES iterations as the left preconditioned equations (43) and (44), while each iteration is faster. In the present setting with highly general materials, however, Eqs. (45) and (46) suffer and were omitted.

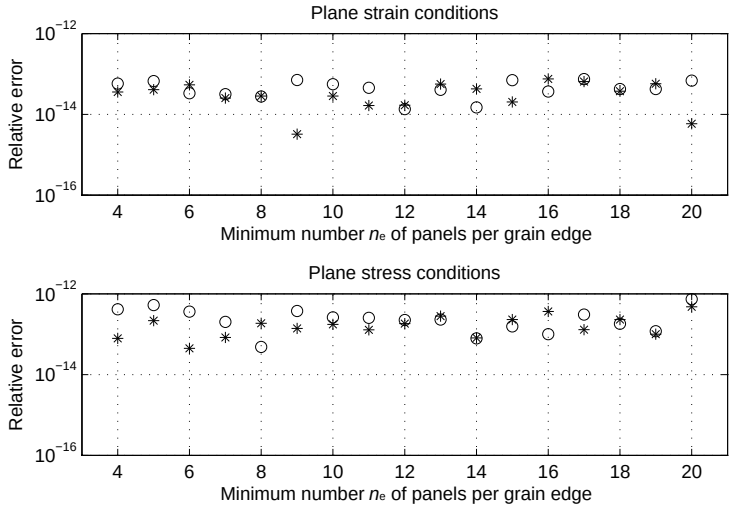

Figure 11: Estimated relative errors in (q_1, q_2) using different panel resolutions for a composite with 218 grains. Stars indicate Eq. (43) and circles Eq. (44).

We first demonstrate the behavior of Eqs. (43) and (44) under increased mesh resolution. Using a structure with $N_{\mathrm{gr}} = 218$ grains we vary the minimum number of panels n_{e} from 4 to 20 and calculate (q_1, q_2) in each instance under both plane strain and plane stress conditions. As reference solution we choose the arithmetic mean of the eight solutions computed with $n_{\mathrm{e}} = 5, 6, 7, 8$. Fig. 11 shows the relative errors. One can see that the methods are stable under overresolution, that the quantity (q_1, q_2) has converged already at $n_{\mathrm{e}} = 4$, and that the error is very low also for problems significantly more complex than those of Section 10.

Next, we vary the number of grains N_{gr} from 1 to 2475 while keeping n_{e} fixed. The discretization of the coarse grid has $n_{\mathrm{e}} = 4$ in light of the preceding paragraph, but reference quantities (q_1, q_2) are also calculated using $n_{\mathrm{e}} = 5$. All six setups are tested. Fig. 12 for plane strain and Fig. 13 for plane stress are similar. The errors are roughly the same, albeit somewhat

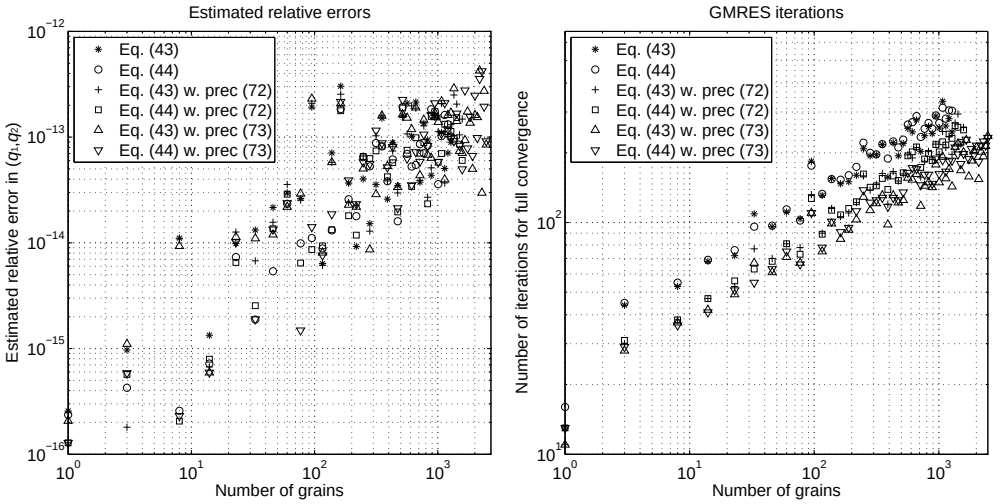

Figure 12: Plane strain conditions. Left: The estimated relative errors of the computed quantity (q_1, q_2) as a function of the number of grains $N_{\rm gr}$. Right: The number of GMRES iterations needed to reach a stopping criterion threshold of $\epsilon_{\rm mach}$ in the estimated relative residual.

higher under plane stress conditions. The number of GMRES iterations are also similar. This is a very favorable characteristic; the setups are equally stable and usable regardless of whether plane strain or plane stress conditions are assumed, and regardless of whether layer densities related to displacements or stresses are sought. The most important difference is that for plane strain conditions, the preconditioner (73) should be used in order to minimize the number of GMRES iterations, while for plane stress conditions (72) is preferable for the same reason. The fact that the number of GMRES iterations increases with the number of grains is a concern, but experiments with more homogeneous materials suggest that the number of iterations will eventually settle around an upper limit.

As for an example of memory and time requirements we choose the largest geometry with $N_{\rm gr} = 2475$, plane stress conditions, and Eq. (43) with the preconditioner (72). The entire computation takes about two hours. Time-wise, the computation is dominated by the calculation of (79), (80) and the GMRES solver. The mean number of recursion steps needed for the blocks $\mathbf{R}_p$ and $\mathbf{X}_p$ to satisify (78) are 50 and 47, respectively, and the total block computation accounts for 38% of the solution time. GMRES, requiring 184 iterations for the estimated relative residual to drop below

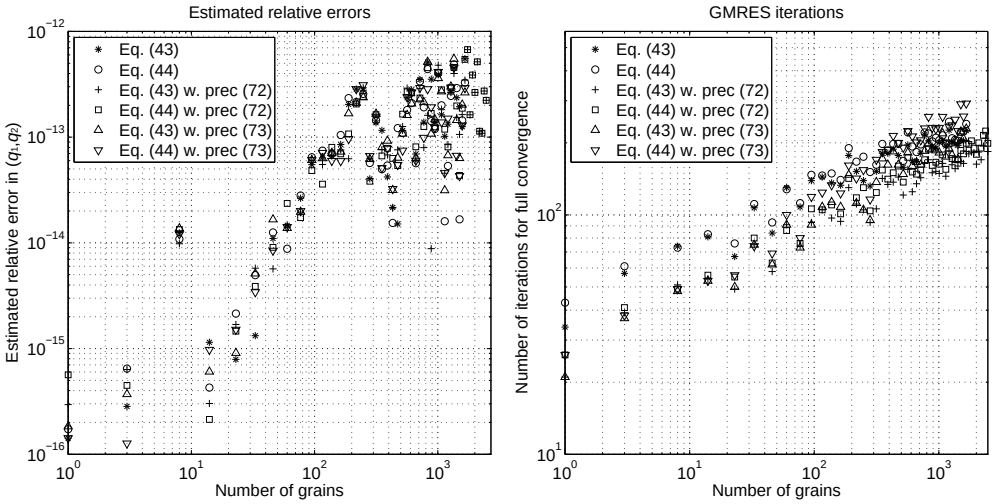

Figure 13: Plane stress conditions. Left: The estimated relative errors of the computed quantity (q_1, q_2) as a function of the number of grains N_{gr}. Right: The number of GMRES iterations needed to reach a stopping criterion threshold of ϵ_{mach} in the estimated relative residual.

ϵ_{mach}, accounts for around 60%. Precomputed blocks and Krylov subspace vectors in GMRES also dominate the memory required. There are 5149 corners and triple-junctions. At each of these the block (79) needs to be stored in RAM for later use in GMRES. These blocks, together, require about 1.5 GB of RAM. Furthermore, the computational grid consists of 693952 points, and with the density $\hat{\boldsymbol{\omega}}_{\mathrm{coa}}$ being complex this means that each Krylov vector in the GMRES solver uses approximately 1.1 MB of RAM. With 184 GMRES iterations, about 2 GB of memory is required. One should keep in mind that this is a very large and complicated geometry which, by necessity, translates into a high use of computational resources. The point is that even these very difficult geometries can be handled efficiently enough to be feasible for off the shelf workstations. In addition, these numbers correspond to solving the problem at hand to maximum accuracy. It should be pointed out that the solver components are tunable in terms of desired precision, saving time and storage accordingly. Setting the stopping criterion in (78) and in the GMRES solver to some specified tolerance $\epsilon_{\mathrm{tol}} > \epsilon_{\mathrm{mach}}$ saves time and lowers the memory required by the Krylov vectors. The amount of memory required by the precomputed blocks can be lowered by computing the singular value decomposition of the blocks (79) and only store

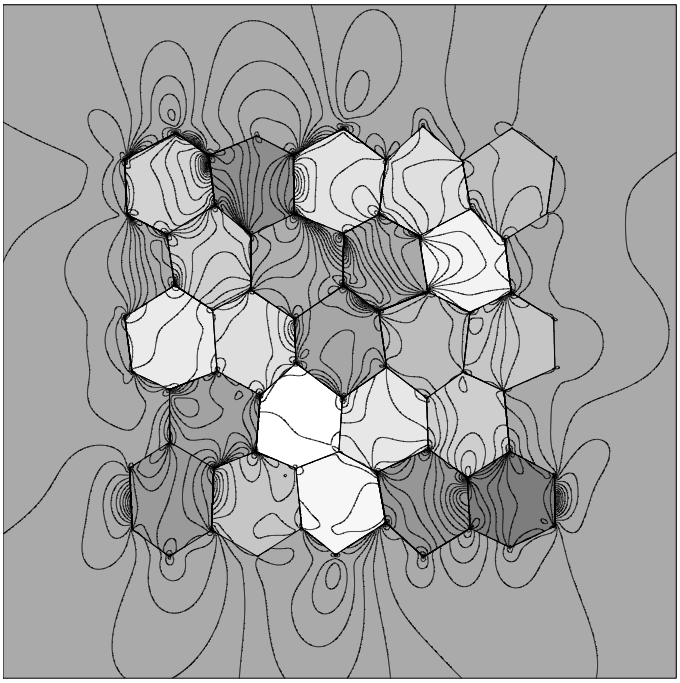

Figure 14: A contour plot of the base-10 logarithm of the von Mises effective stress. The coloring is as in Fig. 10.

the singular vectors corresponding to singular values greater than $100\epsilon_{\text{tol}}$. Setting $\epsilon_{\text{tol}} = 10^{-8}$, for example, roughly cuts solution time and memory requirements in half and the relative error in (q_1, q_2) is kept around 10^{-8} as specified.

Now, we briefly demonstrate an extension to the convergence experiment in Section 10.3 by calculating the von Mises effective stress over an entire domain. We choose a geometry of moderate size – a honeycomb structure consisting of 23 grains. The method used is, basically, the same as in Section 10.3, with the addition that close to grain boundaries we now utilize special-purpose interpolatory quadrature (Helsing and Ojala, 2008a) to improve accuracy. The result is displayed in Fig. 14.

To conclude this section, we briefly summarize the findings. Firstly, there is little difference in the performance of (43) and (44). That is, one is free to solve for quantities corresponding to stresses or displacements, whichever best suits a given problem from a modeling point of view. Secondly, provided the appropriate left preconditioner is used, there is little

difference in performance between plane strain and plane stress conditions. The suggested left preconditioners are :

- the Neumann series inverse preconditioner (72) for plane stress,
- the Muskhelishvili inverse preconditioner (73) for plane strain.

Thirdly, these setups perform well for structures consisting of highly general materials, with $\nu = (-1, 0.5)$. For more common materials, where $\nu > 0$, further efficiency can be gained by using the compressed right inverse preconditioned equations (45) and (46). To sum up: the combination of recursive compression with a suitable left preconditioner is a robust, flexible, efficient, and highly accurate tool in planar elasticity.

12. Outlook and conclusion

This paper is about a recursive compression technique for the accurate resolution of sharp corners and triple junctions in the context of solving integral equations for problems in linear elasticity. Its description requires a fair amount of detail. Once implemented, however, the technique is a versatile and powerful tool.

To limit the size of the paper we have focused on an elastic plane with a multiphase granular inclusion and boundary conditions at infinity, where we compute some functionals of the solution. In the numerical examples we, further, specialize to boundaries made up of straight edges. Objects with curved boundaries can be treated with almost the same ease, as demonstrated for Laplace's equation in Section 10.2 of (Helsing and Ojala, 2008b). The modification in the present context is that (65) no longer holds. The recursion (63) must be used, rather than the simplified (66). More work is required to obtain the d_κ quantities, see Appendix A. Perhaps, then, it would pay off to replace the global regularization (17) with panelwise evaluation throughout $\Gamma_p^\star$.

Should one need to reconstruct the pointwise values of $\omega(z)$ and $\Omega(z)$ on the fine grid from the computed values of their transformed counterparts on the coarse grid, this can be done by, in a sense, running (63) backwards. The pointwise discretization error is the same as if un-compressed equations were used from the start. We may return to this in future work, possibly in combination with more compact notation for describing the entire scheme.

Other geometries, boundary conditions, and output, such as multi-wedge stress intensity factors and plots of elastic fields in finite bodies with mixed boundary conditions, could in principle also be treated. Stress intensity

factors can be extracted via curve fitting of singular basis functions to reconstructed layer densities. Singular exponents may be computed via multi-wedge eigenvalue analysis or determined in the fit itself. Mixed boundary conditions require the derivation of new integral equations. This, too, we leave for future work.

One may argue that real materials seldom have corners that are atomically sharp, that linear elasticity is not valid at atomic length scales, that grains are never perfectly bonded in triple junctions and that one, therefore, somehow should smooth out geometrical difficulties that arise in material modeling as to avoid such 'unphysical complications'. We believe that while some smoothing could perhaps increase realism it will not lead to simpler problems from a computational viewpoint. On the contrary, non-sharp interfaces introduce new numerical complications. Furthermore, this paper shows that the presence of corners and triple junctions in a geometry does not generally increase the conditioning of an underlying elastostatic problem. Why, then, should it increase the computational difficulty in any significant way? Indeed, apart from the need for extra storage, non-smooth boundaries can be treated with the same ease as smooth boundaries. This is the main conclusion of our paper.

Acknowledgements

This work was supported by the Swedish Research Council under contract 621-2007-6234.

Appendices

A. Efficient computation of $\mathbf{D}^{\circ}_{i\mathrm{b}}$, $\mathbf{D}^{\star}_{1\mathrm{b}}$, and $\mathbf{K}^{\star}_{n\mathrm{b}}$

While all matrices entering into the recursion (66,67) have been defined, a discussion about the efficient computation of $\mathbf{D}^{\circ}_{i\mathrm{b}}$, $\mathbf{D}^{\star}_{1\mathrm{b}}$, and $\mathbf{K}^{\star}_{n\mathrm{b}}$ is in order. For this we need another two sets of discretization points. Let $\mathcal{G}^{1\star}_{p}$ denote a set of 32 points on the coarse panels on $\Gamma^{1\star}_{p}$, see Fig. 3. Let $\mathcal{G}^{\circ}_{i\mathrm{b}}$ denote the 32 points on the outermost two panels in the set $\mathcal{G}_{i\mathrm{b}}$, see Fig. 4.

The 32 non-zero elements of $\mathbf{D}^{\circ}_{i\mathrm{b}}$ are needed in each recursion step i. In the last step, $i = n$, they can be written

$$\frac{c_\kappa(\tau_j)}{b(\tau_j)} - d_\kappa(\tau_j, \mathcal{G}_{n\mathrm{b}}) - d_\kappa(\tau_j, \mathcal{G}^{\circ}_{p})\,, \quad \tau_j \in \mathcal{G}^{\circ}_{n\mathrm{b}}\,, \tag{81}$$

see (53). Row summation in $\mathbf{K}_{n\mathrm{b}}^{(1)\circ}$ gives $d_\kappa(\tau_j, \mathcal{G}_{n\mathrm{b}})$ and row summation in $\lambda_1 \mathbf{M}_{1\mathrm{coa}}^{\circ}$ gives $d_\kappa(\tau_j, \mathcal{G}_p^{\circ})$. When $i < n$, the non-zero elements can be written

$$\frac{c_\kappa(\tau_j)}{b(\tau_j)} - d_\kappa(\tau_j, \mathcal{G}_{i\mathrm{b}}) - d_\kappa(\tau_j, \mathcal{G}_{n\mathrm{a}} \setminus \mathcal{G}_{i\mathrm{a}}) - d_\kappa(\tau_j, \mathcal{G}_p^{\circ}), \quad \tau_j \in \mathcal{G}_{i\mathrm{b}}^{\circ}, \tag{82}$$

The quantities $d_\kappa(\tau_j, \mathcal{G}_{n\mathrm{a}} \setminus \mathcal{G}_{i\mathrm{a}})$ and $d_\kappa(\tau_j, \mathcal{G}_p^{\circ})$ can, in general, be computed via interpolation and recursion in the opposite direction of (63). When Γ is a collection of line segments, (82) is independent of i and simple options exists for all its terms. Independence of i is shown as follows: The first term of (82) is a constant. The second term is invariant under the change $d_\kappa(\tau_j, \mathcal{G}_{i\mathrm{b}})$, $\tau_j \in \mathcal{G}_{i\mathrm{b}}^{\circ} \rightarrow d_\kappa(\tau_j, \mathcal{G}_{(i-1)\mathrm{b}})$, $\tau_j \in \mathcal{G}_{(i-1)\mathrm{b}}^{\circ}$ and thus independent of i. From (16) we have

$$\frac{1}{\pi \mathrm{i} b(z)} \int_{\Gamma_1+\Gamma_2+\Gamma_3} \frac{a_\kappa(\tau)\,\mathrm{d}\tau}{\tau - z} = \frac{c_\kappa(z)}{b(z)}, \quad z \in \mathcal{G}_{i\mathrm{b}}^{\circ}, \tag{83}$$

with $\Gamma_1 = \Gamma \setminus \Gamma_{p(i+1)}^{\star}$, $\Gamma_2 = \Gamma_{p(i+1)}^{\star} \setminus \Gamma_{pi}^{\star}$, and $\Gamma_3 = \Gamma_{pi}^{\star}$. The functions $b(z)$ and $c_\kappa(z)$ are constant on each edge, see (7) and (15), so the right hand side of (83) is independent of i and therefore also the left hand side. Now the integrals over Γ_2 and Γ_3 are independent of i and therefore also the integral over Γ_1. Furthermore, the discretization of the integral over Γ_1 is accurate according to Fig. 1 and therefore independent of i. Clearly, the discretization of the integral over Γ_2 is independent of i. The sum of the discretizations of the integrals over Γ_1 and Γ_2 corresponds to the the sum of the third and the fourth term of (82).

The elements in (82) can now be computed choosing $i = n-1$. For the second term we observe that $d_\kappa(\tau_j, \mathcal{G}_{(n-1)\mathrm{b}})$, $\tau_j \in \mathcal{G}_{(n-1)\mathrm{b}}^{\circ}$ and $d_\kappa(\tau_j, \mathcal{G}_{n\mathrm{b}})$, $\tau_j \in \mathcal{G}_{n\mathrm{b}}^{\circ}$ are the same. The latter values were computed in connection with (81). For the third term we observe that $\mathcal{G}_{n\mathrm{a}} \setminus \mathcal{G}_{(n-1)\mathrm{a}} = \mathcal{G}_{n\mathrm{b}}^{\circ}$ and that $\mathcal{G}_{(n-1)\mathrm{b}}^{\circ}$ is a subset of $\mathcal{G}_{n\mathrm{b}}$. For the fourth term we interpolate values of $d_\kappa(\tau_j, \mathcal{G}_p^{\circ})$ at $\tau_j \in \mathcal{G}_p^{1\star}$, available by row summation in $\lambda_1 \mathbf{M}_{1\mathrm{coa}}^{\circ}$, to points in $\mathcal{G}_{(n-1)\mathrm{b}}^{\circ}$. See the last paragraph of Section 5.2. In conclusion, to evaluate (81) and (82) when Γ consists of straight lines, it suffices to do summation in $\mathbf{K}_{n\mathrm{b}}^{(1)\circ}$ and $\lambda_1 \mathbf{M}_{1\mathrm{coa}}^{\circ}$ followed by interpolation . The elements of $\mathbf{K}_{n\mathrm{b}}^{(1)\circ}$ are independent of n and we choose $n = 1$.

The 64 non-zero elements of $\mathbf{D}_{1\mathrm{b}}^{\star}$

$$\frac{c_\kappa(\tau_j)}{b(\tau_j)} - d_\kappa(\tau_j, \mathcal{G}_{1\mathrm{b}}) - d_\kappa(\tau_j, \mathcal{G}_{n\mathrm{a}} \setminus \mathcal{G}_{1\mathrm{a}}) - d_\kappa(\tau_j, \mathcal{G}_p^{\circ}), \quad \tau_j \in \mathcal{G}_{1\mathrm{b}} \setminus \mathcal{G}_{1\mathrm{b}}^{\circ}, \tag{84}$$

are needed in the first recursion step. Again, we specialize to Γ being a collection of line segments. Scale invariance gives that (84) is independent of n and we choose $n=1$. The second term then corresponds to summation in $\mathbf{K}_{n\mathrm{b}}^{(1)}$ with $\tau_j \in \mathcal{G}_{n\mathrm{b}} \setminus \mathcal{G}_{n\mathrm{b}}^{\circ}$. The third term vanishes. For the fourth we interpolate values of $d_\kappa(\tau_j, \mathcal{G}_p^{\circ})$ at $\tau_j \in \mathcal{G}_p^{1\star}$ to points in $\mathcal{G}_{n\mathrm{b}} \setminus \mathcal{G}_{n\mathrm{b}}^{\circ}$. So, again, it suffices to do summation in $\mathbf{K}_{n\mathrm{b}}^{(1)}$ and $\lambda_1 \mathbf{M}_{1\mathrm{coa}}^{\circ}$ followed by interpolation.

The entries of $\mathbf{K}_{n\mathrm{b}}^{\star}$, which come from $\mathbf{K}_{n\mathrm{b}}^{(1)\star}$, $\mathbf{K}_{n\mathrm{b}}^{(3)\star}$ and $\mathbf{K}_{n\mathrm{b}}^{(4)\star}$ and describe interaction of neighboring panels which meet at a vertex, can be computed more accurately using special interpolatory quadrature (Helsing and Ojala, 2008a). For $\lambda_1 M_1$ of (17) this corresponds to *panelwise evaluation*, rather than to global regularization.

B. Speedup of the recursions and storage

Each recursion step of (66) involves a fair amount of matrix inversion and matrix-matrix multiplication. The complex matrix $\mathbf{R}_i$ is 64×64, the complex matrix within square brackets is 96×96 and $\mathbf{P}_{\mathrm{bc}}$ is 96×64, even though a majority of the elements are zero. Using the Schur-Banachiewicz inverse formula for a partitioned matrix (Henderson and Searle, 1981) one can show

$$\left[\begin{array}{cc} \mathbf{P}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{array} \right] \left[\begin{array}{cc} \mathbf{A}^{-1} & \mathbf{U} \\ \mathbf{V} & \mathbf{D} \end{array} \right]^{-1} \left[\begin{array}{cc} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{array} \right] =$$

$$\left[\begin{array}{cc} \mathbf{P}^T\mathbf{A}\mathbf{P} + \mathbf{P}^T\mathbf{A}\mathbf{U}(\mathbf{D}-\mathbf{V}\mathbf{A}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}\mathbf{P} & -\mathbf{P}^T\mathbf{A}\mathbf{U}(\mathbf{D}-\mathbf{V}\mathbf{A}\mathbf{U})^{-1} \\ -(\mathbf{D}-\mathbf{V}\mathbf{A}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}\mathbf{P} & (\mathbf{D}-\mathbf{V}\mathbf{A}\mathbf{U})^{-1} \end{array} \right]. \quad (85)$$

After permutation of (66) and with $\mathbf{A} = 0.5\mathbf{R}_{i-1}$, $\mathbf{P}$ being a collection of blocks of $\mathbf{P}_{\mathrm{bc}}$, and $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{D}$ being collections of blocks of $(\mathbf{I}_{n\mathrm{b}}^{\circ} + \mathbf{D}_{i\mathrm{b}}^{\circ} + \mathbf{K}_{n\mathrm{b}}^{\circ})\mathbf{W}_{n\mathrm{b}}^{-1}$, one can use (85) to speed up the recursion.

It is worth pointing out that the recursion (66), except for the last step $i = n$, can be viewed as a fixed-point iteration. This is so since $\mathbf{D}_{i\mathrm{b}}^{\circ}$ is independent of i for $i < n$ assuming a corner in the shape of a wedge. One can, thus, iterate (66) with $\mathbf{D}_{i\mathrm{b}}^{\circ} = \mathbf{D}_{(n-1)\mathrm{b}}^{\circ}$ until convergence for $\mathbf{R}_{n-1}$, without knowing the precise number n in advance, and then perform one last iteration for $\mathbf{R}_n$ with $\mathbf{D}_{n\mathrm{b}}^{\circ}$.

Some other observations: The recursion (70) can be viewed as a fixed-point iteration, too, for $i < n$. It can be sped up by replacing $\mathbf{R}_{i-1}$ with the converged quantity $\mathbf{R}_{n-1}$ computed above. The blocks of $\mathbf{X}\mathbf{R}^{-1}$ can be stored on disk since they are only needed in the post-processor (49). Symmetries in the non-zero blocks of the $\mathbf{Q}_{\mathrm{cb}}$ and $\mathbf{P}_{\mathrm{bc}}$ matrices can be used

to reduce the number of arithmetic operations needed for their application, see eq. (45) of Helsing and Ojala (2008a). Use of (85), rather than a straightforward implementation of (66), allows for a reduction of the computational cost with about 70% per recursion step.

References

Atkinson, K.E., 1997. *The numerical Solution of Integral Equations of the Second Kind*, Cambridge University Press, Cambridge.

Carpinteri, A., Paggi, M., 2007. Analytical study of the singularities arising at multi-material interfaces in 2D linear elastic problems, Engng Fract. Mech., **74**(1/2), 59–74.

Dong, C.Y., Cheung, Y.K., Lo, S.H., 2002. Regularized domain integral formulation for inclusion problems of various shapes by equivalent inclusion method, Comp. Meth. Appl. Mech. Engng **191**(31), 3411–3421.

Eischen, J.W., Torquato, S., 1993. Determining elastic behavior of composites by the boundary-element method, J. Appl. Phys. **74**(1), 159–170.

Englund, J., 2007. A higher order scheme for two-dimensional quasi-static crack growth simulations, Comp. Meth. Appl. Mech. Engng **196**(21/24), 2527–2538.

Greenbaum, A., Greengard, L., Mayo, A., 1992. On the numerical solution of the biharmonic equation in the plane, Physica D **60**(1/4), 216–225.

Greengard, L., Helsing, J., 1998. On the numerical evaluation of elastostatic fields in locally isotropic two-dimensional composites, J. Mech. Phys. Solids **46**(8), 1441–1462.

Greengard, L., Rokhlin, V., 1987. A fast algorithm for particle simulations, J. Comput. Phys. **73**(2), pp. 325–348.

Helsing, J., Jonsson, A., 2002. Stress Calculations on Multiply Connected Domains, J. Comput. Phys. **176**(2), 456–482.

Helsing, J., Ojala, R., 2008a. On the evaluation of layer potentials close to their sources, J. Comput. Phys. **227**(5), 2899–2921.

Helsing, J., Ojala, R., 2008b. Corner singularities for elliptic problems: Integral equations, graded meshes, quadrature, and compressed inverse preconditioning, J. Comput. Phys., **227**(20), 8820–8840.

Helsing, J., Peters, G., 1999. Integral equation methods and numerical solutions of crack and inclusion problems in planar elastostatics, SIAM J. Appl. Math., **59**(3), 965–982.

Henderson, H.V., Searle, S.R., 1981. On deriving the inverse of a sum of matrices, SIAM Rev. **23**(1), 53–60.

Jin, J., Wu, X., 2004. The singular finite element method for some elliptic boundary value problem with interface, J. Comp. Appl. Math., **170**(1), 197–216.

Jeon, I., Kang, K.J., Im, S., 2008. Stress intensities at the triple junction of a multilevel thin film package, Microel. Reliab., **48**(5), 749–756.

Jonsson, A., 2003. Discrete dislocation dynamics by an O(N) algorithm, Comput. Mater. Sci., **27**(3), 271–288.

Koshelev, V., Ghassemi, A., 2008. Wedge boundary elements for 2D problems with corner points, Engng Anal. Bound. Elem., **32**(2), 168–175.

Kushch, V.I., Shmegera, S.V., Mishnaevsky, L., 2008. Meso cell model of fiber reinforced composite: Interface stress statistics and debonding paths, Int. J. Solids Struct., **45**(9), 2758–2784.

Linkov, A.M., Koshelev, V.F., 2006. Multi-wedge points and multi-wedge elements in computational mechanics: evaluation of exponent and angular distribution, Int. J. Solids Struct., **43**(18/19), 5909–5930.

Liu, Y.J., 2008. A fast multipole boundary element method for 2D multi-domain elastostatic problems based on a dual BIE formulation, Comput. Mech., **42**(5), 761–773.

Liu, Y.J., Nishimura, N., Otani, Y., Takahashi, T., Chen, X.L., Munakata, H., 2005. A fast boundary element method for the analysis of fiber-reinforced composites based on a rigid-inclusion model, ASME J. Appl. Mech., **72**(1), 115–128.

Mantič, V., París, F., Berger, J., 2003. Singularities in 2D anisotropic potential problems in multi-material corners: Real variable approach, Int. J. Solids Struct., **40**(20), 5197–5218.

Mikhlin, S.G., 1964. *Integral Equations and their applications to certain problems in mechanics, mathematical physics and technology, 2nd ed.*, Pergamon Press, London.

Mogilevskaya, S.G., Wang, J., Crouch, S.L., 2007. Numerical evaluation of the effective elastic moduli of rocks, Int. J. Rock. Mech. Min. Sci., **44**(3), 425–436.

Muskhelishvili, N.I., 1953. *Singular Integral Equations, 2nd Ed.*, P. Noordhoff N.V, Groningen.

Noda, N.A., Takase, Y., Hamashima, T., 2003. Generalized stress intensity factors in the interaction within a rectangular array of rectangular inclusions, Arch. Appl. Mech., **73**(5/6), 311–322.

Saad, Y., Schultz, M.H., 1986. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comp. **7**(3), 856–869.

Sethian, J.A., Wilkening, J., 2004. A numerical model of stress driven grain boundary diffusion, J. Comput. Phys., **193**(1), 275–305.

Sherman, D.I., 1959. On the problem of plane strain in non-homogeneous media. In *Non-Homogeneity in elasticity and plasticity*, ed. W. Olszag. (Proceedings of I.U.T.A.M. Symposium, Warsaw, 1958). Pergamon Press, London.

Theocaris, P.S., Ioakimidis, N.I., 1977. Inclusion problem in plane elasticity, Q. J. Mech. Appl. Math., **30**(4), 437–448.

Wang, J., Mogilevskaya, S.G., Crouch, S.L., 2005. An embedding method for modeling micromechanical behavior and macroscopic properties of composite materials, Int. J. Solids Struct., **42**(16/17), 4588–4612.

Wei, Y.J., Bower, A.F., Gao, H.J., 2008. Recoverable creep deformation and transient local stress concentration due to heterogeneous grain-boundary diffusion and sliding in polycrystalline solids, J. Mech. Phys. Solids, **56**(4), 1460–1483.