11 Computational elastostatics

Historically almost all of the insights into elasticity were obtained by means of analytic calculations, carried out by some of the best scientists of the time using the most advanced methods available to them, sometimes even inventing new mathematical concepts and methods along the way. Textbooks on the theory of elasticity are often hard to read because of their demands on the reader for command of mathematics.

In the last half of the twentieth century, the development of the digital computer changed the character of this field completely. Faced with a problem in elastostatics, modern engineers quickly turn to numerical computation. The demand for prompt solutions to design problems has over the years evolved these numerical methods into a fine art, and numerous commercial and public domain programs are now available.

In this short chapter we shall illustrate how it is possible to solve a concrete problem numerically, providing sufficient detail that a computer program can be implemented. It is not the intention here to expose the wealth of tricks of the trade, but just present the basic reasoning behind the numerical approach and the various steps that must be carried out. First, one must decide on the field equations and boundary conditions that are valid for the problem at hand. Second, the infinity of points in continuous space must be replaced by a finite set of points or volumes, often organized in a regular grid, and the fundamental equations must be approximated on this set. Third, a method must be adopted for an iterative approach towards the desired solution, with convergence criteria that enable one to monitor the progress of the computation and calculate error estimates.

11.1 Theory of the numeric method

As we do not, from the outset, know the solution to the problem we wish to solve by numerical means, we must begin by making an educated guess about the initial displacement field. Unless we are incredibly lucky this guess will fail to satisfy the mechanical equilibrium equations and the boundary conditions, resulting in a non-vanishing effective body force, as well as a violation of Newton's third law on the boundary. The idea is now to create an iterative procedure which through a sequence of tiny changes in displacement will proceed from the chosen initial state towards the equilibrium state.

Basic equations

Given an arbitrary (infinitesimal) displacement field $\boldsymbol{u}(\boldsymbol{x})$, the fundamental equations underlying computational elastostatics are,

$$u_{ij} = \frac{1}{2}\big(\nabla_i u_j + \nabla_j u_i\big), \qquad \text{Cauchy's strain tensor (7.20)} \qquad (11.1a)$$

$$\sigma_{ij} = \sum_{kl} E_{ijkl} u_{kl}, \qquad \text{general Hooke's law (8.18)} \qquad (11.1b)$$

$$f_i^* = f_i + \sum_j \nabla_j \sigma_{ij}, \qquad \text{effective body force (6.15)} . \qquad (11.1c)$$

In numeric computation there is no reason to limit the formalism to isotropic materials, so we have used the most general form of Hooke's law. If somebody presents you with a displacement field for a body, these equations represent a "machine" or "program" that through purely local operations (i.e. differentiation) allows you to calculate the effective body force acting in each and every point of the body. In mechanical equilibrium the effective body force $\boldsymbol{f}^*$ must of course vanish everywhere in the volume V of the body, but since we do not know the equilibrium displacement field — otherwise the numerical computation would be unnecessary — $\boldsymbol{f}^*$ will in general not vanish while we are seeking the solution.

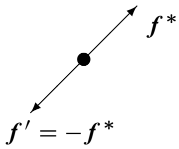
The solution must also satisfy boundary conditions on the surface S of the body. We shall assume that part of the surface, S_0, is permanently "glued" to undeformable and unmoveable external bodies such that the displacement has to vanish here. In the remainder of the body surface, $S_P = S - S_0$, the surface stress is under "user control" and fixed to be $\boldsymbol{P}(\boldsymbol{x})$. The solution must in other words satisfy,

$$\boldsymbol{f}^*(\boldsymbol{x}) = \boldsymbol{0} \qquad \text{for } \boldsymbol{x} \in V, \qquad (11.2a)$$

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{0} \qquad \text{for } \boldsymbol{x} \in S_U, \qquad (11.2b)$$

$$\boldsymbol{\sigma}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) = \boldsymbol{P}(\boldsymbol{x}) \qquad \text{for } \boldsymbol{x} \in S_P. \qquad (11.2c)$$

where $\boldsymbol{n}(\boldsymbol{x})$ is the normal to the surface in $\boldsymbol{x}$. Where the external surface stresses vanish, $\boldsymbol{P}(\boldsymbol{x}) = \boldsymbol{0}$, the body surface is said to be *free*.

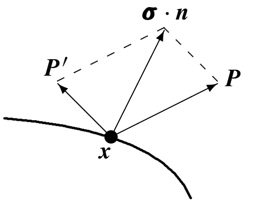


Every material particle can be kept in place by means of an additional virtual force, $\boldsymbol{f}' = -\boldsymbol{f}^*$, that balances the effective internal body force on the particle.

Virtual work and potential energy

Starting with an educated guess for the displacement field, $\boldsymbol{u}(\boldsymbol{x})$, that vanishes on S_0, there is no guarantee that the effective force $\boldsymbol{f}^*$ vanishes nor that the boundary condition $\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{P}$ is fulfilled. To prevent the material particles from beginning to move we must as in section 7.4 add a *virtual body force* $\boldsymbol{f}' = -\boldsymbol{f}^*$ to the material particles and a *virtual surface stress* $\boldsymbol{P}' = \boldsymbol{\sigma} \cdot \boldsymbol{n} - \boldsymbol{P}$ to secure that Newton's third law is fulfilled in each point in S_P. If we now change the displacement field by a tiny field, $\delta\boldsymbol{u}(\boldsymbol{x})$, which also vanishes at S_0, the work of the virtual forces becomes,

$$\boxed{\delta W = -\int_V \boldsymbol{f}^* \cdot \delta\boldsymbol{u}\, dV + \int_{S_P} \delta\boldsymbol{u} \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{n} - \boldsymbol{P})\, dS.} \qquad (11.3)$$



At every point of the surface S_P we must add a virtual stress $\boldsymbol{P}' = \boldsymbol{\sigma} \cdot \boldsymbol{n} - \boldsymbol{P}$ to secure continuity of the stress vector, demanded by Newton's third law.

Carrying out a partial integration as in section 7.4, but using that $\delta\boldsymbol{u} = \boldsymbol{0}$ only on S_0, leads to,

$$\delta W = -\int_V \boldsymbol{f} \cdot \delta\boldsymbol{u}\, dV + \int_V \sum_{ij} \sigma_{ij} \delta u_{ij}\, dV - \int_{S_P} \delta\boldsymbol{u} \cdot \boldsymbol{P}\, dS. \qquad (11.4)$$

The first term is the virtual work against the external body forces, the second the virtual work against the internal stresses, and the last the virtual work against the external surface forces.

Since neither $\boldsymbol{f}$ nor $\boldsymbol{P}$ depend on $\boldsymbol{u}$, the infinitesimal virtual work may immediately be integrated to yield the (potential) energy of any displacement (see for example [Doghri 2000, p. 35] for a more rigorous treatment),

$$\boxed{\mathcal{E} = -\int_V \boldsymbol{f} \cdot \boldsymbol{u}\, dV + \frac{1}{2}\int_V \sum_{ij} \sigma_{ij} u_{ij}\, dV - \int_{S_P} \boldsymbol{u} \cdot \boldsymbol{P}\, dS.} \tag{11.5}$$

Here the first term is the energy of the external body forces (gravity), the second the elastic energy of the deformation, and the last the energy of the external surface forces. Notice that the potential energy of the displacement field always contains the last term, even if the boundary condition (11.2c) is fulfilled. In section 7.4 we did not get this term because we assumed that the displacement was fixed, $\delta\boldsymbol{u} = 0$, on all of the body surface S.

Relaxing towards equilibrium

The potential energy (11.5) is a positive definite quadratic form in the displacement field, and has consequently a unique minimum. At this minimum the energy must be stationary $\delta\mathcal{E} = \delta W = 0$ under any variation $\delta\boldsymbol{u}$ that vanishes on S_0, and it follows immediately from (11.3) that $\boldsymbol{f}^* = \boldsymbol{0}$ in V and $\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{P}$ on S_P. *The unique minimum of the energy is therefore the desired equilibrium solution.* This result forms the basis for obtaining the numerical solution.

Starting with an arbitrary displacement field satisfying $\boldsymbol{u} = \boldsymbol{0}$ on S_0, the iterative *relaxation procedure* consists in designing a sequence of small displacement steps that all drain energy away from the body. In each iteration the total energy of the body will decrease until it reaches the unique minimum. Thus, in the end the relaxation procedure will arrive at the desired equilibrium state.

Gradient descent

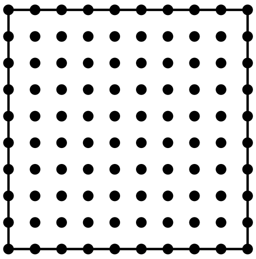
A common relaxation procedure is to choose the displacement change in a step to be,

$$\delta\boldsymbol{u} = \epsilon \boldsymbol{f}^* \quad \text{in } V, \qquad\qquad \delta\boldsymbol{u} = -\epsilon(\boldsymbol{\sigma} \cdot \boldsymbol{n} - \boldsymbol{P}) \quad \text{on } S_P \tag{11.6}$$

where ϵ is a positive quantity, called the *step-size*. Relaxing the displacement in this way guarantees that the integrands in both terms of eq. (11.3) are manifestly negative everywhere in the volume V and on the surface S_P, and thus drains energy away from every material particle in the body that is not already in equilibrium. Since the displacement incessantly "walks downhill" *against* the gradient of the total energy (in the space of all allowed displacement fields), it is naturally called *gradient descent*.

Gradient descent is not a foolproof method, even when the energy (as in linear elastic media) is a quadratic function of the displacement field with a unique minimum. In particular the step-size ϵ must be chosen judiciously. Too small, and the procedure may never seem to converge; too large, and it may overshoot the minimum and go into oscillations or even diverge. Many fine tricks have been invented to get around these problems and speed up convergence [Press et al. 1992, Braess 2001], for example increasing the step-size when the field changes are too small, and decreasing it when they are too large. Another, rather effective method, is *conjugate gradient descent* in which the optimal step-size is calculated in advance by searching for a minimum of the total energy along the chosen direction of descent. Having found this minimum, the procedure is repeated in the new direction of steepest descent which necessarily must be orthogonal to the old one. However, in the remainder of this chapter we shall just use the straightforward technique of the dedicated downhill skier, always selecting the steepest gradient for the next "step".

11.2 Discretization of space

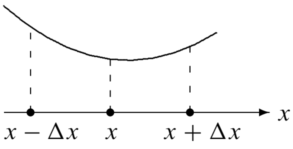


A two-dimensional (10 × 10) square grid. There are 36 points at the boundary and 64 inside. Small grids have a lot of boundary.

The infinity of points in space cannot be represented in a finite computer. In numerical simulations of the partial differential equations of continuum physics, smooth space is often replaced by a finite collection of points, a grid or lattice, on which the various fields "live" (see for example [Anderson 1995, Griebel et al. 1998]). In Cartesian coordinates the most convenient grid for a rectangular volume $a\times b\times c$ is a rectangular lattice with $(N_x+1)\times(N_y+1)\times(N_z+1)$ points that are equally spaced at coordinate intervals $\Delta x = a/N_x$, $\Delta y = b/N_y$, and $\Delta z = c/N_z$. The grid coordinates are numbered by $n_x = 0, 1, \ldots, N_x$, $n_y = 0, 1, \ldots, N_y$ and $n_z = 0, 1, \ldots, N_z$, and the various fields can only exist at the positions $(x, y, z) = (n_x\Delta x, n_y\Delta y, n_z\Delta z)$.

There are many other ways of discretizing space besides using rectangular lattices, for example triangular, hexagonal or even random lattices. The choice of grid depends on the problem itself, as well as on the field equations and the boundary conditions. The coordinates in which the system is most conveniently described may not be Cartesian but curvilinear, and that leads to quite a different discretization. The surface of the body may or may not fit well with the chosen grid, but that problem may be alleviated by making the grid very dense at the cost of computer time and memory. When boundaries are irregular, as they usually are for real bodies, an adaptive grid that can fit itself to the shape of the body may be the best choice. Such a grid may also adapt to put more points where they are needed in regions of rapid variation of the displacement field.

Finite difference operators with first-order errors



Forward and backward finite differences can be very different, and may as here even have opposite signs.

In a discrete space, coordinate derivatives of fields such as $\nabla_x f(x, y, z)$ must be approximated by finite differences between the field values at the allowed points. Using only the nearest neighbors on the grid there are two basic ways of forming such differences at a given *internal* point of the lattice, namely forwards and backwards,

$$\widehat{\nabla}^+_x f(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x}, \qquad \widehat{\nabla}^-_x f(x) = \frac{f(x) - f(x-\Delta x)}{\Delta x}. \tag{11.7}$$

Here and in the following we suppress for clarity the "sleeping" coordinates y and z and furthermore assume that finite differences in these coordinates are defined analogously.

According to the rules of differential calculus, both of these expressions will in the limit of $\Delta x \to 0$ converge towards $\nabla_x f(x)$. Inserting the Taylor expansion

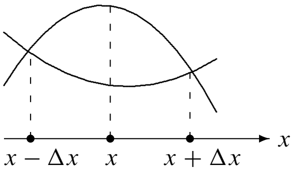
$$f(x+\Delta x) = f(x) + \Delta x \nabla_x f(x) + \tfrac{1}{2}\Delta x^2 \nabla_x^2 f(x) + \tfrac{1}{6}\Delta x^3 \nabla_x^3 f(x) + \cdots,$$

we find indeed

$$\widehat{\nabla}^\pm_x f(x) = \nabla_x f(x) \pm \tfrac{1}{2}\Delta x \nabla_x^2 f(x) + \cdots,$$

with an error (the second term) that is of first order in the interval Δx.

Finite difference operators with second-order errors



The central difference is insensitive to the value at the center. The two curves shown here have the same symmetric difference but behave quite differently.

It is clear from the above expression that the first-order error may be suppressed by forming the average of forward and backward difference operators, called the *central difference*,

$$\widehat{\nabla}_x f(x) = \frac{1}{2}\left(\widehat{\nabla}^+_x + \widehat{\nabla}^-_x\right) f(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x}. \tag{11.8}$$

Expanding the function values to third order we obtain

$$\widehat{\nabla}_x f(x) = \nabla_x f(x) + \tfrac{1}{6}\Delta x^2 \nabla_x^3 f(x) + \cdots,$$

with errors of second order only. The central difference does not involve the field value at the central point x, so one should be wary of possible "leapfrog" or "flipflop" numeric instabilities in which half the points of the lattice behave differently than the other half.

On a boundary, the central difference cannot be calculated, and we are forced to use one-sided differences. On the left boundary one must use the forward difference and on the right the backward one. In order to consistently avoid $\mathcal{O}(\Delta x)$ errors one may instead of one-step differences (11.7) use one-sided two-step difference operators (see problem 11.1),

$$\widehat{\nabla}_x^+ f(x) = \frac{-f(x+2\Delta x) + 4f(x+\Delta x) - 3f(x)}{2\Delta x}, \tag{11.9a}$$

$$\widehat{\nabla}_x^- f(x) = \frac{f(x-2\Delta x) - 4f(x-\Delta x) + 3f(x)}{2\Delta x}. \tag{11.9b}$$

The coefficients are chosen here such that the leading order corrections vanish. Expanding to third order we find indeed

$$\widehat{\nabla}_x^\pm f(x) = \nabla_x f(x) \mp \frac{1}{3}\Delta x^2 \nabla_x^3 f(x) + \cdots,$$

which shows that the one-sided differences represent the derivative at the point x with leading errors of $\mathcal{O}\left(\Delta x^2\right)$ only.

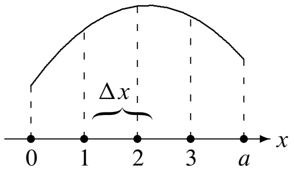
Other schemes involving more distant neighbors to suppress even higher order errors are of course also possible, but here we shall only use the second order expressions.

Numeric integration

In simulations it will also be necessary to calculate various line, surface and volume integrals over discretized space. Since the fields are only known at the points of the discrete lattice, the integrals must be replaced by suitably weighted sums over the lattice points.

Consider, for example, a one-dimensional integral over an interval $0 \leq x \leq a$ along the x-axis. Divide the interval into a set of N sub-integrals over the lattice spacing $\Delta x = a/N$, we may write the integral as an exact sum of sub-integrals,

$$\int_0^a f(x)\,dx = \sum_{n=0}^{N-1} \int_{x_n}^{x_{n+1}} f(x)\,dx, \tag{11.10}$$



The interval $0 \leq x \leq a$ has four subintervals of size Δx numbered $n = 0, 1, 2, 3$.

where $x_n = n\Delta x$. By means of the Taylor expansion at x, we find for a single subinterval

$$\int_x^{x+\Delta x} f(x')dx' = \Delta x f(x) + \frac{1}{2}\Delta x^2 \nabla_x f(x) + \frac{1}{6}\Delta x^3 \nabla_x^2 f(x) + \cdots. \tag{11.11}$$

In the lowest order of approximation we replace the first order derivative by the forward difference operator (11.7a), and get

$$\begin{aligned}\int_{x_n}^{x_{n+1}} f(x')dx') &= \Delta x f(x) + \frac{1}{2}\Delta x^2 \widehat{\nabla}_x f(x) + \mathcal{O}\left(\Delta x^3\right) \\ &= \frac{1}{2}\Delta x\big(f(x_{n+1}) + f(x_n)\big) + \mathcal{O}\left(\Delta x^3\right).\end{aligned}$$

Finally, adding the N contributions, we arrive at

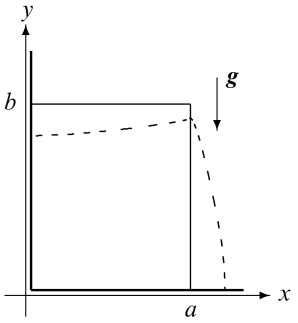
$$\int_0^a f(x)\,dx = \frac{1}{2}\big(f(0) + f(a)\big)\,\Delta x + \sum_{n=1}^{N-1} f(x_n)\,\Delta x + \mathcal{O}\left(\Delta x^2\right). \tag{11.12}$$

This is the well-known trapezoidal rule [Press et al. 1992, p. 131].

In higher dimensions one may integrate each dimension according to this formula. Again there exist schemes for numerical integration on a regular grid with more complicated weights and correspondingly smaller errors, for example Simpson's famous formula [Press et al. 1992, p. 134] which is correct to $\mathcal{O}\left(\Delta x^4\right)$.

11.3 Gravitational settling in two dimensions

One of the simplest non-trivial problems that does not seem to admit an exact analytic solution is the gravitational settling of an infinitely long, horizontal block of elastic material with a rectangular cross section of dimensions $a \times b$. When one of the vertical sides is removed the material bulges out (see the margin figure). In this somewhat academic case we follow the conventions normally used in two dimensions and take the y-axis to be vertical. The wall that is removed is situated at $x = a$ whereas the wall at $x = 0$ remains in place. It is reasonable to assume that there can be no displacement in the z-direction direction, i.e. $u_z = 0$ everywhere, because it would have to move infinitely much material. It is also reasonable to assume that the displacements u_x and u_y only depend on x and y, but not on z. The problem has become effectively two-dimensional, although there are vestiges of the three-dimensional problem, for example the non-vanishing stress along the z-direction, but that can be ignored.



Expected shape of two-dimensional gravitational settling. When the wall at $x = a$ is removed, the elastic material will bulge out, because of its own weight.

Equations

The components of the two-dimensional strain tensor are

$$u_{xx} = \nabla_x u_x, \tag{11.13a}$$

$$u_{yy} = \nabla_y u_y, \tag{11.13b}$$

$$u_{xy} = \tfrac{1}{2}(\nabla_x u_y + \nabla_y u_x). \tag{11.13c}$$

The corresponding stresses are found from Hooke's law (8.9) ,

$$\sigma_{xx} = 2\mu u_{xx} + \lambda(u_{xx} + u_{yy}), \tag{11.14a}$$

$$\sigma_{yy} = 2\mu u_{yy} + \lambda(u_{xx} + u_{yy}), \tag{11.14b}$$

$$\sigma_{xy} = \sigma_{yx} = 2\mu\, u_{xy}. \tag{11.14c}$$

Finally, the components of the effective force are

$$f_x^* = \nabla_x \sigma_{xx} + \nabla_y \sigma_{xy}, \tag{11.15a}$$

$$f_y^* = \nabla_x \sigma_{xy} + \nabla_y \sigma_{yy} - \rho_0 g_0. \tag{11.15b}$$

Note that only first-order partial derivatives are used in these equations. This makes it trivial to convert the equations to the discrete lattice by replacing the derivatives by finite difference operators.

Boundary conditions

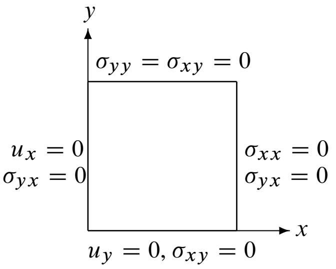
The boundary consists of the two fixed surfaces at $x = 0$ and $y = 0$ and the free surfaces at $x = a$ and $y = b$. We shall adopt the following boundary conditions,

$$\sigma_{xx} = 0, \quad \sigma_{yx} = 0 \qquad \text{free surface at } x = a, \tag{11.16a}$$

$$\sigma_{yy} = 0, \quad \sigma_{xy} = 0 \qquad \text{free surface at } y = b, \tag{11.16b}$$

$$u_x = 0, \quad \sigma_{yx} = 0 \qquad \text{fixed wall at } x = 0, \tag{11.16c}$$

$$u_y = 0, \quad \sigma_{xy} = 0 \qquad \text{fixed wall at } y = 0. \tag{11.16d}$$

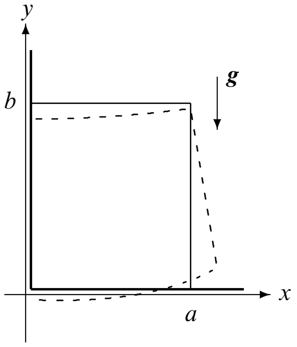


Boundary conditions for the rectangular block.

Here we have assumed that the fixed surfaces are slippery, so that the shear stress must vanish. That is however not the only choice.

Had we instead chosen the fixed walls to be sticky so that the elastic material were unable to slip along the sides, the tangential displacements at these boundaries would also have to vanish, i.e. $u_y = 0$ at $x = 0$ and $u_x = 0$ at $y = 0$. The tangential stress $\sigma_{xy} = \sigma_{yx}$ would, on the other hand, be left free to take any value determined by the field equations.

Shear-free solution



The shear-free solution sinks partly into the bottom of the box. An extra vertical pressure is needed from below in order to fulfill the boundary conditions.

Since the shear stress vanishes at all boundaries, it is tempting to solve the equations by requiring the shear stress also to vanish throughout the block, $\sigma_{xy} = \sigma_{yx} = 0$, as we did for the three-dimensional settling in section 9.2. One may verify that the following field solves the field equations,

$$u_x = \frac{\nu}{1-\nu}\frac{(b-y)x}{D}, \qquad u_y = -\frac{b^2-(b-y)^2}{2D} + \frac{\nu}{1-\nu}\frac{3x^2-a^2}{6D}, \tag{11.17}$$

where the characteristic deformation scale is

$$D = \frac{4\mu(\lambda+\mu)}{(2\mu+\lambda)\rho_0 g_0} = \frac{E}{(1-\nu^2)\rho_0 g_0}. \tag{11.18}$$

The solution is of the same general form as in the three-dimensional case (9.15) on page 145, but the dependence on Poisson's ratio ν is different because of the different geometry. As before, this solution also fails to meet the boundary conditions at the bottom, here $y = 0$. The arbitrary constant in u_y has been chosen such that $\int_0^a u_y \, dx = 0$ at $x = 0$.

Convergence measures

The approach towards equilibrium may, for example, be monitored by means of the integral over the square of the effective force field which should converge towards zero, if the algorithm works. We shall choose the monitoring parameter to be

$$\chi = \frac{1}{\rho_0 g_0}\sqrt{\frac{1}{ab}\int_0^a dx \int_0^b dy \left(f_x^{*2} + f_y^{*2}\right)}. \tag{11.19}$$

It is normalized such that $\chi = 1$ in the undeformed state where $u_x = u_y = 0$ and thus $f_x^* = 0$ and $f_y^* = -\rho_0 g_0$. The integrals are calculated using the trapezoidal rule. The iterative process can then be stopped when the value of χ falls below any desired accuracy, say $\chi \lesssim 0.01$.

Another possibility is to calculate the potential energy (11.5) with $\boldsymbol{P} = \boldsymbol{0}$,

$$\mathcal{E} = \int_0^a dx \int_0^b dy \left[\frac{1}{2}(u_{xx}\sigma_{xx} + u_{yy}\sigma_{yy} + 2u_{xy}\sigma_{xy}) + \rho_0 g_0 u_y\right]. \tag{11.20}$$

This quantity should decrease monotonically from $\mathcal{E} = 0$ in the undeformed state towards its (negative) minimum. Since it — like χ — is well-defined in the continuum, its value should be relatively independent of how fine-grained the discretization is, as long as the lattice is large enough. It is, however, harder to determine the relative accuracy attained.

Iteration cycle

Assuming that the discretized displacement field on the lattice (u_x, u_y) satisfies the boundary conditions, we may calculate the strains (u_{xx}, u_{yy}, u_{zz}) from (11.13) by means of the discrete derivatives, and the stresses $(\sigma_{xx}, \sigma_{yy}, \sigma_{xy})$ from Hooke's law (11.14) . Stress boundary conditions are then imposed and the effective force field (f_x^*, f_y^*) is calculated from (11.15) . At this point the monitoring parameter χ may be checked and if below the desired accuracy, the iteration process is terminated. If not, the corrections

$$\delta u_x = \epsilon f_x^*, \qquad \delta u_y = \epsilon f_y^* \tag{11.21}$$

are added into the displacement field, boundary conditions are imposed on the displacement field, and the cycle repeats.

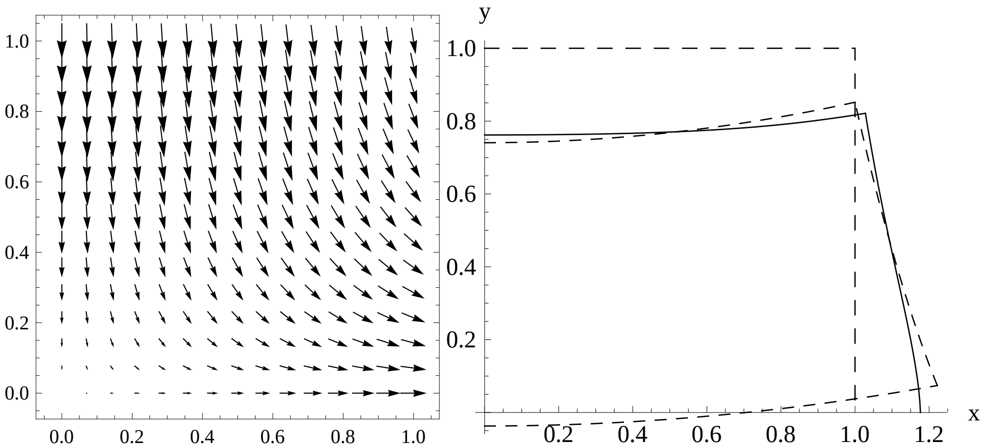


Figure 11.1. Computed deformation of a square two-dimensional block. On the left the equilibrium displacement field is plotted by means of little arrows (not to scale). On the right is plotted the outline of the deformed block. The displacement vanishes as it must at the fixed walls. The protruding material (solid line) has a slightly convex shape rather than the concave shape in the shear-free approximation (dashed lines).

The iteration process may be viewed as a dynamical process which in the course of (computer) time makes the displacement field converge towards its equilibrium configuration. The true dynamics of deformation (see chapter 25) go on in real time and are quite different. Since dissipation in solids is not included here, the true dynamics are unable to eat away energy and make the system relax towards equilibrium. Releasing the block from the undeformed state, as we do here, would instead create vibrations and sound waves that would reverberate forever throughout the system.

Choice of parameters

Since we are mostly interested in the shape of the deformation, we may choose convenient values for the input parameters. They are the box sides $a = b = 1$, the lattice sizes $N_x = N_y = 20$, Young's modulus $E = 2$, Poisson's ratio $\nu = 1/3$ and the force of gravity $\rho_0 g_0 = 1$. The step-size is chosen of the form

$$\epsilon = \frac{\omega}{E} \frac{\Delta x^2 \Delta y^2}{\Delta x^2 + \Delta y^2} \qquad (11.22)$$

where ω is called the *convergence parameter*. The reason for this choice is that the effective force is proportional to Young's modulus E and (due to the second-order spatial derivatives) to the inverse squares of the grid spacings, say $1/\Delta x^2 + 1/\Delta y^2 = (\Delta x^2 + \Delta y^2)/\Delta x^2 \Delta y^2$. The convergence parameter ω is consequently dimensionless and may be chosen to be of order unity to get fastest convergence. In the present computer simulation, the largest value that could be used before numeric instabilities set in was $\omega = 1$.

Programming hints

The fields are represented by real arrays, containing the field values at the grid points, for example

$$UX[i, j] \Leftrightarrow u_x(i\Delta x, j\Delta y), \qquad UY[i, j] \Leftrightarrow u_y(i\Delta x, j\Delta y), \qquad (11.23)$$

and similarly for the strain and stress fields. Allocating separate arrays for strains and stresses may seem excessive and can be avoided, but when lattices are as small as here, it does not matter. Anyway, the days of limited memory are over.

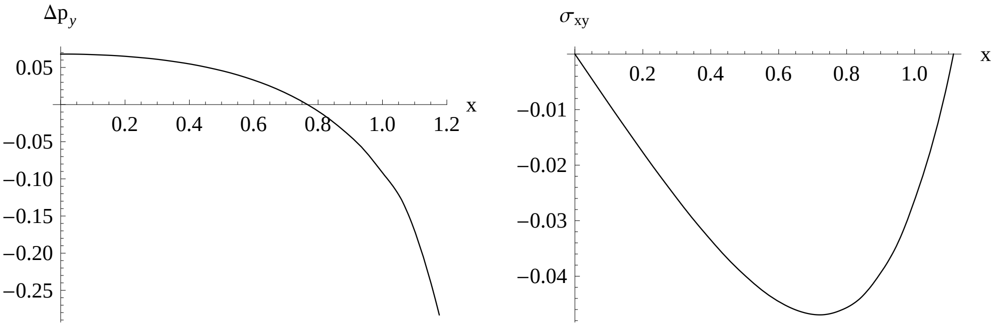


Figure 11.2. The computed vertical pressure excess, $\Delta p_y = p_y - p_0$ with $p_y = -\sigma_{yy}$ and $p_0 = \rho_0 g_0 b$, is plotted on the left for $y = 0$ as a function of the displaced x. On the right, the corresponding shear stress is plotted at $y = 0.5$. The pressure is higher in the central region than the shear-free estimate p_0, and the shear stress is negative (but small) and thus adds to the force exerted by gravity on the central part. The curves have been linearly interpolated between the data points. The small kink in Δp_y just beyond $x = 1$ is an artifact of the coarse lattice.

The iteration cycle is implemented as a loop, containing a sequence of calls to subroutines that evaluate strains, stresses, effective forces and impose boundary conditions, followed by a step that evaluates the monitoring parameters and finally updates the displacement arrays before the cycle repeats. The iteration loop is terminated when the accuracy has reached the desired level, or the number of iterations has exceeded a chosen maximum.

Results

After about 2000 iteration cycles the monitoring parameter χ has fallen from 1 to about 0.01 where it seems to remain without further change. This is most probably due to the brute enforcing of boundary values. The limiting value of χ diminishes with increasing lattice volume $N = N_x N_y$, in accordance with the lessened importance of the boundary which decreases like $1/\sqrt{N}$ relative to the volume.

The final displacement field and its influence on the outline of the original box is shown in figure 11.1. One notes how the displacement does not penetrate into the fixed bottom wall as it did in the shear-free approximation. In figure 11.2 the vertical pressure excess $\Delta p_y = p_y - p_0$ where $p_y = -\sigma_{yy}$ and $p_0 = \rho_0 g_0 b$ is plotted as a function of x at the bottom of the block ($y = 0$). Earlier we argued that there would have to be an extra normal reaction from the bottom in order to push up the sagging solution to the shear-free equations. This is also borne out by the plot of Δp_y which has roughly the same shape throughout the block. Since the vertical pressure in the central region is now larger than the weight of the column of material above (p_0), we expect that there must be a negative shear stress on the sides of the column to balance the extra vertical pressure, as is also evident from figure 11.2.

Problems

11.1 Show that the coefficients in the one-sided two-step differences (11.9) are uniquely determined.