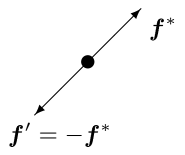
13 Numeric elastostatics

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 [13, 12, 27, 42].

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

In this chapter we shall illustrate how it is possible to solve a concrete problem numerically with so much 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 numerical simulation and the various steps that must be carried out in order to make a successful numerical simulation. First of all one must decide on the field equations and boundary conditions that should be implemented, and what simplifications that can be made on these from the outset. Secondly, the infinity of points in continuous space must be replaced by a finite set, often a regular grid or lattice, and the fundamental equations must be approximated on this set. Thirdly, a method must be adopted for the iterative approach towards the desired solution, and finally one needs to choose convergence criteria that enables one to monitor the progress of the computation and error estimates that give confidence in the solution.

13.1 Relaxing towards equilibrium



Every material particle can be kept in place by acting on it with an additional external force that balances the already existing effective body force on the particle.

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. This guess should satisfy the boundary conditions, but unless we are incredibly lucky it will fail to satisfy the equation of mechanical equilibrium, and thus there will be a non-vanishing effective body force $f_i^* = f_i + \sum_j \nabla_j \sigma_{ij}$. The idea is now to create an iterative procedure by which we may proceed from an arbitrary initial state towards equilibrium, $\boldsymbol{f}^* = \boldsymbol{0}$.

Any mechanical system that is not in equilibrium, is (literally) forced to move. If we nevertheless wish to keep it in place, we must act on it with suitable external forces, like we do when we stretch a rubber band. So, if the effective body force field $\boldsymbol{f}^*$ does not vanish, we must at least in our minds impose another external force field $\boldsymbol{f}' = -\boldsymbol{f}^*$ to balance the system's "own" field. In this way we can keep any displacement field fixed for as long as we wish.

Suppose now that we change the displacement of the body by a tiny amount, $\boldsymbol{u}(\boldsymbol{x}) \to \boldsymbol{u}(\boldsymbol{x}) + \delta\boldsymbol{u}(\boldsymbol{x})$. Then the added external forces provided by us must perform the work

$$\delta W = \int_V \boldsymbol{f}' \cdot \delta\boldsymbol{u}\, dV = -\int_V \boldsymbol{f}^* \cdot \delta\boldsymbol{u}\, dV \; . \tag{13-1}$$

The effective force will itself undergo change due to the change in the displacement, but that is of second order in $\delta\boldsymbol{u}$ and can be disregarded here.

If the above work is negative, the amount of elastic energy stored in the body must decrease, as when we relax a stretched rubber band a bit. As long as $\boldsymbol{f}^* \neq \boldsymbol{0}$ somewhere, we must be able to drain elastic energy away from the body by relaxing the displacement in the neighborhood. This process must, however, come to an end sooner or later, because physical bodies are not allowed to have infinitely negative energy. The conclusion is that an iterative procedure, in which successive changes in the displacement are all chosen to do negative work, will everything else being equal lead to a state with $\boldsymbol{f}^* = \boldsymbol{0}$ everywhere, *i.e.* to a solution for the equilibrium displacement field. Incidentally, this argument also shows that the equilibrium state must correspond to a minimum in the elastic energy (see section 10.4).

There remains a question about what happens at the surface of the body, where external stresses may also perform work under a relaxation of the displacement (see section 10.4). This work is the integrated product of the change in displacement and the stress vector acting on the surface, $\oint_S \delta\boldsymbol{u} \cdot \boldsymbol{\sigma} \cdot d\boldsymbol{S}$. For the most common boundary conditions, in which the displacement is either fixed on the surface or the stress vector is required to vanish, the integral yields zero, and the problem goes away.

Gradient descent

A common procedure is to select the change in displacement to be everywhere proportional to the effective force,

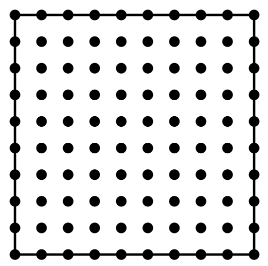
$$\delta \boldsymbol{u} = \epsilon \boldsymbol{f}^* \ , \tag{13-2}$$

where ϵ is a positive quantity, called the *step-size*. Relaxing the displacement in this way by "running along" with the force guarantees that the density of external work, $\boldsymbol{f}' \cdot \delta\boldsymbol{u} = -\boldsymbol{f}^* \cdot \delta\boldsymbol{u} = -\epsilon(\boldsymbol{f}^*)^2$, is negative everywhere and thus drains energy away from every material particle in the body that is not already in equilibrium. Since the displacement 'walks downhill' against the gradient of the total energy (in the space of all displacement fields), it is unsurprisingly called *gradient descent*.

Gradient descent is not a foolproof method, even when the energy (as here) 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, and too large, 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 [35], for example conjugate gradient descent in which the optimal step-size is calculated in advance by searching for a minimum along the chosen direction of descent. Here we shall, however, just use the straightforward technique of the dedicated downhill skier, always looking for the steepest gradient.

13.2 Discretization of space

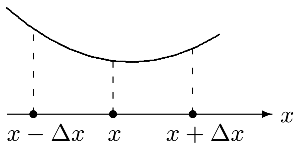
The infinity of points in space cannot be represented in a finite computer. In numerical simulations of the partial differential equations of continuum physics, smooth space is for this reason always replaced by a finite collection of points, a grid or lattice, on which the various fields "live". In 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 $(x, y, z) = (n_x\Delta x, n_y\Delta y, n_z\Delta z)$.


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

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 a cost in 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

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



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

$$\widehat{\nabla}_x^+ f(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} \; , \tag{13-3a}$$

$$\widehat{\nabla}_x^- f(x) = \frac{f(x) - f(x-\Delta x)}{\Delta x} \; . \tag{13-3b}$$

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

$$\begin{aligned} f(x+\Delta x) &= f(x) + \Delta x \nabla_x f(x) + \frac{1}{2}\Delta x^2 \nabla_x^2 f(x) \\ &\quad + \frac{1}{6}\Delta x^3 \nabla_x^3 f(x) + \frac{1}{24}\Delta x^4 \nabla_x^4 f(x) + \cdots \; , \end{aligned}$$

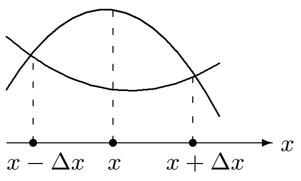
we find indeed

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

with an error that is of first order in the interval Δx.

Finite difference operators with second-order errors

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



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.

$$\widehat{\nabla}_x f(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} \; . \tag{13-4}$$

Expanding the function values to third order we obtain

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

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

On a boundary, the central difference cannot be calculated, and one is forced to use one-sided differences. If the grid is rectangular with a rectangular border, one must use the right hand difference on the boundary to the left and the left hand difference on the boundary to the right. In order to consistently avoid $\mathcal{O}(\Delta x)$ errors one may instead of (13-3) use the one-sided two-step difference operators (see problem 13.3),

$$\widehat{\nabla}_x^+ f(x) = \frac{-f(x+2\Delta x) + 4f(x+\Delta x) - 3f(x)}{2\Delta x} \;, \tag{13-5a}$$

$$\widehat{\nabla}_x^- f(x) = \frac{f(x-2\Delta x) - 4f(x-\Delta x) + 3f(x)}{2\Delta x} \;. \tag{13-5b}$$

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

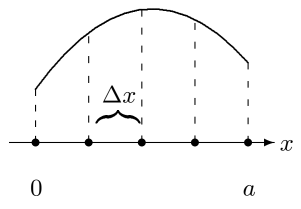
$$\widehat{\nabla}_x^\pm f(x) = \nabla_x f(x) \mp \frac{1}{3}\Delta x^2 \nabla_x^3 f(x) + \cdots \;,$$

which shows that both one-sided differences represent the derivative in 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.

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 on the points of the discrete lattice, the integrals must be replaced by suitably weighted sums over the lattice points.



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

Let us, for example, consider a one-dimensional integral over an interval, say $\int_0^a f(x)\,dx$, on a regular grid with coordinates $x_n = n\Delta x$ where $n = 0, 1, \ldots, N$. The contribution to the integral from n'th subinterval $x_n \le x \le x_{n+1}$ is approximated by the trapezoidal rule [35, p. 131]

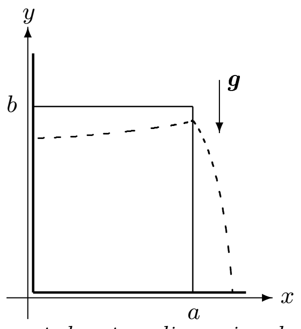
$$\int_{x_n}^{x_{n+1}} f(x)\,dx = \frac{1}{2}\big(f(x_n) + f(x_{n+1})\big)\Delta x + \mathcal{O}\left(\Delta x^3\right) \;,$$

which is analogous to the central difference in suppressing the leading order error. Adding the contributions from the N subintervals together, we obtain the well-known extended trapezoidal rule for numerical integration

$$\int_0^a f(x)\,dx \approx \frac{1}{2}f(0)\,\Delta x + \sum_{n=1}^{N-1} f(n\Delta x)\,\Delta x + \frac{1}{2}f(a)\,\Delta x + \mathcal{O}\left(\Delta x^2\right) \;. \tag{13-6}$$

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 Simpsons famous formula [35, p. 134] which is correct to $\mathcal{O}\left(\Delta x^4\right)$.



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

13.3 Gravitational settling in two dimensions

One of the simplest non-trivial problems that does not seem to admit of an exact analytic solution is the gravitational settling of a rectangular block of elastic material in a long open box of dimensions $a \times b \times c$ with one of the sides removed (see section 11.2 for the case where all sides are removed).

In this 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 the clamping in the z-direction at $z = 0$ and $z = c$ prevents any displacement in that direction, *i.e.* $u_z = 0$ everywhere. Since the block is assumed to be very long in this direction, 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 which is taken up by the walls at the ends.

Equations

The components of the two-dimensional strain tensor are

$$u_{xx} = \nabla_x u_x \ , \tag{13-7a}$$
$$u_{yy} = \nabla_y u_y \ , \tag{13-7b}$$
$$u_{xy} = \frac{1}{2}(\nabla_x u_y + \nabla_y u_x) \ . \tag{13-7c}$$

The corresponding stresses are found from Hooke's law (10-9) and (10-10),

$$\sigma_{xx} = (2\mu + \lambda)u_{xx} + \lambda u_{yy} \ , \tag{13-8a}$$
$$\sigma_{yy} = (2\mu + \lambda)u_{yy} + \lambda u_{xx} \ , \tag{13-8b}$$
$$\sigma_{xy} = \sigma_{yx} = 2\mu\, u_{xy} \ . \tag{13-8c}$$

Finally, the components of the effective force are

$$f_x^* = \nabla_x \sigma_{xx} + \nabla_y \sigma_{xy} \ , \tag{13-9a}$$
$$f_y^* = \nabla_x \sigma_{xy} + \nabla_y \sigma_{yy} - \rho_0 g_0 \ . \tag{13-9b}$$

Notice that only first order partial derivatives are used in these equations.

We could of course substitute the equations into each other to express the effective force in terms of second order derivatives of the displacement fields

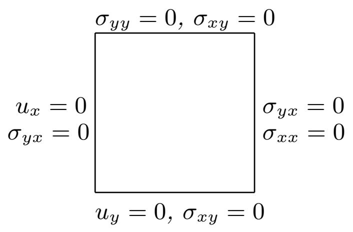
$$f_x^* = (\lambda + 2\mu)\nabla_x^2 u_x + \mu \nabla_y^2 u_x + (\lambda + \mu)\nabla_x \nabla_y u_y \ , \tag{13-10a}$$
$$f_y^* = (\lambda + 2\mu)\nabla_y^2 u_y + \mu \nabla_x^2 u_y + (\lambda + \mu)\nabla_x \nabla_y u_x - \rho_0 g_0 \ . \tag{13-10b}$$

Although there are excellent numerical methods to solve such (elliptic) differential equations, the boundary conditions that involve stresses (see below) are not so easy to implement.

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,

$$\sigma_{xx} = 0\ , \quad \sigma_{yx} = 0 \qquad \text{free surface at } x = a\ , \tag{13-11a}$$
$$\sigma_{yy} = 0\ , \quad \sigma_{xy} = 0 \qquad \text{free surface at } y = b\ , \tag{13-11b}$$
$$u_x = 0\ , \quad \sigma_{yx} = 0 \qquad \text{fixed wall at } x = 0\ , \tag{13-11c}$$
$$u_y = 0\ , \quad \sigma_{xy} = 0 \qquad \text{fixed wall at } y = 0\ . \tag{13-11d}$$


$\sigma_{yy} = 0,\ \sigma_{xy} = 0$
$u_x = 0$
$\sigma_{yx} = 0$
$\sigma_{yx} = 0$
$\sigma_{xx} = 0$
$u_y = 0,\ \sigma_{xy} = 0$

Boundary conditions for the rectangular block.

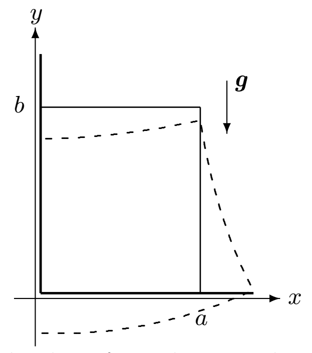
Here we have assumed that the fixed surfaces are slippery, so that the shear stress must vanish. That is however not the only choice. Whereas freedom appears to be unique, there is always more than one way to constrain it.

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 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

Since the shear stress vanishes at all boundaries, it is tempting to solve the equations by requiring the shear stress also to vanish everywhere inside the block, $\sigma_{xy} = \sigma_{yx} = 0$, as we did for the three-dimensional settling in section 11.2. One may verify that the following field solves the field equations

$$u_x = \frac{\nu}{(1-\nu)D}(b-y)x \tag{13-12}$$
$$u_y = -\frac{1}{2D}\left(b^2 - (b-y)^2 + \frac{\nu}{1-\nu}(a^2 - x^2)\right)\ , \tag{13-13}$$



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

where

$$D = \frac{4\mu(\lambda+\mu)}{(2\mu+\lambda)\rho_0 g_0} = \frac{E}{(1-\nu^2)\rho_0 g_0} \tag{13-14}$$

is the characteristic deformation scale. The solution is of the same general form as in the three-dimensional case (11-16), but the dependence on Poisson's ratio ν is different because of the two-dimensionality. As before, this solution also fails to meet the boundary conditions at the bottom, here $y = 0$.

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 must converge towards zero, if the algorithm works. We shall choose the monitoring parameter to be

$$\chi = \frac{1}{\rho_0 g_0}\sqrt{\frac{1}{ab}\int_0^a dx \int_0^b dy \left({f_x^*}^2 + {f_y^*}^2\right)}\ . \tag{13-15}$$

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 integral is calculated as a sum over the two-dimensional lattice (with appropriate weighting of the boundaries). The iterative process can then be stopped when the value falls below any desired accuracy, say $\chi \lesssim 0.01$.

Another possibility is to calculate the total energy (10-30) with the integral replaced by a double sum over the lattice points,

$$\mathcal{E} = \int_0^a dx \int_0^b dy \left[\frac{1}{2}(u_{xx}\sigma_{xx} + u_{yy}\sigma_{yy} + 2u_{xy}\sigma_{xy}) + \rho_0 g_0 u_y\right]\ . \tag{13-16}$$

This quantity should decrease monotonically towards its minimum and, since it like χ also has a well-defined continuum limit, 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 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 (13-7) by means of the discrete derivatives, and the stresses $(\sigma_{xx}, \sigma_{yy}, \sigma_{xy})$ from Hooke's law (13-8). Stress boundary conditions are then imposed and the effective force field (f_x^*, f_y^*) calculated from (13-9). 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

$$\delta u_x = \epsilon f_x^* \tag{13-17}$$
$$\delta u_y = \epsilon f_y^* \tag{13-18}$$

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

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 12) goes on in real time and is quite different. Since dissipation in solids is not included here, this dynamics is unable to eat away energy and make the system relax towards equilibrium. Releasing the block from the undeformed state, as we do here, will instead create vibrations and sound waves that reverberate forever throughout the system.

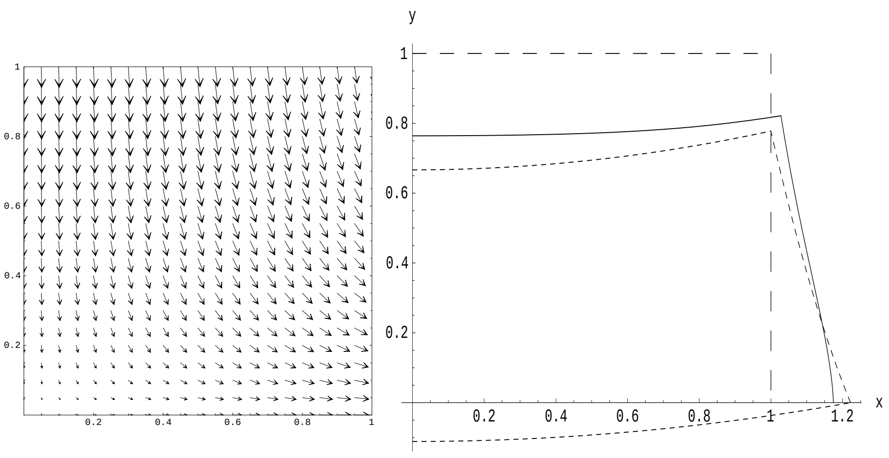


Figure 13.1: *Computed deformation of square two-dimensional block. On the left the equilibrium displacement field is plotted by means of little arrows (not to scale). On the right is plotted the outline of the deformed block. The displacement vanishes as it must at the fixed walls. The protruding material (solid line) has a slightly convex shape rather than the concave shape in the shear-free approximation (small dashes).*

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

$$\epsilon = \frac{\omega}{E} \frac{\Delta x^2 \Delta y^2}{\Delta x^2 + \Delta y^2} \ . \tag{13-19}$$

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 to the inverse squares of the grid spacings, say $1/\Delta x^2 + 1/\Delta y^2 = (\Delta x^2 + \Delta y^2)/\Delta x^2 \Delta y^2$ (because of the second order spatial derivatives). 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

$$UX[i,j] \Leftrightarrow u_x(i\Delta x, j\Delta y) \ , \tag{13-20}$$

$$UY[i,j] \Leftrightarrow u_y(i\Delta x, j\Delta y) \ , \tag{13-21}$$

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.

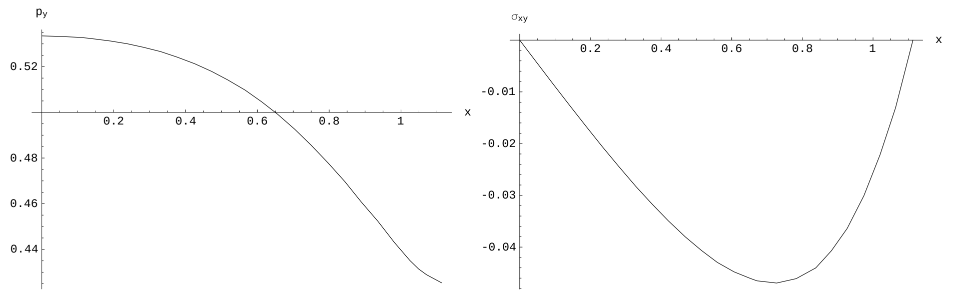


Figure 13.2: *The computed vertical pressure, $p_y = -\sigma_{yy}$, is plotted on the left for $y = 0.5$ as a function of the true x. On the right the corresponding shear stress is plotted at the same height. The pressure is higher in the central region than the shear-free estimate ($p_y = 0.5$) and the shear stress is negative (but small) and thus adds to the force exerted by gravity. The curves have been linearly interpolated between the data points. If the grid is made denser, there will be more detail in the region of the protrusion ($x \gtrsim 1$).*

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, taking only a few seconds on a modern PC, 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 brutal 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 fig. 13.1. One notices how the displacement does not penetrate into the fixed bottom wall as it did in the shear-free approximation. In fig. 13.2 the vertical pressure $p_y = -\sigma_{yy}$ is plotted as a function of x in the middle of the block ($y = 0.5$). 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 p_y which has roughly the same shape throughout the block. Since the vertical pressure is now larger than the weight of the column of material above, 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 fig. 13.2.

Problems

13.1 Show that the one-sided two-step difference (13-5) may be written

$$(\nabla_x^+)_2 f(x) = (\nabla_x^+)_1 f(x) - \frac{1}{2}\Delta x \widehat{\nabla_x^2} f(x + \Delta x) \tag{13-22}$$

where $(\nabla_x^+)_1$ is the one-step operator. Use this to prove that the errors are $\mathcal{O}\left(\Delta x^2\right)$.

13.2 Calculate the errors on the various differences.

13.3 Show that the coefficients in the one-sided two-step differences (13-5) are uniquely determined.