**Supplementary information for:**

**Continent-wide estimates of Antarctic strain rates from Landsat 8-derived velocity grids**

Karen E. Alley, Ted A. Scambos, Robert S. Anderson, Harihar Rajaram, Allen Pope, Terry M. Haran

**S1. Strain-rate derivations**

 The 2d strain-rate tensor is defined as:

|  |  |
| --- | --- |
|  | (S1) |

The main text discusses two strain-rate definitions: nominal strain and logarithmic strain. Approximations of the strain-rate tensor are most easily applied as a difference in velocity over some distance, which is directly equivalent to nominal strain. We can see this equivalence by considering the 1d strain-rate tensor, . This is equal to a change in velocity over an initial distance (), which can be further separated into a change in distance () divided by a change in time ():

|  |  |
| --- | --- |
|  | (S2) |

The right-hand side of (S2) can be seen to be equivalent to main text equation (5), which defines the nominal strain-rate.

The two approaches discussed for calculating strain rates, nominal strain and logarithmic strain, can be shown to be equivalent when exact integration is possible. Assume that we have two particles moving in one dimension through a known velocity profile (Figure S1).

|  |
| --- |
|  |
| **Figure S1: The movement of two particles through a linear velocity field.** |

If the velocity profile is expressed by the linear equation , where is a constant, we can differentiate this equation to find the only component of the strain rate tensor:

|  |  |
| --- | --- |
|  | (S3) |

Because the velocity is increasing linearly in space, the strain rate in this velocity field is spatially uniform.

 We have just found the strain rate of the velocity profile using the derivative of the velocity field, which is a direct application of the 1d strain rate tensor. A discretized solution would find the strain rate by differencing the velocities across some distance, which is the nominal strain rate. Now we will derive the strain rate by asking how the distance between the particles changes as they move through the velocity field. This is an integration that uses the definition of logarithmic strain, which takes into account the velocity history of each particle. In the case considered, the velocity profile is linear, making the integration very straightforward. The positions through time of each of the tracked particles are written:

|  |  |
| --- | --- |
|  | (S4) |
|  | (S5) |

Equations S4 and S5 describe the original particle positions sketched in Figure S1 when we apply the initial conditions and .

 At this point, we have defined the exact velocity profile that is experienced by each of the particles we are tracking. Now we wish to integrate over the domain to obtain the displacement of each particle. Following the first particle, we may separate equation S4 to become:

|  |  |
| --- | --- |
|  | (S6) |
|  | (S7) |

We wish to know the position of the particle, so we solve for :

|  |  |
| --- | --- |
|  | (S8) |
|  | (S9) |
|  | (S10) |

Similarly, for , we have:

|  |  |
| --- | --- |
|  | (S11) |

We can find the initial length between the points through simple subtraction of the original point locations:

|  |  |
| --- | --- |
|  | (S12) |

Similarly, we can find the final length by subtracting the integrated equations (S10) and (S11):

|  |  |
| --- | --- |
|  | (S13) |

This reduces to:

|  |  |
| --- | --- |
|  | (S14) |

Now we can plug the initial and final lengths into the derivative of the logarithmic strain defined in main text equation (6). In this case, rather than writing a discrete time step, we will have taken an exact derivative of the strain with respect to time to obtain the strain rate:

|  |  |
| --- | --- |
|  | (S15) |

This reduces to:

|  |  |
| --- | --- |
|  | (S16) |

Thus, where we can calculate exact derivatives, such as in an exactly linear velocity profile, the two definitions of strain rate yield the same result. However, errors may be introduced if finite differences are used without integration, as in the direct differencing of (the definition of nominal strain). Instead, the integrated logarithmic version should be used for large displacements to most accurately approximate true strain values in glacial ice.

**S2. Nominal strain-rate code**

 In the nominal strain-rate code, the 2d strain-rate tensor (main text equation (1)) is approximated by differencing velocity grids and dividing by the offset distance. For example, Figure S2 diagrams the calculation of the x-direction strain rate (). For any given point, such as the one marked with a black dot, a velocity difference is taken between two points at a distance of in the positive and negative x-directions (Figure S2a). In the code, these differences are achieved simultaneously by offsetting the relevant velocity grid from itself by , subtracting the grids, and dividing by the offset distance (Figure S2b). The calculated strain rates are assigned to points at the center of each offset. Combinations of x- and y-direction velocity grids offset in the x- or y-directions are used to find all components of the strain-rate tensor.

|  |
| --- |
|  |
| **Figure S2:** Calculation of x-direction nominal strain rates. a. For calculation of the strain-rate at the black dot, a velocity difference is taken at a distance of on either side of the center point. b. Offsetting of velocity grids as is implemented in the nominal strain-rate code. |

**S3. Least squares approximation**

In order to adapt Nye’s strain square method to a regular velocity grid, we find values for two components oriented horizontally and vertically relative to the grid ( and , corresponding to the *c* and *a* components in the strain square in main text Figure 1), and the shear strain relative to those directions (, which is measured twice corresponding to the *b* and *d* components in main text Figure 1). To calculate strain rates in any direction from these components, we can write the system of n equations corresponding to n strain component measurements, with three coefficients, as shown below:

|  |  |
| --- | --- |
|  | (S17) |

 The application of equation (S17) to our system yields four equations corresponding to the four directions in Nye’s strain square. The three coefficients in each of the four equations in our system can be constructed as a matrix A, which depends on the angles of the strain component measurements taken, relative to the coordinate system that defines the three components we want to calculate. We have deliberately oriented the strain square so that (the *c* component of the strain square) and (the *a* component of the strain square), so the coefficients for these two equations, according to the form of the system of equations written above, are simply and , respectively. The coefficients for and could be determined equally simply if our coordinate system were rotated by 45˚. Therefore, we perform a rotation on the strain-rate tensor as shown below:

|  |  |
| --- | --- |
|  | (S18) |

When the matrix multiplication is completed, the expression for can be found in the upper left entry, and the expression for is in the lower right entry. These expressions are:

|  |  |
| --- | --- |
|  | (S19) |
|  | (S20) |

If we substitute 45˚ for , we get the relevant coefficients for the remaining rows of the matrix , which are and for and , respectively. Then the whole coefficient matrix can be constructed:

|  |  |
| --- | --- |
|  | (S21) |

The ideal system we are trying to solve is then represented by the matrix equation:

|  |  |
| --- | --- |
|  | (S22) |

where is a column vector of the measured strain rate components , , , and , is the matrix of coefficients that depends on the angles at which the measurements were taken, and is a column vector of the three strain rate components we wish to derive: , , and . We wish to invert this equation to solve for .

 However, because we have four measurements to solve for three unknowns, and because the measurements reflect both random errors and real gradients in strain across the strain square, we take error into account in the calculation. Instead of solving an exact system, we solve the equation:

|  |  |
| --- | --- |
|  | (S23) |

where r here is the error. In the least squares method, we wish to minimize the sum of the squared components of the error vector . Therefore, we square this equation, which can be represented as a sum of the error for each equation in the system in matrix notation:

|  |  |
| --- | --- |
|  | (S24) |

The sum of the squared error R will be minimized when the partial derivatives of each component, with respect to the unknowns, are equal to zero. Taking this partial derivative, we get:

|  |  |
| --- | --- |
|  | (S25) |

Therefore, when least squares are minimized, we can solve for the unknowns that minimize error, , using the equation:

|  |  |
| --- | --- |
|  | (S26) |

**S4. Numerical stake-tracking scheme with adaptive time-stepping**

In the logarithmic strain-rate code, each virtual stake in the current strain square is allowed to move at its current velocity for one time-step, in a simple Euler-forward numerical formulation:

|  |  |
| --- | --- |
|  | (S27) |

where the superscripts indicate the relevant time step, with representing the current time step and representing the following time step. At the end of the time step, the lengths of each strain segment are updated () and the strain and strain rate during that time step are calculated using ,.

 After the time step has been completed, an accuracy check is performed. The Euler-forward method calculates the final stake positions by allowing the stakes to move at their initial velocity throughout the time step. However, in reality each stake’s velocity should change continuously as it moves. If the time step and the velocity gradients are both small, the Euler-forward method should have very little impact on the final stake positions. However, in areas with large velocity gradients, a smaller time step may be required to maintain accuracy. Therefore, we use an adaptive time-stepping approach.

At the end of each time step, after each stake’s position is calculated, a second calculation of the new stake position is made using an improved Euler method. In this method, the averaged velocities at the beginning and approximate end positions are used to move the stakes:

|  |  |
| --- | --- |
|  | (S28) |

where is the velocity extracted at the final stake position calculated by the Euler-forward method. The difference between the positions at the next time step, and , can be used as a metric to ensure that errors remain small in the final result. We choose to use a relative error metric:

|  |  |
| --- | --- |
|  | (S29) |

where the tolerance criterion is set by the user (the default is 10-4). If the error in any stake position during the time step exceeds the defined tolerance limit, the time step is reduced by half and the iteration is repeated. If all stake positions are within the tolerance limit, the final strain segment lengths are updated as the new initial lengths, and the running total of strain experienced by each segment is updated. If necessary, the time step is reset.

The stakes are also assigned new current velocities for the next time steps. These velocities are determined using a bilinear interpolation function. Note that velocities and stake movements are calculated using x- and y-components; therefore, changes in the direction of the stakes are automatically applied at each time step. Then the process repeats, moving the stakes according to their new velocities.

**S5:** **Derivation of a uniform strain field**

A non-divergent, non-rotating velocity field exhibits “pure strain.” The non-divergence condition, which also implies that a parcel will not change in volume, is:

|  |  |
| --- | --- |
|  | (S30) |

We can define a velocity field in coordinates that satisfies this condition using the form:

|  |  |
| --- | --- |
|  | (S31) |
|  | (S32) |

From this field, we also know that the shear strain components and must both be equal to zero. That gives us the strain rate tensor:

|  |  |
| --- | --- |
|  | (S33) |

Note that this yields a uniform strain field only when measurements are oriented relative to the x- and y-directions, with , , and .

We can make this velocity field more flexible by allowing rotation through an arbitrary angle to a new set of coordinates . With defined as positive in the counter-clockwise direction, we can relate to using a coordinate system transformation:

|  |  |
| --- | --- |
|  | (S34) |

We also need to obtain equations for , the components of the velocity field along the new coordinates . Therefore, we rotate the vector through the angle :

|  |  |
| --- | --- |
|  | (S35) |

Now we solve for and in terms of and in the coordinate system. First, carry out matrix multiplication to obtain an expression for :

|  |  |
| --- | --- |
|  | (S36) |

Then substitute the velocity field in the original coordinate systems for and :

|  |  |
| --- | --- |
|  | (S37) |

Now we want to apply the coordinate system rotation to substitute for and :

|  |  |
| --- | --- |
|  | (S38) |

This can be simplified using distribution and trig identities:

|  |  |
| --- | --- |
|  | (S39) |
|  | (S40) |

A similar progression is used to derive an expression for :

|  |  |
| --- | --- |
|  | (S41) |

 Using the two final equations for and , we take the partial derivatives , , , and to obtain the final strain rate tensor:

|  |  |
| --- | --- |
|  | (S42) |

Note that we can also add a constant velocity to the pure strain field at some arbitrary angle :

|  |  |
| --- | --- |
|  | (S43) |
|  | (S44) |

Because the added velocity is constant, it does not change the derivatives that comprise the strain rate tensor.

 Flow vectors for a uniform strain field centered on the origin are shown in main text Figure 2. This field was built with equal to 0.01 and and equal to zero. Using these parameters, the uniform strain field yields theoretical values of , , and . Both the differencing and numerical codes reproduce these values. The numerical code introduces a small amount of error in the slowest sections (of order 10^-5 at the maximum); the differencing code is more accurate because its assumption of linear velocity change is accurate for this field.

**S6: Derivation of a flow field around a Rankine half-body**

We begin with a uniform flow in one direction. For convenience, we will limit this flow to the x-direction and give it a magnitude of . Within this flow, we add a source point from which fluid is released evenly in all radial directions. We represent the source strength as:

|  |  |
| --- | --- |
|  | (S45) |

Then in Cartesian coordinates, the equation for flow from the point source can be represented in the x- and y-directions, respectively, as a simple sum of the uniform and source flows:

|  |  |
| --- | --- |
|  | (S46) |
|  | (S47) |

Figure S3 shows the resulting flow vectors produced from these equations. Note that as x and y become very large, the velocity equations reduce to the far-field uniform flow, . The stagnation point, shown as a red dot in Figure S3, is where all velocity goes to zero. As the uniform velocity field is only in the x-direction, we know that this stagnation point will occur directly in the x-direction from the source point, which we have placed at (0, 0), where the x-direction velocities cancel each other out. By setting and solving for with equal to zero in equation (S4), we find the x-coordinate of the stagnation point:

|  |  |
| --- | --- |
|  | (S48) |

We also find the equation of the “surface streamline,” which is the streamline passing through the stagnation point that outlines the surface of the Rankine half-body:

|  |  |
| --- | --- |
|  | (S49) |

To simulate an obstruction within the flow field, which has no flow through or across it, we simply set every value within the surface streamline to NaN.

|  |
| --- |
|  |
| **Figure S3: Uniform flow field with fluid source (flow around a Rankine half-body).** Arrows show direction of motion; length is proportional to speed. Blue lines are streamlines. Blue dot is fluid source. Red dot is fluid stagnation point. Red surface streamline outlines the Rankine half-body that may be treated as an obstruction in the flow field, similar to a nunatak or ice rise in a glacier or ice shelf. |

From the velocity field equations, we can derive the analytical solutions for the components of the strain rate tensor:

|  |  |
| --- | --- |
|  | (S50) |