

Contents

S1 Introduction	2
S2 Model components	3
S2.1 Iceberg model	3
S2.1.1 Iceberg shape and size	3
S2.1.2 Sampling	5
S2.1.3 Iceberg deterioration	7
S2.1.4 Dynamics	12
S2.2 Locations	13
S3 Extra simulations	15
S3.1 Results	15
S3.1.1 Flow patterns and iceberg trajectories	15
S3.1.2 Potential IRD provenance	16
S3.1.3 Minimum iceberg size	17
S3.1.4 Present-day simulation	18
S3.2 Discussion	19
S3.2.1 Flow patterns and iceberg trajectories	19
S3.2.2 Potential IRD provenance	19
S3.2.3 Minimum iceberg size	20
S3.2.4 Iceberg melt rates	20
S3.2.5 Present-day simulation	22
S3.3 Conclusion	22
S4 Visual material	23
S4.1 Location ODP Site 696	23
S4.2 Circumpolar simulation	24
S4.3 Iceberg properties and melt rates for selected trajectories	25
References	27

S1 Introduction

These supplementary materials contain a full description of the components of the iceberg model (Chapter S2) in addition to the methods described in the main text. In Chapter S3, additional results of simulations not included in the main text are described and placed in context with the main simulations. Finally, section S4 shows additional figures referred to in the main text or sections in this document.

S2 Model components

S2.1 Iceberg model

The iceberg model was built using the Lagrangian framework Parcels (Delandmeter and Van Sebille, 2019) version 2.4, in which the particle dynamics are described using kernels. In the section below, the kernels defined in this study are explained in more detail.

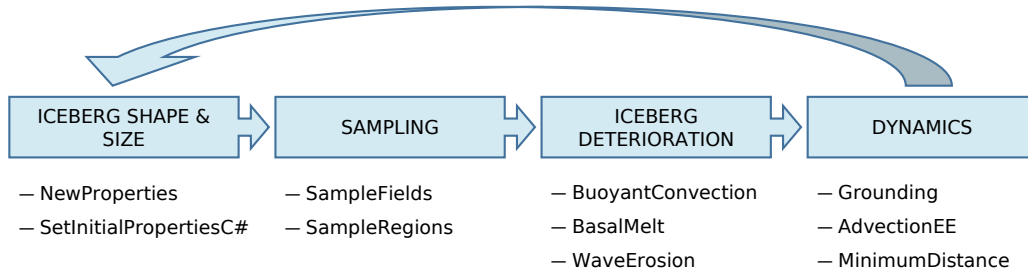


Figure S1: The order and names of kernels used in the iceberg model.

S2.1.1 Iceberg shape and size

The first set of kernels describe the iceberg properties. The kernel SetInitialPropertiesC# (Listing S1), with # the iceberg size class, defines the initial length, width, thickness, mass and draft of the iceberg for the given size class - unless already defined in a previous timestep.

```

1 def SetInitialPropertiesC4(particle, fieldset, time):
2     '''
3     Set the initial properties of the iceberg using:
4     W = L/1.5
5     M = L * W * T * rho_ice
6     D = T * (rho_ice/rho_ocean)
7     for size class C4.
8     '''
9
10    if particle.check < 0.:
11        particle.L = 10000.                # length [m]
12        particle.W = particle.L/1.5        # width [m]
13        particle.T = 500.                  # thickness [m]
14
15        particle.M = particle.L * particle.W * particle.T * fieldset.rho_i # mass [kg]
16
17        particle.depth = particle.T * (fieldset.rho_i/fieldset.rho_o) # draft [m]
18
19        particle.prev_L = particle.L
20        particle.prev_W = particle.W
21        particle.prev_T = particle.T
22
23        particle.check = 1.
24
25    particle.prev_check = particle.check

```

Listing S1: Code of the kernel defining the properties of iceberg size class C4.

In the second kernel in this category, NewProperties (Listing S2), the new iceberg dimensions after melting are calculated - provided the iceberg was defined already in a previous timestep. By calculating the melt at each surface, the change in iceberg volume can be determined. In addition, as basal melt is the only melt process influencing the iceberg base, a new iceberg thickness can be calculated using the basal melt rate. As the length-to-width ratio is set to remain constant

(1.5 : 1), the new length and width of the iceberg follow from the new iceberg volume and thickness. Finally, at the end of this kernel, the capsizing criterion is applied (Wagner et al., 2017b). If needed, the thickness is subsequently updated to the dimension of the iceberg width and the iceberg length and width are recalculated from the iceberg volume.

Note that while generally, each round of execution of the kernels will start anew, here, the first kernel (NewProperties) depends on the melt rates defined - and thus field sampling performed - in the previous timestep. In other words, the preparation for the calculation of the new iceberg dimensions has been performed in the previous time step.

```

1 def NewProperties(particle, fieldset, time):
2     """
3     Calculates the new properties of the iceberg from the melt terms.
4     """
5
6     if particle.check > 0.:
7         ## Calculate old thickness
8         draft = particle.prev_T * (fieldset.rho_i/fieldset.rho_o)
9
10        ## Calculate areas for ...
11        Ab = particle.prev_L * particle.prev_W # ... basal melt [m3]
12        Av = 2. * draft * (particle.prev_L +
13                          particle.prev_W) # ... buoyant convection on all sides below SL [m3]
14        Ae = particle.prev_T * (particle.prev_L
15                              + particle.prev_W) # ... wave erosion on two sides [m3]
16
17        ## Volume changes
18        dV = (particle.prev_Mbr*Ab + particle.prev_Mvr*Av
19              + particle.prev_Mer*Ae) * particle.dt # total volume loss [m3]
20        Vol = (particle.prev_L * particle.prev_W *
21              particle.prev_T) - dV # new volume [m3]
22        if Vol < 0.:
23            particle.delete()
24
25        ## New dimensions
26        particle.T = particle.prev_T - particle.prev_Mbr*particle.dt # new thickness [m]
27        A = Vol/particle.T # horizontal area [m2]
28        particle.W = math.sqrt(A/1.5) # new width [m]
29        particle.L = 1.5 * particle.W # new length [m]
30        particle.depth = particle.T * (fieldset.rho_i/fieldset.rho_o) # new draft [m]
31
32        ## New mass
33        if particle.L >= 0 and particle.W >= 0 and particle.T >= 0:
34            particle.M = particle.T * particle.W * particle.L *
35                      fieldset.rho_i # iceberg mass [kg]
36
37            ep_crit = math.sqrt(6. * (fieldset.rho_i/fieldset.rho_o) *
38                              (1. - (fieldset.rho_i/fieldset.rho_o)))
39            ep = particle.W/particle.T
40
41            if ep < ep_crit: # iceberg tipping (WDE17)
42                particle.T = particle.W # calculate new thickness [m]
43                A = Vol/particle.T # keep volume and ratios correct
44                particle.W = math.sqrt(A/1.5) # calculate new width [m]
45                particle.L = 1.5 * particle.W # calculate new length [m]
46                particle.depth = particle.T * (
47                    fieldset.rho_i/fieldset.rho_o) # new draft [m]
48            else:
49                particle.delete()
50
51        particle.prev_L = particle.L
52        particle.prev_W = particle.W
53        particle.prev_T = particle.T

```

Listing S2: Code of the kernel updating the iceberg properties.

S2.1.2 Sampling

After the iceberg dimensions have either been updated or defined, sampling of the ocean and atmospheric fields is performed at the location of the iceberg in the kernel `SampleFields` (Listing S3). Note that sampling is performed at time $t + \frac{1}{2}dt$ as iceberg melt occurs continuously between two timesteps. First, the surface wind stress components τ_x and τ_y are sampled to approximate the surface wind velocity as this field is not available from the high-resolution model. In general, the relation between wind velocity and wind stress is given by

$$\vec{\tau} = \rho_a C_d \vec{u}_a |\vec{u}_a| \quad (\text{S1})$$

and hence we obtain

$$u_a = \frac{\tau_x}{\rho_a C_d |\vec{u}_a|} \quad (\text{S2})$$

$$v_a = \frac{\tau_y}{\rho_a C_d |\vec{u}_a|} \quad (\text{S3})$$

where we can use the approximation

$$|\vec{u}_a| = \pm \left(\frac{\sqrt{\tau_x^2 + \tau_y^2}}{\rho_a C_d} \right)^{\frac{1}{2}} \quad (\text{S4})$$

Here, $\rho_a = 1.293 \text{ kg m}^{-3}$ is the air density and $C_d = 0.0015$ is the wind drag coefficient.

Next, the water depth at the location of the iceberg is sampled. To avoid sampling below the seafloor during the calculation of the basal and depth-integrated values, the minimum of the water depth and iceberg draft, called z , is determined. Then, after sampling the ocean surface temperature (T_s) and velocity components (u_s, v_s), the ocean properties at the base (depth z) of the iceberg are determined (i.e. T_b, u_b, v_b).

Finally, the integrated ocean velocities (u_d, v_d) and temperature (T_d) are calculated following

$$\vec{u}_d = \frac{1}{D} \sum_{i=0}^{\min(d,D)} \vec{u}_i dz \quad (\text{S5})$$

with D the iceberg draft, d the water depth, \vec{u}_i the velocity (or temperature) in the current layer, and dz the thickness of the current layer - or, if the base of the iceberg lies between two vertical layers, the thickness of the current layer till the iceberg base.

```

1 def SampleFields(particle, fieldset, time):
2     '''
3     At the iceberg's location, sample:
4     1) the surface wind field,
5     2) the ocean bathymetry,
6     3) the surface ocean temperature and velocity,
7     4) the ocean temperature and velocity at the iceberg base, and
8     5) the depth-integrated ocean temperature and velocity along the iceberg's draft.
9     '''
10
11     ### Wind: surface wind stress components and surface wind speed
12     tau_x, tau_y = fieldset.XY[time+(particle.dt/2.), 0., particle.lat, particle.lon]
13     # [g/(s2 cm)]
14     vela = math.sqrt((math.sqrt(tau_x**2+tau_y**2)*0.1)/(fieldset.rho_a*0.0015)) # [m/s]
15     particle.uvela = (tau_x*0.1)/(fieldset.rho_a*0.0015*math.fabs(vela)) # [m/s]
16     particle.vvela = (tau_y*0.1)/(fieldset.rho_a*0.0015*math.fabs(vela)) # [m/s]
17
18     ### Bathymetry
19     particle.bath = fieldset.B[time+(particle.dt/2.), particle.depth, particle.lat,
20     particle.lon] # [m]
```

```

20  ## Determine the deepest point between the iceberg's base and the bathymetry
21  z = min(particle.depth, math.fabs(particle.bath))
22
23  ### Surface (ocean): ocean temperature and velocity components at the surface
24  particle.tempS = fieldset.T[time+(particle.dt/2.), 0., particle.lat, particle.lon]
25  # [°C]
26  uvel, vvel = fieldset.UV[time+(particle.dt/2.), 0., particle.lat, particle.lon]
27  # [°/s]
28  particle.uveloS = uvel*1852*60*math.cos(particle.lat*math.pi/180.) # [m/s]
29  particle.vveloS = vvel*1852*60 # [m/s]
30
31  ### Basal (ocean): ocean temperature and velocity components at the iceberg base
32  particle.tempB = fieldset.T[time+(particle.dt/2.), z, particle.lat, particle.lon]
33  # [°C]
34  uvel, vvel = fieldset.UV[time+(particle.dt/2.), z, particle.lat, particle.lon]
35  # [°/s]
36  particle.uveloB = uvel*1852*60*math.cos(particle.lat*math.pi/180.) # [m/s]
37  particle.vveloB = vvel*1852*60 # [m/s]
38
39  ### Integrated (ocean): depth-integrated ocean temperature and velocity components
40  along the iceberg's draft
41  t, u, v = 0, 0, 0
42  check = 1
43
44  if z >= 10.01244 and check > 0:
45      temp = fieldset.T[time+(particle.dt/2.), 5.00622, particle.lat, particle.lon]
46      uv, vv = fieldset.UV[time+(particle.dt/2.), 5.00622, particle.lat, particle.lon]
47      t += temp * (10.01244-0.0)
48      u += uv * (10.01244-0.0)
49      v += vv * (10.01244-0.0)
50  elif z < 10.01244 and check > 0:
51      temp = fieldset.T[time+(particle.dt/2.), 5.00622, particle.lat, particle.lon]
52      uv, vv = fieldset.UV[time+(particle.dt/2.), 5.00622, particle.lat, particle.lon]
53      t += temp * (z-0.0)
54      u += uv * (z-0.0)
55      v += vv * (z-0.0)
56      check = -1
57
58  .
59  .
60  .
61
62  if z >= 644.3267 and check > 0:
63      temp = fieldset.T[time+(particle.dt/2.), 579.30725, particle.lat, particle.lon]
64      uv, vv = fieldset.UV[time+(particle.dt/2.), 579.30725, particle.lat, particle.
65      lon]
66      t += temp * (644.3267-514.2877)
67      u += uv * (644.3267-514.2877)
68      v += vv * (644.3267-514.2877)
69  elif z < 644.3267 and check > 0:
70      temp = fieldset.T[time+(particle.dt/2.), 579.30725, particle.lat, particle.lon]
71      uv, vv = fieldset.UV[time+(particle.dt/2.), 579.30725, particle.lat, particle.
72      lon]
73      t += temp * (z-514.2877)
74      u += uv * (z-514.2877)
75      v += vv * (z-514.2877)
76      check = -1
77
78  if z >= 814.37573 and check > 0:
79      temp = fieldset.T[time+(particle.dt/2.), 729.35126, particle.lat, particle.lon]
80      uv, vv = fieldset.UV[time+(particle.dt/2.), 729.35126, particle.lat, particle.
81      lon]
82      t += temp * (814.37573-644.3267)
83      u += uv * (814.37573-644.3267)
84      v += vv * (814.37573-644.3267)
85  elif z < 814.37573 and check > 0:
86      temp = fieldset.T[time+(particle.dt/2.), 729.35126, particle.lat, particle.lon]
87      uv, vv = fieldset.UV[time+(particle.dt/2.), 729.35126, particle.lat, particle.
88      lon]
89      t += temp * (z-644.3267)
90      u += uv * (z-644.3267)
91      v += vv * (z-644.3267)
92      check = -1

```

```

84     particle.tempD = t / z                                # [°C]
85     uvel = u / z                                         # [°/s]
86     vvel = v / z                                         # [°/s]
87
88     particle.uveloD = uvel*1852*60*math.cos(particle.lat*math.pi/180.) # [m/s]
89     particle.vveloD = vvel*1852*60                       # [m/s]
90

```

Listing S3: Code of the kernel sampling the model fields.

In a second kernel (SampleRegion, Listing S4) used for backward simulations, the coastal regions used in this study (Fig. 2.2 in the main text, see section S2.2) are sampled.

```

1  def SampleRegion(particle, fieldset, time):
2      '''
3      At the iceberg's location, sample the region (see Carter et al., 2017)
4      '''
5
6      particle.reg = fieldset.R[time+(particle.dt/2.), 0., particle.lat, particle.lon]

```

Listing S4: Code of the kernel sampling the coastal regions.

Lastly, it is important to mention here that as the model is run in the faster JIT mode, we cannot use a for loop and hence the model depth-levels are defined implicitly in the depth-integration. As the depth layers from other models are likely different, a new kernel should be defined. Hence, we defined a kernel SampleFieldsModern for the present-day comparison simulations using the Mercator Ocean International (MOi) hydrodynamics dataset (Gasparin et al., 2018) and ECMWF ERA5 reanalysis data (Hersbach et al., 2023).

S2.1.3 Iceberg deterioration

In the next group of kernels, the different melt processes are described. All melt rates are calculated in units of m d^{-1} unless indicated differently.

Basal melt

As described in the main text, one of the main processes of iceberg melt is basal melting (Listing S5), which is given by the following equation (adapted from Merino et al., 2016):

$$M_b = C |\vec{u}_d - \vec{u}_b|^{0.8} \frac{T_b - T_f}{L^{0.2}} \quad (\text{S6})$$

Here, C is a dimensional constant equal to $C = 0.58 \text{ } ^\circ\text{C}^{-1} \text{ m}^{0.4} \text{ d}^{-1} \text{ s}^{0.8}$ for oceanic conditions around $T_0 = 0^\circ\text{C}$ (Weeks and Campbell, 1973). The higher temperatures of the late Eocene, however, might lead to a different value of C . Indeed, it was shown that for laboratory conditions of $T_0 = 20^\circ\text{C}$ the constant increased to $C = 0.75 \text{ } ^\circ\text{C}^{-1} \text{ m}^{0.4} \text{ d}^{-1} \text{ s}^{0.8}$ (FitzMaurice et al., 2017), suggesting that basal melt rates during the Eocene might be slightly underestimated by using the standard value of C .

The variable T_f is the freezing temperature, which was shown to give better results than using the internal ice temperature of $T_i = -4^\circ\text{C}$ (FitzMaurice and Stern, 2018). Following their approach, we approximated T_f using $T_f \approx \alpha S_o + \beta$, where $\alpha = -5.73 \times 10^{-2} \text{ } ^\circ\text{C PSU}^{-1}$ and $\beta = 9.39 \times 10^{-2} \text{ } ^\circ\text{C}$. Although it would be possible to determine the salinity S_o and hence the freezing temperature T_f for every iceberg and timestep individually, this would require adding the ocean salinity field to the model. As we expect Southern Ocean variability in salinity to be relatively minor, we here opt to determine an average salinity for the Weddell Sea region based on a single field (model day 380101) to limit data usage. This yielded a salinity of $S = 35.06 \text{ PSU}$, and hence $T_f = -1.92^\circ\text{C}$.

Finally, note that basal melt is reduced to zero where $\vec{u}_d = \vec{u}_b$. Generally, this occurs when the wind speed is very small (El-Tahan et al., 1987), but as we only use iceberg advection based on the ocean flow (section S2.1.4), basal melt cannot be computed in the surface layer (10 m in the Eocene model) and hence icebergs will not decrease once their draft decreases below 10 m. Similarly, when running a simulation using only surface properties, in principle no basal melt can be calculated. However, to allow for some inclusion of basal melt, we determine a mean velocity from the (top of) the second layer and the surface velocity itself.

```

1 def BasalMelt(particle, fieldset, time):
2     '''
3     Calculates the mass loss rate [m/s] through basal melt at the iceberg base from:
4     Mb = C * |v_ice - v_ocean|^0.8 * (T_ocean - T_ice) / (L^0.2)
5     with
6     C = 0.58 [°C^-1 m^0.4 d^-1 s^0.8]
7     v_ice: iceberg velocity (depth-integrated) [m/s]
8     v_ocean: ocean velocity at the iceberg base [m/s]
9     T_ocean: ocean temperature at the iceberg base [°C]
10    T_ice = -1.92 [°C]; freezing temperature
11    L: iceberg length [m]
12    See FitzMaurice & Stern (2018) and references therein.
13    '''
14
15    dvelabs = math.sqrt((particle.uveloD-particle.uveloB)**2 +
16                       (particle.vveloD-particle.vveloB)**2)
17
18    Mbr = 0.58 * (dvelabs**0.8) * (
19        (particle.tempB-fieldset.Ti)/(particle.L**0.2)) # [m/d]
20    particle.Mbr = Mbr/fieldset.sec.to_day # [m/s]
21    particle.prev_Mbr = particle.Mbr

```

Listing S5: Code of the (bulk) basal melt kernel.

An alternative to the parameterisation described above, also called the bulk parameterisation and which is usually used within the iceberg community, is the three-equation parameterisation (Holland and Jenkins, 1999). This set of equations is generally applied to ice shelves and includes equations for the dependency of the freezing point on pressure and salinity, and the conservation of heat and salt. Hence, the inclusion of these equations would require the use of both salinity and pressure fields and thus would be more computationally expensive. Therefore, we decided against using the full three-equation system in our simulations.

However, in their comparison of the basal melt equations, FitzMaurice and Stern (2018) proposed a slight alteration of the bulk parameterisation for large tabular icebergs (Listing S6). They found that while for small icebergs the bulk parameterisation provided accurate results, the equation deviated strongly for icebergs larger than the Rossby deformation radius, which is roughly $R_d \approx 15$ km in the polar oceans. For these situations, they proposed an alternative calculation that approximates the full three-equation parameterisation as

$$M_b = \frac{\gamma_{T,3EM} (aT_o + b - T_f)}{\rho_i L_f} \quad (S7)$$

where

$$\gamma_{T,3EM} = \frac{\rho_o c_{po} \sqrt{c_d} (cT_o + d)}{\Gamma_t + \Gamma_m} \quad (S8)$$

is the heat transfer coefficient. In turn, this coefficient is dependent on the turbulent exchange parameter (Holland and Jenkins, 1999)

$$\Gamma_t = \frac{1}{k} \ln \left(\frac{u_* \xi_N \eta_*^2}{f h_\nu} \right) + \frac{1}{2 \xi_N \eta_*} - \frac{1}{k}$$

with friction velocity

$$u_* = \sqrt{c_d} u$$

stability parameter

$$\eta_* = \left(1 + \frac{\xi_N u_*}{f L_o R_c}\right)^{-1/2} \quad (\text{S9})$$

and sublayer thickness

$$h_\nu = 5 \frac{\nu}{u_*}$$

Finally, the molecular exchange parameter

$$\Gamma_m^T = 12.5 \text{Pr}^{2/3} - 6$$

depends on the Prandtl number

$$\text{Pr} = \frac{\nu \rho_o c_p}{k_o^T}$$

The parameters and variables used in these equations are shown in Table S1 below. However, the definition of the stability parameter (eq. S9) depends on the Monin-Lobokuv length L_o , which in turn depends on the buoyancy flux and hence cannot be determined from the current fields. Instead, we estimate the order of size for L_o by reproducing the figures of FitzMaurice and Stern (2018), which gives $L_o \sim 100$ m. This is also similar to the Obukhov length found in the Arctic ocean by Fer and Sundfjord (2007).

Table S1: Parameters used in the alternative basal melt equation (FitzMaurice and Stern, 2018) in this study.

Parameter	Value	Unit
a	0.7 ± 0.1	—
b	-1.2 ± 0.1	$^{\circ}\text{C}$
c	0.004 ± 0.003	$\text{m s}^{-1} ^{\circ}\text{C}^{-1}$
d	0.02 ± 0.01	m s^{-1}
Drag coefficient, c_d	1.5×10^{-3}	—
Specific heat capacity of ocean, c_{po}	4004	$\text{J kg}^{-1} ^{\circ}\text{C}^{-1}$
Coriolis parameter, f		s^{-1}
Viscous sublayer thickness, h_ν		m
von Kármán's constant, k	0.40	—
Thermal conductivity ocean, k_o^T	0.563	W m^{-1}
Latent heat of fusion of ice, L_f	3.34×10^5	J kg^{-1}
Obukhov length, L_o		m
Basal melt, M_b		m s^{-1}
Prandtl number, Pr		—
Critical flux Richardson number, R_c	0.20	—
Freezing temperature, T_f	-1.92	$^{\circ}\text{C}$
Ocean temperature, T_o		$^{\circ}\text{C}$
Velocity, u		m s^{-1}
Friction velocity, u_*		m s^{-1}
Heat transfer coefficient, $\gamma_{T,3EM}$		$\text{W m}^{-2} \text{K}^{-1}$
Stability parameter, η_*		—
Kinematic viscosity seawater, ν	1.826×10^{-6}	$\text{m}^2 \text{s}^{-1}$
Iceberg density, ρ_i	850.0	kg m^{-3}
Ocean density, ρ_o	1027.5	kg m^{-3}
Stability constant, ξ_N	0.052	—
Molecular heat transfer parameter, Γ_m		—
Turbulent heat transfer parameter, Γ_t		—

Finally, in between the large and small regimes, a gradual transition between the two parametrisation should occur. The exact form of this matching, however, is not known. Hence, a jump in melt rates will occur at an iceberg length of 15 km if no extra adaptations are made. For consistency between the simulations and with previous studies, we therefore do not implement the adapted basal melt equation in the main simulations. However, we do run an extra simulation for size class C5 for comparison, which is analysed in Chapter S3.

```

1 def BasalMeltAlt(particle, fieldset, time):
2     '''
3     Calculates the mass loss rate [m/s] through basal melt at the iceberg base following
4     the bulk equation when its length is smaller than 1E2 m and the adapted equation
5     when larger.
6     '''
7
8     if particle.L <= 15000.:
9         dvelabs = math.sqrt((particle.uveloD-particle.uveloB)**2 +
10                             (particle.vveloD-particle.vveloB)**2)
11
12         Mbr = 0.58 * (dvelabs**0.8) * ((particle.tempB-fieldset.Ti)/
13                                         (particle.L**0.2))                # [m/d]
14         particle.Mbr = Mbr/fieldset.sec.to_day                            # [m/s]
15         particle.prev_Mbr = particle.Mbr
16     else:
17         f = math.fabs(2 * fieldset.Om * math.sin(particle.lat*math.pi/180.))
18         u_st = math.sqrt(fieldset.cd) * math.sqrt(particle.uveloD**2 +
19                                                     particle.vveloD**2)
20         eta_st = 1./math.sqrt(1 + fieldset.xiN*u_st/(f*fieldset.Lo*fieldset.Rc))
21         h_nu = 5*fieldset.nu/u_st
22
23         GammaT = fieldset.k_r*math.log(
24             u_st*fieldset.xiN*eta_st**2/(f*h_nu)) + 1./((
25                 2*fieldset.xiN*eta_st) - fieldset.k_r)                # [-]
26         GammaM = 12.5 * fieldset.Pr**(2./3.) - 6                      # [-]
27         gamma_T3EM = (fieldset.rho_o*fieldset.cpo*math.sqrt(fieldset.cd)*(
28             0.004*particle.tempS+0.02)) / (GammaT + GammaM)          # [W/(m2 K)]
29
30         particle.Mbr = gamma_T3EM * ((0.7*particle.tempS-1.2-fieldset.Ti)/
31                                         (fieldset.rho_i * fieldset.Lf))    # [m/s]
32         particle.prev_Mbr = particle.Mbr

```

Listing S6: Code of the alternative (three-equation based) basal melt kernel.

Buoyant convection

A second kernel calculates buoyant convection (Listing S7) along the iceberg sides as given by an empirical fit to observations (El-Tahan et al., 1987) following:

$$M_v = 7.62 \times 10^{-3} T_d + 1.29 \times 10^{-3} T_d^2 \quad (\text{S10})$$

The strong dependence on temperature makes it a negligible term in cold waters (Cenedese and Straneo, 2023).

```

1 def BuoyantConvection(particle, fieldset, time):
2     '''
3     Calculates the mass loss rate [m/s] through buoyant convection at the icebergs sides
4     following:
5     Mv = b1 * T + b2 * T^2
6     with
7     b1 = 7.62E-3 [m/(d °C)]
8     b2 = 1.29E-3 [m/(d °C)]
9     T: depth-integrated temperature [°C]
10    See also Merino et al. (2016).
11    '''
12
13    if particle.depth < math.fabs(particle.bath):                ## if not
14        grounded                                                grounded
15        Mvr = 7.62e-3 * particle.tempD + 1.29e-3 * (particle.tempD)**2    # [m/d]
16    else:                                                        ## if grounded

```

```

15     # meltrate till seafloor
16     Mv1 = 7.62e-3 * particle.tempD + 1.29e-3 * (particle.tempD)**2
17     Mvr1 = Mv1 * math.fabs(particle.bath)
18     # meltrate 'below' seafloor
19     Mv2 = 7.62e-3 * particle.tempB + 1.29e-3 * (particle.tempB)**2
20     Mvr2 = Mv2 * (particle.depth-math.fabs(particle.bath))
21     # total meltrate
22     Mvr = (Mvr1+Mvr2)/particle.depth # [m/d]
23
24     particle.Mvr = Mvr/fieldset.sec_to_day # [m/s]
25     particle.prev_Mvr = particle.Mvr

```

Listing S7: Code of the buoyant convection kernel.

Wave erosion

Finally, part of the iceberg sides are subject to wave erosion (Listing S8) as waves of relatively warm water cause melt and break-off parts of the iceberg. This is parameterised by:

$$M_e = \frac{1}{6} S_s \zeta (T_s - -2.) \quad (\text{S11})$$

where

$$S_s = \frac{3}{2} \sqrt{|\vec{u}_a - \vec{u}_s|} + \frac{1}{10} |\vec{u}_a - \vec{u}_s|$$

describes the sea state, and

$$\zeta = \frac{1}{2} (1 + \cos(\pi A_i^3))$$

sea ice dampening. Note that this dampening effect was likely absent in the sea ice-free Eocene as described in the main text.

```

1  def WaveErosion(particle, fieldset, time):
2      '''
3      Calculates the mass loss rate [m/s] through wave erosion at the icebergs sides:
4      Me = 1/12 Ss * (1+cos(pi*A.ice^3)) * (T_ocean + 2)
5      with sea state:
6      Ss = 3/2 (|v_atm - v_ocean|)^0.5 + 1/10 |v_atm - v_ocean|
7      and
8      A.ice: fractional sea-ice area [m2]; negligible in the Eocene
9      T_ocean: ocean surface temperature [°C]
10     v_atm: surface wind velocity [m/s]
11     v_ocean: ocean surface velocity [m/s]
12     '''
13
14     dvelabs = math.sqrt((particle.uvela-particle.uveloS)**2 +
15                        (particle.vvela-particle.vveloS)**2)
16     Ss = (3./2.) * math.sqrt(dvelabs) + (1./10.) * dvelabs
17
18     Ai = 0.
19     damping = 0.5 * (1 + math.cos(math.pi * Ai**3))
20
21     Mer = (1./6.) * Ss * damping * (particle.tempS - -2.) # [m/d]
22     particle.Mer = Mer/fieldset.sec_to_day # [m/s]
23     particle.prev_Mer = particle.Mer

```

Listing S8: Code of the wave erosion kernel.

Other melt terms

Other factors that influence iceberg mass loss, but are not included in the model here, include surface melting by solar insolation and iceberg breakup due to fracturing.

While melting by solar insolation is very small (El-Tahan et al., 1987) and hence can be neglected, iceberg breakup due to fracturing can have a large impact but is not well constrained (England et al., 2020). Especially for large icebergs ($L > 18$ km), fracturing can be responsible for 80% of the mass loss (Tournadre et al., 2015, 2016), and hence exclusion of this process leads to overestimation of iceberg lifetimes (England et al., 2020).

S2.1.4 Dynamics

Next to iceberg melt, the dynamical processes influencing the iceberg are a second main component of the iceberg model. This includes iceberg grounding, (depth-integrated) iceberg advection, and a calculation of the distance to ODP Site 696 from the iceberg's location.

Grounding

During forward simulations, iceberg grounding is included by changing the depth-integrated velocity components to zero when the iceberg keel would extend below the seafloor (Listing S9). Note that it is important to run the grounding kernel after the deterioration kernels as some of the melt terms depend on \vec{u}_d .

```

1 def Grounding(particle, fieldset, time):
2     """
3     Set the depth-integrated velocity to zero when an iceberg has run aground.
4     """
5
6     if particle.depth >= math.fabs(particle.bath):
7         particle.uveloD = 0.
8         particle.vveloD = 0.

```

Listing S9: Code of the iceberg grounding kernel.

Advection

In the model, the icebergs are advected in the kernel AdvectionEE (Listing S10) by the depth-integrated ocean velocity using an Euler forward scheme. This is a simplification, however, as in reality, more factors play a role in the dynamics of an iceberg.

Generally, forces included are the air, water and sea-ice drag, and the Coriolis, wave-radiation and pressure force, including the effect of the sea surface slope (e.g. Martin and Adcroft, 2010). Nevertheless, the movement of large icebergs is dominated by the Coriolis and sea surface slope forces (Rackow et al., 2017) and hence large icebergs - or icebergs in the absence of strong winds (Wagner et al., 2017a) - generally move with the water. Specifically, Wagner et al. (2017a) found that for icebergs where $L > 12$ km, the wind force amounted to less than 10% of the iceberg motion. Therefore, for icebergs in the range of size class C5 and to a slightly lesser extent C4, using solely ocean velocities to simulate the iceberg dynamics gives likely a suitable representation.

On the other hand, the momentum balance of smaller icebergs is often dominated by the oceanic and especially the atmospheric drag terms (Rackow et al., 2017). For icebergs with $L < 200$ m, the impact of wind drag can be three times as large as that of ocean drag causing icebergs to drift at roughly 2% of the surface wind speed relative to the water (Wagner et al., 2017a). Although Kubat and Sayed (2005) found that using depth-integrated ocean velocities instead of surface velocities to simulate drift significantly improves the iceberg trajectories, we can expect some deviation in the trajectories of the smaller iceberg size classes (C1, C2) due to exclusion of atmospheric drag.

```

1 def AdvectionEE(particle, fieldset, time):
2     """
3     Advection of icebergs using Explicit Euler (aka Euler Forward) integration
4     using the depth-integrated ocean velocity components.
5     """
6
7     particle.lon += particle.uveloD/(1852*60*math.cos(particle.lat*math.pi/180.)) *
8     particle.dt
9     particle.lat += particle.vveloD/(1852*60) * particle.dt

```

Listing S10: Code of the depth-integrated iceberg advection kernel.

Distance to ODP Site 696

The final kernel is not dynamical perse, but depends on the geographical position of the iceberg and hence the dynamics. In this kernel, MinimumDistance (Listing S11), the minimal distance to ODP Site 696 is determined. As we are only interested in a distance on the order of ~ 10 km, it suffices to use an equirectangular approximation for the distance calculation, i.e.:

$$\Delta\varphi = |\varphi_{IB} - \varphi_{ODP696}| \quad (S12)$$

$$\Delta\lambda = |\lambda_{IB} - \lambda_{ODP696}| \cdot \cos\left(\frac{\varphi_{IB} + \varphi_{ODP696}}{2} \frac{\pi}{180^\circ}\right) \quad (S13)$$

$$d = \sqrt{\Delta\varphi^2 + \Delta\lambda^2} \cdot R_{earth} \cdot \frac{\pi}{180^\circ} \quad (S14)$$

with φ latitude and λ longitude, and $R_{earth} = 6371 \times 10^3$ m the radius of the Earth.

During post-processing, the icebergs that come within either one (main simulations) or two (sensitivity simulation) times the grid cell distance (~ 11 km) of ODP 696 are assumed to potentially deposit IRD at the site. However, the location of ODP Site 696 is not always straightforward (section S2.2).

```

1 def MinimumDistance(particle, fieldset, time):
2     '''
3     Calculate the minimum distance of the iceberg to ODP Site 696.
4     '''
5
6     lat_dist = math.fabs(particle.lat - fieldset.ODP_lat) # vertical dist. [°]
7     lon_dist = math.fabs(particle.lon - fieldset.ODP_lon) * (math.cos(
8         ((particle.lat + fieldset.ODP_lat)/2.)*(math.pi/180.))) # horizontal dist.[°]
9
10    c = 6371e3 * (math.pi/180.)
11    dist = math.sqrt(lat_dist**2 + lon_dist**2) * c # total distance [m]
12    if dist < particle.prev_dist: # find minimum distance
13        particle.distance = dist
14    else:
15        particle.distance = particle.prev_dist
16    particle.prev_dist = particle.distance

```

Listing S11: Code of the kernel in which the minimum distance to ODP Site 696 is determined.

S2.2 Locations

As stated in the main text, the location of ODP Site 696, the forward and backward release locations, and coastal regions based on Carter et al., 2017 (2017) needed to be defined for the simulations.

First, we defined the position of ODP Site 696 by comparing the present-day bathymetry (e.g. Fig. 1 in López-Quirós et al., 2021) with the Eocene bathymetry of the SOM (Fig. S8) as we could not use a plate reconstruction program such as GPlates due to the difference in reference frame (Baatsen et al., 2016; van de Lagemaat et al., 2021). Instead, by using the paleoceanographic reconstruction of López-Quirós et al. (2021) as a first-order indication, we obtained an approximate position of ODP Site 696 at $67^\circ 5'S$, $57^\circ W$.

For easier analysis of the data of backward simulations, we defined several regions along the Antarctic coast, for which we follow Carter et al., 2017 (2017). In their study, they divide the southern Weddell Sea coast into regions relating to the Ellsworth Mountains, Offshore Filchner Ice shelf, the Coats Land Shelf, and the Dronning Maud Land Shelf. By comparing the present-day and Eocene bathymetry, these regions can be defined in the Eocene (main text Fig. 2.2). In total, this leads to the following six regions: Palmer Land which was not defined by Carter et al. (2017), Ellsworth Mountains, offshore Filchner Ice Shelf, Coats Land shelf, and the western and

eastern sectors of the Dronning Maud Land shelf as the latter was not completely included by Carter et al., 2017 (2017).

During forward simulations, icebergs are released from the specified regions. Specifically, 25 particles were placed at the 500 m bathymetry contour along this section of the coast with a 5° zonal spacing except along the Antarctic Peninsula where a 2° meridional spacing was used instead. For the backward simulations, the grid cell of ODP Site 696 was seeded with 25 points.

S3 Extra simulations

In addition to the primary results described in the main text, other simulations were run to test specific aspects of the model. During the initial phase of the model development, passive particle simulations were performed to study the surface flow. In addition, the sensitivity of the depth-integrated simulations was tested for 1) a shift in the position of ODP Site 696 by 0.5° to the south and 1° to the west (Fig. S8) for size class C4, and 2) the effect of using the adapted basal melt equation for size class C5. Finally, the present-day simulations are shown here.

S3.1 Results

S3.1.1 Flow patterns and iceberg trajectories

In general, the trajectories from the simulations here are similar to those found in the main text (Fig. S2). However, in the simulation of passive particles, particles seem to leave the domain eastwards around 57°S more easily and can move westward through Drake Passage. Still, no particles released east of Gunnerus Ridge in the western sector of Dronning Maud Land seem to be able to reach ODP Site 696.

Inducing a southwestward shift in the position of ODP Site 696 seems to decrease the number of successful trajectories (Fig. S2) compared to the original position used in the main text, leaving only Coats Land as a viable release location. Finally, during the simulation of icebergs of size class C5 with the adapted basal melt equation, trajectories from all regions seem able to reach ODP Site 696, but again none from east of Gunnerus Ridge. In addition, these initially quite large icebergs are able to reach onto shallow regions, such as over Gunnerus Ridge and the Filchner Ice Shelf, which was not seen in the main simulation. In addition, some icebergs can travel westward through Drake Passage.

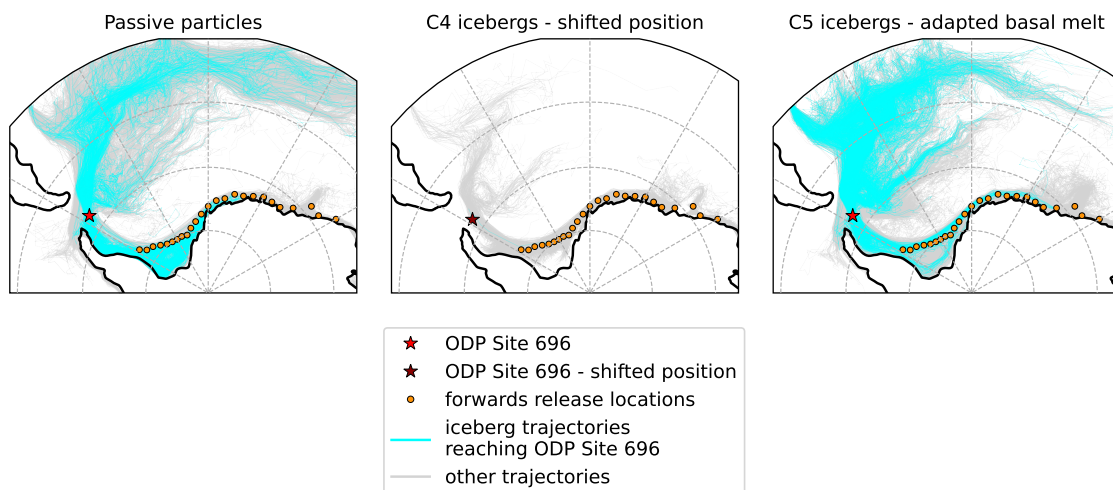


Figure S2: Iceberg trajectories from forward simulations that can (blue) or cannot (grey) reach ODP Site 696 within the distance of one grid cell (~ 11 km) at some point along their trajectory. Shown are the results for passive particles (left), icebergs of size class C4 with a southwestward shift in the position of ODP Site 696 (middle), and icebergs of size class C5 using depth-integration with the adapted basal melt equation (right) (Eq. S7). Note that the marker of ODP Site 696 is roughly half the size of the SOM.

S3.1.2 Potential IRD provenance

Figure S3 and Table S2 show that roughly 1.2 % of the passive particles pass ODP Site 696 within a distance of 11 km. In this case, successful release locations with relatively high numbers compared to most main simulations extend all the way until location 18 in Dronning Maud Land. In addition, particles released at location 21 can reach the SOM, but as for the main simulations, no particles released east of Gunnerus Ridge are successful. Moreover, although the number of successful releases generally decreases with alongshore distance from ODP Site 696, slightly higher numbers are found at locations 11 and 12 (Coats Land).

A southwestward shift in the position of ODP Site 696 for icebergs of size class C4, on the other hand, seems to decrease the number of successful trajectories compared to the main simulation (Fig. S2). In this case, only the fifth release location can release icebergs to ODP Site 696 (Fig. S3). Finally, for the depth-integrated icebergs of C5, the number of successful releases is much higher both in total number and spatial extent especially for the first ten release locations. For this simulation, only locations 19, 20, and 23 to 25 cannot release icebergs to ODP Site 696. Finally, in this case, the number of releases at locations 11 and 12 is slightly lower than at the surrounding release locations.

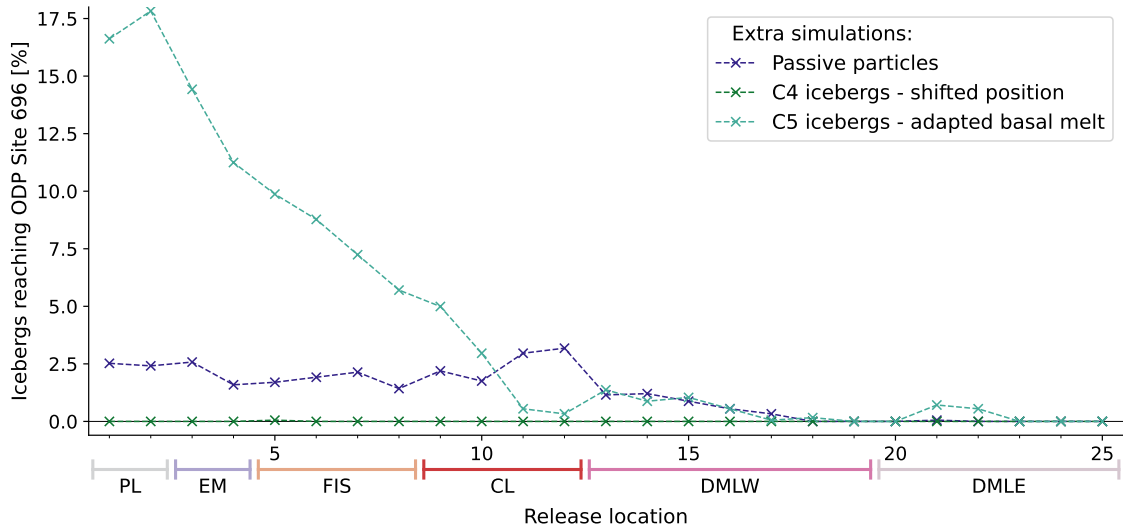


Figure S3: Percentage of total releases per forward release location that reach ODP Site 696 at some point along their trajectory. The coastal regions based on Carter et al. (2017) are indicated on the x -axis where PL = Palmer Land, EM = Ellsworth Mountains, FIS = Filchner Ice Shelf, CL = Coats Land, and DML = Dronning Maud Land (W = west, E = east). The total number and percentages of releases reaching ODP Site 696 per simulation are given in Table S2.

Table S2: Total number and percentage of forward iceberg releases reaching ODP Site 696 within one grid cell distance (~ 11 km) for passive particles, class C4 icebergs for a southwestward shift in the position of ODP Site 696, and class C5 icebergs using depth-integration and the adapted basal melt equation (Eq. S7).

Simulation	Icebergs reaching ODP Site 696	
	#	%
PF	557	1.222
C4 - shifted position	1	0.002
C5 - adapted basal melt	1930	4.235

S3.1.3 Minimum iceberg size

Spatial patterns

Similar to the main simulations, iceberg mass and thickness seem to increase with alongshore distance for both simulations (Fig. S4). In addition, between the two simulations shown here and the normal depth-integrated simulation of size class C1, neither the spatial extent nor the range of magnitudes seems significantly different at first glance. Again, note that icebergs cannot reach the southeastern part of the Filchner Ice Shelf region and a section in front of Coats Land.

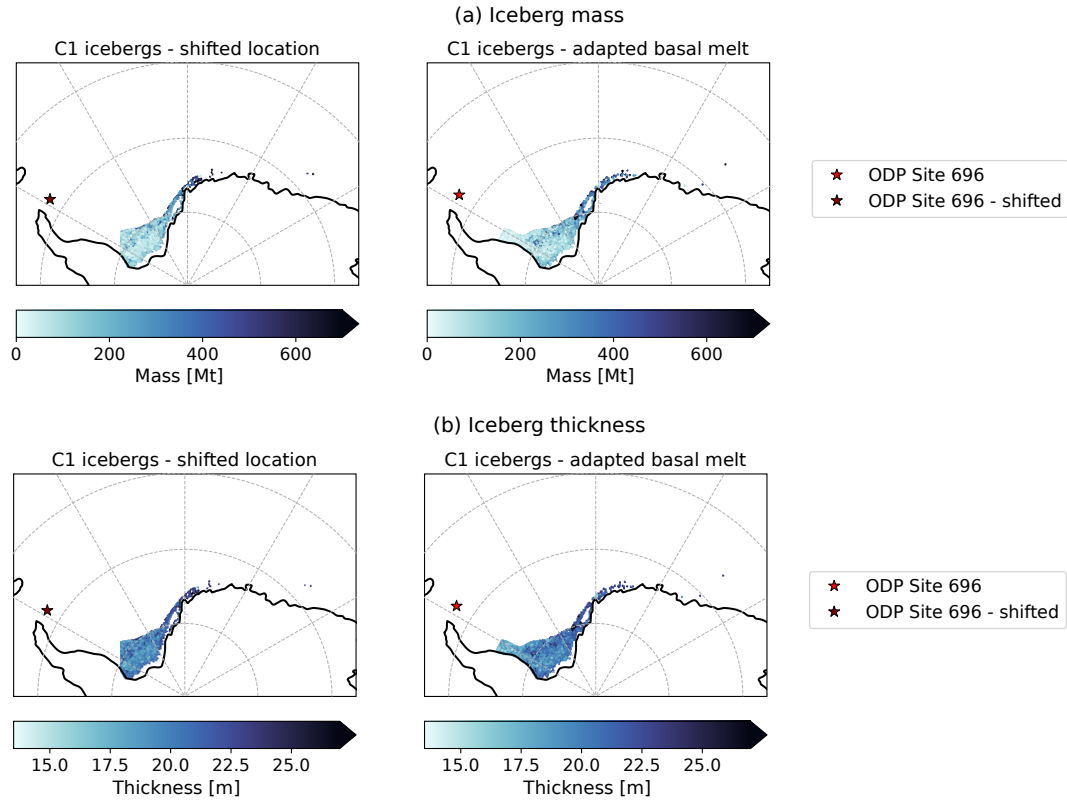


Figure S4: Spatial distribution obtained during backward simulations for (a) iceberg mass and (b) iceberg thickness within the defined coastal region of the Southern Weddell Sea based on Carter et al. (2017). Results are shown for iceberg size class C1 when inducing a southwestward shift in the position of ODP Site 696 (left) or when applying an adapted basal melt equation for $L > 15$ km (right; Eq. S7). The total number of datapoints per simulation are given in Table S3. Note that the marker of ODP Site 696 is roughly a quarter of the size of the SOM.

Regional patterns

Overall, the trends in Figure S5 are similar to those observed from the depth-integrated simulations of the main study. Again, the most probable iceberg size increases with alongshore distance, as does the range of possible values. However, in the eastern sector of Dronning Maud Land, the range of probable values was very large for size class C1. In both simulations here, however, the range remains relatively small and is even smaller than observed in some other regions. Specifically, the most probable iceberg mass varies from roughly 5 Mt in Palmer Land to 450 and 650 Mt in the western and eastern sectors of Dronning Maud Land when shifting the location of ODP Site 696, or from 6 to 400 and 450 Mt when using the adapted basal melt equation. The iceberg thickness ranges from 16.5 or 18 m to roughly 22.5 m for these simulations.

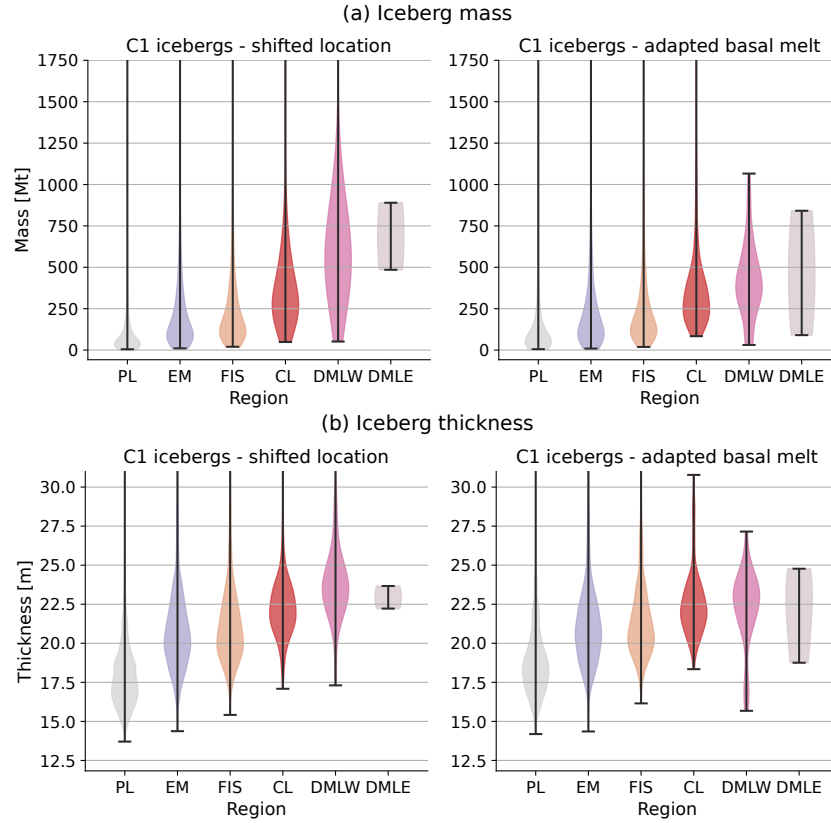


Figure S5: Visualisation (violin plot) of the distribution obtained during backward simulations for the (a) minimum iceberg mass and (b) minimum iceberg thickness in each coastal region where PL = Palmer Land, EM = Ellsworth Mountains, FIS = Filchner Ice Shelf, CL = Coats Land, and DML = Dronning Maud Land (W = west, E = east). Results are shown for iceberg size class C1 when inducing a southwestward shift in the position of ODP Site 696 (left) or when applying an adapted basal melt equation for $L > 15$ km (right; Eq. S7). The number of data points per region are given in Table S3.

S3.1.4 Present-day simulation

In this section, we present the results from the present-day simulation. Contrary to the main simulations, these runs only cover a one-year time period (2021). The time-mean fields are shown in Appendix D of the main text.

Overall, the difference in the temporal resolution used does not seem to significantly alter the iceberg trajectories (Fig. S6a). Note that although the release locations defined for the Eocene were shifted northwards by 5° , part of the locations is still positioned on land or in water depths shallower than the iceberg draft.

The impact of hourly versus monthly wind fields on wave erosion is more substantial (Fig. S6b). The monthly averaged rates of wave erosion over all trajectories are always underestimated using a monthly wind field, leading to deviations between 5 and 41.5 %. On average, wave erosion is underestimated by 30 % by using monthly mean values.

Table S3: Number of data points in each coastal region (Fig. S5; where PL = Palmer Land, EM = Ellsworth Mountains, FIS = Filchner Ice Shelf, CL = Coats Land, and DML = Dronning Maud Land (W = west, E = east)) and in total for the backward simulations of iceberg size class C1 with icebergs starting from a southwestward shifted position of ODP Site 696 (C1s) or with the adapted basal melt equation (C1a; Eq. S7).

Region	Simulation	
	C1s	C1a
PL	5869	5043
EM	3540	3269
FIS	3064	2494
CL	340	350
DMLW	59	65
DMLE	2	3
<i>Total</i>	<i>12874</i>	<i>11224</i>

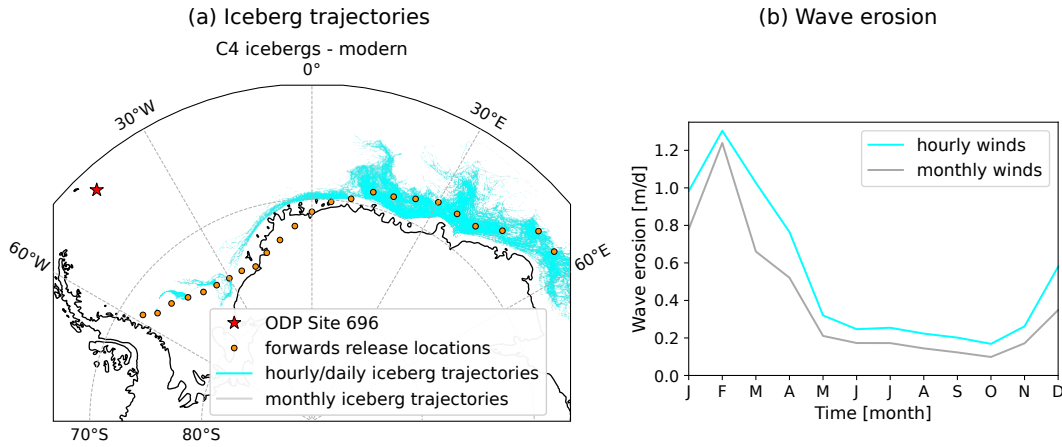


Figure S6: Difference between using present-day hourly wind and daily ocean data (blue) or monthly mean values (grey) for icebergs of size class C4 in a) iceberg trajectories and b) time-mean wave erosion. Note that the marker of ODP Site 696 is roughly a quarter of the size of the SOM.

S3.2 Discussion

S3.2.1 Flow patterns and iceberg trajectories

As the passive particles can only be removed by either leaving the domain (north of 50°S) or reaching the end of the simulation, the length of their trajectories gives an impression of the maximum distance that can be covered within the five-year model period. No particles seem able to circle Antarctica completely within five years. More specifically, a passive simulation has been performed in which passive particles were released in the main current (Fig. S9b). As none of these particles was able to reach the (approximate) initial location (Fig. S9a), we assume icebergs cannot reach ODP Site 696 after circling Antarctica.

S3.2.2 Potential IRD provenance

Although not all simulations shown here include active (iceberg) particles, it is interesting to analyse the differences in their sources. Generally, the passive and surface (main text) simulations show a similar distribution through the regions. As in both cases the particles are advected

using surface velocities, we indeed would expect a similar pattern. However, minor differences between the two arise because of two reasons. Firstly, while the passive particles could move infinitely - as long as they do not leave the modelled domain, the surface particles do experience melt, which might end their trajectories prematurely. Secondly, the passive simulation was run with the more computationally expensive 4th-order Runge-Kutta advection integration method compared to the Euler Forward method used for the surface (and depth-integrated) simulations. Over time, the deviations in estimated locations will add up. Interestingly, the use of depth-integrated velocities for the simulation of icebergs seems to have a positive effect on the number of successful release locations when combined with the adapted basal melt equation.

As was discussed in the main text, releases from Palmer Land (locations 1 and 2) are unlikely sources of IRD due to their young geology. The region offshore the Ellsworth Mountains was previously considered unlikely due to the lack of trajectories reaching ODP Site 696, but seems a reasonable option based on the simulation using the adapted basal melt parameterisation. Releases from offshore the Filchner Ice Shelf are still possible, and, in addition, trajectories from Coats Land now also reach the SOM. Interestingly, while the number of releases from the passive simulation seems increased here, the depth-integrated simulation shows a decrease in numbers. However, as mentioned in the main text, neither Baatsen et al. (2024) nor Van Breedam et al. (2022) suggest glacial conditions in this region, making it an unlikely source location. Finally, a provenance from Dronning Maud Land shows roughly the same likelihood as before.

S3.2.3 Minimum iceberg size

In the previous sections, the region from the Ellsworth Mountains to Dronning Maud Land, with the exclusion of Coats Land, revealed themselves as potential IRD source locations during the adapted basal melt simulation. This gives a mass range of 70 to 400 Mt or 17.5 and 21 m. Although no trajectories from offshore the Ellsworth Mountains could reach ODP Site 696 during the standard forward simulations, the iceberg size in this region is comparable to the values found here. Only in the eastern sector of Dronning Maud Land do the values between the simulations deviate strongly, being more constrained in the current simulations.

S3.2.4 Iceberg melt rates

Although most of the results of the surface simulation are discussed in the main text, it is interesting to study the melt rates of this simulation in more detail as many iceberg models (e.g. Bigg et al., 1997; Gladstone et al., 2001; Martin and Adcroft, 2010) do only include surface fields in their simulations.

The impact of iceberg melt on the iceberg trajectories is already slightly visible by comparing the surface simulation (main text) with the passive simulation, as the number of icebergs leaving the domain north and eastward seems more limited there. Hence, we might conclude that at least part of the icebergs have experienced sufficient melt to disappear before reaching these regions. Indeed, the surface-based melt rates are relatively high compared to the present day (Fig. S7). The difference with the depth-integrated melt rates (Fig. 4.1, main text), however, is relatively minor. Similar trends of increasing melt rates with decreasing latitude are visible. However, the basal melt rate is more constant and very close to zero everywhere. This is due to the small difference in velocity between the layers at the ocean surface used for sampling the basal velocity ($\vec{u}_b = \vec{u}_s$) and the velocity of the iceberg (\vec{u}_{ib}).

In addition, we can study the change in iceberg size through time for the surface-only simulation (Fig. S10). This reveals two main trends in iceberg size changes. In the first case, the mass loss follows a (weak) exponential decrease while iceberg thickness is relatively linear until a change in gradient occurs when small iceberg masses are reached. Secondly, iceberg mass loss occurs faster and more variable, giving rise to a slightly S-like shape. In this case, the iceberg length also shows

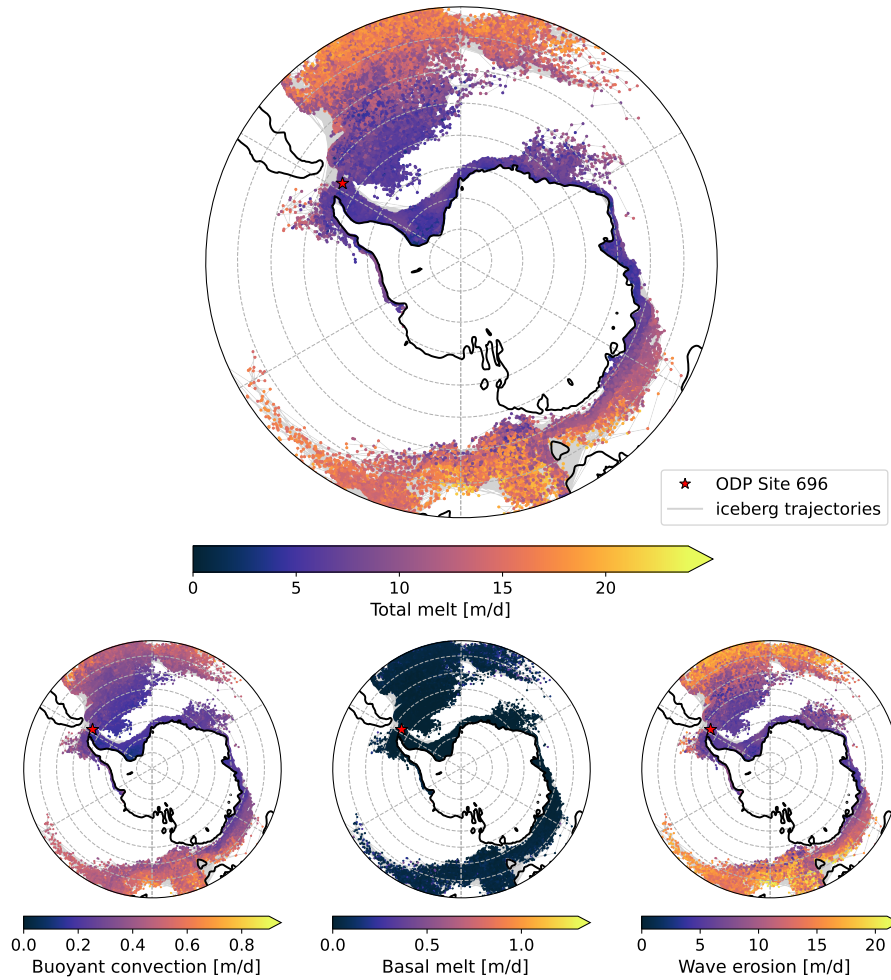


Figure S7: Spatial distribution of the total (top) and individual iceberg melt terms (bottom) along iceberg trajectories for icebergs of size class C4 during a five-year long forward simulation using only surface properties. Note that the marker of ODP Site 696 is roughly half (equal to) the size of the SOM in the top (bottom) panel(s).

a more rapid decline. In both cases, however, changes in iceberg thickness occur very gradually until small iceberg sizes are reached. This is due to the combination of low basal melt rates and iceberg capsizing. In the almost flat part, low basal melt rates (below 0.1 m d^{-1}) cause a very slow decrease in iceberg thickness. At the same time, the larger wave erosion (between roughly 5 and 15 m d^{-1}) and, to a lesser extent, buoyant convection (zero to 0.6 m d^{-1}) cause stronger erosion of the iceberg length and width. Once the iceberg length has decreased to such an extent that its position becomes unstable, the iceberg becomes prone to capsizing. Note that the rate of buoyant convection seems to be seasonally dependent with maxima in Southern Hemispheric summer - which seems reasonable as buoyant convection is strongly dependent on temperature.

A similar approach has been taken for the depth-integrated simulation of size class C4 (Fig. S11). In this case, the iceberg mass decreases roughly exponentially for all icebergs shown. In addition, the iceberg length decreases more or less linearly at a constant rate. Changes in iceberg thickness follow the same pattern as for the surface simulation. Indeed, the basal melt is generally close to zero, explaining the slow decrease in iceberg thickness. In this case, buoyant convection seems more stable around 0.18 m d^{-1} . Finally, wave erosion shows a yearly (seasonal) cyclicity of around 6 m d^{-1} for most icebergs, reaching minimal values in Southern Hemispheric winter. For iceberg 3 in Figure S11 (light green), an animation of its trajectory and mass can be found separately to

the Supplementary Information. This shows that between the first and fifth months of model year 40, the iceberg becomes grounded on the shelf in front of Graham Land. Only at the end of model year 41, it manages to reenter the current system, after which it quickly deteriorates completely. Comparing this interval with Figure S11 shows that buoyant convection and basal melt rates are relatively low and constant here, only the magnitude of wave erosion varies relatively strongly.

S3.2.5 Present-day simulation

At first glance, the iceberg trajectories do not seem very dependent on the resolution of the wind field. This could suggest that in terms of connectivity, the regions found as potential sources in the main study will not change significantly. Since the trajectories are based on the depth-integrated velocities, the iceberg thickness plays a dominant role in determining the iceberg's path. As the thickness is solely dependent on the basal melt rate, which is not influenced by the wind speed, the small deviation in trajectories is not unexpected. However, once the iceberg width and thickness are of similar magnitudes, capsizing can cause a quick decline of the iceberg size by switching the surfaces and, hence, changing the effecting melt term(s) on each iceberg surface. As such, for smaller icebergs, the impact of wave erosion on the trajectories might be larger through iceberg capsizing. In addition, as the modern rates of iceberg melt are much lower than those of the warm Eocene, the Eocene icebergs will change size at a higher rate.

S3.3 Conclusion

Compared to the main simulations, switching from passive particles to depth-integrated icebergs reduces the potential IRD sources, as does shifting the position of ODP Site 696. However, the incorporation of a different melt parameterisation has a significant effect on the iceberg trajectories and origins for size class C5, adding the region offshore the Ellsworth Mountains as a potential source of the IRD at the SOM with minimum size estimates of 100 Mt and 21 m. Unfortunately, as the transition of the heat exchange coefficient between large (three-equation system) and small (bulk equation) icebergs is unknown, the basal melt rate for medium-sized icebergs remains likely underestimated to a certain extent.

S4 Visual material

S4.1 Location ODP Site 696

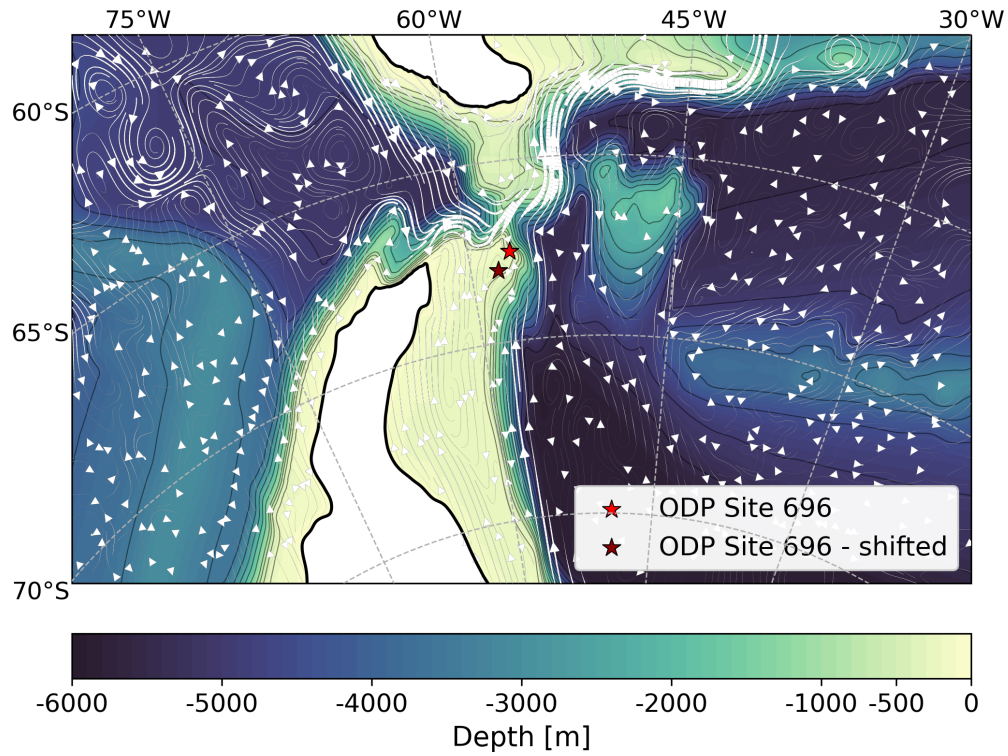


Figure S8: High-resolution Eocene bathymetry and five-year (years 38-42) mean ocean surface currents (Nooteboom et al., 2022) around the approximate late Eocene position of ODP Site 696 used in the main and sensitivity experiments. Note that the markers of ODP Site 696 are less than a quarter of the size of the SOM.

S4.2 Circumpolar simulation

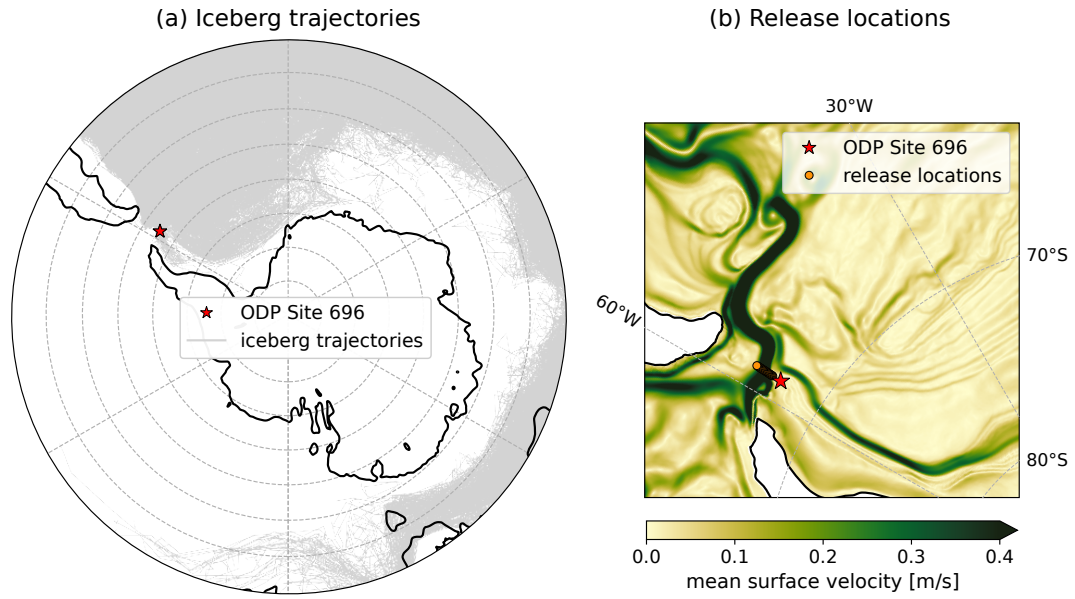


Figure S9: Five-year evolution of (a) particle trajectories and (b) their release locations for passive particles released during the first two model years in the mean current through Drake Passage. Note that the marker of ODP Site 696 is roughly half the size of the SOM in panel (a) or a sixth of the size in panel (b).

S4.3 Iceberg properties and melt rates for selected trajectories

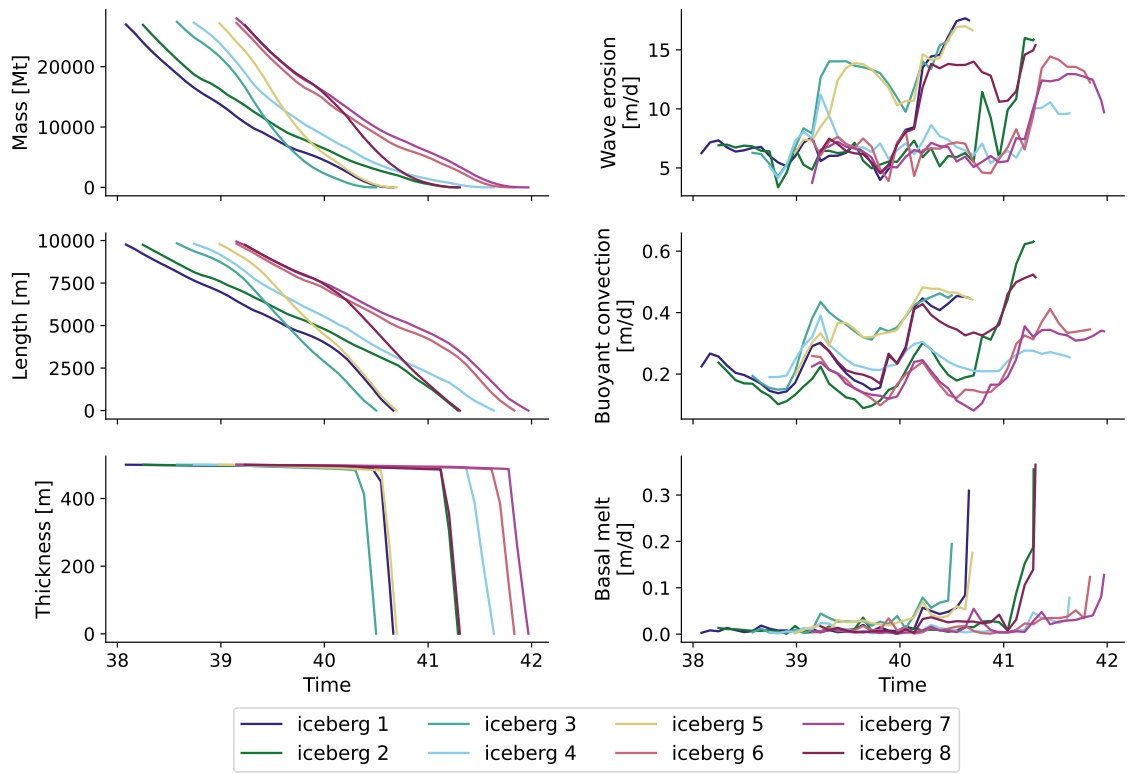


Figure S10: Changes in iceberg size (left) and melt rates (right) for eight selected icebergs of the surface-only forward simulation (size class C4).

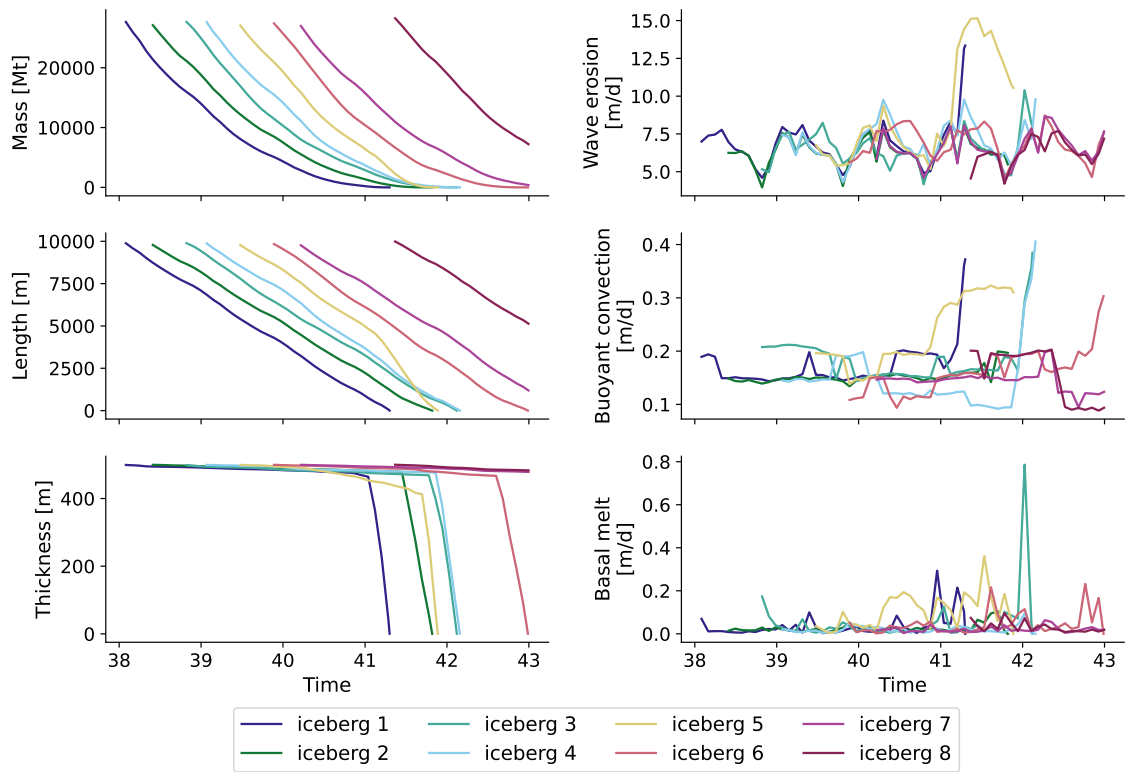


Figure S11: Changes in iceberg size (left) and melt rates (right) for eight selected icebergs of the standard forward depth-integrated simulation (size class C4).

References

- Baatsen, M., Bijl, P., Von Der Heydt, A., Sluijs, A., & Dijkstra, H. (2024). Resilient Antarctic monsoonal climate prevented ice growth during the Eocene. *Climate of the Past*, 20, 77–90. <https://doi.org/10.5194/cp-2023-36>
- Baatsen, M., Van Hinsbergen, D. J., Von Der Heydt, A. S., Dijkstra, H. A., Sluijs, A., Abels, H. A., & Bijl, P. K. (2016). Reconstructing geographical boundary conditions for palaeoclimate modelling during the Cenozoic. *Climate of the Past*, 12(8), 1635–1644. <https://doi.org/10.5194/cp-12-1635-2016>
- Bigg, G. R., Wadley, M. R., Stevens, D. P., & Johnson, J. A. (1997). Modelling the dynamics and thermodynamics of icebergs. *Cold Regions Science and Technology*, 26, 113–135. [https://doi.org/10.1016/S0165-232X\(97\)00012-8](https://doi.org/10.1016/S0165-232X(97)00012-8)
- Carter, A., Riley, T. R., Hillenbrand, C. D., & Rittner, M. (2017). Widespread Antarctic glaciation during the Late Eocene. *Earth and Planetary Science Letters*, 458. <https://doi.org/10.1016/j.epsl.2016.10.045>
- Cenedese, C., & Straneo, F. (2023). Icebergs Melting. *Annual Review of Fluid Mechanics*, 55(1), 377–402. <https://doi.org/10.1146/annurev-fluid-032522-100734>
- Delandmeter, P., & Van Sebille, E. (2019). The Parcels v2.0 Lagrangian framework: New field interpolation schemes. *Geoscientific Model Development*, 12(8), 3571–3584. <https://doi.org/10.5194/GMD-12-3571-2019>
- El-Tahan, M., Venkatesh, S., & El-Tahan, H. (1987). Validation and Quantitative Assessment of the Deterioration Mechanisms of Arctic Icebergs. *Journal of Offshore Mechanics and Arctic Engineering*, 109(1), 102–108. <https://doi.org/10.1115/1.3256983>
- England, M. R., Wagner, T. J., & Eisenman, I. (2020). Modeling the breakup of tabular icebergs. *Science Advances*, 6(51). <https://doi.org/10.1126/sciadv.abd1273>
- Fer, I., & Sundfjord, A. (2007). Observations of upper ocean boundary layer dynamics in the marginal ice zone. *Journal of Geophysical Research*, 112, C04012. <https://doi.org/10.1029/2005JC003428>
- FitzMaurice, A., Cenedese, C., & Straneo, F. (2017). Nonlinear response of iceberg side melting to ocean currents. *Geophysical Research Letters*, 44(11), 5637–5644. <https://doi.org/10.1002/2017GL073585>
- FitzMaurice, A., & Stern, A. (2018). Parameterizing the basal melt of tabular icebergs. *Ocean Modelling*, 130, 66–78. <https://doi.org/10.1016/j.ocemod.2018.08.005>
- Gasparin, F., Greiner, E., Lellouche, J.-M., Legalloudec, O., Garric, G., Drillet, Y., Bourdallé-Badie, R., Traon, P.-Y. L., Rémy, E., & Drévilion, M. (2018). A large-scale view of oceanic variability from 2007 to 2015 in the global high resolution monitoring and forecasting system at Mercator Océan. *Journal of Marine Systems*, 187, 260–276. <https://doi.org/10.1016/j.jmarsys.2018.06.015>
- Gladstone, R. M., Bigg, G. R., & Nicholls, K. W. (2001). Iceberg trajectory modeling and meltwater injection in the Southern Ocean. *Journal of Geophysical Research: Oceans*, 106(C9), 19903–19915. <https://doi.org/10.1029/2000JC000347>
- Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., & Thépaut, J.-N. (2023). ERA5 hourly data on single levels from 1940 to present. <https://doi.org/10.24381/cds.adbb2d47>
- Holland, D. M., & Jenkins, A. (1999). Modeling Thermodynamic Ice–Ocean Interactions at the Base of an Ice Shelf. *Journal of Physical Oceanography*, 29(8), 1787–1800. [https://doi.org/https://doi.org/10.1175/1520-0485\(1999\)029<1787:MTIOLA>2.0.CO;2](https://doi.org/https://doi.org/10.1175/1520-0485(1999)029<1787:MTIOLA>2.0.CO;2)

- Kubat, I., & Sayed, M. (2005). An Operational Model of Iceberg Drift. *International Journal of Offshore and Polar Engineering*, 15(2), 125–131.
- López-Quirós, A., Escutia, C., Etourneau, J., Rodríguez-Tovar, F. J., Roignant, S., Lobo, F. J., Thompson, N., Bijl, P. K., Bohoyo, F., Salzmänn, U., Evangelinos, D., Salabarnada, A., Hoem, F. S., & Sicre, M. A. (2021). Eocene-Oligocene paleoenvironmental changes in the South Orkney Microcontinent (Antarctica) linked to the opening of Powell Basin. *Global and Planetary Change*, 204. <https://doi.org/10.1016/j.gloplacha.2021.103581>
- Martin, T., & Adcroft, A. (2010). Parameterizing the fresh-water flux from land ice to ocean with interactive icebergs in a coupled climate model. *Ocean Modelling*, 34(3-4), 111–124. <https://doi.org/10.1016/j.ocemod.2010.05.001>
- Merino, N., Le Sommer, J., Durand, G., Jourdain, N. C., Madec, G., Mathiot, P., & Tournadre, J. (2016). Antarctic icebergs melt over the Southern Ocean: Climatology and impact on sea ice. *Ocean Modelling*, 104, 99–110. <https://doi.org/10.1016/j.ocemod.2016.05.001>
- Nooteboom, P. D., Bijl, P. K., Van Sebille, E., Sluijs, A., Baatsen, M., Kliphuis, M. A., Dijkstra, H. A., & Von Der Heydt, A. S. (2022). Improved Model-Data Agreement With Strongly Eddying Ocean Simulations in the Middle-Late Eocene. *Paleoceanography and Paleoclimatology*, 37. <https://doi.org/10.1002/essoar.10508749.1>
- Rackow, T., Wesche, C., Timmermann, R., Hellmer, H. H., Juricke, S., & Jung, T. (2017). A simulation of small to giant Antarctic iceberg evolution: Differential impact on climatology estimates. *Journal of Geophysical Research: Oceans*, 122(4), 3170–3190. <https://doi.org/10.1002/2016JC012513>
- Tournadre, J., Bouhier, N., Girard-Ardhuin, F., & Rémy, F. (2015). Large icebergs characteristics from altimeter waveforms analysis. *Journal of Geophysical Research: Oceans*, 120(3), 1954–1974. <https://doi.org/10.1002/2014JC010502>
- Tournadre, J., Bouhier, N., Girard-Ardhuin, F., & Rémy, F. (2016). Antarctic icebergs distributions 1992–2014 [_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/2015JC011178>]. *Journal of Geophysical Research: Oceans*, 121(1), 327–349. <https://doi.org/10.1002/2015JC011178>
- Van Breedam, J., Huybrechts, P., & Crucifix, M. (2022). Modelling evidence for late Eocene Antarctic glaciations. *Earth and Planetary Science Letters*, 586. <https://doi.org/10.1016/j.epsl.2022.117532>
- van de Lagemaat, S. H. A., Swart, M. L. A., Vaes, B., Kusters, M. E., Boschman, L. M., Burton-Johnson, A., Bijl, P. K., Spakman, W., & van Hinsbergen, D. J. J. (2021). Subduction initiation in the Scotia Sea region and opening of the Drake Passage: When and why? *Earth-Science Reviews*, 215, 103551. <https://doi.org/10.1016/j.earscirev.2021.103551>
- Wagner, T. J. W., Dell, R. W., & Eisenman, I. (2017a). An Analytical Model of Iceberg Drift. *American Meteorological Society*, 47, 1605–1616. <https://doi.org/10.1175/JPO-D-16-0262.s1>
- Wagner, T. J. W., Stern, A. A., Dell, R. W., & Eisenman, I. (2017b). On the representation of capping in iceberg models. *Ocean Modelling*, 117, 88–96. <https://doi.org/10.1016/j.ocemod.2017.07.003>
- Weeks, W., & Campbell, W. (1973). Icebergs as a fresh water source: An appraisal. *Journal of Glaciology*, 12(65), 207–33.