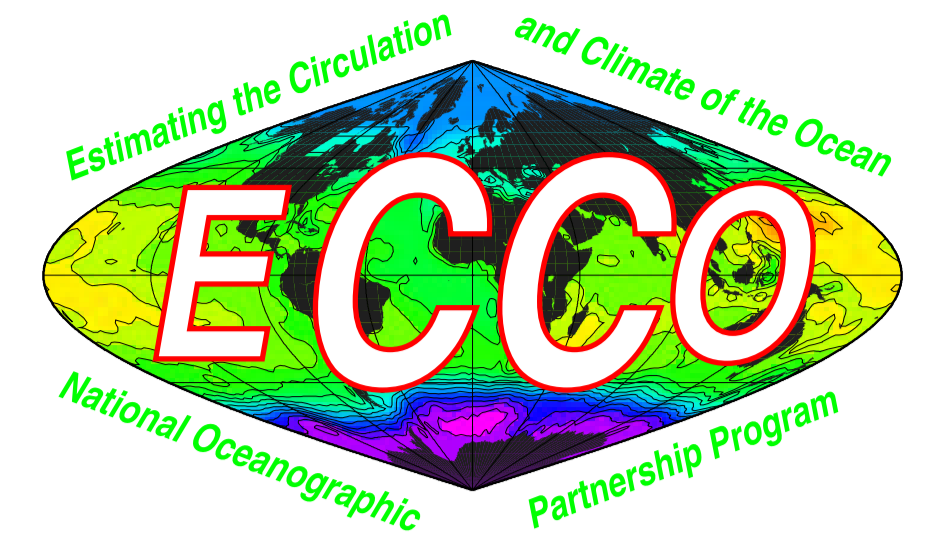




An efficient exact adjoint of the parallel MIT general circulation model, generated via automatic differentiation



Patrick Heimbach¹, Chris Hill¹ and Ralf Giering²

¹ MIT / EAPS, Cambridge (MA), USA
² FastOpt, Hamburg, Germany

E-mail: {heimbach, hill}@mit.edu
 E-mail: ralf.giering@fastopt.de

URL: http://mitgcm.org
 URL: http://www.fastopt.de

1 Introduction

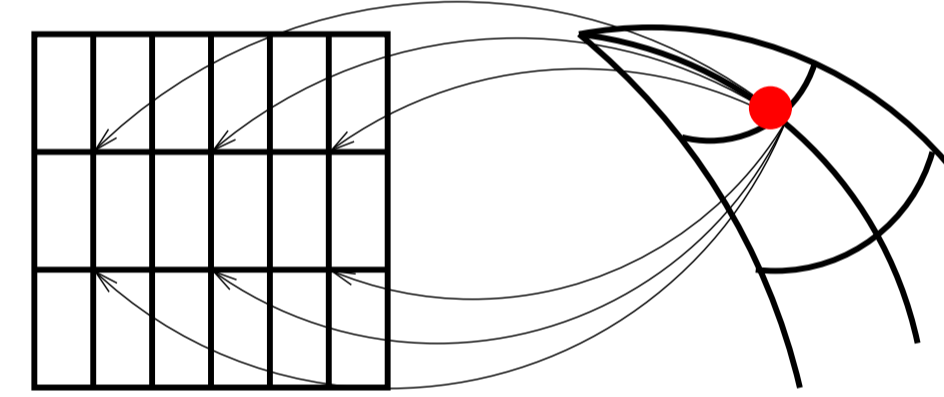
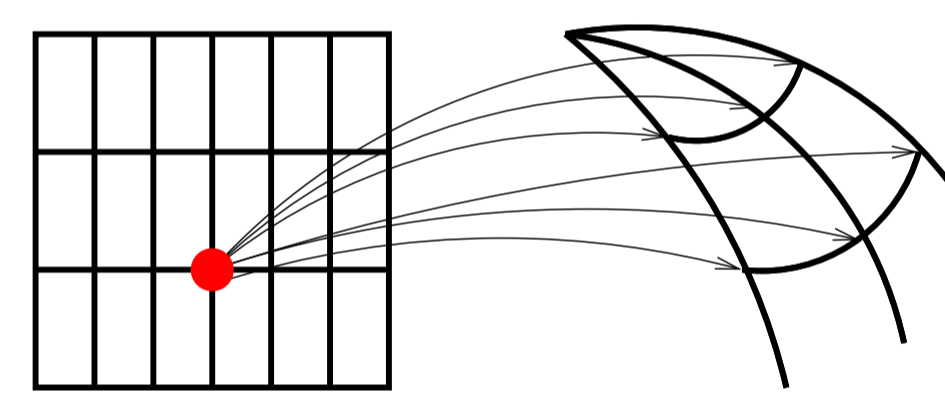
(I) Ocean State Estimation [1, 2]

- Given:
 - a set of (possibly different types of) observations
 - a numerical model & set of initial / boundary conditions
- Question: (estimation / optimal control problem)
 Find optimal trajectory consistent with observations within prior error
- Approach:
 Minimize least square function $\mathcal{J}(\vec{u})$ measuring model vs. data misfit
 → seek $\vec{\nabla}_u \mathcal{J}(\vec{u})$ to infer update $\Delta \vec{u}$ from variation of controls \vec{u}

$$\vec{u}^{n+1} = \vec{u}^n + \Delta \vec{u}$$

(II) Sensitivity analysis [3]

- Finite difference approach:
 - Take a “guessed” anomaly (e.g. SST(x, y, t)) and determine its impact on some model output (e.g. MOC)
 - Perturb each input element (SST(i, j)) to determine its impact on output (MOC).
- Reverse/adjoint approach:
 - Calculates “full” sensitivity field
 - Approach: $\mathcal{J} = \text{MOC}$, $\vec{u} = \text{SST}$
 → $\vec{\nabla}_u \mathcal{J}(\vec{u}) = \frac{\partial \text{MOC}}{\partial \text{SST}(x, y, t)}$

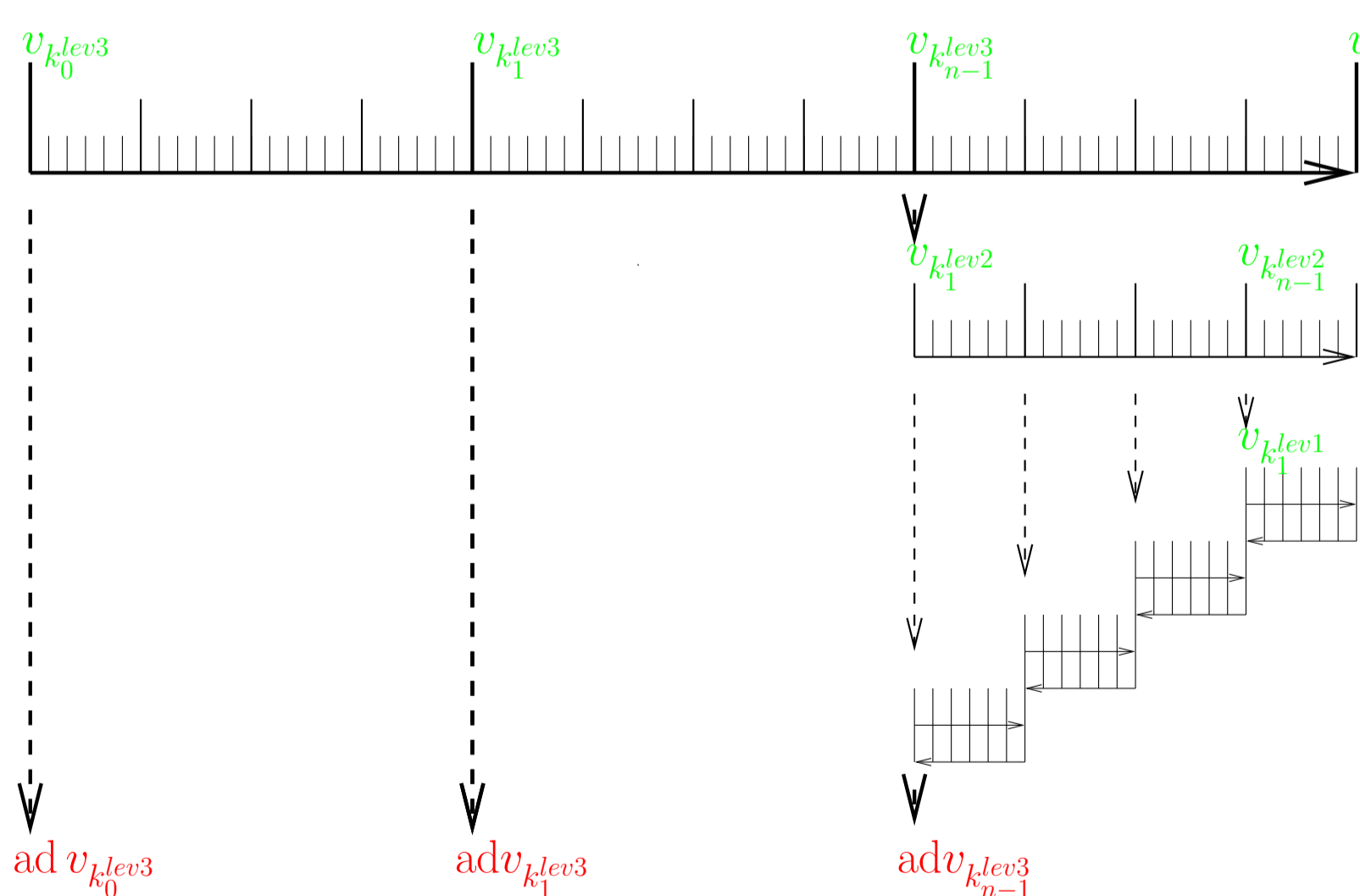


finite difference / forward approach

adjoint / reverse approach

5 Flow reversal and Checkpointing

- Adjoint = transpose of TLM
 → evaluated in reverse order
 → all state stored or recomputed
- Solution: Checkpointing
 e.g. Griewank, 1992
 → balances storing vs. recomputation



e.g. 3-level checkpointing:

$$n_T = n_1 \cdot n_2 \cdot n_3$$

→ Storing: reduced from

$$n_1 \cdot n_2 \cdot n_3 \text{ to } \begin{cases} n_2 + n_3 & \text{to disk} \\ n_1 & \text{to memory} \end{cases}$$

→ CPU: 3 · forward + 1 · adjoint ≈ 5.5 · forward

→ adjoint dump & restart closely related!

2 The Adjoint Method

Consider least-square model vs. data misfit cost function:

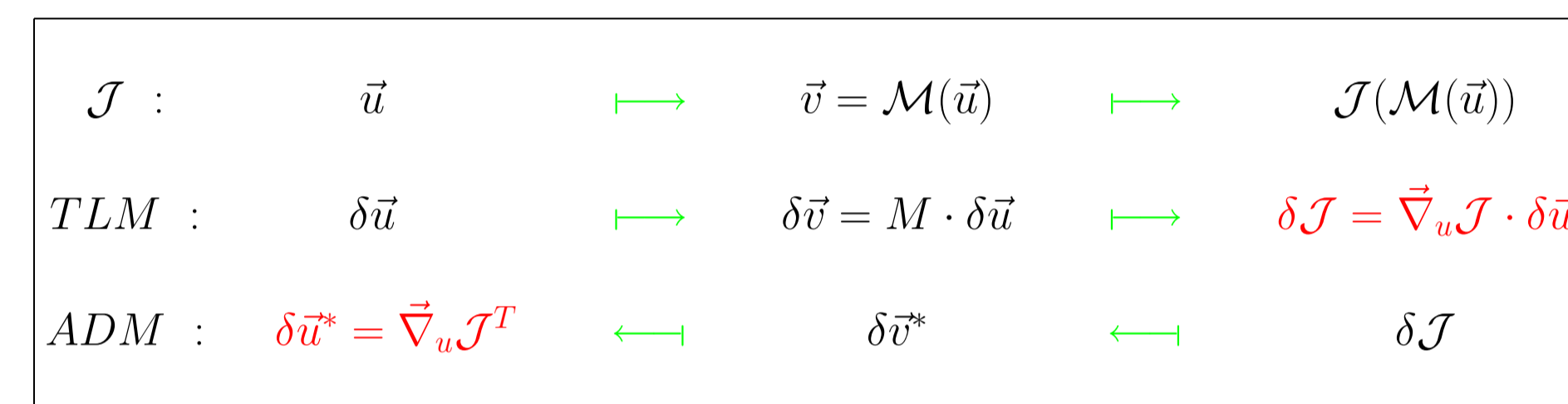
$$\mathcal{J}(\mathcal{M}(\vec{u})) = (\mathcal{H}(\vec{v}) - \vec{d})^T W^{obs} (\mathcal{H}(\vec{v}) - \vec{d}) + (\vec{u} - \vec{u}^{(0)})^T W^u (\vec{u} - \vec{u}^{(0)})$$

- $\vec{v} = \mathcal{M}(\vec{u})$ nonlinear model
- M, M^T tangent linear (TLM) / adjoint (ADM)
- $\delta \vec{u}, \delta \vec{u}^*$ perturbation / dual (or sensitivity)
- $\mathcal{H}, \mathcal{H}^T$ data projection operator / adjoint thereof
- W^{obs}, W^u inverse error weights (metric)

Need gradient of $\mathcal{J} \in \mathbb{R}^1$ w.r.t. control variables $\vec{u} \in \mathbb{R}^m$

$$\min_{\vec{u}} \mathcal{J}(\vec{u}) \Rightarrow \vec{\nabla}_c \mathcal{J}(\vec{u}, \mathcal{M}(\vec{u})) \Big|_{\vec{u}=\vec{u}^{opt}} = 0$$

Use the adjoint method to compute gradient



$$\vec{\nabla}_u \mathcal{J}^T \Big|_{\vec{u}} = 2 M^T \cdot H^T \cdot W \cdot (\mathcal{H}(\vec{v}) - \vec{d}) \sim (\text{adjoint model}) \cdot (\text{model} - \text{data})$$

TLM : m (~ $n_x n_y n_z$) integrations @ 1 · (#forward)
 ADM : 1 integration @ γ · (#forward)

3 Automatic Differentiation (AD)

$$\vec{v} = \mathcal{M}_\lambda (\mathcal{M}_{\lambda-1} (\dots (\mathcal{M}_0 (\vec{u}))) \quad \delta \vec{u}^* = M_0^T \cdot M_1^T \cdot \dots \cdot M_\lambda^T \cdot \delta \vec{v}^*$$

Automatic differentiation:

each line of code is elementary operator \mathcal{M}_λ
 → rules for differentiating elementary operations
 → yield elementary Jacobians M_λ
 → composition of M_λ 's according to chain rule
 yield full tangent linear / adjoint model

TAMC/TAF source-to-source tool [4]



AD and tangent linearity

Consider the nonlinear function

$$f(x, y, z) = 3x + axy^2 + bz$$

Its differential will be of the form

$$df = (3 + a y^2) dx + (2a x y) dy + b dz$$

→ state at $(x = x_0, y = y_0)$ has to be known to evaluate df .

→ evaluation of derivative is tangent
 at each point along nonlinear trajectory

8 Conclusions:

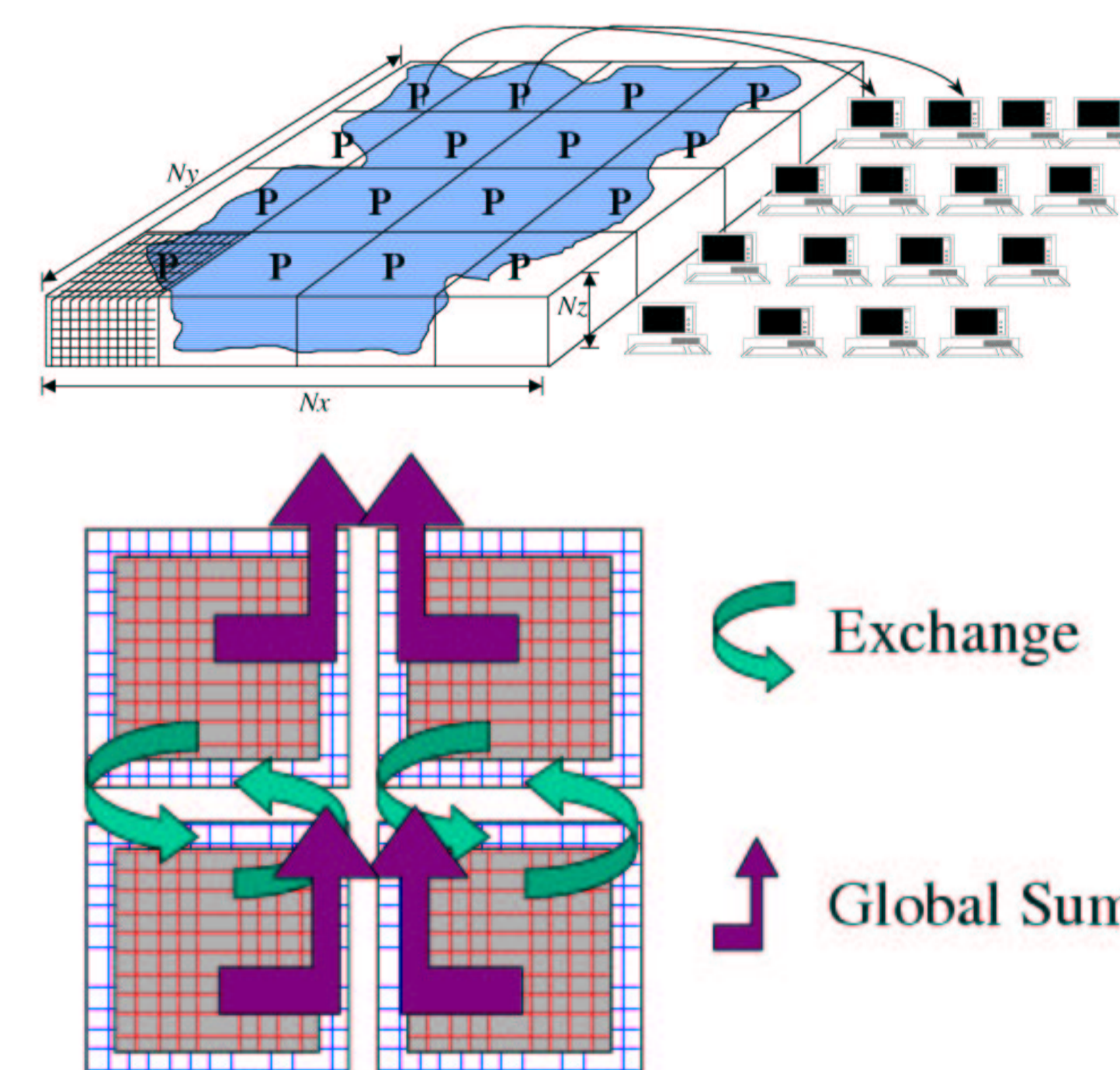
- Scalable adjoint code generation is feasible for fully-fledged models [5, 6]
- AD tools indispensable for ongoing code development
- Nevertheless, challenges remain despite AD:
 Efficient adjoint code generation is at best semi-automatic so far!
- Code development should have AD “in mind”
- Libraries/couplers need derivative forms (e.g. the ESMF, PRISMA model coupling projects)
- AD tools need to be improved

ACTS: Adjoint Compiler Technology & Standard

AD tools ought to be
 – open-source,
 – easier to use,
 – transparent,
 – common platform that facilitates AD tool development by many

6 Adjoint Code Scalability

- domain decomposition (tiles) & overlaps (halos)
- split into extensive on-processor and global phase



Global communication/arithmetic op.'s supported by MITgcm's intermediate layer (WRAPPER) which need hand-written adjoint forms

operation/primitive	forward	reverse
• communication (MPI,...):	send	receive
• arithmetic (global sum,...):	gather	scatter
• active parallel I/O:	read	write

7 Example: Boundary layer limits in KPP

Implement following criteria:

- for neutral stratification: H_{bl} should be smaller than both h_e and L
- generally: H_{bl} should be larger than some minimal value

Model code:

```
CADJ STORE hbl, bfc TO TAPE
do i = 1, Nx * Ny
  if (bfc(i).gt.0.0) then
    hekman = cekman * ustar(i) / max(abs(Coriol(i)), eps)
    hmonob = cmonob * ustar(i) * ustar(i) * ustar(i)
    / vonk / bfc(i)
    hlimit =
    & stable(i) * min(hekman, hmonob)
    + (stable(i) - 1) * zgrid(Nr)
    & hbl(i) = min(hbl(i), hlimit)
  end if
  hbl(i) = max(hbl(i), minKPPhbl)
end do
```

$$h_e = 0.7u' / f$$

$$L = u'^2 / (\kappa B_T)$$

h_u limit for:

– stable case

– unstable case

apply upper limit

apply lower limit

Adjoint code:

```
CADJ RESTORE hbl, bfc FROM TAPE
do i = 1, Nx * Ny
  adhbl(i) = adhbl(i) * (0.5 + sign(0.540, hbl, 2(i) - minKPPhbl))
  if (bfc(i).gt.0.0) then
    c ---> recompute hekman, hmonob, hlimit <-----
    adhlimit = adhlimit + adhbl(i) * (0.5 - sign(0.540, hlimit - hbl, 1(i)))
    adhbl(i) = adhbl(i) * (0.5 + sign(0.540, hlimit - hbl, 1(i)))
    adhekman = adhekman + adhlimit * stable(i) * (0.5 + sign(0.540, hmonob - hekman))
    adhmonob = adhmonob + adhlimit * stable(i) * (0.5 - sign(0.540, hmonob - hekman))
    adstable(i) = adstable(i) + adhlimit * (zgrid(nr) + min(hekman, hmonob))
    adhlimit = 0.40
  ...
end if
end do
```

hbl, 2 NOT available

hbl, 1 available

hbl, 1 available

References

- Stammer, D., Wunsch, C., Giering, R., Eckert, C., Heimbach, P., Marotzke, J., Adcroft, A., Hill, C., Marshall, J.: The global ocean circulation and transports during 1992 – 1997, estimated from ocean observations and a general circulation model. *J. Geophys. Res.* **107(C9)** (2002a) 3118
- Stammer, D., Wunsch, C., Giering, R., Eckert, C., Heimbach, P., Marotzke, J., Adcroft, A., Hill, C., Marshall, J.: Volume, heat and freshwater transports of the global ocean circulation 1993 – 2000, estimated from a general circulation model constrained by WOCE data. *J. Geophys. Res.* (2002b) in press.
- Marotzke, J., Giering, R., Zhang, K., Stammer, D., Hill, C., Lee, T.: Construction of the adjoint MIT ocean general circulation model and application to Atlantic heat transport variability. *J. Geophys. Res.* **104, C12** (1999) 29,529–29,547
- Giering, R., Kaminski, T.: Recipes for adjoint code construction. *ACM Transactions on Mathematical Software* **24** (1998) 437–474
- Heimbach, P., Hill, C., Giering, R.: Automatic generation of efficient adjoint code for a parallel navier-stokes solver. In J.J. Dongarra, P.M.A. Sloot and C.J.K. Tan, ed.: *Computational Science – ICCS 2002*. Volume 2331 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin (Germany) (2002a) 1019–1028
- Heimbach, P., Hill, C., Giering, R.: An efficient exact adjoint of the parallel mit general circulation model, generated via automatic differentiation. *Future Generation Computer Systems (FGCS)* (2002b) submitted.

Acknowledgements

This is a contribution to the ECCO project, supported by NOPP, and with funding from NASA, NSF and ONR. Many members of the ECCO Consortium under the lead of D. Stammer have contributed.