Algorithm

**Initialization Step**

1) Check that:

• *D* is connected

• *g*(*x*) is continuous and differentiable everywhere in *D*

• *h*(*x*) = *ln*(*g*(*x*)) is concave everywhere in *D*

2) Initialize T\_k (vector with *k* elements):

• *Tk* = *{xi*;*i* = 1*, ..., k}*, where *x*1 *≤ ... ≤ xk* are the *k* abscissae in *D* where we will evaluate *h*(*x*) and *h0*(*x*)

• If *D* unbounded on the left, chose *x*1 s.t. *h0*(*x*1) *>* 0

• If *D* unbounded on the right, chose *xk* s.t. *h0*(*xk*) *<* 0

3) Evaluate *h*(*x*) and *h0*(*x*) on *Tk* and store these as two-length *k* vectors, say h\_x and h\_prime\_x 4) Calculate z (vector with *k* + 1 elements):

• *z*0 = lower bound of *D* (or *−∞* if *D* is not bounded below)

• For *j* = 1*, ..., k −* 1, *zj* =*h*(*xj*+1)*−h*(*xj* )*−xj*+1*h0*(*xj*+1)+*xjh0*(*xj* )

*h~~0~~*(*xj* )*−h~~0~~*(*xj*+1)(these are the points at which the

tangents to *h*(*x*) at *xj* and *xj*+1 intersect)

• *zk* = upper bound of *D* (or +*∞* if *D* is not bounded above)

5) Find u\_k (this is the piecewise linear upper hull formed by tangents to *h*(*x*) at *Tk*):

• *uk*(*x*) = *h*(*xj* ) + (*x − xj* )*h0*(*xj* ), where *x ∈* [*zj−*1*, zj* ] and *j* = 1*, ..., k*

• NB: *exp*(*uk*(*x*)) is the rejection envelope on *Tk*

6) Find s\_k:

R

• *sk*(*x*) = *exp*(*uk*(*x*))

*Dexp*(*uk*(*x0*))*dx0*

7) Find l\_k (this is the piecewise linear lower hull formed by connecting adjacent points on *h*(*x*) where *Tk* is evaluated)

• *lk*(*x*) = (*xj*+1*−x*)*h*(*xj* )+(*x−xj* )*h*(*xj*+1)

*xj*+1*−xj*, where *x ∈* [*xj , xj*+1] and *j* = 1*, ..., k −* 1

• For *x < x*1 or *x > xk*, define *lk*(*x*) = *−∞*

• NB: *exp*(*lk*(*x*)) is the squeezing function on *Tk*

**Sampling Step**

1) Sample a value *x∗*from *sk*(*x*)

2) Sample a value *w* independently from *Unif*(0*,* 1)

3) Perform the test:

• If *w ≤ exp{lk*(*x∗*) *− uk*(*x∗*)*}*:

**–** Accept *x∗*

• Else:

1

**–** Evaluate *h*(*x∗*) and *h0*(*x∗*)

**–** If *w ≤ h*(*x∗*) *− uk*(*x∗*)*}*:

∗ Accept *x∗*

**–** Else:

∗ Reject *x∗*

**Updating Step**

1) Follow this recipe:

• If *h*(*x∗*) and *h0*(*x∗*) were evaluated in Sampling Step:

**–** Include *x∗*in *Tk* to form *Tk*+1

**–** Relabel the *xi in* *Tk* in ascending order

**–** Construct new functions *uk*+1(*x*), *sk*+1(*x*), and *lk*+1(*x*)

**–** Increment *k*

**–** Return to Sampling Step if *n* points have not been sampled yet

• Else:

**–** No update necessary, repeat Sampling Step if *n* points have not been sampled yet 2