Sensitivity functions and uncertainty analysis

Ian G. Enting

MASCOS

The University of Melbourne

Acknowledgments

- The Center of Excellence for Mathematics and Statistics of Complex Systems (MASCOS) is funded by the Australian Research Council (ARC).
- My fellowship at MASCOS is supported by CSIRO through a sponsorship agreement.
- Collaborators: Cathy Trudinger and YingPing Wang of CSIRO Marine and Atmospheric Research and members of the MATCH working group on the Brazilian Proposal.

Summary

- Analysing the Brazilian proposal
  - Marginal attribution
  - Sensitivity functions
  - Use in attribution, sensitivity and data uncertainty
- Interpretation as adjoints
- derivation by automatic differentiation.

Brazilian Proposal

Tabled by Brazil during negotiations leading to Kyoto Protocol — Flicked-passed to Subsidiary Body for Scientific and Technical Advice (SBSTA).

Proposes that emission reduction targets should be proportional to nation's relative responsibility for the greenhouse effect.

Issues:

- Indicator? What quantity is used as a measure of the greenhouse effect?
- For what period of emissions is responsibility attributed?
- How are non-linear responses attributed?

Marginal attribution

As example, use indicator $T^* = T_{\mathrm{CO2}}(2100)=$ warming in 2100 from CO_2 emissions.

T^* is to be attributed to emissions $E_j(t)$ from country j with $E(t) = \sum_j E_j(t)$.

Differential attribution of warming to emissions from country j is

$$\frac{\partial}{\partial \alpha_j} T^*[\sum_j (1 + \alpha_j) E_j(t)]$$

Sensitivity functions

Consider additive form: $$\frac{\partial}{\partial\gamma} T^*[E(t) + \gamma f(t)]$$

This is linear: $$\frac{\partial}{\partial\gamma} T^*[E(t) + \gamma(f_1(t) + f_2(t))]$$

$$= \frac{\partial}{\partial\gamma} T^*[E(t) + \gamma f_1(t)] + \frac{\partial}{\partial\gamma} T^*[E(t) + \gamma f_1(t)]$$

Therefore can be represented as inner product:

$$\frac{\partial}{\partial\gamma} T^*[E(t) + \gamma f(t)] = \int_0^{\mathcal{T}} S(t)\, f(t)\, dt$$

Sensitivity function, $S(t)$ is Frechet derivative.

Applications

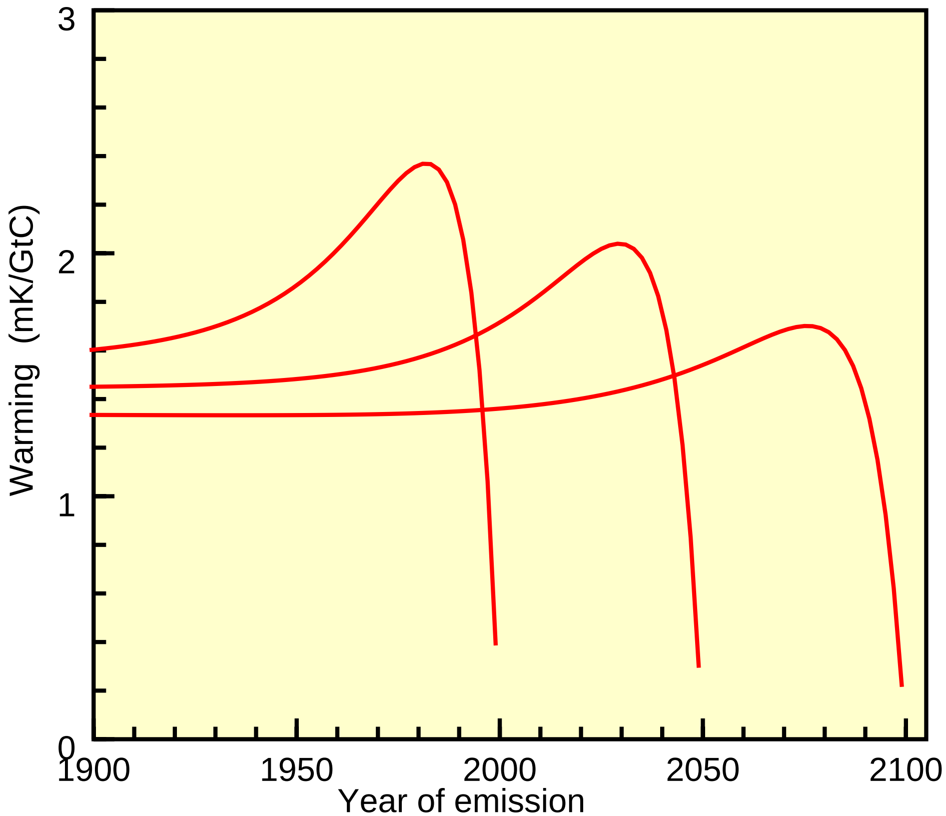
- Attribution
$$\frac{\partial}{\partial \alpha_j} T^*[\sum_j (1+\alpha_j) E_j(t)] = \int S(t) E_j(t)\, dt$$

- Cumulated attribution: $T_j^* = \sum_j \int S(t) E_j(t)\, dt$

- Sensitivities:
$$\frac{\partial}{\partial \beta} T_j^* = \sum_j \int E_j(t) \frac{\partial}{\partial \beta} S(t)\, dt$$

- Sensitivity of T_j^* to uncertainties in emissions can be obtained as
$$\mathrm{Var}[T_j^*] = \int \int S(t) \mathrm{Cov}[E_j(t), E_j(t')] S(t')\, dt'\, dt$$

Results: Frechet Derivatives



Assumes IS92a emissions. Represents temperature by response function. Linear responses for ocean and biotic carbon, coupled non-linearly to atmospheric CO_2 (as in CSIRO study).

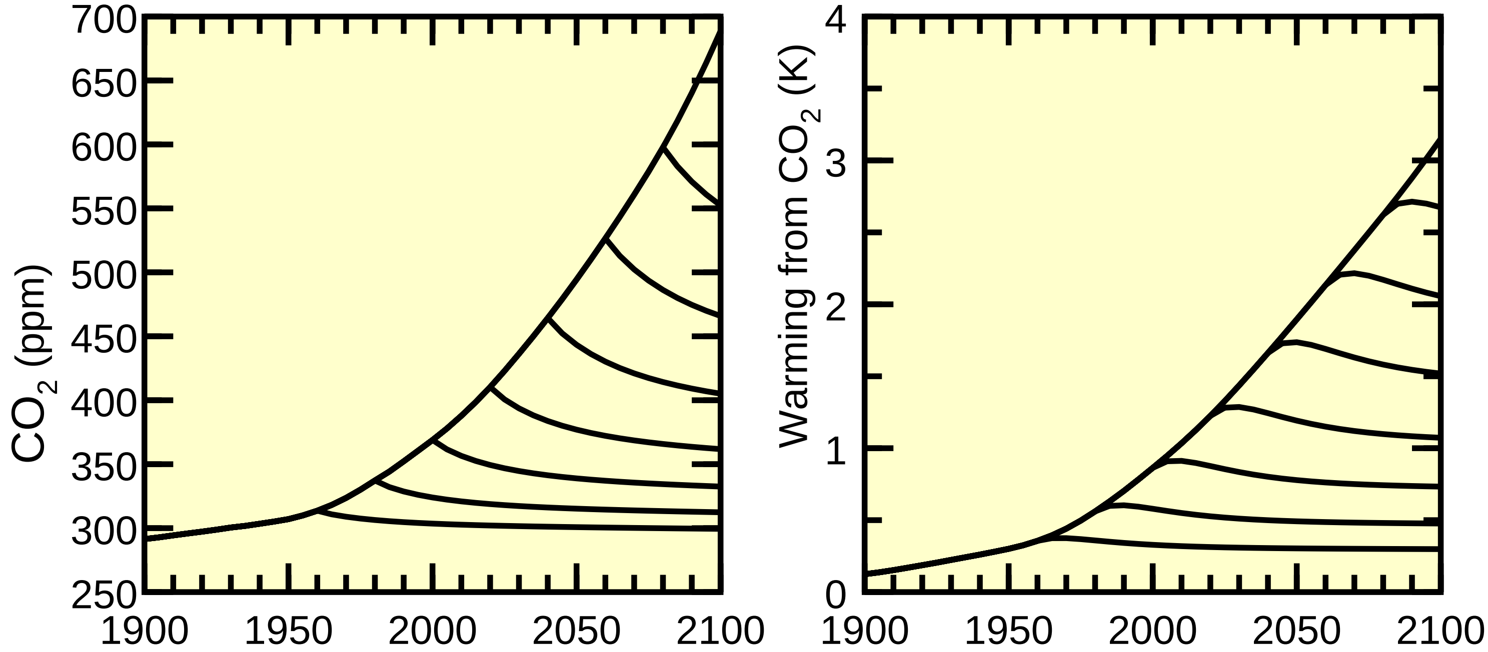
"$\frac{\partial}{\partial E(t)} T(\tau)$" for $\tau = 2000, 2050, 2100$.

Decrease as $t \to \tau$ shows 'committed warming'.
At any time, warming from the most recent releases is yet to happen.

In preparation for MATCH workshop

Timescales

CO_2 concentrations and consequent warming, partitioned according to time of emission.



Lowest bands are from pre-1960 emissions, next from 1960 to 1980 emissions, etc. Increase in contribution to warming after time of emissions from 'committed warming' effect.

In preparation for MATCH workshop

Tangent Linear Model (TLM)

For N DEs: $\frac{d}{dt}x_j = g_j(\{x_k\}, \alpha, t)$ for $j = 1, N$

we can define sensitivities as

$$y_{j,p} = \frac{\partial}{\partial \alpha_p} x_j \quad \text{for } j = 1, N \text{ or}$$

to give 'tangent linear model(s)':

$$\frac{d}{dt} y_{m,p} = \frac{\partial}{\partial \alpha_p} g_m(\{x_k\}, \alpha, t) + \sum_n \frac{\partial}{\partial x_n} g_m(\{x_k\}, \alpha, t)\, y_{n,p}$$

Adjoint relations can give gradients.

Algorithmic Differentiation (AD)

Differentiation by successive use of chain rule.

For binary operation $c = f(a, b)$,

$$\frac{\partial c}{\partial \alpha} = \frac{\partial f}{\partial a} * \frac{\partial a}{\partial \alpha} + \frac{\partial f}{\partial b} * \frac{\partial b}{\partial \alpha}$$

e.g.

$$c = a + b \quad \rightarrow \frac{\partial c}{\partial \alpha} = \frac{\partial a}{\partial \alpha} + \frac{\partial b}{\partial \alpha}$$

$$c = a * b \quad \rightarrow \frac{\partial c}{\partial \alpha} = b * \frac{\partial a}{\partial \alpha} + a * \frac{\partial b}{\partial \alpha}$$

Convert program to code for derivatives, one operation at a time.

Operator Overloading (C++)

Replace real variable x, (type `double`), with composite variable $\tilde{x}$ (type `Xvar`), representing both value x and its derivatives with respect to K model quantities, α_k as:

$$\tilde{x}_0 = x \quad \text{and} \quad \tilde{x}_k = \frac{\partial}{\partial \alpha_k} x \quad \text{for } k = 1, K$$

Operator overloading implements $\tilde{c} = \tilde{a} * \tilde{b}$, representing:

$$\tilde{c}_0 = \tilde{a}_0 * \tilde{b}_0 \quad \text{and} \quad \tilde{c}_k = \tilde{a}_0 * \tilde{b}_k + \tilde{a}_k * \tilde{b}_0$$

Overloaded functions, $\tilde{c} = f(\tilde{a})$, represent:

$$\tilde{c}_0 = f(\tilde{a}_0) \quad \text{and} \quad \tilde{c}_k = f'(\tilde{a}_0) * \tilde{a}_k$$

where $f'(.)$ denotes the derivative of $f(.)$

Class Definitions

Fragment of C++ class definition to implement operator overloading:

```
class Xvar{
public :
static const int ns = _NUMDERIVS+1;
double xs[_NUMDERIVS+1];
Xvar operator*(Xvar);
...
};

Xvar Xvar::operator*(Xvar b){ Xvar c;
for (int i=1; i < ns; i++)
   c.xs[i] = xs[i]*b.xs[0]+xs[0]*b.xs[i];
c.xs[0] = xs[0]*b.xs[0];
return c;} ;
...
```

Usage

Original

```
double F_co2(double c){
double a;
a = log(c/280.0)*5.35;
return a;
};
...
double cc;
...
ff = F_CO2(cc)
```

Transformed

```
Xvar F_co2(Xvar c){
Xvar a;
a = log(c/280.0)*5.35;
return a;
};
...
Xvar cc;
// Derivatives wrt
// initial value of cc
cc.set(280,1);
...
ff = F_CO2(cc)
```

Putting AD into models

Need to modify model by changing:

- type declarations
- output
- initialisation (and input)
- other surprises ????

Use syntax checking of compiler to help ensure validity.

Real = *x_var* should be undefined, detection by compiler implies failure to declare all neccessary variables.

Basic Operator Requirements

A basic set:

Binary

Op	X.op.X	R.op.X	X.op.R	X.op.I	I.op.X
=	Y	N/A	Y	Y	N/A
+	Y	Y	Y	Y	Y
−	Y	Y	Y	Y	Y
*	Y	Y	Y	Y	Y
/	Y	Y	Y	Y	Y
**	−	−	−	Y	−

Unary operations (and intrinsics)

Op	−	sqrt	cos	sin	log	exp
	Y	Y	Y	Y	Y	Y

Conclusions

Brazilian Proposal —

- For a given indicator, T^*, calculation of $S(t)$ allows attribution to any nation.
- $S(t)$ most efficiently calculated from adjoint model, but for multiple indicator times, tangent linear model is not too inefficient.
- Sensitivity of T_j^* to model uncertainties can be obtained as second derivatives.

Algorithmic differentiation —

Operator overloading is a straightforward way of developing tangent linear models (and obtaining higher derivatives if needed).

Further Information

Andreas Griewank, 2000, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, (SIAM, Philadelphia).

MATCH website (Brazilian Proposal): *http://www/match-info.net*

I.G. Enting, 2005, *Automatic differentiation in the analysis of strategies for mitigation of global change*, International Congress on Modelling and Simulation, Melbourne, 2005. Ed. A. Zerger and R. M. Argent, 7pp
http://www.mssanz.org.au/modsim05/papers/enting.pdf