

シミュレーションの可視化と対話性

株式会社環境総合研究所 大西 行雄

コンピュータシミュレーション(環境モデル)は、環境を構成する諸因子についてのきわめて複雑な関係を取り扱う。しかし、人間の頭の処理能力を越えた、言い換えれば、ブラックボックス化した計算結果が合意形成上の説得力を持つとは思えない。複雑な関係性を人間が納得できる形にわかりやすく再構成してはじめて説得力がえられる。可視化、対話性の重要性がそこにある。

シミュレーションとはなにか

現実の環境課題に対して、厳密な意味での科学的な解答を出すことはほとんどの場合に、不可能である。環境に関して数多くの要因が複雑に関連している。言い換えれば、全体システムは多数のサブシステムによって構成されているが、それぞれのサブシステムもまた複雑な構成要素をもっており、それらの要因の関連についての数量的関係は明確にはなっていないものが大部分である。

このような状況であるとしても、特定のサブシステムに関して、その動的な構造をコンピュータ上の数学モデルとして表現し、その振る舞いを調べることが可能な場合があり、そのモデルから現状理解や政策のヒントを得られるケースがある。このような場合にコンピュータシミュレーションが使われる。

「津波」を例にすれば、地震発生後に A 地点に津波が到達するのに 2 時間という時間がかかった理由、津波の高さが 10m であった理由はなにかという疑問に答えるのが「現状理解」である。そこでコンピュータシミュレーションが実施される。シミュレーションの結果が、実際の到達時間、実際の波高に十分に近いものであれば、現状が理解されたと認識される。

シミュレーションモデルによって、現状(あるいは、過去の現象)が再現(説明)できるならば、異なった条件下ではどのような現象が起きるのかを予測することが可能になる。

環境に関するモデリングの歴史をふりかえると、1970 年代に、湖沼や内湾の富栄養化が問題になりはじめるとともに、プランクトンの生態動力学などを対象にした大規模かつ複雑な内容のコンピュータシミュレーションが行われるようにな

った。水環境だけではなく、大気環境に関しても大規模なモデルが作成され、おもだった手法は 1970 年代にほぼ出揃ったといえる。その後、コンピュータの処理能力の向上に伴って、より現実的な条件をとりこんだモデルが試みられるようになり、1980 年代後半には、環境モデルは現象理解と問題解決のための強力な道具のひとつとして認知されるようになった。それと同時に、環境モデルの課題も明らかになってきた。

(1) 環境モデルが複雑なものになるほど、コンピュータでしか処理できない大量のデータが扱われるが、モデルに含まれる関係が多くなりすぎた場合には、コンピュータ内のプロセスを人間が頭の中で追跡理解できないという事態を引き起こす。モデルに組み込むべき諸関係のうち、より本質的なものを取り込み、瑣末な関係は排除して単純化すべきであるが、実際にそのようにモデルが構築されているかどうかを判断するのは人間である。それが困難なほどに複雑なモデルは、逆に合意形成上の説得力をもたなくなる。

(2) 環境モデルが現状を再現したとしても、既存のモデリング手法では、大規模変化によるシステムの構造変化には追従しえない。モデルを使った将来予測の際に、モデルの適用限界を超えた大規模変化を扱うことはしてはならない。モデルのプロセスが複雑すぎる場合には、将来予測がモデルの適用範囲で行われているのは、それを超えているのかの判断が困難になる。そのような場合にも、モデルの説得力は失われる。

可視化

環境モデルの課題のひとつは、コンピュータ内のプロセスをなんとか人間が理解できる形にすることである。実際のモデルでは、インプットデータ、アウトプットデータともにきわめて多量である。これらを数字で印字したのを見ても意味の解釈は困難である。そこで、これらの多量の数字を図に表現することによって解釈可能な形にすることを可視化という。

モデルへのインプットデータは、環境モデルの場合、地理的な分布をもつデータである。そのため、現在では、インプットデータを地理情報システム(GIS)と連動したデータベースに格納することが多い。GIS を使ってインプットデータを整理しておく、必要なときにはいつでも、(データによって色塗りわけをした)地図の形でデータを確認することができる。インプットデータの処理に関して、通常の GIS では実現できない以下のようなプロセスがある。

(1) 環境モニタリングデータ

水環境や大気汚染などのモニタリングデータは、特定のモニタリング地点において離散的に取得されており、面的な分布を直接的に知ることはできない。このような離散的なデータから 2 次元的、あるいは 3 次元的に連続なデータを推定する手法としては、数学的な内挿法や、それ自身、環境モデルのひとつである「診断型モデリング」手法などがある。

(2) 異なる集計単位へのデータの読み替え

人間の社会活動が引き起こす環境問題、典型的には、大気汚染や水質汚濁などの場合、社会活動の規模の大きさや質は、市町村などという社会的単位で集計されたデータを基礎としてあつかう。一方、環境モデルでは、これらのデータは、モデルがあつかう湖、あるいは、地域、それらを分割したグリッドのように地形的な単位に再集計される。この再集計のプロセスは、コンピュータの中で一瞬に行われるのであるが、逆に、ブラックボックス化しやすいといえる。

たとえば、政策目標が汚染濃度の 5%削減であり、環境モデルが、「汚濁負荷量を 10%減らせば、汚染濃度を 5%減らせる」という結果を出しているような場合を考えると、汚濁負荷量 10%削減を、社会的集計単位の上にもどのように割り戻せばいいかということが問題になる。このような場

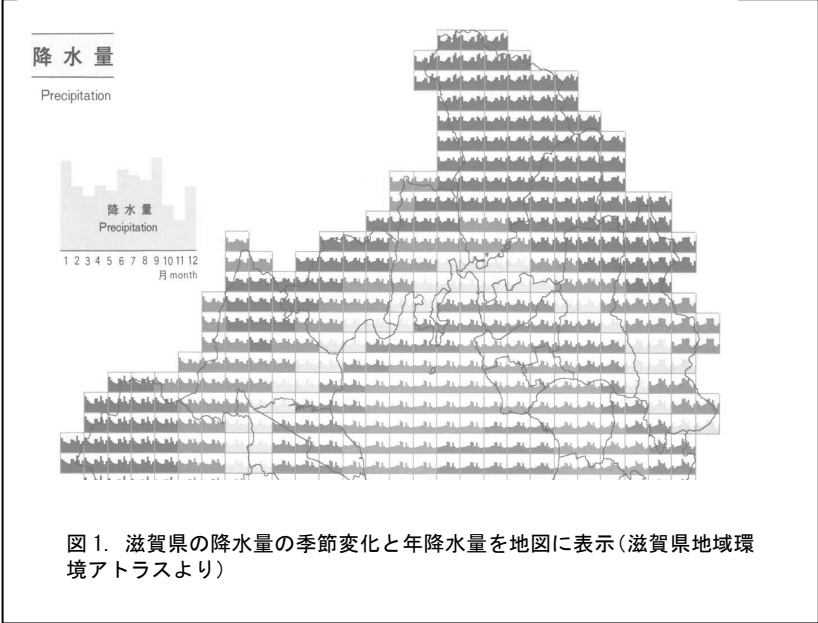


図 1. 滋賀県の降水量の季節変化と年降水量を地図に表示 (滋賀県地域環境アトラスより)

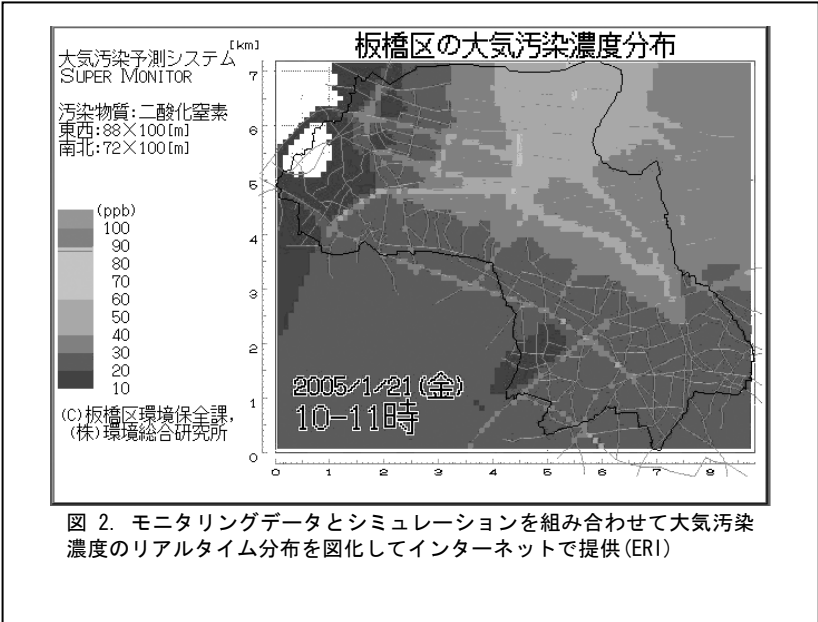


図 2. モニタリングデータとシミュレーションを組み合わせる大気汚染濃度のリアルタイム分布を図化してインターネットで提供 (ERI)



図 3. 河川流域別の土地利用 (棒グラフの幅が流域面積を表わし、それぞれ土地利用状態別に帯グラフになっている) 滋賀県地域環境アトラスより

合に、集計単位の読み替えプロセスをわかりやすく可視化することが重要になる。

環境モデルのアウトプットの可視化は、途中プロセスの可視化と、結果の可視化にわけて考える。初期(1980年代半ばまで?)の環境モデルでは、モデルの計算の途中経過を図示しながら同時に計算を続けるということができないシステムは無い、あるいは稀であった。そのため、計算結果を磁気記憶装置に蓄積しておいて、あとで結果を図示するのが普通であった。現在では、モデルの計算の途中経過をディスプレイ画面上に作図してそれを見ながら処理を進めることは容易に実現できる。オフスクリーンキャンバス(メモリー上の仮想のキャンバス)の上に図示したものを、常時、ディスプレイ画面に表示しながら、必要に応じて、プリンタに出力したり、あるいは、画像ファイルとして保存してインターネットなどを経由して公開することも可能である。

結果の可視化とは、政策目標としての計算結果(たとえば、湖の5年後の年間平均 COD 濃度)が、政策手段(たとえば流域における下水道普及率目標)とどのような関係にあるかを示すような図示である。

途中経過の可視化は、モデルに対する信頼性を得るためのものであり、結果の可視化は、モデルを適用するために必要なものである。

対話性

対話型処理とは、環境モデルへの入力データが変化したときに、アウトプットがどう変化するかを見るために、入力データの変化を画面上で設定し、結果を図示して見せるような処理のことを言う。この意味での「対話型処理」とは、入力データを入力ファイルとして作成し、計算の途中経過は示されず、見えないところで計算が終わるとその結果は出力ファイルに保存されているという「一括(バッチ)処理」の反対の概念である。

環境モデルにおける対話性については、このようなバッチ処理の対概念を超えるものが必要である。大部分の環境モデルにおいて、すでに述べたように入力データは集計

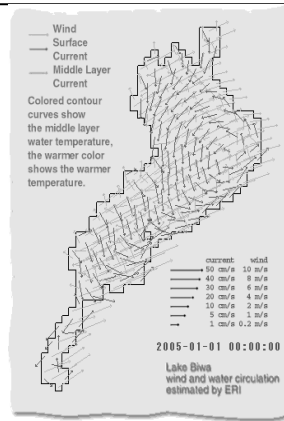


図4. 琵琶湖の風によって作られる流れ。琵琶湖周辺11箇所の風向・風速の観測データを用いて毎日自動的にシミュレーションを行いインターネットで更新公開される。

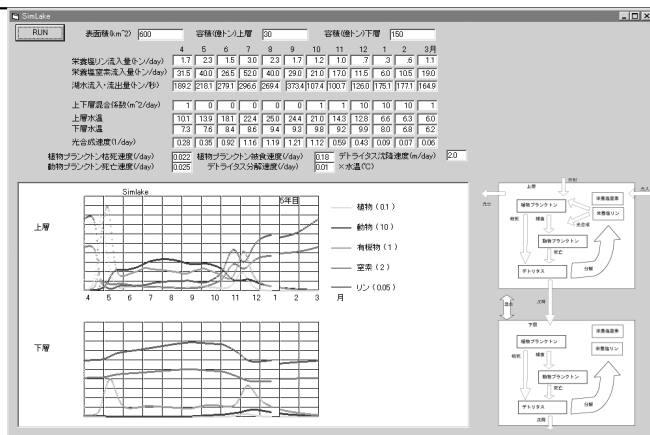


図5. 湖沼富栄養化モデル(1ボックスの一番シンプルなモデル) 栄養塩負荷量と、水の流出入量、上下層の混合係数などによって富栄養化の程度がどのように変化するのかパソコン画面で簡単にシミュレーションできる。

単位の読み替えなど複雑なプロセスを経た後にシミュレーションモデルの入力データになる。そのような中間段階の結果を見ながら、処理の継続や入力条件の変更などを行うような形の対話型処理が必要である。

一方、すでに述べたように現在適用されている多くの環境モデルでは、構造変化を伴うような大きな変化を追従できない。このことを言い換えれば、入力条件の微小変化に対応するモデルのアウトプットの微小変化については、モデルの信頼性はあるとしても、大きな変化を扱うことには慎重でなければならない。さらに言い換えれば、これはシステムの線形応答特性こそ重要であるということと同じである。そうである以上、入力条件のさまざまな変化に対応してむやみに多くのケースの計算結果を出すのではなく、システムの線形応答特性を明らかにして、その上で入力→出力系の応答特性を調べるというような対話型処理が必要とされる。