大学・大学院での教育（有効だったこと）

- 大学（1991～1995年）
  - FORTRANプログラミング講座（大学内部の情報基盤センター）
  - C言語，C＋＋言語，オブジェクト指向的プログラミングは自習した．
- 大学院（1995～2001年）
  - 基礎物理学研究所で開催の並列プログラミング講座（2日くらい）および共同利用計算機の利用
    - はじめて並列計算用ライブラリ（MPI，PVM）を利用した並列プログラミングについて学んだ．
      - よかったこと：並列計算が，とても難しいというものではない印象を受けて，実際に使ってみようというやる気が起きた．
    - 上記の講習会の後，基研のマシンを利用(共同利用)させていただいた．
      - よかったこと：利用の条件が緩やかであったため，計算機の規模は小さいが，ほとんど好きなだけ使うことができた．失敗もしながら実際の利用の仕方を学んでいくことができた．
      - 初心者にとっては，自由に利用できる小～中規模のシステムがあると学習しやすい．
  - 共同利用可能な計算機を利用した研究や技術習得
    - 筑波大学(CP－PACS)，原研，KEK，大阪大学(RCNP)などを利用させていただいた．ネットワークを使って作業ができたため，名古屋にいながら効率的に研究と技術習得を進めることができた．

大学・大学院での教育の不足点・問題点

- 不足点・問題点
  - ソフトウェア工学の知識，パフォーマンス・チューニングの考え方がほとんど得られていなかった．
    - 信頼性の高いソフトウェアをつくるためにはどうしたらいいのか．
    - たとえば，MPI並列計算はパフォーマンス・チューニングの一部分に過ぎない．
  - 情報技術という観点でみると，物理という分野だけでは狭い．（すべてではないが）情報技術のなかに，物理，化学，生物などの分野の知識も集約されてくる．
    - 計算基礎科学のエキスパートを目指す人が物理だけに専念するということは考え難い．
  - (その他)計算基礎科学の大事な役割と思うこと
    - 確立された知を集約する．→産業界で応用し，開発競争力を高める．
    - 確立されていない知の正しさを確かめる．→基礎科学を発展させる．
  - (その他)計算基礎科学が，さまざまな分野の知を統合するように発展するとしても，個々の科学分野の基礎研究（新現象，新法則）はやはり重要と思われる．

企業に入ってから役に立ったか

- 役に立ったこと
  - 企業に入ってからもほとんどそのまま使えること
    - 数学（数値計算，物理(応用)数学）と計算機関係技術（ハードウェア＋ソフトウェア工学）
    - 物理的考え方（保存則，対称性）．
    - 十分に確立された（実績・信頼性のある）物理法則等(力学，流体力学，古典電磁気学，基礎的な量子力学・統計力学)．ただし，最先端・極限的な科学は直ぐには使われない．
  - 企業に入ってから考え方等の変更を要すると思ったこと
    - 具体的な対象を，十分高精度に計算することが容易でないことも多い．
      - 計算機の計算能力の限界．
      - 経験的あるいは粗視化された物理モデルの導入も必要．
      - 初期条件を完全に設定することが難しい．
    - 計算基礎科学の方法を上手く活用する，柔軟な考え方．
    - 計算結果を出すことは企業の最終目標ではないので，計算基礎科学だけでなく，より広い視野で業務を進めるのがいい．
    - 基本法則が十分に分かって正確にシミュレーションができるようになっても，入力(初期)条件には無限の可能性がある．様々な入力条件とその結果を求めることを通じて，自然法則を，道具として思いのままに利用できることが理想である．

提案等

- 計算機科学技術では，①モデリング，②数学，③計算機，の3つの技術のバランスが重要.
  - モデリング技術が欠けると:
    - 必要のない自由度まで計算すると，計算量が非現実的になることがある.
      - 最適な物理モデルを選び出せる技術が望まれる（Lattice QCD ⇔ ハドロン有効模型 ⇔ 原子核模型）．課題を解決するために，どのような物理量を導入するべきか．一般に自由度を増やすと理論はシンプルになるが，自由度が増大して計算が追いつかなくなる傾向がある.
    - モデルの適用限界を何も考えずに利用して，意味のない結果（数値）を信じてしまうのは問題.
      - あるモデルの適用限界を，より高精度のモデルや理論から理解する必要もある.
  - 数学（Numerical Mathematics）の技術が欠けると:
    - 必要以上の計算量を生じさせてしまうことがある．例えば，フーリエ変換におけるFFT(Fast Fourier Transform)や，多粒子系の古典的クーロン力計算におけるFMM（Fast Multipole Method），あるいは，対称性を考慮した計算量の削減.
    - 線形演算などで，正しく得られていない解を誤って使ってしまう，など.
    - (その他)数学は早めにできるだけしっかりやっておくべき．企業に入ってからは，十分に時間をかけて学習する機会が少ない．物理モデルが変わっても，解法に同じ数学的手法を使うことは多い（線形演算等）.

提案等

- 計算機(ハードウェア＋ソフトウェア工学)の技術が不足すると:
  - 信頼性の高いソフトウェアがつくれない(入力条件を変えると簡単に異常終了する など).
    - 信頼性の高いソフトウェアをつくるには，やはりソフトウェア工学の知識・経験も必要.
    - ドキュメントを書くための技術の教育も重要.
    - 保守性を考慮した分かりやすいコーディング(特別な理由がなければ技巧を見せない).
    - バグに対する客観的で確実な対応.
    - 使いやすいインタフェースの開発は重要．アプリケーション開発者がアプリケーション開発の重要な部分に専念できるようにする.
  - OS，コンパイラの知識がまったくないと，計算の高速化を十分に行えないことがある．また，並列化率，ベクトル化率，キャッシュヒット率をただ高くすればいいというものではない.
    - 最終的には，計算時間の短縮が課題である.
  - (その他)国産CPUを主とするハードウェアマニュアルの不足は，優れたOS，アプリ開発に影響があると思われる.