Numerical Simulation of Dynamic Systems XIII

Prof. Dr. François E. Cellier
Department of Computer Science
ETH Zurich

April 9, 2013

PDEs in Multiple Space Dimensions

In principle, the MOL methodology can be extended without modification to the case of PDEs in multiple space dimensions. For example, the two-dimensional heat flow problem:

$$\frac{\partial u}{\partial t} = \sigma \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

discretized using third-order accurate finite difference formulae for both the discretization in the x- and in the y-directions leads to the following ODE at point $x = x_i$ and $y = y_j$:

$$\frac{du_{i,j}}{dt} \approx \sigma \left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\delta y^2} \right)$$

PDEs in Multiple Space Dimensions

In principle, the MOL methodology can be extended without modification to the case of PDEs in multiple space dimensions. For example, the two-dimensional heat flow problem:

$$\frac{\partial u}{\partial t} = \sigma \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

discretized using third-order accurate finite difference formulae for both the discretization in the x- and in the y-directions leads to the following ODE at point $x = x_i$ and $y = y_j$:

$$\frac{du_{i,j}}{dt} \approx \sigma \left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\delta y^2} \right)$$

Yet, the problems are formidable. The first, and most frightening, problem is concerned with the sheer numbers of resulting ODEs.

PDEs in Multiple Space Dimensions

In principle, the MOL methodology can be extended without modification to the case of PDEs in multiple space dimensions. For example, the two-dimensional heat flow problem:

$$\frac{\partial u}{\partial t} = \sigma \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

discretized using third-order accurate finite difference formulae for both the discretization in the x- and in the y-directions leads to the following ODE at point $x = x_i$ and $y = y_j$:

$$\frac{du_{i,j}}{dt} \approx \sigma \left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\delta y^2} \right)$$

Yet, the problems are formidable. The first, and most frightening, problem is concerned with the sheer numbers of resulting ODEs.

Let us assume that we use 50 segments in each space dimension. Then, the *2D problem* has $50 \times 50 = 2500$ ODEs, whereas the *3D problem* has $50 \times 50 \times 50 = 125,000$ ODEs. The **A**-matrix of the 3D problem has $125,000 \times 125,000 = 15,624,000,000$ elements.

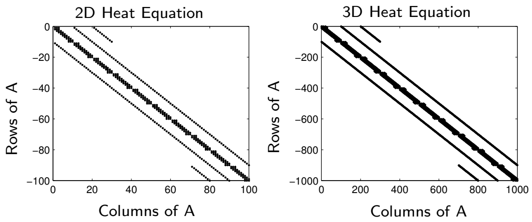
PDEs in Multiple Space Dimensions II

The second problem has to do with the distribution of the non-zero elements in the **A**-matrix. Until now, it always happened that the **A**-matrix of a single linear PDE converted by use of finite differences was *band-structured* with a narrow band width. There exist special matrix routines for very efficient handling of band-structured matrices.

PDEs in Multiple Space Dimensions II

The second problem has to do with the distribution of the non-zero elements in the **A**-matrix. Until now, it always happened that the **A**-matrix of a single linear PDE converted by use of finite differences was *band-structured* with a narrow band width. There exist special matrix routines for very efficient handling of band-structured matrices.

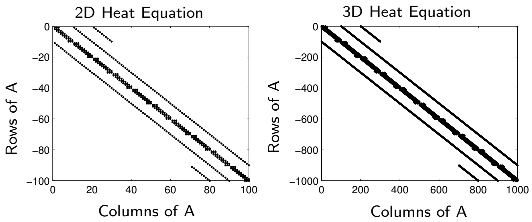
The **A**-matrices of 2D and 3D problems are still band-structured, but the bandwidth is no longer as narrow.



PDEs in Multiple Space Dimensions II

The second problem has to do with the distribution of the non-zero elements in the **A**-matrix. Until now, it always happened that the **A**-matrix of a single linear PDE converted by use of finite differences was *band-structured* with a narrow band width. There exist special matrix routines for very efficient handling of band-structured matrices.

The **A**-matrices of 2D and 3D problems are still band-structured, but the bandwidth is no longer as narrow.



Let n be the number of segments. In the 1D case, the bandwidth was constant. In the 2D case, it grows proportional in n. In the 3D case, it grows proportional in n^2.

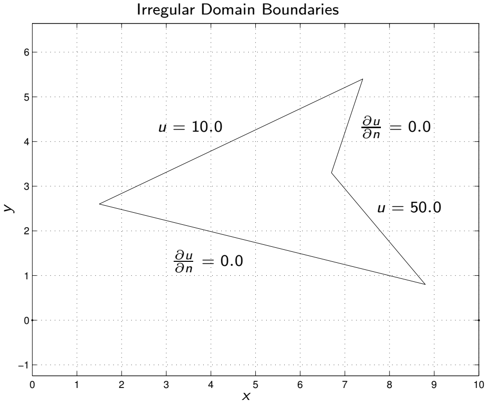
PDEs in Multiple Space Dimensions III

The third problem has to do with the location of the *boundary conditions*. Until now, we could always assume that the boundary conditions were applied at grid points.

PDEs in Multiple Space Dimensions III

The third problem has to do with the location of the *boundary conditions*. Until now, we could always assume that the boundary conditions were applied at grid points.

We can no longer make that assumption in the 2D and 3D cases:



PDEs in Multiple Space Dimensions IV

Let us assume that four neighboring values on grid points in x-direction for $y = y_j$ are $u_{1,j}$, $u_{2,j}$, $u_{3,j}$, and $u_{4,j}$. Let us assume further that the boundary value is known at $x = x_{1.35}$ located between x_1 and x_2.

PDEs in Multiple Space Dimensions IV

Let us assume that four neighboring values on grid points in x-direction for $y = y_j$ are $u_{1,j}$, $u_{2,j}$, $u_{3,j}$, and $u_{4,j}$. Let us assume further that the boundary value is known at $x = x_{1.35}$ located between x_1 and x_2.

If we know the four solution values $u_{1,j}$, $u_{2,j}$, $u_{3,j}$, and $u_{4,j}$, we can use the *Nordsieck vector* approach to compute $u_{1.35,j}$. $u_{1.35,j}$ can be expressed as a weighted sum of $u_{1,j}$, $u_{2,j}$, $u_{3,j}$, and $u_{4,j}$.

PDEs in Multiple Space Dimensions IV

Let us assume that four neighboring values on grid points in x-direction for $y = y_j$ are $u_{1,j}$, $u_{2,j}$, $u_{3,j}$, and $u_{4,j}$. Let us assume further that the boundary value is known at $x = x_{1.35}$ located between x_1 and x_2.

If we know the four solution values $u_{1,j}$, $u_{2,j}$, $u_{3,j}$, and $u_{4,j}$, we can use the *Nordsieck vector* approach to compute $u_{1.35,j}$. $u_{1.35,j}$ can be expressed as a weighted sum of $u_{1,j}$, $u_{2,j}$, $u_{3,j}$, and $u_{4,j}$.

In reality, however, we know $u_{1.35,j}$ (*boundary value*), and $u_{2,j}$, $u_{3,j}$, and $u_{4,j}$ (*through numerical integration - internal to the domain*). What is unknown is $u_{1,j}$ (*external to the domain*).

PDEs in Multiple Space Dimensions IV

Let us assume that four neighboring values on grid points in x-direction for $y = y_j$ are $u_{1,j}$, $u_{2,j}$, $u_{3,j}$, and $u_{4,j}$. Let us assume further that the boundary value is known at $x = x_{1.35}$ located between x_1 and x_2.

If we know the four solution values $u_{1,j}$, $u_{2,j}$, $u_{3,j}$, and $u_{4,j}$, we can use the *Nordsieck vector* approach to compute $u_{1.35,j}$. $u_{1.35,j}$ can be expressed as a weighted sum of $u_{1,j}$, $u_{2,j}$, $u_{3,j}$, and $u_{4,j}$.

In reality, however, we know $u_{1.35,j}$ (*boundary value*), and $u_{2,j}$, $u_{3,j}$, and $u_{4,j}$ (*through numerical integration - internal to the domain*). What is unknown is $u_{1,j}$ (*external to the domain*).

Thus, we need to solve the previously determined equation for the unknown $u_{1,j}$ instead for the known $u_{1.35,j}$.

PDEs in Multiple Space Dimensions V

- PDEs in one space dimension were still lots of fun. PDEs in multiple space dimensions are painful, to say the least.

PDEs in Multiple Space Dimensions V

- PDEs in one space dimension were still lots of fun. PDEs in multiple space dimensions are painful, to say the least.
- A large number of applied mathematicians devote their entire academic careers to nothing but solving these types of challenging numerical PDE problems.

PDEs in Multiple Space Dimensions V

- PDEs in one space dimension were still lots of fun. PDEs in multiple space dimensions are painful, to say the least.
- A large number of applied mathematicians devote their entire academic careers to nothing but solving these types of challenging numerical PDE problems.
- Unfortunately, the recipes that they have come up with so far are often rather *ad hoc*. There are no good theories available yet for which techniques work best when and why.

PDEs in Multiple Space Dimensions V

- PDEs in one space dimension were still lots of fun. PDEs in multiple space dimensions are painful, to say the least.
- A large number of applied mathematicians devote their entire academic careers to nothing but solving these types of challenging numerical PDE problems.
- Unfortunately, the recipes that they have come up with so far are often rather *ad hoc*. There are no good theories available yet for which techniques work best when and why.
- Consequently, there remains a formidable amount of research yet to be explored.

Elliptic PDEs

The class of *elliptic PDEs* is the one easiest to solve. yet, we have saved it until now, because elliptic PDE problems are rarely defined in a single space dimension.

Elliptic PDEs

The class of *elliptic PDEs* is the one easiest to solve. yet, we have saved it until now, because elliptic PDE problems are rarely defined in a single space dimension.

The simplest elliptic PDE is the *Laplace equation*, e.g. in two space dimensions:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0.0$$

Elliptic PDEs

The class of *elliptic PDEs* is the one easiest to solve. yet, we have saved it until now, because elliptic PDE problems are rarely defined in a single space dimension.

The simplest elliptic PDE is the *Laplace equation*, e.g. in two space dimensions:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0.0$$

Let us assume the Laplace equation is defined in a circular domain of radius $r = 1.0$ around the origin. Since the domain is circular, it is much more appropriate to formulate the problem using *polar coordinates*:

$$x = r \cdot \cos\varphi$$
$$y = r \cdot \sin\varphi$$

or:

$$r = \sqrt{x^2 + y^2}$$
$$\varphi = \arctan\left(\frac{y}{x}\right)$$

Elliptic PDEs II

We can express $u(x, y)$ as $\tilde{u}(r(x, y), \varphi(x, y))$. Thus:

$$\frac{\partial u}{\partial x} = \frac{\partial \tilde{u}}{\partial r} \cdot \frac{\partial r}{\partial x} + \frac{\partial \tilde{u}}{\partial \varphi} \cdot \frac{\partial \varphi}{\partial x}$$

or, in short–hand notation:

$$u_x = \tilde{u}_r \cdot r_x + \tilde{u}_\varphi \cdot \varphi_x$$

Elliptic PDEs II

We can express $u(x, y)$ as $\tilde{u}(r(x, y), \varphi(x, y))$. Thus:

$$\frac{\partial u}{\partial x} = \frac{\partial \tilde{u}}{\partial r} \cdot \frac{\partial r}{\partial x} + \frac{\partial \tilde{u}}{\partial \varphi} \cdot \frac{\partial \varphi}{\partial x}$$

or, in short–hand notation:

$$u_x = \tilde{u}_r \cdot r_x + \tilde{u}_\varphi \cdot \varphi_x$$

Using the chain rule and the multiplication rule, we find:

$$\begin{aligned} u_{xx} + u_{yy} = {} & \left(r_x^2 + r_y^2\right) \tilde{u}_{rr} + 2\left(r_x \varphi_x + r_y \varphi_y\right) \tilde{u}_{r\varphi} + \left(\varphi_x^2 + \varphi_y^2\right) \tilde{u}_{\varphi\varphi} \\ & + \left(r_{xx} + r_{yy}\right) \tilde{u}_r + \left(\varphi_{xx} + \varphi_{yy}\right) \tilde{u}_\varphi \end{aligned}$$

Elliptic PDEs II

We can express $u(x, y)$ as $\tilde{u}(r(x, y), \varphi(x, y))$. Thus:

$$\frac{\partial u}{\partial x} = \frac{\partial \tilde{u}}{\partial r} \cdot \frac{\partial r}{\partial x} + \frac{\partial \tilde{u}}{\partial \varphi} \cdot \frac{\partial \varphi}{\partial x}$$

or, in short–hand notation:

$$u_x = \tilde{u}_r \cdot r_x + \tilde{u}_\varphi \cdot \varphi_x$$

Using the chain rule and the multiplication rule, we find:

$$\begin{aligned} u_{xx} + u_{yy} = &\left(r_x^2 + r_y^2\right) \tilde{u}_{rr} + 2\left(r_x \varphi_x + r_y \varphi_y\right) \tilde{u}_{r\varphi} + \left(\varphi_x^2 + \varphi_y^2\right) \tilde{u}_{\varphi\varphi} \\ &+ \left(r_{xx} + r_{yy}\right) \tilde{u}_r + \left(\varphi_{xx} + \varphi_{yy}\right) \tilde{u}_\varphi \end{aligned}$$

or finally:

$$\frac{\partial^2 \tilde{u}}{\partial r^2} + \frac{1}{r} \cdot \frac{\partial \tilde{u}}{\partial r} + \frac{1}{r^2} \cdot \frac{\partial^2 \tilde{u}}{\partial \varphi^2} = 0.0$$

Elliptic PDEs III

The boundary condition might be:

$$\frac{\partial \tilde{u}}{\partial r} = f(\varphi, t)$$

Elliptic PDEs III

The boundary condition might be:

$$\frac{\partial \tilde{u}}{\partial r} = f(\varphi, t)$$

- Notice that there is *no need for any initial condition*, since the PDE doesn't depend on time at all (except possibly through the boundary condition as in the above example).

Elliptic PDEs III

The boundary condition might be:

$$\frac{\partial \tilde{u}}{\partial r} = f(\varphi, t)$$

- Notice that there is *no need for any initial condition*, since the PDE doesn't depend on time at all (except possibly through the boundary condition as in the above example).
- No numerical integration across time will take place at all. **We are thus in trouble with our MOL methodology.**

Elliptic PDEs III

The boundary condition might be:

$$\frac{\partial \tilde{u}}{\partial r} = f(\varphi, t)$$

- Notice that there is *no need for any initial condition*, since the PDE doesn't depend on time at all (except possibly through the boundary condition as in the above example).
- No numerical integration across time will take place at all. **We are thus in trouble with our MOL methodology.**
- We may still be able to apply the MOL approach by either differentiating along r and integrating along φ, or alternatively, by differentiating along φ and integrating along r.

Elliptic PDEs III

The boundary condition might be:

$$\frac{\partial \tilde{u}}{\partial r} = f(\varphi, t)$$

- Notice that there is *no need for any initial condition*, since the PDE doesn't depend on time at all (except possibly through the boundary condition as in the above example).
- No numerical integration across time will take place at all. **We are thus in trouble with our MOL methodology.**
- We may still be able to apply the MOL approach by either differentiating along r and integrating along φ, or alternatively, by differentiating along φ and integrating along r.
- In both cases, however, we would be *lacking one initial condition*, and would instead have *one final condition too many*. This is therefore not an *initial value problem*, but rather a *boundary value problem*.

Invariant Embedding

Let us simplify the boundary condition a bit by assuming that it does not depend on time. In this case, the problem is totally *static*, i.e., the solution is not time-dependent at all. The solution consists simply of a set of u-values at the grid points.

Invariant Embedding

Let us simplify the boundary condition a bit by assuming that it does not depend on time. In this case, the problem is totally *static*, i.e., the solution is not time-dependent at all. The solution consists simply of a set of u-values at the grid points.

We can now embed this problem within another problem as follows:

$$\frac{\partial \tilde{u}}{\partial t} = \frac{\partial^2 \tilde{u}}{\partial r^2} + \frac{1}{r} \cdot \frac{\partial \tilde{u}}{\partial r} + \frac{1}{r^2} \cdot \frac{\partial^2 \tilde{u}}{\partial \varphi^2}$$

with the boundary condition:

$$\frac{\partial \tilde{u}}{\partial r} = f(\varphi)$$

and with arbitrary initial conditions.

Invariant Embedding II

- This is now clearly a *parabolic initial value problem*, which we already know how to solve.

Invariant Embedding II

- This is now clearly a *parabolic initial value problem*, which we already know how to solve.
- Since the PDE is analytically stable, and since the boundary condition is not a function of time, the solution will eventually settle into a *steady state*.

Invariant Embedding II

- This is now clearly a *parabolic initial value problem*, which we already know how to solve.
- Since the PDE is analytically stable, and since the boundary condition is not a function of time, the solution will eventually settle into a *steady state*.
- Once the steady state has been reached, the solution no longer changes with time, thus:

$$\frac{\partial \tilde{u}}{\partial t} = 0.0$$

Invariant Embedding II

- This is now clearly a *parabolic initial value problem*, which we already know how to solve.
- Since the PDE is analytically stable, and since the boundary condition is not a function of time, the solution will eventually settle into a *steady state*.
- Once the steady state has been reached, the solution no longer changes with time, thus:

$$\frac{\partial \tilde{u}}{\partial t} = 0.0$$

- we conclude that the steady-state solution of the parabolic PDE is identical with the solution of the original *elliptic PDE*.

Invariant Embedding II

- This is now clearly a *parabolic initial value problem*, which we already know how to solve.
- Since the PDE is analytically stable, and since the boundary condition is not a function of time, the solution will eventually settle into a *steady state*.
- Once the steady state has been reached, the solution no longer changes with time, thus:

$$\frac{\partial \tilde{u}}{\partial t} = 0.0$$

- we conclude that the steady-state solution of the parabolic PDE is identical with the solution of the original *elliptic PDE*.
- This method of solving elliptic PDEs is called *invariant embedding*.

Invariant Embedding II

- This is now clearly a *parabolic initial value problem*, which we already know how to solve.
- Since the PDE is analytically stable, and since the boundary condition is not a function of time, the solution will eventually settle into a *steady state*.
- Once the steady state has been reached, the solution no longer changes with time, thus:

$$\frac{\partial \tilde{u}}{\partial t} = 0.0$$

- we conclude that the steady-state solution of the parabolic PDE is identical with the solution of the original *elliptic PDE*.
- This method of solving elliptic PDEs is called *invariant embedding*.
- The price that we had to pay for this comfort is formidable. We were able to convert a *boundary value problem* into an *initial value problem* at the expense of increasing the number of dimensions by one.

Finite Element Approximations

Those of you who followed my companion class on *Mathematical Modeling of Physical Systems* know my reservations against writing down mathematical formulae deprived of their physical meaning. Mathematics is no end in itself. **Mathematics is simply the language of physics.**

Finite Element Approximations

Those of you who followed my companion class on *Mathematical Modeling of Physical Systems* know my reservations against writing down mathematical formulae deprived of their physical meaning. Mathematics is no end in itself. **Mathematics is simply the language of physics.**

Voltages and currents in an electronic circuit don't change their values as functions of time, because they observe some differential equations. They change their values in order to bring the system to a state of minimal energy.

Finite Element Approximations

Those of you who followed my companion class on *Mathematical Modeling of Physical Systems* know my reservations against writing down mathematical formulae deprived of their physical meaning. Mathematics is no end in itself. **Mathematics is simply the language of physics.**

Voltages and currents in an electronic circuit don't change their values as functions of time, because they observe some differential equations. They change their values in order to bring the system to a state of minimal energy.

A differential equation is not the cause that makes physics tick, it is only one way of describing, in mathematical terms and after the fact, what happens in the process of energy exchange taking place in the physical system.

Finite Element Approximations II

- Looking at the solution of the previously discussed Laplace equation, we know that the solution will minimize the amount of energy stored in the system.

Finite Element Approximations II

- Looking at the solution of the previously discussed Laplace equation, we know that the solution will minimize the amount of energy stored in the system.

- Consequently, we can write an *energy function* parameterized in the (unknown) solution values, and solve a minimization problem over the set of unknown parameters. This leads to a set of algebraic equations, possibly non-linear, in the unknown solution vector.

Finite Element Approximations II

- Looking at the solution of the previously discussed Laplace equation, we know that the solution will minimize the amount of energy stored in the system.

- Consequently, we can write an *energy function* parameterized in the (unknown) solution values, and solve a minimization problem over the set of unknown parameters. This leads to a set of algebraic equations, possibly non-linear, in the unknown solution vector.

- Approaches that follow this line of reasoning are called *finite element methods*. They come in many shades and colors.

Finite Element Approximations II

- Looking at the solution of the previously discussed Laplace equation, we know that the solution will minimize the amount of energy stored in the system.

- Consequently, we can write an *energy function* parameterized in the (unknown) solution values, and solve a minimization problem over the set of unknown parameters. This leads to a set of algebraic equations, possibly non-linear, in the unknown solution vector.

- Approaches that follow this line of reasoning are called *finite element methods*. They come in many shades and colors.

- The technique was originally developed by civil engineers trying to determine the static stress in bridges and other building structures. However, the method has a much broader range of possible applications.

Finite Element Approximations II

- Looking at the solution of the previously discussed Laplace equation, we know that the solution will minimize the amount of energy stored in the system.

- Consequently, we can write an *energy function* parameterized in the (unknown) solution values, and solve a minimization problem over the set of unknown parameters. This leads to a set of algebraic equations, possibly non-linear, in the unknown solution vector.

- Approaches that follow this line of reasoning are called *finite element methods*. They come in many shades and colors.

- The technique was originally developed by civil engineers trying to determine the static stress in bridges and other building structures. However, the method has a much broader range of possible applications.

- For all practical purposes, it can be viewed as an alternative to the finite difference approaches. Thus, it can conceptually also be used for other than elliptic PDEs.

Finite Element Approximations III

- The two approaches have their own particular advantages and disadvantages.

Finite Element Approximations III

- The two approaches have their own particular advantages and disadvantages.
- Finite elements usually are *less infected by problems with consistency errors* than finite difference methods. Consequently, we can get by with a larger (and irregular) mesh, and thus, with a smaller number of equations.

Finite Element Approximations III

- The two approaches have their own particular advantages and disadvantages.
- Finite elements usually are *less infected by problems with consistency errors* than finite difference methods. Consequently, we can get by with a larger (and irregular) mesh, and thus, with a smaller number of equations.
- On the other hand, finite difference approximations always lead to *sparse matrices*. Finite element approximations do not share this property. As a consequence, although the number of equations is smaller in the finite element case, we may not be able to use sparse matrix techniques, and it is therefore not evident that the smaller system size truly leads to a more economical algorithm.

Finite Element Approximations III

- The two approaches have their own particular advantages and disadvantages.
- Finite elements usually are *less infected by problems with consistency errors* than finite difference methods. Consequently, we can get by with a larger (and irregular) mesh, and thus, with a smaller number of equations.
- On the other hand, finite difference approximations always lead to *sparse matrices*. Finite element approximations do not share this property. As a consequence, although the number of equations is smaller in the finite element case, we may not be able to use sparse matrix techniques, and it is therefore not evident that the smaller system size truly leads to a more economical algorithm.
- Also, a finite difference formulation is usually easier to derive and harder to solve than a finite element formulation.

Finite Element Approximations III

- The two approaches have their own particular advantages and disadvantages.
- Finite elements usually are *less infected by problems with consistency errors* than finite difference methods. Consequently, we can get by with a larger (and irregular) mesh, and thus, with a smaller number of equations.
- On the other hand, finite difference approximations always lead to *sparse matrices*. Finite element approximations do not share this property. As a consequence, although the number of equations is smaller in the finite element case, we may not be able to use sparse matrix techniques, and it is therefore not evident that the smaller system size truly leads to a more economical algorithm.
- Also, a finite difference formulation is usually easier to derive and harder to solve than a finite element formulation.
- However, it is easier to incorporate *irregular and even non-convex domain boundaries* into a finite element description.

Simulation of Elliptic PDEs

- Although elliptic PDEs are the *numerically most benign class of distributed parameter system models*, they are by no means trivial to simulate.

Simulation of Elliptic PDEs

- Although elliptic PDEs are the *numerically most benign class of distributed parameter system models*, they are by no means trivial to simulate.
- In the 1980s, Rice and Boisvert undertook a large research effort to collect an impressive series of algorithms for the numerical solution of elliptic PDE problems. They encoded them in a software called **Ellpack**.

Simulation of Elliptic PDEs

- Although elliptic PDEs are the *numerically most benign class of distributed parameter system models*, they are by no means trivial to simulate.

- In the 1980s, Rice and Boisvert undertook a large research effort to collect an impressive series of algorithms for the numerical solution of elliptic PDE problems. They encoded them in a software called **Ellpack**.

- Although Ellpack represents the fruit of many man-years of research and resulted in a Fortran code with several hundreds of subroutines and many thousands of lines of code, Ellpack was unable to conquer the elliptic PDE market. Ellpack was a research tool that allowed us to quickly experiment with many different combinations of algorithms, but the resulting simulation code was too sluggish to be practically useful. After finding out, which algorithms worked on our specific problems, we then had to recode these algorithms from scratch to get software that could be used for the simulation of large-scale structures.

Simulation of Elliptic PDEs

- Although elliptic PDEs are the *numerically most benign class of distributed parameter system models*, they are by no means trivial to simulate.
- In the 1980s, Rice and Boisvert undertook a large research effort to collect an impressive series of algorithms for the numerical solution of elliptic PDE problems. They encoded them in a software called **Ellpack**.
- Although Ellpack represents the fruit of many man-years of research and resulted in a Fortran code with several hundreds of subroutines and many thousands of lines of code, Ellpack was unable to conquer the elliptic PDE market. Ellpack was a research tool that allowed us to quickly experiment with many different combinations of algorithms, but the resulting simulation code was too sluggish to be practically useful. After finding out, which algorithms worked on our specific problems, we then had to recode these algorithms from scratch to get software that could be used for the simulation of large-scale structures.
- We used this tool primarily for experimenting with electronic device simulations, in particular with simulating breakdown phenomena in reverse-biased power transistors and studying the effects of total dose ionizing radiation on such devices.

Conclusions

- In this third presentation on *distributed parameter system simulation*, we looked at models in multiple space dimensions.

Conclusions

- In this third presentation on *distributed parameter system simulation*, we looked at models in multiple space dimensions.

- In particular, we discussed issues with and solution methods for the class of *elliptic PDEs*.

Conclusions

- In this third presentation on *distributed parameter system simulation*, we looked at models in multiple space dimensions.
- In particular, we discussed issues with and solution methods for the class of *elliptic PDEs*.
- One quite general approach for converting *boundary value problems* to equivalent *initial value problems* is the technique of *invariant embedding*. This technique was demonstrated by embedding an elliptic PDE in two space dimensions into an equivalent parabolic PDE in two space dimensions and one time dimension.

Conclusions

- In this third presentation on *distributed parameter system simulation*, we looked at models in multiple space dimensions.
- In particular, we discussed issues with and solution methods for the class of *elliptic PDEs*.
- One quite general approach for converting *boundary value problems* to equivalent *initial value problems* is the technique of *invariant embedding*. This technique was demonstrated by embedding an elliptic PDE in two space dimensions into an equivalent parabolic PDE in two space dimensions and one time dimension.
- The presentation ended with a very brief introduction to *finite element methods*.

References

1. Davis, K.R., R.D. Schrimpf, F.E. Cellier, K.F. Galloway, D.I. Burton, and C.F. Wheatley, Jr. (1989), "The Effects of Ionizing Radiation on Power-MOSFET Termination Structures," *IEEE Trans. Nuclear Science*, **36**(6), pp.2104-2109.

2. Kosier, S.L., R.D. Schrimpf, F.E. Cellier, and K.F. Galloway (1990), "The Effects of Ionizing Radiation on the Breakdown Voltage of P-Channel Power MOSFETs," *IEEE Trans. Nuclear Science*, **37**(6), pp.2076-2082.

3. Kosier, S.L., R.D. Schrimpf, K.F. Galloway, and F.E. Cellier (1991), "Predicting Worst-Case Charge Buildup in Power-Device Field Oxides," *IEEE Trans. Nuclear Science*, **38**(6), pp.1383-1390.

4. Wu, Q.M., and F.E. Cellier (1986), "Simulation of High-Voltage Bipolar Devices in the Neighborhood of Breakdown," *Mathematics and Computers in Simulation*, **28**, pp.271-284.

5. Wu, Q.M., C.M. Yen, and F.E. Cellier (1989), "Analysis of Breakdown Phenomena in High-Voltage Bipolar Devices," *Transactions of SCS*, **6**(1), pp.43-60.

References II

1. Davis, Kenneth R. (1989), *Two-Dimensional Simulation of the Effects of Total Dose Ionizing Radiation on Power-MOSFET Breakdown*, MS Thesis, Dept. of Electrical & Computer Engineering, University of Arizona, Tucson, AZ.

2. Kosier, Stephen L. (1990), *Breakdown Behavior of Electronic Devices under the Influence of Total Dose Ionizing Radiation*, MS Thesis, Dept. of Electrical & Computer Engineering, University of Arizona, Tucson, AZ.

3. Tan, Leong Hin (1989), *Two-Dimensional Device Simulation of Junction Termination Structures for Determination of Breakdown Behavior*, MS Thesis, Dept. of Electrical & Computer Engineering, University of Arizona, Tucson, AZ.

4. Yen, Chi-Min (1988), *Two-Dimensional Simulation of Power MOSFET Near Breakdown*, MS Thesis, Dept. of Electrical & Computer Engineering, University of Arizona, Tucson, AZ.