Solving the 1D heat equation

Explicit versus implicit finite difference schemes

GEO441 Homework 3

February 16, 2022

Problem statement

The heat equation in 1D

Conservation and constitutive equations

- Heat energy conservation law

$$\rho(x)c_p(x)\partial_t T(x,t) = -\partial_x q(x,t) \tag{1}$$

- Fourier's heat conduction law

$$q(x,t) = -K(x)\partial_x T(x,t) \tag{2}$$

Where, $T(x,t)$ is the temperature, $c_p(x)$ the specific heat at constant pressure, $\rho(x)$ is density, $K(x)$ the thermal conductivity and $q(x,t)$ the heat flux.

The heat equation in 1D

Conservation and constitutive equations

- Heat energy conservation law

$$\rho(x)c_p(x)\partial_t T(x,t) = -\partial_x q(x,t) \tag{1}$$

- Fourier's heat conduction law

$$q(x,t) = -K(x)\partial_x T(x,t) \tag{2}$$

Where, $T(x,t)$ is the temperature, $c_p(x)$ the specific heat at constant pressure, $\rho(x)$ is density, $K(x)$ the thermal conductivity and $q(x,t)$ the heat flux.

Injecting (2) into (1), we obtain the **heterogeneous heat equation**

$$\boxed{\rho(x)c_p(x)\partial_t T(x,t) = \partial_x\left[K(x)\partial_x T(x,t)\right]} \tag{3}$$

The heat equation in 1D

Conservation and constitutive equations

- Heat energy conservation law

$$\rho(x)c_p(x)\partial_t T(x,t) = -\partial_x q(x,t) \tag{1}$$

- Fourier's heat conduction law

$$q(x,t) = -K(x)\partial_x T(x,t) \tag{2}$$

Where, $T(x,t)$ is the temperature, $c_p(x)$ the specific heat at constant pressure, $\rho(x)$ is density, $K(x)$ the thermal conductivity and $q(x,t)$ the heat flux.

Injecting (2) into (1), we obtain the **heterogeneous heat equation**

$$\boxed{\rho(x)c_p(x)\partial_t T(x,t) = \partial_x\left[K(x)\partial_x T(x,t)\right]} \tag{3}$$

Homogeneous heat equation

Now, if we consider constant thermal properties in equation (3), we obtain

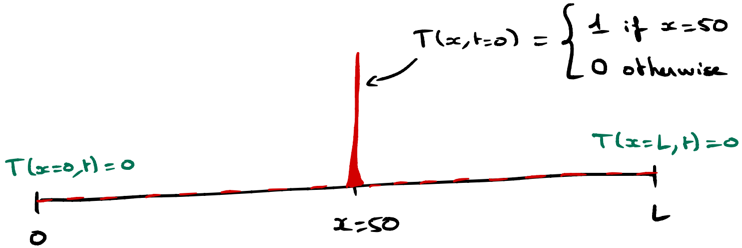
$$\boxed{\partial_t T(x,t) = D\partial_x^2 T(x,t)} \tag{4}$$

where we have introduced the thermal diffusivity $D = \frac{K}{\rho c_p}$.

Problem setup and finite-difference discretization

We consider a rod of length $\mathbf{L} = \mathbf{100}$

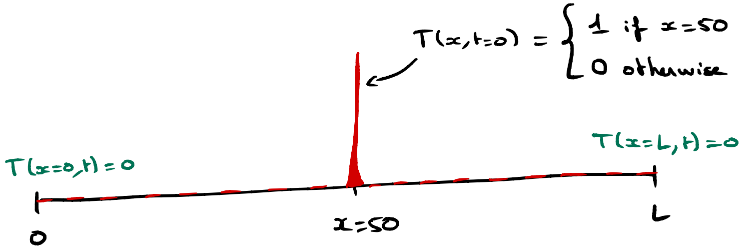
- suddenly heated at time $\mathbf{t} = \mathbf{0}$ and location $\mathbf{x} = \mathbf{50}$ to a temperature $\mathbf{T} = \mathbf{1}$
- whose ends $\mathbf{x} = \mathbf{0}$ and $\mathbf{x} = \mathbf{L}$ present null temperatures $\mathbf{T} = \mathbf{0}$ for any time t



Problem setup and finite-difference discretization

We consider a rod of length $\mathbf{L} = \mathbf{100}$

- suddenly heated at time $\mathbf{t} = \mathbf{0}$ and location $\mathbf{x} = \mathbf{50}$ to a temperature $\mathbf{T} = \mathbf{1}$
- whose ends $\mathbf{x} = \mathbf{0}$ and $\mathbf{x} = \mathbf{L}$ present null temperatures $\mathbf{T} = \mathbf{0}$ for any time t



As usual with finite-difference methods, we **discretize** the problem in time and space:

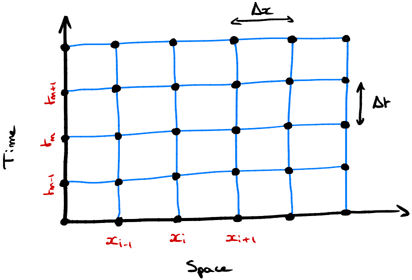
- the rod with N grid points evenly spaced by a grid step $\Delta\mathbf{x} = \frac{\mathbf{L}}{\mathbf{N}-\mathbf{1}}$
- time with M samples evenly spaced by a time step $\Delta\mathbf{t} = \frac{\mathbf{T}}{\mathbf{M}-\mathbf{1}}$

Such that $\mathbf{t} = (\mathbf{n} - 1)\Delta\mathbf{t}$ and $\mathbf{x} = (\mathbf{i} - 1)\Delta\mathbf{x}$, and such that $\mathbf{T}(\mathbf{x}, \mathbf{t}) = \mathbf{T}_{\mathbf{i}}^{\mathbf{n}}$.

Problem setup and finite-difference discretization

We consider a rod of length $\mathbf{L} = \mathbf{100}$

- suddenly heated at time $\mathbf{t} = \mathbf{0}$ and location $\mathbf{x} = \mathbf{50}$ to a temperature $\mathbf{T} = \mathbf{1}$
- whose ends $\mathbf{x} = \mathbf{0}$ and $\mathbf{x} = \mathbf{L}$ present null temperatures $\mathbf{T} = \mathbf{0}$ for any time t



As usual with finite-difference methods, we **discretize** the problem in time and space:

- the rod with N grid points evenly spaced by a grid step $\mathbf{\Delta x} = \frac{\mathbf{L}}{\mathbf{N-1}}$
- time with M samples evenly spaced by a time step $\mathbf{\Delta t} = \frac{\mathbf{T}}{\mathbf{M-1}}$

Such that $\mathbf{t} = (\mathbf{n} - 1)\Delta\mathbf{t}$ and $\mathbf{x} = (\mathbf{i} - 1)\Delta\mathbf{x}$, and such that $\mathbf{T}(\mathbf{x}, \mathbf{t}) = \mathbf{T}_\mathbf{i}^\mathbf{n}$.

Finite-difference approximation of derivatives you will need

Spatial derivatives: central difference

$$\partial_x^2 T_i^n \approx \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} + \mathcal{O}(\Delta x^2) \tag{5}$$

Time derivatives: forward and backward differences

- Forward difference

$$\partial_t T_i^n \approx \frac{T_i^{n+1} - T_i^n}{\Delta t} + \mathcal{O}(\Delta t) \tag{6}$$

$\rightarrow$ leads to an **explicit** time integration

- Backward difference

$$\partial_t T_i^n \approx \frac{T_i^n - T_i^{n-1}}{\Delta t} + \mathcal{O}(\Delta t) \tag{7}$$

$\rightarrow$ leads to an **implicit** time integration

Explicit finite-difference scheme

Explicit scheme: forward difference in time and central difference in space

Explicit problem

Use central differences in space and forward differences in time

$$\partial_x^2 T_i^n \approx \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} \qquad \text{and} \qquad \partial_t T_i^n \approx \frac{T_i^{n+1} - T_i^n}{\Delta t} \tag{8}$$

to solve the homogeneous heat equation

$$\partial_t T(x,t) = D\partial_x^2 T(x,t) \tag{9}$$

Explicit scheme: forward difference in time and central difference in space

Explicit problem

Use central differences in space and forward differences in time

$$\partial_x^2 T_i^n \approx \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} \qquad \text{and} \qquad \partial_t T_i^n \approx \frac{T_i^{n+1} - T_i^n}{\Delta t} \tag{8}$$

to solve the homogeneous heat equation

$$\partial_t T(x,t) = D\partial_x^2 T(x,t) \tag{9}$$

Solution

$$T_i^{n+1} \quad ???? \tag{10}$$

Explicit scheme: forward difference in time and central difference in space

Explicit problem

Use central differences in space and forward differences in time

$$\partial_x^2 T_i^n \approx \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} \qquad \text{and} \qquad \partial_t T_i^n \approx \frac{T_i^{n+1} - T_i^n}{\Delta t} \tag{8}$$

to solve the homogeneous heat equation

$$\partial_t T(x,t) = D\partial_x^2 T(x,t) \tag{9}$$

Solution

$$T_i^{n+1} = T_i^n + \frac{D\Delta t}{\Delta x^2}\left[T_{i+1}^n - 2T_i^n + T_{i-1}^n\right] \tag{10}$$

Explicit scheme: forward difference in time and central difference in space

Explicit problem

Use central differences in space and forward differences in time

$$\partial_x^2 T_i^n \approx \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} \qquad \text{and} \qquad \partial_t T_i^n \approx \frac{T_i^{n+1} - T_i^n}{\Delta t} \tag{8}$$

to solve the homogeneous heat equation

$$\partial_t T(x,t) = D\partial_x^2 T(x,t) \tag{9}$$

Solution

$$T_i^{n+1} = T_i^n + \frac{D\Delta t}{\Delta x^2}\left[T_{i+1}^n - 2T_i^n + T_{i-1}^n\right] \tag{10}$$

Can you write this in matrix form and find coefficients?

$$\begin{bmatrix} d_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & d_{M-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & d_M \end{bmatrix} \begin{bmatrix} T_1^{n+1} \\ T_2^{n+1} \\ T_3^{n+1} \\ \cdots \\ T_{M-1}^{n+1} \\ T_M^{n+1} \end{bmatrix} = \begin{bmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & 0 \\ 0 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 0 & a_{M-1} & b_{M-1} & c_{M-1} \\ 0 & 0 & 0 & 0 & a_M & b_M \end{bmatrix} \begin{bmatrix} T_1^n \\ T_2^n \\ T_3^n \\ \cdots \\ T_{M-1}^n \\ T_M^n \end{bmatrix}$$

Explicit scheme – Assignment

Explicit scheme

The finite-difference scheme

$$T_i^{n+1} = T_i^n + \frac{D\Delta t}{\Delta x^2}\left[T_{i+1}^n - 2T_i^n + T_{i-1}^n\right] \tag{11}$$

is called **explicit** because solutions at T^{n+1} only depends on solution T^n
$\rightarrow$ we can march the solution in time and do not need to form/invert matrices

Stability

Warning: this scheme is only **conditionally stable**[a]

$$\boxed{\Delta t \leq \frac{\Delta x^2}{2D}} \tag{12}$$

[a] See appendix for the derivation of stability condition

Assignment:

- Solve the heat equation with this explicit scheme
- Test different time steps: check sensitivity/stability of the solution

Implicit finite-difference scheme

Implicit problem

Use central differences in space and backward differences in time

$$\partial_x^2 T_i^n \approx \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} \qquad \text{and} \qquad \partial_t T_i^n \approx \frac{T_i^n - T_i^{n-1}}{\Delta t} \tag{13}$$

to solve the homogeneous heat equation

$$\partial_t T(x,t) = D\partial_x^2 T(x,t) \tag{14}$$

Implicit scheme: backward difference in time and central difference in space

Implicit problem

Use central differences in space and backward differences in time

$$\partial_x^2 T_i^n \approx \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} \qquad \text{and} \qquad \partial_t T_i^n \approx \frac{T_i^n - T_i^{n-1}}{\Delta t} \tag{13}$$

to solve the homogeneous heat equation

$$\partial_t T(x,t) = D\partial_x^2 T(x,t) \tag{14}$$

Solution

$$T_i^n \quad ???? \tag{15}$$

Implicit scheme: backward difference in time and central difference in space

Implicit problem

Use central differences in space and backward differences in time

$$\partial_x^2 T_i^n \approx \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} \qquad \text{and} \qquad \partial_t T_i^n \approx \frac{T_i^n - T_i^{n-1}}{\Delta t} \tag{13}$$

to solve the homogeneous heat equation

$$\partial_t T(x,t) = D\partial_x^2 T(x,t) \tag{14}$$

Solution

$$T_i^n - \frac{D\Delta t}{\Delta x^2}\left[T_{i+1}^n - 2T_i^n + T_{i-1}^n\right] = T_i^{n-1} \tag{15}$$

Implicit scheme: backward difference in time and central difference in space

Implicit problem

Use central differences in space and backward differences in time

$$\partial_x^2 T_i^n \approx \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} \quad \text{and} \quad \partial_t T_i^n \approx \frac{T_i^n - T_i^{n-1}}{\Delta t} \tag{13}$$

to solve the homogeneous heat equation

$$\partial_t T(x,t) = D\partial_x^2 T(x,t) \tag{14}$$

Solution

$$T_i^n - \frac{D\Delta t}{\Delta x^2}\left[T_{i+1}^n - 2T_i^n + T_{i-1}^n\right] = T_i^{n-1} \tag{15}$$

Can you write this in matrix form and find coefficients?

$$\begin{bmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & a_{M-1} & b_{M-1} & c_{M-1} \\ 0 & 0 & 0 & 0 & a_M & b_M \end{bmatrix} \begin{bmatrix} T_1^n \\ T_2^n \\ T_3^n \\ \dots \\ T_{M-1}^n \\ T_M^n \end{bmatrix} = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & d_{M-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & d_M \end{bmatrix} \begin{bmatrix} T_1^{n-1} \\ T_2^{n-1} \\ T_3^{n-1} \\ \dots \\ T_{M-1}^{n-1} \\ T_M^{n-1} \end{bmatrix}$$

Implicit scheme – Assignment

Implicit scheme

The finite-difference scheme

$$T_i^n - \frac{D\Delta t}{\Delta x^2}\left[T_{i+1}^n - 2T_i^n + T_{i-1}^n\right] = T_i^{n-1} \tag{16}$$

is called **implicit** because solutions T^{n-1} depends on solution T^n
→ we cannot march the solution in time
→ need to **solve a linear system** at each time step (form and invert matrices)!

$$\mathbf{A}T^n = T^{n-1} \tag{17}$$

Stability

This scheme is **unconditionally stable**[a].

[a] See appendix for the derivation of stability condition

Assignment:

- Solve the heat equation with the implicit scheme
- Check different time steps (same as those used previously): compare sensitivity/stability of the solution

Notes: solving a linear system

Easy way

The linear system $\boxed{\mathbf{A}T^n = T^{n-1}}$ can easily be solved using the *inv* matlab function or using its \ operator

Under the hood

Matlab uses classical linear algebra techniques – you have probably seen before – to invert the matrix or solve the linear system:

- Gaussian elimination
- Forward elimination followed by backward substitutions (probably after an LU factorization of matrix **A**)

Tridiagonal solver – If you feel like it

We are lucky:

- our linear system involves a matrix **A** that is tridiagonal
- **A** does not change from one time iteration to another

We can therefore use an efficient tridiagonal solver (see appendix for the algorithm and some explanations)

Crank–Nicolson method

Crank–Nicolson: a semi-implicit/semi-explicit method

- Until now we have considered two time schemes – forward and backward differences – which are both first order accurate in time but conditionally and unconditionally stables respectively
- If accuracy and stability are needed for your problem, you can use the Crank-Nicolson scheme that is both second-order accurate and implicit in time

Crank-Nicolson problem

Use time average of central differences in space and backward differences in time

$$\partial_x^2 T_i^n \approx 0.5 \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n + T_{i+1}^{n-1} - 2T_i^{n-1} + T_{i-1}^{n-1}}{\Delta x^2} \quad \text{and} \quad \partial_t T_i^n \approx \frac{T_i^n - T_i^{n-1}}{\Delta t} \tag{18}$$

to solve the homogeneous heat equation

$$\partial_t T(x,t) = D\partial_x^2 T(x,t) \tag{19}$$

Assignment:

- Write the discretized form of the homogeneous heat equation with the Crank-Nicolson scheme
- Write it also in matrix form and give corresponding entries of matrices
- Write and use your program with the same configuration you used previously

Good luck!

Stability condition of the explicit scheme

The finite-difference explicit scheme is given by

$$\boxed{T_i^{n+1} = T_i^n + \frac{D\Delta t}{\Delta x^2}\left[T_{i+1}^n - 2T_i^n + T_{i-1}^n\right]} \tag{20}$$

Take plane wave solutions of the form $\boxed{T_i^n = e^{an\Delta t}e^{ikj\Delta x}}$

$$e^{a(n+1)\Delta t}e^{ikj\Delta x} = e^{an\Delta t}e^{ikj\Delta x} + \frac{D\Delta t}{\Delta x^2}\left[e^{an\Delta t}e^{ik(j+1)\Delta x} - 2e^{an\Delta t}e^{ikj\Delta x} + e^{an\Delta t}e^{ik(j-1)\Delta x}\right] \tag{21}$$

Divide by $T_i^n = e^{an\Delta t}e^{ikj\Delta x}$ (to see the growth from one iteration to another)

$$e^{a\Delta t} = 1 + \frac{D\Delta t}{\Delta x^2}\left[e^{ik\Delta x} + e^{-ik\Delta x} - 2\right] = 1 + \frac{D\Delta t}{\Delta x^2}\left[2\cos(k\Delta x) - 2\right] \tag{22}$$

$$= 1 - \frac{2D\Delta t}{\Delta x^2}\left[1 - \cos(k\Delta x)\right] \tag{23}$$

$$= 1 - \frac{2D\Delta t}{\Delta x^2}\left[1 - \{\cos^2(k\Delta x/2) - \sin^2(k\Delta x/2)\}\right] \tag{24}$$

$$= 1 - \frac{4D\Delta t}{\Delta x^2}\sin^2(k\Delta x/2) \tag{25}$$

We require $|e^{a\Delta t}| \leq 1$, since $\frac{4D\Delta t}{\Delta x^2}\sin^2(k\Delta x/2)$ is always positive, we get

$$\boxed{\frac{4D\Delta t}{\Delta x^2} \leq 2} \tag{26}$$

Stability condition of the implicit scheme

The finite-difference implicit scheme is given by

$$\boxed{T_i^n + \frac{D\Delta t}{\Delta x^2}\left[-T_{i+1}^n + 2T_i^n - T_{i-1}^n\right] = T_i^{n-1}} \tag{27}$$

Take plane wave solutions of the form $\boxed{T_i^n = e^{an\Delta t}e^{ikj\Delta x}}$

$$e^{an\Delta t}e^{ikj\Delta x} + \frac{D\Delta t}{\Delta x^2}\left[-e^{an\Delta t}e^{ik(j+1)\Delta x} + 2e^{an\Delta t}e^{ikj\Delta x} - e^{an\Delta t}e^{ik(j-1)\Delta x}\right] = e^{a(n-1)\Delta t}e^{ikj\Delta x} \tag{28}$$

Divide by $T_i^n = e^{an\Delta t}e^{ikj\Delta x}$ (to see the growth from one iteration to another)

$$1 + \frac{D\Delta t}{\Delta x^2}\left[2 - e^{ik\Delta x} - e^{-ik\Delta x}\right] = e^{-a\Delta t} \tag{29}$$

$$1 + \frac{4D\Delta t}{\Delta x^2}\sin^2(k\Delta x/2) = e^{-a\Delta t} \tag{30}$$

Therefore,

$$e^{a\Delta t} = \frac{1}{1 + \frac{4D\Delta t}{\Delta x^2}\sin^2(k\Delta x/2)} \tag{31}$$

We require $|e^{a\Delta t}| \leq 1$, which is always true since $1 + \frac{4D\Delta t}{\Delta x^2}\sin^2(k\Delta x/2) \geq 1$

Hence, this scheme is **unconditionally stable**!

Solving a linear system – Simple example

Consider the following linear system $\mathbf{Ax} = \mathbf{b}$ involving two unknowns

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{32}$$

Multiplying row 1 with a_{21} and row 2 with a_{11}, we get

$$\begin{bmatrix} a_{21}\,a_{11} & a_{21}\,a_{12} \\ a_{11}\,a_{21} & a_{11}\,a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_{21}\,b_1 \\ a_{11}\,b_2 \end{bmatrix} \tag{33}$$

Substracting row 1 from row 2, we can eliminate the x_1 unknown in the second equation

$$\begin{bmatrix} a_{21}\,a_{11} & a_{21}\,a_{12} \\ 0 & a_{11}\,a_{22} - a_{21}\,a_{12} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_{21}\,b_1 \\ a_{11}\,b_2 - a_{21}\,b_1 \end{bmatrix} \tag{34}$$

And solve for x_2

$$x_2 = \frac{a_{11}\,b_2 - a_{21}\,b_1}{a_{11}\,a_{22} - a_{21}\,a_{12}} \tag{35}$$

This process is called **forward elimination** and reduces the system to an upper triangular system. Then x_1 can be obtained by substituting x_2 into the first row

$$x_1 = \frac{a_{21}\,b_1 - a_{21}\,a_{12}x_2}{a_{21}\,a_{11}} \tag{36}$$

This operation is called **backward substitution**.

Solving a linear system – Larger systems

The forward-elimination / backward-substitution generalizes to larger matrices ($n \times n$).

Forward elimination

1. Eliminate x_1 from row 2 to row n[a]
   - First eliminate x1 from row 2
     $\rightarrow$ multiply row 1 by a_{21}/a_{11}
     $\rightarrow$ substract row 1 from row 2
   - Repeat this procedure for the remaining equations (rows)
     $\rightarrow$ multiply row 1 by a_{31}/a_{11}
     $\rightarrow$ substract row 1 from row 3
2. Then eliminate x_2 from row 3 to row n in the same way and repeat until elimination of x_{n-1} from row n

[a] Warning elements a_{ij} and b_i are changed during the process

You get an upper-triangular system whose last equation is $a_{nn}^{(n-1)} x_n = b_n^{(n-1)}$ (here the upper-script means that coefficients have changed $n-1$ times).

Backward substitution

1. Solve $x_n = b_n^{(n-1)} / a_{nn}^{(n-1)}$
2. Repeatedly solve $x_i = \frac{b_i^{(i-1)} - \sum_{j=i+1}^{n} a_{ij}^{(i-1)} x_j}{a_{ii}^{(i-1)}}$ for i=n-1 to i=1

Naive implementation of Gauss algorithm

Objective: solve $\mathbf{Ax} = \mathbf{b}$

```
% Some parameters and glue matrix A with vector b
[m,n] = size(A);
nb = n+1;
Aug = [A b];

% Forward elimination
for k = 1:n-1
    for i = k+1:n
        factor = Aug(i,k)/Aug(k,k);
        Aug(i,k:nb) = Aug(i,k:nb)-factor*Aug(k,k:nb);
    end
end

% Backward substitution
x = zeros(n,1);
x(n) = Aug(n,nb)/Aug(n,n);
for i = n-1:-1:1
    x(i) = (Aug(i,nb)-Aug(i,i+1:n)*x(i+1:n))/Aug(i,i);
end
```

Solving tridiagonal linear system

For a matrix of size $n \times n$, let r be the right hand side vector, f a vector containing diagonal entries of $\mathbf{A}$, e a vector containing sub (lower) diagonal entries and g a vector containing super (upper) diagonal entries.

```
% get size
n=length(f);

% forward elimination
for k = 2:n
    factor = e(k)/f(k-1);
    f(k) = f(k) - factor*g(k-1);
    r(k) = r(k) - factor*r(k-1);
end

% back substitution
x(n) = r(n)/f(n);
for k = n-1:-1:1
    x(k) = (r(k)-g(k)*x(k+1))/f(k);
end
```