Notes

Matthias Thumann

https://github.com/m-thu/sandbox/blob/master/notes.tex

Compiled on Friday 5th October, 2018 at 15:45

Contents

1 Verilog

1.1 Synthesis

Listing 1 shows the basic structure of a Verilog module. Identifiers are case-sensitive.

```
// Single line comment
/* Comment spanning
   several lines */

module some_module (
   // Ports
   input [wire/reg] in1,
   input in2,
   output [wire/reg] out
);
   // Internal signals
   wire signal1, signal2;

   // Vector, bus (32 bits wide)
   wire [31:0] bus;

   // Examples of accessing part of a vector:
   // bus[0], bus[7:0]

   // Local parameter, cannot be changed
   localparam PARAM = 42;

endmodule
```

Figure 1: Basic structure of a Verilog module

Signals can take the values 0 (logic 0), 1 (logic 1), x (undefined, uninitialized) and z (high impedance).

Integer constants: <bits>'<base><literal> (leaving out <bits> defaults to a width of at least 32 bits, underscores in < literal > are ignored). Valid values for <base>: b (binary), d (decimal), o (octal) and h (hexadecimal). Examples: 4'b01_01, 8'hab, 10.

Instantiating modules or primitives connecting ports by sequence:

```
some_module instance_name (
   // Inputs and outputs in the same order
   // as the appear in the declaration of some_module
   in1,
   in2,
   out
);
```

Connecting the ports by name:

```
some_module instance_name (
    .IN1(in1),
    .IN2(in2),
    .OUT(out)
);
```

Dataflow modeling (combinatorial circuit)

```
module my_xor (
    input a,
    input b,
    output y
);

    assign y = a ^ b;

endmodule
```

Operators

Operator	Type	Meaning
&	Bitwise	AND
|	Bitwise	OR
^	Bitwise	Exclusive OR
~	Bitwise	One's complement
{a, b, c}	Other	Concatenate wires or vectores
...	...	...

Table 1: Verilog operators

Behavioral modeling using **always** blocks:

```
// Synchronous logic
// Edge triggered
always @(posedge clock, negedge reset) begin
    if (reset)
        counter <= 0;
    else
        counter <= counter + 1;
end

// Asynchronous logic
// Sensitive to all signals in block, level triggered
always @* begin
    // ...
end
```

Non-blocking assignment: `signal <= /* ... */`

Blocking assignment: `signal = /* ... */`

Switch case

```
case (signal)
    value1: /* instruction / block */
    value2: /* instruction / block */
    // ...
    default: /* instruction / block */
endcase
```

casez treats z, **casex** x and z as don't care values.

Use default values to avoid inference of latches/ registers.

Modeling of memory with arrays:

```
// 256 bytes of memory
reg [7:0] mem [0:255];
//     ^          ^
//     |          |
//     |          -- Number of cells
//     ------------ Memory width
```

Tristate ports: **inout**, Pull-down: **tri0**, Pull-up: **tri1**, Wired-AND: **wand** or **triand**, Wired-OR: **wor** or **trior**

Declaration and instantiation of modules with parameters:

```
module some_module #(
    // Parameters with default values
    parameter PARAM1 = 1,
    parameter PARAM2 = 2
)(
    input in,
    output out
);

    /* ... */

endmodule

// Instantiation of the module above
some_module #(
    .PARAM1(param1),
    .PARAM2(param2)
) instance_name (
    /* ... */
);
```

Functions: only combinational circuits, no registers, delays and non-blocking assignments (are defined in the module in which they are used, can call other functions).

```
function my_xor;
    input a, b;

    begin
        my_xor = a ^ b;
    end
endfunction

assign y = my_xor(c, d);
```

Tasks: no return value, can have ouput and **inout** ports, can have delays

```
task my_inverter;
    // FIXME
```

generate Blocks:

```
    // FIXME
```

Compiler directives: FIXME

1.2 Simulation

Register data types: **integer**, **real**, **time**

Initial blocks:

```
module testbench;
    initial begin
        $display("Hello world!);
        $finish;
    end
endmodule
```

Delays: #10;

System tasks: **$display**("Format string", /* ... */)

$monitor("Format string", /* ... */): automatically generates output if one of the values changes

VCD (Value Change Dump) files:

```
$dumpfile("traces.vcd")
$dumpvars(0, testbench)
```

2 VHDL

2.1 Lexical Elements

Types of *comments* (can't be nested):

```
-- Single line comment
/* Comment spanning multiple
   lines (VHDL-2008 only) */
```

Valid *identifiers* consist of the characters 'a'-'z', 'A'-'Z', '0' - '9', '_' and may only start with an alphabetic letter. Identifiers with more than one consecutive underscore or identifiers that end with an underscore are illegal. Furthermore, identifiers are case insensitive in VHDL.

Examples of *numeric literals*:

```
-- Integer literals (base 10)
1 42 10e5 10E6
1_234             -- Identical to 1234, underscores get ignored

-- Real literals (base 10, always contain the decimal point '.' character)
3.1415927 1.6022e-19 8.8542E-12

-- Other bases
2#10101010#    -- Binary integer
8#777#         -- Octal integer
16#aa# 16#FF# -- Hexadecimal integer
2#1.101#       -- Binary real
```

Examples of *character and string literals*:

```
-- Characters (ISO 8859 Latin-1 8 bit character set)
'a' 'b' 'X' '_'
'''               -- Single quote

-- Strings (may not be separated by linebreaks)
"Hello world!"
""""              -- Double quote
""                -- Empty string
```

Bit string literals are enclosed in double quotes with a prefix that indicates the size of the string in bits (VHDL-2008 only) and the base. Examples:

```
b"101010" B"1111_0000" -- Binary (underscores are ignored)
"11001100"             -- Equivalent to b"11001100"
o"777" O"777"          -- Octal
x"aabb" X"FFFF"        -- Hexadecimal

-- VHDL-2008 only
8b"1111_0000" -- 8 bit long binary number
d"42" D"23"   -- Decimal number, size can't be specified

X"0X1"        -- Equivalent to B"0000XXXX1111"
B"1111_ZZ00" -- Equivalent to B"1111ZZ00"

UB"" UO"" UX"" -- Unsigned binary, octal, hexadecimal
SB"" SO"" SX"" -- Signed       "       "          "
```

2.2 Fundamental Data Types

Declaring *constants* (one of several types of objects, namely constants, variables, signals and files):

constant identifier {, ...}: type_name [:= initial_value];

Examples:

```
constant pi : real    := 3.1415927;
constant n  : natural := 42;
```

Declaring *variables*:

variable identifier {, ...}: type_name [:= initial_value];

Examples:

```
variable count : integer := 0;
variable x     : natural;
```

Assignments to variables are non-blocking (in contrast to assignments to signals), i.e. the value of the variable gets updated immediately:

[label_name:] identifier := value;

Examples:

```
x     := 1;
count := count + 1;
```

New *types* can be declared using

type identifier **is** definition ;.

Declaring new *integer* types:
type identifier **is range** bound1 (**to**|**downto**) bound2;

Example:

```
type day_of_week is range 0 to 6;

constant JAN : integer := 1;
constant DEC : integer := 12;
type month is range JAN to DEC;  -- Bounds for range can be evaluated
                                 -- at compile time
```

Operations that are defined for integer types include +, −, *, /, **mod**, **rem**, **abs** and **.

New *floating point* types are declared exactly as their integer counterparts, except the bounds for their range have to evaluate to real numbers. If the default value for floating point variables is left out, they default to the lowest possible value.

Physical types include a physical unit and can be declared as follows:

```
type identifier is range bound1 (to|downto) bound2
   units
      identifier;
      {identifier2 = (integer_literal|real_literal) unit;}
   end units [identifier];
```

Example:

```
type voltage is range 0 to 1e9
   units
      volt;              -- Base unit
      kvolt = 1000 volt;
   end units voltage;
```

VHDL includes a predefined physical type time, that defines the following unit prefixes: fs, ps, ns, us, ms, sec, min and hr.

Declaring *enumeration* types:
type identifier **is** ((identifier | character_literal) {, ...});

Examples:

```
type fsm_state is (reset, state1, state2);
variable state : fsm_state := reset;

type hexdigit is ('0','1','2','3','4','5','6','7','8','9',
                  'a','b','c','d','e','f'); -- Lowercase hex digits
variable x : hexdigit := 'a';

-- Predefined character enumeration type
variable c : character := 'x';
```

The predefined boolean enumeration type can take the values true and false. Valid expressions including booleans may use the operators = (is equal), /= (is not equal), **and**, **or**, **nand**, **nor**, **not**, **xor** and **xnor**.

Predefined bit type: **type** bit **is** ('0', '1'); (overloads '0' and '1' character literals).

Declaring *subtypes*:
subtype identifier **is** type_name **range** bound1 (**to**|**downto**) bound2;

Examples:

```
subtype uint8 is integer range 0 to 255;

variable b : uint8 := 255;

-- Predefined subtype for delays >=0 fs
variable delay : delay_length;
```

The *type qualification* operator typename'(expression) can be used to distinguish between overloaded enumeration literals:

```
type fsm1_state is (reset, a1, a2);
type fsm2_state is (reset, b1, b2);

fsm1_state'(reset) -- Type of reset literal: fsm1_state
fsm2_state'(reset) -- Type of reset literal: fsm2_state
```

Examples of *type conversions*:

```
real(42)          -- Integer to real
integer(2.71828) -- Real to integer
```

List of *attributes* applicable to scalar data types (T denotes a type name):

Attribute	**Description**
T'ascending	True: ascending range, false: descending range
T'low	Lowest possible value
T'high	Highest possible value
T'left	Leftmost element
T'right	Rightmost element
T'image(value)	Converts value to string representation
T'value(string)	Converts string to value

List of additional attributes applicable to discrete and physical types (T denotes a type name):

Attribute	Description
T'pos(value)	Index of value
T'val(j)	Value at index j
T'succ(value)	Value incremented by one
T'pred(value)	Value decremented by one
T'leftof(value)	Element to the left of value
T'rightof(value)	Element to the right of value

2.3 Sequential Control Flow Constructs

Syntax of *if* statements:

```
[label_name:]
-- VHDL-2008 only: implicit conversion of 'expression' to boolean
if expression then
   {...}
{elsif expression then
   {...}}
[else                    -- Default branch
   {...}]
end if [label_name];
```

Example:

```
entity bincnt is
   port (clk : in std_logic;
         led1, led2, led3, led4, led5 : out std_logic);
end bincnt;

architecture behavioral of bincnt is
   signal cnt : unsigned (4 downto 0) := (others => '0');
begin
   process (clk)
      variable delay : unsigned (23 downto 0) := (others => '0');
   begin
      if rising_edge(clk) then
         if delay = 2_999_999 then
            cnt <= cnt + 1;
            delay := x"000000";
         else
            delay := delay + 1;
         end if;
      end if;
   end process;

   (led5,led4,led3,led2,led1) <= std_logic_vector(cnt);
end behavioral;
```

Syntax of *conditional variable assignments* (VHDL-2008 only):

```
[label_name:]
variable := value when expression
   {else value when expression}
   [else expression];              -- Default branch
```

Syntax of *case* statements:

```
[label_name:]
case expression is
   when (expression
        | bound1 (to|downto) bound2 -- Discrete range
        | subtype_name              -- Name of subtype
        | others)                   -- Default branch
   => {...}
end case [label_name];
```

Multiple alternatives in a **when** statement can be separated with the “|” character, e.g.

```
case x is
   when 1|2    =>
      y <= 1;
   when 3      =>
      y <= 2;
   when others =>
      y <= 3;
end case;
```

Another example of a case statement:

```
entity mux4 is
   port (a0, a1, a2, a3 : in  bit;
         sel0, sel1     : in  bit;
         y              : out bit);
   constant T_PD : delay_length := 4.5 ns;
end entity mux4;

architecture behav of mux4 is
   signal sel : bit_vector(0 to 1);
begin
   sel <= (sel0, sel1);

   process (sel, a0, a1, a2, a3) is
   begin
      case sel is
         when "00" => y <= a0 after T_PD;
         when "01" => y <= a1 after T_PD;
         when "10" => y <= a2 after T_PD;
         when "11" => y <= a3 after T_PD;
      end case;
   end process;
end architecture behav;
```

Syntax of *selected variable assignments* (VHDL-2008 only):

```
[label_name:]
with expression select
variable := {value when expression,}
   value when expression;
```

Null statements (no operation): [label_name:] **null**;.

General syntax of *loops*:

```
[label_name:]
[while expression                              -- Boolean expression
 | for identifier in bound1 (to|downto) bound2] -- Discrete range
loop                                           -- Loops forever without
                                               -- while or for
   {...}
end loop [label_name];
```

Loops can be aborted with the **exit** statement: [label_name:] **exit** [label_loop] [**when** expression];, to continue with the next iteration of the loop one can use the **next** statement: [label_name:] **next** [label_loop] [**when** expression];.

Example of a loop without **while** or **for**:

```
entity counter is
   port (clk   : in bit;
         count : out natural);
end entity counter;

architecture behav of counter is
begin
   process is
      variable c : natural := 15;
   begin
      count <= c;
      loop
         wait until clk = '1';
         c := c - 1 when c /= 0
              else 15;
         count <= c;
      end loop;
   end process;
end architecture behav;
```

Example of a *for loop*:

```
entity average is
   port (clk  : in bit;
         din  : in real;
         dout : out real);
end entity average;

```

```
architecture behav of average is
   constant N : integer := 4;
begin
   process is
      variable sum : real;
   begin
      loop
         sum := 0.0;
         for i in 0 to N-1 loop
            wait until clk = '1';
            sum := sum + din;
         end loop;

         dout <= sum / real(N);
      end loop;
   end process;
end architecture behav;
```

Syntax of *assertions* (for simulation, synthesis or formal verification):

```
[label_name:] assert expression
         [report expression] [severity level];

-- Valid values for level:
-- note, warning, error, failure
```

Report statements are equivalent to unconditional assertions:
[label_name:] **report** expression [**severity** level];.

2.4 Composite Data Types

Declaring new *array* types:

```
type identifier is array
   (discrete_subtype                                    -- Index is a discrete
                                                        -- subtype
    | discrete_subtype range bound1 (to|downto) bound2  -- Index is a discrete
                                                        -- subtype with
                                                        -- constrained range
    | bound1 (to|downto) bound2                         -- Index range is
   )                                                    -- specified directly
    {, ...}                                             -- Multidimensional
                                                        -- array
   ) of subtype;
```

Example of a one-dimensional array:

```
entity reg is
   port (din         : in bit_vector (31 downto 0);
         read, write : in integer range 0 to 15;
         wen         : in bit;
```

```vhdl
        dout        : out bit_vector (31 downto 0));
end entity reg;

architecture behav of reg is
begin
   process (din, read, write, wen) is
      type mem is array (0 to 15) of bit_vector (31 downto 0);
      variable memory : mem;
   begin
      if wen = '1' then
         memory(write) := din;
      end if;
      dout <= memory(read);
   end process;
end architecture behav;
```

Array literals can be constructed out of scalar data types using *array aggregates*:

```vhdl
([expression                     -- Named association: indices explicitly
                                 --                    specified
  | discrete_subtype
  | bound1 (to|downto) bound2
  | others => ]                  -- Wildcard for any indices that haven't been
                                 -- specified previously (must be last)
  expression                     -- Positional association is used
                                 -- if the [... => ] part is left out
{, ...})
```

Example:

```vhdl
type narray is array (0 to 3) of natural;

variable n1 : narray := (1, 2, 3, 4);      -- Positional association
variable n2 : narray := (0 => 1, 1 => 2,   -- Named association
                         2 => 3, 3 => 4);
```

Multiple indices can be separated with the "|" character:

```vhdl
type rarray is array (1 to 10) of real;

variable r : rarray := (1|2|4 => 3.14, 5 to 7 => 2.718, others => 0.0);
```

Array aggregates can also be used as the target in both variable and signal assignments:

```vhdl
[label_name:] (identifier|array_aggregate) := expression;
[label_name:] (identifier|array_aggregate) <= expression;

-- Example:
variable a, b, c, d : bit;
variable x          : bit_vector (3 downto 0);

(a, b, c, d) := x;
```

In addition to just scalar types, VHDL-2008 also allows the use of sub-arrays in the specification of array aggregates.

Attributes that can be used in conjunction with array types are (A denotes either an array type or object, the index j =1,2,... which dimension the attribute shall refer to. If the parentheses are left out, j defaults to one.):

Attribute	Description
A'left(j)	Leftmost index
A'right(j)	Rightmost index
A'low(j)	Lowest index
A'high(j)	Highest index
A'**range**(j)	Index range
A'reverse_range(j)	Reversed index range
A'length(j)	Length of index range
A'ascending(j)	True: ascending range, false: descending range
A'element	Subtype of the elements in A (VHDL-2008 only)

Array types that leave the index range(s) unspecified are called *unconstrained* array types and can be declared in the following way:

```
type identifier is array
   (type_name range <>
   {, ...})           -- Multidimensional array
) of subtype;
```

Example:

```
package array_type is
   type int_array is array (natural range <>) of integer;
end package array_type;

use work.array_type.all;

entity max is
   port (din  : in int_array;
         dout : out integer);
end entity max;

architecture behav of max is
begin
   process (din) is
      variable max : integer;
   begin
      assert din'length /= 0
         report "Array has zero length!"
         severity failure;

      max := din(din'left);
      for i in din'range loop
         if din(i) > max then
```

```
            max := din(i);
         end if;
         dout <= max;
      end loop;
   end process;
end architecture behav;
```

Predefined unconstrained array types include string and bit_vector, VHDL-2008 additionally provides boolean_vector, integer_vector, real_vector and time_vector.

Parts of an array can be referenced using *array slices*:
identifier (bound1 (**to**|**downto**) bound2).

Declaring new *record* types:

```
type identifier is record
   identifier {, ...} : type_name;
   {...}
end record [identifier];
```

Single members in a record type can be referred to using *selected names*: identifier .record_member.

Example:

```
type coordinate is record
   x, y, z : real;
end record coordinate;

variable v1, v2 : coordinate;

v1.x = 1.0; v1.y = 2.0; v1.z = 3.0;
v2 = v1;
```

Record aggregates can be constructed similarly to array aggregates:

```
([expression      -- Named association: record members explicitly specified,
                  -- multiple choices may be separated with the '|' character
  | others => ]   -- Wildcard for any members that haven't been
                  -- specified previously (must be last)
  expression      -- Positional association is used
                  -- if the [... => ] part is left out
{, ...})
```

Example:

```
v1 = (1.0, 2.0, 3.0);                 -- Positional association
v1 = (x => 1.0, y => 2.0, z => 3.0); -- Named association
v2 = (others => 0.0);
```

2.5 Entities and Architectures, Behavioral and Structural Modeling

Syntax of *entity* declarations:

```
entity identifier is
   [generic (
      identifier {, ...} : typename
         [:= default_value]
      -- Generic type (VHDL-2008 only)
      | type generic_type_identifier
      {; ...}
   );]

   [port (
      identifier {, ...} : (in|out|buffer|inout) typename
         [:= default_value]
      -- Generic type (VHDL-2008 only)
      | generic_type_identifier ...
      {; ...}
   );]

   -- Declarations common to all implementations
   {...}

[begin
   -- Concurrent assertion statements
   -- Passive concurrent procedure calls
   -- Passive processes
   {...}
]
end [entity] [identifier];
```

Procedure calls and processes are *passive* if they don't include any signal assignments.

In and **out** ports are unidirectional, **inout** ports are bidirectional and **buffer** ports are **out** ports that can be read internally (VHDL-2008 also allows reading **out** ports).

Syntax of *architectures*:

```
architecture identifier of entity_name is
   -- Declarations
   {...}
begin
   -- Concurrent statements
   {...}
end [architecture] [identifier];
```

Signals (internal connections within an architecture) are declared as follows:
signal identifier {, ...} : type_name [:= initial_value];.

Syntax of *signal assignments*:

```
-- Default delay mechanism: inertial
```

```
[label_name:]
identifier <=
   [transport
    | [reject time_expression] inertial]
   value [after time_expression] {, ...};
   | unaffected;
```

If one assigns the **unaffected** value to a signal, its value remains unchanged (VHDL-2008: **unaffected** may also be used in sequential signal assignments). Not specifying any delay is equivalent to specifying a delay of 0 fs (*delta delay*).

Syntax of *conditional signal assignments* (VHDL-2008 only, replace with **if** in VHDL<2008):

```
[label_name:]
identifier <=
   [transport                          -- Delay mechanism
    | [reject time_expression] inertial]
   value [after time_expression] {, ...};
   | unaffected when expression
   {else ... when expression}
   [else ...]; -- Default branch
```

Syntax of *selected signal assignments* (VHDL-2008 only, replace with **case** in VHDL<2008):

```
[label_name:]
with expression select
identifier <=
   {... when ...,}
   [transport                          -- Delay mechanism
    | [reject time_expression] inertial]
   value [after time_expression] {, ...};
   | unaffected when expression;
```

List of attributes applicable to signals (S denotes a signal, T an expression of type time):

Attribute	Description
S'active	True if signal S got updated with a new value in the current simulation cycle (*transaction*), false otherwise.
S'delayed(T)	Delays signal S by a time interval T.
S'event	True if the new value of a signal S is unequal to its old value (*eve* false otherwise.
S' last_active	Time since last transaction on signal S.
S' last_event	Time since last event on signal S.
S' last_value	Last value of signal S before current event.
S'quiet	True if no events occured on signal S for a time span T, false otherwise (return type: **signal**).
S' stable (T)	True if no events where scheduled on signal S for a time period T false otherwise (return type: **signal**).
S' transaction	Toggles between '0' and '1' each time a transaction occurs on signal S (return type: **signal**).

Syntax of *wait statements*:

```
[label_name:]
-- Type of identifier: signal
wait [on identifier {, ...}] -- Sensitivity list
   [until expression]        -- Condition clause
   [for time_expression];    -- Timeout
```

Combinational logic can be modeled using **wait on** statements. The execution of the process in the following example resumes every time an event occurs on any signal in its *sensitivity list*:

```
my_or: process
begin
   y <= x1 or x2;
   wait on x1, x2;
end process;

-- Equivalent to:
my_or2: process (x1, x2)
begin
   y <= x1 or x2;
end process;
```

In contrast, **wait until** statements suspend a process until the condition becomes true once again. When combining both aforementioned forms of wait statements, events on signals in the sensitivity list have a higher priority than any conditions. A timeout for maximum suspension of a process may be specified using the **for** statement. In simulation, a process can be stopped using a single **wait**; instruction.

Syntax of *process statements*:

```
-- Type of identifier: signal
[label_name:]
```

```
process [(identifier {, ...} | all)] [is]
   -- Declarations (including variables)
   {...}
begin
   -- Sequential statements
   {...}
end process [label_name];
```

A process gets activated on any event that occurs on the signals listed in its *sensitivity* list. If there is no sensitivity list, one has to use at least one **wait** statement inside the body of the process. The usage of a sensitivity list and **wait** statements is mutually exclusive. Specifying **all** in the sensitivity list of a process in VHDL-2008 is equivalent to explicitly listing all signals the process treats as inputs.

Concurrent signal assignments are signal assignment statements that reside inside an architecture body. Accordingly, conditional signal assignments inside an architecture body are called *concurrent conditional signal assignments* and selected signal assignments *concurrent selected signal assignments*. This enables *functional descriptions* of digital systems.

Example of a concurrent selected signal assignment:

```
entity mux4 is
   port (a0, a1, a2, a3 : in bit;
         sel0, sel1     : in bit;
         y              : out bit);
   constant T_PD : time := 4.5 ns;
end entity mux4;

architecture behav of mux4 is
begin
   with bit_vector'(sel0, sel1) select
      y <= a0 when "00",
           a1 when "01",
           a2 when "10",
           a3 when "11";
end architecture behav;
```

Similarly, assertions inside the body of an architecture or entity are referred to as *concurrent assertion statements*.

A *structural model* of a system can be constructed using *component instantiation statements* inside an architecture body:

```
label_name:
entity identifier [(architecture_identifier)]
   [generic map (
      [identifier =>] expression
                    | open
      {, ...}
   )]

   [port map (
        [identifier =>] signal_identifier
                        | expression
                        | open
        {, ...}
    )]
;
```

Specifying **open** in the port association list leaves the corresponding port unconnected.

2.6 Procedures and Functions

Syntax of a *procedure* declaration:

```
procedure identifier
[generic (
    -- C.f. entity declaration
)]
[[parameter] (
    -- Formal parameters
    [constant | variable | signal]
    -- Out parameters may only be read in VHDL-2008
    identifier {, ...} : [in|out|inout] typename
        [:= default_value]
    {; ...}
)] is
    -- Declarations (local to the procedure)
    {...}
begin
    -- Sequential statements
    {...}
end [procedure] [identifier];
```

The statement `[label_name:] return;` returns immediately from a procedure.

Syntax of *procedure calls*:

```
[label_name:]
procedure_identifier[(
    [parameter_identifier =>]
        expression
        | signal_identifier
        | variable_identifier
        | open
    {, ...}
)];
```

Example of a procedure:

```
entity sum is
end entity sum;

```

```
architecture behav of sum is
begin
   process is
      type real_array is array (natural range <>) of real;
      constant deviations : real_array := (1.0, 2.0, 3.0);
      variable sum_of_squares : real;

      procedure calc_sos (dev : in real_array; sos : out real) is
         variable tmp : real := 0.0;
      begin
         for i in dev'range loop
            tmp := tmp + dev(i)**2;
         end loop;

         sos := tmp;
      end procedure calc_sos;
   begin
      calc_sos(deviations, sum_of_squares);
      report "Sum of squares: " & to_string(sum_of_squares);
      wait;
   end process;
end architecture behav;
```

Syntax of *function* declarations:

```
-- Default: pure
[pure | impure]
function identifier
[generic (
   -- C.f. entity declarations
)]
[[parameter] (
   -- Default: constant
   [constant | signal]
   -- Default: in
   identifier {, ...} : [in] typename
      [:= default_value]
   {; ...}
)] return type_name is
   -- Declarations (local to the function)
   {...}
begin
   -- Sequential statements
   {...}
end [function] [identifier];
```

Pure functions do not read any variables or signals other than those specified in its parameter list; the impure keyword indicates that a function may have side effects. The statement `[label_name:] return value;` exits the function immediately returning `value`. Functions may not include any **wait** statements.

Syntax of *function calls*:

```
function_identifier [(
   [parameter_identifier =>]
      expression
      | signal_identifier
      | open
   {, ...}
)];
```

The predefined function now returns the current simulation time as a delay_length.

Example of a function:

```
entity check_edge is
end entity check_edge;

architecture behav of check_edge is
   function valid_edge(signal x : in std_ulogic) return boolean is
   begin
      if (x'last_value = '0' or x'last_value = 'L')
         and (x = '1' or x = 'H') then
         return true;
      elsif (x'last_value = '1' or x'last_value = 'H')
            and (x = '0' or x = 'L') then
         return true;
      else
         return false;
      end if;
   end function valid_edge;

   signal x : std_ulogic := 'U';
begin
   process is
   begin
      x <= '1';
      wait until x = '1';
      report "U -> 1: " & to_string(valid_edge(x));
      x <= '0';
      wait until x = '0';
      report "1 -> 0: " & to_string(valid_edge(x));
      x <= 'L';
      wait until x = 'L';
      report "0 -> L: " & to_string(valid_edge(x));
      x <= 'H';
      wait until x = 'H';
      report "L -> H: " & to_string(valid_edge(x));
      wait;
   end process;
end architecture behav;
```

Example of *operator overloading*:

```
entity overloading is
end entity overloading;

```

```
architecture behav of overloading is
   function "and" (left, right : integer) return boolean is
      variable l, r : boolean;
   begin
      l := false when left = 0 else true when left /= 0;
      r := false when right = 0 else true when right /= 0;
      return l and r;
   end function "and";

   function "or" (left, right : integer) return boolean is
      variable l, r : boolean;
   begin
      l := false when left = 0 else true when left /= 0;
      r := false when right = 0 else true when right /= 0;
      return l or r;
   end function "or";
begin
   process is
   begin
      report "0 and 1 = " & to_string(0 and 1);
      report "1 and 1 = " & to_string(1 and 1);
      report "0 or  0 = " & to_string(0 or  0);
      report "0 or  1 = " & to_string(0 or  1);
      wait;
   end process;
end architecture behav;
```

2.7 FSM

Possible state encoding formats:

Decimal	Binary	Gray	Johnson	One-Hot
0	0000	0000	00000000	0000000000000001
1	0001	0001	00000001	0000000000000010
2	0010	0011	00000011	0000000000000100
3	0011	0010	00000111	0000000000001000
4	0100	0110	00001111	0000000000010000
5	0101	0111	00011111	0000000000100000
6	0110	0101	00111111	0000000001000000
7	0111	0100	01111111	0000000010000000
8	1000	1100	11111111	0000000100000000
9	1001	1101	11111110	0000001000000000
10	1010	1111	11111100	0000010000000000
11	1011	1110	11111000	0000100000000000
12	1100	1010	11110000	0001000000000000
13	1101	1011	11100000	0010000000000000
14	1110	1001	11000000	0100000000000000
15	1111	1000	10000000	1000000000000000

FSM with separate register, next state and output logic:

```vhdl
entity fsm1 is
  port (clk, rst  : in std_logic;
        x         : in std_logic;
        y         : out std_logic);
end entity;

architecture behav of fsm1 is
  type fsm_state is (S0, S1, S2);
  signal state, next_state : fsm_state;
begin
  -- Next state logic
  process (state, x)
  begin
    case state is
      when S0 =>
        next_state <= S1;
      when S1 =>
        if x = '1' then
          next_state <= S2;
        else
          next_state <= S1;
        end if;
      when S2 =>
        next_state <= S0;
      when others =>
        next_state <= S0;
    end case;
  end process;

  -- Register
```

```vhdl
    process (clk, rst)
    begin
        if rst = '1' then
            state <= S0;
        elsif rising_edge(clk) then
            state <= next_state;
        end if;
    end process;

    -- Moore output (concurrent selected signal assignment)
    with state select y <=
        '1' when S0,
        '1' when S1,
        '0' when S2,
        '0' when others;
end architecture;
```

FSM with combined register and next state logic, separate output logic:

```vhdl
entity fsm2 is
    port (clk, rst : in std_logic;
          x        : in std_logic;
          y        : out std_logic);
end entity;

architecture behav of fsm2 is
    type fsm_state is (S0, S1, S2);
    signal state : fsm_state;
begin
    -- Register, next state logic
    process (clk, rst)
    begin
        if rst = '1' then
            state <= S0;
        elsif rising_edge(clk) then
            case state is
                when S0 =>
                    state <= S1;
                when S1 =>
                    if x = '1' then
                        state <= S2;
                    else
                        state <= S1;
                    end if;
                when S2 =>
                    state <= S0;
                when others =>
                    null;
            end case;
        end if;
    end process;

    -- Moore output (separate process)
    process (state)
    begin
        case state is
            when S0 =>
                y <= '1';
            when S1 =>
                y <= '1';
            when S2 =>
                y <= '0';
            when others =>
                y <= '0';
        end case;
    end process;
end architecture;
```

FSM with combined next state and output logic, separate register:

```
entity fsm3 is
    port (clk, rst  : in std_logic;
          x         : in std_logic;
          y         : out std_logic);
end entity;

architecture behav of fsm3 is
    type fsm_state is (S0, S1, S2);
    signal state, next_state : fsm_state;
begin
    -- Next state and output logic
    process (state, x)
    begin
        case state is
            when S0 =>
                -- Next state
                state <= S1;
                -- Moore output
                y <= '1';
            when S1 =>
                -- Next state
                if x = '1' then
                    state <= S2;
                else
                    state <= S1;
                end if;
                -- Moore output
                y <= '1';
            when S2 =>
                -- Next state
                state <= S0;
                -- Moore output
                y <= '0';
            when others =>
                state <= S0;
                y <= '0';
        end case;
    end process;
```

```

   -- Register
   process (clk, rst)
   begin
      if rst = '1' then
         state <= S0;
      elsif rising_edge(clk) then
         state <= next_state;
      end if;
   end process;
end architecture;
```

FSM with combined register, next state and output logic:

```
entity fsm4 is
   port (clk, rst : in std_logic;
         x        : in std_logic;
         y        : out std_logic);
end entity;

architecture behav of fsm4 is
begin
   process (clk, rst)
      type fsm_state is (S0, S1, S2);
      variable state : fsm_state;
   begin
      -- Register, next state logic
      if rst = '1' then
         state := S0;
      elsif rising_edge(clk) then
         case state is
            when S0 =>
               state := S1;
            when S1 =>
               if x = '1' then
                  state := S2;
               else
                  state := S1;
               end if;
            when S2 =>
               state := S0;
            when others =>
               state := S0;
         end case;
      end if;

      -- Moore output
      case state is
         when S0 =>
            y <= '1';
         when S1 =>
            y <= '1';
         when S2 =>
            y <= '0';
            when others =>
                y <= '0';
        end case;
    end process;
end architecture;
```

2.8 Miscellaneous

Syntax of *package* declarations:

```
package identifier is
    -- Declarations
    {...}
end [package] [identifier];
```

Syntax of *package bodies*:

```
package body identifier is
    -- Declarations
    {...}
end [package body] [identifier];
```

Implicit context clause in each design unit:

```
library std, work;
use std.standard.all;
```

Declaring aliases for data objects:
alias new_identifier **is** old_identifier ;

3 ED

3.1 Vector Analysis

Identities involving vector products ($\vec{a}, \vec{b}, \vec{c} \in \mathbb{R}^3$, the wedge operator $\wedge$ denotes the vector product):

$$\vec{a} \cdot (\vec{b} \wedge \vec{c}) = \vec{b} \cdot (\vec{c} \wedge \vec{a}) = \vec{c} \cdot (\vec{a} \wedge \vec{b})$$
$$\vec{a} \cdot (\vec{c} \wedge \vec{b}) = \vec{b} \cdot (\vec{a} \wedge \vec{c}) = \vec{c} \cdot (\vec{b} \wedge \vec{a})$$

$$\vec{a} \wedge (\vec{b} \wedge \vec{c}) = \vec{b}(\vec{a} \cdot \vec{c}) - \vec{c}(\vec{a} \cdot \vec{b})$$
$$(\vec{a} \wedge \vec{b}) \wedge \vec{c} = \vec{b}(\vec{a} \cdot \vec{c}) - \vec{a}(\vec{b} \cdot \vec{c})$$

Definition of the *nabla* (*del*) operator:

$$\vec{\nabla} = \begin{bmatrix} \partial/\partial x \\ \partial/\partial y \\ \partial/\partial z \end{bmatrix}$$

Gradient of a scalar field f:

$$\text{grad}(f) = \vec{\nabla} f = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \\ \partial f/\partial z \end{bmatrix}$$

Divergence of a vector field $\vec{F}$:

$$\text{div}(\vec{F}) = \vec{\nabla} \cdot \vec{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z}$$

Curl of a vector field $\vec{F}$:

$$\text{curl}(\vec{F}) = \vec{\nabla} \wedge \vec{F} = \begin{vmatrix} \hat{e}_x & \hat{e}_y & \hat{e}_z \\ \partial/\partial x & \partial/\partial y & \partial/\partial z \\ F_x & F_y & F_z \end{vmatrix}$$

Product rules for gradients (f, g: scalar fields; $\vec{F}, \vec{G}$: vector fields):

$$\vec{\nabla}(fg) = f(\vec{\nabla} g) + g(\vec{\nabla} f)$$

$$\vec{\nabla}(\vec{F} \cdot \vec{G}) = \vec{F} \wedge (\vec{\nabla} \wedge \vec{G}) + \vec{G} \wedge (\vec{\nabla} \wedge \vec{F}) + (\vec{F} \cdot \vec{\nabla})\vec{G} + (\vec{G} \cdot \vec{\nabla})\vec{F}$$

Product rules for divergences (f: scalar field; $\vec{F}, \vec{G}$: vector fields):

$$\vec{\nabla} \cdot (f\vec{G}) = f(\vec{\nabla} \cdot \vec{G}) + \vec{G} \cdot (\vec{\nabla} f)$$

$$\vec{\nabla} \cdot (\vec{F} \wedge \vec{G}) = \vec{G} \cdot (\vec{\nabla} \wedge \vec{F}) - \vec{F} \cdot (\vec{\nabla} \wedge \vec{G})$$

Product rules for curls (f: scalar field; $\vec{F}, \vec{G}$: vector fields):

$$\vec{\nabla} \wedge (f\vec{G}) = f(\vec{\nabla} \wedge \vec{G}) - \vec{G} \wedge (\vec{\nabla} f)$$

$$\vec{\nabla} \wedge (\vec{F} \wedge \vec{G}) = (\vec{G} \cdot \vec{\nabla})\vec{F} - (\vec{F} \cdot \vec{\nabla})\vec{G} + \vec{F}(\vec{\nabla} \cdot \vec{G}) - \vec{G}(\vec{\nabla} \cdot \vec{F})$$

Definition of the *Laplacian* (f: scalar field):

$$\vec{\nabla} \cdot (\vec{\nabla} f) = \vec{\nabla}^2 f = \Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$$

The curl of a gradient and the divergence of a curl always vanish:

$$\vec{\nabla} \wedge (\vec{\nabla} f) \equiv \vec{0}$$

$$\vec{\nabla} \cdot (\vec{\nabla} \wedge \vec{F}) \equiv 0$$

Curl of a curl:

$$\vec{\nabla} \wedge (\vec{\nabla} \wedge \vec{F}) = \vec{\nabla}(\vec{\nabla} \cdot \vec{F}) - \vec{\nabla}^2 \vec{F}$$

Fundamental theorem for gradients (f: scalar field):

$$\int_{\vec{a}}^{\vec{b}} (\vec{\nabla} f) \cdot \mathrm{d}\vec{r} = f(\vec{b}) - f(\vec{a})$$

Gauss's theorem ($\vec{F}$: vector field):

$$\int_V (\vec{\nabla} \cdot \vec{F}) \, \mathrm{d}V = \oint_{\partial V} \vec{F} \cdot \mathrm{d}\vec{a}$$

Stokes' theorem ($\vec{F}$: vector field):

$$\int_S (\vec{\nabla} \wedge \vec{F}) \cdot \mathrm{d}\vec{a} = \oint_{\partial S} \vec{F} \cdot \mathrm{d}\vec{r}$$

Spherical coordinates:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = r \cdot \begin{bmatrix} \sin(\vartheta) \cdot \cos(\varphi) \\ \sin(\vartheta) \cdot \sin(\varphi) \\ \cos(\vartheta) \end{bmatrix}$$

Unit vectors:

$$\begin{aligned} \hat{e}_r &= \sin(\vartheta) \cdot \cos(\varphi) \cdot \hat{e}_x + \sin(\vartheta) \cdot \sin(\varphi) \cdot \hat{e}_y + \cos(\vartheta) \cdot \hat{e}_z \\ \hat{e}_\vartheta &= \cos(\vartheta) \cdot \cos(\varphi) \cdot \hat{e}_x + \cos(\vartheta) \cdot \sin(\varphi) \cdot \hat{e}_y - \sin(\vartheta) \cdot \hat{e}_z \\ \hat{e}_\varphi &= -\sin(\varphi) \cdot \hat{e}_x + \cos(\varphi) \cdot \hat{e}_y \end{aligned}$$

Line and volume elements:

$$\begin{aligned} \mathrm{d}\vec{r} &= \mathrm{d}r \cdot \hat{e}_r + r \, \mathrm{d}\vartheta \cdot \hat{e}_\vartheta + r \sin(\vartheta) \, \mathrm{d}\varphi \cdot \hat{e}_\varphi \\ \mathrm{d}V &= r^2 \sin(\vartheta) \, \mathrm{d}r \, \mathrm{d}\vartheta \, \mathrm{d}\varphi \end{aligned}$$

Gradient:

$$\vec{\nabla} f = \frac{\partial f}{\partial r} \cdot \hat{e}_r + \frac{1}{r} \frac{\partial f}{\partial \vartheta} \cdot \hat{e}_\vartheta + \frac{1}{r \sin(\vartheta)} \frac{\partial f}{\partial \varphi} \cdot \hat{e}_\varphi$$

Divergence:

$$\vec{\nabla} \cdot \vec{F} = \frac{1}{r^2} \frac{\partial}{\partial r}(r^2 \cdot F_r) + \frac{1}{r \sin(\vartheta)} \frac{\partial}{\partial \vartheta} (\sin(\vartheta) \cdot F_\vartheta) + \frac{1}{r \sin(\vartheta)} \frac{\partial}{\partial \varphi} F_\varphi$$

Curl:

$$\begin{aligned}\vec{\nabla} \wedge \vec{F} &= \frac{1}{r\sin(\vartheta)} \cdot \left[\frac{\partial}{\partial\vartheta}\left(\sin(\vartheta)\cdot F_\varphi\right) - \frac{\partial}{\partial\varphi}F_\vartheta\right] \cdot \hat{e}_r \\ &+ \frac{1}{r} \cdot \left[\frac{1}{\sin(\vartheta)}\frac{\partial}{\partial\varphi}F_r - \frac{\partial}{\partial r}(r\cdot F_\varphi)\right] \cdot \hat{e}_\vartheta \\ &+ \frac{1}{r} \cdot \left[\frac{\partial}{\partial r}(r\cdot F_\vartheta) - \frac{\partial}{\partial\vartheta}F_r\right] \cdot \hat{e}_\varphi\end{aligned}$$

Laplacian:

$$\Delta f = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\cdot\frac{\partial f}{\partial r}\right) + \frac{1}{r^2\sin(\vartheta)}\frac{\partial}{\partial\vartheta}\left(\sin(\vartheta)\cdot\frac{\partial f}{\partial\vartheta}\right) + \frac{1}{r^2\sin^2(\vartheta)}\cdot\frac{\partial^2 f}{\partial\varphi^2}$$

Cylindrical coordinates:

$$\begin{bmatrix}x\\y\\z\end{bmatrix} = \begin{bmatrix}\varrho\cdot\cos(\varphi)\\ \varrho\cdot\sin(\varphi)\\ z\end{bmatrix}$$

Unit vectors:

$$\begin{aligned}\hat{e}_\varrho &= \cos(\varphi)\cdot\hat{e}_x + \sin(\varphi)\cdot\hat{e}_y \\ \hat{e}_\varphi &= -\sin(\varphi)\cdot\hat{e}_x + \cos(\varphi)\cdot\hat{e}_y \\ \hat{e}_z &= \hat{e}_z\end{aligned}$$

Line and volume elements:

$$\begin{aligned}\mathrm{d}\vec{r} &= \mathrm{d}\varrho\cdot\hat{e}_\varrho + \varrho\,\mathrm{d}\varphi\cdot\hat{e}_\varphi + \mathrm{d}z\cdot\hat{e}_z \\ \mathrm{d}V &= \varrho\,\mathrm{d}\varrho\,\mathrm{d}\varphi\,\mathrm{d}z\end{aligned}$$

Gradient:

$$\vec{\nabla} f = \frac{\partial f}{\partial\varrho}\cdot\hat{e}_\varrho + \frac{1}{\varrho}\frac{\partial f}{\partial\varphi}\cdot\hat{e}_\varphi + \frac{\partial f}{\partial z}\cdot\hat{e}_z$$

Divergence:

$$\vec{\nabla}\cdot\vec{F} = \frac{1}{\varrho}\frac{\partial}{\partial\varrho}(\varrho\cdot F_\varrho) + \frac{1}{\varrho}\frac{\partial}{\partial\varphi}\cdot F_\varphi + \frac{\partial}{\partial z}\cdot F_z$$

Curl:

$$\begin{aligned}\vec{\nabla}\wedge\vec{F} &= \left(\frac{1}{\varrho}\frac{\partial}{\partial\varphi}\cdot F_z - \frac{\partial}{\partial z}\cdot F_\varphi\right)\cdot\hat{e}_\varrho \\ &+ \left(\frac{\partial}{\partial z}F_\varrho - \frac{\partial}{\partial\varrho}\cdot F_z\right)\cdot\hat{e}_\varphi \\ &+ \frac{1}{\varrho}\cdot\left[\frac{\partial}{\partial\varrho}(\varrho\cdot F_\varphi) - \frac{\partial}{\partial\varphi}\cdot F_\varrho\right]\cdot\hat{e}_z\end{aligned}$$

Laplacian:

$$\Delta f = \frac{1}{\varrho}\frac{\partial}{\partial\varrho}\left(\varrho\cdot\frac{\partial f}{\partial\varrho}\right) + \frac{1}{\varrho^2}\cdot\frac{\partial^2 f}{\partial\varphi^2} + \frac{\partial^2 f}{\partial z^2}$$

Delta distribution (differentiation with respect to $\vec{r}$, $\vec{r}\,' \equiv$ const.):

$$\vec{\nabla}\cdot\left(\frac{\vec{r}-\vec{r}\,'}{|\vec{r}-\vec{r}\,'|^3}\right) = 4\pi\cdot\delta^{(3)}(\vec{r}-\vec{r}\,')$$

$$\Delta\left(\frac{1}{|\vec{r}-\vec{r}\,'|}\right) = -4\pi\cdot\delta^{(3)}(\vec{r}-\vec{r}\,')$$

Properties of *curl-less* fields:

1. $\vec{\nabla}\wedge\vec{F}\equiv\vec{0}$
2. $\int_\gamma \vec{F}\cdot d\vec{r}$ is independent of the path γ
3. $\oint_\gamma \vec{F}\cdot d\vec{r}\equiv 0 \quad \forall\gamma$
4. $\vec{F} = -\vec{\nabla}\phi$ (ϕ: scalar potential)

Properties of *divergence-less* field:

1. $\vec{\nabla}\cdot\vec{F}\equiv 0$
2. $\int_\mathcal{S} \vec{F}\cdot d\vec{a}$ is independent of the surface $\mathcal{S}$
3. $\oint_\mathcal{S} \vec{F}\cdot d\vec{a}\equiv 0 \quad \forall\mathcal{S}$
4. $\vec{F} = \vec{\nabla}\wedge\vec{A}$ ($\vec{A}$: vector potential)

3.2 Electrostatics

Coulomb's law (force on a test charge Q caused by a point charge q at location $\vec{r}\,'$, ε_0: permittivity of free space):

$$\vec{F}(\vec{r}) = \frac{qQ}{4\pi\varepsilon_0}\cdot\frac{\vec{r}-\vec{r}\,'}{|\vec{r}-\vec{r}\,'|^3}$$

Force on a test charge exerted by several point charges $q_1,\dots,q_n$:

$$\vec{F} = Q\cdot\vec{E}$$

Electric field:

$$\vec{E}(\vec{r}) = \frac{1}{4\pi\varepsilon_0}\cdot\sum_{j=1}^{n} q_j\cdot\frac{\vec{r}-\vec{r}_j}{|\vec{r}-\vec{r}_j|^3}$$

Electric field generated by the charge distribution $\varrho(\vec{r})$:

$$\vec{E}(\vec{r}) = \frac{1}{4\pi\varepsilon_0} \cdot \int \varrho(\vec{r}\,') \cdot \frac{\vec{r} - \vec{r}\,'}{|\vec{r} - \vec{r}\,'|^3} \, \mathrm{d}^3 r'$$

Electric flux through a surface $\mathcal{S}$:

$$\Phi = \int_{\mathcal{S}} \vec{E} \cdot \mathrm{d}\vec{a}$$

Gauss's law (differential form):

$$\begin{aligned}\vec{\nabla} \cdot \vec{E} &= \frac{1}{4\pi\varepsilon_0} \cdot \int \mathrm{d}^3 r' \, \varrho(\vec{r}\,') \, \vec{\nabla} \cdot \left(\frac{\vec{r} - \vec{r}\,'}{|\vec{r} - \vec{r}\,'|^3} \right) = \frac{1}{4\pi\varepsilon_0} \cdot \int \mathrm{d}^3 r' \, \varrho(\vec{r}\,') \, 4\pi \cdot \delta^{(3)}(\vec{r} - \vec{r}\,') \\ &= \frac{1}{\varepsilon_0} \cdot \varrho(\vec{r})\end{aligned}$$

Gauss's law (integral form):

$$\begin{aligned}\oint_{\partial\mathcal{V}} \vec{E} \cdot \mathrm{d}\vec{a} &= \int_{\mathcal{V}} \vec{\nabla} \cdot \vec{E} \, \mathrm{d}V = \frac{1}{\varepsilon_0} \cdot \int_{\mathcal{V}} \varrho(\vec{r}) \, \mathrm{d}V \\ &= \frac{1}{\varepsilon_0} \cdot Q_{\text{total}}\end{aligned}$$

Closed paths, curl of an electric field:

$$\oint_{\gamma} \vec{E} \cdot \mathrm{d}\vec{r} \equiv 0 \quad \forall \gamma$$

$$\vec{\nabla} \wedge \vec{E} \equiv \vec{0}$$

Definition of the electric potential ($\vec{r}_{\text{ref}}$: reference point):

$$\phi(\vec{r}) = -\int_{\vec{r}_{\text{ref}}}^{\vec{r}} \vec{E}(\vec{r}\,') \cdot \mathrm{d}\vec{r}\,' \curvearrowright \vec{E}(\vec{r}) = -\vec{\nabla}\phi(\vec{r})$$

Poisson's equation:

$$\vec{\nabla}^2 \phi = \Delta\phi = -\frac{\varrho}{\varepsilon_0}$$

Laplace's equation ($\varrho \equiv 0$):

$$\vec{\nabla}^2 \phi = \Delta\phi = 0$$

Potential of a charge distribution ($\vec{r}_{\text{ref}} \longrightarrow \infty$):

$$\phi(\vec{r}) = \frac{1}{4\pi\varepsilon_0} \cdot \int \frac{\varrho(\vec{r}\,')}{|\vec{r} - \vec{r}\,'|} \, \mathrm{d}^3 r'$$

Energy stored in an electric field:

$$W = \frac{\varepsilon_0}{2} \cdot \int_{\mathbb{R}^3} \vec{E}^2 \, \mathrm{d}V$$

Properties of an ideal conductor:

1. $\vec{E} \equiv \vec{0}$ on the inside
2. $\varrho \equiv 0$ on the inside
3. Any net charge resides on the surface.
4. $\phi(\vec{r}) \equiv \text{const.}$
5. $\vec{E} \perp$ surface

Boundary conditions at the proximity of a surface with surface charge density σ ($\hat{n}$: unit vector perpendicular to the surface):

$$\vec{E}_{\text{above}} - \vec{E}_{\text{below}} = \frac{\sigma}{\varepsilon_0} \cdot \hat{n}$$

Multipole expansion in cartesian coordinates (P_n: Legendre polynomials, $\alpha := \angle(\vec{r}, \vec{r}\,')$):

$$\phi(\vec{r}) = \frac{1}{4\pi\varepsilon_0} \cdot \sum_{n=0}^{\infty} \frac{1}{|\vec{r}|^{n+1}} \int |\vec{r}\,'|^n \, P_n(\cos(\alpha)) \, \varrho(\vec{r}\,') \, \mathrm{d}^3 r'$$

Dipole moments:

$$\vec{p} = \int \vec{r}\,' \, \varrho(\vec{r}\,') \, \mathrm{d}^3 r'$$

$$\phi^{(2)}(\vec{r}) = \frac{1}{4\pi\varepsilon_0} \cdot \frac{\vec{p} \cdot \hat{e}_r}{|\vec{r}|^2}$$

3.3 Electric fields in matter

Polarization:

$$\vec{P} = \frac{\text{dipole moment}}{\text{unit volume}}$$

Electric displacement:

$$\vec{D} = \varepsilon_0 \vec{E} + \vec{P}$$

$$\vec{\nabla} \wedge \vec{D} = \varepsilon_0 (\vec{\nabla} \wedge \vec{E}) + \vec{\nabla} \wedge \vec{P} = \vec{\nabla} \wedge \vec{P} \neq \vec{0}$$

Gauss's law (ϱ_{f}: free charge density):

$$\vec{\nabla} \cdot \vec{D} = \varrho_{\text{f}}$$

$$\oint_{\partial \mathcal{V}} \vec{D} \cdot \mathrm{d}\vec{a} = Q_{\text{total}}$$

Boundary conditions (σ_f: free surface charge density):

$$D^\perp_\text{above} - D^\perp_\text{below} = \sigma_\text{f}$$
$$D^\parallel_\text{above} - D^\parallel_\text{below} = P^\parallel_\text{above} - P^\parallel_\text{below}$$

Polarization in linear dielectrics (χ_e: electric susceptibility):

$$\vec{P} = \varepsilon_0 \chi_\text{e} \vec{E}$$

Permittivity:

$$\varepsilon = \varepsilon_0 (1 + \chi_\text{e})$$

Electric displacement:

$$\vec{D} = \varepsilon \vec{E}$$

Dielectric constant:

$$\varepsilon_\text{r} = 1 + \chi_\text{e} = \frac{\varepsilon}{\varepsilon_0}$$

3.4 Magnetostatics

Lorentz force law:

$$\vec{F} = q \cdot (\vec{E} + \vec{v} \wedge \vec{B})$$

Force on a wire with impressed current I in a magnetic field $\vec{B}$:

$$\vec{F}_\text{mag} = \int I \, (\mathrm{d}\vec{l} \wedge \vec{B})$$

Surface current density (σ: surface charge density):

$$\vec{K} = \frac{\mathrm{d}\vec{I}}{\mathrm{d}l_\perp} = \sigma \vec{v}$$

Force on a surface current in a magnetic field $\vec{B}$:

$$\vec{F}_\text{mag} = \int (\vec{K} \wedge \vec{B}) \, \mathrm{d}a$$

Volume current density (ϱ: volume charge density):

$$\vec{J} = \frac{\mathrm{d}\vec{I}}{\mathrm{d}a_\perp} = \varrho \vec{v}$$

Force on a volume current in a magnetic field $\vec{B}$:

$$\vec{F}_{\text{mag}} = \int (\vec{I} \wedge \vec{B})\, \mathrm{d}V$$

Continuity equation (local charge conservation):

$$\vec{\nabla} \cdot \vec{J} = -\frac{\partial \varrho}{\partial t}$$

Biot-Savart law (magnetic field of a steady line current):

$$\vec{B}(\vec{r}) = \frac{\mu_0}{4\pi} \cdot \int \vec{I} \wedge \frac{\vec{r} - \vec{r}'}{|\vec{r} - \vec{r}'|^3}\, \mathrm{d}l' = \frac{\mu_0}{4\pi} I \cdot \int \mathrm{d}\vec{l}' \wedge \frac{\vec{r} - \vec{r}'}{|\vec{r} - \vec{r}'|^3}$$

Ampere's law (differential form):

$$\vec{\nabla} \wedge \vec{B} = \mu_0 \cdot \vec{J}$$

Ampere's law (integral form, I_{total}: total enclosed current):

$$\oint \vec{B} \cdot \mathrm{d}\vec{l} = \mu_0 \cdot I_{\text{total}}$$

Definition of the vector potential:

$$\vec{B} = \vec{\nabla} \wedge \vec{A}$$
$$\vec{\nabla} \cdot \vec{A} \equiv 0$$

Ampere's law:

$$\vec{\nabla}^2 \vec{A} = -\mu_0 \cdot \vec{J}$$

General solution ($\vec{J}(|\vec{r}| \longrightarrow \infty) \longrightarrow \vec{0}$):

$$\vec{A}(\vec{r}) = \frac{\mu_0}{4\pi} \cdot \int \frac{\vec{J}(\vec{r}')}{|\vec{r} - \vec{r}'|}\, \mathrm{d}^3 r'$$

Boundary conditions at a surface current ($\hat{n}$: unit vector perpendicular to the surface):

$$\vec{B}_{\text{above}} - \vec{B}_{\text{below}} = \mu_0 \cdot \vec{K} \wedge \hat{n}$$

Magnetic dipole moment:

$$\vec{m} = I \int \mathrm{d}\vec{a} = I\, \vec{a}$$

Vector potential of a magnetic dipole:

$$\vec{A}^{(2)}(\vec{r}) = \frac{\mu_0}{4\pi} \cdot \frac{\vec{m} \wedge \hat{e}_r}{|\vec{r}|^2}$$

3.5 Magnetic Fields in Matter

Magnetization:

$$\vec{M} = \frac{\text{magnetic dipole moment}}{\text{unit volume}}$$

Definition of the magnetic field:

$$\vec{H} = \frac{1}{\mu_0} \cdot \vec{B} - \vec{M}$$

Ampere's law ($\vec{J}_\mathrm{f}$: free current, I_total: total free current):

$$\vec{\nabla} \wedge \vec{H} = \vec{J}_\mathrm{f}$$

$$\oint \vec{H} \cdot \mathrm{d}\vec{l} = I_\mathrm{total}$$

Divergence of the magnetic field:

$$\vec{\nabla} \cdot \vec{H} = -\vec{\nabla} \cdot \vec{M}$$

Boundary conditions at a surface current ($\hat{n}$: unit vector perpendicular to the surface, $\vec{K}_\mathrm{f}$: free surface current):

$$\vec{H}^{\parallel}_\mathrm{above} - \vec{H}^{\parallel}_\mathrm{below} = \vec{K}_\mathrm{f} \wedge \hat{n}$$

Magnetization in linear matter (χ_m: magnetic susceptibility):

$$\vec{M} = \chi_\mathrm{m} \cdot \vec{H}$$

Magnetic flux density in linear matter (μ: permeability, μ_r: relative permeability, μ_0: permeability of free space):

$$\vec{B} = \mu_0 \cdot (\vec{H} + \vec{M}) = \mu_0 \cdot (1 + \chi_\mathrm{m}) \cdot \vec{H} = \mu_0\, \mu_\mathrm{r} \cdot \vec{H} = \mu \cdot \vec{H}$$

3.6 Electrodynamics

Ohm's law (σ: conductivity, $\varrho = 1/\sigma$: resistivity):

$$\vec{J} = \sigma \vec{E}$$

Resistance:

$$R = \frac{V}{I}$$

Electromotive force (emf, $\vec{f}_\mathrm{s}$: source):

$$\mathcal{E} = \oint \vec{f}_\mathrm{s} \cdot \mathrm{d}\vec{l}$$

Magnetic flux through a surface $\mathcal{S}$:

$$\Phi = \int_{\mathcal{S}} \vec{B} \cdot \mathrm{d}\vec{a}$$

Flux rule for motional emf:

$$\mathcal{E} = -\frac{\mathrm{d}\Phi}{\mathrm{d}t}$$

Faraday's law (differential form):

$$\vec{\nabla} \wedge \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

Mutual inductance of two loops (Φ_1, I_1: flux and current through loop one; Φ_2, I_2: flux and current through loop two):

$$\Phi_2 = M_{21}\, I_1$$

Neumann formula:

$$M_{21} = \frac{\mu_0}{4\pi} \cdot \oint \oint \frac{\mathrm{d}\vec{l}_1 \cdot \mathrm{d}\vec{l}_2}{|\vec{r}_1 - \vec{r}_2|}$$
$$\curvearrowright\ M_{21} = M_{12}$$

Self inductance:

$$\Phi = L\, I$$

Energy of a magnetic field:

$$W = \int_{\mathbb{R}^3} \vec{B}^2\, \mathrm{d}V$$

Maxwell's equations:

$$\begin{aligned}
\vec{\nabla} \cdot \vec{E} &= \frac{1}{\varepsilon_0} \cdot \varrho \\
\vec{\nabla} \cdot \vec{B} &= 0 \\
\vec{\nabla} \wedge \vec{E} &= -\frac{\partial \vec{B}}{\partial t} \\
\vec{\nabla} \wedge \vec{B} &= \mu_0 \cdot \vec{J} + \mu_0\, \varepsilon_0 \cdot \frac{\partial \vec{E}}{\partial t}
\end{aligned}$$

Maxwell's equations in matter:

$$\begin{aligned}
\vec{\nabla} \cdot \vec{D} &= \varrho_{\text{free}} \\
\vec{\nabla} \cdot \vec{B} &= 0 \\
\vec{\nabla} \wedge \vec{E} &= -\frac{\partial \vec{B}}{\partial t} \\
\vec{\nabla} \wedge \vec{H} &= \vec{J}_{\text{free}} + \frac{\partial \vec{D}}{\partial t}
\end{aligned}$$

General boundary conditions at a surface ($\hat{n}$: unit vector perpendicular to the surface, σ_f: free surface charge density, $\vec{K}_\text{f}$: free surface current density):

$$\begin{aligned} D_1^\perp - D_2^\perp &= \sigma_\text{f} \\ B_1^\perp - B_2^\perp &= 0 \\ \vec{E}_1^\parallel - \vec{E}_2^\parallel &= \vec{0} \\ \vec{H}_1^\parallel - \vec{H}_2^\parallel &= \vec{K}_\text{f} \wedge \hat{n} \end{aligned}$$

General boundary conditions at a surface in linear matter:

$$\begin{aligned} \varepsilon_1 \cdot \vec{E}_1^\perp - \varepsilon_2 \cdot \vec{E}_2^\perp &= \sigma_\text{f} \\ B_1^\perp - B_2^\perp &= 0 \\ \vec{E}_1^\parallel - \vec{E}_2^\parallel &= \vec{0} \\ \frac{1}{\mu_1} \cdot \vec{B}_1^\parallel - \frac{1}{\mu_2} \cdot \vec{B}_2^\parallel &= \vec{K}_\text{f} \wedge \hat{n} \end{aligned}$$

Energy density in an electromagnetic field:

$$u = \frac{1}{2} \left(\varepsilon_0 \cdot \vec{E}^2 + \frac{1}{\mu_0} \cdot \vec{B}^2 \right)$$

Poynting vector (energy flux density):

$$\vec{S} = \frac{1}{\mu_0} (\vec{E} \wedge \vec{B})$$

Poynting's theorem:

$$\frac{\mathrm{d}W}{\mathrm{d}t} = -\frac{\mathrm{d}}{\mathrm{d}t} \int_\mathcal{V} u \,\mathrm{d}V - \oint_{\partial\mathcal{V}} \vec{S} \cdot \mathrm{d}\vec{a}$$

General wave equation:

$$\vec{\nabla}^2 f = \frac{1}{v_\text{ph}^2} \cdot \frac{\partial^2 f}{\partial t^2}$$

Maxwell's equations in a vacuum:

$$\begin{aligned} \vec{\nabla} \cdot \vec{E} &= 0 \\ \vec{\nabla} \cdot \vec{B} &= 0 \\ \vec{\nabla} \wedge \vec{E} &= -\frac{\partial \vec{B}}{\partial t} \\ \vec{\nabla} \wedge \vec{B} &= \mu_0 \, \varepsilon_0 \cdot \frac{\partial \vec{E}}{\partial t} \end{aligned}$$

Derivation of the wave equations for $\vec{E}$ and $\vec{B}$:

$$\vec{\nabla} \wedge (\vec{\nabla} \wedge \vec{E}) = \vec{\nabla}\underbrace{(\vec{\nabla} \cdot \vec{E})}_{=0} - \vec{\nabla}^2 \vec{E} = -\frac{\partial}{\partial t}(\vec{\nabla} \wedge \vec{B}) = -\mu_0\, \varepsilon_0 \cdot \frac{\partial^2 \vec{E}}{\partial t^2}$$

$$\curvearrowright \quad \vec{\nabla}^2 \vec{E} = \mu_0\, \varepsilon_0 \cdot \frac{\partial^2 \vec{E}}{\partial t^2}$$

$$\vec{\nabla} \wedge (\vec{\nabla} \wedge \vec{B}) = \vec{\nabla}\underbrace{(\vec{\nabla} \cdot \vec{B})}_{=0} - \vec{\nabla}^2 \vec{B} = \mu_0 \varepsilon_0 \cdot \frac{\partial}{\partial t}(\vec{\nabla} \wedge \vec{E}) = -\mu_0\, \varepsilon_0 \cdot \frac{\partial^2 \vec{B}}{\partial t^2}$$

$$\curvearrowright \quad \vec{\nabla}^2 \vec{B} = \mu_0\, \varepsilon_0 \cdot \frac{\partial^2 \vec{B}}{\partial t^2}$$

Speed of light in a vacuum:

$$c_0 = \frac{1}{\sqrt{\mu_0\, \varepsilon_0}}$$

(Complex) plane waves ($\vec{k}$: wave vector, $|\vec{k}| = 2\pi/\lambda$: wave number, $\hat{n}$: polarization vector, $\omega = c \cdot k$):

$$\underline{\vec{E}}(\vec{r}, t) = \underline{E_0} \cdot \mathrm{e}^{\mathrm{i}(\vec{k}\cdot\vec{r} - \omega\, t)} \cdot \hat{n}$$
$$\underline{\vec{B}}(\vec{r}, t) = \frac{1}{c} \cdot \underline{E_0} \cdot \mathrm{e}^{\mathrm{i}(\vec{k}\cdot\vec{r} - \omega\, t)} \cdot (\hat{e}_k \wedge \hat{n}) = \frac{1}{c} \cdot (\hat{e}_k \wedge \underline{\vec{E}})$$

Maxwell’s equations in linear matter in absence of free charge and current:

$$\vec{\nabla} \cdot \vec{E} = 0$$
$$\vec{\nabla} \cdot \vec{B} = 0$$
$$\vec{\nabla} \wedge \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$
$$\vec{\nabla} \wedge \vec{B} = \mu\, \varepsilon \cdot \frac{\partial \vec{E}}{\partial t}$$

Speed of light in linear matter:

$$c = \frac{1}{\sqrt{\mu\, \varepsilon}} = \frac{c_0}{n}$$

Index of refraction:

$$n = \sqrt{\mu_r\, \varepsilon_r}$$

Boundary conditions on the inside of a wave guide:

$$\vec{E}^{\parallel} = \vec{0}$$
$$B^{\perp} = 0$$

Plane electromagnetic wave travelling in z-direction:

$$\underline{\vec{E}} = \underline{\vec{E}_0} \cdot \mathrm{e}^{\mathrm{i}(k\,z-\omega\,t)}$$
$$\underline{\vec{B}} = \underline{\vec{B}_0} \cdot \mathrm{e}^{\mathrm{i}(k\,z-\omega\,t)}$$

Complex amplitudes:

$$\underline{\vec{E}_0} = \begin{bmatrix} E_{0,x} \\ E_{0,y} \\ E_{0,z} \end{bmatrix}; \quad \underline{\vec{B}_0} = \begin{bmatrix} B_{0,x} \\ B_{0,y} \\ B_{0,z} \end{bmatrix}$$

Maxwell's equations on the inside:

$$\vec{\nabla} \cdot \vec{E} = 0$$
$$\vec{\nabla} \cdot \vec{B} = 0$$
$$\vec{\nabla} \wedge \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$
$$\vec{\nabla} \wedge \vec{B} = \frac{1}{c^2} \cdot \frac{\partial \vec{E}}{\partial t}$$

4 QM

Schroedinger equation in one dimension:

$$\mathrm{i}\hbar \frac{\partial \Psi}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \Psi}{\partial x^2} + V \cdot \Psi$$

Statistical interpretation:

$$P(a \leq x \leq b; t) = \int_a^b |\Psi(x,t)|^2 \, \mathrm{d}x$$

Expectation value (ϱ: probability density):

$$\langle f(x) \rangle = \int_{-\infty}^{\infty} f(x) \cdot \varrho(x) \, \mathrm{d}x$$

Variance:

$$\sigma^2 \equiv \langle (x - \langle x \rangle)^2 \rangle$$
$$\sigma^2 = \langle x^2 \rangle - \langle x \rangle^2$$

Normalization requirement:

$$\int_{-\infty}^{\infty} |\Psi(x,t)|^2 \, \mathrm{d}x = 1$$

Uncertainty principle:

$$\sigma_x \, \sigma_p \geq \frac{\hbar}{2}$$

Hamiltonian ($p \longrightarrow \frac{\hbar}{\mathrm{i}} \frac{\partial}{\partial x}$):

$$H = \frac{p^2}{2m} + V(x) = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x)$$

Time-independent Schroedinger equation:

$$H\psi(x) = E\psi(x)$$

General solution of the separable time-dependent Schroedinger equation:

$$\Psi(x,t) = \sum_{n=1}^{\infty} c_n \cdot \psi_n(x) \cdot \mathrm{e}^{-\mathrm{i}E_n \cdot t/\hbar}$$

1D infinite square well:

$$V(x) = \begin{cases} 0, & 0 \leq x \leq a \\ \infty, & \text{otherwise} \end{cases}$$

$$E_n = \frac{n^2 \pi^2 \hbar^2}{2ma^2}, \quad n = 1, 2, \ldots$$

$$\psi_n = \sqrt{\frac{2}{a}} \cdot \sin\left(\frac{n\pi}{a} \cdot x\right)$$

Hermite polynomials (Rodrigues formula):

$$H_n(\xi) = (-1)^n \cdot \mathrm{e}^{\xi^2} \cdot \left(\frac{\mathrm{d}}{\mathrm{d}\xi}\right)^n \mathrm{e}^{-\xi^2}$$

1D harmonic oscillator ($\xi \equiv \sqrt{\frac{m\omega}{\hbar}} \cdot x$, H_n: Hermite polynomials):

$$V(x) = \frac{1}{2} m\omega^2 x^2$$

$$E_n = \left(n + \frac{1}{2}\right) \cdot \hbar\omega \quad n = 0, 1, 2, \ldots$$

$$\psi_n(x) = \left(\frac{m\omega}{\pi\hbar}\right)^{1/4} \cdot \frac{1}{\sqrt{2^n n!}} \cdot H_n(\xi) \cdot \mathrm{e}^{-\xi^2/2}$$

Free particle ($k \equiv \pm\frac{\sqrt{2mE}}{\hbar}$, dispersion relation: $\omega = \frac{\hbar k^2}{2m}$):

$$\Psi(x,t) = \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{\infty} \phi(k) \cdot \mathrm{e}^{\mathrm{i}\left(kx - \frac{\hbar k^2}{2m} \cdot t\right)} \, \mathrm{d}k = \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{\infty} \phi(k) \cdot \mathrm{e}^{\mathrm{i}(kx - \omega t)}$$

$$\phi(k) = \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{\infty} \Psi(x, 0) \cdot \mathrm{e}^{-\mathrm{i}kx} \, \mathrm{d}x$$

Group velocity:

$$v_\mathrm{g} = \frac{\mathrm{d}\omega}{\mathrm{d}k}$$

Phase velocity:

$$v_\mathrm{ph} = \frac{\omega}{k}$$

Definition of bound and scattering states ($V(x \to \infty) = 0$):

- $E < 0$: bound state
- $E > 0$: scattering state

Delta function potential:

$$V(x) = -\alpha\delta(x)$$

Bound state:

$$\psi(x) = \frac{\sqrt{m\alpha}}{\hbar} \cdot \mathrm{e}^{-m\alpha|x|/\hbar^2}$$
$$E = -\frac{m\alpha^2}{2\hbar^2}$$

Inner product:

$$\langle\alpha|\beta\rangle = a_1^* b_1 + a_2^* b_2 + \cdots + a_N^* b_N$$
$$\langle\beta|\alpha\rangle = \langle\alpha|\beta\rangle^*$$
$$\langle f|g\rangle = \int_a^b f(x)^* g(x)\,\mathrm{d}x$$

Schwarz inequality:

$$|\langle f|g\rangle| \leq \sqrt{\langle f|f\rangle \cdot \langle g|g\rangle}$$

Orthonormal set of functions $\{f_n\}$:

$$\langle f_m|f_n\rangle = \delta_{mn}$$

Complete set of functions $\{f_n\}$:

$$f(x) = \sum_{n=1}^{\infty} c_n \cdot f_n(x)$$
$$c_n = \langle f_n|f\rangle$$

Expectation value of an observable:

$$\langle Q\rangle = \int \psi^* \hat{Q}\psi\,\mathrm{d}x = \langle\psi|\hat{Q}|\psi\rangle$$

Hermitian operators:

$$\langle f|\hat{Q}f\rangle = \langle \hat{Q}f|f\rangle$$

Properties of eigenfunctions of hermitian operators $\hat{Q}$ with discrete spectra:

1. $\hat{Q}f = qf \curvearrowright q \in \mathbb{R}$
2. $\hat{Q}f = qf,\ \hat{Q}g = q'g,\ q \neq q' \curvearrowright \langle f|g\rangle = 0$

Momentum space wave function:

$$\Phi(p,t) = \frac{1}{\sqrt{2\pi\hbar}} \cdot \int_{-\infty}^{\infty} \mathrm{e}^{-\mathrm{i}px/\hbar} \cdot \Psi(x,t)\,\mathrm{d}x$$
$$\Psi(x,t) = \frac{1}{\sqrt{2\pi\hbar}} \cdot \int_{-\infty}^{\infty} \mathrm{e}^{\mathrm{i}px/\hbar} \cdot \Phi(p,t)\,\mathrm{d}p$$

Commutator of two operators $\hat{A}, \hat{B}$:

$$[\hat{A},\hat{B}] := \hat{A}\hat{B} - \hat{B}\hat{A}$$

Anticommutator of two operators $\hat{A}, \hat{B}$:

$$\{\hat{A},\hat{B}\} := \hat{A}\hat{B} + \hat{B}\hat{A}$$

Generalized uncertainty principle (observable A: $\sigma_A^2 = \langle \left(\hat{A} - \langle A\rangle\right)\Psi|\left(\hat{A} - \langle A\rangle\right)\Psi\rangle$, observable B: $\sigma_B^2 = \langle \left(\hat{B} - \langle B\rangle\right)\Psi|\left(\hat{B} - \langle B\rangle\right)\Psi\rangle$):

$$\sigma_A^2\sigma_B^2 \geq \left(\frac{1}{2\mathrm{i}}\langle[\hat{A},\hat{B}]\rangle\right)^2$$

Time dependence of an expectation value (Q: observable, H: Hamiltonian):

$$\frac{\mathrm{d}}{\mathrm{d}t}\langle Q\rangle = \frac{\mathrm{i}}{\hbar}\langle[\hat{H},\hat{Q}]\rangle + \langle\frac{\partial\hat{Q}}{\partial t}\rangle$$

Schroedinger equation in three dimensions ($\vec{p} \to \frac{\hbar}{\mathrm{i}}\vec{\nabla}$):

$$\mathrm{i}\hbar\frac{\partial\Psi}{\partial t} = -\frac{\hbar^2}{2m}\Delta\Psi + V\Psi$$

Normalization requirement:

$$\int_{\mathbb{R}^3} |\Psi|^2\,\mathrm{d}V = 1$$

Probability current and conservation of probability:

$$\vec{j} = \frac{i\hbar}{2m} \cdot (\Psi \vec{\nabla} \Psi^* - \Psi^* \vec{\nabla} \Psi)$$
$$\vec{\nabla} \cdot \vec{j} = -\frac{\partial}{\partial t} |\Psi|^2$$

Time-independent Schroedinger equation:

$$-\frac{\hbar^2}{2m} \Delta \psi + V \psi = E \psi$$

General solution of the separable time-dependent Schroedinger equation:

$$\Psi(\vec{r}, t) = \sum_{n=1}^{\infty} c_n \psi_n(\vec{r}) \cdot e^{-iE_n \cdot t/\hbar}$$

Canonical commutation relations ($i, j \in \{1, 2, 3\}$):

$$[r_i, p_j] = i\hbar \delta_{ij}$$
$$[r_i, r_j] = [p_i, p_j] = 0$$

Legendre polynomials (Rodrigues formula):

$$P_l(x) = \frac{1}{2^l l!} \cdot \left(\frac{d}{dx} \right)^l \cdot (x^2 - 1)^l$$

Associated Legendre functions:

$$P_l^m(x) = (1 - x^2)^{|m|/2} \cdot \left(\frac{d}{dx} \right)^{|m|} \cdot P_l(x)$$

Spherical harmonics (normalized angular wave function):

$$\text{Azimuthal quantum number: } l = 0, 1, 2, \ldots$$
$$\text{Magnetic quantum number: } m = -l,\ -l+1,\ \ldots,\ l-1,\ l$$
$$Y_l^m(\vartheta, \varphi) = \sqrt{\frac{2l+1}{4\pi} \cdot \frac{(l-|m|)!}{(l+|m|)!}} \cdot e^{im\varphi} \cdot P_l^m(\cos(\vartheta)) \cdot \begin{cases} (-1)^m, & m \geq 0 \\ 1, & m < 0 \end{cases}$$

Orthonormality:

$$\int_0^{2\pi} d\varphi \int_0^{\pi} d\vartheta\ \sin(\vartheta)\, Y_l^m(\vartheta, \varphi)^* \cdot Y_{l'}^{m'}(\vartheta, \varphi) = \delta_{ll'} \delta_{mm'}$$

Radial equation ($\psi(r,\vartheta,\varphi) = R(r)\cdot Y(\vartheta,\varphi)$, $u(r) \equiv r\cdot R(r)$):

$$-\frac{\hbar^2}{2m}\cdot\frac{\mathrm{d}^2u}{\mathrm{d}r^2} + \underbrace{\left[V + \frac{\hbar^2}{2m}\cdot\frac{l(l+1)}{r^2}\right]}_{\text{effective potential}}\cdot u = E\cdot u$$

Radial wave equation for the hydrogen atom:

$$-\frac{\hbar^2}{2m_\mathrm{e}}\cdot\frac{\mathrm{d}^2u}{\mathrm{d}r^2} + \left[-\frac{e^2}{4\pi\varepsilon_0}\cdot\frac{1}{r} + \frac{\hbar^2}{2}\cdot\frac{l(l+1)}{r^2}\right]\cdot u = E\cdot u$$

Laguerre polynomial:

$$L_q(x) = \mathrm{e}^x\cdot\left(\frac{\mathrm{d}}{\mathrm{d}x}\right)^q(\mathrm{e}^{-x}\cdot x^q)$$

Associated Laguerre polynomial:

$$L^p_{q-p}(x) = (-1)^p\cdot\left(\frac{\mathrm{d}}{\mathrm{d}x}\right)^p L_q(x)$$

Energies (degeneracy of E_n: n^2):

$$E_n = -\left[\frac{m_\mathrm{e}}{2\hbar^2}\cdot\left(\frac{e^2}{4\pi\varepsilon_0}\right)\right]\cdot\frac{1}{n^2}, \quad n = 1,2,\dots$$

Bohr radius:

$$a_\mathrm{B} = \frac{4\pi\varepsilon_0\hbar^2}{m_\mathrm{e}e^2}$$

Hydrogen wave function:

$$\psi_{nlm}(r,\vartheta,\varphi) = \sqrt{\left(\frac{2}{na}\right)^3\cdot\frac{(n-l-1)!}{2n\,[(n+l)!]^3}}\cdot\mathrm{e}^{-r/na}\cdot\left(\frac{2r}{na}\right)^l\cdot\left[L^{2l+1}_{n-l-1}(2r/na)\right]\cdot Y^m_l(\vartheta,\varphi)$$
$$n = 1,2,3,\dots;\ l = 0,1,2,\dots,n-1;\ m = -l,-l+1,\dots,l-1,l$$

Angular momentum operator:

$$\hat{\vec{L}} = \hat{\vec{r}}\wedge\left(\frac{\mathrm{i}}{\hbar}\vec{\nabla}\right)$$

Commutation relations ($L^2 \equiv L_x^2 + L_y^2 + L_z^2$):

$$[L_x,L_y] = \mathrm{i}\hbar L_z;\quad [L_y,L_z] = \mathrm{i}\hbar L_x;\quad [L_z,L_x] = \mathrm{i}\hbar L_y$$
$$[L^2,\vec{L}] = 0$$

Eigenvalues (Y_l^m: spherical harmonics):

$$L^2\, Y_l^m = \hbar^2 l(l+1)\, Y_l^m; \quad L_z\, Y_l^m = \hbar m\, Y_l^m$$
$$l = 0,\ 1/2,\ 1,\ 3/2,\ \ldots; \quad m = -l, -l+1, \ldots, l-1, l$$

Commutation relations for spin:

$$[S_x, S_y] = \mathrm{i}\hbar S_z; \quad [S_y, S_z] = \mathrm{i}\hbar S_x; \quad [S_z, S_x] = \mathrm{i}\hbar S_y$$

Eigenvectors and eigenvalues of S^2 and S_z:

$$S^2\, |s\, m\rangle = \hbar^2 s(s+1)\, |s\, m\rangle\,; \quad S_z\, |s\, m\rangle = \hbar m\, |s\, m\rangle$$
$$s = 0,\ 1/2,\ 1,\ 3/2,\ \ldots; \quad m = -s, -s+1, \ldots, s-1, s$$

Eigenstates of spin 1/2 ($s = 1/2$):

- Spin up: $|\frac{1}{2}\ \frac{1}{2}\rangle = |\frac{1}{2}\ \uparrow\rangle$
- Spin down: $|\frac{1}{2}\ \left(-\frac{1}{2}\right)\rangle = |\frac{1}{2}\ \downarrow\rangle$

Spinor:

$$\chi = \begin{bmatrix} a \\ b \end{bmatrix} = a\chi_+ + b\chi_-$$
$$\chi_+ = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ \chi_- = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$|a|^2 + |b|^2 = 1$$

Pauli matrices:

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_y = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix}, \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Spin operator ($\vec{\sigma} = [\sigma_x, \sigma_y, \sigma_z]$):

$$\vec{S} = \frac{\hbar}{2} \cdot \vec{\sigma}$$

Hamiltonian of a charged particle at rest in a magnetic field (γ: gyromagnetic ratio):

$$H = -\gamma \vec{B} \cdot \vec{S}$$

Triplet states ($s = 1$):

1. $|1\, 1\rangle = \uparrow\uparrow$
2. $|1\, 0\rangle = \frac{1}{\sqrt{2}}(\uparrow\downarrow + \downarrow\uparrow)$

3. $|1\,(-1)\rangle = \downarrow\downarrow$

Singlet state ($s = 0$):

$$|0\,0\rangle = \frac{1}{\sqrt{2}}(\uparrow\downarrow - \downarrow\uparrow)$$

Addition of spins s_1 and s_2:

$$s = (s_1 + s_2),\ (s_1 + s_2 - 1),\ (s_1 + s_2 - 2), \ldots, |s_1 - s_2|$$

Combined states ($C^{s_1 s_2 s}_{m_1 m_2 m}$: Clebsch-Gordan coefficients):

$$|s\,m\rangle = \sum_{m_1+m_2=m} C^{s_1 s_2 s}_{m_1 m_2 m} \cdot |s_1\,m_1\rangle\ |s_2\,m_2\rangle$$

$$|s_1\,m_1\rangle\ |s_2\,m_2\rangle = \sum_s C^{s_1 s_2 s}_{m_1 m_2 m} \cdot |s\,m\rangle$$

Symmetrization requirement ("+": bosons, "−": fermions):

$$\psi(\vec{r}_1, \vec{r}_2) = \pm\psi(\vec{r}_2, \vec{r}_1)$$

5 CM

Drude's model

Avogadro's number:

$$N_\mathrm{A} = 6.022 \cdot 10^{23}\,\frac{1}{\mathrm{mol}}$$

Mass density:

$$\varrho_\mathrm{m} = \frac{m}{V}, \quad [\varrho_\mathrm{m}] = \frac{\mathrm{g}}{\mathrm{cm^3}}$$

Moles per cm³ (M: molar mass, $[M] = \mathrm{g/mol}$):

$$\frac{\varrho_\mathrm{m}}{M}$$

Conduction electron density (Z: number of valence electrons):

$$n = \frac{N}{V} = N_\mathrm{A} \cdot Z \cdot \frac{\varrho_\mathrm{m}}{M}, \quad [n] = \frac{1}{\mathrm{cm^3}}$$

Volume per conduction electron (modeled as a sphere):

$$\frac{V}{N} = \frac{1}{n} = \frac{4\pi}{3} \cdot r_\mathrm{s}^3 \curvearrowright r_\mathrm{s} = \left(\frac{3}{4\pi} \cdot \frac{1}{n}\right)^{1/3}$$

Assumptions:

- no e^-–e^- interactions: *independent electron* approximation
- no e^-–ion interaction: *free electron* approximation
- probability of a collision per unit time: $1/\tau$ (τ: *relaxation time*)

Resistivity ($\vec{E}$: electric field, $[E] = \text{V}/\text{m}$, $\vec{j}$: current density, $[j] = \text{A}/\text{m}^2$):

$$\vec{E} = \varrho \cdot \vec{j}, \quad [\varrho] = \Omega \cdot \text{m}$$

Conductivity:

$$\sigma \equiv \frac{1}{\varrho}, \quad [\sigma] = \frac{\text{S}}{\text{m}} \equiv \frac{\text{mho}}{\text{m}} \equiv \frac{\mho}{\text{m}}$$

Current density of n electrons per unit volume moving with velocity $\vec{v}$:

$$\vec{j} = -ne\vec{v}$$

Average electron velocity:

$$\vec{v}_{\text{average}} = -\frac{1}{m_\text{e}} \cdot e\tau \cdot \vec{E} \curvearrowright \vec{j} = \frac{ne^2\tau}{m_\text{e}} \cdot \vec{E}$$

Current density at time t:

$$\vec{j} = -ne \cdot \frac{\vec{p}(t)}{m_\text{e}}$$

Probability of a collision before $t + \text{d}t$:

$$\frac{\text{d}t}{\tau}$$

Probability of no collision before $t + \text{d}t$:

$$1 - \frac{\text{d}t}{\tau}$$

Additional momentum due to external force $\vec{F}(t)$:

$$\text{d}\vec{p} = \vec{F}(t)\,\text{d}t + O\left((\text{d}t)^2\right)$$

Contribution of electrons without any collision during $t + \text{d}t$ to the average momentum (correction due to electrons with a collision before $t + \text{d}t$: $O\left((\text{d}t)^2\right)$):

$$\vec{p}(t + \text{d}t) = \left(1 - \frac{\text{d}t}{\tau}\right) \cdot \left[\vec{p}(t) + \vec{F}(t)\,\text{d}t + O\left((\text{d}t)^2\right)\right]$$

$$\frac{\vec{p}(t + \text{d}t) - \vec{p}(t)}{\text{d}t} = -\frac{1}{\tau} \cdot \vec{p}(t) + \vec{F}(t)$$

$$\curvearrowright \dot{\vec{p}}(t) = -\frac{1}{\tau} \cdot \vec{p}(t) + \vec{F}(t) \quad \text{(e.o.m)}$$

AC electrical conductivity

$$\vec{E}(t) = \Re\left\{\vec{E}(\omega)\cdot \mathrm{e}^{-\mathrm{i}\omega t}\right\}$$

e.o.m:

$$\dot{\vec{p}} = -\frac{1}{\tau}\cdot\vec{p} - e\cdot\vec{E}$$

Steady state solution:

$$\vec{p}(t) = \Re\left\{\vec{p}(\omega)\cdot \mathrm{e}^{-\mathrm{i}\omega t}\right\}$$

$$\curvearrowright\ -\mathrm{i}\omega\cdot\vec{p}(\omega) = -\frac{1}{\tau}\cdot\vec{p}(\omega) - e\cdot\vec{E}(\omega) \qquad \curvearrowright\ \vec{p}(\omega) = -\frac{e\cdot\vec{E}}{{}^1\!/\!\tau - \mathrm{i}\omega}$$

$$\vec{j}(t) = \Re\left\{\vec{j}(\omega)\cdot \mathrm{e}^{-\mathrm{i}\omega t}\right\} = \Re\left\{-\frac{ne}{m_\mathrm{e}}\cdot\vec{p}(\omega)\cdot \mathrm{e}^{-\mathrm{i}\omega t}\right\}$$

$$\curvearrowright\ \vec{j}(\omega) = -\frac{ne}{m_\mathrm{e}}\cdot\vec{p}(\omega) = -\frac{ne}{m_\mathrm{e}}\cdot\left(-\frac{e\cdot\vec{E}(\omega)}{{}^1\!/\!\tau - \mathrm{i}\omega}\right) = \frac{ne^2}{m_\mathrm{e}}\cdot\frac{\vec{E}(\omega)}{{}^1\!/\!\tau - \mathrm{i}\omega}$$

$$\curvearrowright\ \vec{j}(\omega) = \sigma(\omega)\cdot\vec{E}(\omega),\quad \sigma(\omega) = \frac{\sigma_\mathrm{DC}}{1-\mathrm{i}\omega\tau},\quad \sigma_\mathrm{DC} = \frac{ne^2\tau}{m_\mathrm{e}}$$

$\lambda \gg$ electronic mean free path:

$$\curvearrowright\ \vec{j}(\vec{r},\omega) = \sigma(\omega)\cdot\vec{E}(\vec{r},\omega)$$

Maxwell's equations (cgs units):

$$\vec{\nabla}\cdot\vec{E} = 0$$
$$\vec{\nabla}\cdot\vec{H} = 0$$
$$\vec{\nabla}\wedge\vec{E} = -\frac{1}{c}\cdot\frac{\partial\vec{H}}{\partial t}$$
$$\vec{\nabla}\wedge\vec{H} = \frac{1}{c}\cdot\left[4\pi\vec{j} + \frac{\partial\vec{E}}{\partial t}\right]$$

$$\vec{\nabla}\wedge(\vec{\nabla}\wedge\vec{E}) = \vec{\nabla}(\vec{\nabla}\cdot\vec{E}) = \vec{\nabla}(\underbrace{\vec{\nabla}\cdot\vec{E}}_{=0}) - \vec{\nabla}^2\vec{E} = -\frac{1}{c}\cdot\frac{\partial}{\partial t}(\vec{\nabla}\wedge\vec{H}) = -\frac{1}{c}(-\mathrm{i}\omega)\cdot(\vec{\nabla}\wedge\vec{H})$$

$$= \frac{\mathrm{i}\omega}{c}\cdot(\vec{\nabla}\wedge\vec{H}) = \frac{\mathrm{i}\omega}{c^2}\cdot\left[4\pi\sigma(\omega)\cdot\vec{E}(\omega) - \mathrm{i}\omega\cdot\vec{E}(\omega)\right]$$

$$\curvearrowright\ -\vec{\nabla}^2\vec{E} = \vec{E}\cdot\left(\frac{\mathrm{i}\omega}{c^2}\cdot 4\pi\sigma(\omega) + \frac{\omega^2}{c^2}\right) = \left(1 + \frac{4\pi\mathrm{i}\sigma(\omega)}{\omega}\right)\cdot\frac{\omega^2}{c^2}\cdot\vec{E}$$

Complex dielectric constant:

$$\varepsilon(\omega) \equiv 1 + \frac{4\pi i\sigma(\omega)}{\omega}$$

$$\curvearrowright\ -\vec{\nabla}^2\vec{E} = \varepsilon(\omega)\cdot\frac{\omega^2}{c^2}\cdot\vec{E}$$

For $\omega\tau \gg 1$:

$$\varepsilon(\omega) = 1 + \frac{4\pi i}{\omega}\cdot\sigma_{\mathrm{DC}}\cdot\frac{1}{\underbrace{1 - i\omega\tau}_{\approx -i\omega\tau}} \approx 1 - \frac{4\pi}{\omega^2\tau}\cdot\sigma_{\mathrm{DC}} = 1 - \frac{1}{\omega^2}\cdot\frac{4\pi ne^2}{m_e}$$

$$= 1 - \frac{\omega_p^2}{\omega^2}$$

Plasma frequency:

$$\omega_p^2 = \frac{4\pi ne^2}{m_e}$$

- $\omega < \omega_p$: $\varepsilon < 0$: no propagation of e.m. waves
- $\omega > \omega_p$: $\varepsilon > 0$: propagation of e.m waves is possible

Thermal conductivity

Wiedemann-Franz law (κ: thermal conductivity, $[\kappa] = {}^{W}/_{m\cdot K}$; σ: electrical conductivity, $[\sigma] = {}^{1}/_{\Omega\cdot m}$; T: temperature, $[T] = K$):

$$\frac{\kappa}{\sigma} \propto T$$

Fourier's law ($\vec{j}^q$: thermal conductivity, $[j^q] = {}^{W}/_{m^2}$; $\vec{\nabla}T$: temperature gradient, $[\nabla T] = {}^{K}/_{m}$):

$$\vec{j}^q = -\kappa\cdot\vec{\nabla}T$$

Sommerfeld's model

Maxwell-Boltzmann distribution:

$$f_{\mathrm{MB}}(\vec{v}) = n\cdot\left(\frac{m}{2\pi k_B T}\right)^{3/2}\cdot\exp\left\{-\frac{1}{2}\cdot\frac{mv^2}{k_B T}\right\}$$

Ground state ($T \equiv 0$)

Time independent Schroedinger equation for one free and independent electron:

$$H\,\Psi(\vec{r}) = E\,\Psi(\vec{r})\ \curvearrowright\ -\frac{\hbar^2}{2m_e}\vec{\nabla}^2\Psi(\vec{r}) = E\,\Psi(\vec{r})$$

Born–von Karman/periodic boundary conditions for an electron confined to a cube of volume $V = L^3$:

$$\begin{aligned}\Psi(x+L,y,z) &= \Psi(x,y,z)\\ \Psi(x,y+L,z) &= \Psi(x,y,z)\\ \Psi(x,y,z+L) &= \Psi(x,y,z)\end{aligned}$$

Trial solution ignoring the boundary condition:

$$\Psi_{\vec{k}}(\vec{r}) = \frac{1}{\sqrt{V}} \cdot \mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}$$

$$\curvearrowright\ -\frac{\hbar^2}{2m}\left(\partial_x^2+\partial_y^2+\partial_z^2\right)\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}} = \frac{\hbar^2}{2m}\cdot\left(k_x^2+k_y^2+k_z^2\right)\cdot\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}} = \frac{\hbar^2|\vec{k}|^2}{2m}\cdot\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}$$

$$\curvearrowright\ E(\vec{k}) = \frac{\hbar^2\vec{k}^2}{2m}$$

$$\int_V \mathrm{d}^3x \left|\frac{1}{\sqrt{V}}\cdot\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}\right|^2 = \frac{1}{V}\cdot\int_V \mathrm{d}^3x = 1$$

$\Psi_{\vec{k}}(\vec{r})$: eigenstate of the momentum operator:

$$\hat{\vec{p}} \equiv \frac{\hbar}{\mathrm{i}}\vec{\nabla}$$

$$\curvearrowright\ \hat{\vec{p}}\,\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}} = \frac{\hbar}{\mathrm{i}}\cdot\mathrm{i}\vec{k}\cdot\mathrm{e}^{\mathrm{i}\vec{k}\cdot r} = \hbar\vec{k}\cdot\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}$$

Momentum and velocity of an electron in level $\Psi_{\vec{k}}(\vec{r})$:

$$\vec{p} = \hbar\vec{k} \qquad \vec{v} = \frac{\hbar\vec{k}}{m_\mathrm{e}}$$

de Broglie wavelength ($\vec{k}$: wave vector, $k \equiv |\vec{k}|$: wave number):

$$\lambda = \frac{2\pi}{k}$$

Applying the boundary conditions:

$$\mathrm{e}^{\mathrm{i}k_x\cdot L} = \mathrm{e}^{\mathrm{i}k_y\cdot L} = \mathrm{e}^{\mathrm{i}k_z\cdot L} = 1$$

$$\curvearrowright\ k_x = \frac{2\pi n_x}{L}, k_y = \frac{2\pi n_y}{L}, k_z = \frac{2\pi n_z}{L} \quad n_x, n_y, n_z \in \mathbb{Z}$$

Number of allowed states in k-space of volume Ω ($V = L^3$):

$$\frac{\Omega}{\underbrace{(2\pi/L)^3}_{\text{Volume of one state}}} = \frac{\Omega\cdot V}{8\pi^3}$$

Number of allowed $\vec{k}$ values within the *Fermi sphere* (radius k_F):

$$\frac{V}{8\pi^3} \cdot \Omega = \frac{V}{8\pi^3} \cdot \frac{4\pi}{3} k_\mathrm{F}^3 = \frac{V}{6\pi^2} \cdot k_\mathrm{F}^3$$

N electrons with spin $\pm\frac{\hbar}{2}$:

$$N = 2 \cdot \frac{V}{6\pi^2} \cdot k_\mathrm{F}^3 = \frac{V}{3\pi^2} \cdot k_\mathrm{F}^3$$

Ground state of a system with N electrons (density $n = N/V$, k_F: Fermi wave vector):

$$n = \frac{k_\mathrm{F}^3}{3\pi^2}$$

Fermi momentum and *Fermi velocity*:

$$p_\mathrm{F} = \hbar k_\mathrm{F},\ v_\mathrm{F} = \frac{p_\mathrm{F}}{m_\mathrm{e}}$$

Fermi energy:

$$E_\mathrm{F} = \frac{\hbar^2 k_\mathrm{F}^2}{2m_\mathrm{e}}$$

Ground state energy of N electrons:

$$E = \underbrace{2}_{\text{Spin } \pm\hbar/2} \cdot \sum_{k<k_\mathrm{F}} \frac{\hbar^2 k^2}{2m_\mathrm{e}}$$

$$\sum_{\vec{k}} F(\vec{k}) = \left(\frac{V}{8\pi^3}\right) \cdot \sum_{\vec{k}} F(\vec{k}) \cdot \left(\frac{8\pi^3}{V}\right)$$

$$\curvearrowright \lim_{V\to\infty} \frac{1}{V} \cdot \sum_{\vec{k}} F(\vec{k}) = \int \frac{\mathrm{d}\vec{k}}{8\pi^3} F(\vec{k})$$

$$\curvearrowright \underbrace{\frac{E}{V}}_{\text{Energy density}} = 2 \cdot \int_{k<k_\mathrm{F}} \frac{\mathrm{d}^3 k}{8\pi^3} \frac{\hbar^2 k^2}{2m_\mathrm{e}} = \frac{\hbar^2}{8m_\mathrm{e}\pi^3} \cdot \int_0^\pi \mathrm{d}\vartheta\ \sin(\vartheta) \int_0^{2\pi} \mathrm{d}\varphi \int_0^{k_\mathrm{F}} \mathrm{d}r r^4$$

$$= \frac{\hbar^2}{8m_\mathrm{e}\pi^3} \cdot 2 \cdot 2\pi \cdot \frac{1}{5} k_\mathrm{F}^5 = \frac{\hbar^2 k_\mathrm{F}^5}{10\pi^2 m_\mathrm{e}}$$

Energy per electron:

$$\frac{E}{N} = \frac{E}{V} \cdot \frac{3\pi^2}{k_\mathrm{F}^3} = \frac{3}{10} \cdot \frac{\hbar^2 k_\mathrm{F}^2}{m_\mathrm{e}} = \frac{3}{5} \cdot E_\mathrm{F}$$

Fermi temperature:

$$T_\mathrm{F} \equiv \frac{E_\mathrm{F}}{k_\mathrm{B}} = \frac{\hbar^2 k_\mathrm{F}^2}{2 k_\mathrm{B} m_\mathrm{e}}$$

Crystal lattices

Bravais lattice

Position vectors in a Bravais lattice ($\vec{a}_i$: *primitive vectors*):

$$\vec{R} = n_1\,\vec{a}_1 + n_2\,\vec{a}_2 + n_3\,\vec{a}_3; \quad n_1, n_2, n_3 \in \mathbb{Z}$$

Nearest neighbours:

Points in a Bravais lattice closest to a given point ≡ *coordination number*

sc/simple cubic ($\hat{x}, \hat{y}, \hat{z}$: orthogonal unit vectors):

$$\vec{a}_1 = a\,\hat{x},\ \vec{a}_2 = a\,\hat{y},\ \vec{a}_3 = a\,\hat{z}$$

coordination number: 6

bcc/body-centered cubic:

$$\vec{a}_1 = a\,\hat{x},\ \vec{a}_2 = a\,\hat{y},\ \vec{a}_3 = \frac{a}{2}\,(\hat{x}+\hat{y}+\hat{z})$$

$$\vec{a}_1 = \frac{a}{2}\,(\hat{y}+\hat{z}-\hat{x}),\ \vec{a}_2 = \frac{a}{2}\,(\hat{z}+\hat{x}-\hat{y}),\ \vec{a}_3 = \frac{a}{2}\,(\hat{x}+\hat{y}-\hat{z})$$

coordination number: 8

fcc/face-centered cubic:

$$\vec{a}_1 = \frac{a}{2}\,(\hat{y}+\hat{z}),\ \vec{a}_2 = \frac{a}{2}\,(\hat{z}+\hat{x}),\ \vec{a}_3 = \frac{a}{2}\,(\hat{x}+\hat{y})$$

coordination number: 12

Primitive (unit) cell of a lattice:

Fills all of space without voids or overlap when translated along the primitive vectors $\vec{a}_i$; contains exactly one point of the lattice.
$\vec{r} = x_1\,\vec{a}_1 + x_2\,\vec{a}_2 + x_3\,\vec{a}_3,\ x_i \in [0, 1]$

Reciprocal lattice

Definition of the *reciprocal* lattice ($\vec{R}$: Bravais lattice, $\vec{K}$: set of wave vectors):

$$\mathrm{e}^{\mathrm{i}\vec{K}\cdot(\vec{r}+\vec{R})} \overset{!}{=} \mathrm{e}^{\mathrm{i}\vec{K}\cdot\vec{r}} \quad \forall \vec{r}, \vec{R}$$

$$\curvearrowright\ \mathrm{e}^{\mathrm{i}\vec{K}\cdot\vec{R}} = 1 \quad \forall \vec{R}$$

Primitive vectors spanning the reciprocal lattice ($\vec{a}_i$: primitive vectors generating the *direct lattice*):

$$\vec{b}_1 = 2\pi \, \frac{\vec{a}_2 \wedge \vec{a}_3}{\vec{a}_1 \cdot (\vec{a}_2 \wedge \vec{a}_3)}$$

$$\vec{b}_2 = 2\pi \, \frac{\vec{a}_3 \wedge \vec{a}_1}{\vec{a}_1 \cdot (\vec{a}_2 \wedge \vec{a}_3)} = 2\pi \, \frac{\vec{a}_3 \wedge \vec{a}_1}{\vec{a}_2 \cdot (\vec{a}_3 \wedge \vec{a}_1)}$$

$$\vec{b}_3 = 2\pi \, \frac{\vec{a}_1 \wedge \vec{a}_2}{\vec{a}_1 \cdot (\vec{a}_2 \wedge \vec{a}_3)} = 2\pi \, \frac{\vec{a}_1 \wedge \vec{a}_2}{\vec{a}_3 \cdot (\vec{a}_1 \wedge \vec{a}_2)}$$

$$\curvearrowright \; \vec{b}_i \cdot \vec{a}_j = 2\pi \, \delta_{ij}$$

$$\vec{k} = k_1 \, \vec{b}_1 + k_2 \, \vec{b}_2 + k_3 \, \vec{b}_3$$

$$\vec{R} = n_1 \, \vec{a}_1 + n_2 \, \vec{a}_2 + n_3 \, \vec{a}_3 \quad n_i \in \mathbb{Z}$$

$$\curvearrowright \; \vec{k} \cdot \vec{R} = 2\pi \, (k1 \, n_1 + k_2 \, n_2 + k_3 \, n_3) \overset{!}{=} m \cdot 2\pi \quad m \in \mathbb{Z}$$

$$\curvearrowright \; k_i \in \mathbb{Z}$$

Simple cubic (sc):

$$\vec{a}_1 = a \, \hat{x}, \; \vec{a}_2 = a \, \hat{y}, \; \vec{a}_3 = a \, \hat{z}$$

$$\vec{b}_1 = 2\pi \, \frac{a^2 \, (\hat{y} \wedge \hat{z})}{a^3 \, \hat{x} \cdot (\hat{y} \wedge \hat{z})} = \frac{2\pi}{a} \, \hat{x}$$

$$\vec{b}_2 = \frac{2\pi}{a} \, (\hat{z} \wedge \hat{x}) = \frac{2\pi}{a} \, \hat{y}$$

$$\vec{b}_3 = \frac{2\pi}{a} \, (\hat{x} \wedge \hat{y}) = \frac{2\pi}{a} \, \hat{z}$$

$$\curvearrowright \; \text{Reciprocal lattice: sc}$$

Face-centered cubic (fcc):

$$\vec{a}_1 = \frac{a}{2} \, (\hat{y} + \hat{z}), \; \vec{a}_2 = \frac{a}{2} \, (\hat{z} + \hat{x}), \; \vec{a}_3 = \frac{a}{2} \, (\hat{x} + \hat{y})$$

$$\vec{b}_1 = 2\pi \, \frac{(a/2)^2 \, (\hat{z} + \hat{x}) \wedge (\hat{x} + \hat{y})}{(a/2)^3 \, (\hat{y} + \hat{z}) \cdot [(\hat{z} + \hat{x}) \wedge (\hat{x} + \hat{y})]} = \frac{4\pi}{a} \, \frac{\hat{y} - \hat{x} + \hat{z}}{1 + 1} = \frac{4\pi}{a} \, \frac{1}{2} \, (\hat{y} + \hat{z} - \hat{x})$$

$$\vec{b}_2 = \frac{4\pi}{a} \, \frac{1}{2} \, [(\hat{x} + \hat{y}) \wedge (\hat{y} + \hat{z})] = \frac{4\pi}{a} \, \frac{1}{2} \, (\hat{x} + \hat{z} - \hat{y})$$

$$\vec{b}_3 = \frac{4\pi}{a} \, \frac{1}{2} \, [(\hat{y} + \hat{z}) \wedge (\hat{z} + \hat{x})] = \frac{4\pi}{a} \, \frac{1}{2} \, (\hat{x} + \hat{y} - \hat{z})$$

$$\curvearrowright \; \text{Reciprocal lattice: bcc}$$

Body-centered cubic (bcc):

$$\curvearrowright \; \text{Reciprocal lattice: fcc}$$

Volume of the reciprocal lattice primitive cell (v: volume of the primitive cell of the direct lattice):

$$\frac{(2\pi)^3}{v}$$

First Brillouin zone:

Wigner-Seitz primitive cell of the reciprocal lattice.

Lattice planes

Family of lattice planes:

Set of parallel, equally spaced (distance d) lattice planes (contain all points of the Bravais lattice).

Classification of lattice planes:

- For any family of lattice planes, there exists a set of reciprocal lattice vectors with length $2\pi/d$.
- For any reciprocal lattice vector $\vec{K}, |\vec{K}| = 2\pi/d$, there exists a family of lattice planes normal to $\vec{K}$ seperated by distance d.

Plane with *Miller indices h, k, l*:

Plane normal to the reciprocal lattice vector $\vec{k} = h\,\vec{b}_1 + k\,\vec{b}_2 + l\,\vec{b}_3$ $(h, k, l \in \mathbb{Z})$; $|\vec{k}| \overset{!}{=} \min$.

Electrons in a periodic potential

Potential with the periodicity of a Bravais lattice $\vec{R}$:

$$U(\vec{r} + \vec{R}) = U(\vec{r})$$

Schroedinger equation for single electrons in a periodic potential ($\equiv$ *Bloch electrons*):

$$H\,\Psi = \left[-\frac{\hbar^2}{2m_\mathrm{e}}\vec{\nabla}^2 + U(\vec{r})\right]\Psi = E\,\Psi$$

Bloch's theorem (n: *band index*):

$$\Psi_{n\vec{k}}(\vec{r}) = \mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}} \cdot u_{n\vec{k}}(\vec{r}), \quad u_{n\vec{k}}(\vec{r} + \vec{R}) = u_{n\vec{k}}(\vec{r})$$
$$\curvearrowright\ \Psi_{n\vec{k}}(\vec{r} + \vec{R}) = \mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{R}} \cdot \Psi_{n\vec{k}}(\vec{r})$$

Definition of the *translation operator*:

$$T_{\vec{R}}\,f(\vec{r}) = f(\vec{r} + \vec{R}) \quad \forall f$$
$$\curvearrowright\ T_{\vec{R}}H\Psi = H(\vec{r} + \vec{R})\Psi(\vec{r} + \vec{R}) = H\Psi(\vec{r} + \vec{R}) = HT_{\vec{R}}\Psi \quad \forall\Psi$$
$$\curvearrowright\ T_{\vec{R}}\,H = H\,T_{\vec{R}} \leftrightarrow [H, T_{\vec{R}}] = 0$$
$$T_{\vec{R}}T_{\vec{R}'}\Psi(\vec{r}) = T_{\vec{R}'}T_{\vec{R}}\Psi(\vec{r}) = \Psi(\vec{r} + \vec{R} + \vec{R}')$$
$$\curvearrowright\ T_{\vec{R}}T_{\vec{R}'} = T_{\vec{R}'}T_{\vec{R}} = T_{\vec{R}+\vec{R}'} \leftrightarrow [T_{\vec{R}}, T_{\vec{R}'}] = 0$$

Translation operator and Hamiltonian commute → simultaneous eigenfunctions exist:

$$H\,\Psi = E\,\Psi$$
$$T_{\vec{R}}\,\Psi = c(\vec{R})\,\Psi$$

$$T_{\vec{R}'}T_{\vec{R}}\Psi = c(\vec{R})T_{\vec{R}'}\Psi = c(\vec{R})c(\vec{R}')\Psi$$
$$T_{\vec{R}'}T_{\vec{R}}\Psi = T_{\vec{R}+\vec{R}'}\Psi = c(\vec{R}+\vec{R}')\Psi$$
$$\curvearrowright\; c(\vec{R}+\vec{R}') = c(\vec{R})\cdot c(\vec{R}')$$

$\vec{a}_i,\ i = \{1,2,3\}$: primitive vectors for the Bravais lattice

$$\int \mathrm{d}^3x\,|\Psi(\vec{r}+\vec{R})|^2 = \int \mathrm{d}^3x\,|c(\vec{R})|^2\,|\Psi(\vec{r})|^2 \overset{!}{=} \int \mathrm{d}^3x\,|\Psi(\vec{r})|^2$$
$$\curvearrowright\; |c(\vec{R})| = 1 \;\curvearrowright\; c(\vec{R}) = \mathrm{e}^{\mathrm{i}\theta(\vec{R})}$$
$$c(\vec{a}_i) = \mathrm{e}^{2\pi\mathrm{i}x_i}$$
$$c(\vec{R}) = c(\underbrace{\vec{a}_1+\cdots+\vec{a}_1}_{\times n_1})\,c(n_2\vec{a}_2)\,c(n_3\vec{a}_3) = c(\vec{a}_1)^{n_1}\,c(\vec{a}_2)^{n_2}\,c(\vec{a}_3)^{n_3}$$
$$\vec{k} = x_1\vec{b}_1 + x_2\vec{b}_2 + x_3\vec{b}_3$$
$$\curvearrowright\; c(\vec{R}) = \mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{R}} = \exp[\mathrm{i}(x_1n_1\underbrace{\vec{b}_1\cdot\vec{a}_1}_{2\pi} + x_2n_2\vec{b}_2\cdot\vec{a}_2 + x_3n_3\vec{b}_3\cdot\vec{a}_3)] =$$
$$= \left(\mathrm{e}^{2\pi\mathrm{i}x_1}\right)^{n_1}\left(\mathrm{e}^{2\pi\mathrm{i}x_2}\right)^{n_2}\left(\mathrm{e}^{2\pi\mathrm{i}x_3}\right)^{n_3}$$
$$\curvearrowright\; T_{\vec{R}}\,\Psi = \Psi(\vec{r}+\vec{R}) = c(\vec{R})\,\Psi(\vec{r}) = \mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{R}}\,\Psi(\vec{r})$$

Periodic boundary conditions ($N = N_1N_2N_3$: number of primitive cells in a crystal):

$$\Psi(\vec{r}+N_i\vec{a}_i) = \Psi(\vec{r}), \quad i = 1,2,3$$
$$\curvearrowright\; \Psi_{n\vec{k}}(\vec{r}+N_i\vec{a}_i) = \mathrm{e}^{\mathrm{i}N_i\vec{k}\cdot\vec{a}_i}\,\Psi_{n\vec{k}}(\vec{r}) \overset{!}{=} \Psi_{n\vec{k}}(\vec{r})$$
$$\curvearrowright\; \mathrm{e}^{\mathrm{i}N_i\vec{k}\cdot\vec{a}_i} = \exp\left[\mathrm{i}N_i(x_1\vec{b}_1\cdot\vec{a}_i + x_2\vec{b}_2\cdot\vec{a}_i + x_3\vec{b}_3\cdot\vec{a}_i)\right]$$
$$\curvearrowright\; \mathrm{e}^{2\pi\mathrm{i}N_ix_i} = 1 \;\curvearrowright\; x_i = \frac{m_i}{N_i},\ m_i \in \mathbb{Z}$$

Allowed Bloch wave vectors:

$$\vec{k} = \sum_{i=1}^{3}\frac{m_i}{N_i}\,\vec{b}_i$$

Crystal momentum $\hbar\vec{k}$ ($\vec{k}$ can always be confined to the first Brillouin zone):

$$\hat{\vec{p}}\,\Psi_{n\vec{k}} = \frac{\hbar}{\mathrm{i}}\vec{\nabla}\left(\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}\,u_{n\vec{k}}(\vec{r})\right) = \hbar\vec{k}\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}u_{n\vec{k}} - \mathrm{i}\hbar\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}\vec{\nabla}u_{n\vec{k}}(\vec{r}) =$$
$$= \hbar\vec{k}\,\Psi_{n\vec{k}}(\vec{r}) + \frac{\hbar}{\mathrm{i}}\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}\vec{\nabla}u_{n\vec{k}}(\vec{r})$$

Solutions of the Schroedinger equation for $\Psi(\vec{r}) = \mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}u(\vec{r})$:

$$H_{\vec{k}}\Psi = -\frac{\hbar^2}{2m_\mathrm{e}}\vec{\nabla}^2\left[\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}u(\vec{r})\right] + U(\vec{r})\Psi = E_{\vec{k}}\Psi =$$
$$= -\frac{\hbar^2}{2m_\mathrm{e}}\left[\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}\vec{\nabla}^2u_{\vec{k}}(\vec{r}) + 2\left(\vec{\nabla}\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}\right)\left(\vec{\nabla}u_{\vec{k}}(\vec{r})\right) + u_{\vec{k}}(\vec{r})\vec{\nabla}^2\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}\right] + U(\vec{r})\Psi$$
$$= \frac{\hbar^2}{2m_\mathrm{e}}\mathrm{e}^{\mathrm{i}\vec{k}\cdot\vec{r}}\left[-\vec{\nabla}^2u_{\vec{k}}(\vec{r}) - 2\mathrm{i}\vec{k}\vec{\nabla}u_{\vec{k}}(\vec{r}) + u_{\vec{k}}\vec{k}^2\right]$$
$$\curvearrowright\ \frac{\hbar^2}{2m_\mathrm{e}}\left[\left(\frac{\vec{\nabla}}{\mathrm{i}} + \vec{k}\right)^2 + U(\vec{r})\right]u_{\vec{k}}(\vec{r}) = E_{\vec{k}}(\vec{r}), \quad u_{\vec{k}}(\vec{r}+\vec{R}) = u_{\vec{k}}(\vec{r})$$

Band structure ($E_{n\vec{k}}$: enery band):

$$\Psi_{n,\vec{k}+\vec{K}}(\vec{r}) = \Psi_{n,\vec{k}}(\vec{r})$$
$$E_{n,\vec{k}+\vec{K}} = E_{n,\vec{k}}$$

Mean velocity of an electron:

$$\vec{v}_n(\vec{k}) = \frac{1}{\hbar}\vec{\nabla}_{\vec{k}}\,E_n(\vec{k})$$

Phonons

Position of an atom in a Bravais lattice:

$$r(\vec{R}) = \vec{R} + \underbrace{u(\vec{R})}_{\text{small deviation}}$$

Total potential energy of the crystal:

$$U = \frac{1}{2}\sum_{\vec{R},\vec{R}'}\phi\left(r(\vec{R}) - r(\vec{R}')\right) = \frac{1}{2}\sum_{\vec{R},\vec{R}'}\phi\left(\vec{R} - \vec{R}' + u(\vec{R}) - u(\vec{R}')\right)$$

Hamiltonian (M: atomic mass):

$$H = \sum_{\vec{R}}\frac{\vec{P}^2}{2M} + \frac{1}{2}\sum_{\vec{R},\vec{R}'}\phi(\underbrace{\vec{R} - \vec{R}'}_{\equiv\vec{x}} + \underbrace{u(\vec{R}) - u(\vec{R}')}_{\equiv\Delta\vec{x}})$$

Taylor series (H: Hessian, $U^{(\mathrm{eq})}$: equilibrium energy, $U^{(\mathrm{h})}$: harmonic approximation):

$$f(\vec{x} + \Delta\vec{x}) = f(\vec{x}) + \left(\vec{\nabla} f\right) \cdot (\Delta\vec{x}) + \frac{1}{2} (\Delta\vec{x})^t \cdot H \cdot (\Delta\vec{x}) + \dots$$
$$H_{ij} = \frac{\partial^2 f}{\partial x_i \, \partial x_j}$$

6 P

Lagrangian (T: kinetic energy, V: potential energy):

$$L \equiv T - V$$

Action:

$$S[x] = \int_{t_a}^{t_b} L(x, \dot{x}, t) \, \mathrm{d}t \overset{!}{=} \min$$

Variation of the action:

$$\delta S = S[x + \delta x] - S[x] \overset{!}{=} 0$$

Endpoints of the extremum path x shall be fixed:

$$\delta x(t_a) = \delta x(t_b) = 0$$

$$\curvearrowright \; S[x + \delta x] = \int_{t_a}^{t_b} L(x + \delta x, \dot{x} + \delta\dot{x}, t) \, \mathrm{d}t = \int_{t_a}^{t_b} \left[L(x, \dot{x}, t) + \frac{\partial L}{\partial x} \delta x + \frac{\partial L}{\partial \dot{x}} \delta\dot{x} \right] \mathrm{d}t$$
$$= S[x] + \int_{t_a}^{t_b} \left[\frac{\partial L}{\partial x} \delta x + \frac{\partial L}{\partial \dot{x}} \delta\dot{x} \right] \mathrm{d}t$$
$$\curvearrowright \; \delta S = \int_{t_a}^{t_b} \left[\frac{\partial L}{\partial x} \delta x + \frac{\partial L}{\partial \dot{x}} \frac{\mathrm{d}}{\mathrm{d}t} \delta x \right] \mathrm{d}t = \int_{t_a}^{t_b} \frac{\partial L}{\partial x} \delta x \, \mathrm{d}t + \underbrace{\left. \frac{\partial L}{\partial \dot{x}} \delta x \right|_{t_a}^{t_b}}_{=0} - \int_{t_a}^{t_b} \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial L}{\partial \dot{x}} \right) \delta x \, \mathrm{d}t$$
$$= \int_{t_a}^{t_b} \delta x \left[\frac{\partial L}{\partial x} - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{x}} \right] \mathrm{d}t \overset{!}{=} 0 \quad \text{for arbitrary } \delta x$$
$$\curvearrowright \; \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = 0$$

Euler-Lagrange equation(s) for n coordinates:

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0, \quad i = 1, \dots, n$$

Problem 2–1: Free particle with Lagrangian $L = \frac{1}{2}m\dot{x}^2$. Show:

$$S = \frac{m}{2}\frac{(x_b - x_a)^2}{t_b - t_a}$$

Solution:

$$x(t) = x_a + \frac{x_b - x_a}{t_b - t_a} \cdot (t - t_a)$$
$$\curvearrowright\ \dot{x}(t) = \frac{x_b - x_a}{t_b - t_a}$$
$$\curvearrowright\ S = \frac{m}{2}\int_{t_a}^{t_b} \frac{(x_b - x_a)^2}{(t_b - t_a)^2}\,\mathrm{d}t = \frac{m}{2}\frac{(x_b - x_a)^2}{(t_b - t_a)^2} \cdot t\Big|_{t_a}^{t_b} = \frac{m}{2}\frac{(x_b - x_a)^2}{t_b - t_a}$$

Problem 2–2: Harmonic oscillator with Lagrangian $L = \frac{m}{2}(\dot{x}^2 - \omega^2 x^2)$, $T \equiv t_b - t_a$. Show:

$$S = \frac{m\omega}{2\sin(\omega T)}\left[(x_a^2 + x_b^2)\cos(\omega T) - 2x_a x_b\right]$$

Solution:

$$x(t) = A\sin(\omega t) + B\cos(\omega t)$$
$$x(t_a \equiv 0) = B = x_a$$
$$x(t_b - t_a \equiv T) = A\sin(\omega T) + x_a\cos(\omega T) = x_b$$
$$\curvearrowright\ A = \frac{x_b - x_a\cos(\omega T)}{\sin(\omega T)}$$
$$\curvearrowright\ x(t) = \frac{x_b - x_a\cos(\omega T)}{\sin(\omega T)}\sin(\omega t) + x_a\cos(\omega t)$$
$$\dot{x}(t) = A\omega\cos(\omega t) - x_a\omega\sin(\omega t)$$
$$S = \frac{m}{2}\int_{t_a}^{t_b} \dot{x}\dot{x}\,\mathrm{d}t - \frac{m\omega^2}{2}\int_{t_a}^{t_b} x^2\,\mathrm{d}t = \frac{m}{2}\dot{x}x\Big|_{t_a}^{t_b} - \frac{m}{2}\int_{t_a}^{t_b} \underbrace{\ddot{x}}_{-\omega^2 x}\, x\,\mathrm{d}t - \frac{m\omega^2}{2}\int_{t_a}^{t_b} x^2\,\mathrm{d}t$$
$$= \frac{m}{2}x\dot{x}\Big|_{t_a}^{t_b}$$
$$\curvearrowright\ S = \frac{m}{2}\left[x_b\dot{x}(T) - x_a\dot{x}(0)\right] = \frac{m}{2}\left[x_b A\omega\cos(\omega T) - x_b x_a\omega\sin(\omega T) - x_a A\omega\right]$$
$$= \frac{m\omega}{2\sin(\omega T)}\left[x_b\left(x_b - x_a\cos(\omega T)\right)\cos(\omega T) - x_b x_a\sin^2(\omega T) - x_a\left(x_b - x_a\cos(\omega T)\right)\right]$$
$$= \cdots = \frac{m\omega}{2\sin(\omega T)}\left[(x_a^2 + x_b^2)\cos(\omega T) - 2x_a x_b\right]$$

Problem 2–3: Particle under constant force F with $L = \frac{1}{2}m\dot{x}^2 + Fx$, $T \equiv t_b - t_a$. Show:

$$S = \frac{m(x_b - x_a)^2}{2T} + \frac{FT(x_b + x_a)}{2} - \frac{F^2T^3}{24m}$$

Solution:

$$
\begin{aligned}
x(t) &= x_a + \dot{x}(t_a)\cdot(t-t_a) + \frac{1}{2}\frac{F}{m}\cdot(t-t_a)^2 \\
x(t_b) = x_b &= x_a + \dot{x}(t_a)T + \frac{F}{2m}T^2 \\
\curvearrowright\ \dot{x}(t_a) &= \frac{x_b-x_a}{T} - \frac{FT}{2m} \\
\curvearrowright\ x(t) &= x_a + \left(\frac{x_b-x_a}{T} - \frac{FT}{2m}\right)\cdot(t-t_a) + \frac{F}{2m}\cdot(t-t_a)^2 \\
\frac{m}{2}\dot{x}^2 &= \frac{m}{2}\left(\frac{x_b-x_a}{T} - \frac{FT}{2m}\right)^2 + \left(\frac{x_b-x_a}{T} - \frac{FT}{2m}\right)F\cdot(t-t_a) + \frac{F^2}{2m}\cdot(t-t_a)^2 \\
Fx &= Fx_a + \left(\frac{x_b-x_a}{T} - \frac{FT}{2m}\right)F\cdot(t-t_a) + \frac{F^2}{2m}\cdot(t-t_a)^2 \\
\frac{m}{2}\dot{x}^2 + Fx &= \frac{F^2}{m}\cdot(t-t_a)^2 + 2F\left(\frac{x_b-x_a}{T} - \frac{FT}{2m}\right)\cdot(t-t_a) + Fx_a + \frac{m}{2}\left(\frac{x_b-x_a}{T} - \frac{FT}{2m}\right)^2 \\
\curvearrowright\ S = \int_{t_a}^{t_b}\left(\frac{m}{2}\dot{x}^2 + Fx\right)\mathrm{d}t &= \frac{F^2T^3}{3m} + F\left(\frac{x_b-x_a}{T} - \frac{FT}{2m}\right)T^2 + Fx_aT + \frac{mT}{2}\left(\frac{x_b-x_a}{T} - \frac{FT}{2m}\right)^2 \\
&= \cdots = \frac{m}{2T}(x_b-x_a)^2 + \frac{FT}{2}(x_b+x_a) - \frac{F^2T^3}{24m}
\end{aligned}
$$

Infinitesimal transformation ($\varepsilon \longrightarrow 0$):

$$
\begin{aligned}
q &\longrightarrow q + \varepsilon\delta \\
\curvearrowright\ L(q,\dot{q}) &\longrightarrow L(q+\varepsilon\delta q, \dot{q}+\varepsilon\delta\dot{q}) = L(q,\dot{q}) + \varepsilon\delta q\,\frac{\partial L}{\partial q} + \varepsilon\delta\dot{q}\,\frac{\partial L}{\partial \dot{q}} \\
&= L(q,\dot{q}) + \varepsilon\delta q\,\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}} + \varepsilon\frac{\partial L}{\partial \dot{q}}\frac{\mathrm{d}}{\mathrm{d}t}\delta q = L(q,\dot{q}) + \frac{\mathrm{d}}{\mathrm{d}t}\left(\varepsilon\delta q\,\frac{\partial L}{\partial \dot{q}}\right) \\
\curvearrowright\ \delta L &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}}\,\varepsilon\delta q\right)
\end{aligned}
$$

Noether current:

$$j \equiv \frac{\partial L}{\partial \dot{q}}\,\delta q$$

Hamiltonian ($p \equiv \frac{\partial L}{\partial \dot{q}}$: momentum):

$$H \equiv T + V = p\dot{q} - L$$

Lorentz transformation along the x-axis ($\gamma \equiv \frac{1}{\sqrt{1-\beta^2}}$, $\beta \equiv \frac{v}{c}$):

1. $x' = \gamma(x - vt)$
2. $y' = y$

3. $z' = z$

4. $t' = \gamma(t - \frac{v}{c^2}x)$

Position four-vector:

$$x^0 \equiv ct,\ x^1 = x,\ x^2 = y,\ x^3 = z$$

Lorentz transformation using four-vectors:

1. $x'^0 = \gamma(x^0 - \beta x^1)$
2. $x'^1 = \gamma(x^1 - \beta x^0)$
3. $x'^2 = x^2$
4. $x'^3 = x^3$

Matrix form:

$$\begin{bmatrix} x'^0 \\ x'^1 \\ x'^2 \\ x'^3 \end{bmatrix} = \begin{bmatrix} \gamma & -\gamma\beta & 0 & 0 \\ -\gamma\beta & \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x^0 \\ x^1 \\ x^2 \\ x^3 \end{bmatrix}$$

$$\curvearrowright\ x'^\mu = \sum_{\nu=0}^{3} \Lambda^\mu_\nu x^\nu \quad \mu:\ \text{row},\ \nu:\ \text{column}$$

Definition of a four-vector (Λ: Lorentz transformation):

$$a'^\mu = \Lambda^\mu_\nu x^\nu$$

Contravariant four-vector:

$$a^\mu = (a^0, a^1, a^2, a^3)^t$$

Covariant four-vector:

$$a_\mu = (-a^0, a^1, a^2, a^3)$$

Scalar product of two four-vectors (invariant under Lorentz transformations):

$$a \cdot b = a_\mu b^\mu = a^\mu b_\mu = -a^0 b^0 + a^1 b^1 + a^2 b^2 + a^3 b^3$$

Minkowski metric $(-, +, +, +)$:

$$g_{\mu\nu} = g^{\mu\nu} = \mathrm{diag}(-1, 1, 1, 1)$$
$$\curvearrowright\ a_\mu = g_{\mu\nu} a^\nu$$

Distance:

- $a_\mu a^\mu > 0$: spacelike
- $a_\mu a^\mu = 0$: lightlike
- $a_\mu a^\mu < 0$: timelike

Proper time ($\vec{u} = \frac{\mathrm{d}\vec{l}}{\mathrm{d}t}$: velocity in an inertial system $\equiv$ ordinary velocity):

$$\mathrm{d}\tau = \sqrt{1 - \frac{u^2}{c^2}}$$

Proper velocity:

$$\vec{\eta} = \frac{\mathrm{d}\vec{l}}{\mathrm{d}\tau} = \frac{1}{\sqrt{1 - \frac{u^2}{c^2}}}\vec{u}$$

$$\eta^\mu = \frac{\mathrm{d}x^\mu}{\mathrm{d}\tau}; \;\; \eta'^\mu = \Lambda^\mu_\nu \eta^\nu$$

Relativistic momentum:

$$\vec{p} = m\vec{\eta} = \frac{m\vec{u}}{\sqrt{1 - \frac{u^2}{c^2}}}$$

Relativistic energy:

$$E = \frac{mc^2}{\sqrt{1 - \frac{u^2}{c^2}}} \quad \curvearrowright p^0 = \frac{E}{c}; p^i = \vec{p}^i$$

Rest and kinetic energy:

$$E_{\text{rest}} = mc^2; \quad E_{\text{kin}} = E^2 - mc^2$$

Relativistic energy:

$$p \cdot p = p_\mu p^\mu = -(p^0)^2 + \vec{p}^2 = -m^2c^2$$
$$\curvearrowright E^2 = m^2c^4 + p^2c^2$$

Second rank tensor:

$$t'^{\mu\nu} = \Lambda^\mu_\lambda \Lambda^\nu_\sigma t^{\lambda\sigma}$$

$$\Lambda = \begin{bmatrix} \gamma & -\gamma\beta & 0 & 0 \\ -\gamma\beta & \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$t^{\mu\nu} = \begin{bmatrix} t^{00} & t^{01} & t^{02} & t^{03} \\ t^{10} & t^{11} & t^{12} & t^{13} \\ t^{20} & t^{21} & t^{22} & t^{23} \\ t^{30} & t^{31} & t^{32} & t^{33} \end{bmatrix}$$

(Anti-)symmetric tensor:

$$t^{\mu\nu} = (-)t^{\nu\mu}$$

General antisymmetric tensor:

$$\begin{bmatrix} 0 & t^{01} & t^{02} & t^{03} \\ -t^{01} & 0 & t^{12} & t^{13} \\ -t^{02} & -t^{12} & 0 & t^{23} \\ -t^{03} & -t^{13} & -t^{23} & 0 \end{bmatrix}$$

Field tensor:

$$F^{\mu\nu} = \begin{bmatrix} 0 & {}^{E_x}\!/_{c} & {}^{E_y}\!/_{c} & {}^{E_z}\!/_{c} \\ -{}^{E_x}\!/_{c} & 0 & B_z & -B_y \\ -{}^{E_y}\!/_{c} & -B_z & 0 & B_x \\ -{}^{E_z}\!/_{c} & B_y & -B_x & 0 \end{bmatrix}$$

Dual tensor: ($\frac{\vec{E}}{c} \longrightarrow \vec{B},\ \vec{B} \longrightarrow -\frac{\vec{E}}{c}$):

$$G^{\mu\nu} = \begin{bmatrix} 0 & B_x & B_y & B_z \\ -B_x & 0 & -{}^{E_z}\!/_{c} & {}^{E_y}\!/_{c} \\ -B_y & {}^{E_z}\!/_{c} & 0 & -{}^{E_x}\!/_{c} \\ -B_z & -{}^{E_y}\!/_{c} & {}^{E_x}\!/_{c} & 0 \end{bmatrix}$$

Current density 4-vector:

$$j^{\mu} = (c\varrho,\ j_x,\ j_y,\ j_z)^t$$

Continuity equation:

$$\frac{\partial j^{\mu}}{\partial x^{\mu}} = 0$$

Maxwell's equations (summation over ν):

$$\frac{\partial F^{\mu\nu}}{\partial x^{\nu}} = \mu_0 j^{\mu}, \quad \frac{\partial G^{\mu\nu}}{\partial x^{\nu}} = 0$$

Vector potential of the magnetic field ($\vec{\nabla} \cdot \vec{B} = 0$):

$$\vec{B} = \vec{\nabla} \wedge \vec{A}$$

$$\curvearrowright\ \vec{\nabla} \wedge \vec{E} = -\frac{\partial}{\partial t}\vec{B} = -\frac{\partial}{\partial t}\left(\vec{\nabla} \wedge \vec{A}\right)$$

$$\curvearrowright\ \vec{\nabla} \wedge \left(\vec{E} + \frac{\partial \vec{A}}{\partial t}\right) = 0$$

$$\curvearrowright\ \exists\phi :\ \vec{\nabla}\phi = -\left(\vec{E} + \frac{\partial \vec{A}}{\partial t}\right)$$

$$\curvearrowright\ \vec{E} = -\vec{\nabla}\phi - \frac{\partial \vec{A}}{\partial t}$$

$$\vec{\nabla} \cdot \vec{E} = \vec{\nabla} \cdot \left(-\vec{\nabla}\phi - \frac{\partial \vec{A}}{\partial t} \right) = -\vec{\nabla}^2 \phi - \frac{\partial}{\partial t} \left(\vec{\nabla} \cdot \vec{A} \right) = \frac{1}{\varepsilon_0} \varrho$$
$$\curvearrowright \; \vec{\nabla}^2 \phi + \frac{\partial}{\partial t} \left(\vec{\nabla} \cdot \vec{a} \right) = -\frac{1}{\varepsilon_0} \varrho$$

$$\vec{\nabla} \wedge \vec{B} = \vec{\nabla} \wedge \left(\vec{\nabla} \wedge \vec{A} \right) = \mu_0 \vec{j} + \mu_0 \varepsilon_0 \frac{\partial \vec{E}}{\partial t} = \mu_0 \vec{j} - \mu_0 \varepsilon_0 \vec{\nabla} \left(\frac{\partial \phi}{\partial t} \right) - \mu_0 \varepsilon_0 \frac{\partial^2 \vec{A}}{\partial t^2}$$
$$\curvearrowright \; \left(\vec{\nabla}^2 \vec{A} - \mu_0 \varepsilon_0 \frac{\partial^2 \vec{A}}{\partial t^2} \right) - \vec{\nabla} \left(\vec{\nabla} \cdot \vec{A} + \mu_0 \varepsilon_0 \frac{\partial \phi}{\partial t} \right) = -\mu_0 \vec{j}$$

$$\vec{A}' = \vec{A} + \vec{\alpha}, \quad \vec{\nabla} \wedge \vec{A} = \vec{B}, \vec{\nabla} \wedge \vec{A}' = \vec{B}$$
$$\curvearrowright \; \vec{\nabla} \wedge \vec{\alpha} = 0$$
$$\vec{\nabla} \wedge (\vec{\nabla} f) \equiv 0 \, \forall f \; \curvearrowright \; \vec{\alpha} = \vec{\nabla} \lambda$$

$$\phi' = \phi + \beta, \quad \vec{\nabla} \phi = -\vec{E}, \vec{\nabla} \phi' = -\vec{E}$$
$$\curvearrowright \; \vec{\nabla} \beta + \frac{\partial \vec{\alpha}}{\partial t} = \vec{\nabla} \left(\beta + \frac{\partial \lambda}{\partial t} \right) = 0 \; \curvearrowright \; \beta = -\frac{\partial \lambda}{\partial t}$$

Gauge transformations:

$$\vec{A} \longrightarrow \vec{A} + \vec{\nabla} \lambda$$
$$\phi \longrightarrow \phi - \frac{\partial \lambda}{\partial t}$$

Coulomb gauge:

$$\vec{\nabla} \cdot \vec{A} = 0$$

Lorentz gauge:

$$\vec{\nabla} \cdot \vec{A} = -\mu_0 \varepsilon_0 \frac{\partial \phi}{\partial t}$$

Four-vector potential:

$$A^\mu = \left(\frac{\phi}{c}, A_x, A_y, A_z \right)^t$$

Field tensor:

$$F^{\mu\nu} = \frac{\partial A^\nu}{\partial x_\mu} - \frac{\partial A^\mu}{\partial x_\nu} = \partial^\mu A^\nu - \partial^\nu A^\mu$$

Four-gradient:

$$\frac{\partial}{\partial x^\mu} = \left(\frac{1}{c}\frac{\partial}{\partial t}, \vec{\nabla}\right) \equiv \partial_\mu$$
$$\frac{\partial}{\partial x_\mu} = \left(-\frac{1}{c}\frac{\partial}{\partial t}, \vec{\nabla}\right) \equiv \partial^\mu$$

d'Alembert operator:

$$\Box \equiv \frac{\partial}{\partial x_\mu}\frac{\partial}{\partial x^\mu} = \partial^\mu \partial_\mu = -\frac{1}{c^2}\frac{\partial^2}{\partial t^2} + \vec{\nabla}^2$$

Definition of Lorentz transformations:

$$x'^2 = \eta_{\mu\nu} x'^\mu x'^\nu = \eta_{\mu\nu}(\Lambda^\mu_{\ \alpha} x^\alpha)(\Lambda^\nu_{\ \beta} x^\beta) \overset{!}{=} x^2 = \eta_{\alpha\beta} x^\alpha x^\beta$$
$$\curvearrowright \eta_{\mu\nu}\Lambda^\mu_{\ \alpha}\Lambda^\nu_{\ \beta} = \eta_{\alpha\beta}$$

Lagrangian ($\mathcal{L}$: Lagrangian density):

$$L = \int \mathcal{L}\, \mathrm{d}^n x$$

Action:

$$S = \int L\, \mathrm{d}t = \int \mathcal{L}\, \mathrm{d}t\, \mathrm{d}^n x$$

Variation of the action ($\delta\phi(x, t_a) = \delta\phi(x, t_b) = 0$, $\delta\phi(x \to \pm\infty) \to 0$):

$$\delta S = \delta \int_{t_a}^{t_b} \mathrm{d}t \int_{-\infty}^{\infty} \mathrm{d}x\, \mathcal{L}(\phi, \dot{\phi}, t)$$

$$= \int_{t_a}^{t_b} \mathrm{d}t \int_{-\infty}^{\infty} \mathrm{d}x\, \left[\mathcal{L}(\phi + \delta\phi, \dot{\phi} + \delta\dot{\phi}, \partial_x\phi + \partial_x\delta\phi, t) - \mathcal{L}(\phi, \dot{\phi}, \partial_x\phi, t)\right]$$

$$= \int_{t_a}^{t_b} \mathrm{d}t \int_{-\infty}^{\infty} \mathrm{d}x\, \left[\frac{\partial\mathcal{L}}{\partial\phi}\delta\phi + \frac{\partial\mathcal{L}}{\partial\dot{\phi}}\delta\dot{\phi} + \frac{\partial\mathcal{L}}{\partial(\partial_x\phi)}\partial_x\delta\phi\right]$$

$$\int_{t_a}^{t_b} \mathrm{d}t \int_{-\infty}^{\infty} \mathrm{d}x\, \frac{\partial\mathcal{L}}{\partial\dot{\phi}}\frac{\mathrm{d}}{\mathrm{d}t}\delta\phi = \underbrace{\int_{-\infty}^{\infty} \mathrm{d}x\, \frac{\partial\mathcal{L}}{\partial\dot{\phi}}\delta\phi\bigg|_{t_a}^{t_b}}_{=0} - \int_{-\infty}^{\infty} \mathrm{d}x\, \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial\mathcal{L}}{\partial\dot{\phi}}\right)\delta\phi$$

$$\int_{-\infty}^{\infty} \mathrm{d}x \int_{t_a}^{t_b} \mathrm{d}t\, \frac{\partial\mathcal{L}}{\partial(\partial_x\phi)}\frac{\mathrm{d}}{\mathrm{d}x}\delta\phi = \underbrace{\int_{t_a}^{t_b} \mathrm{d}t\, \frac{\partial\mathcal{L}}{\partial(\partial_x\phi)}\delta\phi\bigg|_{-\infty}^{\infty}}_{=0} - \int_{-\infty}^{\infty} \mathrm{d}x \int_{t_a}^{t_b} \mathrm{d}t\, \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\partial\mathcal{L}}{\partial(\partial_x\phi)}\right)\delta\phi$$

$$\curvearrowright\ \delta S = \int_{t_a}^{t_b} \mathrm{d}t \int_{-\infty}^{\infty} \mathrm{d}x\, \delta\phi \left[\frac{\partial\mathcal{L}}{\partial\phi} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial\mathcal{L}}{\partial\dot{\phi}}\right) - \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\partial\mathcal{L}}{\partial(\partial_x\phi)}\right)\right] \overset{!}{=} 0 \quad \text{for arbitrary } \delta\phi$$

$$\curvearrowright\ \frac{\partial\mathcal{L}}{\partial\phi} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial\mathcal{L}}{\partial\dot{\phi}}\right) - \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\partial\mathcal{L}}{\partial(\partial_x\phi)}\right) = 0$$

Euler-Lagrange equations for n fields:

$$\partial_\mu\left(\frac{\partial\mathcal{L}}{\partial\left(\partial_\mu\phi_i\right)}\right) - \frac{\partial\mathcal{L}}{\partial\phi_i} = 0 \quad i = 1, \ldots, n$$

Noether current:

$$j^\mu = \frac{\partial\mathcal{L}}{\partial\left(\partial_\mu\phi\right)}\delta\phi$$

$$\phi \to \phi + \varepsilon\delta\phi : \delta\mathcal{L} = 0 \curvearrowright \partial_\mu j^\mu = -\frac{\partial j^0}{\partial t} + \vec{\nabla}\cdot\vec{j} = 0$$

$$j^0 : \text{charge density}, \vec{j} : \text{current density}$$

Hamiltonian density and momentum:

$$\mathcal{H} = \frac{\partial\mathcal{L}}{\partial\dot{\phi}_\mu}\dot{\phi}_\mu - \mathcal{L}$$

$$\Pi^\mu = \frac{\partial\mathcal{L}}{\partial\dot{\phi}_\mu}$$

Maxwell's equations for $c = \mu_0 = \varepsilon_0 = 1$:

$$\vec{\nabla} \cdot \vec{E} = \varrho$$
$$\vec{\nabla} \cdot \vec{B} = 0$$
$$\vec{\nabla} \wedge \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$
$$\vec{\nabla} \wedge \vec{B} = \vec{j} + \frac{\partial E}{\partial t}$$

Lagrangian density for classical electrodynamics:

$$\mathcal{L} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} - j^{\mu} A_{\mu}$$