
USING PHYSICAL QUANTITIES IN APPLIED MATHEMATICAL PROBLEMS WITH MAPLE

Tsvetanka Kovacheva

Abstract. *The present article discusses units of measure and their base units, work environments built in the Units package of the computer algebra system Maple. An analysis is drawn of the tools of the application in connection with the use of physical quantities and their features. Maple's main commands are arranged in groups depending on the function. Some applied mathematical problems are given as examples making use of derivative, integral and differential equations.*

Key words: *conversion, dimension, Maple, unit, systems of quantities, systems of units, Units package.*

ACM Classification Keywords: *G. Mathematics of Computing, I. Computing Methodologies*

1. Introduction

When teaching Maths students of the technical majors at Universities an emphasis should be placed on the applicable nature of the mathematical terms introduced, such as derived, integral, differential equation, order, etc. with the purpose of mastering essential knowledge and skills associated with their future engineering occupation. The need to solve various production problems entails various calculations where the application of dimensionless quantities. However, in a number of areas there is a need to use physical quantities with indication and their units' measuring This could be measuring the strain on a beam, calculating the thermal duty cycle of a transistor, determining the movement resistance of an object in liquid, etc. [8]. The units of measure provide quantity and quality evaluation of the calculation. Each science deals with units of measure rendering the results of its application. Those can be data about a process at its various stages of development. This information can display how a process evolves, what steps are necessary to be taken for its enhancement towards the desired end purpose. In a number of cases the mathematical model of processes or phenomena is demonstrated through complex systems of differential or integral differential equations. This demands determining of a numeric solution using and developing appropriate algorithms and methods, for which it is necessary to apply higher-level algorithmic languages. When using computer applications, however, we only deal with the numerical values of variables and not with their corresponding units. When giving a mental solution to a physical problem we operate entirely with physical quantities. Therefore, with languages it is vital to keep in mind the rule that all physical quantities ought to be in one system of units. They have to be without factors such as miles, mega, etc. [3,10]. The next step in utilizing computer technology when solving physical-mathematical, engineering-technical as well as study problems is represented by computer algebra system (CAS) like Maple, Mathematica, etc. [4]. One of the most common CAS is Maple which is used with solving scientific-technical, engineering and study problems [4,5]. The product represents a powerful computer system, who gives opportunities for efficient solutions of Algebraic and Geometrical problems, tasks of Mathematical Analysis, Probability Theory and Statistics, numerical calculations, etc. The system allows for analytical and numerical solving of equations and systems, drawing graphs, optimization, mathematical modeling, transformations, statistical data processing, symbolic mathematics, etc. It also allows for determining of the numerical values of quantities when solving specific problems and their dimensions. All this facilitates the reference to physical dimension formulae. Its advantage is that by only a couple of lines of commands one unit of measure or quantity can be converted into another, which is quite handy for mathematicians, engineers, economists, etc. These calculations are carried out with the help of the Units package for calculations of physical quantities.

2. Quantities, units and systems [1,2,8]

2.1. Quantities and units

A physical quantity is either a quantity within physics that can be measured, as mass, volume, etc. or the result of a measurement. *Physical quantities* are divided into categories. *Quantities of the same category* are those that can be compared. e.g. a category with quantities like length, distance, diameter, height, etc. *Unit or quantity for comparison* is a constant quantity that allows comparison of quantities within one category. Any other quantity of

the category can be displayed as a function of this unit by multiplying it by a number (the numerical value of the quantity, represented by this unit). e.g. tension of an electric circuit $U=0,05 \cdot 10^3$ $V=0,05kV$.

2.2. System of quantities

Quantities of different categories are those that connect by equations manifesting the physical laws. In order to define systems of quantities and to introduce the term *physical dimension* (was introduced by Fourier, 1822), certain quantities have to be regarded as independent, i.e. they are considered base quantities. In 1960 at the 11th General Conference in Measures and Weights a set of base quantities of the *International System of Units* (SI) was accepted, each field of study having a need only of a few of those quantities: Mechanics: length, mass and time; Electricity and magnetism: length, mass, time and electrical current.

2.3. System of units

Units in a system are selected arbitrary, however, it is more convenient if the base and derived units are in accordance with the chosen system of quantities. Such a system of units is called coherent (SI system). The coherent unit for all dimensionless quantities is the unit one, indicated with 1.

3. Systems of Units built in Maple [6,7,8,9,11]

The following systems of units are built in Maple: *Atomic, CGS, EMU, ESU, FPS, MKS, MTS, SI*.

3.1. Atomic

Atomic units form a system of units convenient for atomic physics, electromagnetism and quantum electrodynamics, particularly for describing the properties of electrons. They have been chosen such that the fundamental electron properties are all equal to one atomic unit. The numerical values of the following six physical constants are all unity by definition: *two properties of the electron, its mass and charge* - the electron mass m_e ($9.1093897 \cdot 10^{-31}$ kg) and elementary charge e ($1.60217733 \cdot 10^{-19}$ C); *two properties of the hydrogen atom, its Bohr radius and electric potential energy in the ground state* - Bohr radius a_0 ($5.291 772 108(18) \cdot 10^{-11}$ m) and Hartree energy E_h ($4.359 744 17(75) \cdot 10^{-18}$ J); *two constants* - angular momentum - Planck's constant $\hbar/2\pi$ ($1.054 571 68(18) \cdot 10^{-34}$ J s) and electrostatic force constant - Coulomb's constant $1/(4\pi\epsilon_0)$ ($8.9875516 \cdot 10^9$ C⁻² N m²). These six quantities are not independent; to normalize all six quantities to 1, it suffices to normalize any four of them to 1. Derived Atomic Units are time, velocity, force, current, temperature, pressure.

3.2. The Centimeter-Gram-Second (CGS)

The system goes back to a proposal made in 1832 by the German mathematician Carl Gauss. The mechanical units are the same in all CGS systems, but there are several variants of electric additions. Length, mass and time, however are not sufficient to define electric and magnetic quantities; a fourth dimension must be included. The system has four varieties: electrostatic, electromagnetic, Gaussian and Heaviside-Lorentz system. The different CGS systems arose from differing choices of the additional dimension. Like SI it has same base units, but they have other decimal prefix. Derived CGS are the units dyne, erg, poise, stokes, gauss, oersted and maxwell.

3.3. The Electromagnetic System of Units (EMU)

The electromagnetic system of units (EMU) has the centimeter, gram, second and biot as its base units, where the biot is energy-equivalent to the unit square root dyne ($(\text{cm})^{1/2} \cdot \text{g}^{1/2}/\text{s}$). For naming consistency, the biot is called the abampere.

3.4. The Electrostatic System of Units (ESU)

The electrostatic system of units (ESU) has the centimeter, gram, second and franklin as its base units, where the franklin is energy-equivalent to the unit square root dyne centimeter ($\text{cm}^{3/2} \cdot \text{g}^{1/2}/\text{s}$). For naming consistency, the franklin is called the abcoulomb.

3.5. The Foot-Pound-Second (FPS)

The FPS (the British, imperial or often called in USA, the inch-pound) system of units has the foot, pound and second as its base units. Unlike modern systems of units, composite dimensions are not necessarily represented by a product of powers of the base units.

3.6. The Meter-Kilogram-Second (MKS)

The MKS system of units has the meter, kilogram and second as its base units. Although the MKS system does not include units of electricity or magnetism, the SI system, which is also based on the meter, kilogram and second, does.

3.7. The Meter-Tonne-Second (MTS)

The MTS system of units has the meter, tonne and second as its base units.

3.8. International system SI [4]

The International System of Units SI consists of a set of units together with a set of prefixes. The system is based on: seven nominally dimensionally independent base units: kilogram, meter, second, mole, ampere, kelvin and candela; two dimensionless units - radian and steradian; prefix (kilo, nano, etc.) - a name or associated symbol that precedes a unit of measure (or its symbol) to form a decimal multiple or submultiple. They are used to reduce the quantity of zeroes in numerical equivalencies; derived units with special names - hertz, newton, joule, watt, pascal, etc.; compound units derived from SI units - square meter, cubic metre, radian per second, newton second, etc.; Non-SI units accepted for use with SI - minute, hour, degree of arc, minute of arc, second of arc, litre, tonne, electronvolt, astronomical unit, atomic mass unit. In addition to the SI units there are also a set of non-SI units accepted for use with SI.

4. Maple tools for the conversion of data [9]

The conversion of data in Maple is provided for by its Units package. It has a graphic customer interface for the transformation of units in separate environments for the completion of calculations. The commands for dimensions and units are in the dialogue window Unit Converter and are described in the package *Unit Converter Support Package*. Each command in the package is accessible in two *forms*: short and long. The long form of command is necessary when the short one is not defined by the *with* command, `with(PackageName, command)` or when there is a command of the same name in the Maple package `PackageName` in use.

4.1. Work environments

Maple package offers 3 work environments: *Default* – it does not accept any use of units. However, there are conversion routines that convert units. It consists of the following commands: `convert[units]`, `convert[temperature]`, `convert[conversion_table]`, `convert[dimensions]`, `convert[system]`; *Standard* – some of the options are modified to support entering of units as expressions. Typing in `with(Units[Standard])` command displays options for entering of units while the use of short forms from the Units package for dimensions, units and systems of units is not supported. `*Unit(unit_name)` command is used to enter units as expressions. The result appears in square brackets. The different environment allows for quantities to be indicated in common symbols, used as names of the units, where *m* refers to 'meter' and *s* to 'seconds'; *Natural* – the list of commands for the calculation of units is the same, with the difference of options. Here the natural numerical system for the designation of signs (`*unit_name`) is used whereas units are entered as expressions.

4.2. The Convert command

In Maple the `convert` command can be applied directly for single transformations without the `with(Units)` command. Determinant for the transformation is a variable without sign. The *conversion factor* can be determined with using an unassigned variable in the `convert` command, e. g. commands for conversion from 5 feet to meters and the conversion formula from feet to meters are:

```
>convert(5.0, units, ft, m); 1.524          >convert(x,units,ft,m); 381/1250 x
```

Some important conversions:

- *temperature conversions* – the units conversion changes. To convert absolute temperatures from one scale to another must use the temperature option to convert ().

```
>convert(15, units, degC, degF); 27          >convert(0, temperature, degC, degF); 32
```

- *angle conversions* – the trigonometric functions in Maple expect angles in radians. In the Standard environment can simply enter the angle in any units and Maple automatically performs the conversion.

```
>with(Units[Standard]): α := 45 * Unit(deg); α := 45[arc deg]
```

```
>cos(α); √2/2          >evalf(%); 0.7071067810
```

4.3. Commands in the Units package

The package offers 20 different commands. The `with(Units)` command allows work with physical units where a list of options is presented. Commands could be categorized in the following *groups*:

- commands for adding or changing a unit, dimension, system of units (beginning with `Add`):
 - `AddBaseUnit` – add a base unit and associated base dimension.

- >AddBaseUnit(unit, context=unit_context, dimension=dimension_name, opts);
 - *AddDimension* – add a base unit and associated base dimension.
- >AddBaseUnit(unit, context=unit_context, dimension=dimension_name, opts);
 - *AddSystem* – add or modify a system of units. > AddSystem(system, opts, units);
 - *AddUnit* - Units[AddSystem] - add or modify a system of units. > AddSystem(system, opts, units);
- commands for restoring a quantity, information, list of dimensions, units, systems of units (*beginning with Get*):
 - *GetDimension* – return dimension as a product of powers of base dimensions. >GetDimension(dim);
 - *GetDimensions* – list all dimensions. >GetDimensions(opts);
 - *GetSystem* – list units in a system of units. >GetSystem(system)
 - *GetSystems* – list all systems of units. > GetSystems();
 - *GetUnit* – get unit information. > GetUnit(unit, opt1, opt2, ...);
 - *GetUnits* – list all unit names. >GetUnits(opts);
- commands for testing (beginning with Has):
 - *HasDimension* – test whether a dimension exists. > HasDimension(dim);
 - *HasSystem* – test whether a system of units exists. > HasSystem(system);
 - *HasUnit* – test whether a unit exists. > HasUnit(unit, opts);
- commands for changing a dimension, system of units (beginning with Remove):
 - *RemoveDimension* – remove a dimension. > RemoveDimension(dim);
 - *RemoveSystem* – remove a system of units. >RemoveSystem(system)
- commands for setting a context, system of units (beginning with Use):
 - *UseContexts* – set default context(s). >UseContexts(ctx1, ctx2, ...);
 - *UseSystem* – set the default system of units. >UseSystem(system, opts);
- commands for displaying a set context, systems of units (beginning with Using):
 - *UsingContexts* – list the default unit contexts. > UsingContexts();
 - *UsingSystem* – return the default system of units. > UsingSystem();
- commands for deriving output:
 - *Unit* – return argument as a unit in standard form. > Unit(expr);

4.4. Conversion between measuring systems

Maple automatically converts the solution from the given systems and units.

```
>with(Units[Standard]); Dist:=5*Unit(ft)+2.5*Unit(inches); Dist := 1.587500000 [m]
```

It is easier to solve the above problem in the standard US system (FPS: foot-pound-second), using the *convert* command: >Dist:=convert(Dist,system,FPS); Dist := 5.208333333 [m]

4.5. Prefixes in the units of measure

In order to write a set of units in a short form and to avoid lengthy or short names, the System of Units SI and the International Electrical-Technical Commission have approved a list of prefixes that can be used together with the base units. For each set, prefixes can be added to the name of the unit and their symbol can be added to the symbol of the base unit. Prefixes cannot be added to the symbol and the prefix of the symbol cannot be added to the symbol of the unit. Prefixes should be used together with the unit. It cannot be expressed as numerical value. e.g. *k* cannot be represented as 1000. Prefixes cannot be combined to form complex prefixes. e.g. *nanometer* cannot be written out as *millimicrometer*. The prefixes of the units of measure in Maple are given in Table 1. They can only be added to those inbuilt units of measure that allow for that. For instance, if in Maple we operate with meters (m), decimeters (dc), centimeters (cm), millimeters (mm), etc., then after using the *convert* command:

```
>with(Units): convert([m,dm,cm,mm],conversion_table,output=grid);
```

The conversion table is given in Figure 1.

5. Some examples from the mathematical analysis [12]

5.1. Determining velocity and acceleration of an object through a derivative

Find out the acceleration a of a free-falling object, if the dependency of the distance s on the time t :

$$s = \frac{1}{2} \cdot g \cdot t^2 + v_0 \cdot t + s_0 \tag{1}$$

where – $g = 9,8 \text{ m/s}^2$ - the acceleration of gravity; $s_0 = s|_{t=0}$ - value of s if $t=0$; $v_0 = v|_{t=0}$ - value of v if $t=0$.

Table 1

10^n	Prefix	Symbol	10^{-n}	Prefix	Symbol
10^{24}	yotta	Y	10^{-1}	deci	d
10^{21}	zetta	Z	10^{-2}	centi	c
10^{18}	exa	E	10^{-3}	milli	m
10^{15}	peta	P	10^{-6}	micro	mu,u,mc
10^{12}	tera	T	10^{-9}	nano	n
10^9	giga	G	10^{-12}	pico	p
10^6	mega	M	10^{-15}	femto	f
10^3	kilo	k	10^{-18}	atto	a
10^2	hecto	h	10^{-21}	zepto	z
10^1	deka	da,dk	10^{-24}	yocto	y

Figure 1

UnitName	Symbol	To:	m	dm	cm	mm
meters	m		1	10	100	1000
decimeters	dm		$\frac{1}{10}$	1	10	100
centimeters	cm		$\frac{1}{100}$	$\frac{1}{10}$	1	10
milimeters	mm		$\frac{1}{1000}$	$\frac{1}{100}$	$\frac{1}{10}$	1

Solution: 1. The values for of the three known parameters - acceleration of gravity g , distance s_0 and velocity v_0 are entered in Maple in their respective units meter per second, meter and meter per second, as Maple output reflects the unit:

```
> with(Units[Standard]): g:=9.8*Unit(m/s^2); g := 9.8[m/s^2] > s0*Unit(m); s0[m] > v0*Unit(m/s); v0[m/s]
```

2. Determine the dependence of the distance of the time using formula (1) – enter meter per second and second for the initial velocity v_0 and time t .

```
> distance:=0.5*g*(t*Unit(s))^2+v0*Unit(m/s)*t*Unit(s)+s0*Unit(m); v := 4.90t^2 + v0t + s0[m]
```

3. By using the *diff* command twice is received the velocity and acceleration of the object. To differentiate with respect to a variable having a unit – second in this case – use the syntax *diff(function,variable*Unit(unit))*.

```
> velocity:=diff(distance,t*Unit(s)); v := 9,80t + v0 [m / s]
```

```
> acceleration:=diff(velocity,t*Unit(s)); a := 9,80 [m / s^2]
```

5.2. Calculating of power with the help of a defined integral.

The contraction S of a spring is proportional to the applied power F . Find out the work of the power F when the contraction of the spring is by 5 cm, if to contract it by 1 cm a power 10H is needed.

Solution: As set in the condition, the power and move are bound by the following dependency

$$F = k \cdot S \tag{2}$$

The measure units of quantities are as follows: S in meters, F in newtons. The work of the power is defined by the formula:

$$A = \int_a^b F(s) \cdot ds \tag{3}$$

1. Enter the following quantities and their units meter and newton: $S = 0.01\text{m}$, $F = 10\text{N}$:

```
> with(Units[Standard]): F:=10*Unit(N); F0 := 10[N] > S:=0.01*Unit(m); s0 := 0.01[m]
```

2. Calculate the value of k using the formula (2): $k:=F/S$; $k := 1000[kg/s^2]$

3. Determine the power - formula (2): $F:=k \cdot S$; $F := 1000 \cdot s^2[kg/s^2]$

3. Define the work of the power A by using the *int* command– formula (3)

> A:=int(F,s=a..b); $A := (500.b^2 - 500.a^2)[kg/s^2]$

4. Enter the units of the limits a and b of the defined integral

> Values:=[a=0*Unit(m),b=0.05*Unit(m)]; $Values := [a = 0, b = 0.05[m]]$

5. Compute the work of the force using the command eval - > eval(A,Values); 1.2500[J]

In the Standard Units environment, the int function integrates an expression with respect to a name that can have a unit. The result is the integral of the expression, with respect to the variable of integration, with a unit, the integrand unit multiplied by the variable of integration unit.

5.3. Calculating the velocity of radium's decay and its period of half-decay with the help of a differential equation

The velocity of decay of radium is in direct ratio to its amount at each given time. Find out: A. the law of change of the radium's mass m depending on the time t , if when $t=0$ radium's mass is m_0 ; B. the period of half-decay.

Solution: A. The velocity of decay of radium is

$$\frac{dm}{dt} = -k.m \quad (4)$$

where $-k$ is a coefficient of direct ratio ($k > 0$). Equation (4) is a separable differential equation. The sign on the right is a minus, as with passing of time the radium's mass gets reduced. After separating the variables and integrating the two sides of the equation, its general solution is:

$$m = C.e^{-k.t} \quad (5)$$

When using the given initial condition in the problem $m(0)=m_0$, the partial solution is:

$$m = m_0.e^{-k.t} \quad (6)$$

B. For the time t_0 the radium's mass is decayed by $\alpha\%$ from its original mass. Therefore, coefficient k can be defined by the correlation:

$$(1 - \alpha/100)m_0 = m_0.e^{-k.t_0} \quad (7)$$

where $k = -1/t_0 \cdot \ln(1 - \alpha/100)$

Thus it is deduced for the radium that $k=0.000436$ (the unit for time is year). The period of half-decay of radium, i.e. the interval of time for which half of its initial mass decays is defined by (6) for the set value of k and replacing of m_0 with $m_0/2$. There are other problems in physics and chemistry similar to this type of equation (4). The solution with Maple is:

A. DE's solution:

1. Enter the differential equation (4) and by the *dsolve* command its general solution

> with(Units[Standard]): equ:=diff(m(t),t)=-k*m(t); $edu := dm(t)/dt = -k.m(t)$

2. The partial solution of (4) is reached by entering the initial condition:

> IC := m(0)=m0; $IC := m(0) = m0$ > mass := dsolve({equ,IC},m(t)); $mass := m(t) = m0.e^{(-kt)}$

3. To incorporate units into the partial solution specify k , $m0$ and t in the list:

>Values:=[k=0.000436*Unit(1/yr),m0=m0*Unit(kg),t=t*Unit(yr)];

$Values := [k = 0.000436 * Unit[1/yr], m0 = m0 * Unit[kg], t = t * Unit[yr]]$

4. Compute the mass m as a function of t alone by evaluating in on those parameters using the eval command:

>eval(mass,Values); $m(t[yr]) = m0[kg].e^{(-0.000436 [1/yr]t[yr])}$

B. Defining the period of half-decay of radium

1. Replace m_0 with $m0/2$ and t with T in (6): > equ1:=subs(m(t)=m0/2,mass); $equ1 := m0/2 = m0e^{(-kT)}$

2. Solve the equation about the period T : >T:=solve(%,T); $T = \ln(2)/k$

3. Enter the units and value of k : >Values: = [k=0.000436 * Unit(1/yr)]; $Values := k = [0.000436 * Unit[1/yr]]$

4. Compute the period T using the eval command, evaluate the result and convert the hours into years:

>T:=eval(T,Values); $T = 2293.577982.\ln(2)[yr]$ >T:=evalf(%); $T = 1589.787112[yr]$

6. Conclusion

One of the most important issues students should dwell on when solving applied problems are whether the required quantity is physical or non-physical and if the unit of the quantity in question is known or not, etc. Any computation in a specific problem should commence only after all quantities are converted into the right units according to SI, if no other system of units is referred to explicitly. After solving a specific problem they should do a check through the physical dimensions of the output quantities. If all is correct, then this is a guarantee of a right solution. Solving applied problems enhances students' engineering thought, i.e. their ability to observe links between Maths and Physics, on the one side, and various technical applications of these sciences, on the other, also to envisage possibilities for applying this knowledge to practice and to realize scientific ideas on a practical level.

Bibliography

- [1] "A Dictionary of Units of Measurement" <http://www.unc.edu/~rowlett/units/index.html>
- [2] "Index to Units & Systems of Units" <http://www.sizes.com/units/index.htm>
- [3] Engineering and scientific calculations http://www.megabook.ru/pc/encyclop.asp?topic=pc_542&rubr=pc_542
- [4] Kovacheva Ts. "Mathematical packages for teaching and research in Internet applications and information support", "Information [theories@Application](#)", v.11,4,2004, 387-393.
- [5] "Maple Application Center", "Waterloo Maple Software" www.maplesoft.com
- [6] "New packets of the system Maple 7" <http://ais.khstu.ru/Maple7/Glava%2016/Index40.htm>
- [7] "Products and Solution" www.maplesoft.com
- [8] "Reference book for Units" <http://www.bgnews.bg/media.html?media=47549806>
- [9] "The Units Package in Maple" <http://www.adeptscience.co.uk/download/dlddsp/9304/0/All/A4+Maple+Units+Poster.html>
- [10] Units in Mathcad & Maple http://tw.mpei.ac.ru/ochkov/work2_eng.htm
- [11] "SI Units, Conversion, Measurement skill" <http://www.wacek.co.za/>

Author's Information

d-r Tsvetanka Kovacheva – Lecturer at Department Mathematics of Technical University, Bulgaria, 9010 Varna, 1 Studentska str., e-mail: Tsveta_Kovacheva@tu-varna.acad.bg

ABOUT ONTOLOGY APPLICATION TO THE DESCRIPTION OF SYLLABUS

Evgeny Eremin

Abstract: Publication describes the experience in application of ontology technique to structuring of educational materials. Several topics of physics were formalized by means of *Protégé* software tool. Some principal problems in building of knowledge structure were found, so the discussion may interest not only ontology users, but also the developers of ontology tools.

Keywords: ontology, program, education, course, *Protégé*, knowledge structure, objects, inheritance, classes.

ACM Classification Keywords: K.3.1 Computer Uses in Education – Computer-assisted instruction; E.1 Data Structures – Trees; E.2 Data Storage Representations – Object representation; I.2.4 Knowledge Representation Formalisms and Methods – Frames and scripts; I.2.6 Learning – Knowledge acquisition.

Introduction

At present time ontologies – formal descriptions of terms and relations between them in some knowledge domain – are increasingly using for structuring of the comprehensive expertise, accumulated by mankind, and its transformation into computer representation. This method of structure storage already has many functioning complete applications in various areas. Perspective theoretic studies, aimed on support of the correct semantic search in electronic documents (including data selection by net agents) and automatic building of ontologies from