Structure of Development
or
Development and Topology of Biological Shape exemplified on the early embryology in the egg of the nematode C. elegans

Dissertation
zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Christian-Albrechts-Universität
zu Kiel

vorgelegt von
Alexander Krämer

Kiel
Jul 2002

Referent/in: Prof. H.-G. Busse
Korreferent/in: Prof. U.P. Hansen
Tag der mündlichen Prüfung: 07/11/2002
Zum Druck genehmigt: Kiel, 07/11/2002

Der Dekan

Contents

1 Introduction **1**
- 1.1 The worm C. elegans . . . 2
- 1.2 Visualization of images on the computer screen . . . 8
- 1.3 The task . . . 10

2 Devices, Media and Methods **12**
- 2.1 Devices . . . 12
  - 2.1.1 Microscopes . . . 12
    - Inverse microscope . . . 12
      - Type Olympus IMT-2 . . . 12
    - TILL System Time Laps . . . 12
      - Type: Olympus IX70 . . . 12
    - Confocal Laser-scanning Microscope . . . 13
      - Olympus Fluoview 500 . . . 13
      - Leica CLSM . . . 13
  - 2.1.2 Computers . . . 13
    - Personal Computers . . . 13
    - Cluster Computers . . . 13
  - 2.1.3 Device for preparation of worms and video recording . . . 13
  - 2.1.4 Microscopy mount device . . . 14
- 2.2 Media . . . 15
  - Worm Agar . . . 15
  - Egg Buffer Solution . . . 17
- 2.3 Methods . . . 17
  - 2.3.1 Experimental methods . . . 17
    - Pulling capillaries for glass electrodes . . . 17
    - Growth and harvest of the worm . . . 18
    - Stock of C. elegans . . . 20
    - Picking individual worms . . . 20
    - Removing eggs from worms . . . 22
    - Preparation of eggs for microscopy . . . 23
    - Loading of Microscopy mounts . . . 23
- 2.4 Microscopic records . . . 24

2.5 Computational methods . . . 25
2.5.1 Data formats . . . 26
2.5.2 Fundamental data format . . . 27
2.5.3 Tables for records of a database . . . 27
2.5.4 Database of the SIMI BioCell browser . . . 29
2.5.5 Data formats of visualization programs . . . 30
2.5.6 Internal data structure in C and MATLAB . . . 32
2.5.7 Images and image stacks . . . 33
2.5.8 SIMI Browser of BioCell Inc. . . . 33
2.5.9 SIMI analysis module . . . 35

3 Results **37**
3.1 Abstraction of the cell lineage . . . 37
3.1.1 Simplification of the cell linage tree by abstraction. . . . 37
3.1.2 First level of abstraction . . . 43
3.1.3 Second level of abstraction and generating function . . . 46
3.2 Contact matrices . . . 48
3.2.1 Incidence matrix normalized by TDO . . . 56
3.2.2 Reordering by membership to strips and surfaces . . . 58
3.3 Adjacency and distance of cells and their nuclei . . . 58
3.4 Reconstruction of the position of nuclei . . . 65
3.5 First abstraction level for spatial distances of nuclei . . . 67
3.6 Development in 4D . . . 68
3.7 Triangulation of surface (outer cells) . . . 72
3.8 Segmentation . . . 74
3.8.1 Segmentation results . . . 74
3.9 Coding the topology . . . 77
3.10 Spring embedding . . . 78
3.11 Physical space of chains (topology) . . . 78

4 Discussion **83**
4.1 Goal of the work and cell properties . . . 83
4.1.1 Cell cycle events . . . 85
4.1.2 Movement of cells - Migration . . . 87
4.2 The cell-lineage pattern of C. elegans . . . 88
4.2.1 Mutants of spatial position and lineage patterns in early embryology . . . 89
4.3 Experimental records of the development in the egg. . . . 91
4.4 Visualization of the database records . . . 92
4.5 Temporal and spatial abstraction . . . 93
4.6 Structure inherent in the adjacency . . . 94
4.7 Reconstruction from a distance matrix . . . 95
4.8 Development in 4D time-space coordinates . . . 97

4.9 Developmental potentials, forces and fluxes 98
4.10 Algorithm, Data Structures and Convex Hulls 100
4.10.1 Computational Geometry 100
4.10.2 Algorithms, data structures and access functions 101
4.11 Outlook 103
4.11.1 Relation to cells of the immune system 104

5 Appendixes **106**

A Simplex computation **107**
A.0.1 Simplices and simplicial complexes 107
A.0.2 Delaunay triangulation of points and volume of simplices . 111
A.0.3 Computation of areas in Voronoi tessellation 114
A.0.4 Volume computation of territories in the hull of a Voronoi tessellation 115

B Triangulation, tessellation and vector-chains **117**
B.0.1 Delaunay triangulation and Voronoi tessellation as chains . 117
B.0.2 Orientation of simplices and chains 120
B.0.3 Boundary of chains and co-chains in simplicial complexes . 121
B.0.4 Matrices of boundary operators and network theory 123
B.0.5 Structure of discrete and contineous Systems and memory 128
B.0.6 The coding of a simplex complex by stripes 130

C Segmentation **133**
C.0.1 Normalized Cuts 133
Graph partitions and normalized cuts 133
C.0.2 Color, texture and conture segmentation 135

D Videos **137**
D.0.1 Growth of an embryo in the egg until hatching 137
D.0.2 Preparation procedures of an egg from the alive worm . . . 139
D.0.3 Demonstration of the rigidity of an egg 143

E Electronic storage media **144**
E.0.1 Thesis 144
E.0.2 Reproducible research 144
E.0.3 Browser and database 145
E.0.4 Suplemental programs 145
Programs 145
Matlab Code 146
C Code 146
E.0.5 Utilities 146
Matlab Code 146

C Code 146
E.0.6 Miscellaneous 146
Matlab Code 146
C Code 146

G Literature **147**

H Summary **154**

Chapter 1

Introduction

After the disclosure of whole genomes in the last decade of the second millennium there are only a few biological secrets left to be uncovered. The function of the brain and the development of biological organisms are two examples. In the development of organisms, two major disciplines, the phylogeny and the ontogeny, are distinguished. The phylogeny describes the development of the organism under pressure the of adaption into their environment over a long period of time, much longer than their life span. The information gained to survive in a given environment is stored in the genome and preserved by that for future generations. The information is passed on in the germ line of the organism from genome to genome. The individual organism serves as a protecting hull (body) of the germ genome to keep and maintain the information in the environment for the lifetime of the body and to pass it on to its progenies. Apparently the memory of the brain has a similar task regarding the information gained in the society.

The ontogeny is the development, or more precisely, the self-development of the protecting body (the soma) of an individual organism. The main process in the ontogeny is the cell division in its different forms. For archae-, pro-, and eucaryotic single cell microorganism, it is the only event which can forward the information of the genome to future generations. However, in an inhospitable

environment, such organisms have developed several strategies to survive. One of the most advanced is the generation of multicellular organisms, in which the cells can differentiate and perform different tasks in coordinated ways.

At the time the ultimate development is coordinated by the brain, which allows the formation, development and interaction of an (individual) organism and of a society.

The self organization of the body of the multicellular organism and its adaption into the environment are still not satisfactorily explained by the classical science of nature [Sla83]. Also the mechanism that leads to the decay of the body is not known sufficiently. To study such processes, one usually has to choose the right object at first and secondly to apply the right method to it, before the mechanism of nature that drives these processes of live can be understood.

Among others the nematode Caenorhabditis elegans (a small worm) and the fruit fly Drosophila melanogoster have been studied according to these principals. In particular, after the molecular biology has allowed the manipulation of the genome, the study of gene function in developing organisms has made progress (e.g. gene clusters as the homeo box clusters see [KG90]). Whereas the fruit fly Drosophila is already a quite complex and huge organism compared to the worm C. elegans, the later has several advantages over the fly in the study of its embryonic development. The most important of these seems to be the invariant cell lineage of the wild type and the small and constant number of cells, which constitute the individual. Therefore, C. elegans has been selected for this study.

1.1 The worm C. elegans

The adult nematode Caenorhabditis elegans (C. elegans) is a worm of the length ≈ 1.3 mm and the diameter $\approx 1/20$ of its length. It exists as a female hermaphrodite and as a male (see figure 1.2 and the movie 1.1)

Figure 1.1: *The video freeze image shows a C. elegans population under the microscope at 40× magnification. Nematodes of all types (hermaphrodites and male) and all developmental stages (eggs L1 to L4 larvae and adults) are worming on the agar layer (≈ 4mm thick). The big worms are adult hermaphrodites, whereas the small ones are larvae stages and males (appear at low frequency). The agar surface is humidified (with egg buffer see section 2.3.1) such that the worms move transparent on the surface but do not swim.*

Essentially, the life cycle (see figure 1.3) consists of six phases: the embryogenesis in the egg (≈ 14 h), which ends at the hatching from the egg, subsequent four larval stages of a duration of ≈ 12h, 7h, 7h, and 9h, each ending with a skinning, and the final adult stage (5-20 days) that can produces more than 1000 eggs. The egg stays up to the ≈30 cell stage (or more, depending on the environmental conditions) in the uterus of the worm (see movie in figure 1.4).

At hatching, a L1 larva is ≈250 μm long and consists of 558 cells in the female, and 560 cells in the male worm. The embryogenesis (i.e. the phase from zygote to hatching) in the egg may be sub-divided in three more phases:

Phase 1: the creation of embryonic founder cells and the definition of the embryonic axes (of orientation) ($\approx 2h$).

Phase 2: cell proliferation, migration including gastrulation ($\approx 5h$).

Phase 3: morphogenesis, differentiation and organogenesis ($\approx 7h$).

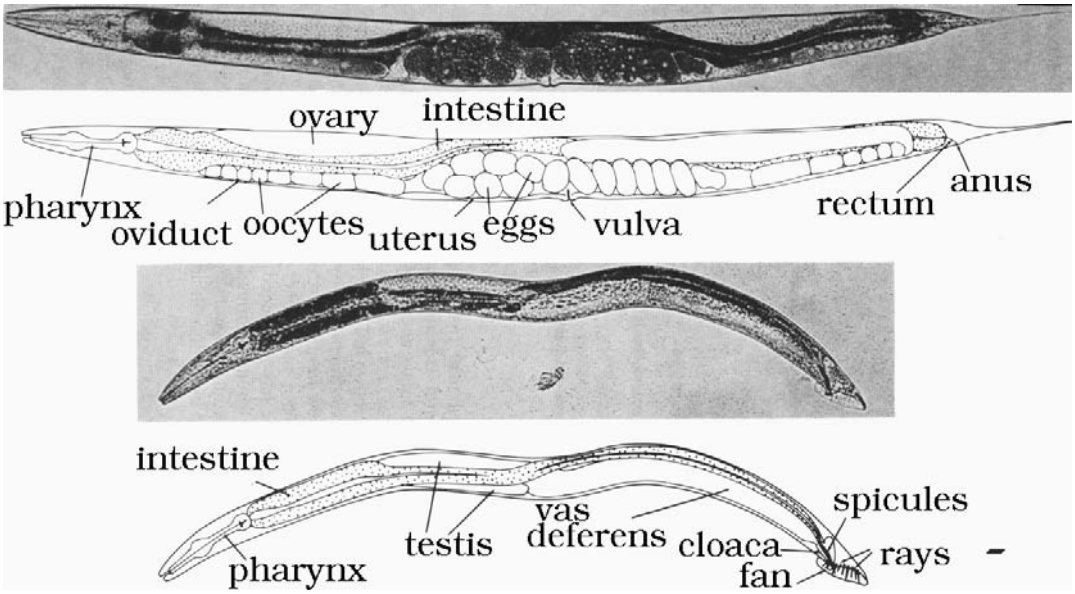


Figure 1.2: *Micrographs of an adult hermaphrodite (upper figures) and a male (lower figures) of C. elegans. The corresponding graph displays the anatomical features of the body plan. The major organs are indicated (see [Woo88]). The differences in the body plan originate mainly from the big part of the sexual organs. The width of the males micrograph is* ≈ 1 *mm (from [SH77]).*

After 7 hours (in middle of embryogenesis), the embryo is still a spheroid of $\approx$ 550 cells.The cell proliferation ceases at this time and the elongation of the eggs cell mass as well as differentiation starts. In particular, the neuronal interconnection is established.

Within $\approx$ 20 minutes after fertilization the egg forms a rigid shell around itself, which consists of an inner vitellin membrane (mostly impermeable to hydrophilic substances), a chitinous layer in the middle and an outer layer of lipids and collagen. An egg has a length of $\approx 60\mu$m and a diameter of $\approx 30\mu$m. The first cleavage takes place $\approx$ 35 minutes after the fertilization. Then, the founder cells (AB, C, D, E, MS) (see figure 1.5) are split off the germ line sequence (P_0, P_1, P_2, P_3, P_4 and Z).

The cell division of the founder cells into daughter cells of unequal size and of asymmetric morphology is a characteristic, which is seen in stem cell patterns in other organisms also (smaller germ-line and larger founder cells [zS06]). In the

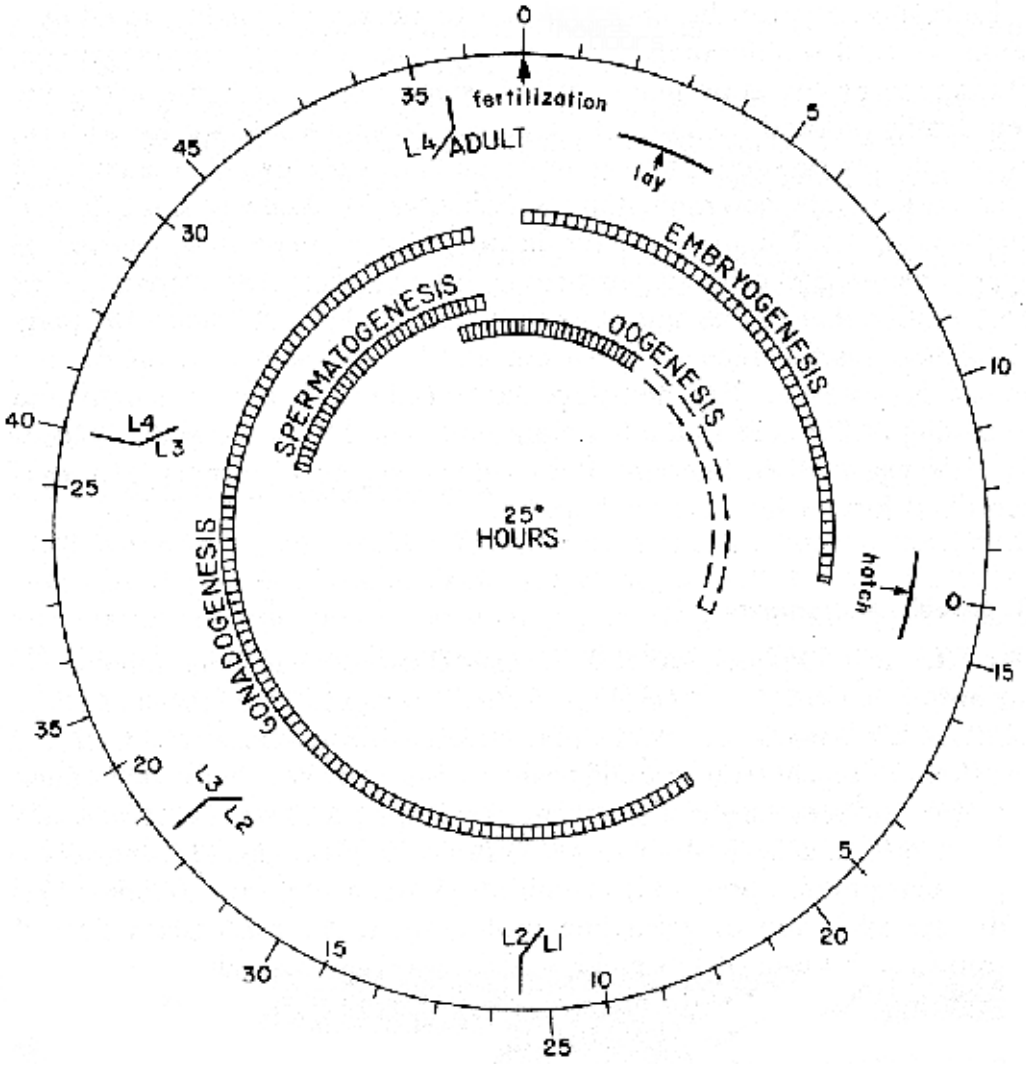


Figure 1.3: *Life cycle of C. elegans displaying the approximate time course of the 4 larval stages L1, L2, L3 , L4 and the embryogenesis in the egg, which is subdivided into two nearly equal phases: a pre-morpho-genetic one of* $\approx 7h$ *duration, in which proliferation and migration of cells dominate, and the morpho-genetic one of* $\approx 7h$ *duration in which the early worm is build. Numbers at the outer circle are approximate hours at room temperature (25 °C) after fertilisation. Numbers at the inside of the circle indicate hours after hatch. The larval stage L3 may be bridged by a dauer stage which prolong the life cycle substantially. (from [Woo88] and [WHC80])*

Figure 1.4: *The video freeze image, which shows a female in the moment of laying an egg in the* $\approx$ 60*-cell state. The egg is just leaving the uterus through the vulva. In this state, the embryo inside the egg is able to complete its development outside the uterus. Magnification* 200$\times$*.*

notation of cell-names, the founder cell carries capital letters and the progenies (descendents) are named by adding a lower case letter to the parent name (with C. elegans the letters : a = anterior, p= posteriori, l=left, r=right, d=dorsal, and v=ventral according to the interior coordinate system, e.g. ABp is the daughter born posterior from founder cell AB). By this convention, all cells in the cell lineage are identified and their axes of division is specified roughly. After some cell divisions the phase of gastrulation begins. At this time the egg consists of the following progenies: 16 from AB, 4 from C, 4 from MS, 2 from E, and of one D and P_4. Now, $\approx$ 100 minutes have passed since the first cleavage (see figure 1.5). The phase lasts for $\approx$ 150 minutes, until the $\approx$ 360 nuclei state has been reached.

Beginning in the late gastrulation, there are many programmed cell deaths (apoptosis). The apoptosis of the cells has been discussed in several papers (e.g. see [WC02]). Cell death by apoptosis is an invariant of the cell lineage and a cell-autonomous event. Note that cell death is not always bilateral in space inside the egg.

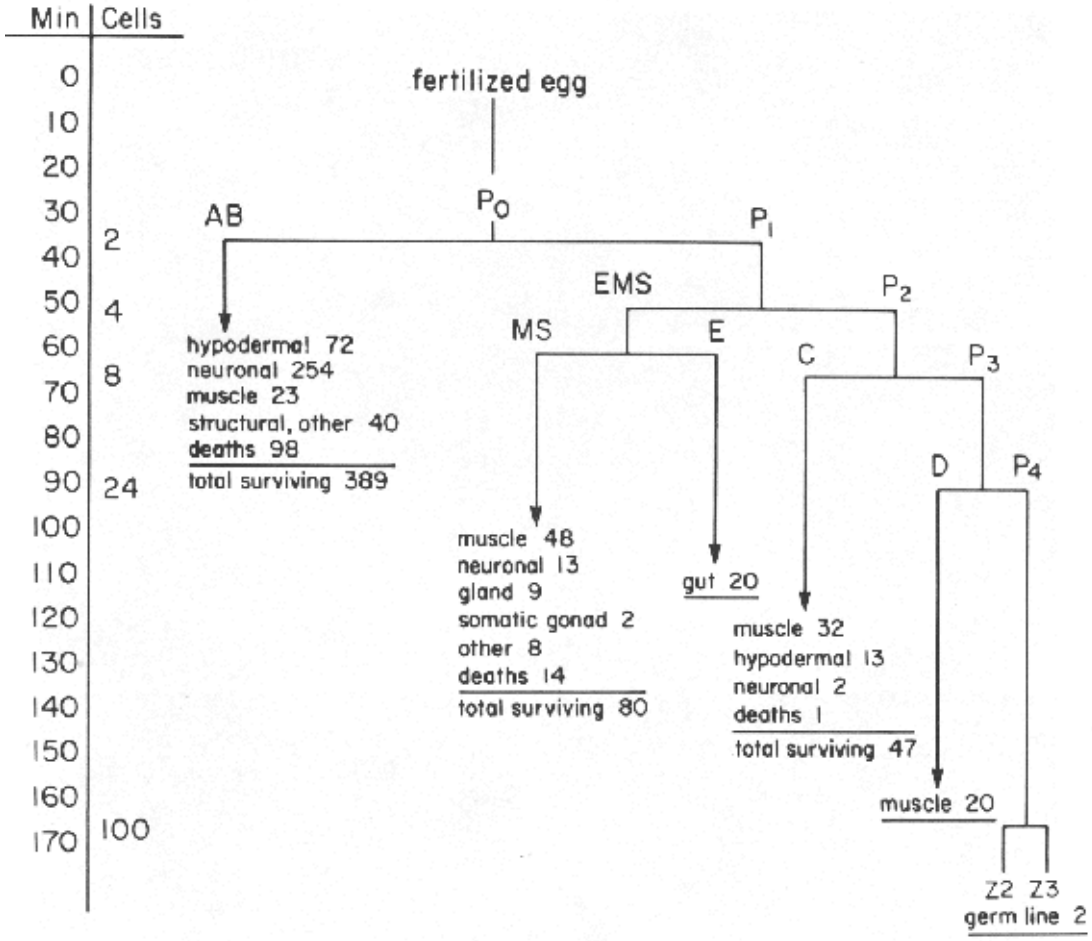


Figure 1.5: *Early cell-lineage pattern in the egg of the nematode C. elegans. The derivation of the five founder cells AB,MS,E,C,D and P_4 is displayed and for each founder cell at hatch, the number and the fate of the progeny are given. The left scale shows the approximate time after fertilization at 25 °C and the number of cells present at times of low cell division activity (from [SSWT83]).*

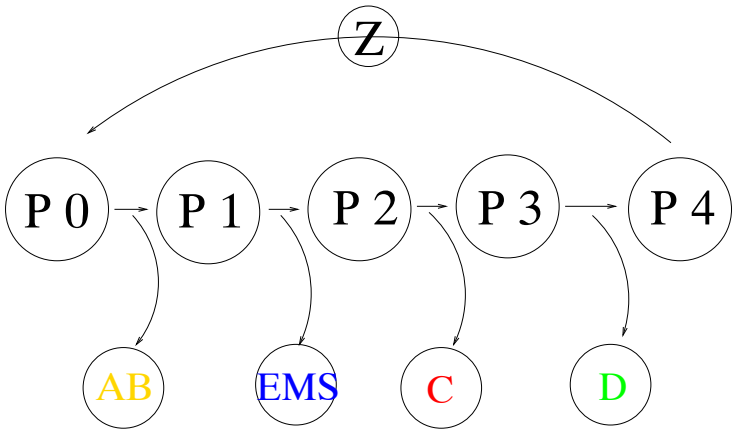


Figure 1.6: *Propagation of genetic information in the germ line from the P_0 to P_4 cell and on to the Z cells, which reproduce the eggs P_0 cell and the generation of the founder cells.*

In the morpho-genetic phase, the spheroid of ≈ 550 cells lengthens (folds) in the egg into a wormlike shape (see figure 1.7). The shape of the worm is finally stabilized by a tight cuticle surrounding it before hatching (see [PH86]). There are many mutants available for the study of defects (see e.g. [Woo88] page 497-559) of various types. The many lineage and pre-morphologic mutants would be of special interest for this work (see [Woo88] page 528-532). The mutants may be genotypic or phenotypic. Such mutants are of particular interest to the study of the function of genes by molecular biological and genetic methods. Since the complete genome has been sequenced (see www.wormbase.org or elegans.swmed.edu), the mutations may be located on the genome. For a review of the literature, see [Woo88] and [HM02].

1.2 Visualization of images on the computer screen

Programming and visualization environments to display scenes on computer monitors have been developed in the last years considerably. Many of them are avail-

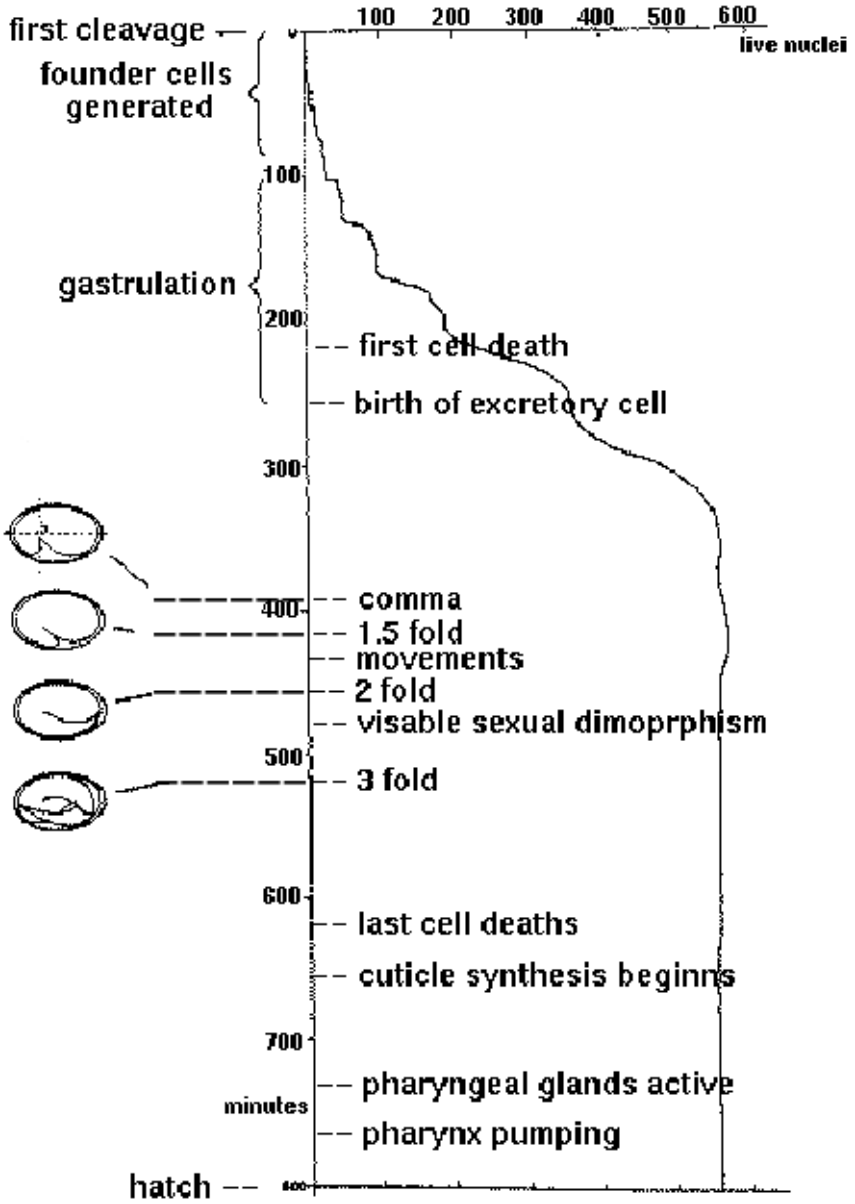


Figure 1.7: *Number of nuclei during embryogenesis in the egg of the nematode C. elegans. The stages of the development are indicated on the time scale. (from [SSWT83])*

able free of charge from the Internet (e.g. IBM Data Explorer "openDX" [IBM02] or the visualization toolkit "vtk" [SML98] and others). The algorithm in the packages may be classified in those that act on images (raster images) and those that are based on geometrical background. The former algorithm are used to edit (2D and 3D pictures e.g. movies), the later to construct and display a scene (e.g. of geometrical objects in engineering). Between the two packages, algorithms may be placed which analyze images for segments (segmentation) or extract objects from them. These algorithms are important to the dissection of the egg into the cells it is composed of, the cell nucleus and other substructural elements. Whereas segmentation is still in an incomplete state, the visualization by geometry is quite advanced. Here, the distinction between the topological and the geometric elements of a scene has facilitated the development of algorithms. The topology gives the relations of objects in space (or time) without assigning a position to them (i.e. it states only if an object is behind or adjacent to another). The geometry places the objects in their specific positions in space, such that the topological relations are still valid. (For details on this topic, see [SML98] where also the use of homogenous coordinates is treated.)

1.3 The task

In the pre-morpho-genetic phase in the embryology of C. elegans, cell division and migration are the major visible events. Both events support the cell sorting process, which brings the cells from their place of birth into their predefined position (destination), from which morphogenesis to a worm starts. In the process of sorting, the cell has to know its present position and its destination. Sorting as such does not need an absolute position, but an ordering of the cells relative to each other. The question addressed in this work is: what is needed to sort the cells to take the order in the spherical cell mass at the beginning of the morpho-genetic-phase. This specific order (relation) is the prerequisite for the inner cell

structure, which determines the outgrowth of the shape of the organisms.

For this purpose the database of C. elegans wild-type [SHMS97], where division times and spatial positions of nuclei are recorded, should be used to analyze the development of cell contacts in the pre-morpho-genetic phase up to the state of stable contacts (sorted cells) at the end of the pre-morpho-genetic phase. In particular, it should be examined, if the development could be simplified by reconstruction and abstraction (i.e. neglecting details).

The topological and geometrical structure inherent to the cell mass should be analyzed for a basis in which cells may sort and forces may lead them to their places of destination. The development as a spatio-temporal phenomenon should be considered in a 4 dimensional (4D) space-time concept, in which the dynamic development becomes static. This should be done as close as possible to reality.

Chapter 2

Devices, Media and Methods

2.1 Devices

2.1.1 Microscopes

The following microscopes were used for data acquisition from mounted specimens.

Inverse microscope

Type Olympus IMT-2 manufacturer Olympus, (for device setup see figure 2.1 and section 2.1.3).

TILL System Time Laps

Type: Olympus IX70 manufacturer Olympus, equipment: Piezo Stepper for Z-focus, CCD camera: Imago 1/2" 640/480, Polychrome IV, Software: TILLvision, contrast method: bright field hoistway light (WI-OBCD - condensor, Olympus), Objective: UPLFL 40x

Confocal Laser-scanning Microscope

Olympus Fluoview 500 Manufacturer: Olympus. Type: BX61, Objectives: 1# UPLAPO60xW and 2# UPLAPO40x, Laser: HeNe-R : $633nm$, contrast method: differential Interferenz-contrast Normarski (DIC), Software: Olympus Fluoview, with 12 bit photomultiplier (4096 channels i.e gray scale values).

Leica CLSM Manufacturer Leica microsystems, type (TCS-SP) wavelengths $(351, 360, 458, 476, 488, 514, 543, 633nm)$ Software: LEICA TCS Service Pack 1 V.258 (see figure 2.4).

2.1.2 Computers

Personal Computers

Intel Pemtium machines with 300Hhz or 500Mhz processors with 700MB to 800MB Ram. Operating Systems: Suse Linux version 7.2 and 7.3, Microsoft Windows 98 and 2000 Server, CODA virtual fileserver system version 5.3.19.

Cluster Computers

Siemens HPCLine, 8 knots with two processors at each knot. Processor type: Pentium III at 650Mhz. The memory is 512 MByte at each knot where both processors access the memory. Hard drives: $8\times$ 20 GB. Inter machine communication network: SCI, Scali SSP 2.1.2. Operating System is Suse Linux version 7.3. The machine is used as a "compute-server" for sequential and parallel applications.

2.1.3 Device for preparation of worms and video recording

Worms are prepared under the microscope (see figure 2.1) with two micro-electrodes. Each micro-electrodes (see section 2.3.1) is attached to a micro-

manipulator (Narishige: MO-303, Serial:9203027). The micro-manipulators are attached to tripods. Microscope and tripods are fixed on a work dish which stands on a flexible tube (diameter of inner tube $\approx 1.5m$) filled with water for shock and vibration resistance. The protection against vibrations is required for data acquisition free of blurriness.

A video camera (JVC: TK-1280E) attached on top of the microscopes was connected to a video recorder system (Panasonic Time Lapse AG 6730) and a computer video processor system (Pinacle: Studio PCTV Rave).

Figure 2.1: *Experimental device setup for video microscopy. To the left, the recording computer, in the middle of the figure, the microscope with video mount, and to the right, the video recording. (scale: hight of computer is 38 cm)*

2.1.4 Microscopy mount device

The mount device carries the egg during observation and provide the necessary condition for development. It is made of two microscopy slides (AL 24 × 32 mm Nr.9.161 032) separated by more longish splinters (to avoid squirt of glass splinters, fold a piece of paper around a glas-slide, crack the paper loader by

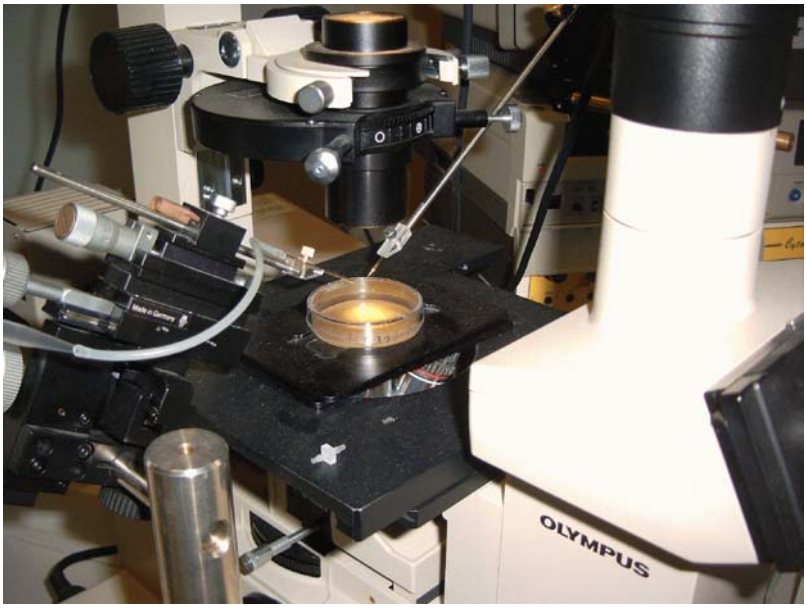


Figure 2.2: *View of the microscope of figure 2.1 with a petri dish in the center (ø5.5 cm) and a micromanipulator to the left.*

forcing the slide with the thumb to curve over an edge of the your desk – unfold the paper and pick with a tweezer longish splinters) (see figure 2.3).

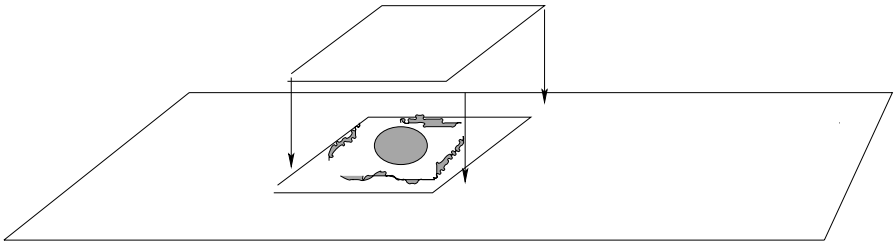


Figure 2.3: *Scheme of a Confocal Laser Scanning Microscope mounting device.*

2.2 Media

Worm Agar

Cultures of C. elegans were grown on petri dishes ($\varnothing = 5.5cm$ in diameter from Greiner, Solingen) prepared by a modified procedure from the literature [Woo88]

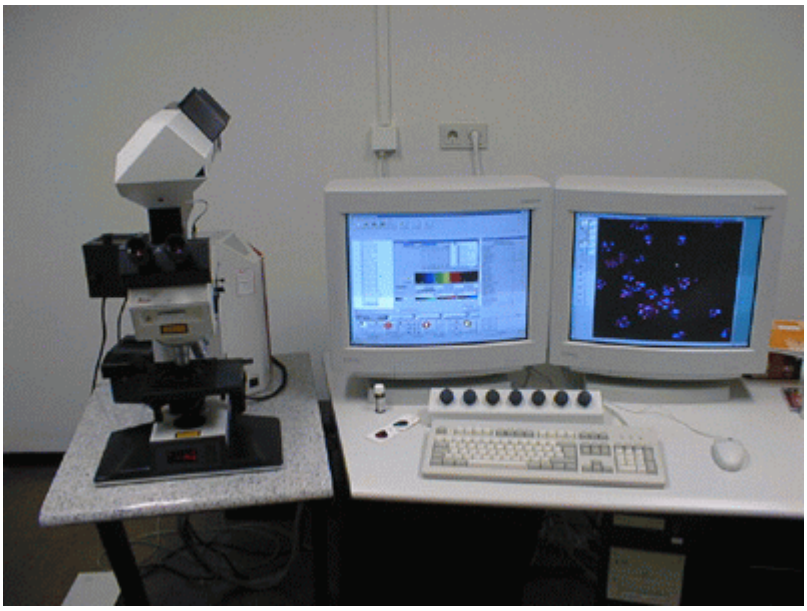


Figure 2.4: *Image of the Leica Confocal Laser Scanning Microscope (CLSM). The microscope is places on the left side on a schock resistant work desk. The computer monitor in the middle is used to control the microscope operation. The monitor to the left displays the processed data. (Image by Dipl. Ing. Pfeiffer, Central Microscopy Lay, CAU Kiel)*

where alternative methods are found also:

Dissolve the following substances in de-ionized sterile H_2O and autoclave at 131^oC and 2 bar for 20 minutes .

- $3g$ $NaCl$ (Sigma,St Louis,USA,Item S-3014)
- $17g$ Agar (Fluka,Buchs,Swiss,Item 05040)
- $2.5g$ peptone (Gibco BRL Item:30392-021)
- $1ml$ of cholesterol ($5mg/ml H_20$) (Fluka,Buchs,Swiss,Item 26740)
- $975ml$ H_2O

Add the following autoclaved sterile solutions and mix after each addition:

- $1ml$ $CaCl_2$ (1M) (Merk,Darmstadt,Germany,Item 2389)
- $1ml$ $MgSO_4$ (1M) (Sigma,St Louis,USA,Item M-9397)
- $25ml$ K_2HPO_3 (1M) (Merk,Darmstadt,Germany,Item 4873)

Pour the hot agar (cooled down to $\approx 70^oC$) into petri dishes ($\varnothing = 5.5cm$ from Greiner) in an airflow bench (ceag shirp D424) to a $\approx 4mm$ thick layer. Let the agar in the petri dishes cool down while it is half covered by the lid in the bench to avoid moisture on the lid until the agar is solid. The petri dishes are kept in a refrigerator stored upside down until inoculated with worms.

Egg Buffer Solution

Prepare a 5% weight/volume solution of agar in H_2O: Take an arbitrary volume of solid sterile agar from the agar preparation (see section 2.2 worm agar) – preferably a small piece from a layer in a petri dish e.g. ($1cm^2 \times 4mm$) thick – as 1/20 and add 19/20 volume of H_2O.

Heat the suspension on a plate (IKA GmbH, Staufen, Type:RTC-B Item:00.12.1663) with intense stirring (use max scale setting) to increase the O_2-concentration, until the solid agar is completely dissolved. Otherwise, the O_2-concentration is too low to grow eggs in $20\mu l$ buffer on the mount device (see section above 2.3.1). If the buffer concentration is too low (i.e.$\leq$ 2% of agar in H_20), the eggs may burst after being exposed into the buffer media (see movie figure D.10).

2.3 Methods

2.3.1 Experimental methods

Pulling capillaries for glass electrodes

The turnkey factor are electrodes ready for use and made, without any post processing steps (e.g. like polishing edging).

The glass electrodes are made from boro-silicate glass capillary tubes with filament (Hilgenberg: article number 14 032 01) using a microelectrode puller (HSE -Huge Sachs Elektronik- cd: type 801 device 8328). The capillary tubes

are of the dimensions: length $l = 100mm$, outer diameter $\emptyset_{Da} = 1.50mm$, inner diameter $\emptyset_{Di} = 1.018mm$, wall thickness $da = 015mm$. The puller settings are: heating off:$HA = 1.45$, magnet on $ME = 2.95$, magnet off $MA = 1.98$, the heating energy 5.0 scale units equivalent to a current of $A = 10.1$ Ampere and a magnetic force of 5.1 scale units. These parameters can but do not necessarily produce electrodes with a tip opening of $\emptyset \approx 10\mu m$ diameter.

Changing the heating energy, one may also get electrodes suitable for injections into cells or for egg handling. The applied change is so small $\pm 0.05A$, that it hardly can be expressed in scale units. The way to proceed is: Position the heating energy button just below the limit between closed and open electrodes. The first runs will produce electrodes with closed tips (see figure 2.5). Every run erodes the heating coil in a way that the wire becomes thinner by chemical reaction with the rooms atmosphere (it wears out). This results in a thinner wire, thus the resistance of the wire increases a tiny amount and so does the temperature. The change of heating temperature modifies the electrodes in the desired way. In a series of $\approx$ 30 runs, all of the necessary electrode shapes are designed since the capillary tubes melting behavior is a characteristic property of the material.

Growth and harvest of the worm

A culture of C. elegans type N2 is obtained from the Schnabel Laboratory [Sch] on agar in a petri dish. Cultures are passed on by transferring a few mature worms (three or four) and some small worms (six to eight) under a stereo microscope (WILD Heerburg/Leitz No.: 173889) with a hand held electrode. The worms are picked up on the electrode tip and placed immediately and gently on a fresh agar plate (see section 2.2). Within a few days of growth, a population of worms is seen. In early transfers the agar surface was coated by a layer of bacteria (E. coli strain LE 392). This turned out to be unnecessary since by the former

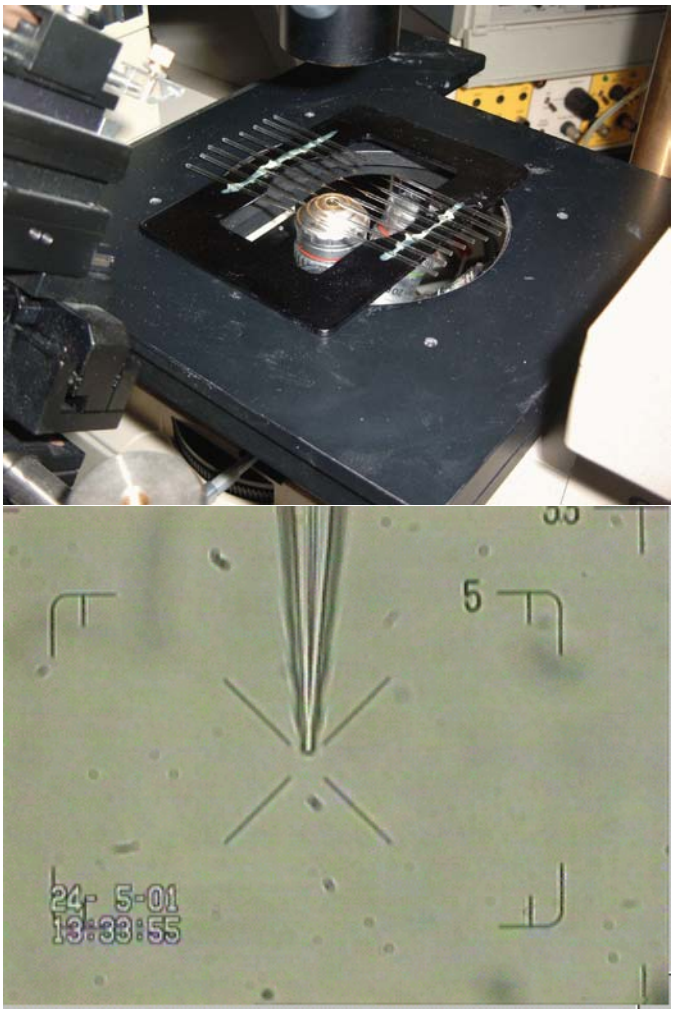


Figure 2.5: *Upper figure: A series of electrodes mounted on the dish of the microscope. Lower figure: The tip of a glas electrode under the microscope (magnification* $40\times$*). The video linked with the bottom image, demonstrates the erosion with the increasing number of runs of the heating coil (i.e. increasing heating) of the puller and the change of the electrodes shape.*

method, enough bacteria attached to the worms are transfered to allow growth of C. elegans. A few hours later some drops (three to five) of LB media (Luria-Bertani medium [Se89]) may be added on the agar surface to keep such a degree of humidity that the worms do not swim in LB but become transparent for light microscopy.

Stock of C. elegans

Petri dishes with a population of C. elegans on agar (see section 2.2) where kept in a refrigerator at 4 ^oC for several months. They served as stocks from which new cultures were prepared if needed.

Picking individual worms

The population on agar in a petri dish is observed under the microscope Olympus IMT-2 (see section 2.1.3). The movie of figure 1.1 shows, how a population is searched for a worm that is harboring eggs (usually one or a few). Firstly, the population is scanned with a 4× magnifying objective. An interesting worm is examined closer with a 20× lense (see movie of figure 2.6).

Figure 2.6: *The video freeze image shows a search for an egg in the early 1- or 2-cell state inside a living worm. At the center of the image, an egg in the one cell state is visible. Magnification* 200×.

As described above the agar surface should be humid since worms become transparent and are easy to scan for eggs, if they are kept in liquid. Individual worms are picked with a hand held electrode, and immediately transfered into a previously prepared drop ($\approx 150\mu l$) of egg buffer solution in a petri dish for later

microscopic study (Olympus-IMT2 see section 2.1.3). In this petri dish (Greiner: $\emptyset = 9cm$ in diameter) the bottom was replaced by a $0.14mm$ thick microscopy glass disk. This allowed the use of a $100\times$ lense with oil (Zeiss Immersionsoel 518C DIN 58 884) on the inverse microscope.

The microscope is equipped with an optical system to revert the left and right hand side motions (i.e. moving the electrode to the right it actually moves to the right in its image also and not to the left) to avoid the extra skill required to hunt a worm. The transfered and fixed worm is shown in a movie of figure 2.7.

Figure 2.7: *The video freeze image shows the procedure how to capture a living worm between two electrodes on the bottom of the petri dish for further treatment.*

It seems that the egg production may be enhanced by placing three drops of egg buffer on the surface of an agar plate (just wetting the surface) taken directly from the refrigerator ($+4^oC$). Too much liquid results in difficulties is catching a worm. About two hours later (the desired early stage) egg-bearing worms are usually observed. Note, isolated worms are easily fixed between and prepared with micro-electrodes when they are cooled down since the almost do not move at $+4^oC$.

Removing eggs from worms

There are methods to prepare eggs in masses e.g. using $NaOCl$ and KOH (see [Woo88] page 603). However, individual preparations is employed here and in part documented by videos (see appendix D). They demonstrate the handling in more detail than a verbal description can do and may be explored if needed.

When the desired worm is placed in the drop of egg buffer solution on the petri dish for microscopy (by the IMT-2 microscope) without proper care, the worm may be damaged or even cracked by unrolling (or touching) the electrodes tip on the bottom of the dish. The tip of the electrodes may brake and should be removed immediately from of the liquid since capillary forces suck egg buffer and particles (egg, specimen) into the electrode (the electrodes splinters may remain at the scene). If the worm is damaged and the eggs are set free, follow section 2.3.1 (see movie in figure D.4). In case the worm is still alive and moving around, injection electrodes are attached to the micro-manipulators. The electrode tips are positioned in a way that both tips are grounded on the bottom of the petri dish and touch each other in focus. A piece of glass splinter may be used as target performing this pre-positioning procedure. The electrodes are separated a little so that a worm almost can pass through the space (a triangle) spanned by the two tips and the bottom of the dish. Now the petri dish is moved in a way that the worm is placed exactly in this triangle (see movie in figure 2.7).

A slight forward thrust of one electrode fixates the worm. After carefully selecting where to cut (e.g. 1/4 from the head or tail of the worm) the worm is dissected with a reasonable force applied to the tripods carrying the micro-manipulators. The worm breaks immediately or after a short while producing tissue, leaving the worm though the cut under pressure (see video sequences in figures 1.4, D.7, D.8 and D.11). The eggs are expelled (see movies in figures D.4, D.5 and D.6) and fall to the bottom of the dish after further tissue dissection. They are examined with a 40× lense or even using a 100× lens with

oil (see movies in figures 2.9, D.2, D.3 and D.12).

The egg of interest is finally drawn into the electrode by capillary forces or under support of an attached Eppendorf pipet (see movie in figure D.9). Note, that in pure H_20, the worm may burst and eggs may be damaged as well (see movies in figures D.10 and D.11).

A video shows how the eggs contained in the tip of the electrode are released onto the prepared mount device (see movie in figure D.9).

Preparation of eggs for microscopy

Eggs in the petri dish are collected using electrodes (see section 2.3.1) with an opening of $\approx 100\mu m$ diameter. Capillary forces imbibe the eggs with egg buffer into the electrodes. Liquid imbibed can be pressed out again, by flexible tubes attached to the electrodes. The specimen contained in the electrode can now be transfered to the microscopy mounting device (see section 2.1.4) or could be frozen (for applying freezing procedures see [Woo88] page 589).

Loading of Microscopy mounts

A $20\mu l$ drop of egg buffer is placed in the center of the lower slide framed by glass splinters (see scheme of figure 2.3). After the egg is released into the drop of buffer, the second slide is placed gently on top of splinters. Now mineral oil (Sigma Mineral Oil: Item M-3516) is supplied to the corners of the sandwich seen in figure 2.3 to seal the buffer against evaporation. Mineral oil is not toxic to the eggs and still allows oxygen to diffuse into the buffer. The lower and the upper microscopy slide finally are glued together at two opposite corners using a tiny drop($\approx 5\mu l$) of glass adhesion (UHU sekunden alleskleber). The glass adhesion is separated by oil from the egg buffer.

In Confocal Laser Scanning Microscopy sessions (see section 2.1.1), the objective imposes a hydraulic pressure onto the top slide when used with immersion oil

and moves the egg out of the focus point. Therefore, the mount device is additionally stabilized with a slot aperture (see figure 2.8). The material of the aperture is copper evaporated with gold, the dimensions are: disk radius $r \approx 1.5mm$; hight: h=60μm and the slot measures $1 \times 1/10mm$. The slot aperture is centered as seen in figure 2.3 between the lower and the upper cover slide. The egg is placed into and kept throughout the microscopy session in the aperture slot.

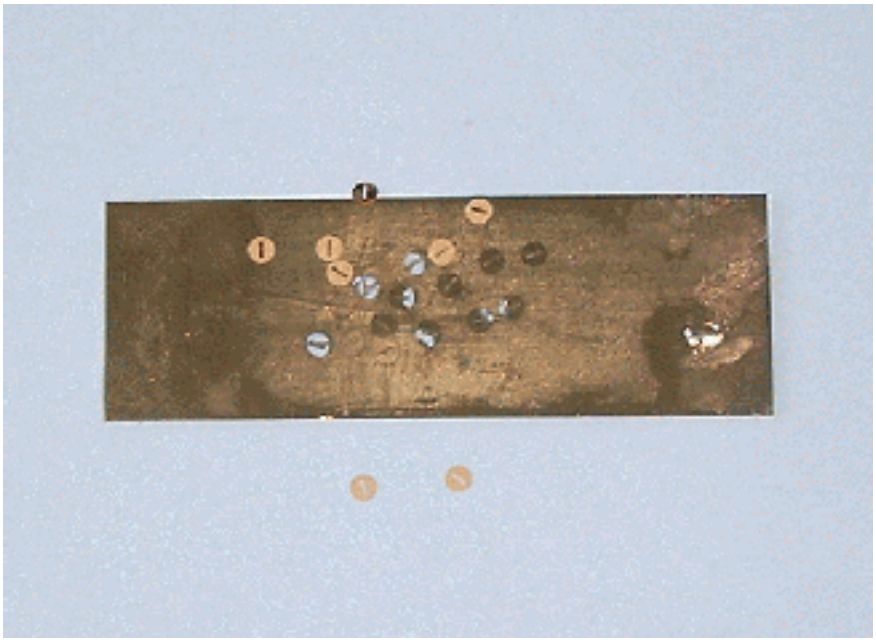


Figure 2.8: *The image shows a set of slot apertures on an object slide evaporated with gold. During preparation the slot of the apertures is filled with an egg in buffer.*

Since the egg develops fast (i.e. only a few minutes separate early cell divisions), the mount device is placed under the pre-configured microscope (ready for action) immediately and data acquisition is started.

2.4 Microscopic records

An egg taken from the nematode is placed in a microscope (Nomarski Differential Interference Optic or Laser Scanning Confocal Microscope (see section 2.1.1)), is

Figure 2.9: *The video freeze image shows an egg in egg buffer on a microscopy mount device. The embryo is in an* ≈ 100 *cell state at room temperature. Under these conditions the egg usually develops until hatching. Magnification* $1000\times$.

≈1500 times enlarged and scanned in space and time (see movie 2.10).

The microscope x-y images at a defined z position of the egg are stored as a 3D stack of 2D x-y digitized images (of adequate resolution with 256 gray scales or 4096 gray scale colors depending on the microscope device (photomultiplier) used). A time sequence of such (≈1700 in ≈5 to 7 hours) 3D-stacks form a 4D data set i.e. a movie in 3D format. The data set represents the development of an egg from its zygote (one cell) or a few cell state until prior to hatching (≈550 cell state). An AVI video file can be found on the Internet (see [SHMS97] or in E.0.1). The movie in figure 2.10 shows how the egg develops a cell structure up to morpho-genesis of the worm state in one particular z-plane.

2.5 Computational methods

The microscopy data are processed with a variety of computer programs E.0.4 designed for this work. Special interfaces facilitate the interplay and usage with already available software as e.g. the comfortable browser SIMI (of BioCell Inc.)

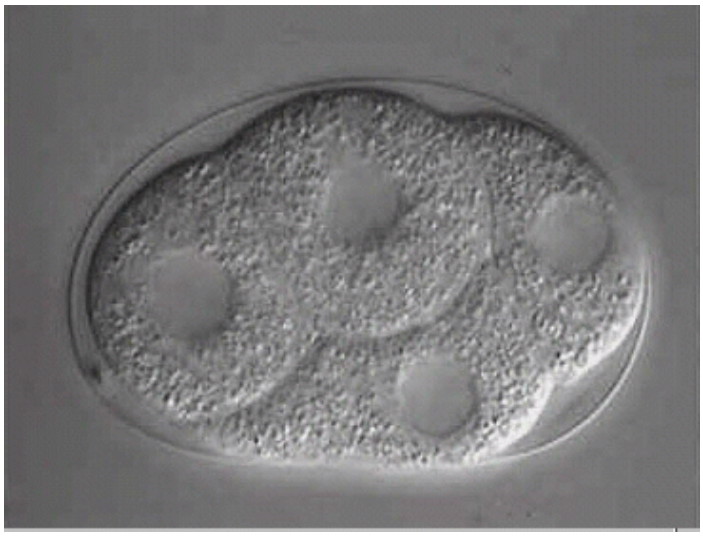


Figure 2.10: *The video freeze image shows one focal plane of a C. elegans egg in the 4-cell state. The egg under a Nomarski microscope with* $\approx 1000\times$ *magnification is studied in one focal plane for the following* $\approx$*10 hours (from www.simi.com)*

for building and manipulation of the cell lineage data for one individual organism. Thus, unified data formats and conventions are defined early for all computational methods and data process operations.

2.5.1 Data formats

The storage of data has to be selected carefully, since otherwise the interconversion of data into different formats may be time consuming. In particular, not only much work to program the interconversion may be avoided, but also programs become more readable under a well planed storage and access strategy.

The different data sources (e.g. microscope, programs etc.) produce specific formats, which have to be converted for further use of the individual programs. In computer science, databases are a convenient method to store and to retrieve great amounts of data. Therefore it was attempted to import, export and represent data as early as possible in a format readable by database programs.

2.5.2 Fundamental data format

A table is a basic format to store data in most databases. A row in a table is called record, which consists of a fixed number of equivalent fields, the entries of data in this row. Each column in the table keeps data of the same type (e.g. text, integers, dates, etc.). The structure or scheme of a record is the fixed sequence of types of records (i.e. a column in tables 2.1 and 2.2) which contain data of the same type. The relation between data records stored in different tables is the key index. If the content of a "key-index entry" in the corresponding column of two different tables is equal then the two records are related (i.e. they are linked to each other). Tables strengthen data storage in unified formats, which facilitate later processing by programs written in different computer languages.

Record-oriented data can be read without additional programming in most computer languages and database programs. The key-index allows an easy assignment of the data into arrays of computer languages for further manipulation and visualization.

2.5.3 Tables for records of a database

The data of the cell lineage may be represented as records for each cell in a tree with the structure of e.g. table 2.1 and 2.2.

Such a table is maintained for each individual worm. There is one special (or default) table of this kind which holds data of a reference cell lineage (the prototype).

The first record in the table (or file) usually contains the headings of columns (i.e. the identifiers for the computer program that reads the data). Tables may be modified easily by adding or deleting columns when they are imported into a spreadsheet (e.g. Excel of Microsoft) or database (e.g. Access of Microsoft or Oracle) and operated on with the database language SQL. The data of coordinates, e.g. the position of the nuclei with the time of division may be stored in

a structure given in table 2.2.

heading of column	type field	comment
key index	integer	unique number to access record
name	text	name of the cell
child1	text	name of child 1
child 2	text	name of child 2
dtime	integer	time at cell devision
fate	integer	fate of cell
xdiv	integer	x direction of division
ydiv	integer	y direction of division
zdiv	integer	z direction of nucleus
remark	text	additional information

Table 2.1: *Scheme for records of a database that contain data of a cell lineage and cell division*

heading of column	type field	comment
key index	integer	unique number to access record
name	text	name of the cell
state	integer	number of cells at that stage
fate	integer	fate of cell
xpos	integer	x-coordinate of center of nucleus
ypos	integer	y-coordinate of center of nucleus
zpos	integer	z-coordinate of center of nucleus
ndia	integer	nuclear diameter
info	text	additional information

Table 2.2: *Scheme for records of a database that contains data on the cell stage and the position of nuclei*

The tables could be linked by the name of the cells as the primary key (i.e. the key-index). However, for many computer calculations, a numeric key facilitates the access to different tables strongly (compare here also the program that uses the tables on E.0.5). Programs that converts data from and to the structure datatype (see section 2.5.5) to the SIMI or stage database (see following section 2.5.4) are gathered in the directory “utilities” on the DVD (see

appendix E.0.5).

2.5.4 Database of the SIMI BioCell browser

The program SIMI BioCell (www.simi.com) solves the task on the one hand to display the 3D-stacks of 2D-slices (see E.0.3) recorded by the microscope and on the other hand works on 2D-slices to locate either the position of a specified nucleus and to insert the marked coordinates of this nucleus into the SIMI database or to find the time of a cell division and insert it. A skeleton of the database which comprises a reference database (a prototype) with records for each cell (i.e. name of cell and entries for a reference lineage) is in this way filled with data of the actual cell lineage and positional data for cells.

The database for a project (i.e. a cell lineage tree) consists of a pair of files. The text file with the extension " `*.sbc` " (abbreviation for Simi Browser Configuration) summarizes the access information (see section 2.5.8) for the real database of the lineage contained in the file with the extension "`*.sbd`" (abbreviation for Simi Browser Database) (see E.0.3). Both file extensions have the same file name (i.e. `DEMO.sbc` and `DEMO.sbd`).

This file displays the database as a sequence of records which are separated by lines − − − (see the file `DEMO1.sbd` in section E.0.3). The skeleton of a record is given by lines of data:

```
− − −
1 1 0 0 ABal      ; 4 tree data, and name of cell
16200 3 -1 29     ; data of prototype lineage tree
-1 0 -1 29 ABal   ; data of actual lineage tree
0                 ; number of nuclear position that follow
− − −
```

Table 2.3: *Record of the initial database.*

Each record contains data for one cell in the developing organism. A record consists essentially of four lines, where the fourth line in table 2.4 displays the

```
– – –
0 0 0 0 ABarppapaa
127575 0 2 30 V1L     ;1. integer = time of division in units
127200 0 2 30 V1L     ;3. integer = fate of cell; 5. entry = generic name;
2
129000 413 384 4      ; nuclear position per line as
138000 407 411 3      ; time , x- , y- and z- position
– – –                 ; where x and y are positions pixel in a slice
                      ; and z is the number of the slice in the 3D-stack
```

Table 2.4: *A record with positional data. In case coordinates of the cell nucleus or other data have been determined, they may be inserted or stored in additional lines as follows.*

number of rows that follow with different time and spatial position of the nucleus. The first line shows essentially the name (identification of the cell as text (see [SSWT83])). The second line contains data of the reference tree and has as first number the time of cell division, as third the fate and as the fifth the generic name of the cell. The third line display data of the actual tree. The first number is the time of division of the organism under study. An entry of -1 in a line means not determined.

Another database obtained from Schierenberg (personal communication) contains the data up to the 100 cell state (see E.0.3 for details).

2.5.5 Data formats of visualization programs

Two visualization packages are inspected, firstly the Geometry Center "Geomview" [Geo] and secondly the "visualization tool kit (VTK)" [SML98]. Both of the packages work on similar structures of data sets, which split the data essentially into the three categories: topological, geometrical and other. As an example, the output for the same (six cell state) data set of C. elegans is given firstly as "unstructured grid" by VTK in table 2.5 and secondly as the `qhull` format [BDH96] in table 2.6).

```
# vtk DataFile Version 2.0
vtk output
ASCII
DATASET UNSTRUCTURED_GRID
POINTS 6 float
0.175 0.368 0.240 0.216 0.266 0.384      ;data
0.325 0.195 0.096 0.415 0.195 0.360      ;of points in space
0.418 0.389 0.216 0.578 0.309 0.192

CELLS 3 15                               ;topological data
4 5 3 2 4                                ;connectivity of points
4 4 0 2 1
4 4 2 1 3

CELL_TYPES 3                             ;type of topological cell
10
10
10
```

Table 2.5: *Dataset of the six cell state of C. elegans in the format of "unstructured_grid" of the program package VTK*

```
4
6 3 6
175 368 240 223649   ;geometric data
216 266 384 264868   ;(coordinates of points in space)
325 195 96 152866
415 195 360 339850
418 389 216 372701
578 309 192 466429
4 1 4 2 0            ;topological data
4 4 3 2 5            ;(connectivity of points)
4 4 3 1 2
3
1667236              ;volume of cells
1618960
1931236
```

Table 2.6: *The dataset of table 2.5 in the format generated by the program* `qhull` *(see [Geo]).*

In both cases, the geometric data are represented by spatial coordinates of points (in C. elegans the coordinates of the nuclei) preceded by information on the number of points (six) (table 2.5 line 5: "POINTS **6** float"; and in table 2.6 line 2:"**6** 3 6") and in line 1 of table 2.6 the dimension of the point space plus 1 is given. The topological data are given as an object (cell or simplex) per line where the first integer in a line is the number of integers in the sequence of integers that follows. Each of these integers represents a point of the geometric data (i.e. its name)

Each point is named by an unique integer (the integer name represents the position in the sequence of points in the geometric section). A sequence of integers of points in the topological section characterizes the object (cell) by its constituting points.

In the VTK dataset the topological part of the output is headed by the number of objects (cells,simplices) (**3**) that follow in (table 2.5 the line "CELLS **3** 15" where **15** is the total number of integers in this section) and in table 2.6 line 2 "6 **3** 6". VTK distinguishes different cell types to which the three integer sequences are assigned (table 2.5) the three integers following the line "CELL_TYPES **3**". In table 2.6 the volumes of the simplices of the topological section follow.

The utility programs "VTK2struc" and "struc2VTK" convert structured data into VTK unstructured grid format forth and back. "qhull2struc" and "struc2qhull" are the counterparts for qhull output and input data conversion and SIMI2struc and struc2SIMI for the SIMI database.

2.5.6 Internal data structure in C and MATLAB

The above conversion algorithms are designed as functions. As such they deliver besides the conversion of the ascii text file a function value in the program calling them. C as well as Matlab has a datatype "structure" (see table 2.7) into which the content of table 2.2 and 2.1 as well as tables 2.5 and 2.6 may be converted.

Such structure datatype may be e.g. an array of points and of topological data.

```
struc worm
                char name
                int state
                int x-pos, y-pos, z-pos
                int div-time, fate, n-diameter
                char info
end
```

Table 2.7: *Internal representation of record of table 2.2 in the computer languages MATLAB and C as datatype structure.*

Each record (cell) gives an entry in the array of structured records 2.7 which could be easily processed or accessed e.g. as by a command "worm(i).name" to get the name of the cell from a record with the key index i.

2.5.7 Images and image stacks

Time series of images are usually both recorded on a sVHS format video tape for documentation and digitized on hard drives in AVI formats (Intel Indeo 5.1 Codec in the DVD directory "miscellaneous" in appendix E.0.6) using the program supplied with the PCTV-Studio and the DC10-plus devices (www.pinnaclesys.com).

2.5.8 SIMI Browser of BioCell Inc.

A browser to visualize data of nematodes as C. elegans is commercially available from SIMI Inc. (www.simi.com). A trial version of the browser for the operating system Windows was free of charge available and is used in this work (the trial version is included in the DVD (E.0.3) of this work). The self designed programs produce data that can operate with the SIMI trial version. Figure 2.11 shows the browsers layout of the screen with a part of the cell lineage tree (left) and the spatial distribution of the nuclei (right). The menu to operate the browser is self explaining.

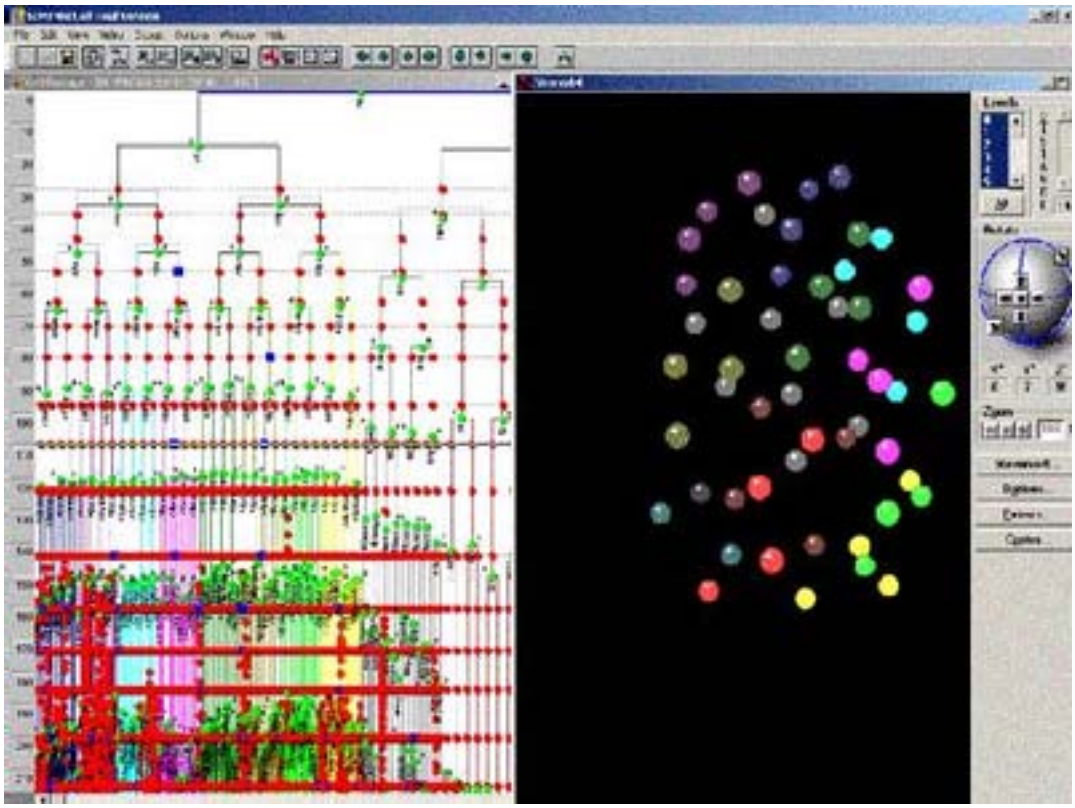


Figure 2.11: *Screen layout of the browser SIMI. The left window shows parts of the cell lineage of C. elegans. The lineage tree of an individual animal is shown in color whereas the reference lineage is displayed in gray color. On the left side of the window, a scale shows the time for the developing organism in [min]. Green circles mark cell division. Red circles mark times, at which the spatial positions of the nuclei were determined. The bluish horizontal line defines the time of the spatial nuclei distribution in the right window. The nuclei on the right side are colored according to their stem precursor cells (like in table 3.3). The reference times of division are displayed as gray skeleton in the background, whereas the tree generated by the actual times is colored and appears as foreground.*

The action of a mouse button may be explored by trail and error without the danger of modifying the data except for the edit function e.g. the spatial path of the cell may be followed back to its original precursor.

The SIMI-Browser operates on a database of an individual organism (`*.sbd`). The database (cell-lineage tree) can be build (in a 4D microscopy session), edited and browsed. The database is assigned to a session as a project by choosing in the menu the project in specifying the database file e.g. `DEMO1` or `DEMO2` (see

appendix E.0.3). These files may be produced, processed and modified by the self designed programs ready for use (e.g. `DEMO2` on DVD (in appendix E.0.3) contains the clustered version of a cell lineage tree).

A program called "struc2SIMI " may be used to create a database readable by SIMI-Browser from tables and data formats used in this thesis. The program asks for the location (path) where it finds the files of reference and actual position of nuclei data and where to place the SIMI database.

Besides the individual database (see section 2.5.4), the SIMI browser reads an initialization (Configuration) file with the same name as the database file `DEMO1.SBC` ending with the extension `"*.sbc"`. This file contains e.g. the scale factors for the conversion of times in the database to minutes and details for spatial coordinates and other settings of the recording of the microscopes session. It is a text file and may be inspected by a text processor for details (see appendix E.0.3).

2.5.9 SIMI analysis module

The browser has a build-in video player to display video sequences (in AVI format) of the microscopic images to view the development of the nematode. The overlay technique may display the position of a chosen nucleus in the lineage tree in the video. Individual nuclei may be identified.

The SIMI Browser incorporates a module to extract positional data from the video sequence of the video player and writes them with their corresponding time values into the database. The positions of interest are the central coordinates of the nuclei.

The division time of individual cells may be determined by scanning through the set of images with the program SIMI and inserted in line 3 (see table 2.3). Also, the program allows the determination of the position of a cell at a specific time and inserts it into the record (see table 2.4).

This analysis is the manual counterpart of automatic segmentation (see section 3.8), which dissects the images into objects and finds the position of the nuclei or other objects in the 3D-space.

Chapter 3

Results

3.1 Abstraction of the cell lineage

3.1.1 Simplification of the cell linage tree by abstraction.

The C. elegans database holds information on every cell in the life span of the nematode. Ordering the data systematically would reduce it without affecting the information about the biological process of the development of the organism. Further, it would allow an easier manipulation of data and may reveal laws, which govern the process of development. The cell lineage tree is determined essentially by the times, at which cell visually divide. A histogram of all division times (see figure 3.1) in the database `Demo1.sbd` (see [SHMS97]) of the virtual prototype (see methods 2.5.4) demonstrates clearly that there are time gaps, during which no, or only a few, cells divides.

The histograms of the prototype and individuals look quite similar. The gaps present in the lineage of the prototype are more pronounced in the individual sub-lineages of the founder cells. Figure 3.2 shows an individual sub-lineage of the founder cell AB from the database `DEMO1.sbd` as a histogram. The gaps between clusters of division times are enhanced in the sub-lineage compared to figure 3.1, as expected.

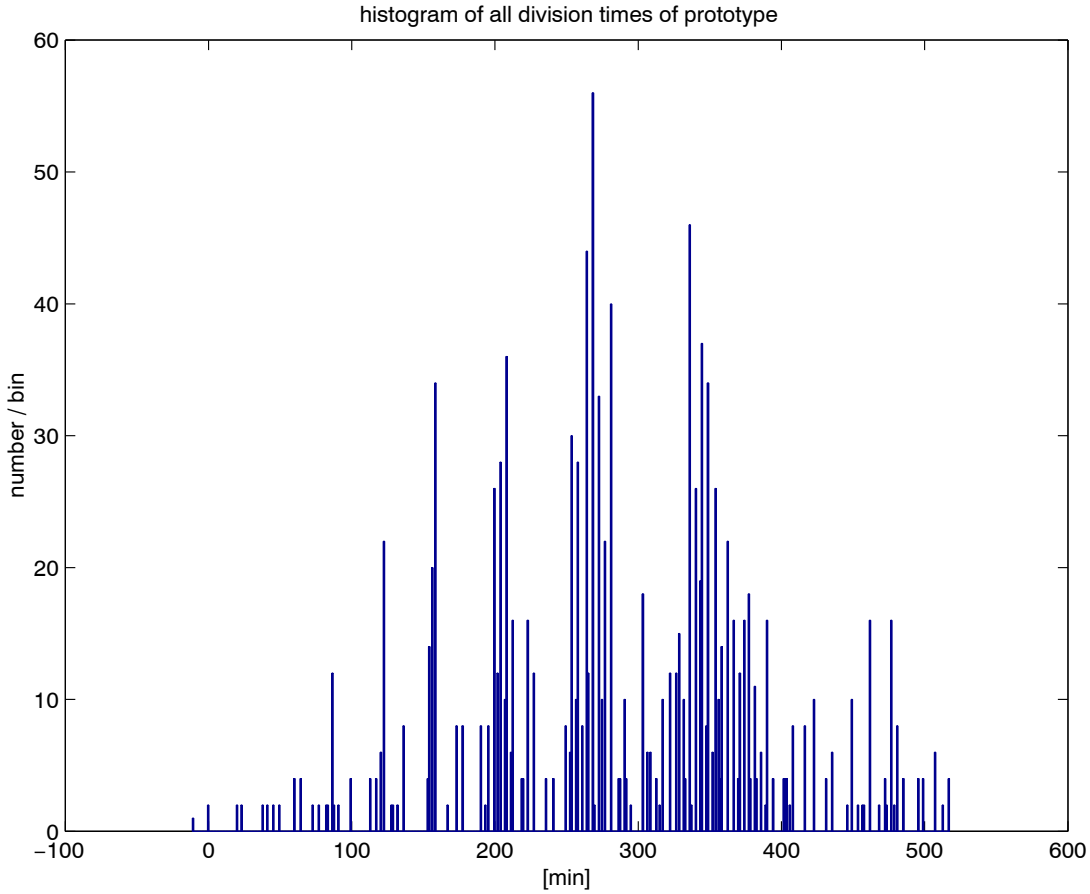


Figure 3.1: *Histogram of all times of division of the prototype cell lineage. The x-axis gives the times of division in the database (`DEMO1.SDB` in E.0.3). The y-axis gives the number of times in each bin. Times of division are unevenly distributed, but clusters are formed around certain bins. Resolution 500 bins.*

A cluster analysis aggregates cells that divide at nearly the same time in a cluster. The time of division of individual cells may be replaced by a mean division time of the cells in their cluster. Thus, the divisions of cells in the cluster are synchronized, although they are indeed somehow de-synchronized in nature's reality. It will be assumed that clusters are selected in such a way that de-synchronization is nearly statistical and will not affect the biological development. By this method the number of parameters (different cluster division times) needed to describe the cell lineage is drastically reduced. The process to reduce the parameters may be considered as an abstraction which removes the

variability from individual division times (i.e. reduces or eliminates noise).

The cluster analysis for the lineage tree of the founder cell AB has been studied with the computer program `Clustan` [Inca]. A result of the analysis is shown in figure 3.3 as a tree diagram and in figure 3.4 as a dendrogram. In the dendrogram one may draw a horizontal line that intersect with branches of the tree. The number of intersections is the number of clusters formed at that level (i.e. at the horizontal line). An estimate for the optimal number of clusters may be obtained by moving the horizontal line up or downwards and comparing the distinguished groups of peaks that are separated by gaps in the corresponding histogram (see figure 3.2).

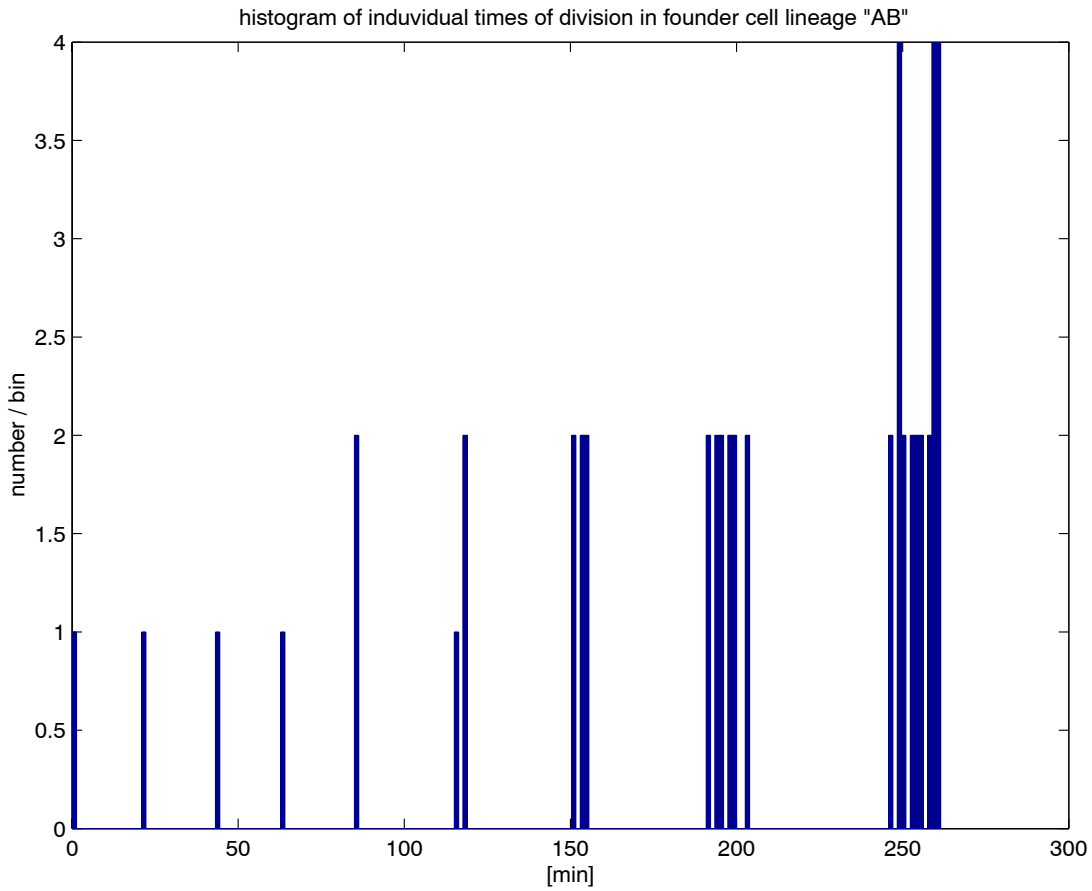


Figure 3.2: *Histogram displaying times of division since the first division of an individual's cell lineage in the subtree of the founder cell AB (from the database* `DEMO1.SDB` *in appendix E.0.3). Resolution 200 bins*

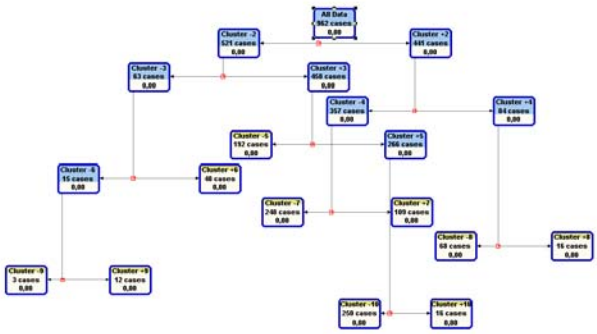


Figure 3.3: *Screen shot of the cluster analysis by the program* `Clustan` *of the sub-lineage of the founder cell AB. The data of the analysis are the individual times of division in the database* `DEMO1.sbd` *(E.0.3). There are 10 clusters at the level shown i.e. leaves in the analysis tree*

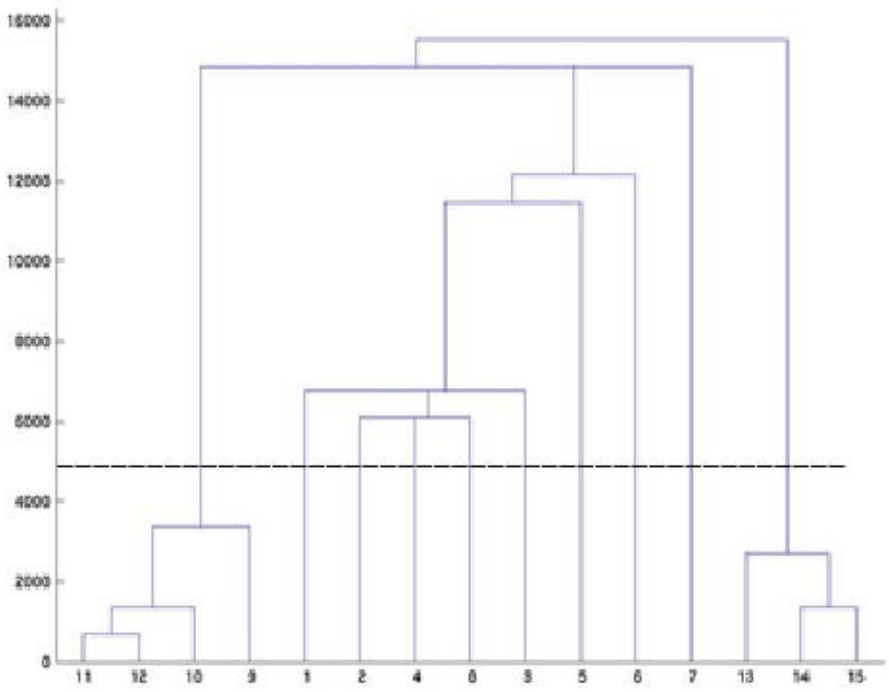


Figure 3.4: *Dendrogram of the cluster analysis of the subcell lineage of the founder cell "AB". The bottom integers represent cluster numbers. Y-axis is the distance of the complete linkage in the cluster analysis.*

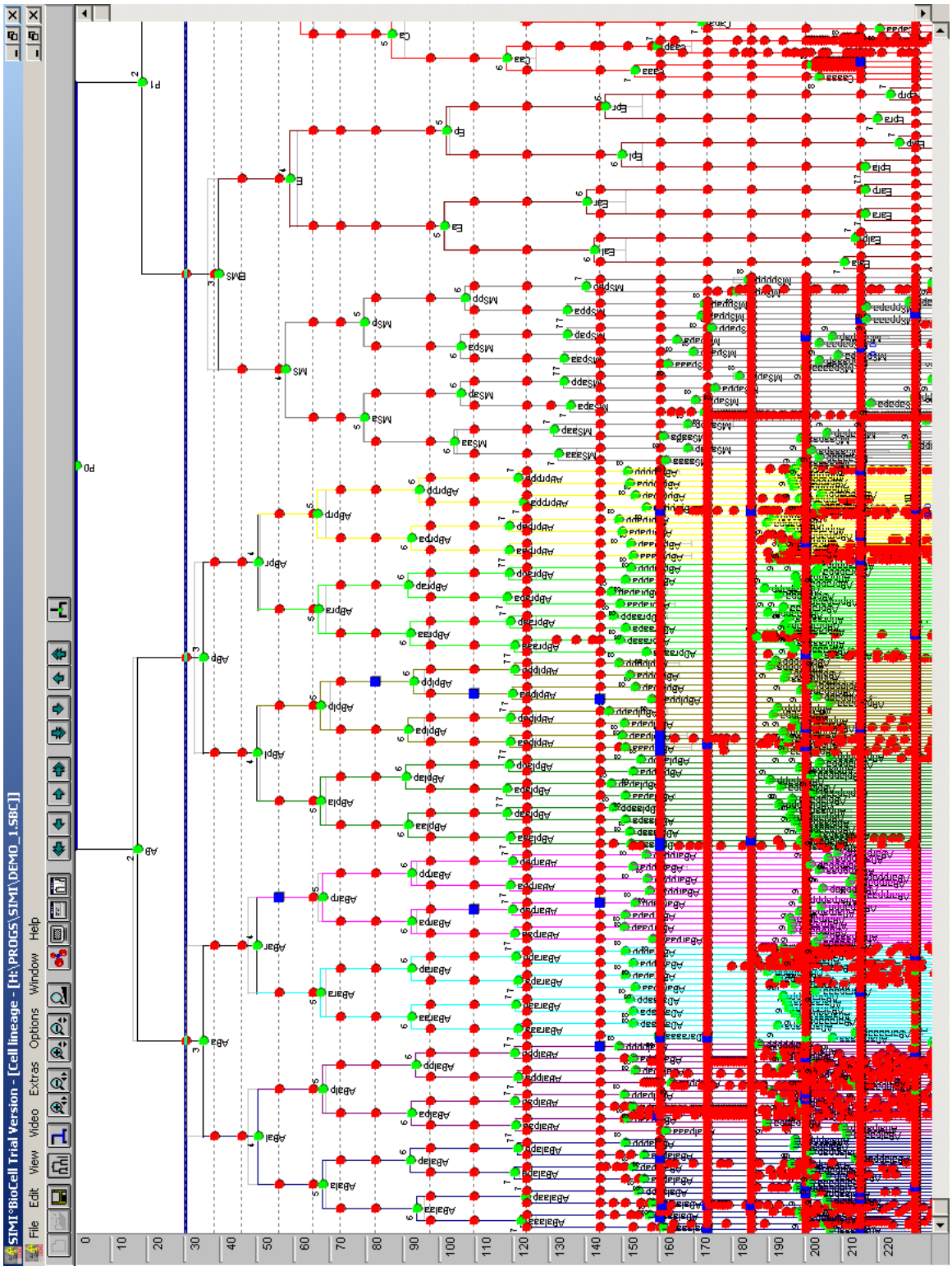


Figure 3.5: *Screen shot by the SIMI browser of the cell lineage tree of the nematode C. elegans from database* `DEMO1.sbd`. *Left scale time in [min]. Green dots mark cell division events labeled with name of dividing cell. Red dots mark times of cells at which its nuclear position is determined. The dashed horizontal lines indicate states at which the positions of all nuclei are determined. The gray lineage in the background is the prototype (default database). The numbers at a cell division indicate the number of division in history since fertilization.*

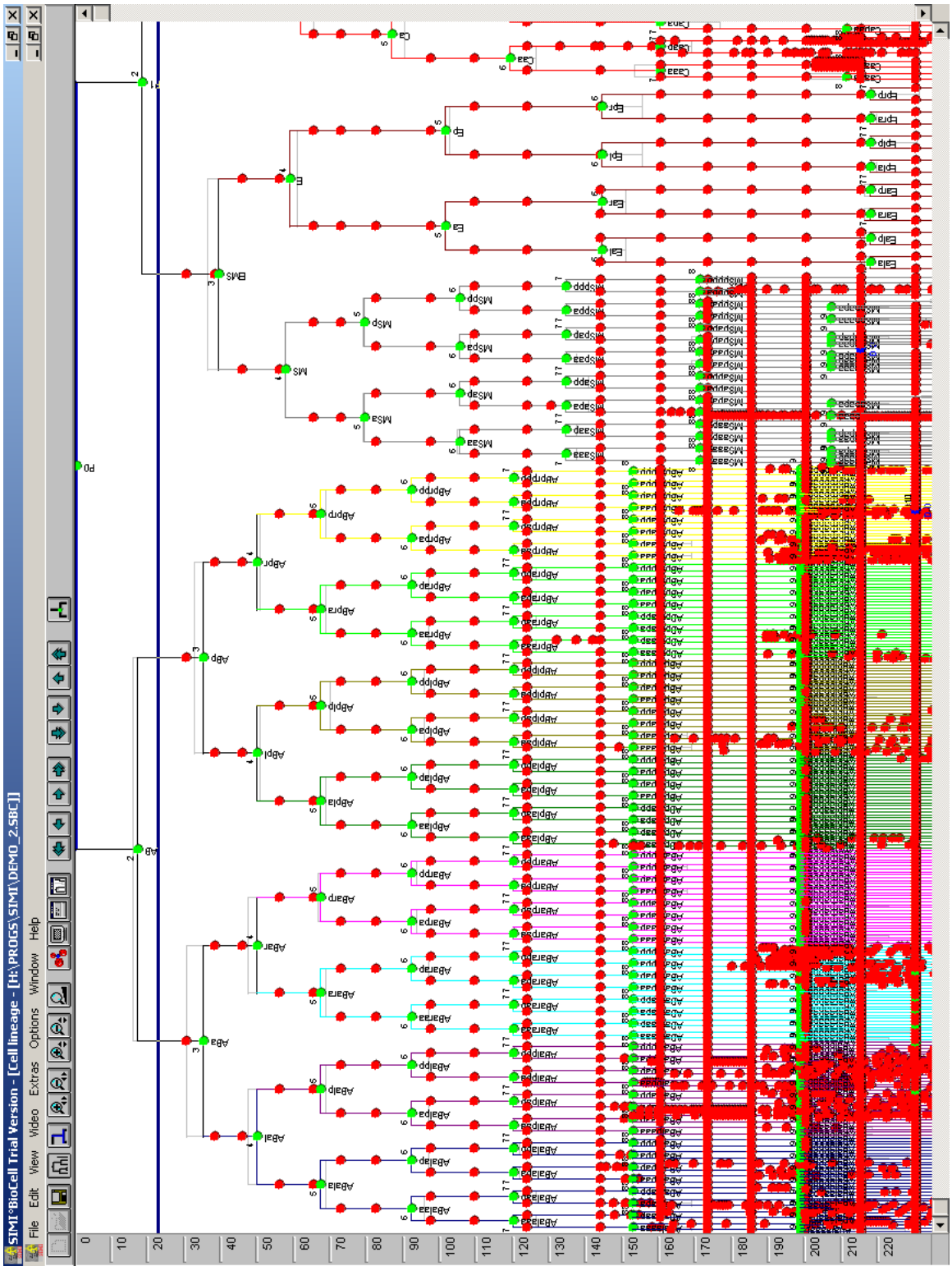


Figure 3.6: *Screen shot by the SIMI browser of the cell lineage tree of the nematode C. elegans from the database* `DEMO2.sbd` *(see appendix E.0.3). The same layout as in figure 3.5 except that the individual times of cell division are replaced by their corresponding mean values from the cluster analysis.*

3.1.2 First level of abstraction

In further studies, the licensed computer program `Clustan` was replaced by the MATLAB statistics toolbox since these algorithms could be integrated more readily in the calculation of clusters of the complete cell lineage tree and data conversion between different programs and operating systems could be avoided.

In the cell lineage tree of an individual, always some division times are not determined. For an analysis of the total development, these times may be replaced by their times estimated from the prototype. Since the deviations of the times of the prototype and of the individual may differ significantly, it was preferred to use the prototype (of database `DEMO1.sbd` or `DEMO2.sbd` (see E.0.3)) for the following cluster analysis.

There may be small differences in the cluster analysis on individual wild type cell lineage trees (e.g. an additional cluster) but the overall picture look very similar. This may be verified by applying the analysis program on the DVD in appendix E.0.3 to an individual database of a cell lineage. The results of the cluster analysis for the different sub-lineages of founder cells of the prototype are shown in the figures 3.7 to 3.12.

It is apparent that the clusters are separated by time gaps, in which cells do not divide. The division times in clusters may be replaced by a representative mean time of the cluster (see figures 3.7 to 3.12 red circle). These representative times may be inserted in the cell lineage database and the result may be displayed by the SIMI browser. The actual division times may be considered as statistical deviations from the representative cluster time. The replaced times of division give the first abstraction level of the cell lineage tree. The subtrees are characterized by only a few representative division times (see table 3.1).

Table 3.1 may be used to generate a cell lineage tree by an algorithm in conjunction with the skeleton of the database, which considers dead cells.

Given the mean cluster times, the times of division may be replaced in the

founder cells	division times										
AB	0.00	19.84	39.69	61.66	86.29	121.11	156.59	205.14	265.39	352.41	475.03
C	90.72	136.08	189.94	238.14	316.10	412.49					
MS	49.61	82.22	113.40	158.76	212.98	274.62	380.56				
E	55.28	128.28	192.78	278.30	428.79						
D	45.36	72.29	99.22	135.02	175.06	222.02	308.89	416.18	517.39		
P	-11.34	0.00	22.68	49.61	90.72	167.26					
division counts	1	2	3	4	5	6	7	8	9	10	11

Table 3.1: Mean times of cell division obtained by cluster analysis for the founder cell sub-lineages. The mean times of cell division clusters are listed in the sequence of the clusters with in increasing order.

prototype field (see methods table 2.3) in the database to yield an impression of how the cell lineage tree is deformed by this simplification (or first abstraction).

Figure 3.6 demonstrates that the lineage is quite well simulated by this first level of abstraction (compare the gray background of the prototype against the colored lineage tree with the inserted mean cluster times). For comparison, the prototype (reference) tree and the original individual data is depicted in figure 3.5. Both trees coincide quite well. The real development may be approximated well in many respects by this simplification.

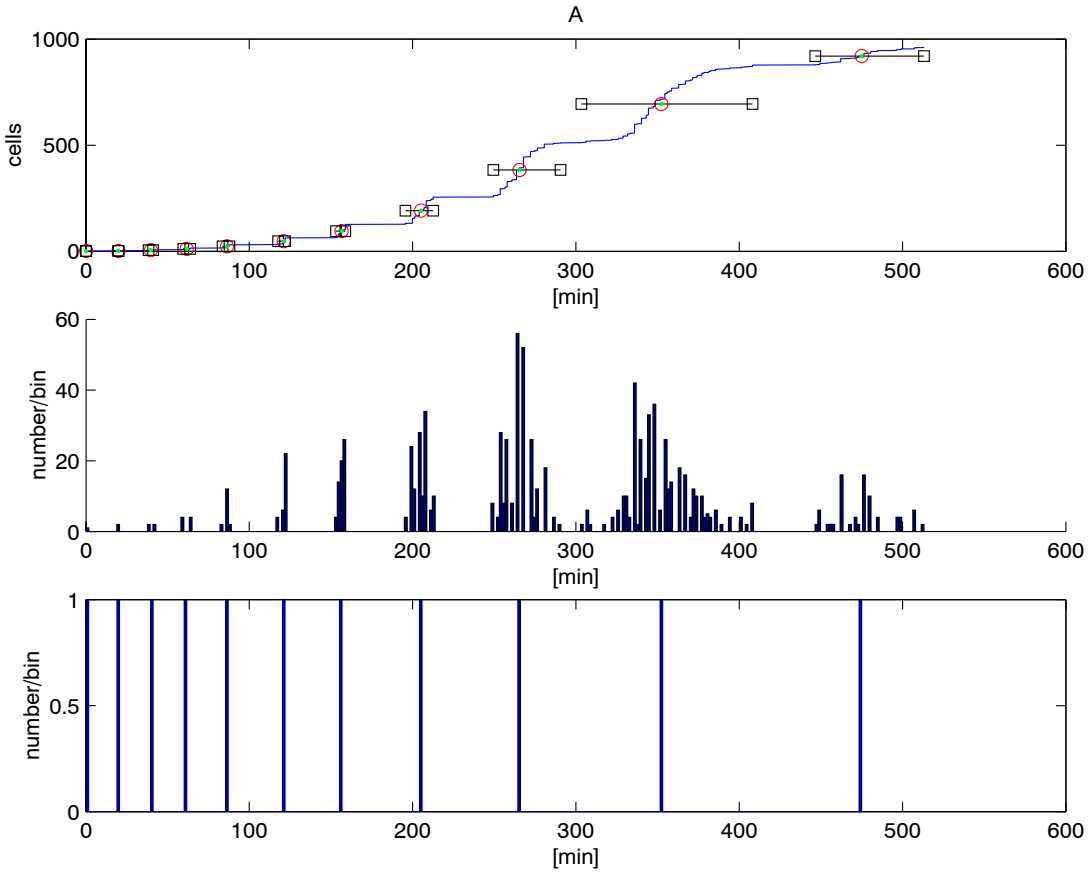


Figure 3.7: *Cluster analysis of the sub-lineage of the founder cells. Upper panel: the cumulative number of cells in the sublineage of the founder cell "AB" plotted against the time of division. Red circles are the mean values of the clusters generated by the cluster analysis corresponding to the values in the bottom panel. The minimal and the maximal deviation from this mean value in the cluster are given by a black horizontal bar with squares connecting the extrema in the cluster. Middle panel: Histogram of the sublineage of the founder cell. The division times are packed into 300 bins. Bottom panel: Corresponding mean cluster values of the times of division in the cluster. The x-scale is the same in all panels.*

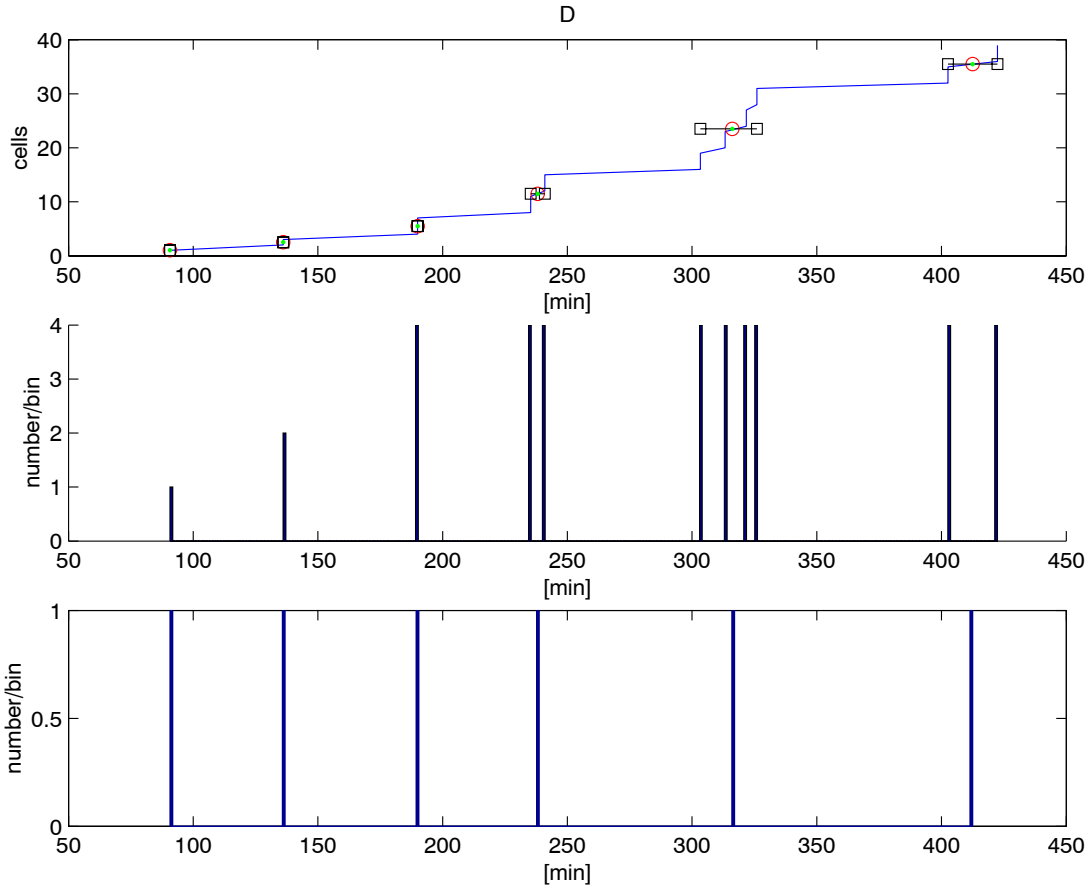


Figure 3.8: *Cluster analysis of the cell lineage of the founder cell "D" (for details see figure 3.7)*

3.1.3 Second level of abstraction and generating function

In figure 3.13, the histogram of all the mean cluster times is displayed. When the above cluster analysis is applied to these times again, the parameters (mean cluster times) are reduced further, so that a few times in the table 3.1 obtain the same value.

Figure 3.14 displays the sorted cluster times of table 3.1 in ascending order. If the few times of the second cluster analysis on the data of figure 3.13 are removed from the data of figure 3.14, then the curve still becomes smoother (which is not shown in a figure here).

The aim of abstraction is to reduce the number of parameters in the database. By fitting the data of figure 3.14 to a function, a correlation between the data

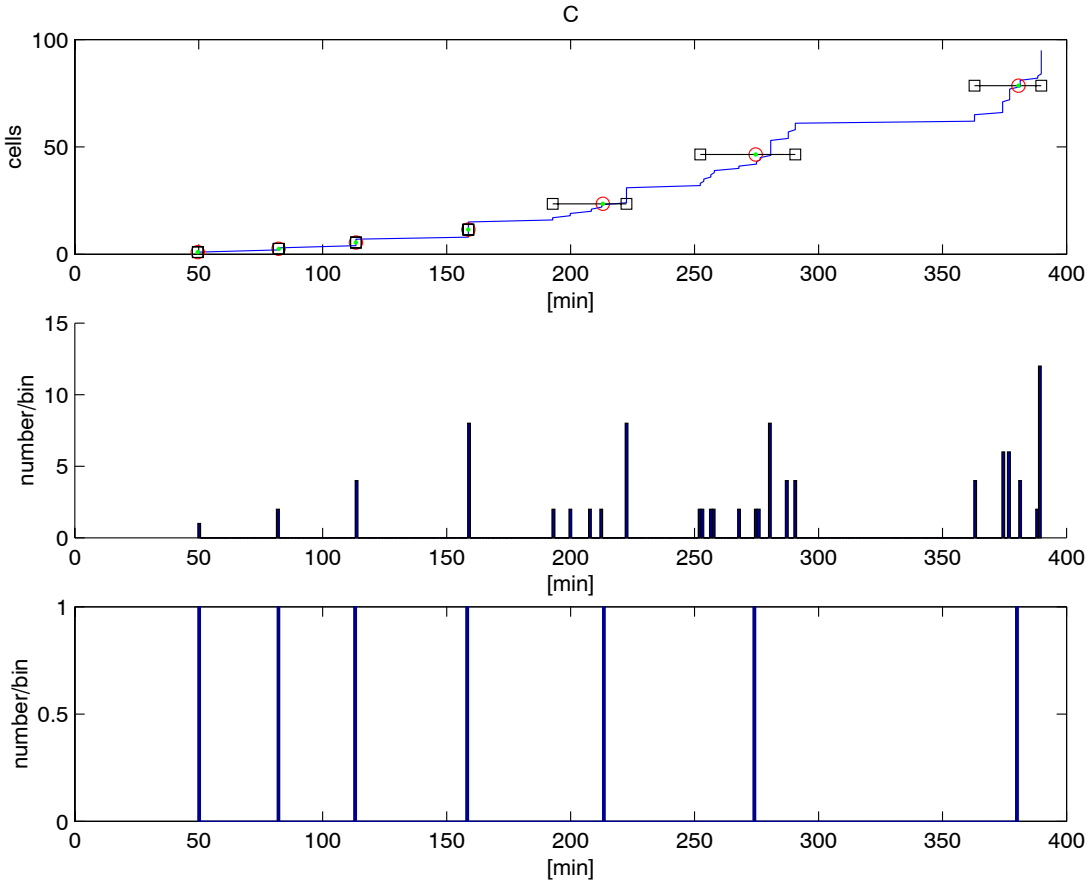


Figure 3.9: *Cluster analysis of the cell lineage of the founder cell "C" (for details see figure 3.7)*

in table 3.1 and the fitted function values could be established. This reduces the number of parameters needed to describe the lineage tree further to the parameters (coefficients) in the function. However, always a check has to be done to reproduce the lineage tree within a tolerance that is not effecting the spatial arrangement of the cells. Nevertheless, an algorithm (functions) is developed which reproduces table 3.1 from the curve in figure 3.14 i.e. from the first abstraction of the cell lineage.

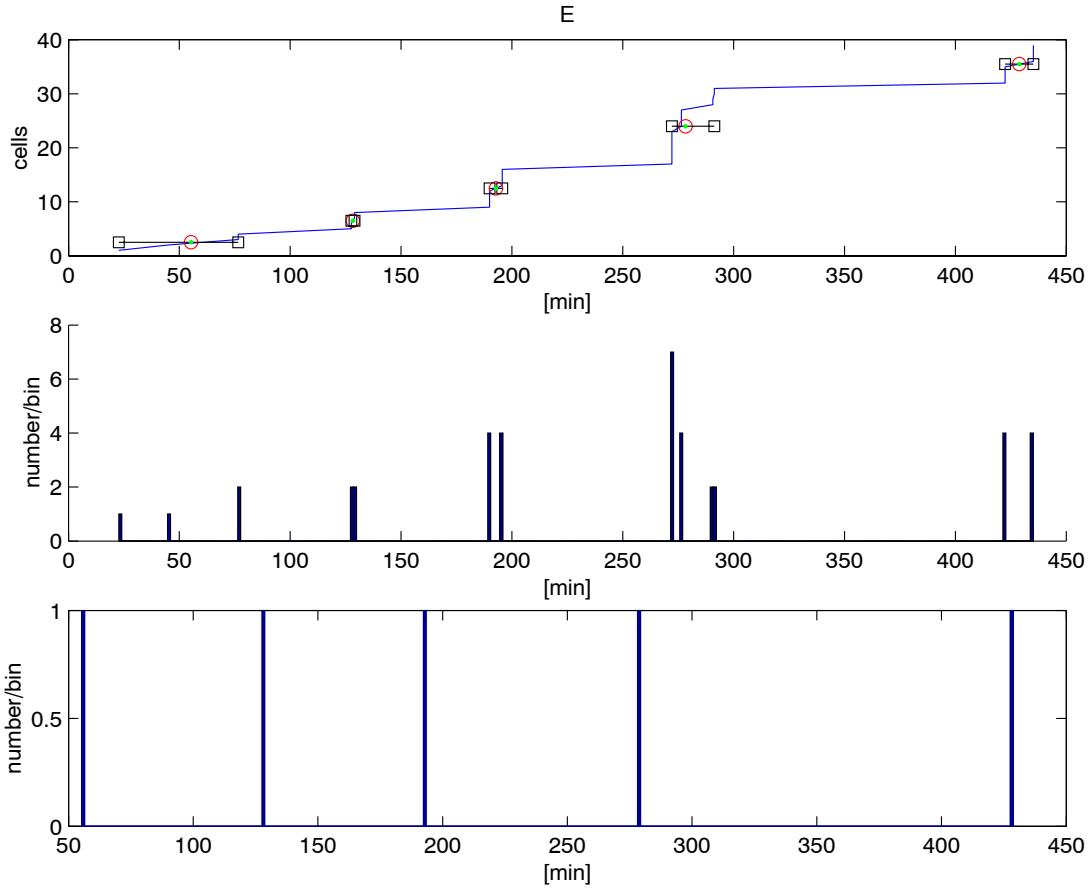


Figure 3.10: *Cluster analysis of the cell lineage of the founder cell "E" (for details see figure 3.7)*

3.2 Contact matrices

The cell lineage shows the time sequence, in which the cells divide and exist during the development of the organism. Since the nuclei of the cells and their division are clearly visible in the microscope, the nuclear position in space and time of division, as well as that of membranes surrounding the cells can be monitored.

The contact between cells (i.e. when their membranes touch) may be represented in a matrix at a definite time of the development. The columns and rows of the matrix are indexed in the same sequence with the cell identification (e.g. the cell names in the lineage tree). If a cell has contact with another cell a 1 is entered as matrix element at the corresponding row and column index. For

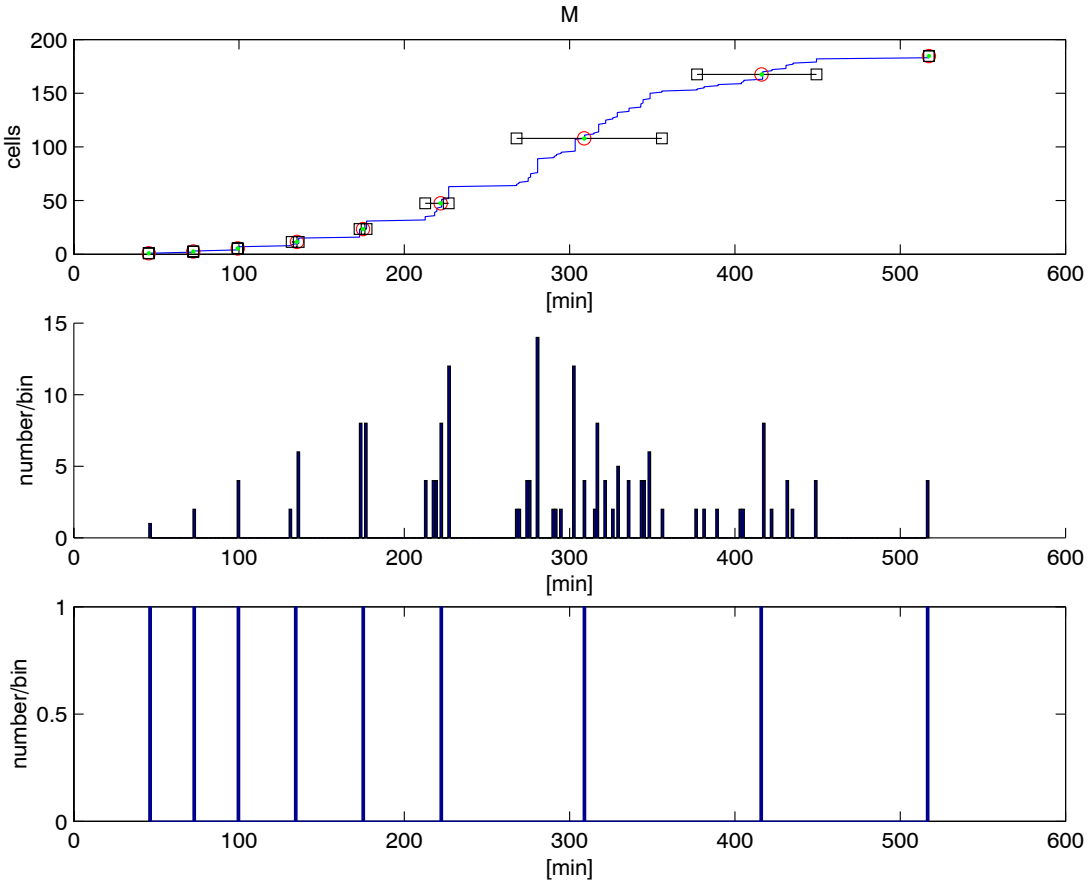


Figure 3.11: *Cluster analysis of the cell lineage of the founder cell "MS" (for details see figure 3.7)*

the eight cell state the matrix table 3-2.1 gives the contacts (connectivity) in the aggregate of cells at this definite state of development.

	ABal	*ABar*	*ABpl*	*ABpr*	*MS*	*E*	*C*	P3
ABal	0	1	1	0	1	0	0	0
ABar	1	0	1	1	1	0	1	0
ABpl	1	1	0	1	1	1	1	0
ABpr	0	1	1	0	1	1	1	1
MS	1	1	1	1	0	1	1	1
E	0	0	1	1	1	0	1	1
C	0	1	1	1	1	1	0	1
P3	0	0	0	1	1	1	1	0

(3-2.1)

The matrix is symmetrical since a cell contact is bilateral (not oriented). The elements of the matrix are either 0 or 1 entries (later they obtain a value

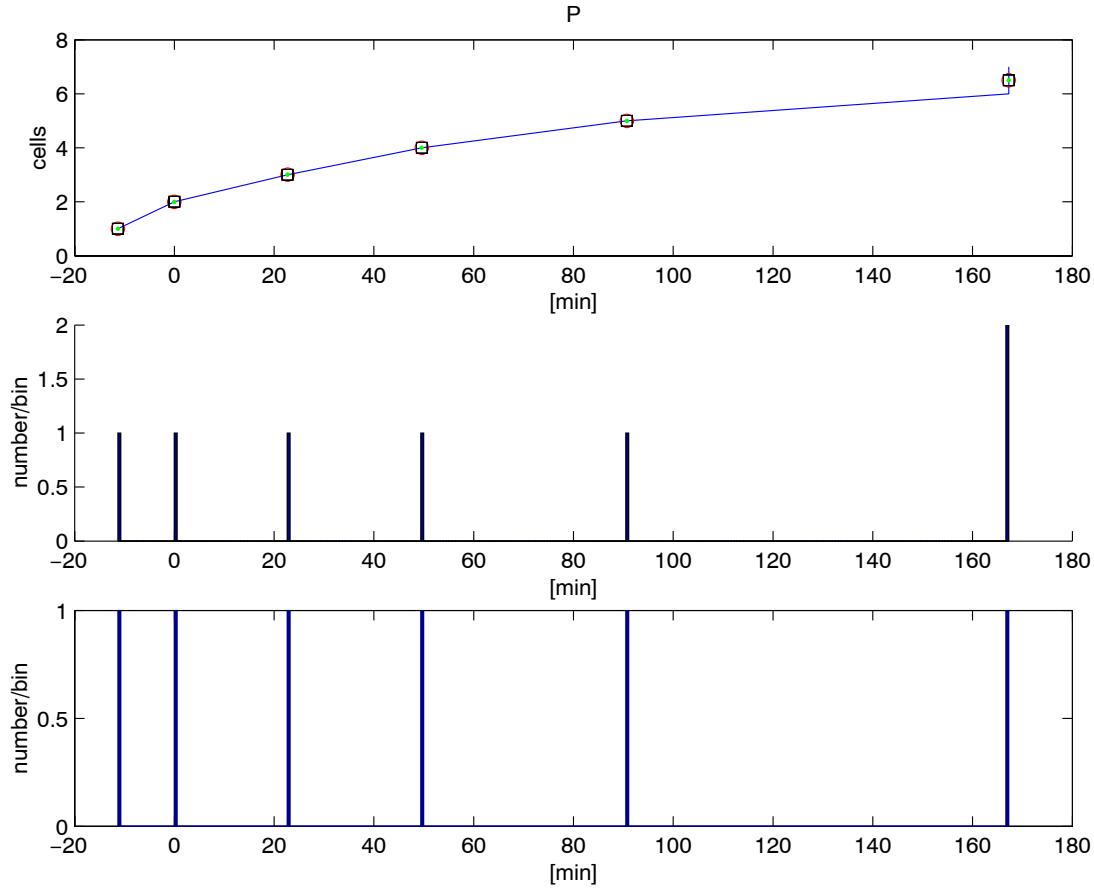


Figure 3.12: *Cluster analysis of the cell lineage of the founder cell "P" (for details see figure 3.7)*

characterizing the connectivity e.g. the orientation). Such matrices may be graphed by dot plots in which the matrix entries are plotted as dots, if a 1 is at that position in the matrix. Figure 3.15 displays a dot plot of the 8-cell state in the egg.

The sequence labeling the entries of matrix indices of cell names (or ID's) is arbitrary. A change of this sequence (in the same way for rows and columns of the matrix) will change the position of the matrix entries and its dot plot, but it leaves the cell contacts invariant. The natural cause of a sequence would be the order given by the cell lineage tree. However, there are other conditions for reordering the rows and columns of table 3-2.1. In figure 3.16, a few are summarized, but these do not give a unique representation either.

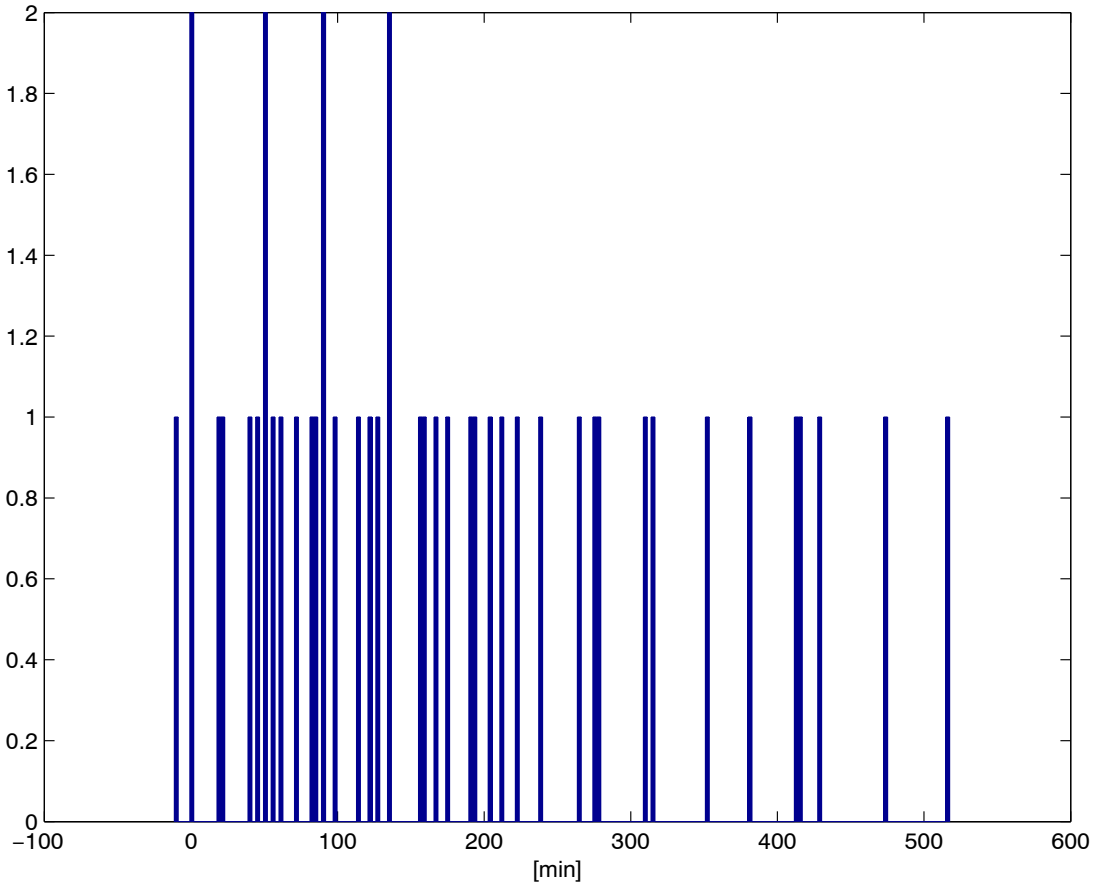


Figure 3.13: *Histogram of all clustered times of division obtained as mean cluster times. A few times are still quite close. The total number of clustered times is* 44. *The bins, which carry two times, originate from the subtree of the founder cell "P", since their times appear twice, firstly in the germ line and secondly in the founder cell split of the germ line. Resolution 200 bins.*

The non-zero count method counts the number of contacts of a cell and sorts the cell sequence (i.e. sorts the matrix index) according to the contact numbers of the cells (i.e. sum of entries in its row). The Cuthill-McKee method orders the sequence so that the contacts are as close as possible to the diagonal of the resulting dot plot matrix [JMS92]. Here, cells appear close together in the sequence, if they are close together in the egg, also. The minimal degree method is a refined non-zero method, which order degenerated subsequences by additional parameters. This is of interest when e.g. the elimination or addition of cells to the aggregate is considered.

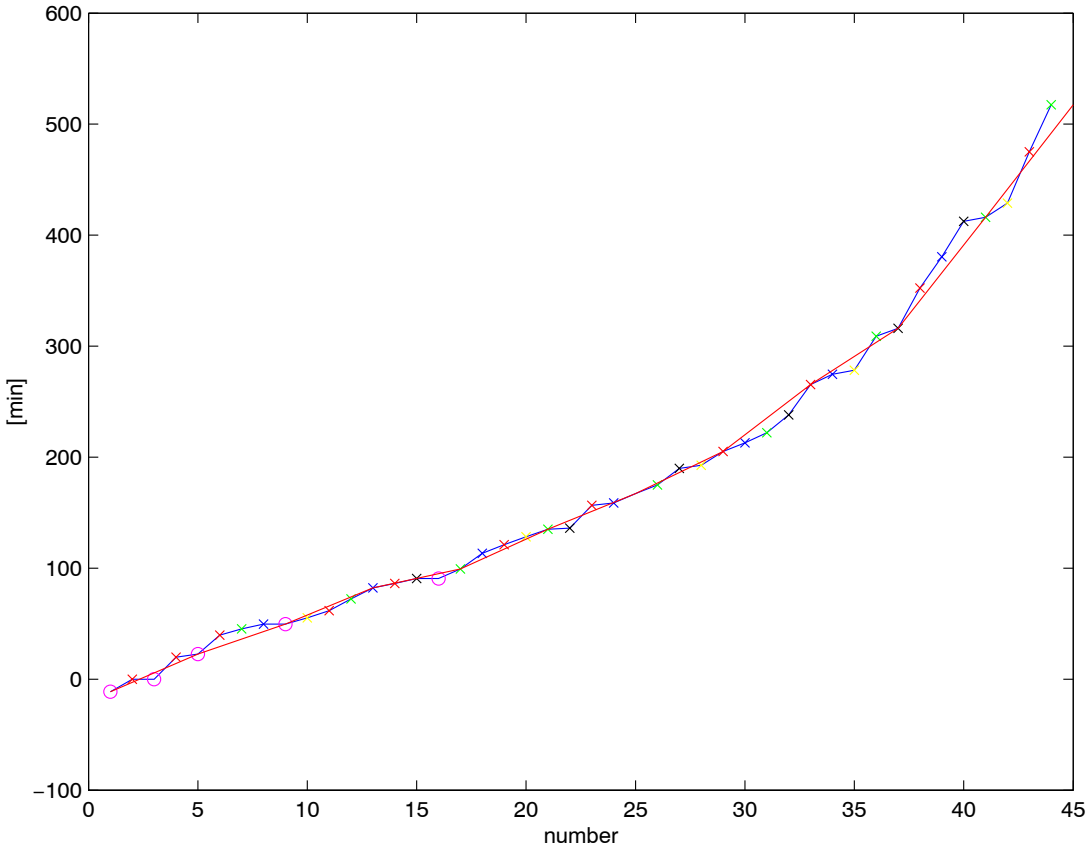


Figure 3.14: *All clustered times of division in table 3.1 sorted in ascending order and plotted against their order number. The magenta circles are times from the germ line denoted by "P". The crosses indicate times from founder cell lineage according to the color table (see table 3.3). Compared with e.g. figure 3.7 upper panel, the curve appears to be quiet smooth already. The red line is the spline fitted curve.*

Figure 3.17 shows the dot plot of the incidence matrix of the eight cell stage of table 3-2.1. The contacts c_1 to c_{21} to cells are listed (a row represents exactly one contact). Since two cells contact each other only once, each row in the matrix has only two non-zero entries. In the incidence matrix, a contact may be oriented or directed (i.e. it points from one to the other cell like a vector describing a force). In the oriented case, $+1$ or -1 values are entered in the matrix depending on the direction of contacts. The directed values for each row in table 3.2 sums to zero.

At the time, no real cell contacts are completely available for larger aggregates

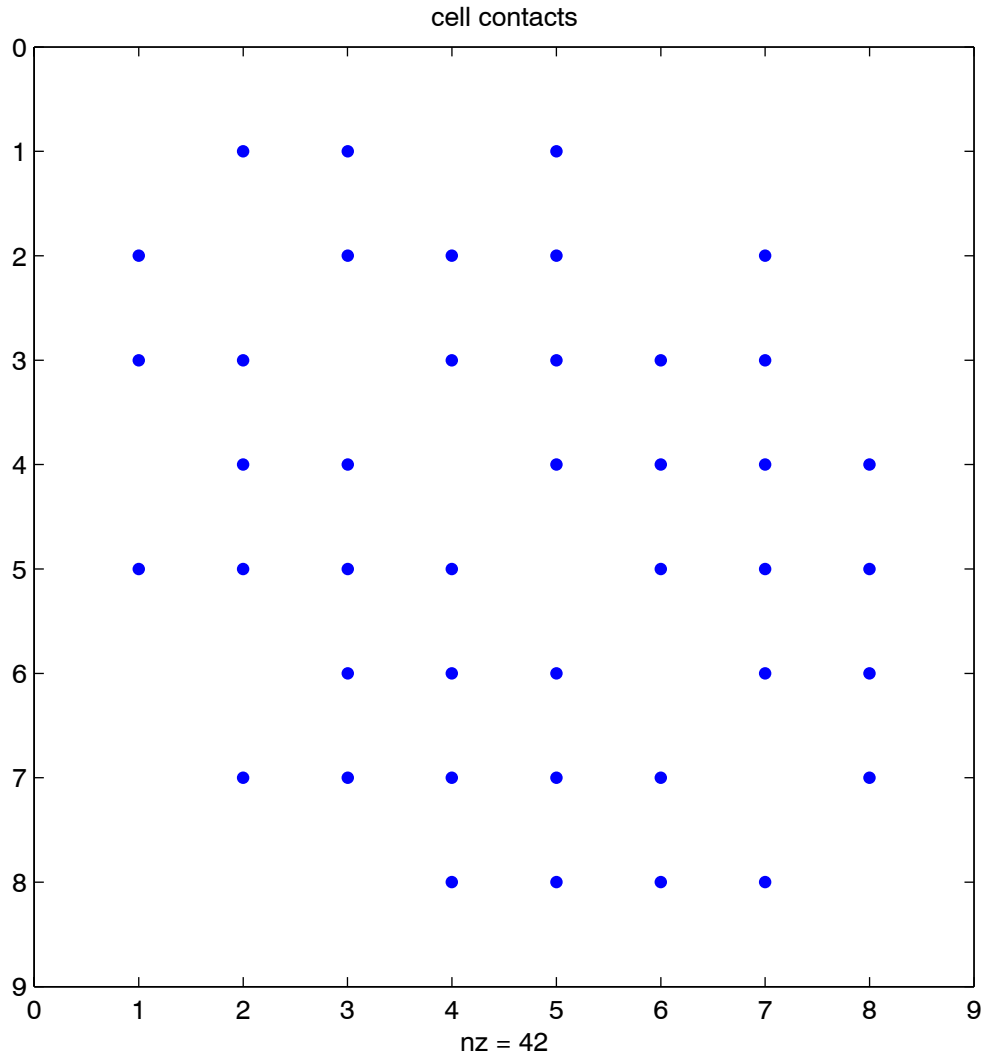


Figure 3.15: *Dot plot of cell contacts in the 8 cell state at time 45 [min] after first division. A dot means that the cells are in contact. The integer labels at the x-axis and y-axis correspond to the name of a cell:* $1 = ABal; 2 = ABar; 3 = ABpl; 4 = ABpr; 5 = MS; 6 = E; 7 = C; 8 = P3$. *There are 21 contacts corresponding to 42 dots in the symmetric plot.*

of cells. Therefore, the adjacency of cells is computed from the position of the nuclei by the Delaunay method. The method delivers a connectivity of the cell-(nuclei) aggregate as a mesh of tetrahedra. Each edge of the tetrahedra is a contact between cells. All the contacts may be casted into an adjacency matrix as described above and can be treated likewise (see algorithm in E.0.4 for an

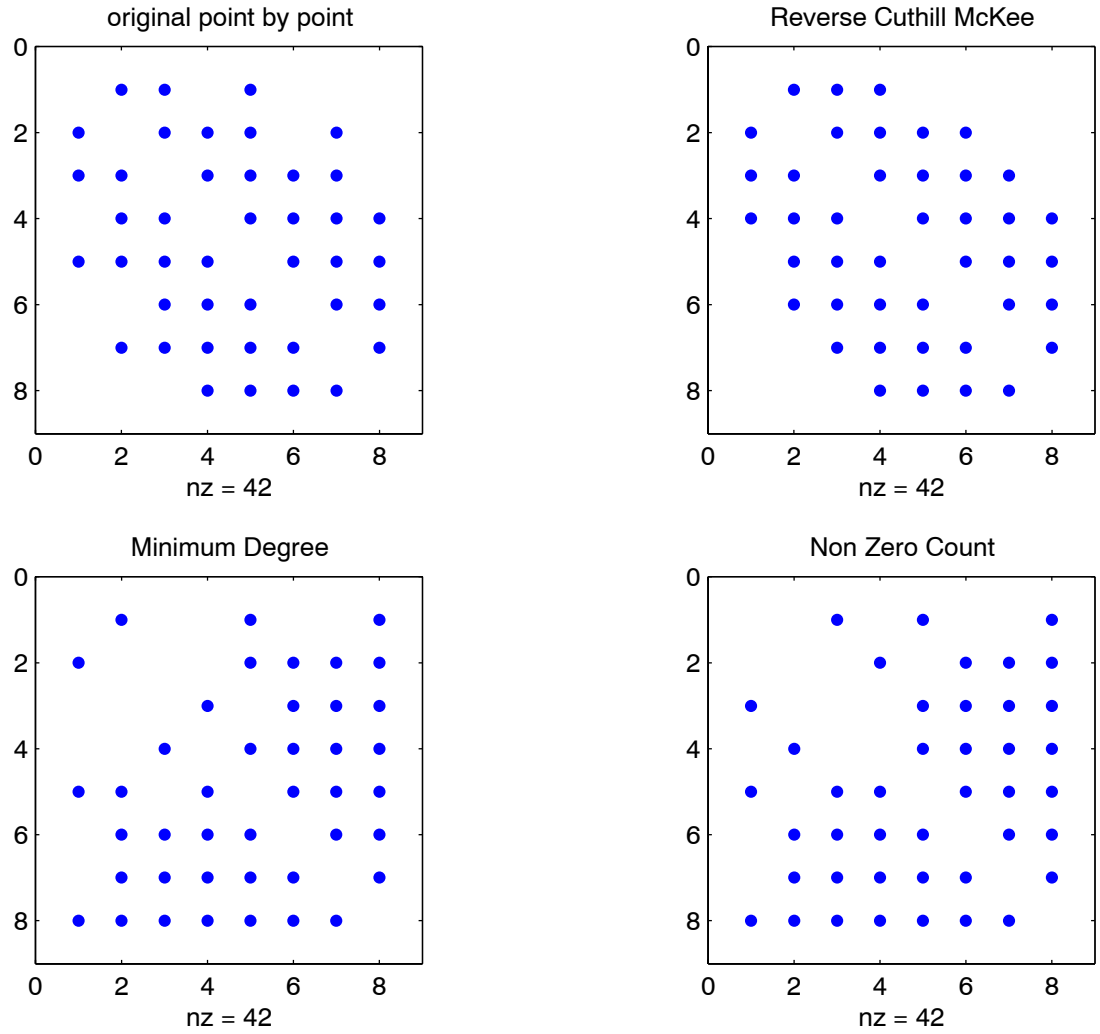


Figure 3.16: *Dot plots of contacts of cells in the 8 cell state. Permutation of the columns and the rows in the underlying matrix leaves the contacts between cells invariant. Upper left panel: is the same as figure 3.15 Upper right panel: rows and columns are permuted according to the method "Reverse Cuthill Mc Kee" Lower left panel: rows and columns are permuted according to the method "Minimum Degree" Lower right panel: rows and columns are permuted according to the method "Non Zero Count"*

n-cell stage).

An extension of the connectivity may be considered. The tetrahedra not only consists of edges but also of triangles (facets of the tetrahedron). The triangles in the net of tetrahedra have an adjacency matrix, in which two triangles are considered adjacent if they have an edge in common. The same is valid for the

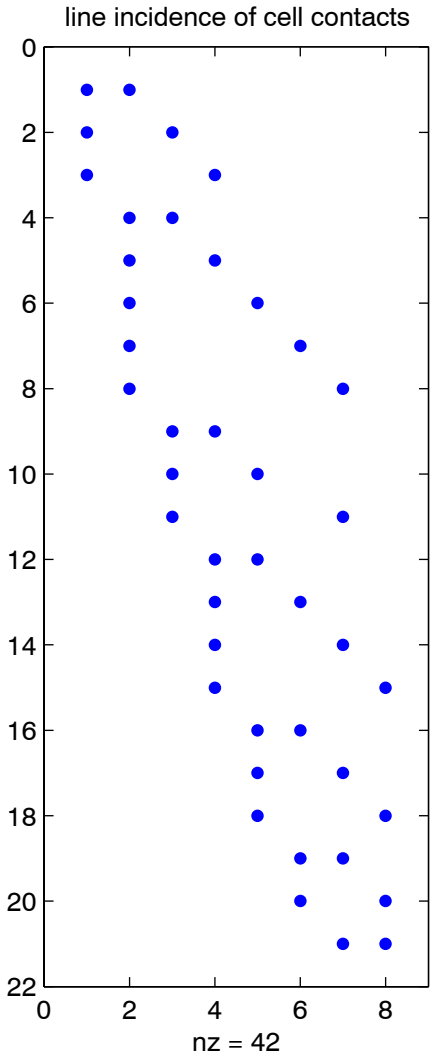


Figure 3.17: *Dot plot of the incidence of cells in the 8 cell state. A dot means a cell contact (y-axis) that involves the cell on the x-axis. The integer on the x-axis labels the cell names which are given by:* $1 = ABal; 2 = ABar; 3 = ABpl; 4 = ABpr; 5 = MS; 6 = E; 7 = C; 8 = P3$. *The y-axis numbers the 21 cell contacts.*

tetrahedra, for which adjacency is defined between two tetrahedra, if they have a triangle in common. The corresponding dot plots may be obtained for different stages by the algorithm in section E.0.4.

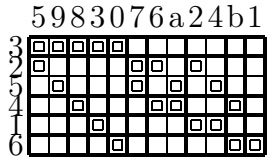
	ABal	*ABar*	*ABpl*	*ABpr*	*MS*	*E*	*C*	P3
c_1	1	-1	0	0	0	0	0	0
c_2	1	0	-1	0	0	0	0	0
c_3	1	0	0	-1	0	0	0	0
c_4	0	1	-1	0	0	0	0	0
c_5	0	1	0	-1	0	0	0	0
c_6	0	1	0	0	-1	0	0	0
c_7	0	1	0	0	0	-1	0	0
c_8	0	1	0	0	0	0	-1	0
c_9	0	0	1	-1	0	0	0	0
c_{10}	0	0	1	0	-1	0	0	0
c_{11}	0	0	1	0	0	0	-1	0
c_{12}	0	0	0	1	-1	0	0	0
c_{13}	0	0	0	1	0	-1	0	0
c_{14}	0	0	0	1	0	0	-1	0
c_{15}	0	0	0	1	0	0	0	-1
c_{16}	0	0	0	0	1	-1	0	0
c_{17}	0	0	0	0	1	0	-1	0
c_{18}	0	0	0	0	1	0	0	-1
c_{19}	0	0	0	0	0	1	-1	0
c_{20}	0	0	0	0	0	1	0	-1
c_{21}	0	0	0	0	0	0	1	-1

Table 3.2: *Incidence matrix of the 8 cell stage (the matrix 3-2.1)*

3.2.1 Incidence matrix normalized by TDO

Neither the incidence matrix in table 3.2, nor the adjacency or contact matrix in table 3-2.1 give a unique representation of the stage of biological development. The algorithm "Tactical Decomposition by Ordering" (TDO) permutes rows and columns of the incidence matrix so that a normalized matrix results (see for details D. Betten [BBT01] and [BB99]). To seek a unique mathematical representation of the 8-cell development stage, the incidence matrix 3.17 may be normalized using an algorithm that eventually results in another unique index permutation to reorder the cells and their contacts (see matrix plot of figure 3.18).

The normalized form represents a set of geometries, in which one geometry is associated with one stage of development. The number of possible topologies



$|Aut| = 2$

$Aut = \langle (2\,3) \rangle$

Figure 3.18: *Incidence matrix of figure 3.17 normalized by the algorithm "Tactical Decomposition by Ordering" (TDO). Rows are labeled by cell numbers as in figure 3.17. Columns are labeled as in figure 3.17, but the counting is continued by letters in ascii sequence. Aut indicates the automorphisms where (2 3) means that the rows 2 and 3 may be interchanged to get the two automorphic geometries for this incidence matrix*

is given by the automorphism group (see figure 3.18). Such normalized forms are needed for a systematic representation or classification of connectivity in a network. In figure 3.19 and 3.20, a sequence of normalized incidence matrices ordered by the TDO method are given for the very early developmental states obtained by Delaunay triangulation of nuclear positions. Since the incidence matrix becomes large for increasing numbers of nodes (or cells), only the 4-cell-, 6-cell-, 8-cell-state in figure 3.19 and the 12-cell-, 15-cell-states in figure 3.20 are displayed here. The number of automorphism is one for cell-states larger than 8, i.e. for 12-cell-states and higher there is a unique geometry in the development.

A part of the development of the organism is expressed in the transformation from the 4-cell-state to the 15-cell-state and further by mapping the consecutive matrices (i.e. the geometry they represent) to the next state of development. Since the matrices (matroid's, i.e. matrices with only the entries 0, 1 or -1) may be converted in a geometry, the maps describe a development of a geometric shape in space. Particularly in 4D space (time and 3D space coordinates), a TDO-normalized matroid describes the complete development of shape. Although in 3D, the sequence of the maps may give information about the rules that govern the developing shape of an organism, it is more promising to search in the 4D

incidence matrix for a certain pattern or editing mechanism of the pattern for the reconstruction of the organism.

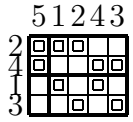
3.2.2 Reordering by membership to strips and surfaces

A special reordering of the sequence of cells in the adjacency matrix is to sort them into two classes depending on the property belonging to the surface or to the interior of the cell aggregate. Originally the program `qhull` (see [BDH96] and [PS88]) was written to find the convex hull around a number of given points (nuclei) (i.e. the smallest convex set containing the points). The output from `qhull` could be used to find the classes i.e. nuclei in the hull and the others (the interior). Note, in the 8 cell state, all cells have a contact to the outside. Therefore, this classification is effective at higher numbers of cells (see E.0.4). This reordering splits the force-flow relation (see appendix B) into surface relations and volume relations. This might be of help to find the real sorting mechanism, which reorders the cell from the place of birth to their final position in the cell aggregate.

Stripes (see appendix B.0.6) may be used as an additional ordering of cells in the hull (surface) or the interior. This reordering process uses topological information obtained by a Delaunay decomposition of a simplicial-complex, which is based on contacts between the nearest neighbors and rejects contacts to cells far apart in the adjacency matrix (as e.g. contacts of neurites).

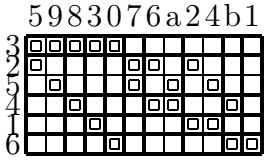
3.3 Adjacency and distance of cells and their nuclei

Two cells (objects) are adjacent if they are neighbors (close together, but do not necessarily touch each other). Thus, two nuclei are usually adjacent, if the corresponding cells touch. The adjacency is measured by the distance of the



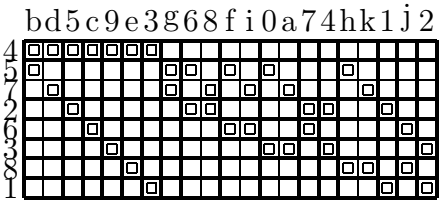
$|Aut| = 4$

$$Aut = \langle (1\,2), (3\,4) \rangle$$



$|Aut| = 2$

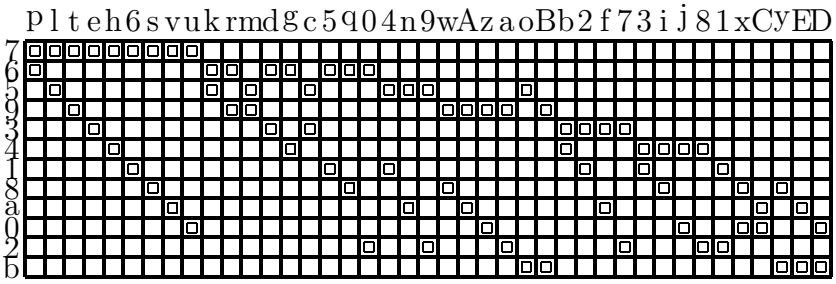
$$Aut = \langle (2\,3) \rangle$$



$|Aut| = 2$

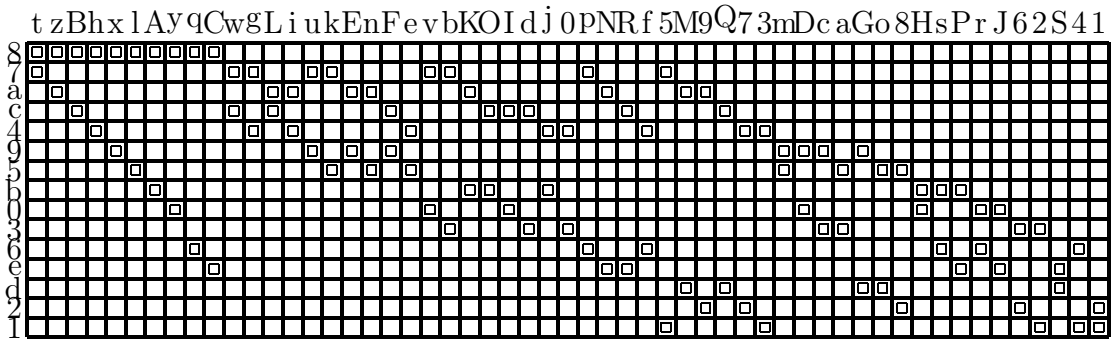
$$Aut = \langle (2\,3) \rangle$$

Figure 3.19: *Output of the program "Tactical Decomposition by Ordering" (TDO) to calculate (see section 3.2.1) normalized incidence matrices. The output represents the connectivity between cells of the development of the* $4-, 6-, 8-$*cell state (corresponds to the number of rows in the graphs of matrices). "Aut" gives the possible automorphism. Row indices are cells, whereas the column indices are cell contacts. The number at rows denotes the cells ordered according to their position in the cell lineage tree. The one digit and letter at the top of the matroid are a symbolic ascii sequence for the connectivity elements starting with* 1 *proceeding with lower case and upper case letters. Rows and columns are permutated according to the TDO algorithm from the original sequential order. A square in a grid mesh of the matrix is equivalent to a dot in the dot plots and means that the cell in the row is involved in the contact of the column.*



$|Aut| = 1$

$Aut = \langle id \rangle$



$|Aut| = 1$

$Aut = \langle id \rangle$

Figure 3.20: *Same as figure 3.19, except that the incidence matrices of the 12-cell state (upper graph) and 15-cell state (lower graph) are represented.*

objects. A simple distance measure for nuclei is their Euclidean distance in space (see appendix A and section 3.4).

The square of Euclidean distance d_{n_i,n_j} between the nuclei n_i and n_j is :

$$d^2_{n_i,n_j} = (x_{n_i} - x_{n_j})^2 + (y_{n_i} - y_{n_j})^2 + (z_{n_i} - z_{n_j})^2 \qquad (3\text{-}3.2)$$

where (x_{n_i},y_{n_i},z_{n_i}) and (x_{n_j},y_{n_j},z_{n_j}) are the coordinates of the nucleus n_i and n_j

respectively.

d_{n_i,n_j}	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8
n_1	0	231.90	295.17	225.07	395.61	397.91	423.75	487.23
n_2	231.90	0	348.43	237.03	348.73	273.46	242.81	364.80
n_3	295.17	348.43	0	296.67	205.93	406.47	381.22	402.95
n_4	225.07	237.03	296.67	0	265.79	203.16	292.01	294.76
n_5	395.61	348.73	205.93	265.79	0	268.58	247.33	212.88
n_6	397.91	273.46	406.47	203.16	268.58	0	161.61	141.74
n_7	423.75	242.81	381.22	292.01	247.33	161.61	0	164.22
n_8	487.23	364.80	402.95	294.76	212.88	141.74	164.22	0

(3-3.3)

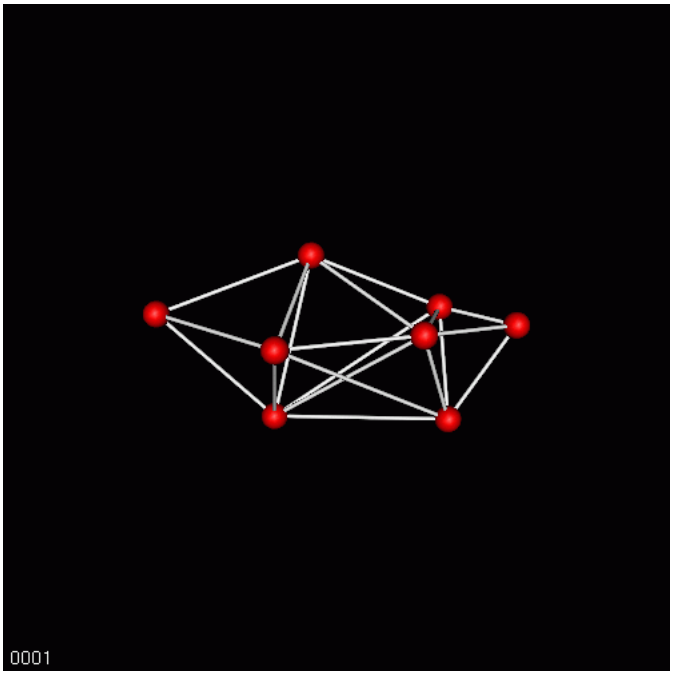


Figure 3.21: *Ball-stick model of the 8 cell state. The balls are at the positions of the nuclei. The sticks connect balls (cell neighbors) as identified by Delaunay triangulation.*

As distance the values d_{n_i,n_j} may be summarized as elements (i, j) in a distance matrix. For the 8-cell state of figure 3.21, the distance is formulated as a distance matrix between the individual nuclei in table 3-3.3. This matrix is symmetric with zeros on the diagonal. The distance matrix gives the distances between all nuclei.

Of particular interest for the network of cells is their neighbor relationship. The mathematical construction of the Delaunay triangulation produces a rela-

tionship between the nearest neighbors (see [GR97]). It connects the nuclei such that a net of triangles (in the plane) and of tetrahedra (in 3D-space) results (see figure 3.21), so that no nuclei lie within a triangle or tetrahedron, and even no nuclei lie within the enclosing circles or spheres through the points (or nuclei) that form the triangle or the tetrahedron. The matrix table 3-3.4 shows the result for the nuclei (n_1 to n_8) that are adjacent according to their distance.

	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8
n_1	0	1	1	0	1	0	0	0
n_2	1	0	1	1	1	0	1	0
n_3	1	1	0	1	1	1	1	0
n_4	0	1	1	0	1	1	1	1
n_5	1	1	1	1	0	1	1	1
n_6	0	0	1	1	1	0	1	1
n_7	0	1	1	1	1	1	0	1
n_8	0	0	0	1	1	1	1	0

(3-3.4)

The entries in the matrix are given by the rule:

> "if two nuclei (points) are connected in the triangulation net, a 1 else a 0 is entered as a matrix element"

As the contact matrix, this matrix is also symmetric.

The adjacency or contact may be represented by a stick or bond connecting the nuclei like in figure 3.21. In this adjacency matrix of nuclei, the contacts may not necessarily reflect the real neighborhood of the cells harboring the corresponding nuclei. Cells being neighbors due to the adjacency matrix may in nature not be in contact with each other, since the nucleus is not always positioned at the center of a cell (compare with a nerve cell as an extreme example). The triangulation only approximates the contacts of cells in nature. Nevertheless, if there is no more information available, then adjacency of cells obtained from the position of the cell nuclei it is a good approximation to cell contacts in reality. For the establishment of the real neighborhood, the hulls (membranes) of the cells should be known to touch.

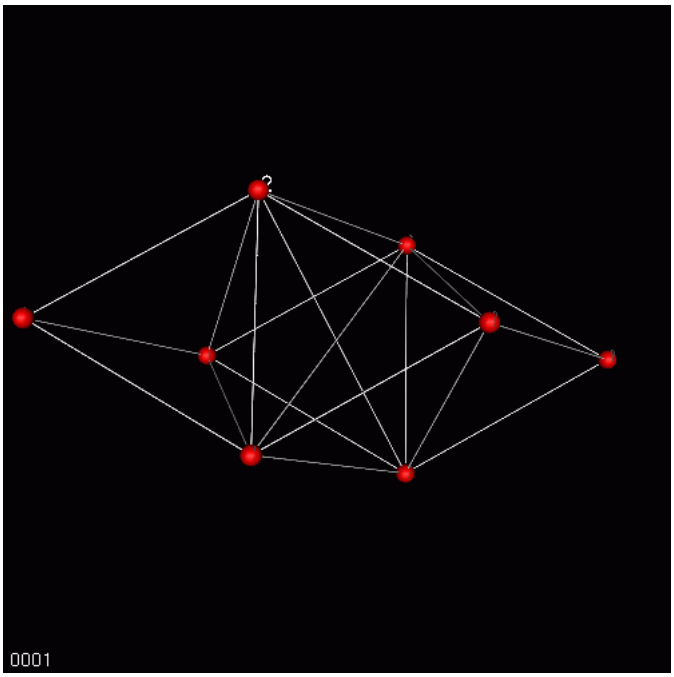


Figure 3.22: *Same as figure 3.21 except that all sticks (connections between neighboring nuclei) are of the same length. Thus, the distances between all nuclei of connected cells are as far as possible the same, i.e. this figure shows an abstraction with equal distances of nuclei of the 8 cell state of figure 3.21*

In the considered case both matrices for the contact and for the adjacency (compare 3-3.4 and 3-3.5) of the eight cell state are identical.

The adjacency matrix 3-3.4 and the distance matrix (table 3-3.3) may be combined, if they are multiplied element for element to give the matrix 3-3.5. Thus a sparse matrix is obtained, which saves machine resources at high numbers of nuclei. Moreover, the full matrix 3-3.3 could be regained from matrix 3-3.5 by the method described in section 3.4.

	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8
n_1	0	231.90	295.17	225.07	0	0	0	0
n_2	231.90	0	348.43	237.03	348.73	273.46	242.81	0
n_3	295.17	348.43	0	296.67	205.93	0	381.22	0
n_4	225.07	237.03	296.67	0	265.79	203.16	292.01	294.76
n_5	0	348.73	205.93	265.79	0	268.58	247.33	212.8849
n_6	0	273.46	0	203.16	268.58	0	161.61	141.74
n_7	0	242.81	381.22	292.01	247.33	161.61	0	164.22
n_8	0	0	0	294.76	212.88	141.74	164.22	0

(3-3.5)

Besides, in the case of the tetrahedra adjacency there are other adjacency relations of sub-complexes, i.e. between objects of same types, as e.g. triangles, lines, pentahedra etc. (see appendix A and B), that have a common border. A connectivity between similar objects exists (see DVD E.0.4) for these sub-complexes also.

The above procedure yields another adjacency matrix between the tetrahedra or triangles of the triangulation. According to the vector space concept (see appendix B), there exists another distance matrix in the co-space. Here, the points are the corners, at which the membrane of several cells meet (in the 3D-space, usually 4 cells have a common corner). In the topological view, to each corner of the co-space is assigned a tetrahedron in the space of the Delaunay triangulation (see appendix B). The distance matrix of these corners (points in the co-space) relates the tetrahedra and generate an adjacency matrix for the tetrahedra themselves (see matrix 3-3.6). In that case, the tetrahedra T_i are listed instead of the single nuclei n_i. A 1 is entered if two tetrahedra have a common triangle (surface or faces that is shared by two tetrahedra in touch). The matrix 3-3.6

	T_1	T_2	T_3	T_4	T_5	T_6	T_7
T_1	0	0	1	0	0	0	0
T_2	0	0	0	0	1	1	0
T_3	1	0	0	1	1	0	0
T_4	0	0	1	0	1	0	0
T_5	0	1	1	1	0	1	0
T_6	0	1	0	0	1	0	1
T_7	0	0	0	0	0	1	0

(3-3.6)

may be derived by the computer program `qhull` (see [PS88] and [BDH96]), which calculates the Delaunay triangulation and lists the tetrahedra by its constituting points (nuclei) from which the common triangles may be obtained.

3.4 Reconstruction of the position of nuclei

From a distance matrix of nuclei (e.g. table 3-3.5), the distribution of nuclei in the egg can be reconstructed. The first step in the reconstruction of the nuclear positions is to find a coordinate system. For that purpose, e.g. in 3D space four points are needed (one point for the origin and three points to define the three coordinate axis). One may select four points arbitrarily by their columns and their corresponding rows from the distance matrix. The sub-matrix formed by the four columns and rows should have no zero elements except on the diagonal. One point may be considered as the origin $(0, 0, 0)$ e.g. P_0. The coordinates of the other three points should fulfill the restrictions given by the distance sub-matrix:

	P_0	P_1	P_2	P_3
P_0	0	s_{01}	s_{02}	s_{03}
P_1	s_{01}	0	s_{12}	s_{13}
P_2	s_{02}	s_{12}	0	s_{23}
P_3	s_{03}	s_{13}	s_{23}	0

(3-4.7)

A second point may be selected arbitrarily from the three points left over (e.g. P_1 in figure 3.23), which is given the coordinates (s_{01},0,0). The vector in the direction from P_0 to P_1 could be chosen as the x-axis of the coordinate system.

From the further two points left, one may select a point as P_2 to find the y-axis. P_2 should have the distances s_{02} to P_0 and s_{12} to P_1

$$x_2^2 + y_2^2 = s_{02}^2 \tag{3-4.8}$$

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 = s_{12}^2 \tag{3-4.9}$$

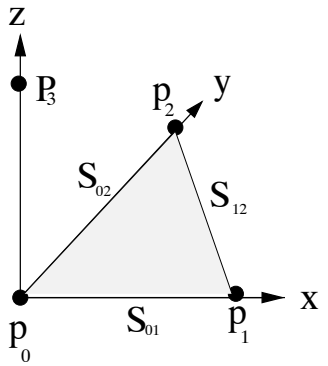


Figure 3.23: *Sketch for construction of a coordinate system by the four points labeled P_0 to P_3*

where the coordinates of P_1 and P_2 are (x_1, y_1, z_1) and (x_2, y_2, z_2) resp. and $z_2 = 0, z_1 = 0, y_1 = 0$.

From the equation system 3-4.8 to 3-4.9, one gets the coordinates for P_2 as

$$(x_2, \sqrt{s_{02}^2 - x_2^2}, 0), \qquad \text{where } x_2 = -\frac{1}{2}\frac{s_{12}^2 - s_{02}^2 - s_{01}^2}{s_{01}} \tag{3-4.10}$$

The y-axis points from P_0 to P_2. Note the orientation of the coordinate axis may be chosen arbitrarily. The coordinates (x_3, y_3, z_3) of the final point P_3 may be obtained from the equations

$$x_3^2 + y_3^2 + z_3^2 = s_{03}^2 \tag{3-4.11}$$

$$(x_3 - x_2)^2 + (y_3 - y_2)^2 + z_3^2 = s_{23}^2 \tag{3-4.12}$$

$$(x_3 - x_1)^2 + y_3^2 + z_3^2 = s_{13}^2 \tag{3-4.13}$$

The z-axis completes the coordinate system as the direction from P_0 to P_3.

Each point added later has to be placed in the space spanned by the coordinate system to fulfill the given distance requirements fixed by the distance matrix.

This method may be simplified and performed more systematically if the triangulation is known. Beginning with a tetrahedron (where three of the nuclei

form the coordinate system, and the remaining one serves as the origin). Three nuclei of a tetrahedron constitute one of its surface triangles. The adjacency matrix of tetrahedra (see matrix 3-3.6) may show, which tetrahedron has also these triangles as surface (facets). Now, the fourth nuclei of the neighbor tetrahedron is taken, and its position is calculated from the distances to those triangle points (nuclei) that constitute the shared facet. Usually, the calculation gives two positions for the fourth tetrahedron point, one on each side of the common triangle (i.e. shared faced).

Only the position that is not placed on the side of the last tetrahedron is valid (see E.0.4 for the reconstruction program). This method of growing a tetrahedra sequence yields the positions in a straight forward way. The nuclear distribution in figure 3.22 is calculated by the described method.

3.5 First abstraction level for spatial distances of nuclei

In figure 3.22 the topology of the 8-cell state was so reconstructed in that all possible distances of the triangulation were set to one. Another interpretation of the reconstructed state is obtained by scaling all distances to mean distance values of the matrix (see 3-3.5). This will not change the nuclear distribution of figure 3.22, but will scale the bonds between the nuclei only. Figure 3.24 shows in a histogram the distribution of distances of matrix 3-3.5.

More obvious is the usefulness of the spatial distance distribution between nuclei, if the histogram in figure 3.24 is computed for the 98-cell state (see figure 3.25) instead of the 8 cell state.

The distribution is close to a normal distribution at high numbers of cell states. This abstraction suggests that in the early embryonal development the

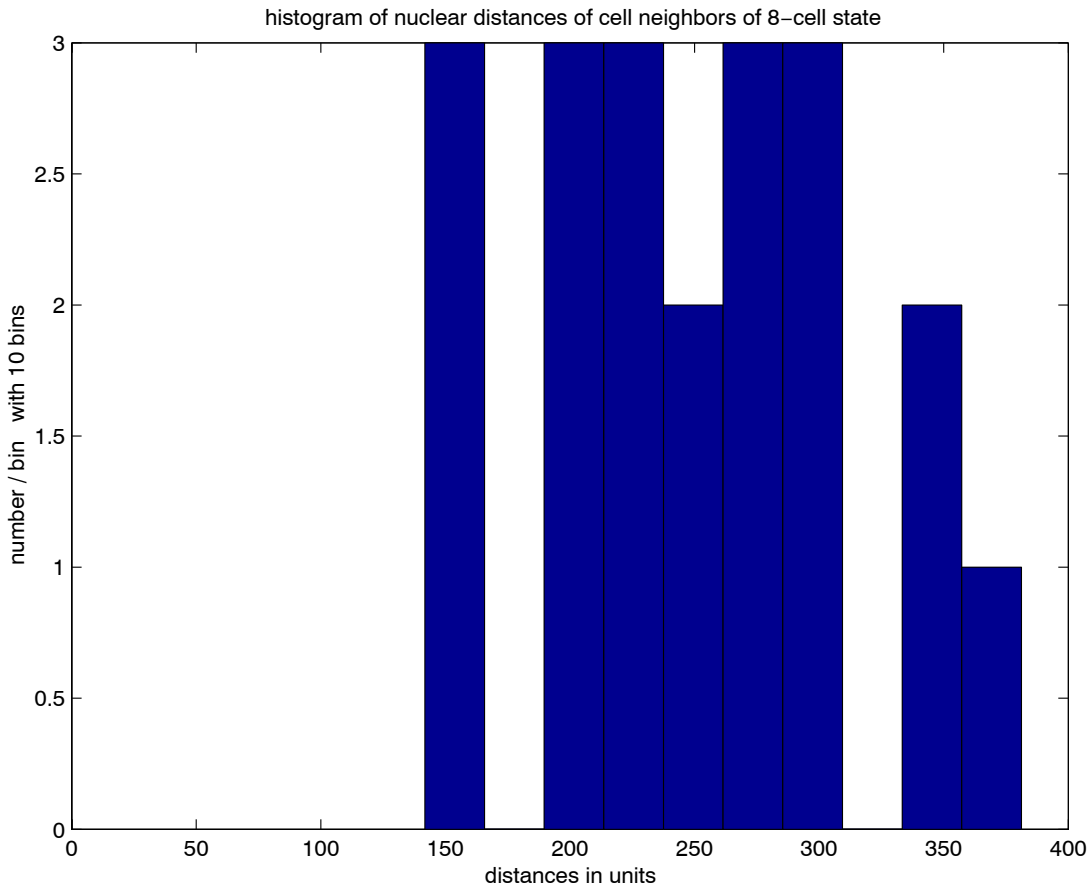


Figure 3.24: *Histogram of the distances of the nuclei in table 3-3.5 in the Delaunay triangulation of the 8 cell state of figure 3.21.*

spatial topology is more essential than distances between and size of the cell for the aggregation or the sorting of the cell mass are.

3.6 Development in 4D

The representation in Delaunay triangulation and Voronoi tessellation may be extended into four dimensions (space and time) or even higher (see appendices A and B). Space-time coordinates may be introduced to facilitate the simplification and to organize knowledge about an underlying network of developmental processes.

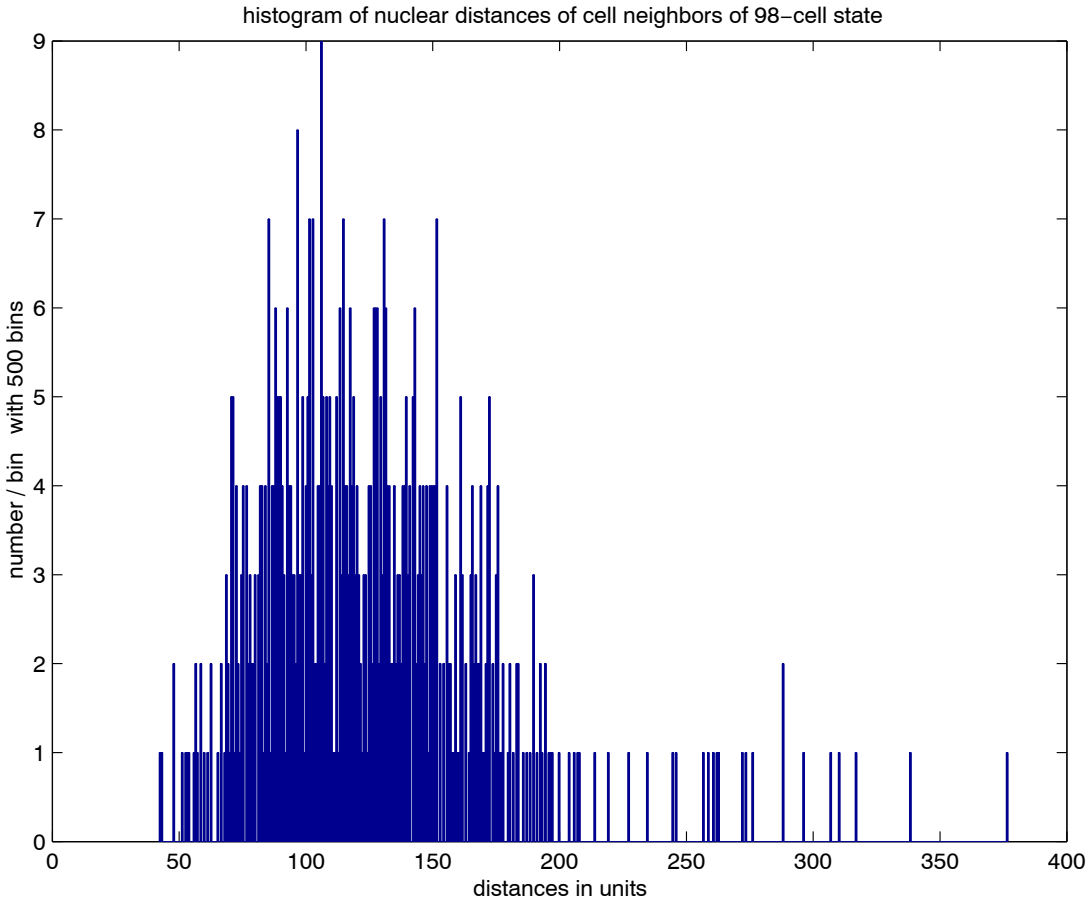


Figure 3.25: *Histogram of spatial distances of nuclei as in figure 3.24 but for the 98 state.*

In the four dimensional space a cell has a 4D volume, which is the 3D volume extended in the time direction for the life span of the cell. In the 4D space the development is static. In 4D, the shell of the egg forms a rigid cylindrical body in which the development takes place until hatching. The cylindrical 4D volume is subdivided into 4D-domains, the cells which are generated and topologically ordered by the time the cell exists. The topology of the 4D representation (i.e. the adjacency of 4D sub-domains) reflects the whole early development. There are several ways of characterizing the development in the egg by coordinates of 4D points i.e. by assigning a 4D-point to a 4D-cell-volumes.

One way is to assign to each 4D-cell-volume a center that has as time, the division time and as space coordinates, the space coordinates of the nuclei just

prior to division. Another way is to select the mean time of the life span (from originating division to terminal division) as the time of the center and the position of the nucleus at that time in space or a weighted mean (like the center of mass) may be chosen.

The 4D representation is hard to visualize but advantageous in a systematic treatment, as e.g. by an abstraction, a reconstruction or to find growth rules in the topology. Besides, the large amount of algebraic data to be processed and the difficulty in imagining the Delaunay triangulation and Voronoi tessellation in 4D space, a movie may be extracted and visualized as seen in 3.28.

Moreover, as in the 3D case adjacency matrices may be obtained in the 4D space, also. However, the matrices are huge and sparse, since they not only comprise all 3D matrices but additionally their connectivity in the 4D context. Again, if the real contact of the cells would be obtained manually, it would be even more time consuming than in the 3D case. Therefore, the adjacency matrix is approximated in analogy to the 3D case by Delaunay triangulation. A program to compute and display such matrices can be found in the program collection to this work on the DVD (see E.0.4). Instead of tetrahedra, pentahedra (the regular polyhedron in 4-dimensional space made of five 4D points i.e. 4-simplex) are generated by the Delaunay algorithm. The Delaunay decomposition is performed by the program `qhull` (see [BDH96]) with parameter settings for the central 4D point in the cell. Since the dot density is quite high, a dot plot of a part of the 4D adjacency matrix (i.e. connectivity of pentahedra) is shown in figure 3.27 only.

Figure 3.26 displays the nuclei adjacency for the 300-cell state in space and time from the beginning of the zygote division. For missing spatial coordinates of a cells in the database, interpolated values are used.

The adjacency matrix displays the neighborhood only and states nothing about the size and extension of the pentahedra from which it is derived. Further, the scale of space and time coordinates had to be adjusted. A factor of 0.002 for the time units was applied (see section 4.8).

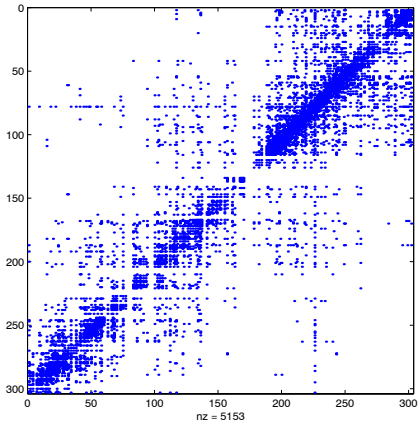


Figure 3.26: *Dot plot of the adjacency matrix of 4D cell centers obtained by Delaunay triangulation of 300 4D-cells. The cell lineage tree is contained in this representation as a spanning tree.*

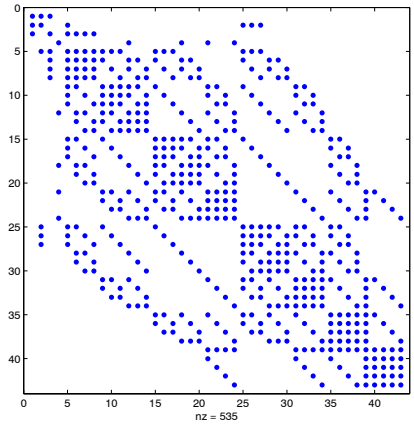


Figure 3.27: *Dot plot of the pattern of the connection of 43 pentahedra by their facets (tetraherda) of the 4D Delaunay triangulation of figure 3.26*

3.7 Triangulation of surface (outer cells)

The egg is a limited object and has as such a surface or a boundary. There is a close relation between the surface or the hull and a triangulation. A computer algorithm (like e.g. `qhull` [PS88]) may be used to find the hull or cells forming the shell of cells enclosing the interior mass of the egg. The decomposition of the egg in cells that form a hull (or a sheet of cells under the eggs shell) and those that comprise the inner is biological relevant since cells in nature may be programmed to form sheets or rods. These cells do not move into a third dimension but remain in the two dimensional layer.

The program `qhull` does not only generate the Delaunay and Voronoi decomposition of the space, but also the hull of a set of points (nuclei). The cells in the hull form a surface layer. In the beginning of the development, all cells belong to the hull, and with increasing cell numbers, more cells appear in the interior. However, in the interior cell mass, holes may be formed. At this state, the formation of holes has been neglected in this work and in the computation of figure 3.28, but it may be dealt with later [ELZ00]. Holes as construction elements of nature have to be considered to approximate reality, since they allow further decomposition of the cell mass.

The hull visualization in 3D looks similar to the visualization of the whole net of ball-sticks (see figure 3.28) since firstly only the interior connections are missing, which are mostly hardly visible and secondly, in the beginning of the development all cells belong to the hull.

The time sequence of ball stick pictures, like the one shown in the movie of figure 3.28 visualizes the early development in the egg. To emphasize the events in relation to the cell lineage, the nuclei may be colored, according to the table of colors 3.3. The color reflects their origin with respect to the founder cells (fig. 1.1) (AB, C, D, E, MS), which form the body of the nematode. P_0 to P_4 are the germ cells of the germ line that carry on (forward) the genetic information

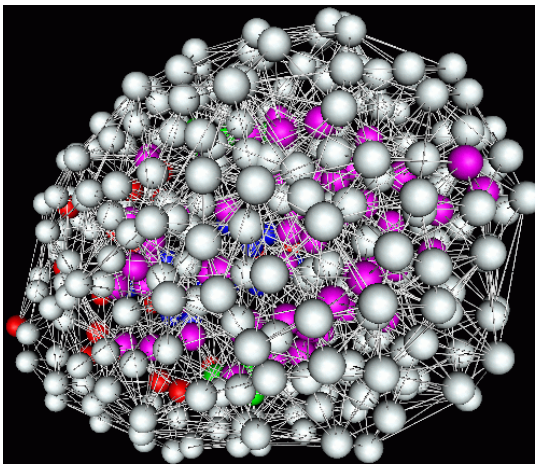


Figure 3.28: *The video freeze image shows a ball-stick (wire frame) model of the* $\approx$ 360 *cell state. Founder cell sub-lineage colored AB = silver, C = red, D = green, E = blue Other = magenta according to color table 3.3*

founder cell	color
AB	silver
C	red
D	green
E	blue
MS	yellow
other	magenta

Table 3.3: *Color map of the founder cells*

protected either by the body of the organism, or as huge number eggs excreted from the body.

Note firstly, in movie of figure 3.28, the hull is constituted only by cells derived from the lineage tree of the *AB* founder tree and secondly, that at some times, only at the polar ends of the hull, a cell of another type of founder cell is present. To explore this aspect, more detailed studies are necessary, e.g. using a colormap that represents cell fates or biological origin in the early development, like endo- or mesoderm etc.

3.8 Segmentation

The essential data for the database (see methods 2.5.4) have to be extracted from the experimental 3D-stack of 2D microscopy images. The automated segmentation of the images of an egg into nuclei, membrane contacts and cell areas would be a prerequisite for the effective analysis of the 3D stacks. However, segmentation is a difficult task and the algorithm are still not working satisfactorily. Auto segmentation was studied and will only be considered superficially. At the time the certain (almost error free) location of the nuclei- and of the cell-wall-position is done manually. Membranes of cells could be clearly stained in thin sections for the electron microscope (see [Sch78]) to establish the contacts of cells in the egg. However, this is a resource-consuming task, which at the time could be done only at a fixed time on a fixed specimen.

The practical data analysis is realized with a tool that is still at the experimental state. The tool basically processes images into a topological data set of attributes. The tool (named Topologizer see fig. 3.31) is written in Tcl/Tk, Perl, C++ and applies VTK (www.kitware.com) , MATLAB (www.mathworks.com). It is made of three components which are included in the DVD (see appendix E.0.4).

3.8.1 Segmentation results

The development of the cell mass up to the end of the pre-embryonic phase is recorded by a microscope (see methods 2.1.3). The microscopy data records include images and configuration information (i.e. time, position, devices settings, etc.). These data records are processed to build a database containing all the developmentally relevant data. One processing aspect is the analysis of the images for the positions of cell-like objects (i.e. nuclei, encircling membrane, granulation etc.). The knowledge of these cell data is conditional for the analysis of structures in the development.

The decomposition of images into meaning full objects is usually performed by the human visual system. The process of perceptional recognition of e.g. a cell-like structures is an inherent ability of the human brain. However, attempts are made to design computer algorithms with partitioning capabilities that are consistent with the human segmentations (see [TM01]).

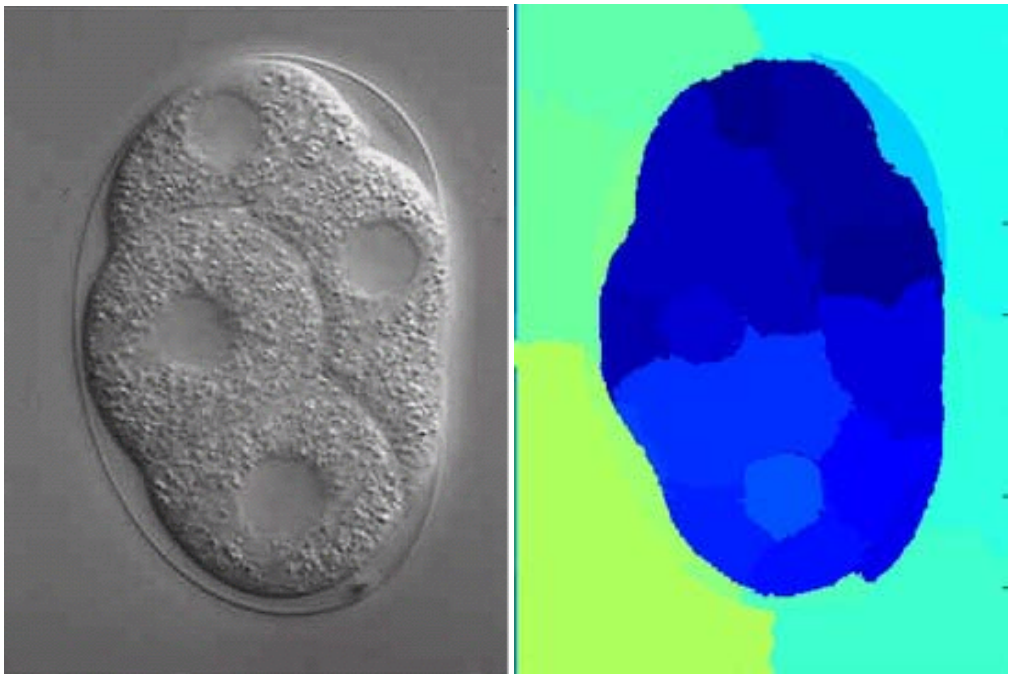


Figure 3.29: *Segmentation of the left micrographic image (Nomarski) of the database (demo1.sbd on DVD E.0.3) with the algorithm "normalized cuts"(see [TM01]) gives the colored right panel. x-y pixel resolution 364x275 at 256 gray values (8-bit)*

The computer algorithm system is briefly described in appendix C and the results of its application on a microscopy image is demonstrated in figure 3.29 and 3.30 on images of different resolution and origen. The selected data in figure 3.29 left panel and the typical data in figure 3.30 upper left panel are delivered by microscopes (see section 2.1.1). The resolution of the Nomarsky image (figure 3.29 left panel) is 256 gray scales instead of 4096 gray scales of the CLMS image in figure 3.30 upper left panel. In the Nomarski image of figure 3.29 left

panel the eye recognizes known structures as e.g. nuclei of cells. The normalized cuts segmentation delivers the nuclei as segments whereas almost all other studied algorithms (as the Matlab Image Processing Toolbox, vtk, or techniques like tensor voting (see [MTL00])) fail or work only with a very limited set of data or with detailed a priori information of the segments searched for. The normalized cuts segmentation of the CLMS data of figure 3.30 upper left panel delivers the visible nuclei and in addition it dissects the eggs interior during early development (see figure 3.30 lower left panel) because it uses texture, conture and color properties in the algorithm. Furthermore, the algorithm resolves differences in the gray scale which are not detected by the eye.

Even for data at lower resolution (see figure 3.30 upper right panel) the segmentation is stable and estimates roughly the nuclei positions (see figure 3.30 lower right panel).

The segmentation was performed with a default set of parameter values. This parameter set is stored in a file ("ncuts_param.txt" see DVD directory E.0.6) and read by the algorithm. The parameter set is not jet optimized for the segmentation problem studied in this work.

The "false-color" of one segment is chosen due to its texture, color and conture characteristics. Unfortunately the present algorithm of normalized cuts represents the same segments in two successive images in different colors. One may view the video frame by frame and compare the structure in each image to get an impression about changes in time of segments in the considered z-layer of the egg.

The segmented images may be used as object mask for the further 3D segmentation in the program Topologizer of figure 3.31.

3.9 Coding the topology

The data of the object to be visualized consists of a geometric (metric) part (i.e. point coordinates or distances) and a topological part. The later describes the connectivity within or between the object. Usually the geometrical data are represented by real numbers (coordinates.), where the topological data are given by relations between abstract names. In the set of points, to each point (nucleus) is assigned an integer (as its name). Since the nuclei of the cells in the egg could be named according to the cell lineage (see lineage in figure 1.5 and figure 3.5), these names could be used in the datasets as well. However, that is not recommended since the programs applied are coded with integers as "point-names". Thus, the cell lineage tree may be encoded to integers in the alphabetical sequence of names. Now, the connection even in the 4D net can be represented as a sequence of integers for lines, poly-lines and polygons (closed poly-lines) as well as for triangles, triangle strips (poly triangles) and closed triangle strips (which are an equivalent to polygons or "trianglegons").

This sequencing procedure may be extended to the tetrahedron, the tetrahedron strips and the closed tetrahedron strips in the 3D space and further to pentahedron, pentahedron strips and closed pentahedron strips in 4D space. In the 4D case of the eggs development, the "pentahedron strip representation" of the Delaunay triangulation gives a very compact representation of the connectivity of the structure (i.e. the topology of development processes in 4D). Note, that the egg has a closed surface and the nuclei of the cells located in the 2D-surface form a closed net of triangles. Also this applies to the developing structure represented in 4D as a surface of connected tetrahedra.

3.10 Spring embedding

Spring embedding is a method (see figure 3.32 [SSL00]) to reduce the degree of freedom in an n-dimensional entity. Here it is the 4D connectivity. The method has been applied to the dataset of the connection of the 4D triangulation of section 3.6 to study its possibilities in lowering of the dimension of the original space (embedding).

3.11 Physical space of chains (topology)

In the appendix B, the topological relationships of the neighbors (adjacency) is interpreted as the base of a system of vector spaces. In the following this interpretation will be illustrated with the 8-cell state (see figure 3.21) as an example. For other cell states, a program is available (in the program package of this work on DVD in appendix E.0.4), which can be activated to display dot plots and matrices in the different vector spaces. The additional property of the topological space to become a physical space, is the orientation of the topology (see appendix B.0.2).

In the incidence matrix 3.2 of the 8-cell state, the 21 line segments (bonds of cell contacts) constitute the rows. In the distance matrix 3-3.5, the lengths of the line segments are given as distances.

The 21 line segments form a base of the vector space (see appendix B). A vector chain in this space may be constructed, if the distance elements of the matrix (see 3-3.5) are selected as coefficients.

$$c = 231.9c_1 + 295.17c_2 + \ldots + \ldots c_{21} \tag{3-11.14}$$

where the base vectors are named according to the index of the rows in matrix 3.2. Note that the sparse matrix 3-3.5 is turned into a chain.

The same could be done with the triangles, which will result in a 2-chain, c^2, having the areas of the triangles as coefficients in units of 1.0e+04 square pixels

$$c^2 = 3.4696c_1^2 + 2.3142c_2^2 + 3.3877c_3^2 + \ldots + \ldots + 3.0799c_n^2 \qquad (3\text{-}11.15)$$

and for the 3-chain of tetrahedra having their volumes as coefficients in units of 1.0e+06 volume pixels.

$$\begin{aligned} c^3 \;=\; & 2.1098c_1^3 + 0.7791c_2^3 + 2.0630c_3^3 + 0.6887c_4^3 + 1.6814c_5^3 + \\ & +0.3070c_6^3 + 0.7124c_7^3 + 1.0380c_8^3 \end{aligned}$$

The areas of the triangles and the volumes of the tetrahedra may be calculated by the equations 1-0.3 to 1-0.5 given in appendix A.

The same procedure may be applied to the co-chains to represent volumes of cells in the co-space and the contacting surfaces between cells as chains in the vectors co-space.

These chains, which represent a geometric content, are of interest for the shape of the object they describe. Since the time parameter is missing, they do not reflect the development. However, in the 4D space (spatial space and time coordinates) the development is incorporated in the coefficients and hence contain the description of the evolution of the object (i.e. the organism development) in time. In the 4D case, the chains are huge and should be verified by the corresponding program in the program collection (see DVD in appendix E.0.4).

The abstractions of section 3.1 and section 3.4 simplify the chains in the 3D- and 4D-space, because many of the coefficients are equal, but the base of the chains is still large and rules have to be established, which reduce the base.

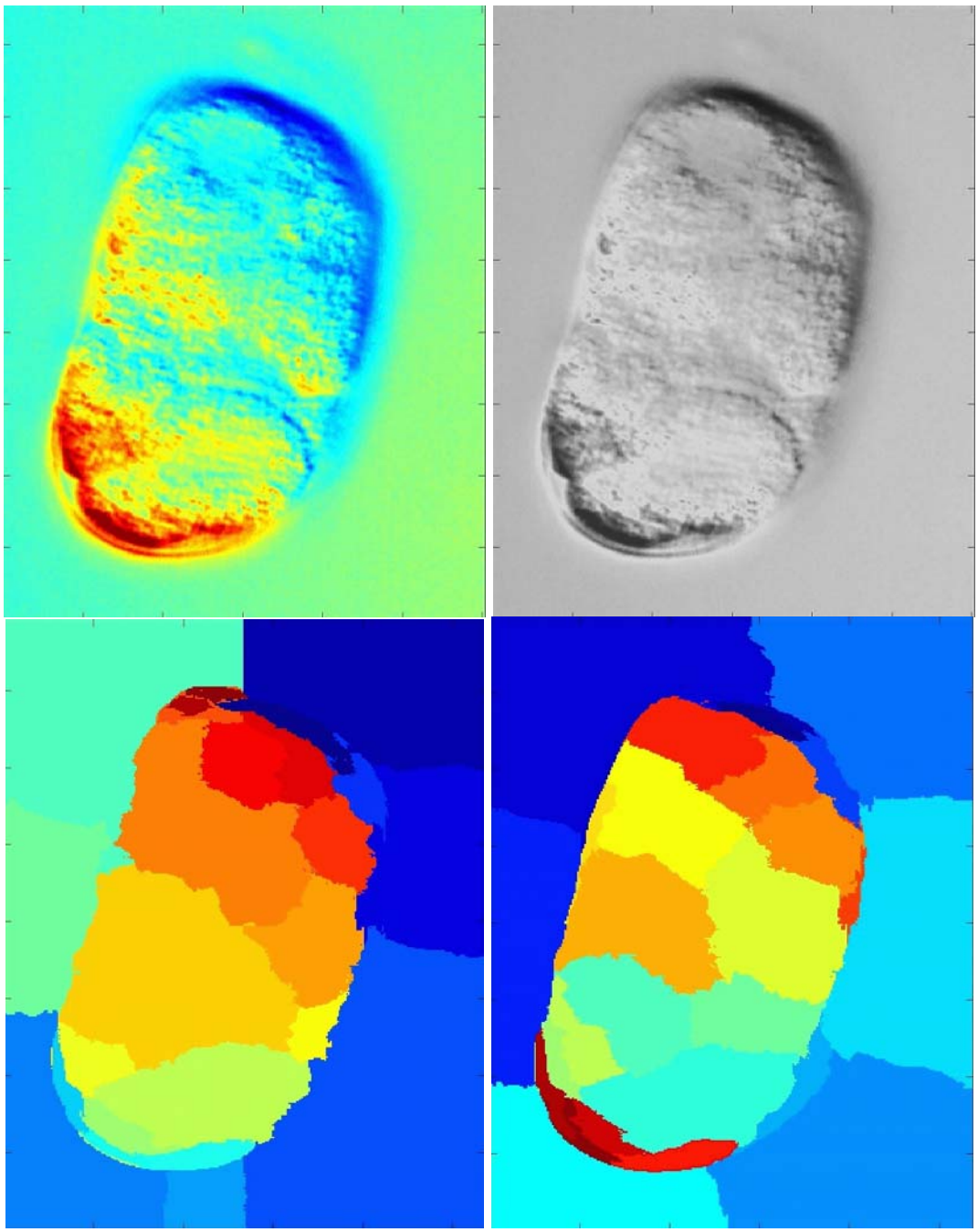


Figure 3.30: *Segmentation of Confocal Laser Scanning Microscope images (see section 2.1.1) with the algorithm "normalized cuts" (see [TM01]). The left upper panel is a 2D image of a 3D stack with a x-y pixel resolution 435x341 at 4096 gray values (12-bit). In the upper right panel the intensity of the same images are scaled down to 8-bit or 256 gray values. The lower panels show the corresponding segmented images obtained by the algorithm normalized cuts. The upper and lower left panel are linked to a video sequence by which the segmentation in the developing egg may be studied (exemplified on 30 frames).*

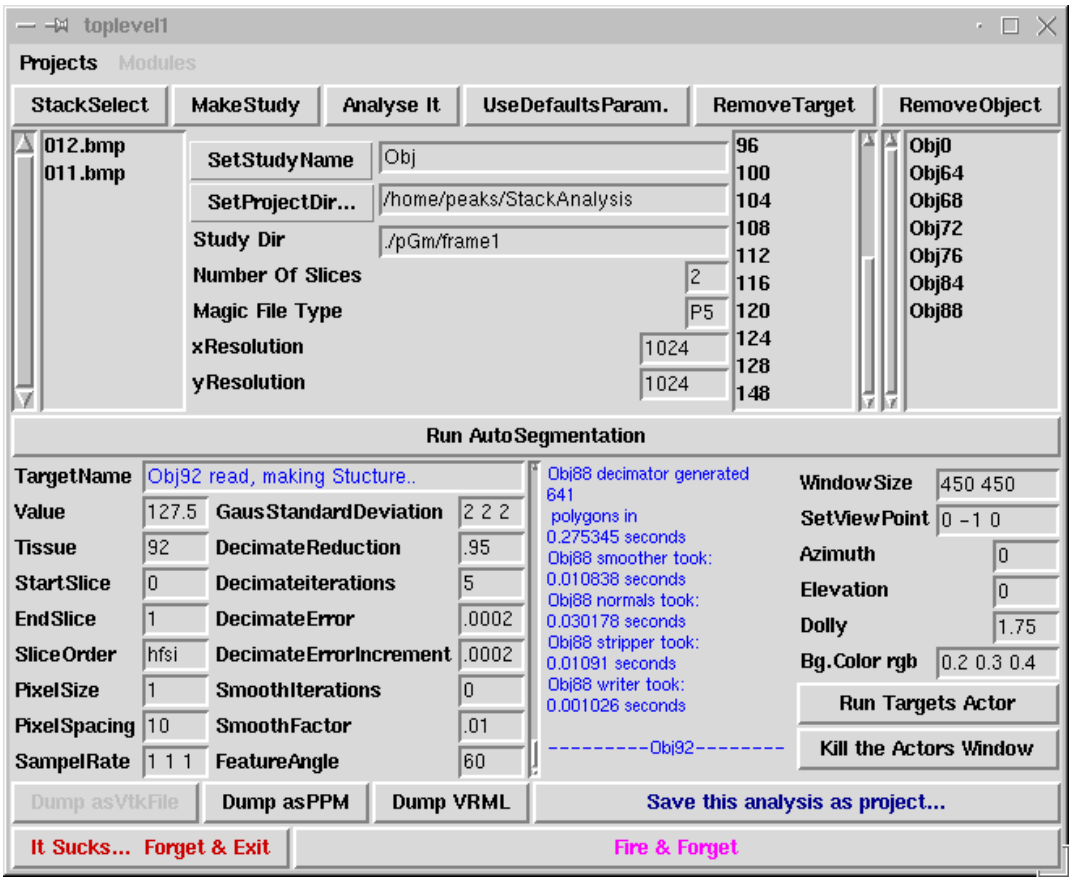


Figure 3.31: *Interface of analysis tool Topologizer (see [KB00]). It was designed to process sliced data of 3D stacks for varying parameter sets.*

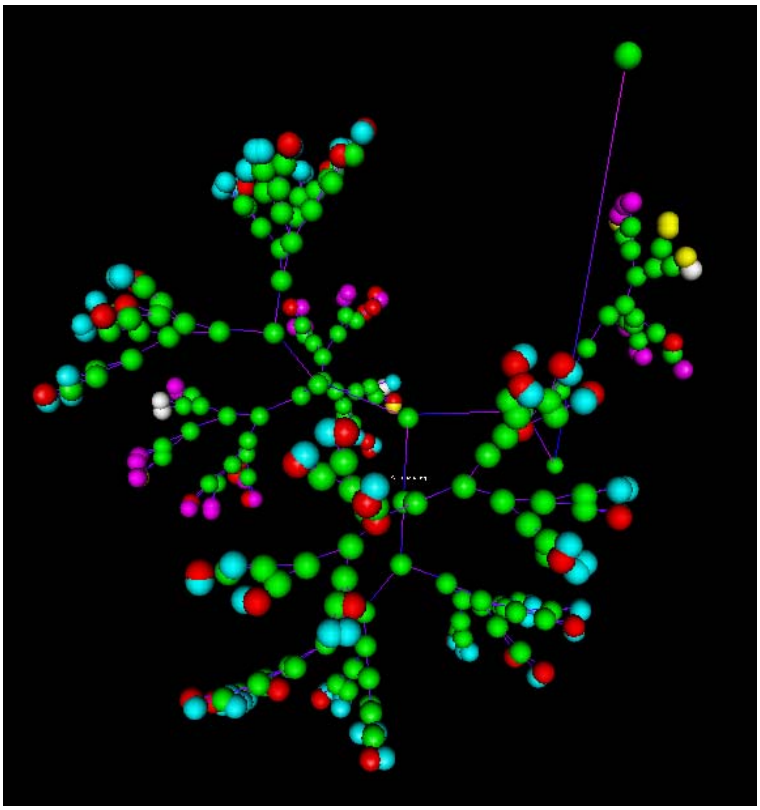


Figure 3.32: *Graphic output of the algorithm of spring embedding applied to the 4D triangulation of figure 3.26.*

Chapter 4

Discussion

4.1 Goal of the work and cell properties

The goal of this work is to study the topological aspects of the development of the shape and its function in embryology. Topology considers the adjacency of objects in space and time. In embryology, the building blocks (objects) are cells, of which topology considers their neighborhood in space and time. The development of shape in embryology is considered as a quite complex process. It is hoped that topological methods deliver a concept to understand, describe and supply models for the process that governs the biological development of shape and consequently of some important functions.

The sequence of steps needed to form an organism is roughly:

> DNA ⇒ mRNA ⇒ Protein ⇒ chemical reaction networks ⇒ cell ⇒ cell proliferation, sorting and determination of functions ⇒ organogenesis and morphogenesis ⇒ organism.

All these steps are interrelated and regulated strongly. To study one specific step or phenomenon, one should consider an object, in which the chosen step is mostly independent of the others, in order to display the phenomenon to be analyzed in a clear manner. A promising model organism for the study of the

step of cell proliferation and positioning in an early phase of embryogenesis is the nematode C. elegans, since it seems that there is little interaction (biochemical communication) between cells and other cell components, e.g. their DNA. This means that the cells themselves (i.e. their interior) act essentially as passive objects in the proliferation and sorting process during the early embryogenesis.

C. elegans is a well studied animal that is easy to cultivate under laboratory conditions with bacteria (E.coli) on agar dishes (see section 2.3.1). Furthermore, it has a short life cycle ($\approx$ 3 days), and the worm, as well as the eggs, are transparent in the study with light microscopy.

A further advantage is the invariance of the wild-type cell lineage from one individual to another and hence of the number of cells in each stage of development. This invariance is particularly important with respect to the intended application of topological methods to the developing structure in the organism (the nematode C. elegans), since topological relations are expected to be at least partially invariant too. The invariance allowed the construction of a unique database for the wild-type animal in which each cell of the fixed number of cells is stored as a record (e.g. position, daughter cell, fate, time of division and others). Note, that the database contains cell-data as entry elements and not DNA sequences or the proteoms of cell types [EMB].

On the other hand, there exist cell lineage mutants which display an altered cell lineage (see [Woo88]). There are closely related nematodes which have an invariant cell lineage too which differs (sometimes only slightly) from the one of the C. elegans wild-type. Whereas this work considers C. elegans only, in the future their phylogenetic relatives should be studied as well.

The figure 1.3 in the introduction displays the life cycle of C. elegans. In the egg up to hatching (558 soma cells in female and 560 cells in male), the major steps of the development take place. In the later larval stages, the development is completed (to a soma that has 811 cells apart from the gonades). The time of development in the egg is divided into halves. The first half is the

pre-morpho-genetic phase, whereas the second half is the morpho-genetic phase of the developing embryo. In the morpho-genetic phase the cell-mass sorted in the pre-morpho-genetic phase take the shape of a worm. The shape is maintained by the cuticula (boundary) of the worm, since experiments damaging the cuticula reshape the interior to a spherical cell-mass [Woo88] and [PH86].

In the first half (pre-morpho-genesis), the major events are cell division (proliferation) and migration (positioning) in such a way that at the entrance to second half (morpho-genesis), nearly all cells have reached their position of destination to initialize morphogenesis and organogenesis. This distinction let the nematodes (and in particular C. elegans) be an excellent object to study the development of topological relations by cell division and cell sorting in the pre-morpho-genetic phase, in which the cell function of the later individuals seem to be negligible with respect to positioning.

A few details about both processes, the division and the movement are discussed and summarized in the following.

4.1.1 Cell cycle events

The cell cycle is considered to be governed by a non-linear oscillation in the cell [HÖ1] and [Gol97]. Since the cell cycle (see figure 4.1) consists of essentially four phases (G_1, M, G_2, S), where the G_1 with the additional G_0 phase and the M phase are of interest in conjuction with the cell division. In the context of the non-linear oscillators, the G_1 and G_2 phases are the switching points (i.e. flip flop mechanism) between the state M (mitosis) and S (DNA replication). Non-linear oscillators have a time interval, in which they may be modulated (either triggered or delayed). In the cell cycle, the G_1 switch is sensitive to modulation, because here the cycle may be delayed (prolonged G_1 phase) by an additional time, the G_0 phase.

Non-linear oscillators may show a jitter, i.e. not every period has exactly the

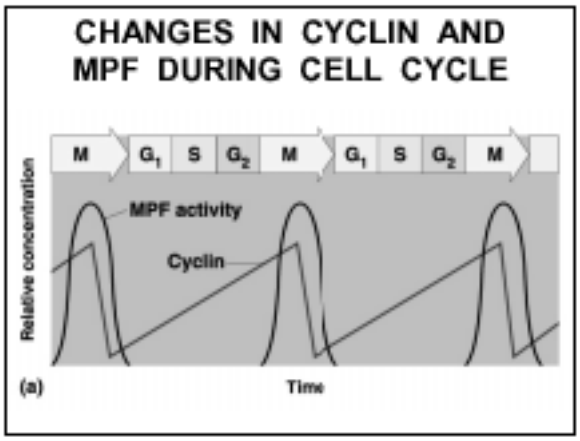


Figure 4.1: *Schematic time course of the concentration of the cell cycle components (a cyclin and mitosis promoting factor (MPF)) displaying the characteristics of a non-linear oscillation. (from http://www.biosci.ohio-state.edu/~pfuerst/courses/mitosis.pdf)*

same duration, but still the oscillation is stable against moderate disturbances. Thus, it is a quasi periodic oscillation. A period may be decomposed into two time sections (G_0, G_1, M, G_2) and (G_2, S, G_0). In the phase M, the cell rounds up and reorganizes the cell interior. In this phase, the cells usually have only a few adhesion contacts to other cells and they usually do not move. In the following phase (G_2, S, G_0), the cell performs its function, keeps essentially its shape and can move. In a movie on the "glia" cells in figure 4.2 (personal communications [Str]), the phases are recognized in a clear manner, since the flat cell detaches from the supporting protein layer, takes a specific shape, divides in a direction parallel to the support surface and finally both cells attach to the support and become flat again.

Therefore, for C. elegans one would expect that the cells will not move, while being in the process of division, but they regain the ability to move, when the division phase is finished. On the other hand, the cell can assume a more or less rounded form in the mitotic M-phase (cell division).

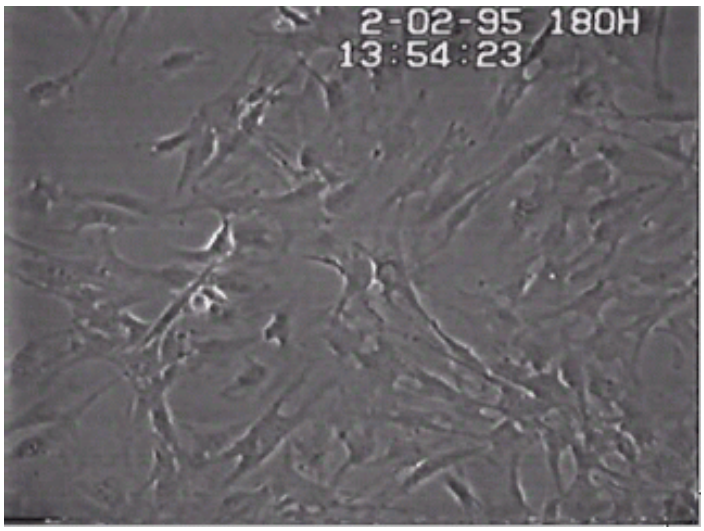


Figure 4.2: *The video freeze image demonstrates a layer of glia-cells, which shows the events during the cell cycle. A cell releases itself from the supporting film (plane) to divide and attaches again to the film later [Str].*

4.1.2 Movement of cells - Migration

Cells move in a developing organism quite often a distance from the place, in which they are born (by division) to their final destination. The movement is active and directed towards the destination, where they implement a specific function. In the post embryogenesis of C. elegans most cells remain at their initial position of division and only a few cells undertake significant movements. In pre-embryogenesis most cells perform substantial movements to reach their final position in the embryo. The forces for the movement are generated while the cell adheres on a support, which may be other cells or guiding material like the eggs shell or fibers. Inside the cell, the filaments are believed to be responsible for the force generation by configurational changes of proteins. The cell apparently sticks to the support by chemical affinity forces through anchors to the support. If such a chemical affinity is unequally distributed over the cells surface, then the sum of forces results in a effective force, and the cell eventually moves in the direction of the effective force.

Inner and outer signals may be responsible for the distribution of the force

directions over the membranes of the cells [Gra93]. The direction may be determined at one time and kept unchanged for a certain time without any check. The external signals are expected usually to be gradients of chemical affinity.

There is a possibility that a cell does not move either because it can not move at all (it rests), or since it has stopped moving because the forces acting on it are in equilibrium (i.e. result in an effective force of zero)

Cells unable to move are prone to serve as a germ for moving cells, to adhere at them and to come to a rest finally. Besides, they may serve as a signal generating source to (attract) other cells.

With C. elegans, it might well be that each cell gets its position in the cell aggregate determined in the cell lineage like many other cell fates in the mosaic development of the nematode.

There are migrations in pre-embryogenesis that have corresponding ones in post-embryogenesis [SH81]. Moreover, mutants have been isolated that show both, different post-embryonic movements and a different patterns in the cell lineage.

It is expected that non-linearity spatial phenomena ([BH73] and [MMYY95]) govern the adhesion and position finding processes also because of their complex and likely non-linear regulation. Analogeous to the cell cycle (as limit cycle) one may assume spatial structures which are quite stable and self regenerating against perturbations.

4.2 The cell-lineage pattern of C. elegans

The cell lineage tree of C. elegans (ancestor tree) is invariant with respect to cell division events and fates (e.g. to be a muscle, a nerve or a skin cell). The tree is usually ordered alphabetically i.e. cells with names in the alphabet prior to others are to the left, except for the founder cells (see figure 1.5). This order needs not to reflect the later position of the cell, which may change during the development.

The relative invariance of the wild type cell lineage places the organism in an extreme position compared to organisms, in which the behaviour of larger cell groups seems to be more important than the fate of single cells.

In case of the development of vertebrates, cell groups undergo a sequence of subdivisions, in which each differentiated group is the base of a subset of tissues.

> Generally the development of living organisms proceeds via a lineage of developmental units (cell groups or types). In the case of the nematode, these units happen for the most part to be directly observable as individual cells.

Since the function of a cell correlates with its position in the organism (i.e. a muscle cell has to find its final place in a muscle) (see [Sch89]), it is likely that the position is determined as a property of the individual cell in the cell lineage tree, also.

This is not the case in other animal species, where the position in the cell lineage is not necessarily coupled to the late function (task) of the cell.

The programmed cell death (see [EH86]) is a unique fate of a cell in the lineage tree. Of the total amount of 1090 produced cells in the lineage tree of C. elegans, 131 cells undergo programmed cell death. The function of cell death in morphological changes is not studied well, but different genes have been identified to be involved in the determination and initiation of programmed cell death [WC02]. There are genes, which induce neighbors to engulf the dying cell. As all other fates, programmed cell death can be initiated by cell-internal or cell-external factors.

4.2.1 Mutants of spatial position and lineage patterns in early embryology

As in all other living matter, genetic mutation may arise spontaneously or can be induced, e.g. by chemicals or radiation, and propagated as well as mixed by

genetic crosses. The methods of molecular biology allow site-specific mutagenesis on the DNA and thereby to allow the straight forward construction and analysis of the effects of mutations by trans-genetic animals. It is expected to obtain at least some experimental answers from mutations that effect the early embryogenesis. For this work, it would be interesting to search for genes that e.g. could test the model suggested in section 4.11.1. Also the many characterized mutants of early embryogenesis and in particular the ones that effect the lineage may help to elucidate the development of the topological structure. Of primary interest are the naturally occurring mutants with an inversion of the left-right bilateral symmetry (invers mutation).

Among the fertile mutants, the expressed phenotype may vary between individuals as well as in their cell lineage patterns. The contrast between these variable lineage patterns of mutant phenotypes and the nearly invariant cell lineage of the wild type from one individual animal to another remains unexplained so far (may be the wild type cell lineage has reached a global minimum in the evolution of the "development- or growth- potential" associated with every biological process of development that is embedded in an environment).

Temperature shift experiments produce information about the time of gene actions and cell-ablation experiments are useful to the study of the spatial location of gene actions in the organism. A review of the fertile mutant phenotypes that have been analyzed so far may be found in the literature (see [Woo88]).

Since this work supports the model that topology may be coded as a "construction plan" in the DNA, mutation experiments are likely to play an essential part in the verification and they are also contemplated to supply further information on the dynamics of a developing topology.

Another interesting attribute is the ability of a cell to transform fates. Missing cells in mutant cell-lineages may be replaced with extra copies of certain types of cells, if the originating type of cell has been lost. One cell may adopt the fate of a particular other cell. This phenomenon is called homeotic. There are spatial,

temporal and sexual transformation of this type.

4.3 Experimental records of the development in the egg.

The basic and fundamental data for these studies are obtained from experimental recordings of the development in the egg. Since the egg is transparent to light, microscopy is the adequate analyzing tool. The contacts between the cells or membranes should be recorded for the intended topological research. This is possible (see section 3.8) but it requires special experimental effort. Therefore the already established method of locating the nuclei by 3D microscopy with time in the egg is chosen (see [Sch78]). It can be implemented without modifying the egg or its interior experimentally by stains.

Both methods (Nomarski and CLSM 2.1.1) yielded pictures, which could be analyzed for the three dimensional position of the nuclei, their division event, and in states with a low number of cells even for the cell (membrane) contacts. However, for any analysis of cell-contacts, electron micrographs are far superior, since the membranes are definitely visible (see [Hal01]). After the cell division, the closure of the membrane could be defined. This might happen quite a long time after the division has finished (see [Sch78]), since it is likely that the unequal daughter cells need to exchange material to become unequal. Nevertheless, the high amount of time-consuming experimental work to determine the "true cell contacts" was postponed for later studies, and the approximation of likely contacts obtained by assuming the nucleus to be centered in the middle of the cell was preferred. Thus, the contacts are calculated by the Delaunay algorithm explained in appendix A (i.e. `qhull`).

The worm was easy to keep in culture on wetted agar plates with bacteria throughout this work and stored in the refrigerator for months. There are several

methods described in the literature (for summary see [Woo88]) to maintain, store and culture C. elegans. The preparation of eggs from these cultures has been adopted as a routine in the current laboratory. Therefore, the movies, which demonstrate the procedure of isolating an egg, are included (see section 2.3.1).

The egg begins to develop quite rapidly, which keeps the time short between the isolation and the preparation for recording, if the intention is to follow the beginning of the development with the two cell state.

4.4 Visualization of the database records

In the preceeding, the production of a database of C. elegans by experimental methods was discussed. The following deals with the extraction of information from the database and its systematic use to explore the principles of embryonal development.

The first step in this direction is to visualize the information stored in the database on a computer monitor. By the visualization, the set of hard to overview records in the database becomes more intelligible. The computer program which transforms the data records into a picture that is easy to grasp, is a browser. A browser has been developed by the company SIMI Inc. especially for the C. elegans database, which serves to visualize the temporal and spatial developmental situations of C. elegans. Besides, the program has a module to analyze the microscopy data and to build a database from them. There are other programs to visualize processes in the development, which intend to show the development in the Internet (see www.cs.umd.edu/users/seanl/gp/ in programming language java) or to display the development with a module that considers cells as balls [GWZM98]. The browser was selected to visualize data in this work, since at the time, it was more flexible to data manipulation, even though it could not be used with the Internet (compare the `Astrodrid project` www.astrogrid.ac.uk/project.htm). Also, modified data may be explored by the browser (see figure 3.22).

4.5 Temporal and spatial abstraction

One method to find rules behind complex events is to simplify the complex processes by abstraction (see [Plaay]). In the developmental sequence of the embryo in the egg, the cell contacts (i.e. the position of a cell relative to other cells) and the proliferation (i.e. cell multiplication by division) are the major events. These processes were characterized by distances, which are for cell divisions the time of division relative to the origin (i.e. the first cell division) [DVB91]. In the 3D space, the Euclidian distance of their two nuclei is taken as the distance between two cells. The consideration of the cell cycle demonstrates that the cycle as a limit cycle may vary in length (i.e. may be quasi-periodic or may have a jitter). The development should not change, if the cell-division times are shifted by small amounts. In section 3.1.2, the set of division times is clustered into subsets of cells with narrow division times. The mean division time may be taken as representative for all times in that subset or cluster. The so modified cell lineage tree may be visualized with the browser (see figure 3.6) and looks still quite normal with the exception that most of the statistical variations in cell divisions have disappeared and that the parameter set, which describes the cell lineage tree, has been reduced drastically (see table 3.1). The same procedure applied to the spatial distances of cell-nuclei (as designated centers of cells) at definite times (see section 3.5) reduces the parameter set still more. The reduction implies that the cells are spheres of almost same diameter at each considered time. The records of the database, in which the mean distance of nuclei and mean division times are inserted, may be displayed in the browser and verified by the dataset on the DVD (`Demo2.sbd` see appendix E.0.4). The result is an organism, which develops in time steps, and which does not always fit into the egg shell, but otherwise approximates the real situation. One may conclude that the principles governing the development are still intact in the simplified model organism at this level of abstraction. Other methods of finding the means of lineage tree parameters may

be applied also. So far, the process of abstraction is nothing more than removing the noise from data.

However, when the simplified model from the view of topology is considered, new aspects arise.

4.6 Structure inherent in the adjacency

The topological structure coded in the vector spaces of chains may be represented in an adjacency matrix, in which the entry "1" means that the column and row objects are adjacent, whereas the entry "0" means that they are not. The objects may be simplices of any dimension or their equivalent counterparts in the co-vector space (i.e. areas, volumes or others), since the adjacency does not imply anything about the objects in mutual contact. The network structure of objects (i.e. simplicial complexes) is described by these matrices and may be displayed geometrically by graphs (see [JMS92]).

An ambiguity is present in the sequence, in which the objects are ordered in the rows and columns (indexing the adjacency matrix). There are several algorithms to reorder the sequence under restrictions or conditions. One specific structure may have different matrices. Thus, different corresponding dot plots arise, when the reordering criteria are changed. A normalized form would eliminate the ambiguity and could identify matrices coding the same structure.

An incidence matrix could be found for each adjacency matrix associated with a simplicial complex made of simplices of the same order (see section 3.2). For incidence matrices, an ordering algorithm has been reported (see [BBT01]) that assigns to each of them a unique normalized form and a unique dot plot representation (see figure 3.18), except for the automorphism.

Another way to code the structure in the adjacency matrices is the adjacency of simplicial stripes (i.e. simplicial complexes see appendix B). Among the visualization tools, triangle stripes are well known, and algorithms to operate on

them are implemented. A structure is thereby characterized by a set of stripes (usually written as a sequence of numbers, letters or as a combination of both). Such a sequence may be interpreted as a word of a language, in which the letters, it is composed of, belong to an alphabet. The set of words is a text, in which the structure inherent in the development is written. The text is usually not composed arbitrarily of letters but underlies (meets) a grammar (i.e. equations for text see [HRL75] and [Ber79]). In computer science, such grammars for text are called "a programming-language" and are used to write computer programs. It would be challenging, if the biological development could be written equivalently as a program in a "development-language". At least it is apparent, that probably on the DNA the plan of development (i.e. the development program) is deposited in a way that it creates structures in an appropriate environment.

A further method of coding the structure information is string embedding (see section 3.10 and [SSL00]). Here, it is attempted to display the structure in a space of a lower dimension than the original geometric space. A one dimensional arrangement is a sequence, like a text.

Moreover, the adjacency may be considered from the point of view of vector spaces and their boundary operators (see appendix B). The matrix for object boundaries characterizes the adjacency in the network of objects. In addition to the adjacency vector spaces introduce an orientation. Therefore, a common boundary is represented in the matrix by either -1 or $+1$ entries, depending on which side of the orientation the object is situated. The matrices for boundary operators are "matroid's" (matrices having only $+1$, 0 and -1 entries), which meet the boundary condition (see appendix B)

4.7 Reconstruction from a distance matrix

The reconstruction of the basic geometric shape is a central aspect of the elucidation of the principles behind the development. In this work, the starting point

of reconstruction is the distance matrix between the nuclei (assumed to be the centers of the biological cell). Only the distances between the cell-nuclei, which are neighbors, are needed for the reconstruction. A 1-chain summarizes just these distances from the distance matrix 3-3.5. The distance matrix implies that there is a certain number of objects (the number of rows, respectively columns), for which an algebraic metric (see [Bau92]) or distance measure exists. The matrix elements are the distances between different objects. A coordinate system has to be established to give the distance a geometric meaning (see section 3.4 for details how this is done). This coordinate system is relative with respect to the objects (the nuclei) forming the distance matrix. For an observer of the egg, this relative coordinate system has to be translated and rotated depending on the movement of the egg in space to yield the real positions of the nuclei in a global coordinate system. In this internal or relative coordinate system, the spatial position of each nuclei could be calculated, because the coordinates have to satisfy the distances given by the distance measure (matrix) (see 3-4.7). Since the reconstruction method explained before is not limited to the 3D spatial geometry, it may be extended in the same way to the 4D time-space geometry, as described before (see section 3.6). Thus, the development of the geometrical shape (see [FE96]) of the organism may be reduced in a 4D distance matrix or an equivalent 1-chain and displayed geometrically in a corresponding internal coordinate system.

So far, reality has been visualized with the reconstruction (see [Fer01], [GWZM98], [KHKL98] and [Hea01]). The algebraic concept of chains (see appendix B) supplies the distance data with the algebraic construct of vector spaces. The distance data are split into the vector space base (σ in appendix B) and the coefficients in the chain for each base vector (i.e. the distances between neighboring cells). The structure of the neighborhood of the cells is incorporated in the vector base (in relations of a topology) and the metrical, or geometric part, in the coefficients. In the coefficients, the real geometrical shape is coded, and in the structure (vector base), the gross outline prescribed (e.g. a road, a disk, a

sphere, etc.)

By setting all coefficients of the chain to one, an abstraction to a prototype object is obtained, which comprises the structure generated with the chain base (see section 3.11 for the 8-cell state). It should be noted that not every prototype could be reconstructed with the Euclidian metric. The vector space concept of chains in different dimensions implies a vector space homology (cascade) (see [dR69] and appendix B), in which the coefficients could not only be interpreted as distances, but also as areas, volumes, etc.

4.8 Development in 4D time-space coordinates

In a 4D space, a cell fills a volume, which is given by the 3D cell-volume at all times of its life (i.e. from birth to death). The position in 4D space may be the mean of the nuclear positions in its time and space coordinates. In 4D space, the complete development of the embryo is represented at once. As in the previous 3D case, the distances may be defined between neighboring cell positions. Here, either the real or the simplified (i.e. abstracted) database records may be used. Because time and space are measured in different dimensions (i.e. [min] and [μm]), the units must be adapted. The time axis has to be scaled to spatial extension. There are many ways of scaling the time axis. Among those tried, the one that approximates the mean of the numbers of the space distance and time distance gave a reasonable topological decomposition. A logarithmic time scale gave a closely similar decomposition. This demonstrates the stability against perturbations or disturbances.

Also by setting the distances of neighboring cells as equal as possible, an abstraction may be obtained in the 4D space and a topology in the 4D time-space coordinates of embryology results.

A remarkable property in this 4D space is, that it incorporates the cell lineage tree. The distance of the mother cell to its daughter cell declares the daughter

cells as neighbors to the mother. Hence, the lineage tree connects all cells in the 4D representation and is called a spanning tree. A spanning tree is a path over the cell neighbors to every cell in the cell aggregate. In a graph representation of a spanning tree, the cells are nodes and the cell neighborship is displayed as branches connecting the neighbors. In that graph, the spanning tree (cell lineage) is made of branches. All branches, which are not belonging to the spanning tree are called the "tree complement" (so to speak its soma). The tree complement is an important set in the network theory (see [DK69]) and may be an essential component, when the forces and their fluxes that drive the development are to be established.

4D structures are difficult to imagine and to visualize. Nevertheless, the visualization of such 4D scenes is attempted by the group at the geometry center (www.geomview.org). The evolving topology in 4D may disclose a set of abstract rules which govern the growth in the developing egg.

Cell division proceeds along an axis in 3D-space, on which both nuclei move apart to form the daughter cells, and which marks their common plane (membrane). This process distributes (or passes on) the cell contacts from the parent to the daughter cells. This is an operation that is applied in the decimation programs of graphic visualization of objects (see [SML98]), usually in the revers order. In the 4D-space, the cell division is recovered along the time axis and may restrict the topological time direction.

4.9 Developmental potentials, forces and fluxes

Whereas the structure (base of chains) inherent in the network of cells was considered in the last section, now the coefficients of the chain vectors will be. The vector space view of the structure may be used advantageously as a vector base for defining the physical forces and their currents (i.e. displacements or fluxes) that the forces generate. Usually, currents and forces are coupled by physical

laws. Some more work and additional analysis of the processes in developing organisms is necessary to define the forces that drive the development and to find the laws that convert forces into currents or fluxes (i.e. real movements).

The vector spaces may be used as a space to formulate the forces, which stabilize cells, move cells and contribute to the morphological appearance of the organism. Here, only the pre-morpho-genetic events are considered, but the "force–motion" view will probably be of major interest, when the connected cell mass form a worm-like entity. Nevertheless, the vector space view may guide the way as a base, in which the development is formulated. Particularly the 4D space of the development may be useful, since here the total development is a static topological structure and as such, the fixed vector space describes the structure of the development.

In the appendix B, two essential properties of these vector spaces have been pointed out. Firstly, the property that the forces in the network may be derived from a potential (see electrical potentials in a circuit [DK69] and appendix B) and secondly, that the laws of "Telegen's Theorem" apply. The later states relationships between the vector space and its co-space. In thermodynamics this theorem is related to irreversibility (see [Ons31]).

A possiblity to introduce forces into a flexible cell mass is given by [Gra93] and [GS93]. In their work differences in adhesion forces may separate the cell mass into those forming a cell surface and those that prefer to stay in the interior of the mass. In deed, looking at the dynamics of the movie in figure 3.28 one may recognize that there are cells that try to stay in the interior or move to the surface of the cell mass.

4.10 Algorithm, Data Structures and Convex Hulls

A part of the current work is based on computer algorithms, in particular on the algorithm dealing with convex hulls (see [BDH96]). This is because an attempt is made to reduce the generation of biological shape to a set of geometrical objects, which could be formulated by a coordinate system, into corresponding algebraic sets of equations. The algebraic equations are accessible to computer algorithms. The result of the algorithm could be retranslated into biological reality, and used to explain and describe its processes and phenomena.

Mathematically, the convex hull of a set of points is the smallest convex set, which comprises all points of the set (i.e. hull or boundary of the set). The calculation of the convex hulls is an essential problem of the field of " computational geometry". Algorithms for the calculation of the convex hull of a set of geometrical objects like points are available (e.g. formulated in the computer language C++ [PS88]).

In this work, the algorithm `qhull` [BDH96] is used. It calculates beside the n-dimensional convex hull of geometrical objects (set of points), also its Delaunay triangulation and Voronoi decomposition. The details of the algorithm and the tools, on which it is based, may be obtained from the literature [PS88]. Some general aspects of geometry are mentioned in the following, since geometry seems to be fundamental to the study of biological shape.

4.10.1 Computational Geometry

In early times, geometry (the science of taxing land) was formulated on the proof of the construction of geometric objects (like triangles or circles) by tools (like rulers or compasses etc.). Later, it was discovered that geometric objects could be represented in a coordinate system and treated by algebraic tools (like equations,

operations, proofs, etc.). Then, based on analysis, the metric geometry (of which a distance measure is a special case) of the convex sets was introduced. The theory of convex sets characterizes the global properties of geometric objects and handles extreme value problems. The distance between mathematical functions leads to the power-full concept of the functional space.

Combinatorial geometry is another concept, which fits well to the spirit of algorithmic geometry. Here, geometric objects are characterized by properties of their sub-sets. Often, the runtime of the algorithms becomes unlimited since the scene of objects is too complex and may consist of an unlimited number of sub-sets needed to be characterized.

Algorithms of combinatorial geometry are contained in software for graph-editing and pattern recognition. The advancements in computational geometry (see [MP69]) consists in the development of efficient algorithms to solve tasks in visualization, in linear programming, in spanning and in segmenting trees. Particularly, run-times for an algorithm may be estimated (e.g. upper and lower limits of runtime), since an algorithm may be decomposed into discrete constructions steps for geometric tasks (or objects). The runtime results from the total sum of steps carried out in the algorithm. The constructive aspects are no longer ruler and compass, but instead tools of algorithms. In this view, the development of shape may be seen as the result of an algorithm (program) to create geometric objects.

4.10.2 Algorithms, data structures and access functions

Two relevant components constitute a language that describes objects of computational geometry, firstly the algorithm and secondly the data structure. An algorithm is formulated in a programming language to be executable on a physical computing machine. The data structures contain the information (data) to be operated on in an ordered set or sequence. Languages need an operating sys-

tem on the computer to be executed. Most of the programming environments run on the widespread operating systems like UNIX, WINDOWS and MacOS. In this work, most of the computing is done in LINUX (a UNIX based public platform obtained from SUSE (www.suse.de)) and some under WINDOWS 2000 Server (www.Microsoft.com). The programming environments are for C++ (Visual Studio 6), tcl/tk (VisualGipsy www.prs.de), VTK (www.kitware.com) and MATLAB (www.mathworks.com). The programs written for this work are gathered on the DVD storage included in this work. The major high level language used to program algorithm was MATLAB and C++. MATLAB programs are basically of the interpreter type, i.e. the program is first interpreted by an interpreter program and afterwards executed. The advantage is a high flexibility in the development of programs with the tradeoff that the interpretation increases the computational load. The MATLAB program code can be translated into the language "C" or "C++" and compiled to executable binaries on different platforms (Windows or Linux). These binaries of MATLAB programs may run approximately a hundred times faster than MATLAB programs interpreter code does.

To facilitate programming, the data structures available in a language are important.

A comprehensive introduction into data structures that are specially suited for complex combinatorial operations may be found in [AU77]. The fundamental data organization are sets, sequences (ordered sets), tree structures and lists, which may be edited and accessed by the operations:

> MEMBER, INSERT, DELETE, FIND, UNION, MIN, SPLIT and CONCATENATE

MATLAB is a language for engineers, which is based on matrix data structures and their manipulations. It complements the linear algebraic methods to the above data structures.

The large amount of data on the developing nematode is packed into a database (see appendix E.0.3), which are accessed and operated by different programs (SIMI browser,Matlab,VTK etc.). Interconversion programs were written to transform (see section 2.5.5) the database records into a language internal data format to use the individual properties of the different languages effectively. This essentially reduce the programming time needed to obtain a program that was free of errors. The availability of geometric algorithm and the appropriate data structures may be of help in the future to formulate for biological objects the adequate biological operation of cell division, cell movement and programmed cell death, in extended data structures (see table 2.7 and [ELZ00]), as more data of the development are feed into the database. For details concerning the calculation of the runtime and the computational load of algorithms see [PS88].

4.11 Outlook

Besides the geometric view, in which the topology within cell aggregates is visualized in the form of simplices and Voronoi territories, there is an algebraic representation. Here, basic elements of a vector space (e.g. simplices) could be thought of as functions of functionals. This brings up a new aspect, since a vector space having functions as a base e.g. a Hilbert space [Hil99], is well known in fields of physics. At present, a special type of Hilbert space (by Wavelet functions) is studied in several respects e.g. in compression of huge data sets as digitized speech and movie records. Looking in the fundamental properties in the theory of wavelets [Dau92], especially with the package decomposition, one discovers striking similarities to the nature in biology. There are two operators (evolving translation and dilatation), which in many ways resemble the process of cell division and cell movement in the context of the adjacency of the objects (see the J rotation operator and the T translation operator [Kai92]). Moreover, the property of the "Uncertainty Principle" in such spaces may lead to another

interpretation in biology e.g. the duality of a signal and its spectrum may find an equivalent in the germ line (with the DNA sequence as a spectrum) and the time-limited appearance of the organism (soma as a signal). However, this aspect needs further consideration to be derived rigorously.

Another aspect of vector spaces are transformations (i.e. maps), e.g. movements of objects in space, which may leave properties as the volume invariant [Kle72] and may be performed on homogenous coordinates.

4.11.1 Relation to cells of the immune system

A biological system displaying many of the properties necessary to realize a mechanism that is able to build and shape a biological object according to a rule or a "construction-plan", is the immune system.

In the phylogenetic development of organisms the immune system appears late and only in vertebrate animals. A tissue system showing a high complexity of connectivity is the nervous system [HBBW$^+$00]. Since the analysis of the genome, it has become obvious that there are several common features in the immune and the nervous system expressed e.g. in a super-family of genes [WM99]. The common features include proteins in the membrane, which are designed and function in a similar manner in the two systems.

Both, the nerve cell and the immune cell can move to find their target cell. They may form dendrites. Both systems have a high competence in recognizing other cells or cell parts. This suggest that biochemical, biophysical and genetic mechanisms and components involved in the later recognition events may organize similarly developmental events, since nature often modifies biological systems to fit them to new tasks.

Here, like in the immune system, leucocyte cells have to move to a specified site and make a contact. In the immune system, small peptides (e.g. obtained by digesting proteins) are signals to locate targets. If one assumes that peptides

(eventually acting like a hapten) are morpho-genetic guides in the embryonic cell sorting process, it can be explained, how the cell can move in a predefined manner and direction.

Moreover, a genetic coding of the morpho-genetic process would be useful (as a memory of the construction plan). Cell may adhere by adhesion molecules, which for some cell types are characterized as proteins with sugar moieties. If a subpopulation of cells adhere by one tissue specific type of adhesion molecule, then the subpopulation may form a tissue (as e.g. a musle) by adhesion and staying together while moving. They create a hull, because the aggregate has a surface which may take a special shape, if forces interact between the cells. However, the position of a cell in such an aggregate of cells is likely not to be locally fixed. A few different adhesion molecules are involved in the contact processes. They may subdivide the cell aggregate further into sub-regions of different size [WMSS01] and [SLJ$^+$93]. The genetic properties coupled to this subdivision are jet only intuitive regarding the formation of organs (as e.g. muscles).

How a coded adhesion or connecting surface protein may be designed from a gene section in the genome is depicted with immunoglobulines [WM99]. The many recognition domains of antibodies on other proteins in the immun system is coded on a few genes regions only [Pau93]. This is a striking example to code a connectivity with genes.

Chapter 5

Appendixes

Appendix A

Simplex computation

In the following, notions are introduced for a rigoreous treatment (see eg. literuture on topology [Mun84]).

A.0.1 Simplices and simplicial complexes

From the geometrical point of view a simplex is understood intuitively: A simplex is a geometric object (volume), which has the number of points that constitute it as edge-nodes (see figure A.1).

A d-simplex is a $(d + 1)$-hedron spanned by $(d + 1)$ points in a space of dimension $\geq d$. Each point is connected by a line (edge) with every other point in the simplex. Any d points of the $(d+1)$ points form a $(d-1)$-simplex (facet). By induction, any $(d - 1)$-points of the $(d - 1)$-simplex form a $(d - 2)$-simplex (sub-facet) etc. (see figure A.1)

a 0-simplex is a point (called a vertex also).

a 1-simplex is a line segment (or edge) between 2 points in space

a 2-simplex is a triangle spanned by 3 points

a 3-simplex is a tetrahedron spanned by 4 points

a 4-simplex is a pentahedron spanned by 5 points.

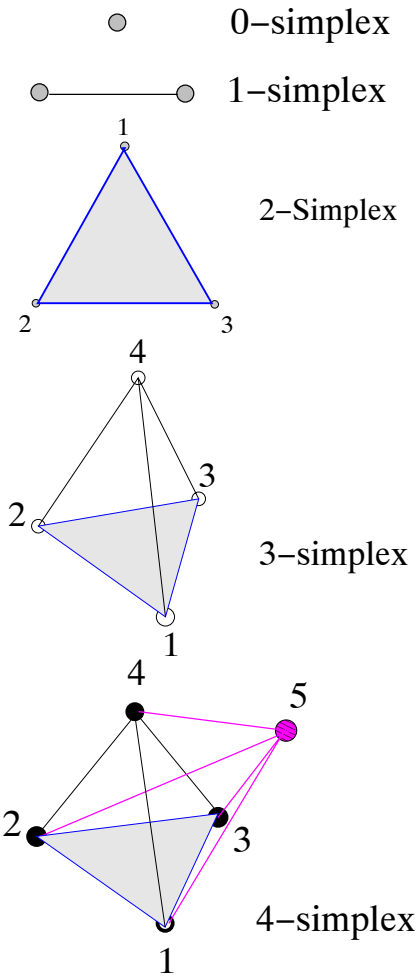


Figure A.1: *The geometric representation of simplices of dimension 0 to 4. The point "5" (color magenta) in the 4-simplex represents the time coordinate for the points 1 to 4.*

The number of d-simplices in a simplex of m points is obtained from the binominal coefficients (see table A.1).

Binomdistribution											d	Number of simplices
					1						0	
				1		1					0	1 0-s.
			1		2		1				1	1 1-s. 2 0-s.
		1		3		3		1			2	1 2-s. 3 1-s. 3 0-s.
	1		4		6		4		1		3	1 3-s. 4 2-s. 4 0-s.
1		5		10		10		5		1	4	1 4-s. 5 3-s. 10 2-s. 10 1-s. 5 0-s.

Table A.1: *Simplices (e.g. facets) that constitute a d-simplex (polyhedron). In the right half of the table e.g. the entry "4 2-s" means: "four×2-simplices" (see figure A.1 for simplices of different dimensions "d").*

A d-simplex may be defined by its $d+1$ points $(P_1, P_2, \ldots, P_{d+1})$, of which it is made and denoted by brackets $[P_1, P_2, \ldots, P_{d+1}]$, e.g. a triangle with edge-nodes named $1, 2$ and 3 is written by this notation $[1, 2, 3]$.

A simplex is regular, if all edges have the same length. A point (edge-node) of the simplex is named a vertex. The geometric view of simplices may be alternatively represented algebraically by inequalities. For example a triangle is given by three equations (see figure A.2).

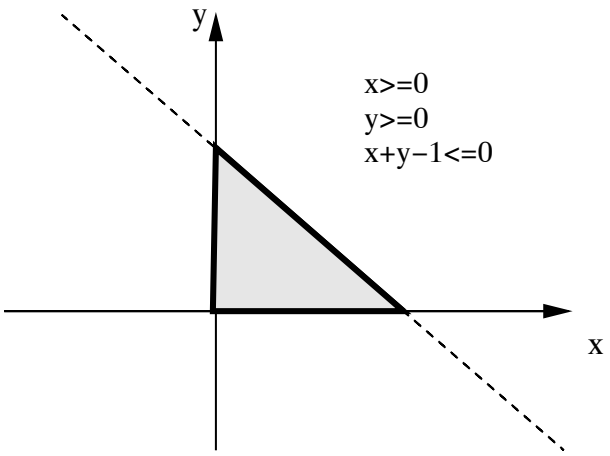


Figure A.2: *Algebraic representation of a simplex (triangle) as half-spaces.*

In the plane (x,y) the triangle is defined by:

- the inequality $x \geq 0$ which has the half space above the x-axis, and
- the inequality $y \geq 0$ which has the half space to the right of the y-axis as solution, and
- the inequality $x + y - 1 \geq 0$ has all points to the left of the line $x + y = 1$ as solution.

Any point (x, y) that satisfied the equations above is a point belonging to the triangle defined by the equations, i.e. the inequalities have as solution all points within the triangle limited by the x-axis ($x = 0$),the y-axis ($y = 0$) and the line ($x + y = 1$). The equal sign creates a system of equations from the inequalities, where a subset of the system of equations has as solutions the points and the lines respectively, of which the triangle (simplex) is composed. In a similar way, inequalities may be written for other simplices. The method of an algebraic definition of simplices by inequalities are their half space representation, named so since an inequality has a half space as a solution.

Simplices may be embedded in a space of higher dimension than their own interior space i.e. a triangle as a 2-dimensional (plane) entity may be embedded in the 3-dimensional space.

A simplicial complex is a set of simplices, which are connected to each other as shown with the triangles in figure A.3.

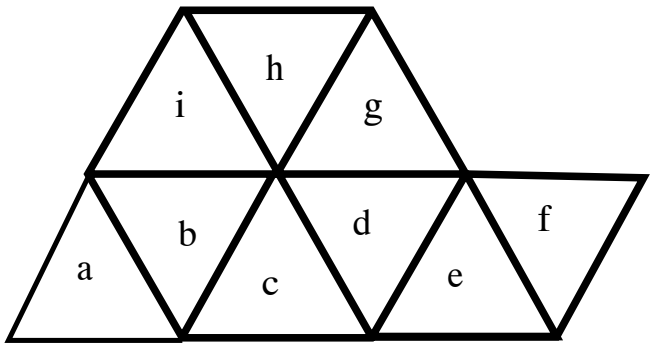


Figure A.3: *Simplicial complex of the triangles (2-simplices) a to i.*

For a set of points (or vertices), such a triangulation can be calculated by an

algorithm, which is based on the minimal distance between vertices (see Delaunay triangulation [Del34]). By the Delaunay method a simplicial complex of any dimensional simplices may be generated. Computer algorithms are implemented (see [BDH96] and [SML98]) to create a Delaunay triangulation (see [Wat81] and [Moe89]) from a set of points. Such simplicial complexes constitute vector spaces (see appendix B).

A.0.2 Delaunay triangulation of points and volume of simplices

A Delaunay triangulation is a connected set of points (nodes) such that the convex space filled by the points is decomposed into simplices. The volume of a simplex that is spanned by its nodes in spatial space is computed by their point coordinates.

The volume for

- a straight line connecting the two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ in the 2-dimensional space is the length given by the determinant

$$V_1 = \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} \tag{1-0.1}$$

  or if $P_1(x_1)$ and $P_2(x_2)$ are placed in a one dimensional space

$$V_1 = \begin{vmatrix} x_1 & 1 \\ x_2 & 1 \end{vmatrix} \tag{1-0.2}$$

  .

- a triangle formed by the points $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ and $P_3(x_3, y_3)$ in the plane the area is given by the determinant

$$V_2 = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \tag{1-0.3}$$

- a tetrahedron in the 3-dimensional space having as edge-nodes the points $P_1(x_1, y_1, z_1) \ldots P_4(x_4, y_4, z_4)$ is given by the determinant

$$V_3 = \begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix} \tag{1-0.4}$$

- a pentahedron spanned by five points in a four dimensional space the volume is

$$V_4 = \begin{vmatrix} x_1 & y_1 & z_1 & w_1 & 1 \\ x_2 & y_2 & z_2 & w_1 & 1 \\ x_3 & y_3 & z_3 & w_3 & 1 \\ x_4 & y_4 & z_4 & w_4 & 1 \\ x_5 & y_5 & z_5 & w_5 & 1 \end{vmatrix} \tag{1-0.5}$$

  and so on for d-hedrons ((d-1)-simplices) in (d-1)-dimensional spaces. In our case, only four dimensions (space and time) are of interest.

The above calculation is not applicable to simplices embedded in higher dimensions in this simple form e.g. for triangles in the 3D space.

An alternative way to compute the volume is by the length of the edges of the simplex. The length s of an edge (line) in the n-dimensional space, which is bounded by the points $P_1(x_1, \ldots, x_n)$ and $P_2(y_1, \ldots, y_n)$, is computable using the Euclidean metric

$$s^2 = (x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2 = (P_1 - P_2)^2. \tag{1-0.6}$$

- For triangles defined by the three edge-points (vectors in space) $P_1(x_1, \ldots, x_n)$, $P_2(x_1, \ldots, x_n)$ and $P_3(x_1, \ldots, x_n)$ the length of the three sides are s_{12}, s_{23} and s_{31} (s is indexed with the indices of points between which the distance (length) is measured.)

  The square length of three triangle sides

$$\begin{aligned} s_{12}^2 &= (P_1 - P_2)^2 \\ s_{13}^2 &= (P_1 - P_3)^2 \\ s_{23}^2 &= (P_2 - P_3)^2 \end{aligned}$$

may be written in generalized form also as

$$V_{ij}^2 = s_{ij}^2 = \frac{1}{2} \begin{vmatrix} 0 & (P_j - P_i)^2 & 1 \\ (P_i - P_j)^2 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix} \tag{1-0.7}$$

where (V_{ij}^2 is the generalized volume of a line-segment between the end-nodes P_i and P_j)

In the above form the area $V_{P_1P_2P_3}$ of the triangle is systematically computed from the length of the three line-edges s_{12}, s_{13} and s_{23} using the symmetrical determinant, e.g. for the triangle constituted by the points P_1,P_2 and P_3

$$V_{123}^2 = \frac{1}{16} \begin{vmatrix} 0 & s_{12}^2 & s_{13}^2 & 1 \\ s_{12}^2 & 0 & s_{23}^2 & 1 \\ s_{13}^2 & s_{23}^2 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{vmatrix} \tag{1-0.8}$$

- For the tetrahedron with the defining edges-nodes $P_1 \ldots P_4$, the volume results from the length of the six line-edges $s_{12} \ldots s_{34}$ of the tetrahedron

$$V_{1234}^2 = \frac{1}{288} \begin{vmatrix} 0 & s_{12}^2 & s_{13}^2 & s_{14}^2 & 1 \\ s_{12}^2 & 0 & s_{23}^2 & s_{24}^2 & 1 \\ s_{13}^2 & s_{23}^2 & 0 & s_{34}^2 & 1 \\ s_{14}^2 & s_{24}^2 & s_{34}^2 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{vmatrix} \tag{1-0.9}$$

- For a pentahedron the volume results from its 10 line-edges

$$V_{12345}^2 = \frac{1}{9216} \begin{vmatrix} 0 & s_{12}^2 & s_{13}^2 & s_{14}^2 & s_{15}^2 & 1 \\ s_{12}^2 & 0 & s_{23}^2 & s_{24}^2 & s_{25}^2 & 1 \\ s_{13}^2 & s_{23}^2 & 0 & s_{34}^2 & s_{35}^2 & 1 \\ s_{14}^2 & s_{24}^2 & s_{34}^2 & 0 & s_{45}^2 & 1 \\ s_{15}^2 & s_{25}^2 & s_{35}^2 & s_{45}^2 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{vmatrix} \tag{1-0.10}$$

- and so on for higher dimensions.

The various volumes of the simplex (n-hedrons) components (e.g. their facets) are computable in this comfortable way (e.g. area, volume, total volume etc).

The elements s_{ij} in the above matrix could be obtained from the distance matrix d (metric) of points (objects) as the squares of its entries.

$$\mathrm{d} = \begin{array}{c|ccccc} & P_1 & P_2 & P_3 & \cdots & P_n \\ \hline P_1 & 0 & s_{12} & s_{13} & \cdots & s_{1n} \\ P_2 & s_{21} & 0 & s_{23} & \cdots & s_{2n} \\ P_3 & s_{31} & s_{23} & 0 & \cdots & s_{3n} \\ \cdots & & & & \cdots & \\ P_n & s_{n1} & s_{n2} & s_{n3} & \cdots & s_{nn} \end{array} \qquad \text{(1-0.11)}$$

A.0.3 Computation of areas in Voronoi tessellation

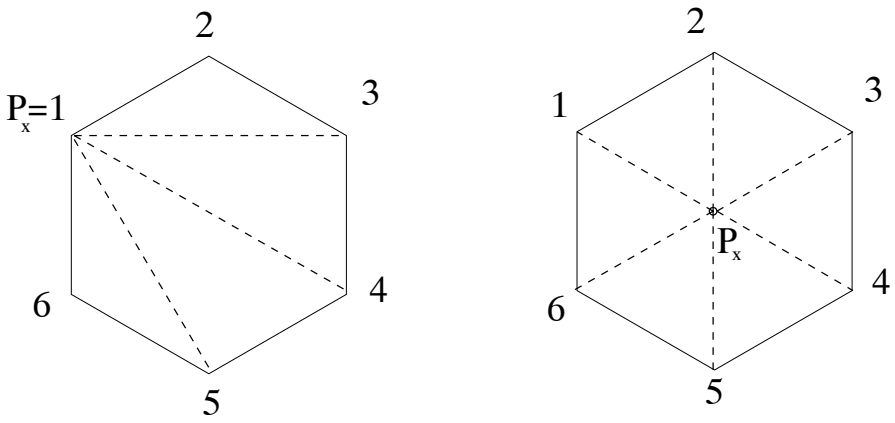


Figure A.4: *Triangulation of the Voronoi territory for Volume calculation.*

The volumes of the territories of a Voronoi tessellation may be computed with the aid of simplices. Given the extreme points of a flat Voronoi territory in the plane $[P_1, P_2 \ldots P_m]$, where the points may be ordered, such that the resulting polygon is cycled in mathematically positive sense (Fig. A.4), then the area of the polygons may be computed by segmenting the Voronoi territory in triangles (see dotted lines in fig. A.4) which have one common point (e.g. P_x) either at the boundary (figure A.4 left panel) or eg. P_x inside the territory (figure A.4 right panel). The area F of the territory is the sum of the triangle areas (in case of figure A.4) of $[P_1P_2P_3] + [P_1P_3P_4] + [P_1P_4P_5] + [P_1P_5P_6]$. The triangle areas are

computable with equation 1-0.8 for Delaunay triangles, i.e.

$$F_{Polygon} = V^{1/2}_{P_1P_2P_3} + V^{1/2}_{P_1P_3P_4} + V^{1/2}_{P_1P_4P_5} + V^{1/2}_{P_1P_5P_6} \tag{1-0.12}$$

where the edge length can be read of the symmetric distance matrix of vertices in the Voronoi tessellation (as e.g. of matrix 1-0.11).

The points of this distance matrix are the vertices of the Voronoi diagram. Remarkable is that no longer every vertex is connected with all other vertices as in the case of simplices (i.e. the distance matrix may be sparse for a Voronoi territory).

The volume of Voronoi territories in 3D space may be computed analogously, if one extra point, P_x, inside the territory is chosen. A Delaunay triangulation into tetrahedra of the Voronoi territory and the inner point P_x is computed. The sum of all tetrahedron volumes calculated by the equation 1-0.9 is the volume of the spatial Voronoi territory. If the reference point P_x is a point of the framing polyhedron (hull), then the volume may be calculated from tetrahedra of the Delaunay triangulation with the points of the hull (the border of the 3D-territory). The calculation of volumes in the 4D space or to higher dimensional spaces may be performed analogously.

Also, the volume V of the bounding surfaces of a territory is computable with the above operation. For more detail, see the literature (e.g. [GR97] and E.0.4).

A.0.4 Volume computation of territories in the hull of a Voronoi tessellation

The surface territories (tile of the boundary) of a Voronoi tessellation has naturally no limit (see ground node in figure B.1). To create a limit, one vertex is introduced, which is positioned at infinite distance. This infinitely displaced vertex is named “the ground vertex” and connects to lines which lead into infinity. The network of a Voronoi tessellation projected onto a sphere, where e.g.

the north-pole is the ∞–vertex at infinity, may illustrate the connectivity in the network.

Since the considered objects are cells (e.g. finite spatial structures), the territories may be limited artificially, e.g. by introducing a limited cell volume or by a distance measure (i.e. like a radius of a cell). In the concrete case, a fixed volume is assigned to the surface cells for simplicity.

Appendix B

Triangulation, tessellation and vector-chains

B.0.1 Delaunay triangulation and Voronoi tessellation as chains

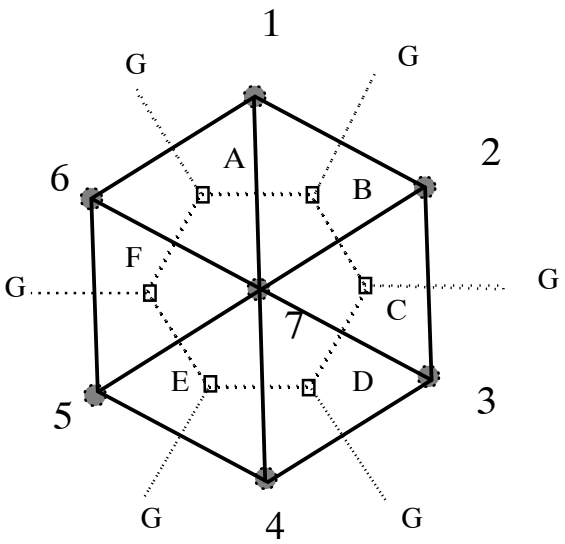


Figure B.1: *Delaunay triangulation for the seven vertices* $1, 2, \ldots, 7$ *connected by the solid lines and the coresponding dual Voronoi tessellation drawn in dashed lines. The dashed lines connect the vertices* A *to* F *and* G *(the ground-node at infinity* ∞*).*

To understand the relationship between the geometry of the structure and vector space chains, the following may be verified before one proceeds to work with the more difficult percepted 4D structures.

The Delaunay triangulation in figure B.1 (drawing with solid lines) consists of

7 **vertices** (also named points or edge-nodes) the seven 0-simplices
$[1], [2], [3], [4], [5], [6], [7]$.

12 **branches** (also named lines or edges), the twelve 1-simplices
$[1,2], [2,3], [3,4], [4,5], [5,6], [6,1], [1,7], [2,7], [3,7], [4,7], [5,7], [6,7]$.

6 **triangles**, the six 2-simplices
$[1,2,7], [2,3,7], [3,4,7], [4,5,7], [5,6,7], [6,1,7]$.

In figure B.1 the Voronoi tessellation (dashed lines), the dual to the Delaunay triangulation (solid lines), consist of the

seven 2D-Voronoi territory **areas** (the 2-co-chains)
$[A,B,C,D,E,F]$ $[A,B,G]$ $[B,C,G]$ $[C,D,G]$ $[D,E,G]$ $[E,F,G]$ $[F,A,G]$.

twelve dashed **lines** (branches) (the 1-co-chains)
$[A,B]$ $[B,C]$ $[C,D]$,...,$[F,G]$ and

six **vertices** (the 0-co-chains)
$[A][B][C][D][E][F]$ and $[G]$

For each triangle exists one vertex in figure B.1, which are the vertices A to F within each triangle. Each solid line of the Delaunay triangulation is associated with exactly one dashed line in the Voronoi tessellation (if dashed lines to the ground vertex G are included). Each Delaunay vertex is enclosed by one territory limited by Voronoi edges. There is a one to one correspondence for

- Delaunay vertices to Voronoi territories (areas)
- Delaunay lines to Voronoi lines (branches)
- Delaunay triangles to Voronoi vertices.

In the simplicial complex of figure B.1 the seven vertices may be taken as the basis of the algebraic construct of a vector space. Similary, the twelve branches could be taken as a base of another vector space and the six triangles could serve as a base for a further vector space. A vector in these spaces is called a chain c and could be written as a linear combination with its base elements σ_i:

$$c = \sum_i a_i \cdot \sigma_i \tag{2-0.1}$$

where a_i are the coefficients and σ_i are the simplices constituting the base.
A p-chain is a chain c with a base of p-simplices.

In a linear vector space exists a scalar product:

$$< c, c^* >= \sum_i a_i \cdot a_i^* \tag{2-0.2}$$

which assigns to each chain a co-chain c^* (the dual of the chain c)

$$c^* = \sum_i a_i^* \sigma_i^* \tag{2-0.3}$$

where a_i^* are the vector components in the co-space and σ_i^* is the basis in the co-space.

Since the vector space and its co-space have a similar structure (they are isomorph), usually no distinction is made between them. However, since they are different spaces, they have to be considered as physically different, even if their structure is similar (see the network of figure B.1)

This duality between the vector space of chains and the space of co-chains shows up in the structural relations of the geometric object, as the one of figure B.1. In higher dimensional Delaunay triangulation, the elementary objects forming the basis of chains are in 3-D tetrahedra and in 4-D pentahedra. For biological objects, the 3-D duality is striking, each vertex (nucleus) is associated with an Voronoi territory of a 3-D volume (the biological cell).

B.0.2 Orientation of simplices and chains

Orientation is an important property which a simplex may posses. Intuitively the sequence of vertices in the simplex notation $[P_1, P_2, P_3, \ldots, P_{d+1}]$ defines an orientation (path). For a line segment $[P_1, P_2]$ the orientation is indicated by an arrow usually (see figure B.2) and for a triangle $[P_1, P_2, P_3]$ by circular arrows (see figure B.3) which may be clock or counter clock wise oriented. For a tetrahedron $[P_1, P_2, P_3, P_4]$ an upward or down ward spiral (right or left handed) marks the orientation (see figure B.4).

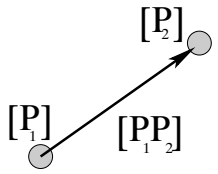


Figure B.2: *Orientation of a 1-simplex displayed by an arrow.*

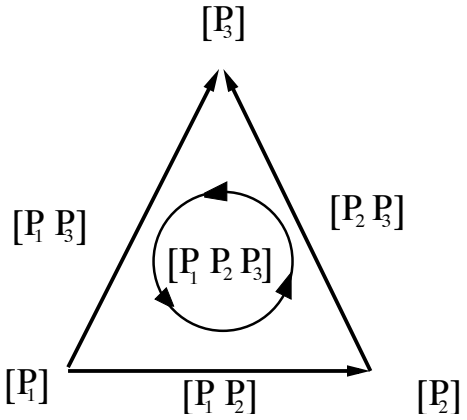


Figure B.3: *Orientation of a 2-simplex (triangle) by clock- or anticlockwise rotation.*

The sequence of vertices in the simplex notation $[P_1, \ldots P_n]$ is arbitrary. However, this is no longer the case, if the simplex itself is oriented, i.e. in $[P_1, P_2]$ the arrow points to P_2 and in $[P_2, P_1]$ it point to P_1. If the position of two points in the sequence of a triangle is changed (2-simplex), the orientation of the triangle

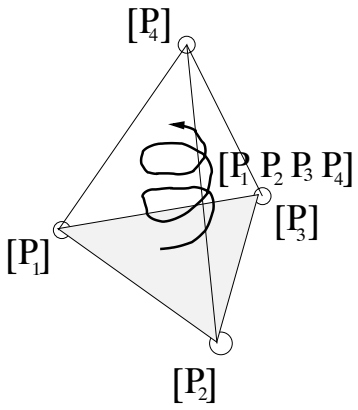


Figure B.4: *Orientation of a 3-simplex (tretahedron) by up- or downward spiral.*

is changed (from clock-wise to counter clock-wise). The same holds for the tetrahedron. In general, the orientation (class) is given by all even permutations of vertices and the other by all odd permutation.

The same rule applies to the co-space base i.e. $[A, B]$ and $[B, A]$ have opposite orientation in figure B.1. Note that vertices have no orientation.

B.0.3 Boundary of chains and co-chains in simplicial complexes

A very important operator in a linear vector space is the linear map named the boundary operator ∂, which acts on vectors (chains) by transforming (i.e. map) the basis:

$$\partial c = \sum_i a_i \partial \sigma_i \tag{2-0.4}$$

In the co-space, a co-boundary operator ∂^* is defined by the scalar product:

$$< \partial c, c^* > = < c, \partial^* c^* > \tag{2-0.5}$$

The co-boundary operator acts on co-chains:

$$\partial^* c^* = \sum_i a_i^* \partial^* \sigma_i^* \tag{2-0.6}$$

The action of the boundary operator is to find the boundary (border) of the objects (chains, co-chains) to which it is applied.

The boundary operation for simplices delivers:

- an oriented line $[P_1, P_2]$ has as boundary, the two end-nodes (see figure B.2) given by vector basis $[P_1], [P_2]$;

$$\partial[P_1, P_2] = [P_1] - [P_2] \tag{2-0.7}$$

- a vertex has no more boundary $\partial[P_i] = 0$
- a triangle has as boundary (see figure B.3) its oriented edges $\partial[P_1, P_2, P_3] = [P_1, P_2] + [P_2, P_3] - [P_3, P_1]$
- a tetrahedron gives upon boundary operation $\partial[P_1, P_2, P_3, P_4] = [P_1, P_2, P_3] - [P_2, P_3, P_4] + [P_3, P_4, P_1] - [P_4, P_1, P_2]$ etc. (compare figure B.4)

In general for an oriented simplex the boundary operation is given by:

$$\partial[P_1, P_2, P_3, \ldots, P_i, \ldots, P_d] = \sum_{j=0}^{d} (-1)^j \; [P_1, P_2, P_3, \ldots, \hat{P}_j, \ldots, P_d] \tag{2-0.8}$$

where $\hat{P}_j$, meaning the vertex j, is omitted in the sequence.

In figure B.1, the boundary operator performs a contraction in a way that the area of each triangle is contracted to its triangle facets (i.e. bounding sides) [Fla63].

If the orientation in figure B.1 is considered, each side of a triangle, which belongs to two triangles, may be canceled, because it appears in opposite orientation in the adjacent triangles. Hence the facets (sides of the triangle) inside to the hexagon of figure B.1 annihilate each other but, the outer facets do not. Thus, only the outer boundary of the hexagon is left after applying the boundary operator.

The boundary operator generates the bounding surface to the volume and lowers by that the dimension of the space (volume) by one (to a surface). The fundamental property of a boundary operator is that the boundary of a boundary vanishes. For a chain c:

$$\partial \cdot \partial c = \emptyset \tag{2-0.9}$$

because the surface that surrounds a volume has no border since it is closed.

Again, using the inner scalar product $< \partial\partial c, c^* >=< c, \partial^*\partial^* c^* >= \emptyset$, a property analogous to $\partial\partial c = \emptyset$ may be derived for the co-chain c^*, i.e. the boundary of a boundary vanishes for co-chains also:

$$\partial^* \cdot \partial^* c^* = \emptyset \tag{2-0.10}$$

The boundary operation divides the chains c into two classes:

- Chains c, for which $\partial c = 0$ (named cycles or cyclic-chains) which represent from a geometrical point of view already a closed surface (closed chain).
- Chains c where the boundary operator delivers the boundary $\overline{c}$ (named boundary or boundary-chain) $\overline{c} = \partial c$ (e.g. the surface of a volume).

Analogously, co-chain c^* that fulfill $\partial^* c^* = 0$ are called co-cycles. The co-chains $\overline{c}^*$ obtained by the boundary operation $\overline{c}^* = \partial^* c^*$ are named co-boundaries.

Moreover, the inner scalar product $< z, \partial^* c^* >=< \partial z, c^* >= 0$ states that cycles are orthogonal to co-boundaries, since for cycles (here named z) $\partial z = 0$. Hence, a chain can be decomposed into the sum of a cycle-chain and a boundary-chain.

B.0.4 Matrices of boundary operators and network theory

The adjacency of objects in a network may be given in a matrix (see table B.1) by indexing the rows i and columns j in the same sequence of objects. In figure B.1 for the seven vertices, the adjacency in the Delaunay triangulation is given by an

entry of 1 at the position (i, j), if the vertex i is connected with vertex j by a solid line.

	1	2	3	4	5	6	7
1	0	1	0	0	0	1	1
2	1	0	1	0	0	0	1
3	0	1	0	1	0	0	1
4	0	0	1	0	1	0	1
5	0	0	0	1	0	1	1
6	1	0	0	0	1	0	1
7	1	1	1	1	1	1	0

Table B.1: Adjacency matrix of vertices of the Delaunay triangulation of figure B.1

The square matrix is symmetric, since if vertex i is adjacent to j, vertex j is adjacent to i. If the orientation of the connectivity is incorporated the matrix would be antisymmetric. The adjacency of the six triangles in figure B.1 is given in the table B.2

	$[1,2,7]$	$[2,3,7]$	$[3,4,7]$	$[4,5,7]$	$[5,6,7]$	$[6,1,7]$
$[1,2,7]$	0	1	0	0	0	1
$[2,3,7]$	1	0	1	0	0	0
$[3,4,7]$	0	1	0	1	0	0
$[4,5,7]$	0	0	1	0	1	0
$[5,6,7]$	0	0	0	1	0	1
$[6,1,7]$	1	0	0	0	1	0

Table B.2: Adjacency matrix of the triangles of the Delaunay triangulation of figure B.1

The connectivity expressed in an adjacency matrix may be converted into a matrix table B.3 relating the 7 vertices to the 12 connecting line segments (branches) in the Delaunay triangulation of figure B.1.

In that matrix the rules are applied.

$$a_{ij} = \begin{cases} +1 & \text{if the branch [i,j] is orientated towards the vertex j} \\ -1 & \text{if the branch [i,j] is orientated towards the vertex i} \\ 0 & \text{if no branch exists connecting the vertices i and j} \end{cases}$$

a_{ij}	$[1,2]$	$[1,7]$	$[2,3]$	$[2,7]$	$[3,4]$	$[3,7]$	$[4,5]$	$[4,7]$	$[5,6]$	$[5,7]$	$[6,1]$	$[6,7]$
1	1	1	0	0	0	0	0	0	0	0	-1	0
2	-1	0	1	1	0	0	0	0	0	0	0	0
3	0	0	-1	0	1	1	0	0	0	0	0	0
4	0	0	0	0	-1	0	1	1	0	0	0	0
5	0	0	0	0	0	0	-1	0	1	1	0	0
6	0	0	0	0	0	0	0	0	-1	0	1	1
7	0	-1	0	-1	0	-1	0	-1	0	-1	0	-1

Table B.3: *Incidence matrix of the boundary operator of the 1-chain of the Delaunay triangulation of figure B.1*

In the example of figure B.1, it may be assumed that the branches (line segments) are oriented always from the vertex named with the lower number to the one with the higher number.

The matrix of table B.3 is also called the "node to branch" incidence matrix or simply incidence matrix. By the definition of the boundary operator (see 2-0.4) it is the matrix representation that maps the chain of 1-simplices of the Delaunay triangulation to 0-simplices (vertices). The elements of the matrix (table B.3) may be obtained from equation 2-0.4 for chains by setting the chain coefficients to the orientation (i.e. $+1$,-1) of the chain of 1-simplices in the base and applying equation 2-0.8.

Similarly, the matrix B.4 for the boundary operator could be calculated between the chain of 2-simplices (triangles) in figure B.1 and the chain of 1-simplices by setting the coefficient of the chain of 2-simplices to their orientation (i.e. $+1$,-1) and applying equation 2-0.8.

This matrix is known as "the branch to mesh matrix", since a triangle may be considered as a mesh in the network of branches. The entries in the matrix

m_{lk} may be obtained in the same sequence of the edge-nodes (branches) as in table B.3 as columns and the triangles (2-simplex as rows in table B.4).

$$m_{lk} = \begin{cases} +1 & \text{if the branch [l,k] is oriented in direction of the loop or mesh} \\ -1 & \text{if the branch [l,k] is not oriented in direction of the loop} \\ 0 & \text{if no branch [l,k] exists} \end{cases}$$

m_{kl}	$[1,2]$	$[1,7]$	$[2,3]$	$[2,7]$	$[3,4]$	$[3,7]$	$[4,5]$	$[4,7]$	$[5,6]$	$[5,7]$	$[6,1]$	$[6,7]$
$[1,2,7]$	-1	1	0	-1	0	0	0	0	0	0	0	0
$[2,3,7]$	0	0	-1	1	0	-1	0	0	0	0	0	0
$[3,4,7]$	0	0	0	0	-1	1	0	-1	0	0	0	0
$[4,5,7]$	0	0	0	0	0	0	-1	1	0	-1	0	0
$[5,6,7]$	0	0	0	0	0	0	0	0	-1	1	0	-1
$[6,1,7]$	0	-1	0	0	0	0	0	0	0	0	-1	1

Table B.4: *The "branch to mesh matrix" of the boundary operator of the 2-chain of the Delaunay triangulation of figure B.1*

The matrices for the co-boundary operator are derived from the boundary operator by the structural similarity of their dual (see appendix B.0.3) i.e. the vertex, the line segment and the triangle specification in table B.3 and table B.4 are replaced by their co-space equivalent e.g.:

- vertex [1] by the area [A,B,G], vertex [2] by [B,C,G], ..., vertex [7] by [A,B,F]
- line segments [1,2] by [B,G], [2,3] by [C,G], ..., [6,1] by [A,G] and [1,7] by [A,B], [2,7] by [B,C], ..., [6,7] by [F,A]
- triangles [1,2,7] by [B], [2,3,7] by [C], ..., [6,1,7] by [A]

This similarity in the operators displays again that the Delaunay space and the Voronoi co-space are topologically isomorph.

The major topological identity 2-0.9:

$$\partial \cdot \partial = 0 \tag{2-0.11}$$

may be verified by the multiplication of the corresponding matrices of the boundary operator (see tables B.3 and B.4)

$$\sum_j a_{ij} \cdot m_{jk} = 0 \tag{2-0.12}$$

The topological relationship above is part of the network theory. Network theory is far developed for electronic circuits. Here, Kirchhoff's laws are well known. In the following, the relation of Kirchhoff's laws to boundary operators is depicted. In network theory the two boundary matrices are named the "node to branch" and the "branch to loop" (mesh) matrix.

In electrical networks, in each branch a current flows and across each branch of the network, there is a potential drop, since in a branch an electrical element is located (e.g. a resistor or capacitor etc.). The electrical network theory formulates Kirchhoff laws for currents as:

$$\sum_j a_{ij} \cdot J_j = 0 \qquad \text{cycles of boundary operator } a_{ij} \tag{2-0.13}$$

which states that the sum of currents J_j in and out of each node is conserved. Kirchhoff voltage law:

$$\sum_j m_{ij} \cdot U_j = 0 \qquad \text{co-cycles of co-boundary operator } m_{ij} \tag{2-0.14}$$

states that the potential U_j across a branch is unique in the whole network and cycling around a loop (mesh) there is a unique potential at every point.

In the developing aggregate of cells in the egg, the electrical properties of the network are at the first instance not as interesting as e.g. the forces that move the cell to a new position or as those forces that modify the shape of cells. Actually, these forces are essentially not of electrical nature.

Fortunately, Kirchhoff laws apply to all thermodynamic systems (an electric system is a thermodynamic systems also) i.e. they apply to generalized forces and fluxes. Hence, one may replace the electrical potential U_j by the mechanical

forces K_j and the electrical flow J_j by the mechanical displacement S_j The laws read now as follows: For the equilibrium of the forces:

$$\sum_j m_{ij} \cdot K_j = 0 \tag{2-0.15}$$

and for the conservation of the displacement:

$$\sum_j a_{ij} \cdot S_j = 0. \tag{2-0.16}$$

As we have an electrical potential $U(j)$, there exists in the mechanical case a vector potential $K(j)$ of forces, which drives the cellular motion by the displacement $S(j)$.

Tellegen's Theorem is another topological identity, which states for currents J_j (chains) and voltages U_j (co-chains):

$$\sum_j J_j \cdot U_j = 0 \tag{2-0.17}$$

and for forces K_j and displacements S_j:

$$\sum_j K_j \cdot S_j = 0 \tag{2-0.18}$$

The meaning of the equations 2-0.17 and 2-0.18 is a restriction to the movement within the network. It says that the egg as a hole does not actively move.

Even in the more general form, where the "forces" and "fluxes" originate in different vector spaces, these laws are valid, even if K_j' and S_j' are generalized forces and fluxes, resp. in another space of same dimension (for more details see [OD71]).

B.0.5 Structure of discrete and contineous Systems and memory

The structure of systems may be coded by linear vector spaces of chains. These vector spaces are related by the "de Rham" [dR69] homology (co-homology) using the boundary operation with its remarkable property:

$$\partial \cdot \partial = \emptyset \tag{2-0.19}$$

Up to now, the systems considered were discrete. Let the volumes of the systems e.g. of the simplices, tend to zero and let the system still be finite, then a continuous system is obtained. The system structure of continuous systems is usually a regular lattice (like that of an indefinite crystal). Differential geometry delivers the tools, in which such systems may be described (see e.g. [Fla63]). The analog to a chain is a differential form and the dual to the co-chain is a partial derivative. The boundary operator of a boundary relation is expressed as:

$$d \cdot d = 0 \tag{2-0.20}$$

and in the co-space as partial derivatives:

$$\partial \cdot \partial = \emptyset \tag{2-0.21}$$

For the natural 3D space, the boundary operators are identified as:

$$div \cdot curl = 0 \tag{2-0.22}$$

and

$$curl \cdot grad = 0 \tag{2-0.23}$$

The corresponding scalar products are recognized as the "Greens Law" (see [Fla63]) on volume and surface integrals. In the case of chains the adjacency is expressed on the basis of the vector space and displays the topology, whereas the events expressed themselves in the coefficients of the chains. In continuous systems, the adjacency is incorporated in the base of the independent dx_i and the events are expressed in the laws depending on dx_i. Note, that the base of a chain may be composed of other geometrical objects than simplices embedded in a geometrical space.

Whereas, the neighbor relationship in the space lattice given by the dx is ordering of points in a 1-D line it can be more complex by far, if adjacency matrices are used to display it. The structure inherent in an adjacency matrix

needs a specific coding to be able e.g. to realize the same system later on. The coding might be considered as the construction plan of the system. The code might be used to store the systems structure e.g. in a memory. If the system itself carries the structural information, it might be considered having a memory. Especially interesting is the case, in which the system can be assembled from the memory and can modify the memory. The adjacency may be considered in this context as a binding of system compartments. Such connections may be coded in a memory and may be used on demand by the system. Systems of this kind have additional properties compared with those with fixed structure (connections), which are treated in classical physical environments usually. The added properties of memories may well be regarded as a piece of development, as it is seen in biological nature and as is tried to show here in the early development of the worm C. elegans.

Systems,which have incorporated a memory capable of realizing the basis of dynamics and morphogenesis apparently have more degrees of freedom than the classic systems of natural science.

Moreover, the independent variables that determine the dynamics need not be the variables used in topological representation, even though they might.

B.0.6 The coding of a simplex complex by stripes

In the topology a d-simplex may be represented by d+1 points that constitute it as $[P_1, P_2, \ldots, P_{d+1}]$ (see appendix A.0.1). To facilitate the writing, P may be omitted and indices of points only indicated i.e. $[1, 2, \ldots, d+1]$ instead of explicitly $[P_1, P_2, \ldots, P_{d+1}]$ written. Thus, a triangle (2-simplex) with the points 1,2 and 3 is written as the sequence $[1, 2, 3]$

A sequence of neighboring triangles shown in figure B.5 could be given as e.g.:

[4,1,5] [1,5,2] [5,2,6] [2,6,3] [6,3,7]

or may be formulated as a sequence of numbers (as triangle names) in a strip

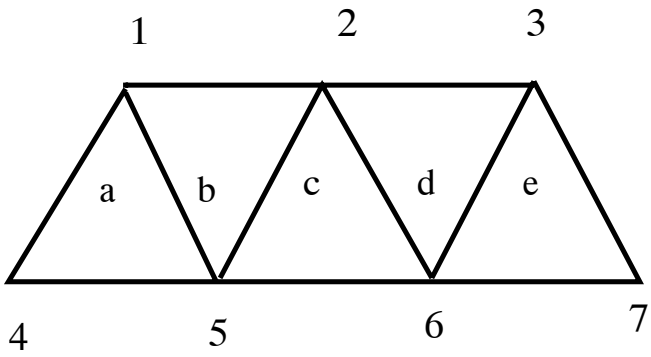


Figure B.5: *Trianglestrip [4,1,5,2,6,3,7] of the connected triangles a to e*

$$[4, 1, 5, 2, 6, 3, 7]$$

in which each consecutive three digits constitute a triangle (e.g. [4,1,5]). Note, that the numbers in a "set" for a triangle may be reordered cyclically, e.g. $[1, 4, 5]$, $[4, 1, 5]$, $[4, 5, 1]$ etc., all represent the same triangle.

A strip is a specific neighbor relation between the elements comprising a strip because:

- 1. they are adjacent
- 2. the common facet is given by the latest edge-nodes (the latest two for a triangle) in the sequence of the strip

Starting at a triangle, the strip may be extended in both directions, until the strip terminates either by reaching its other end (closed strip), or if no more triangles could be found i.e. added to the strip (open strip). A compact description of a complex of d-simplices is obtained if triangle (or d-simplex) strips are terminated, as soon as they touch another strip, which is already established in the procedure of searching for strips (see [SML98] and the "strip algorithm" in appendix E.0.4). By this method, a set of independent simplex-strips is obtained, which completely describes the topological structure of the simplicial complex. Note, none of these strips of the complex have a triangle in common but every triangle is contained in one strip.

The facets, i.e $(d-1)$-simplices of a d-simplex, are a closed strip of $(d-1)$-simplices by the above notion, i.e. the boundary operation on a d-simplex leads to a closed strip.

Appendix C

Segmentation

C.0.1 Normalized Cuts

Graph partitions and normalized cuts

There are many possibilities for segmenting an image into objects (up to the extreme case that every pixel is one object or entity). The question remains, how to pick the right partition, so that a segmentation into meaning-full objects results. Several approaches, which compare images on the pixel scale, involve a priori knowledge in a given context. If no a priori knowledge is available, an unbiased measure is required about pixels that belong to an object and on the definition of the complement pixel set.

An unbiased measure for image segmentation was introduced by [JL01] and [TM01]. The aim was to partition the image into a hierarchical set of subsets (i.e. a tree-like structure that assembles the scene beginning with coarse scale partitions and ending with fine scale partitions). The partitioning should use information about the coherence of luminescence, contour, texture, color and motion to extract a global impression of a scene.

The segmentation problem may be formulated as a problem of graph theory, where the graph describes an image assembled as a set of features. The parti-

tioning of a graph reduces to a generalized eigenvalue problem. The solution of the generalized eigenvalue problem involves a measure, i.e. the normalized cut of the graph. This is an unbiased measure with the characteristics, that if it is minimized, the similarity of pixels within one group and at the same time the dissimilarity between partitioned sub-groups is maximized.

The algorithmic procedure starts with an image, in which the pixels are taken as a set of features in a feature space. The feature set is represented as a weighted, undirected graph $G = (V, E)$ (V =Vertices, E =Edges), in which the nodes of the graph are points in the feature space, and the edges between pairs of nodes are weighted with the similarity function $w(i, j)$ of the nodes i and j. (Note, the similarity functions $w(i, j)$ may be forces in the sense of appendix B). The aim is to partition the set of vertices V in such a way into sub-sets $V_1, V_2, \ldots, V_n$ that the similarity is high within one sub-set and low between the partitioned sub-sets. The same partitioning process may now be applied recursively to each subset. The graph is separated into two partitions, A and B, by cutting edges between certain nodes. The partition of the graph is now chosen in such a way that the sum is minimized of all weighted edges that are cut.

$$cut(A, B) = \sum_{i \in A, j \in B} w(i, j)$$

There are efficient algorithms to compute the minimal cut. The cut criteria and the weighing function determine the partitioning of the graph. The cut criteria as well as the weighing function must be chosen adequately to yield segmentation results which are consistent with human segmentations (i.e. by human intuition). The equations for the cut criteria may be minimized and reduced to the generalized eigenvalue system

$$(D - W)x = \lambda Dx$$

where D and W are square matrices. D is a diagonal matrix with d on its diagonal, with $d(i) = \sum_j w(i, j)$ is the total weight from node i to all other nodes.

W is a symmetric matrix $W_{i,j} = w(i,j)$. x is an indicator vector, with $x_i = 1$, if the node i is in the partition A and $x_i = -1$ otherwise. λ_i are eigenvalues.

The generalized eigenvalue system has several eigenvectors as solution. It can be shown that the second smallest eigenvector is a solution to the minimized cut problem and may be used to partition the graph into disjunct sub-sets. This procedure may now be applied to each subset for further partitioning. The decomposition into a hierarchically ordered set system may terminate, when a certain limit of the cut value is reached.

Sofar, the procedure has made use of gray scale luminescent histogram characteristics.

C.0.2 Color, texture and conture segmentation

The color information contained in an image must be applied by a color based similarity measure to the above partition of subsets to achieve a useful segmentation. Therefore, the similarity measures for contour, texture and color must be combined.

For a given partition, the color statistics is collected across one subset of the partition. The partition results are used to construct windows, in which the color statistics is calculated. This leads to partitions into colored textures where one texture is made of different colored elements. The similarity measures for color is defined as a difference between color histograms over the texture window (see [TM01]). Two histograms h_i and h_j are compared with the χ^2 test (see [Sac02])

$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \tag{3-0.1}$$

The color similarity measure W_{ij}^{color} for two pixels i and j is now defined with the standard deviation σ_{color} as:

$$W_{ij}^{color} = exp^{(\frac{-\chi^2(h_i,h_j)}{\sigma_{color}})} \tag{3-0.2}$$

The combination of the analogous similarity measures for texture $W_{ij}^{texture}$ and contour $W_{ij}^{contour}$ with W_{ij}^{color} is assumed to be multiplicative. The total measure of these three statistical measures is obtained by a logical AND operation:

$$W_{ij} = W_{ij}^{color} \times W_{ij}^{texture} \times (W_{ij}^{contour} + \epsilon) \tag{3-0.3}$$

where ϵ is a small deviation parameter for error tolerance (for details see [JL01]).

Appendix D

Videos

In this section, the video sequences are compiled, which are referred to in the text.

D.0.1 Growth of an embryo in the egg until hatching

Figure D.1: *The video freeze image shows the same sequence of events as in figure 2.9, at the lower magnification* 400×

Figure D.2: *The video freeze image shows an embryo after* ≈ 14 *hours of development hatching from the egg (magnification* $400\times$*)*

Figure D.3: *The video freeze image shows the scene of figure D.2* ≈ 1 *hour later. The worm is alive and leaves the egg (magnification* $400\times$*.)*

D.0.2 Preparation procedures of an egg from the alive worm

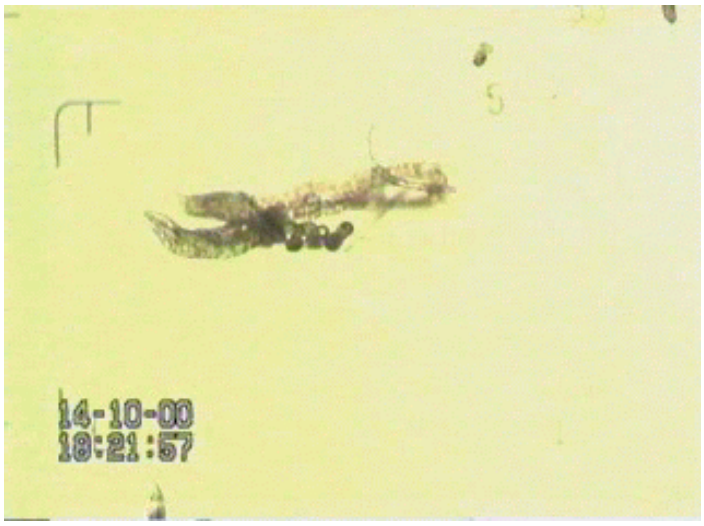


Figure D.4: *The video freeze image shows a typical scene after a worm is cut. The worm cracks and folds into an "X"-like shape. In one branch (lower right), the uterus is still harboring the eggs.*

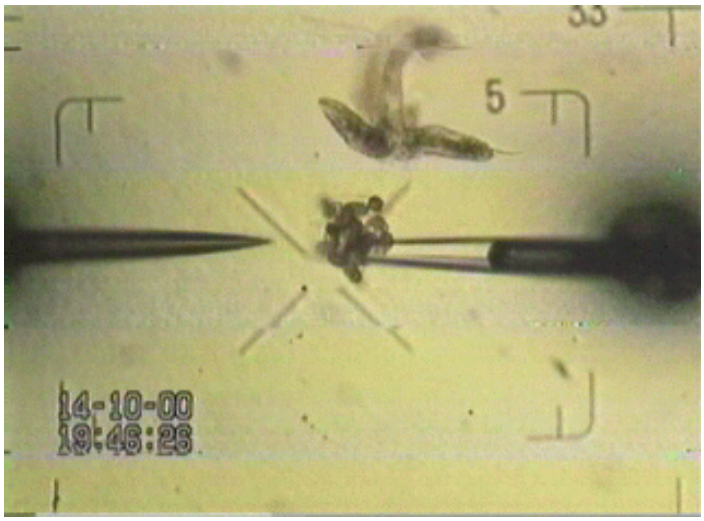


Figure D.5: The video freeze image demonstrates the procedure of dissecting the uterus from the remaining body and isolating one specific egg.

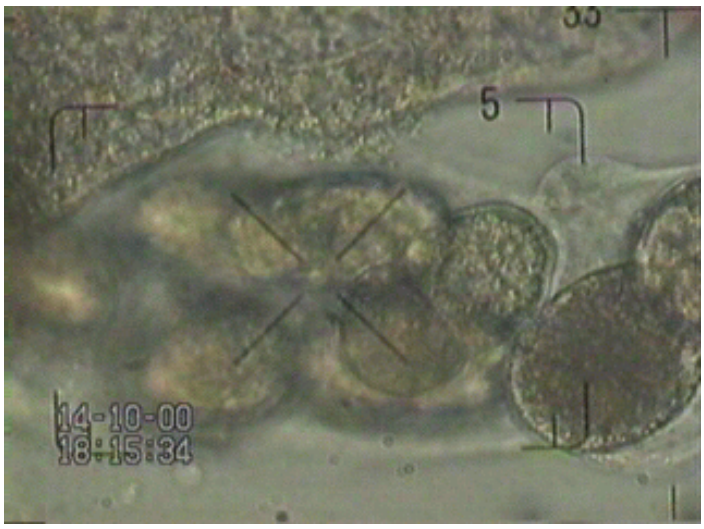


Figure D.6: *The video freeze image shows the uterus with eggs. The egg positioned slightly to right of the center is in the 2-cell state and will be isolated for further treatment (magnification* 400×*).*

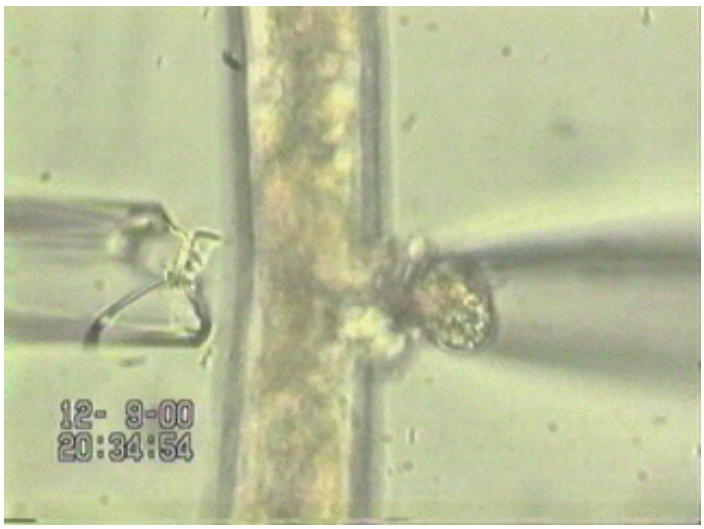


Figure D.7: The video freeze image demonstrates the procedure of isolating a specific egg from a worm. The worm is cut and the egg of interest is extracted. The extracted egg is fixed at the right electrodes tip. (The worm is still alive after the egg was isolated by surgery) (magnification 200×).

Figure D.8: *The video freeze image shows a scene fractions of a second after a worm is cut with electrodes into halves. The worm explodes and eggs are thrown out of the ruptured uterus. The same result is achieved as in figure D.11, without time consuming fixation (magnification* 200×*).*

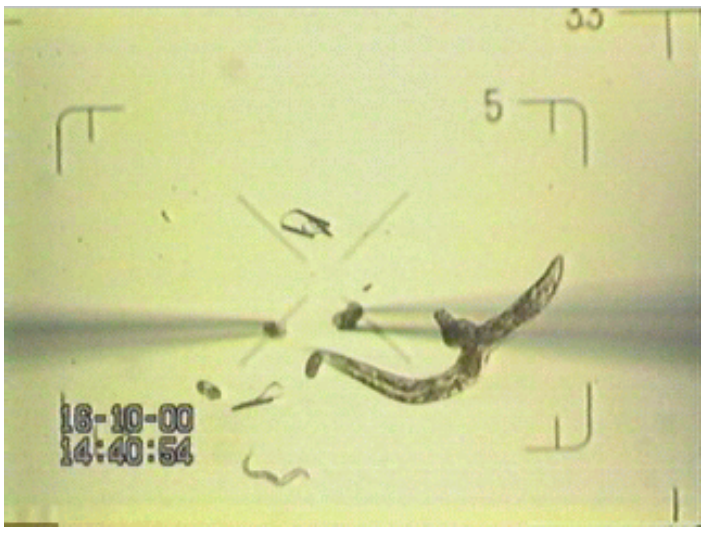


Figure D.9: *The video freeze image shows a collection of separated eggs. Eggs of interest are drawn into micro electrodes and transfered by them to further preparation (magnification* 40×*.*

Figure D.10: *The video freeze image demonstrates the effect of the osmotic pressure in a 2‰ egg buffer instead of a 5‰ as suggested (see section 2.2). Eggs leaving the uterus may explode. The worm may finally explode also.*

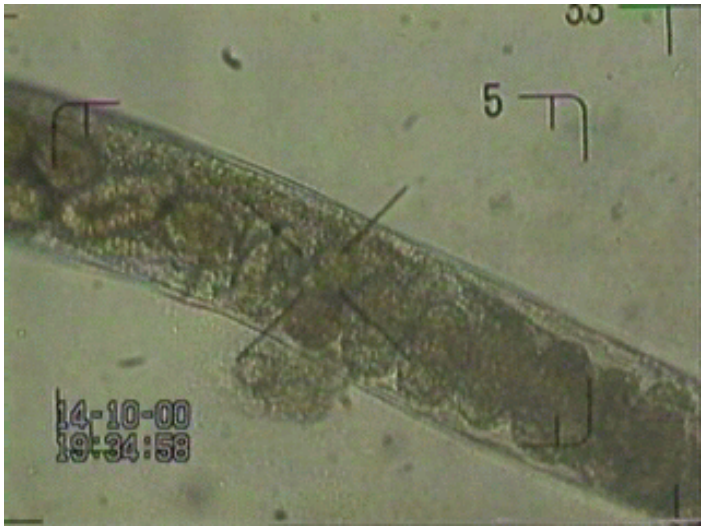


Figure D.11: *The video freeze image shows, how a worm explodes. The female is in a* ≈ 3.5‰ *egg buffer, which is comfortable to developing embryos. The worn is damaged by a considerable cut into the cuticula. Tissue movements inside the worm are visible and tissue is leaving the worm through the cut. Finally, the worm explodes and sets almost all the eggs free. The eggs are lying around and may be captured easily. This procedure is by far the most effective one to isolate the egg of interest (magnification* $200\times$*).*

D.0.3 Demonstration of the rigidity of an egg

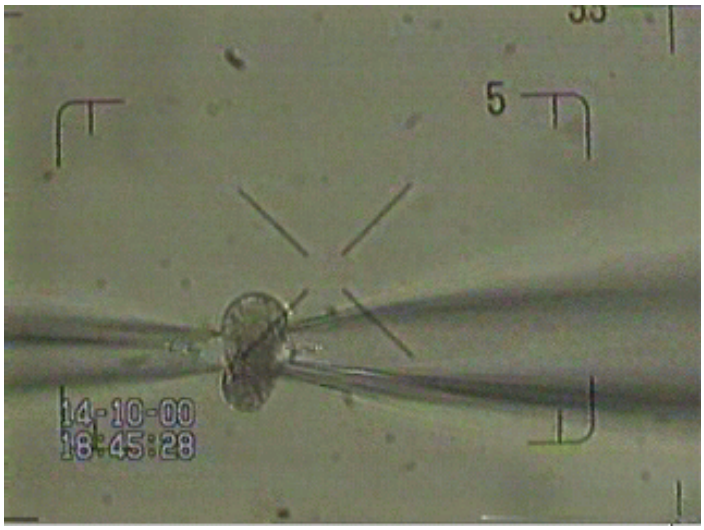


Figure D.12: *The video freeze image demonstrates the stability of a C. elegans egg shell enclosing an embryo in the ≈100-cell state. The egg measures ≈60 μm in length. The egg is placed between two electrodes, of which each is attached to a micromanipulator. The egg and electrodes are placed in a ≈150 μl egg buffer in a petri dish. The egg is squeezed without being cracked. One gets an impression of the eggs stability and forces which must be applied by a micro injection in this state (magnification* 200×*).*

Appendix E

Electronic storage media

This section list the mayor topics on the electronic media.

E.0.1 Thesis

A copy of the thesis in pdf format with activatable videos

Content:

Thesis_tex

Videos

D_Thesis.pdf

D_Thesis.html

E.0.2 Reproducible research

The programs in this section reproduce the figures. It may consulteted if the details on how a figure or a table is calculated are of interest. Call either the program 'figures' or 'tables' and mark the figure or table which should be recalculated by a 'Y'.

Content:

Figures

Tables

programs

E.0.3 Browser and database

Here the demonstration version of the browser of SIMI-BioCell and the corresponding database is found.

Content:

sbctrial.exe

demo_1.sbc

demo_1.sbd

demo_2.sbc

demo_2.sbd

n2_lev00.avi

E.0.4 Suplemental programs

Several programs are collected in this section which either have been used in or written for this work.

Content:

SelectProgram

VTKprogram

programs

Programs

Matlab Code

C Code

E.0.5 Utilities

Content:

Ascii2SIMI

SIMI2Ascii

qhull2struc

struc2qhull

struc2tab

tab2struc

struc2vtk

vtk2struc

Matlab Code

C Code

E.0.6 Miscellaneous

Matlab Code

C Code

Appendix G

Literature

Bibliography

[AU77] A. Aho and J.D. Ullman. Principles of compiler design. *Addison Wesley*, 1977.

[Bau92] H. Bauer. *Maß und Integrations Theorie*. NewYork: de Gruyter, 1992.

[BB99] Anton Betten and Dieter Betten. The proper linear spaces on 17 points. *Discrete Applied Mathematics*, 95:83–108, 1999. http://www.mathe2.uni-bayreuth.de/betten/PUB/pub_proper17.html.

[BBT01] A. Betten, D. Betten, and V. D. Tonchev. Unitals and codes. *Preprint submitted to Elsevier Science*, 2001.

[BDH96] C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. The quickhull algorithm for convex hulls. ACM Transaction on Mathematical Software, Dec 1996. http://www.acm.org/pubs/toc/Abstracts/toms/235821.html http://www.geom.umn.edu/software/qhull/ ftp://geom.umn.edu/pub/software/qhull-96.ps.Z.

[Ber79] J. Berstel. *Transductions and Context-Free Languages*. B.G. Teubner, Stuttgart, 1979.

[BH73] HG. Busse and B. Hess. Information transmission in diffusion-coupled oscillatory chemical systems. *Nature*, 244:203–5, 1973.

[Dau92] I. Daubechies. Ten lectures on wavelets. *SIAM Philadelphia PA.*, 1992.

[Del34] B. Delaunay. Sur la sphere vide. *Bull. Acad. Sci. USSR(VII) Classe Sci. Mat. Nat.*, 1934.

[DK69] C.A. Desoer and E.S. Kuh. *Basic Circuit Theory*. McGraw-Hill New York, 1969.

[dR69] G. de Rham. Lectures on introduction to algebraic topology. *Bombay : Tata Inst. of Fundamental Research*, 1969.

[DVB91] J. Das, E. Valkema, and H.-G. Busse. *Boundary operator and distance measure for the cell lineage of C. elegans and for the pattern in Fusarium Solani*, chapter 14, pages 183–192. Complexity, Chaos and Biological Evolution. Plenum Press New York, 1991.

[EH86] H.M. Ellis and H.R. Horvitz. Genetic control of programmed cell death in the nematode c. elegans. *Cell*, 44:817–829, 1986.

[ELZ00] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. In "Proc. 41st IEEE Sympos. Found. Comput. Sci. 2000" 454-463, Department of Computer Science at Duke University, 2000. http://www.cs.duke.edu/ edels/TriTop/.

[EMB] EMBO. European molecular biology organization. http://www.embo.org/.

[FE96] M. Facello and H. Edelsbrunner. *Geometric techniques for molecular shape analysis.* PhD thesis, University of Illinois, 1996.

[Fer01] Jean-Albert Ferrez. *Dynamic triangulations for efficeint 3d simulation of granular materials.* PhD thesis, Ecole Polytechnique Federale de lausanne, 2001. http://rosowww.epfl.ch/jaf/data/these-jaf.pdf.

[Fla63] H. Flanders. *Differential forms with applications to the physical sciences.* New York : Acad. Press, 1963.

[Geo] Geomview.org. Geomview 1.7.7 geometry center university of minnesota. software@geomview.org. http://www.geomview.org/.

[Gol97] A. Goldbeter. *Biochemical oscillations and cellular rhythms : the molecular bases of periodic and chaotic behaviour.* Number ISBN 0-521-40307-3. Cambridge University Press,, 1997.

[GR97] Jacobs E. Goodman and Joseph O. Rouke. *Handbook of Discrete and Computational Geometry.* Number ISBN 0-8493-8524-5. CRC Press New York, 1997.

[Gra93] F. Graner. Can surface adhesion drive cell-rearrangement? part 1: Biological cell-sorting. *J. theor. Biol.*, 164:455–476, 1993.

[GS93] F. Graner and Y. Sawada. Can surface adhesion drive cell-rearrangement? part 2: A geometric model. *J. theor. Biol.*, 164:477–506, 1993.

[GWZM98] M. Gumbel, O. Werner, G. Zajicek, and H.P. Meinzer. Simulation and 3-d visualization of the intestinal crypt. *In Proceedings of the 12th European Simulation Multiconference (Manchester, UK)*, pages 310–312, 1998.

[HÖ1] M.T. Hütt. *Datenanalyse in der Biologie - Methoden der nichtlinearen Dynamik, fraktalen Geometrie und Informationstheorie.* Springer-Verlag, Heidelberg, 2001.

[Hal01] David Hall. The center for c. elegans anatomy. *Albert Einsten College of Medicine NY.*, 2001. http://www.aecom.yu.edu/wormem/.

[HBBW+00] N. Hilschmann, H.U. Barnikol, S. Barnikol-Watanabe, H. Götz, H. Kratzin, and F.P. Thinnes. Das immun- und das nervensystem. *Nachrichten der Akademie der Wissenschaften in Göttingen*, 5:1–67, 2000.

[Hea01] Paul J. Heid and et. al. A computer-assisted system for reconstructing ... *2001 International Worm Meeting abstract 841*, 2001.

[Hil99] D. Hilbert. *Foundations of geometry.* Open Court La Salle, IL., 1899. Reprint 1971.

[HM02] A. Hartley and C. Marks. *ATLAS OF C. elegans ANATOMY.* http://www.wormatlas.org/handbook, 2002. in preparation.

[HRL75] G. T. Herman, G. Rozenberg, and A. Lindenmayer. *Developmental Systems and languages.* Number ISBN 0 7204 2806 8. North Holland Publishing Company, 1975.

[IBM02] IBM. *openDX IBM Data Explorer.* IBM Research, 2002. http://www.opendx.org/ and http://www.research.ibm.com/dx.

[Inca] Clustan Inc. Clustan. www.clustan.com.

[Incb] Simi Inc. Simibiocell. www.simi.com.

[JL01] S. Jianbo and M. Litendra. Normalized cuts and image segmentation. Technical report, Robotics Institute, Carnegie Mellon, Nov 2001. www.ri.cmu.edu /pub_files/pub3/shi_jianbo_2000_1 /shi_jianbo_2000_1.pdf.

[JMS92] R. Jhon, C. Moler, and R. Schreiber. *MATLAB: Sparse Matrices in MATLAB and Design Implementations*, chapter 9, pages 333–356. Number 1. MATLAB Inc, 1992.

[Kai92] G. Kaiser. An algebraic theory of wavelets. *SIAM J. Math. Anal.*, 23:222–243, 1992.

[KB00] A. Krämer and H.-G. Busse. Topologizer for analysis and visualization of microscopic slide stack by vtk. In *Visualization Development Environments 2000*. Princeton Plasma Physics Laboratory, Princeton University, New Jersey, USA, 2000. http://w3.pppl.gov/vde2000/eproceedings/P02kraemer.PDF.

[KG90] M. Kessel and P. Gruss. Murine development control genes. *Science*, 249:374–379, 1990.

[KHKL98] H. Kitano, S. Hamahashi, J. Kitazawa, and S. Luke. The perfect c. elegans project: An initial report. *Sony Computer Science Laboratory*, 1998.

[Kle72] F. Klein. *Das Erlanger Programm: Vergleichende Betrachtung über neuere geometrische Forschungen*. Akademischer VerlagsGesellschaft Greest und Portig, 1872. Reprint Leipzig 1974.

[MMYY95] MA. Marek, SC. Müller, T. Yamaguchi, and K. Yoshikawa. *Dynamism and Regulation in Nonlinear Chemical Systems*. North-Holland,Amsterdam, 1995.

[Moe89] J. Moeller. Random tesselations in $\mathbb{R}^d$. *Adv. Appl: Prob.*, 21:37–73, 1989.

[MP69] M.I. Minsky and S. Papert. *Perceptrons*. MIT Press Mass., 1969.

[MTL00] G. Medioni, C.-K. Tang, and M.-S. Lee. Tensor voting : Theory and applications. *12eme Congres Francophone AFRIF-AFIA ...(RFIA)*, February 2000.

[Mun84] James R. Munkres. *Elements of Algrebraic Topology*. Addison-Wesley, Menlo Park, 1984.

[OD71] G.F. Oster and C. A. Desoer. Tellegen's theorem and thermodynamic inequalities. *Journal of Theoretical Biology*, 32:219–241, 1971.

[Ons31] L. Onsager. Reciprocal relation in irreversible process. *Phys. Rev.*, 32:405–426, 1931.

[Pau93] W.E. Paul. *Fundamental Immunology*. Raven Press, 1993.

[PH86] J. Pries and D. Hirsh. Caenorhabditis elegans morphogenesis: The role of the cytoskeleton in the elongation of the embryo. *Dev. Biol.*, 117(156-173), 1986.

[Plaay] S. Plattner. Rural market networks. *Scientific American*, 232:66–79, 1975 May.

[PS88] F. Preparata and M. Skamos. Computational geometry. *Springer N.Y.*, 1988.

[Sac02] L. Sachs. *Angewandte Statistik*. Springer N.Y., 2002.

[Sch] R. Schnabel. Schnabel laboratory, technical university braunschweig, germany. personal gift.

[Sch78] E. Schierenberg. *Die embryonalentwicklung des Nematoden Caenorhabditis elegans als Modell*. PhD thesis, Universitt Goettingen, 1978.

[Sch89] E. Schierenberg. Cytoplasmic determination and distribution of developmental potential in the embryo of caenorhabditis elegans. *BioEssays*, 10:99–104, 1989.

[Se89] Joseph Sambrook and et.al. *Molecular Cloning*. Cold Spring Harbor Laboratory Press, 1989. appendix A1.

[SH77] J.E. Sulston and H. R. Horvitz. Postembryonic cell lineage of the nematode caenorhabditis elegans. *Develpmental Biology*, 56:110–156, 1977.

[SH81] J.E. Sulston and H.R. Horvitz. Abnormal cell lineages in mutants of the nematode caenorhabditis elegans. *Dev. Biol.*, 56:110–156, 1981.

[SHMS97] R. Schnabel, H. Hutter, D. Moermann, and H. Schnabel. Accessing normal embryogenesis in caenorhabditis elegans using a 4d microscope: Variability of development and regional specification. *Developmental Biology*, 184:234–265, 1997. http://www.TU-BS.de/institute/genetik/schnabel/ce-home.

[Sla83] J.M.W. Slack. *From egg to embryo*. Number ISBN 0 521 24205 3. Cambridge University Press, 1983.

[SLJ+93] GM. Shackleford, Shivakumar S.and Shiue L., Mason J., Kenyon C., and Varmus HE. Two wnt genes in caenorhabditis elegans. *Oncogene 1993 Jul;8(7):1857-64*, 8(7):1857–64, Jul 1993. http://www.stanford.edu/ rnusse/wntgenes/eleganswnt.html.

[SML98] W. Schroeder, K. Martin, and B. Lorensen. *The visualization toolkit vtk*. Number ISBN: 0-13-954694-4. Prentice Hall PTR, 1998.

[SSL00] F. Simon, F. Steinbrckner, and C. Lewerentz. 3d-spring embedder for complete graphs. *Computer Science Reports*, 2000.

[SSWT83] J. E. Sulston, E Schierenberg, J G White, and J.N. Thomson. The embryonic cell lineage in the nematode caenorhabditis elegans. *Developmental Biology*, 100:64–119, 1983.

[Str] G. Struckhoff. Glia cell movie. Cell division of glia cells; personal communications.

[TM01] D. Tal and J. Malik. Combining color, texture and contour cues for image segmentation. Technical report, Computer Science Division University of California, Berkeley, 2001. http://www.cs.berkeley.edu/ doron/.

[Wat81] D.F. Watson. Computing the n-dimensional delaunay tesselation with application to voronoi polytops. *The Computer J.*, 24:167–172, 1981.

[WC02] N. Wittenburg and B. Conrad. Tod oder lebendig. studien zum programmierten zelltod in c. elegans. *BIOspektrum*, Nr.2-8(ISSN:0947-0867), 2 2002.

[WHC80] W.B. Wood, R. Hecht, and N. Carr, S. Vanderslice. Parental effects and phenotypic characterization in caenorhabditis elegans. *Dev. Biol.*, 74:446–469, 1980.

[WM99] Qiang Wu and Tom Maniatis. A striking organization of a large family of human neural cadherin-like cell adhesion genes. *Cell*, 97:779–790, 1999.

[WMSS01] R. Winklbauer, A. Medina, RK. Swain, and H. Steinbeisser. Frizzled-7 signalling controls tissue separation during xenopus gastrulation. *Nature*, 413(6858):856–60, Oct 25 2001.

[Woo88] W.B. Wood. *The nematode Caenorhabditis Elegans*. Number ISBN 0-87969-307-X. Cold Springer Harbour Laboratory, 1988.

[zS06] O. zur Strassen. *Die Geschichte der T-Riesen von Ascaris megalocephala*. E. Schweitzerbartsche Verlagsbuchhandlung (E. Nägele), Stuttgart, 1906.

Appendix H

Summary

The pre-morpho-genetic development of nematode *Caenorhabditis elegans* in the egg is studied to formulate a base system, in which rules that govern the formation of shape and functions in biological organisms can be defined. The development in the pre-morpho-genetic phase in the egg is recorded by a microscope through which the egg is scanned as a stack of 2D images. The 3D image stack for a definite time of the development is analyzed for the spatial positions of the nuclei. A browser of the computer-aided analysis of a database (see [Incb]) supports the data-retrieval and offers different views, e.g. as a cell lineage tree, as the spatial distribution of cells colored according to their sub-lineages or fates or as a movie of development. Also, the preparation of the egg from the worm C. elegans is displayed in movies.

A segmentation program "ncuts" operates on the images to partition the image data into meaning-full objects (i.e. nuclei).

Computer programs are developed to extract data from the database for processing in the environments of languages like C++,MATLAB, VTK, Tcl/Tk and others.

Since the cell cycle has no exact periods, the times of cell division are analyzed by cluster analysis and the division times in a cluster are replaced by the cluster

mean time values. Thus, groups of dividing cells are obtained and only about 40 different time values are needed to generate the lineage tree. The same procedure delivers mean values of the spatial distances of nuclei for each stage. This reduces the necessary parameters stored in the database further. The development is represented in four dimensional (4D) space with spatial and temporal coordinates. In the 4D space, the whole development is considered at once. Cells in this space fill a 4D volume giving the real 3D volume and the life span of the cell. By assigning the center of the 4D cell to the weighted position of its nucleus in time, a 4D-net of 4D cell-nuclei is obtained, of which the distances between the 4D nuclei could be treated as in the case of the 3D space. The cell contacts to other cells is described in an adjacency matrix. For stages with higher cell numbers, the experimental evaluation of the adjacency is replaced by the Delaunay algorithm, which determines the nearest neighbors according to a triangulation method and is a good first approximation to nature. The algebra behind the algorithm leads to the chain concept of vector spaces, by which a net of cell contacts is split into a topological part (the connectivity) and a metric part (determining the real geometric shape). The topology inherent in the vector space leads to the boundary operators, of which the matrices are evaluated and normalized to a TDO format. The relation to the concept of forces and fluxes in thermodynamic network systems is discussed in the context of development.