UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Geometric Modeling, Simulation, and Visualization Methods for Plasmid DNA Molecules

Adriano Nunes Raposo

Tese para obtenção do Grau de Doutor em
Engenharia Informática
(3º ciclo de estudos)

Orientador: Prof. Doutor Abel João Padrão Gomes

Covilhã, março de 2015

To my family

Acknowledgements

This dissertation is not just a collection of pages containing fancy words and pretty figures, it is the final destination of a very long, and hard, journey. As all journeys, it had its ups and its downs. High mountains that, at first sight, seemed impossible to climb, long deserts that seemed impossible to cross, and some of the deepest swamps that one can find. Fortunately, at the top of each mountain there was a moment of joy, and at each oasis in the middle of a desert there was a feeling of accomplishment. It was, most of the time, a lonely journey, but it had been impossible to fulfil without the support of some very important people.

Foremost, I want to express my endless gratitude to my supervisor, Prof. Abel Gomes. Prof. Abel is, not only my supervisor and mentor, but also a dear friend. This dissertation would not exist without his support. Every time I thought about quitting, and it happened more than once, he convinced me to continue. I also want to thank Prof. Abel for his efforts and commitment in the process of writing the papers that were published, and submitted for publication, along this dissertation. I hope we can work together in the future.

I also want to thank Prof. João Queiroz for the original idea that led to the work presented in this thesis. I sincerely want to thank him for his support, specially in the early stages of this journey and in the scientific revision of the publications.

I am also grateful to the contrary forces that, here and there, tried to difficult my journey. As the great philosopher Friedrich Nietzsche once said: "*That which does not kill us makes us stronger*". Like in the Japanese myth about how koi become dragons, "demons" laughed at my efforts and sadistically increased the height of the waterfalls every time I tried to make a leap to the top. I feel strong enough now, and this dissertation is my final leap to the top of the waterfall.

I want to thank to my closest family for their unconditional support along these years. But above all, I want to apologize to my wife Alexandra and to my daughter Eduarda for not paying them enough attention when I was working on this thesis.

Finally, I want to thank God.

Resumo

Plasmídeos são um tipo especial de moléculas de ADN usadas, entre outras aplicações, em vacinas de ADN e em terapia génica. Este tipo de moléculas de ADN caracteriza-se por se apresentar no seu estado natural numa conformação circular fechada e super-enrolada. A produção de ADN plasmídeo com recurso a bactérias hospedeiras implica um processo de purificação cujo objectivo é separar as moléculas de plasmídeo do ADN da bactéria hospedeira e de outros contaminantes. Este processo de purificação, e todas as alterações físico-químicas envolvidas, tais como variações de temperatura, podem originar alterações na conformação das móleculas de plasmídeo, desenrolando-as ou até mesmo fazendo-as assumir uma conformação linear aberta, o que as torna inviáveis para aplicação terapêutica. Por este motivo, os investigadores procuram novas técnicas de purificação que maximizem a quantidade de ADN plasmídeo obtida na sua conformação super-enrolada. As simulações em computador e a visualização em 3D de moléculas de ADN plasmídeo podem trazer muitas vantagens porque permitem aos investigadores prever o que poderá acontecer a determinadas moléculas deste tipo quando sujeitas a condições específicas. Assim, foi necessário desenvolver modelos geométricos fiáveis e precisos especificamente para moléculas de ADN plasmídeo. Esta dissertação apresenta um novo algoritmo desenvolvido especificamente para a construção em 3D de moléculas de ADN plasmídeo. Este novo algoritmo é totalmente adaptativo no sentido em que permite aos investigadores construir moléculas de ADN com qualquer sequência de pares de bases sobre qualquer conformação arbitrária, real ou teórica, desde que o seu comprimento seja compatível com a sequência da molécula de ADN plasmídeo. Esta capacidade é especialmente útil para simulações de ADN plasmídeo que geram conformações sobre as quais é posteriormente necessário empilhar a sequência de pares de bases da molécula, o que não é possível utilizando os convencionais métodos predictivos de construção de moléculas ADN. Ao contrário dos métodos tradicionais de visualização molecular baseados na estrutura atómica das moléculas, este novo algoritmo utiliza as superfícies moleculares tridimensionais dos nucleótidos, com uma cor diferente para cada tipo de nucleótido, como peças de construção básicas. Esta nova abordagem não só reduz a quantidade de objectos gráficos e, consequentemente, acelera a renderização, mas também torna mais fácil a identificação visual dos nucleótidos nas cadeias de ADN. O algoritmo usado para triangular as malhas das superfícies moleculares dos nucleótidos também é apresentado como novidade nesta dissertação. Este novo algoritmo para triangulação de superfícies implícitas moleculares introduz um novo mecanismo de divisão espacial da estrutura atómica em regiões de influência, acelerando assim a triangulação uma vez que não tem em conta os átomos que não influenciam a construção da superficie molecular na região em causa. Este novo método garante a continuidade da superfície molecular. Sendo o objectivo desta dissertação apresentar um conjunto completo de ferramentas integradas para visualização e simulação de ADN plasmídeo, também é apresentado um novo algoritmo de deformação da conformação deste tipo de moléculas para ser utilizado em simulações baseadas em métodos de simulação de Monte Carlo. Este novo algoritmo de deformação utiliza uma polilinha tridimensional para representar a conformação do ADN plasmídeo e efectua pequenas deformações nessa mesma polilinha, mantendo o comprimento e conectividade dos segmentos de recta que a compõem. As experiência realizadas com o objectivo de comparar este novo método de deformação com os métodos tradicionalmente usados em simulações deste tipo mostraram que o novo método é mais eficiente, isto é, a taxa de aceitação das suas conformações é mais elevada e converge mais rapidamente para o estado

de equilibrio relativamente à energia elástica do ADN plasmídeo. Em suma, esta dissertação apresenta um conjunto completo de modelos e algoritmos para modelação geométrica e simulação de ADN plasmídeo.

Palavras-chave

Modelação geométrica de ADN
Simulação de ADN plasmídeo
Superfícies moleculares Gaussianas
Empilhamento adaptativo de ADN
Deformação de ADN plasmídeo

Abstract

Plasmid DNA molecules are a special type of DNA molecules that are used, among other applications, in DNA vaccination and gene therapy. These molecules are characterized by, when in their natural state, presenting a closed-circular conformation and by being supercoiled. The production of plasmid DNA using bacteria as hosts implies a purification step where the plasmid DNA molecules are separated from the DNA of the host and other contaminants. This purification process, and all the physical and chemical variations involved, such as temperature changes, may affect the plasmid DNA molecules conformation by uncoiling or even by open them, which makes them useless for therapeutic applications. Because of that, researchers are always searching for new purification techniques that maximize the amount of supercoiled plasmid DNA that is produced. Computer simulations and 3D visualization of plasmid DNA can bring many advantages because they allow researchers to actually see what can happen to the molecules under certain conditions. In this sense, it was necessary to develop reliable and accurate geometric models specific for plasmid DNA simulations. This dissertation presents a new assembling algorithm for B-DNA specifically developed for plasmid DNA assembling. This new assembling algorithm is completely adaptive in the sense that it allows researchers to assemble any plasmid DNA base-pair sequence along any arbitrary conformation that fits the length of the plasmid DNA molecule. This is specially suitable for plasmid DNA simulations, where conformations are generated by simulation procedures and there is the need to assemble the given base-pair sequence over that conformation, what can not be done by conventional predictive DNA assembling methods. Unlike traditional molecular visualization methods that are based on the atomic structure, this new assembling algorithm uses color coded 3D molecular surfaces of the nucleotides as the building blocks for DNA assembling. This new approach, not only reduces the amount of graphical objects and, consequently, makes the rendering faster, but also makes it easier to visually identify the nucleotides in the DNA strands. The algorithm used to triangulate the molecular surfaces of the nucleotides building blocks is also a novelty presented as part of this dissertation. This new triangulation algorithm for Gaussian molecular surfaces introduces a new mechanism that divides the atomic structure of molecules into boxes and spheres. This new space division method is faster because it confines the local calculation of the molecular surface to a specific region of influence of the atomic structure, not taking into account atoms that do not influence the triangulation of the molecular surface in that region. This new method also guarantees the continuity of the molecular surface. Having in mind that the aim of this dissertation is to present a complete set of methods for plasmid DNA visualization and simulation, it is also proposed a new deformation algorithm to be used for plasmid DNA Monte Carlo simulations. This new deformation algorithm uses a 3D polyline to represent the plasmid DNA conformation and performs small deformations on that polyline, keeping the segments length and connectivity. Experiments have been performed in order to compare this new deformation method with deformation methods traditionally used by Monte Carlo plasmid DNA simulations These experiments shown that the new method is more efficient in the sense that its trial acceptance ratio is higher and it converges sooner and faster to the elastic energy equilibrium state of the plasmid DNA molecule. In sum, this dissertation successfully presents an *end-to-end* set of models and algorithms for plasmid DNA geometric modelling, visualization and simulation.

Keywords

DNA geometric modelling
Plasmid DNA simulation
Gaussian molecular surfaces
Adaptive DNA stacking
Plasmid DNA deformation

Resumo Alargado

Introdução

Desde os seus primórdios, os computadores têm sido usados para modelar, manipular, simular e visualizar moléculas. Desde cálculos simples com pequenas moléculas, até aos mais recentes avanços em simulação 3D de macromoléculas ou na análise de sequências de ADN, os algoritmos e métodos computacionais têm desempenhado um importante papel nas áreas da biologia molecular, bioquímica e biotecnologia. Estes algoritmos e métodos computacionais aplicados à biologia deram origem a áreas de investigação como a biologia computacional, química computacional e bioinformática.

A faceta mais conhecida da bioinformática é, provavelmente, a sequenciação e análise de ADN e o reconhecimento de padrões genéticos. Importa, no entanto, referir que a bioinformática abrange outros ramos. Um desses ramos dedica-se ao estudo de algoritmos e métodos matemáticos e computacionais usados na modelação geométrica tridimensional de átomos, moléculas, genes e cromossomas. Este ramo da bioinformática é vulgarmente conhecido como modelação molecular (ou *molecular modeling* ou *graphics-based molecular modeling* ou, ainda, *molecular graphics*).

Um dos objetivos da modelação molecular é possibilitar aos investigadores visualizarem e manipularem representações tridimensionais das moléculas. A computação gráfica, em particular a computação geométrica, fornece as ferramentas geométricas necessárias para visualizar e interagir com modelos moleculares através de um computador, sendo as superfícies 3D uma das ferramentas geométricas mais usadas na modelação e visualização molecular. De uma maneira geral, e quanto à sua formulação matemática, as superfícies moleculares tanto podem ser definidas de forma paramétrica como de forma implícita. Esta dissertação, além de apresentar resumidamente os modelos de superfícies moleculares mais populares na literatura, apresenta uma adaptação de um algoritmo de continuação para superfícies implícitas de uso genérico [RG06] como um novo método para construção de superfícies moleculares Gaussianas [RQG09].

A modelação molecular tem como objetivo modelar todos os tipos de moléculas, desde as pequenas com algumas dezenas de átomos, até macromoléculas com milhares de átomos como, por exemplo, moléculas de ADN. As moléculas de ADN são um tipo especial de moléculas que estão presentes nas células de todos os seres vivos (à exceção dos vírus), sendo bastante conhecida a sua longa e estreita estrutura atómica em forma de dupla hélice [WC53]. E é exatamente em relação à sua estrutura que, de uma forma simplificada, podemos dizer que o ADN pode ser decomposto em duas cadeias. Cada uma dessas cadeias é composta por uma sequência de nucleótidos (bases) ligados por uma "coluna vertebral" de açúcares e fosfatos. Nos tipos de ADN mais comuns podemos encontrar quatro tipos de nucleótidos, nomeadamente: adenina (A); citosina (C); guanina (G); e tiamina (T). Sabendo que cada nucleótido de uma cadeia de ADN está emparelhado de forma unívoca com um nucleótido da cadeia complementar (A sempre com T, e C sempre com G), é possível codificar o ADN usando apenas a sequência de nucleótidos de uma das suas cadeias, sequência essa a que vulgarmente se dá o nome de *sequência de pares de bases*.

Devido à existência de todas estas particularidades na estrutura do ADN houve a necessidade de desenvolver modelos e métodos computacionais específicos para este tipo de macromoléculas, especialmente para a construção de moléculas de ADN a partir de sequências de pares de bases sem conhecer explicitamente a sua estrutura atómica, que é a abordagem usada pela maior parte dos modelos moleculares computacionais.

Neste dissertação, não só são apresentados os modelos convencionais para a assemblagem de sequências de ADN, mas também se apresenta um algoritmo de assemblagem de ADN completamente novo [RG12], em particular para ADN da forma B (*B-DNA*). Ao contrário dos métodos mais populares que tentam prever a conformação, isto é, a trajetória das moléculas de ADN a partir da sua sequência de bases, este novo algoritmo consegue adaptar uma sequência de ADN a qualquer conformação arbitrária. Importa referir que este novo algoritmo foi desenvolvido especificamente para ser usado em simulação de ADN, situação na qual a sequência de bases tem de ser assemblada sobre uma conformação previamente existente normalmente representada por uma aproximação a uma curva tridimensional. Importa ainda referir que este novo algoritmo utiliza um conjunto de quatro peças de assemblagem tridimensionais realistas, em representação de cada um dos quatro nucleótidos, e que a superfície de cada uma destas peças tridimensionais foi construída utilizado o novo algoritmo de triangulação de superfícies moleculares que também é apresentado nesta dissertação [RQG09].

As moléculas de *ADN plasmídeo* (*pDNA*), ou simplesmente *plasmídeos*, são um tipo de moléculas usadas em vacinação de ADN, terapia génica ou biofármacos recombinantes. São moléculas de ADN conhecidas por se apresentarem numa conformação circular fechada, ou seja, o primeiro par de bases da dupla hélice está ligado ao último, formando uma estrutura em lacete.

Tanto as experiências de produção de plasmídeos em laboratório como as experiências de purificação, procedimento que é usado para separar os plasmídeos que se pretendem obter do ADN do hospedeiro e de outros contaminantes, têm elevados custos e consomem bastante tempo. A utilização de computadores para simular *in silico* procedimentos laboratoriais pode trazer muitos benefícios. As condições físico-químicas das experiências laboratoriais, tais como temperatura e concentração de sal usadas durante o processo de purificação, podem ser replicadas virtualmente usando modelos matemáticos e algoritmos de simulação adequados.

Esta dissertação foca-se essencialmente nos métodos de simulação de Monte Carlo, provavelmente o tipo de simulação mais utilizado quando se trata de plasmídeos. No caso particular dos métodos de simulação de Monte Carlo para plasmídeos, baseiam-se em estatísticas e em cálculos de energia elástica dos plasmídeos para aceitar ou rejeitar as conformações que vão sendo geradas a cada iteração do algoritmo. Basicamente, o que este tipo de algoritmos faz é deformar uma dada conformação, o que origina uma outra conformação. Esta nova conformação é aceite se minimizar a energia elástica do plasmídeo, isto é, se a sua energia elástica for menor que a conformação anterior ou, se mesmo que isso não aconteça, se existir uma determinada probabilidade de ocorrência dessa mesma conformação.

Nesta dissertação apresenta-se um novo algoritmo de deformação conformacional de plasmídeos para utilização em métodos de simulação de Monte Carlo. As técnicas utilizadas na deformação das conformações têm, no essencial e salvo pequenos melhoramentos, sido as mesmas ao longo dos anos. Com base nos resultados experimentais apresentados nesta dissertação, podemos afirmar que o novo algoritmo de deformação tem várias vantagens e apresenta um desempenho mais eficiente quando comparado com as técnicas de deformação tradicionais. Além disso, importa ainda referir que este novo algoritmo de deformação é de aplicação genérica, ou seja,

não serve apenas para deformação de plasmídeos, mas para deformação aleatória de qualquer curva tridimensional que seja aproximada por uma sequência de segmentos de recta.

De uma maneira geral, esta dissertação pretende contribuir para a otimização dos métodos de modelação e simulação de ADN plasmídeo. Para isso, começa-se por apresentar o *estado da arte* da modelação molecular de ADN, dando especial atenção aos modelos geométricos e de simulação específicos para plasmídeos. Depois, segue-se a parte nuclear da dissertação, a qual inclui um conjunto de novas ferramentas totalmente integráveis, nomeadamente: um novo algoritmo de triangulação suave de superfícies moleculares definidas de forma implícita; um novo algoritmo de assemblagem tridimensional de ADN a partir das suas sequências de bases e adaptável a qualquer conformação arbitrária; e, finalmente, um novo e eficiente algoritmo para deformação de plasmídeos para simulações que utilizem métodos de Monte Carlo.

Trajeto dos Trabalhos de Tese

A principal ideia que esteve na origem desta dissertação foi a possibilidade de visualizar num computador o comportamento conformacional do enrolamento/desenrolamento de moléculas de ADN plasmídeo durante experiências laboratoriais, especialmente durante processos de purificação aos quais este tipo de moléculas é sujeito após a sua produção. Partindo do princípio que as moléculas de ADN plasmídeo devem manter a sua conformação natural super-enrolada para poderem ser eficazes quando aplicadas a vacinação de ADN e a terapia génica, e sabendo que alterações às condições físico-químicas a que são submetidas durante a purificação podem afectar a sua conformação final, a possibilidade de visualizar o processo em computador pode trazer vantagens para esta área de investigação.

O primeiro passo foi estudar os modelos moleculares genéricos já existentes, e determinar em que medida podiam ser usados na modelação de ADN. Foi dada especial atenção aos modelos de superfícies moleculares tradicionalmente usados em modelação molecular, nomeadamente: superfícies de van der Walls, *solvent accessible surface* (SAS), *solvent excluded surface* (SES) e superfícies moleculares Gaussianas. Assim, depois de constatar que os algoritmos existentes para triangulação de superfícies moleculares Gaussianas pertenciam na sua esmagadora maioria à categoria de *marching cubes*, o que os tornava inadequados para triangular parcial or totalmente móleculas com grandes quantidades de átomos de forma dinâmica, como é o caso das móleculas de ADN, surgiu a ideia de adaptar um algoritmo baseado no princípio da continuação para a triangulação de superfícies moleculares Gaussianas. Tendo em consideração que é feita uma divisão espacial da moléculas em regiões específicas, este algoritmo pode, se tal for necessário, ser usado para construir apenas pequenas porções da superfície molecular. Isto pode traduzir-se numa vantagem quando se trata da triangulação de grandes moléculas de ADN.

No entanto, a partir de um certo ponto tornou-se claro que os modelos moleculares tradicionais, especialmente os modelos de superfícies moleculares, têm várias limitações, em particular no que diz respeito a triangulações. Na verdade, triangular a superfície de uma molécula de ADN com uma grande quantidade de átomos, como é o caso do ADN plasmídeo, coloca uma série de constrangimentos quando se é obrigado a re-triangular a superfície em consequência da sua deformação conformacional.

Mas subsistia outro problema, que resultava do facto de a estrutura atómica dos plasmídeos não estar explicitamente disponível nas várias bases de dados internacionais (e.g., Protein Data

Bank ou PDB), nas quais estas moléculas de ADN são normalmente descritas pela sua sequência de pares de bases em ficheiros de formato GBK (GenBank). A ideia seguinte foi passar para outra escala de abstração, mais concretamente do nível do átomo para o do nucleótido. Historicamente, esta tem sido a abordagem tradicional quando se trata de moléculas de ADN. No entanto, os métodos tradicionais não usam uma representação consentânea ou próxima da realidade, pelo que surgiu a ideia de utilizar as superfícies moleculares dos quatro nucleótidos essenciais como peças para a assemblagem de ADN. Além disso, ao contrário dos algoritmos mais populares que tentam prever a conformação das moléculas de ADN, este novo algoritmo permite empilhar ou assemblar pares de nucleóbtidos de ADN ao longo de uma qualquer conformação arbitrária a partir de uma sequência textual de bases, o que faz com que este método seja ideal para cenários de simulação conformacional de ADN plasmídeos.

O passo final para alcançar o objectivo último desta tese foi efectuar simulações de ADN plasmídeo. Foi adoptado o método de simulação de Monte Carlo por ser, historicamente, o mais usado em simulação de plasmídeos. No entanto, os métodos usados tradicionalmente para a deformação das conformações podem gerar deslocamentos súbitos de partes significativas da molécula de uma só vez, o que não reflete a realidade, não sendo por isso adequados para a visualização em tempo real do processo de simulação. Assim, foi desenvolvido um novo método de deformação para plasmídeos que gera deformações mais suaves e realistas das conformações ao longo do tempo. Por fim, várias experiências foram realizadas de maneira a verificar a eficiência do novo método quando comparado com os métodos usados tradicionalmente em simulações de plasmídeos através de métodos de Monte Carlo. Foi possível concluir que o novo algoritmo provoca deformações mais graduais no espaço e no tempo, convergindo mesmo assim mais rapidamente para o ponto de equilíbrio energético do plasmídeo.

Principais Contribuições

Em resultado da investigação que conduziu à presente tese, eis as suas principais contribuições:

- Nesta dissertação propõe-se a adaptação de um algoritmo de triangulação genérico de superfícies implícitas para a construção de malhas trianguladas enquanto aproximações de superfícies moleculares Gaussianas. Uma das novidades deste algoritmo é que não necessita de ter em conta a estrutura atómica completa da molécula. Em vez disso, este novo algoritmo de triangulação divide o domínio que encerra a molécula em regiões denominadas caixas de influência, com as suas correspondentes esferas de influência, e constrói a malha da superfície molecular dentro de cada caixa de influência de forma independente relativamente às caixas de influência vizinhas, ou seja, desprezando a contribuição dos átomos que se encontram fora da mesma caixa de influência. Isto significa que a triangulação da superfície de qualquer molécula de ADN não depende do seu tamanho, ou seja, não depende da quantidade total de átomos da molécula mas apenas dos átomos que estejam numa determinada região. A conectividade e continuidade da superfície molecular é garantida por um mecanismo de mistura de funções Gaussianas locais na intersecção das esferas de influência vizinhas.

- Partindo do princípio que o ADN pode ser decomposto em submoléculas, designadas por nucleótidos, é proposto um algoritmo adaptativo para assemblagem de ADN que utiliza as

superfícies moleculares Gaussianas dos nucleótidos como peças de montagem. Este algoritmo pode ser utilizado em procedimentos de simulação de plasmídeos, nomeadamente quando são utilizados métodos de Monte Carlo. A representação do eixo do ADN como um esqueleto tridimensional arbitrário, e a atribuição de um par de bases a cada um dos segmentos desse esqueleto, permite posicionar cada par de nucleótidos utilizando distâncias e ângulos específicos. A maioria dos métodos de assemblagem de ADN são preditivos, isto é, dada uma sequência de pares de bases tentam prever a conformação da respectiva molécula de ADN. Sendo completamente adaptativo, este novo método é capaz de adaptar qualquer sequência de pares de bases a uma conformação arbitrária. Além disso, a utilização das superfícies moleculares dos nucleótidos como peças de montagem confere uma aparência mais realista às moléculas. Finalmente, visto que se baseia numa abstração ao nível dos nucleótidos, é bastante menos onerosa em termos computacionais, no que respeita à utilização de recursos de memória e de processador.

- Esta dissertação também propõe um novo e eficiente algoritmo para deformação de plasmídeos em simulações de Monte Carlo. Quando comparado com os métodos de deformação tradicionalmente usados em simulações de Monte Carlo de plasmídeos, pode dizer-se que este novo método faz deformações mais graduais em termos de espaço e tempo. No entanto, a deformação acumulada é maior ao longo do tempo quando comparada com a deformação acumulada dos métodos tradicionais. Assim, podemos concluir que o novo método é mais eficiente. A eficiência deste método de deformação pode ser medida pela taxa de aceitação do método de Monte Carlo relativamente às conformações produzidas pelo método de deformação, taxa essa que veio a revelar-se mais elevada no caso do novo algoritmo de deformação, quando comparado com os métodos de deformação mais populares. Além disso, também converge mais rapidamente, ou seja, em menos iterações, para o ponto de equilíbrio da energia elástica dos plasmídeos. Importa ainda referir que este algoritmo pode ser usado como um algoritmo de deformação genérico de modelos geométricos baseados em polilinhas com segmentos do mesmo tamanho.

- Utilizando os novos métodos de assemblagem e deformação propostos nesta teste, é possível visualizar a simulação de plasmídeos em tempo real e a três dimensões em qualquer computador pessoal. A utilização dos nucleótidos como peças elementares de assemblagem, em vez dos átomos, reduz o volume de dados e a quantidade de objectos nas cenas, com resultados visuais semelhantes ou, em alguns casos, até melhores uma vez que os nucleótidos são visualmente identificáveis pelo seu código de cores. Além disso, a utilização de um algoritmo de deformação com uma taxa de aceitação de conformações mais elevada e com deformações mais locais e progressivas, acaba por gerar uma animação mais consistente durante a simulação, sem longos períodos em que nada acontece, e em que não há deslocamentos grandes e repentinos de grandes porções dos plasmídeos, como acontece quando se usam os métodos de deformação convencionais.

No entanto, a maior contribuição desta tese é, provavelmente, que o resultado é maior do que a soma de todas as contribuições, uma vez que apresenta um conjunto completo de métodos para modelação geométrica, simulação e visualização de plasmídeos. Todo o trabalho apresentado é articulado ao longo da tese no sentido em que cada método é fundamental para o funcionamento do método seguinte. O novo algoritmo para triangulação de superfícies moleculares é usado para gerar as peças de montagem usadas pelo algoritmo de assemblagem de ADN. Por sua vez o algoritmo de assemblagem de ADN é usado pelo algoritmo de deformação para visualização em

tempo real dos processos de simulação.

Publicações

O trabalho de investigação que conduziu a esta tese de doutoramento deu origem às seguintes publicações:

1. Adriano N. Raposo and Abel J. P. Gomes: *Polygonization of multi-component non-manifold implicit surfaces through a symbolic-numerical continuation algorithm*. In Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and South-East Asia (GRAPHITE'06), Kuala Lumpur, Malaysia, November 29 - December 2, ACM Press, pp. 399-406, 2006.

2. Adriano N. Raposo, João A. Queiroz, and Abel J. P. Gomes: *Triangulation of Molecular Surfaces Using an Isosurface Continuation Algorithm*. In Proceedings of the International Conference on Computational Science and Its Applications (ICCSA'09), Yongin, Korea, June 29 - July 2, IEEE Computer Society Press, pp. 145-153, 2009.

3. Adriano N. Raposo and Abel J. P. Gomes: 3D molecular assembling of B-DNA sequences using nucleotides as building blocks. *Graphical Models* **74**(4): 244-254 (2012). ISI Web of Knowledge: IF = 0.967 (2013); Q2 (Computer Science, Software Engineering).

4. Adriano N. Raposo and Abel J. P. Gomes: Efficient deformation algorithm for plasmid DNA simulations, *BMC Bioinformatics* **15**(301), 2014.ISI Web of Knowledge: IF = 2.672 (2013); Q1 (Mathematical & Computational Biology).

5. Adriano N. Raposo and Abel J. P. Gomes: isDNA: A Tool for Real-Time Visualization of Plasmid DNA Monte-Carlo Simulations in 3D, *Springer Lecture Notes in Bioinformatics* **9044 Part II**(566-577), 2015.

6. Adriano N. Raposo, Abel J. P. Gomes: Computational 3D Stacking Methods for DNA: a Survey (aceite provisoriamente para publicação na revista *IEEE/ACM Transactions on Computational Biology and Bioinformatics*).

Limitações da Investigação

Esta secção apresenta algumas das limitações encontradas no decorrer do trabalho de investigação. O objectivo é esclarecer se os métodos e técnicas propostas nesta dissertação podem, ou não, dar resposta a outros problemas específicos em modelação molecular ou em domínios de conhecimento afins. É nosso entendimento que indicar as limitações deste trabalho pode contribuir para uma discussão mais alargada da qual podem resultar melhoramentos em relação às soluções propostas.

Assim, algumas das limitações do algoritmo de triangulação de superfícies moleculares Gaussianas são:

- A abordagem de dividir-para-conquistar poderá não ser a melhor opção para moléculas com elevada densidade atómica. O tamanho constante das caixas e esferas de influência torna o algoritmo mais lento em regiões com elevada densidade atómica, isto é, regiões em que existe uma grande quantidade de átomos por unidade de volume, como pode ser o caso de algumas proteínas.

- Embora este algoritmo de triangulação distribua os átomos de uma molécula por caixas e esferas de influência, importa esclarecer que não se trata de um algoritmo de partição espacial, mas sim de um algoritmo de continuação. Assim, no caso de existir uma elevada concentração de átomos por unidade de volume, poderá dar-se o caso de furos toroidais e outras cavidades de uma dada molécula deixarem de ser visíveis ou identificáveis, o que resulta da mistura excessiva de funções Gaussianas associadas aos referidos átomos.

Por sua vez, algumas das possíveis limitações do algoritmo de assemblagem de ADN poderão ser:

- Uma vez que se trata de um algoritmo de assemblagem adaptativa de ADN, no sentido de que permite a construção 3D de uma sequência de pares de bases sobre qualquer conformação arbitrária, pode acontecer que se usem conformações com ângulos e distâncias pouco convencionais entre alguns nucleótidos. Importa no entanto referir que a ideia de utilizar as superfícies moleculares dos nucleótidos como peças elementares na construção tridimensional de moléculas de ADN pode também ser usada pelos métodos preditivos convencionais.

- O algoritmo proposto nesta tese para a assemblagem de ADN é adequado apenas para ADN do tipo *B* (B-DNA), a forma mais comum em plasmídeos. Isto significa que este algoritmo, na forma como é proposto nesta tese, não é adequado à construção de moléculas de ADN de outros tipos como, por exemplo, A (A-DNA) ou Z (Z-DNA). No entanto este algoritmo poderá ser adaptado a estes outros tipos de ADN desde que se ajustem os parâmetros geométricos utilizados.

Finalmente, algumas das possíveis limitações do algoritmo de deformação de plasmídeos para simulações de Monte Carlo são:

- Ao fim de algumas centenas de iterações poderão surgir conformações de plasmídeos com dobras bastante acentuadas (*kinks*, do inglês), o que confere uma aparência menos natural às conformações, o que é pouco provável em experiências laboratoriais. O algoritmo não inclui um mecanismo de suavização destas dobras porque o objectivo era comparar o seu desempenho com os métodos convencionais sem influenciar os resultados finais, uma vez que um mecanismo de suavização iria alterar a energia elástica dos plasmídeos.

- Relativamente às condições físico-químicas, o método de Monte Carlo utilizado apenas tem em consideração alterações na temperatura. Nas experiências realizadas não foram tidas em conta outras variáveis como, por exemplo, a concentração de sal, como é usual em experiências laboratoriais de purificação de plasmídeos.

- Ao contrário do que acontece nas experiências laboratoriais, em que os investigadores manipulam muitas moléculas ao mesmo tempo, todas as experiências de simulação realizadas no contexto deste trabalho de dissertação usaram apenas uma molécula de cada vez.

O autor acredita que algumas das limitações apontadas poderão ser ultrapassadas em trabalhos futuros, como se indica na secção seguinte.

Trabalho Futuro

No decorrer do presente trabalho de tese surgiram várias ideias que não foram implementadas devido a limitações de tempo para submissão da tese ou porque se considerou que saíam do plano inicial proposto para esta tese. No entanto, considerou-se que valia a pena incluir algumas dessas ideias nesta secção como possíveis caminhos para investigação futura. É intenção do autor implementar algumas destas ideias como parte, por exemplo, de um possível projecto de pós-doutoramento que dê continuação à presente dissertação.

Entre possíveis ideias de trabalho futuro, eis algumas que se julgou por bem mencionar:

- Implementação do algoritmo de triangulação de superfícies moleculares Gaussianas com caixas e esferas de influência de diferentes tamanhos. A ideia é dotar o algoritmo de um mecanismo de balanceamento de carga de acordo com a densidade atómica das moléculas. A ideia poderia passar por usar caixas de influência mais pequenas em regiões onde houvesse uma maior concentração de átomos.

- Paralelização do algoritmo de triangulação de superfícies molecular Gaussianas. A ideia seria usar a partição espacial do domínio que encerra a molécula em caixas e esferas de influência, atribuindo depois estas caixas a diferentes processadores paralelos em GPU por forma a calcular e triangular localmente a superfície molecular.

- Paralelização do algoritmo de empilhamento ou de assemblagem de ADN. A ideia passaria por dividir o esqueleto do ADN em partes, atribuindo então essas partes a diferentes processadores paralelos em GPU. Cada processador teria de construir apenas a porção da sequência de bases correspondente à porção do esqueleto que lhe tivesse sido atribuída.

- Desenvolver um algoritmo híbrido (adaptativo/preditivo) que conjugue a adaptabilidade da assemblagem com gamas de valores tradicionalmente tidos como aceitáveis para os parâmetros geométricos do ADN.

- Utilizar um algoritmo de suavização juntamente com o algoritmo de deformação de plasmídeos. O objectivo seria mitigar a ocorrência de dobras (ou *kinks*) mais acentuadas e menos realistas.

- Comparar resultados de simulações computacionais de plasmídeos específicos com resultados de experiências laboratoriais de purificação desses mesmo plasmídeos. A ideia é validar os modelos e métodos propostos. Tanto as experiências laboratoriais como as experiências computacionais deverão ser realizadas para os mesmos plasmídeos e nas mesmas condições físico-químicas. A ideia seria, portanto, comprovar em que medida o enrolamento/desenrolamento *in silico* mimetiza o enrolamento/desenrolamento *in vitro*.

Conclusões

Tal como inicialmente proposto, esta dissertação apresenta um conjunto completo de métodos para modelação geométrica, simulação e visualização de ADN plasmídeo. A integração destes métodos permite a construção tridimensional da sequência de pares de bases de qualquer plasmídeo, real ou teórico, sobre uma conformação arbitrária.

No cerne da tese, encontra-se um novo algoritmo de assemblagem de ADN e um método de deformação conformacional de ADN. O algoritmo de construção tridimensional de ADN faz uso de uma abstração mais elevada da forma no sentido de que a representação molecular do ADN se baseia em nucleótidos e não em átomos, utilizando-se portanto as superfícies moleculares dos nucleótidos como peças modulares de construção. Desta maneira, o modelo geométrico do ADN acaba por ficar computacionalmente menos oneroso, o que torna possível a visualização em tempo real de simulações de plasmídeos.

Esta dissertação apresenta ainda um novo algoritmo de deformação de plasmídeos para simulações de Monte Carlo. Quando comparado com os métodos de deformação convencionais, o novo algoritmo revelou-se mais eficiente e mais realista na mimetização do processo de deformação conformacional *in vitro*, ainda que sem comprometer a convergência para o estado de equilíbrio relativamente à energia elástica dos plasmídeos. A eficiência deste algoritmo contribui também para a visualização em tempo real dos processos de simulação.

Além disso, e tendo em consideração que o métdodo de deformação conformacional satisfaz o princípio da micro-reversabilidade, estão criadas as condições essenciais para reproduzir os processos de enrolamento/desenrolamento *in silico*.

Contents

List of Figures

List of Tables

List of Abbreviations

3D	Three dimensional
3DNA	CAM-based software package
A-DNA	A-form deoxyribonucleic acid
ADN	Deoxyribonucleic acid
ANSI	American National Standards Institute
APB	atoms per box
B-DNA	B-form deoxyribonucleic acid
BD	Brownian dynamics
bp	base pairs
C-DNA	C-form deoxyribonucleic acid
CADD	computer added drug discovery
CAM	Cambridge meeting method
CEHS	Cambridge University Engineering Department Helix Computation Scheme
CSV	Comma-separated values
CUDA	Compute Unified Device Architecture
DNA	Deoxyribonucleic acid
dsDNA	Double stranded deoxyribonucleic acid
EMBO	European Molecular Biology Organization
GBK	GenBank format file
isDNA	DNA stacking library
IUB	International Union of Biochemistry
IUPAC	International Union of Pure and Applied Chemistry
MC	Monte Carlo
MD	Molecular dynamics
ms	milliseconds
NAB	Nucleic Acid Builder
NDB	Nucleic Acid Database
NMR	Nuclear magnetic resonance
NURBS	Non Uniform Rational tensor-product B-Spline
OpenCL	Open Computing Language
OpenGL	Open Graphics Library
PDB	Protein Data Bank
pDNA	Plasmid deoxyribonucleic acid
RNA	Ribonucleic acid
SPA	Solid phase amplification
ssDNA	Single stranded deoxyribonucleic acid
UCSF	University of California San Francisco
VDW	van der Waals
w3DNA	CAM-based software package
WAG	Wedge angle method
Z-DNA	Z-form deoxyribonucleic acid

Chapter 1

Introduction

Plasmid DNA (pDNA) is a special type of DNA molecules largely used in DNA vaccination, gene therapy or recombinant biopharmaceuticals. pDNA is known for having a closed conformation in which the beginning of the double-helix is attached to its end. From its production in laboratory to its purification, a procedure that is used to extract the wanted conformations of pDNA by separating it from the relaxed or open conformations and other contaminants, performing *in-vivo* experiments with pDNA has high costs and is very time consuming. Using computers to simulate the same procedures *in silico*, may bring many benefits to life sciences, so that the design and development of computational methods for modeling, simulation and visualization of pDNA molecules on computers constitute the grand challenge that is behind this doctoral work.

In true, this thesis does not intend to replicate *in silico* pDNA purification procedures, which requires us to be able to simulate the pDNA uncoiling phenomenon starting from their natural supercoiled conformations. Instead, our intent is to introduce geometric models and methods for pDNA molecules as a basis for the simulation of pDNA purification on computers via uncoiling/coiling procedures.

1.1 Thesis Statement

The geometric simulation of uncoiling/coiling of DNA on computer require the fulfillment of the following requirements:

- Efficient triangulation methods for graphical visualization of large macromolecules.
- DNA stacking methods that adapt to successive conformations.
- Efficient Monte Carlo-based deformation methods for coiling and uncoiling phenomena.

As much as we know, there is no efficient way of triangulating surfaces concerning large macromolecules (i.e., those with at least one hundred of thousands of atoms), unless one uses parallel computation via CUDA, OpenCL or else; for example, the pUC19, which is a sort of pDNA, has about 180,000 atoms. Even worse it is the lack of techniques to deal with triangulations of macromolecules, which are made up of other subsidiary molecules; for example, a DNA molecule is composed by other molecules called nucleotides. In the case of DNA, what is done is using unrealistic, rigid, rectangular building blocks for nucleotides that are stacked in order to form DNA conformations. Furthermore, one uses the atom-based model for fine representations of DNA, and the worm-like model as its coarse representation. That is, one uses three different geometric representations for DNA, depending on the desired level of detail. In this thesis, we have achieved the goal of quickly modeling and rendering pDNA molecules via a molecular composition approach in terms of triangulations of the DNA macromolecule and its subsidiary molecules.

In respect to DNA stacking methods, those found in the literature are essentially predictive, i.e., they try to predict the conformation of a DNA molecule from its sequence of base pairs. It happens that this is not suitable for pDNA stacking in simulation procedures (e.g., Monte Carlo) that produce arbitrary conformations. Thus, the stacking of the pDNA base-pair sequences needs to adapt itself to arbitrary conformations, what has never been done before the research work carried out in the scope of this thesis.

In regard to Monte Carlo simulation of pDNA, let us say that traditional deformation techniques used by MC simulations have been essentially the same for many years. As shown later in this thesis, these deformation techniques are not efficient in many ways, in particular for real-time visualization of those simulations. In fact, the classic deformation method does not allow for smooth transitions from a DNA conformation to another. In this thesis, we introduce a new deformation method for DNA molecules that allow us to simulate and visualize the coiling of DNA in real-time and in a smooth 3D realistic way. Moreover, this deformation method satisfies the reversibility principle, so that it can be used to perform the coiling and uncoiling of DNA on computer.

Taking into consideration the requirements mentioned above, we can formulate the *thesis statement* as follows:

> *It is geometrically possible to replicate and render the DNA uncoiling/coiling processes in real-time using a commodity PC without parallel computing and graphics acceleration.*

As shown throughout this thesis, we intend to prove in practice that this statement is true, provided that we use molecular composition for triangulations, adaptive DNA stacking, and a reversible deformation method that allows for smooth transitions from a DNA conformation into another.

1.2 Research Questions

Considering the general problem of 3D plasmid DNA modeling, simulation, and visualization on computer, as well as requirements and the thesis statement presented in the previous section, the main research questions that need to be answered are:

Is it possible to model and render large DNA molecules, in particular the pUC19 that has about 180,000 atoms, in real-time without using parallel computing techniques?

The size of molecules, in terms of number of atoms, directly affects the efficiency and performance of triangulation algorithms traditionally used to build meshes of molecular surfaces. DNA macromolecules like pUC19, even if they have just a few thousands of base pairs, end up having many hundreds of thousands of atoms. Triangulating the entire molecular surface of pUC19 takes too long, while re-triangulating it every time it changes its conformation looks an impossible task. That is, it is not reasonable to triangulate the molecular surface of the DNA every time its conformation changes. Here, we follow a different strategy that consists of pre-triangulating its four types of subsidiary molecules, called nucleotides, namely: adenine (A), guanine (G), thymine (T), and cytosine (C). Then, we use molecular composition to build up the entire DNA molecule along its conformation trajectory, a process called DNA stacking.

Are current DNA stacking methods able to adapt the base pair sequence of a specific plasmid DNA molecule (e.g., the pUC19) to an arbitrary conformation without making use of its atomic structure?

The most popular DNA stacking methods are predictive, i.e., they try to predict the conformation of a given DNA molecule based on statistics and on their base-pair sequences. Consequently, they are not able to adapt a plasmid DNA base pair sequence to an arbitrary conformation. There are only a few DNA stacking methods with adaptive capabilities, but none of them was specifically designed for plasmid DNA modeling and simulation. In this thesis we propose a fully adaptive algorithm for DNA stacking. As said above, our method uses triangulated representations of the nucleotides as building blocks to assemble DNA base pairs sequences along arbitrary conformations very quickly, not to say instantaneously.

Are current DNA deformation methods adequate to visualize the simulation of uncoiling/-coiling of plasmid DNA molecules in real-time on computer?

For many years, the most popular deformation method for Monte Carlo simulations of pDNA has been the *crankshaft* move. The main problem is that many of the conformations generated by the *crankshaft* move are rejected by the Monte Carlo criteria. It is clear that this makes the crankshaft method not very efficient in terms of the acceptance ratio of conformation trials. In this thesis we propose a new reversible deformation method that is more efficient than the traditional *crankshaft* move. Moreover, our method does not generate the sudden displacement of large portions of the DNA like the *crankshaft* move does. This allow us to achieve a natural, smooth, and realistic visualization of plasmid DNA simulations in real-time.

1.3 Research Context

Since their early days, computers have been widely used to model, handle, simulate, and visualize molecules. From the simplest calculation methods for small molecules to the most recent, advanced 3D simulations of macromolecules and DNA sequence analysis, computational methods and algorithms play a more and more important role in molecular biology, biochemistry and biotechnology. In general terms, these computational methods and algorithms applied to the life sciences field originated knowledge domains like *computational biology*, *computational chemistry*, and *bioinformatics*.

Sometimes, computational biology is also referred to as bioinformatics. Although they have similar aims and use similar approaches, some researchers distinguish one from another in terms of scale; more specifically, bioinformatics has more to do with the organization and analysis of basic biological data, while the focus of computational biology is more on the construction of theoretical models of biological systems, as well as computational techniques for those models, in a way as mathematical biology does with respect to mathematical models.

On the other hand, *molecular modeling* pervades a number of knowledge fields, namely computational chemistry, drug design, computational biology and materials science, in the sense that it includes all theoretical methods and computational techniques that are used to model the molecules and their behaviours (or functions). Furthermore, *molecular graphics*, also known as graphics-based molecular modeling, refers to the use of computer graphics to display and manipulate molecules on computer screen. The term 'graphics' refers to image synthesis (i.e.,

the conversion of geometry into image), while the term 'molecular' refers to molecules and their components. Putting it into a simplistic manner, molecular graphics aims at studying molecules through the modelling, simulation, and visualization of the molecules themselves and their component parts. It is clear that molecular graphics plays a key role in the study of proteins and nucleic acids (e.g., pDNA). Therefore, this thesis falls into the domain of molecular graphics.

Molecular graphics studies all types of molecules, from smaller molecules with a few dozens of atoms, to macromolecules with up to thousands of atoms such as, for example, DNA. DNA is a special type of molecules that is present in all living cells, being well known for its double-helix structure [WC53], and for being very long and narrow molecules. In terms of structure, DNA can be decomposed into subsidiary constituent parts: strands; backbones; and nucleotides (or bases). The DNA double-helix is composed by two strands, each one of them being composed by a sugar-phosphate backbone and a sequence of nucleotides. In the most common types of DNA, there are four types of nucleotides: *adenine*, *cytosine*, *guanine*, and *thymine*. Because each nucleotide in a strand is paired with a nucleotide in the complementary strand, this allows researchers to codify DNA using its sequence of nucleotides, usually known as DNA *base-pair sequence*. Because of this particular chemical and geometric structure, it was necessary to create specific models for DNA molecules, specially for DNA stacking from its base-pair sequence without using its atomic structure, as usual in general-purpose molecular models. The majority of traditional DNA stacking models are predictive, i.e., they try to predict the conformation of a DNA double helix based on its base-pair sequence. However, simulation methods used for pDNA generate themselves a number of conformations along which it is necessary to assemble the pDNA sequence. Traditionally, these simulation methods have modeled pDNA just as a closed polyline. But, a more realistic and adaptive way of assembling DNA can bring many benefits for pDNA simulation procedures. Furthermore, adding a coarser level of detail, from atoms to nucleotides, is a way of enhancing the performance of DNA stacking algorithms.

Computer graphics, specially *computer aided geometric design*, provides general purpose geometric modeling and rendering techniques to visualize and interact with molecular models. 3D surfaces are probably one of the most used geometric tools in molecular graphics and visualization. More specifically, either parametric surfaces or implicit surfaces are used to build molecular surfaces. Over the years many models and algorithms have been proposed for molecular surfaces and their triangulations. However, we believe that not enough attention was given to Gaussian molecular surfaces [Bli82], a specific type of implicit surfaces that are simple to render and easy to deform, in particular to represent the nucleotides in DNA stacking.

Considering that the final goal of this thesis is to make possible pDNA simulation in real-time, it is worth to note that *Monte Carlo* simulation methods likely are the most popular for this type of DNA molecules. As most simulation methods, Monte Carlo methods are iterative. Monte Carlo methods are based on pDNA elastic energy calculations and statistics to accept or reject the pDNA conformation generated in each iteration step. In a simple way, starting with any arbitrary pDNA conformation, a new trial is generated in each new iteration step by deforming the previous one, and the new trial is accepted as the new conformation if it minimizes the pDNA elastic energy or, alternatively, if there is a certain probability of its occurrence. The methods used to generate new conformation trials have been in their essence the same for many years with slight improvements. We believe that the low acceptance ratio of the conformation trials generated by these methods, well as the occurrence of sudden-moving deformations of the molecules, does not make them suitable for real-time visualization of the simulation procedures.

More efficient and natural deformation would allow us to visualize pDNA simulations in a more realistic way.

1.4 The Course of the Research Work

The rationale behind this dissertation was the possibility of visualizing on computer screen what happens to plasmid DNA molecules conformations during laboratory experiments, specially during purification processes to which these molecules are submitted after its production. Assuming that plasmid DNA must keep its natural supercoiled conformation for DNA vaccination and gene therapy purposes, and knowing that physical and chemical changes on the purification process, such as temperature changes, may affect the final conformation of a given DNA molecule, the visualization of the process *in silico* would certainly bring benefits to plasmid DNA research.

The first stage of work presented in this thesis involved the study of the existing general-purpose molecular models, and how they could be useful for DNA modeling. A special attention was given to molecular surface models used in molecular modeling. Interestingly, we noted that the Gaussian molecular surface was not one of the most popular surfaces in molecular modeling. Also, there were not many triangulation algorithms for this specific kind of molecular surface. The idea was then to adapt a general-purpose implicit surfaces triangulation algorithm, already published by the author and his supervisor, to triangulate Gaussian molecular surfaces.

Moreover, it became clear that traditional molecular models, in particular the molecular surface models, have several limitations, in largely because they are essentially based on the atomic structure of molecules. It became apparent that triangulating large molecules like the plasmid DNA, which has hundreds of thousands of atoms, was prohibitive in terms of time performance. As a consequence, atom-based molecular surface models were not capable of handling properly large molecules such as plasmid DNA, mainly because every time it was necessary to deform the plasmid DNA conformation into another conformation, it was also necessary to geometrically rebuild the triangulation of entire molecular surface.

But there was also another major problem. In fact, the atomic structure was not available for many plasmid DNA molecules, which are usually described by their base-pair sequence, instead of their atoms. This led us to the idea of using nucleotide's level of detail instead of atom's level of detail for DNA stacking. Historically, this has been the traditional approach for larger DNA molecules. However, traditional DNA stacking models do not have a realistic appearance because their building blocks are literally blocks; hence the idea of using smooth, triangulated molecular surfaces of the four essential nucleotides as building blocks. Besides, unlike traditional algorithms which try to predict the DNA conformations, with our DNA stacking algorithm, and adopting the DNA axis as its skeleton, it was made possible to assemble any base-pair sequence over any arbitrary conformation in a completely adaptive way, what makes it specially suited for plasmid DNA conformation simulations.

The final step to achieve the objective of this thesis was to implement and run real plasmid DNA simulations. The Monte Carlo simulation method has been adopted because, historically, it has been the most used for plasmid DNA simulations. However, traditional deformation methods generate very sudden displacements of big portions of the molecule, what is not completely adequate to real-time visualization of the simulation evolution. In that sense, it was developed a new deformation method for plasmid DNA that generates smoother and more realistic defor-

mations of molecule's conformation over time. Finally, several experiments were performed in order to measure the effectiveness of this new method when compared to the deformation methods traditionally used in plasmid DNA Monte Carlo simulations. From these experiments it was possible to conclude that the new deformation method is smoother and converges faster to plasmid DNA energy equilibrium state.

1.5 Contributions

In spite of being a mix of computer science and molecular biology, this thesis essentially is a computer science dissertation, so that it adopts a language oriented to the computer science audience. As a result of the research work behind this thesis, its main contributions to the advance of the knowledge in molecular graphics are the following:

- The first continuation algorithm for triangulating smooth Gaussian molecular surfaces meshes. Moreover, unlike other triangulation algorithms for convolution surfaces, this algorithm does not need to take into account the whole atomic structure of the molecule to sample its surface; the sampling is done using only the atoms nearby each sampled surface point. In fact, this new triangulation algorithm divides the molecule's atoms into regions called influence boxes, with corresponding influence spheres, and triangulates the portion of molecular surface inside each influence box independently from its neighbour boxes, taking into account just the atoms that are inside the influence sphere that contains the influence box and ignoring the contribution of other atoms outside de influence sphere. The continuity and connectivity of the molecular surface is guaranteed by a mechanism of overlapping neighbor influence spheres.
- It proposes a new assembling or stacking algorithm for DNA that uses Gaussian molecular surfaces of the nucleotides as building blocks, which is specially suitable for plasmid DNA simulations. Most traditional DNA assembling methods are predictive, i.e., given a base-pair sequence they try to predict the conformation of the DNA molecule. The novelty of the DNA stacking method presented in this dissertation is the fact that it is not a predictive method. Being a completely adaptive method, it is able to assemble any base-pair sequence over any arbitrary conformation. Besides, because it uses molecular surfaces of nucleotides as building blocks, it gives a more realistic 3D appearance to the DNA molecules. Finally, because it builds upon nucleotides instead of atoms, it its lighter in terms of memory and time performance.
- This dissertation also presents a new efficient deformation algorithm for plasmid DNA Monte Carlo simulations that can be used for real-time visualization of any plasmid DNA simulation procedure. The novelty about this deformation algorithm, when compared to the classic deformation methods used in plasmid DNA Monte Carlo simulations, is that in each iteration it deforms only small pieces of the molecule, having a much higher acceptance ratio of the trial conformations than other methods. Nevertheless, the accumulated deformation over time is higher than in traditional methods. Thus, it is more efficient than traditional deformation methods. Besides, it also converges sooner, in less iterations, to plasmid DNA elastic energy equilibrium. It is worthy noting that this deformation algorithm can also be used as a general-purpose random deformation algorithm for any geometric model based on a polyline with equally sized segments.

Nevertheless, the major contribution of this thesis likely is the sum of all contributions, because this dissertation presents an end-to-end integrated framework for plasmid DNA geometric modeling, simulation and visualization. All the work is cumulative in a way that each method is essential to the next one: the general purpose triangulation algorithm is used by the molecular surface triangulation algorithm; the building blocks used by the DNA assembling algorithm are generated using the molecular surface triangulation; and the plasmid DNA deformation algorithm is used together with the assembling algorithm to allow the real-time visualization over time.

1.6 Publications

The authorship and full credits of the publications, new methods and software presented in this dissertation belong entirely to the author and his supervisor. None of the new algorithms or implementations presented are the result of some kind of collaboration work with other researchers or research groups. During this research work, the author was not a full-time PhD student, having a full-time job not related with the subject of the dissertation. This research was not financed by any PhD grant or research project, with the exception of the registration and attendance to conferences which were supported by the Instituto de Telecomunicações and Universidade da Beira Interior.

The work done during this dissertation gave origin to the following publications:

Adriano N. Raposo, Abel J. P. Gomes: Polygonization of multi-component non-manifold implicit surfaces through a symbolic-numerical continuation algorithm. In Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and South-East Asia (GRAPHITE'06), Kuala Lumpur, Malaysia, 29 November-2 December, pp. 399-406, ACM Press, 2006.

Abstract: *In computer graphics, most algorithms for sampling implicit surfaces use a 2-points numerical method. If the surface-describing function evaluates positive at the first point and negative at the second one, we can say that the surface is located somewhere between them. Surfaces detected this way are called sign-variant implicit surfaces. However, 2-points numerical methods may fail to detect and sample the surface because the functions of many implicit surfaces evaluate either positive or negative everywhere around them. These surfaces are here called sign-invariant implicit surfaces. In this paper, instead of using a 2-points numerical method, we use a 1-point numerical method to guarantee that our algorithm detects and samples both sign-variant and sign-invariant surface components or branches correctly. This algorithm follows a continuation approach to tessellate implicit surfaces, so that it applies symbolic factorization to decompose the function expression into symbolic components, sampling then each symbolic function component separately. This ensures that our algorithm detects, samples, and triangulates most components of implicit surfaces.*

Adriano N. Raposo, João A. Queiroz, Abel J. P. Gomes: Triangulation of Molecular Surfaces Using an Isosurface Continuation Algorithm. In Proceedings of the International Conference on Computational Science and Its Applications (ICCSA'09), College of Electronics and Information, Kyung Hee University-Global Campus, Yongin, Korea, June 29-July 2, 2009, pp. 145-153, IEEE Computer Society Press, 2009.

Abstract: *There are several computational models and algorithms for the visualization of biomolecules (e.g. proteins and DNA), usually as three-dimensional surfaces called* molecular surfaces. *Based on the Van-der-Waals model, which represents a molecule as a set of spheres (i.e. atoms), a major approach in modeling and visualization of molecular surfaces, called* blobby molecules, *represents an implicitly-defined molecular surface as the result of the sum of implicit functions, where each function describes a Van-der-Walls sphere (i.e. the geometry of an atom). In general, these surfaces are polygonized using the well known* marching cubes *algorithm or similar space-partitioning method. In contrast, this paper presents a very accurate continuation algorithm to polygonize* blobby molecules *with a very high mesh quality, smoothness and scalability.*

Adriano N. Raposo, Abel J. P. Gomes: 3D molecular assembling of B-DNA sequences using nucleotides as building blocks. Graphical Models 74(4): 244-254 (2012).

Abstract: *Unlike the current atomistic DNA models, this paper proposes a new 3D space-filling model for sequences of DNA base pairs using nucleotides, instead of atoms, as building blocks of DNA molecules. This nucleotide-based model is more scalable than the traditional atomistic model, and has the advantage that easily adapts to any topological conformation of DNA. Interestingly, this model also allows the building of the molecular surface of the DNA, either partly or entirely, as needed for energy computations in molecular applications. Moreover, it allows us to grasp the DNA shape at different levels of shape composition: atom, nucleotide, and DNA macromolecule as a whole.*

Adriano N. Raposo, Abel J. P. Gomes: Efficient deformation algorithm for plasmid DNA simulations. BMC Bioinformatics, 15(301), 2014.

Abstract: *Plasmid DNA molecules are closed-circular molecules that are widely used in life sciences, playing a major role in gene therapy research. Monte-Carlo methods have been used for several years to simulate the conformational behaviour of the molecules. These simulation methods need to randomly generate a new trial conformation in each iteration step to test it for acceptance according to energy calculations and stochastic rules. The simulation trials are generated by a method based on a crankshaft motion that, apart slight improvements, has been the same for many years. In this paper we present a new algorithm for the deformation of plasmid DNA molecules for Monte Carlo simulations. Our algorithm preserves both the size and connectivity of the segments of the plasmid DNA skeleton polyline. We also present the results of three experiments in which we compared our algorithm with the traditional deformation method in terms of acceptance ratio of the trials, energy and temperature evolution, and average displacements of the molecule. Our algorithm can also be used as a generic geometric algorithm for the deformation of regular polygons or polylines that preserves connections and segments lengths. When compared with the traditional method based on the crankshaft motion, we concluded that our algorithm generates simulation trials with higher acceptance ratio and smoother deformations, what makes it suitable for real-time visualization of plasmid DNA coiling. We have done this with success using a DNA assembling algorithm that uses nucleotides as building blocks.*

Adriano N. Raposo, Abel J. P. Gomes: isDNA: A Tool for Real-Time Visualization of Plasmid DNA Monte-Carlo Simulations in 3D, Lecture Notes in Bioinformatics, 9044, Part II, (566-577), Springer, 2015.

Abstract: *Computational simulation of plasmid DNA (pDNA) molecules, owning a closed-circular*

shape, has been a subject of study for many years. Monte-Carlo methods are the most popular family of methods that have been used in pDNA simulations. However, though there are many software tools for assembling and visualizing DNA molecules, none of them allows the user to visualize the course of the simulation in 3D. As far as we know, we present here the first software (called isDNA) allowing the user to visualize 3D MC simulations of pDNA in real-time. This is sustained on an adaptive DNA assembly algorithm that uses Gaussian molecular surfaces of the nucleotides as building blocks, and an efficient deformation algorithm for pDNA's MC simulations.

Adriano N. Raposo, Abel J. P. Gomes: Computational 3D Stacking Methods for DNA: a Survey. (provisionally accepted for publication in IEEE/ACM Transactions on Computational Biology and Bioinformatics).

Abstract: *DNA encodes the genetic information of most living beings, except viruses that use RNA. Unlike other types of molecules, DNA is not usually described by its atomic structure being instead usually described by its base-pair sequence, i.e., the textual sequence of its subsidiary molecules known as* nucleotides *(*adenine *(A),* cytosine *(C),* guanine *(G) and* thymine *(T)). The three-dimensional stacking (or assembling) of DNA molecules based on its base-pair sequence has been, for decades, a topic of interest for many research groups all over the world. In this paper we survey the major methods found in the literature to assemble and visualize DNA molecules from their base-pair sequences. We divided these methods into two categories:* predictive methods *and* adaptive methods. *As the name suggests, the aim of* predictive methods *is to predict the conformation of the DNA axis from its base pair sequence, whereas the goal of* adaptive methods *is to assemble DNA base-pairs sequences along previously known conformations. DNA stacking of base pairs along existing conformations is needed, for example, in scenarios such as DNA Monte Carlo simulations where arbitrary conformations can occur. We also present the major software tools that implements both predictive and adaptive methods.*

1.7 Software and Hardware Tools

In terms of software development, all the code was exclusively and fully designed and written by the author. The Gaussian molecular surface triangulation algorithm and the DNA assembling algorithm were implemented using Java 3D. The DNA assembling algorithm was later migrated to C++ and OpenGL for performance enhancement. The pDNA deformation algorithm and Monte Carlo simulations were implemented only in C++ and OpenGL. The development environments used were NetBeans for Java programming and Visual Studio Express for C++/OpenGL programming. With the exception of Harris' knot detection library [HH99] used in the implementation of the pDNA deformation algorithm, no other third-party software was used. No special hardware like parallel or high-performance computers, or even high performance graphics cards, were used at all, just a regular laptop computer.

In respect to codes, let us say that the one concerning the triangulation algorithm for Gaussian molecular surfaces can be found at:

`https://github.com/ISDNA/isMol`

The code of the stacking DNA algorithm can be found at:

`https://github.com/ISDNA/isDNA`

The code of the DNA deformation algorithm and the Monte Carlo algorithm can be found at:

`https://github.com/ISDNA/isDNASim`

1.8 Organization of the Thesis

This thesis has been written as a regular dissertation. It is not organized in published papers, but each one of the core chapters corresponds to one or more papers published or accepted for publication in/to conference proceedings or journal. More specifically, this thesis is organized as follows:

Chapter 1. This chapter introduces the thesis statement, the main research questions, and the contributions that sustain the thesis itself, as well as the context and the motivation that has led to the writing of this thesis.

Chapter 2. This chapter reviews the state-of-the-art in 3D stacking methods for DNA, with a focus on the predictive and adaptive methods.

Chapter 3. This chapter describes a continuation algorithm for the triangulation of Gaussian molecular surfaces defined as implicit surfaces. It is a predictor-corrector continuation algorithm. Thus, the algorithm uses the tangent plane to predict surface points in the neighborhood of the expanding front of the surface, and a Newton corrector to push the predicted points towards the surface.

Chapter 4. This chapter details the new B-DNA stacking algorithm that uses molecular surfaces of nucleotides as building blocks. It is an adaptive method since it adapts to the any trajectory of the DNA axis or conformation.

Chapter 5. This chapter introduces the new deformation method for pDNA Monte Carlo simulations. This method satisfies the principle of micro-reversibility, being thus adequate for the simulation of DNA coiling/uncoiling.

Chapter 6. This chapter presents the final conclusions to draw from this thesis, pointing also to a number of open issues for future work.

Finally, there should mention that the core of this thesis lies in the DNA stacking described in Chapter 4, though its impact is more apparent when applied to the deformation method introduced in Chapter 5. This impact is not only in terms of the efficiency of the deformation method in its convergence toward energy equilibrium, but also on its ability to mimic the real deformations in laboratory, and all of this happens in a computational environment that works at interactive rates, or in real-time.

Chapter 2

Computational 3D Stacking Methods for DNA

DNA encodes the genetic information of most living beings, except viruses that use RNA. Usually, a molecule is described by its atomic structure, but DNA is more commonly described by its base-pair sequence, i.e., by its symbolic string describing the sequence of its constituent molecules (called *nucleotides*), as it is the case of *adenine* (A), *cytosine* (C), *guanine* (G) and *thymine* (T). The three-dimensional stacking (or assembling) of DNA molecules based on its base-pair sequence has been, for decades, a topic of interest for many research groups all over the world. In this chapter, we review the major methods found in the literature to assemble and visualize DNA molecules from their base-pair sequences. We divided these methods into two categories: *predictive methods* and *adaptive methods*. Predictive methods aim at predicting the conformation of the DNA axis from its base pair sequence, whereas adaptive methods aim at assembling DNA base-pairs sequences along previously known conformations. DNA stacking of base pairs for existing conformations is needed, for example, in scenarios such as DNA Monte Carlo simulations where arbitrary conformations can occur. This chapter also presents the major software tools that implements both predictive and adaptive methods.

2.1 Introduction

Since the early days of computer graphics and computer simulation, life sciences and computer science researchers realized that they could use computers as a vehicle for the visualization and simulation of the behavior of atoms and molecules. Before that, the only way to represent molecules was to draw a two-dimensional sketch in a sheet of paper or to build three-dimensional physical models, but only for relatively small molecules. Bigger molecules, also known as *macromolecules*, composed by many hundreds to thousands of atoms could no longer be built up using such physical models. In this respect, computers help us to represent, visualize, and simulate the behavior of molecules and molecular complexes like proteins and DNA molecules, no matter their sizes in terms of atoms.

From the simplest models of the past to the sophisticated models of today, we have witnessed a remarkable evolution in computational representation of molecules. Based on the fact that atoms can be seen not just as points in space, but also as 3D electronic fields that can be represented by spherical surfaces, the earlier molecular models gave way to more realistic surface models. However, there is not a unique way to model the surface of an atom or a molecule. In fact, a large variety of methods and algorithms have been introduced in the literature to represent molecular surfaces [Con96, BLMP97, RPS+07, RQG09]. Nevertheless, in this chapter, we mainly address the problem of DNA stacking in the context of computer models and graphics.

It is well known that DNA is a very important issue in life sciences research because it encodes the genetic information of most living organisms. But, DNA is a molecule composed by atoms that are attached to each other in a very specific and recognizable way: the famous DNA dou-

ble helix. Essentially, what geometrically distinguishes DNA from other *macromolecules* is the way its constituents respect well defined assembling rules, forming very long and narrow helical molecules. Thus, to handle DNA molecules it is very important to know all its structural constituents and geometry. In a simple way, we can say that DNA is made up of four subsidiary molecules called *nucleotides*: *adenine* (A); *cytosine* (C); *guanine* (G); and *thymine* (T). When connected in a sequential way, the nucleotides form a strand that connects to a complementary strand to give origin to a double-stranded helix. Taking into account that the atomic structure of the nucleotides is well known, and because there is a unique correspondence between the nucleotides in the two strands, the DNA molecule is usually represented by its nucleotides sequence, also known as base-pair sequence.

Although DNA molecules can be built using generic molecular models based on the atomic structure, a number of assembling methods specifically for DNA, called DNA stacking methods, have been introduced in the literature in the last few decades. These methods aim at building up DNA molecules from their base-pair sequences, without explicitly knowing the atomic structure. There are two main families of DNA stacking methods: (a) the *predictive methods*; and (b) *adaptive methods*.

The predictive methods include the Cambridge meeting (CAM) method [Dic89] and the wedge angle (WAG) method [KST82, BMHT91]. These methods try to predict possible conformations of a DNA molecule from its DNA base-pair sequence. Predicting a conformation is based on statistical values for angles and displacements between consecutive DNA nucleotides. This DNA assembling paradigm is specifically suitable for scenarios (e.g., conformation and curvature analysis for DNA-protein interactions) where the DNA base-pair sequence is known in advance, but not its conformation.

On the other hand, adaptive methods include those due to Raposo and Gomes [RG12] and to Hornus et al. [HLLF13]. These methods allow to assemble DNA nucleotides along the axis of a given conformation, i.e., they adapt a base-pair sequence of a DNA molecule to an arbitrary conformation. Therefore, adaptive DNA assembling methods are specifically suitable for simulation scenarios because it is where typically DNA conformations (in particular, plasmid DNA conformations) are determined and generated before the DNA assembling.

The remainder of this chapter is organized as follows. Section 2.2 presents essential DNA concepts, a DNA taxonomy, as well as the topology of the closed-circular DNA, also called plasmid DNA. Section 2.3 reviews the *predictive methods* for DNA stacking. Section 2.4 presents the *adaptive methods* for DNA stacking. The previous two sections also approach software tools that incorporate predictive and adaptive methods. Section 2.5 briefly discusses the current state-of-the-art of DNA stacking, with some hints for future reserach. Finally, Section 2.6 concludes the present chapter.

2.2 Background

2.2.1 DNA Basics

In 1953, Watson and Crick made one of the most important discoveries in life sciences: the DNA double helix structure [WC53]. They discovered the primary structure of DNA, which consists of two sequences of subsidiary molecules, called *nucleotides*, that form a double stranded helix.

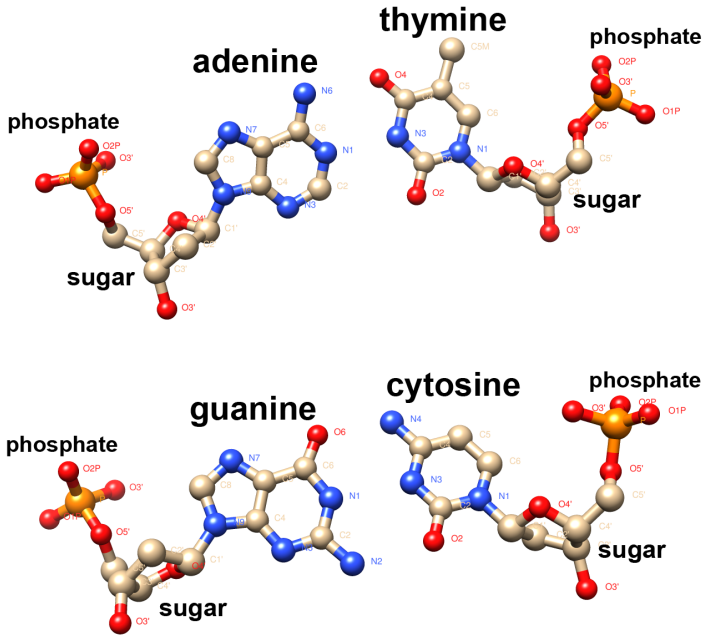


Figure 2.1: Two possible DNA base-pairs: adenine-thymine (top); guanine-cytosine (bottom). Each nucleotide is composed by a base, a sugar, and a phosphate. Each nucleotide is attached to the corresponding pair by hydrogen bonds.

Every nucleotide has three components: a *base*, a *sugar* (2'-deoxyribose), and a *phosphate* (see Figure 2.1). The base of a nucleotide can be one of the following types: (A) *adenine* ; (C) *cytosine*; (G) *guanine*; and (T) *thymine*. *Adenine* and *guanine* are purine bases, while *cytosine* and *thymine* are pyrimidine bases. Each nucleotide's base is attached to the 1'-position of its sugar, which in turn is attached to the phosphate of the next nucleotide in the DNA sequence (phosphodiester bond), originating a structure know as *polynucleotide chain* or *DNA chain*. The sugar-phosphate structure in a DNA chain is also called *backbone*. It is usual to refer to the nucleotide triplet by name of its constituent base, being *A* an adenine nucleotide, *C* a cytosine nucleotide, *G* a guanine nucleotide, and *T* a thymine nucleotide.

Therefore, a double-stranded DNA molecule has two polynucleotide chains. Each nucleotide in the first chain is attached to a single nucleotide in the second chain by hydrogen bonding. This coupling is known as *base-pairing*, while the set of the two bonded nucleotides is called a *base pair*. In the Watson-Crick model, adenine always pairs with thymine and guanine always pairs with cytosine by means of, respectively, two and three hydrogen bonds (see Figure 2.1). Other possible base-pairing schemes, such as the Hoogsteen base-pairing found in DNA triplexes (DNA with three strands) [Hoo63], are out of the scope of this thesis.

Additionally, small sequences of consecutive nucleotides in the same polynucleotide chain have specific names [JKSS96]. A *dimer* (or *dinucleotide*), a *trimer*, a *tetramer*, a *pentamer*, a *hexamer*, a *heptamer*, a *octamer*, a *nonamer*, a *decamer* and a *dodecamer* are, respectively, sequences of 2, 3, 4, 5, 6, 7, 8, 9, 10 and 12 nucleotides. Taking into consideration that we have four sorts of nucleotides, we end up having sixteen different *dinucleotides*: AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT, TA, TC, TG and TT. As shown further ahead, dinucleotides

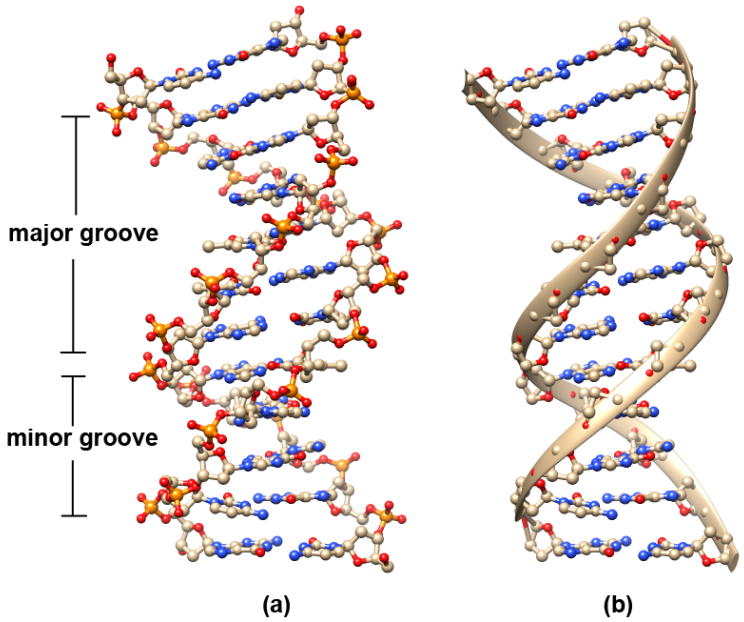


Figure 2.2: Double stranded DNA segment with 12 base-pairs (1BNA from the Protein Data Bank [Mey97]): (a) ball-and-stick representation where it is possible to see the DNA major and minor grooves; (b) ribbon representation of the two DNA backbones.

are particularly important in DNA assembling. In fact, the direction of a DNA polynucleotide chain is defined by the phosphodiester bond (5' $\rightarrow$ 3') and in double stranded DNA the two chains have opposite directions. The two strands are coiled around each other in a right-handed way. Having two coiled strands, where bases are stacked on each other, makes DNA molecules to be hydrophobic in the inside and hydrophilic in the two sugar-phosphate backbones. As shown in Figure 2.2, the DNA *minor groove* occurs where the strand backbones come close together, while the *major groove* is where they are far apart. The major groove is of utmost importance as it makes room for other molecules, such as proteins, to easily interact with the DNA bases [PS84].

2.2.2 DNA Taxonomy

The helicoidal shape of DNA is largely due to the way the nucleotides are stacked, i.e., the conformation of the DNA molecule depends on its sequence of base-pairs. In fact, a specific DNA form presents particular helical parameters in respect to the number of base-pairs per turn, helical handedness (right handed or left handed), helix diameter, and major and minor grooves. Therefore, there are two main forms of DNA, namely: B-form and A-form. Other rare forms of DNA, such as *C*, *D* [BoCGB06] or *Z* [HR96] are out of the scope of this thesis. We will focus essentially on B-DNA with some references to A-DNA whenever found appropriate.

2.2.2.1 B-form

This is the most common DNA found in cells, which is also known as *B-DNA* [BM05]. Its helical parameters and characteristics are the following: (1) a right handed helix with an approximated diameter of 20 Å; (2) base-pairs nearly perpendicular to the helix axis and with an average

distance between them of 3.3 Å; (3) a wide and deep major groove and a narrow and deep minor groove; and (4) approximately 10.5 base-pairs per turn.

2.2.2.2 A-form

It is also known as *A-DNA*, and features the following values for its helical parameters: (1) a right handed helix with an approximated diameter of 23 Å; (2) base-pairs tilted and laying well off the helix axis; (3) a wide and deep major groove and a wide and shallow minor groove; and (4) approximately 11 base-pairs per turn. In some circumstances, it may be difficult to distinguish between the B-DNA and A-DNA forms provided that their characteristics can be affected by solution conditions and by the base-pair sequence itself [BM05].

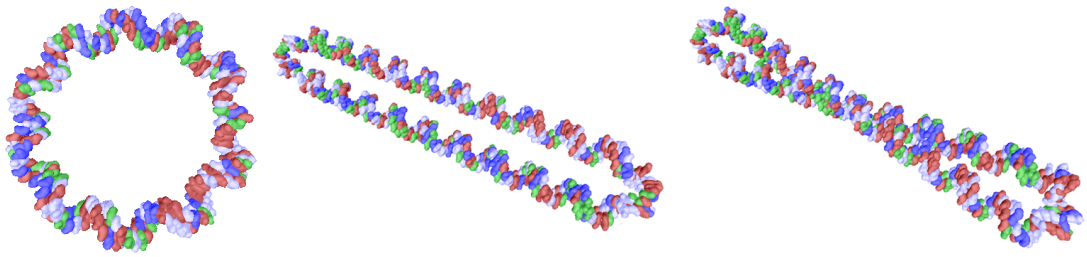
2.2.2.3 Z-form

Also known as *Z-DNA*, it was the first form of DNA to be discovered as having a left handed double helix [BM05]. The 'Z' had its origin in the zigzag path of the sugar-phosphate backbones. Z-DNA does not possess a major groove and its minor groove is narrow and deep. The occurrence of Z-DNA in nature still is a controversial issue [HR96].

Let us here refer that DNA conformation changes under thermal perturbations over time, so that its energy plays a very important role in its packaging and its function inside the cell. This is consistent with the fact that base-pair stacking interactions determine DNA's elastic response; hence, the efforts to come up with a precise method to measure local chain *curvature* and the local chain *flexibility*, as necessary to the study of DNA-protein interactions [ZSC$^+$01]. Recall that chain flexibility refers to the capability of a DNA molecule to be bent or twisted; the flexibility is said to be *isotropic* if DNA equally bents in all directions, and *anisotropic* if DNA bends in a preferred direction. To a more comprehensive discussion on DNA biophysics the reader is referred to [FK97].

2.2.3 Closed-circular DNA

In 1963, Vinograd's and Vogt's research groups were studying the polyoma virus and found that the DNA of this virus had two components: *I* and *II* [WV63, DV63]. Both components were found to be identical in terms of their base-pair sequences. However, *component I* was more compact and more resistant to denaturation when exposed to higher temperatures or pH increase. According to those researchers, this suggested that, in the case of *component I*, they could have found "circular base-paired duplex molecules without chain ends" [VLR$^+$65]. Basically, they found a double-stranded molecule, with helically winded strands, in which the first nucleotide of each strand was attached to the last nucleotide of the same strand, like every other ordinary dimer in the sequence. This special family of DNA molecules is known as *closed-circular DNA* and is illustrated in Figure 2.3. Later, researchers found that polyoma's *component II* was also a *closed-circular DNA* molecule and tried to explain why those two closed-circular DNA molecules (i.e., components I and II) with the same base-pair sequence had different behaviors. At this point, researchers observed that the main difference between the two components was their conformations, i.e., *component I* molecules had many crossings of the double strands and *component II* was mainly relaxed with fewer crossings.



(a) (b) (c)

Figure 2.3: Closed-circular DNA: (a) closed-circular DNA with 120 base-pairs; (b) closed-circular DNA with 240 base-pairs in a relaxed conformation; and (b) closed-circular DNA with 240 base-pairs in a twisted conformation. These pictures were generated using the Raposo-Gomes model [RG12] and the isDNA software [RG15].

Vinograd and colleagues were the first to suggest that component *I* of closed-circular molecules could be "twisted", being this the reason why *component I* was more compact and had more double-strand crossings than *component II* [VLR+65]. This twisting of the double-strands in closed-circular DNA molecules (which must not be mistaken as the helix twisting) is known as *supercoiling* and is one of the most important concepts related to this particular type of DNA molecules (see Figure 2.3(c)). In the case where closed-circular DNA is not *supercoiled* it is usually referred to as being *relaxed* (see Figure 2.3(a) and (b)). DNA supercoiling can be: *positive* if it is formed in the same direction as the DNA helix, i.e., if it overtwists the helix; or *negative* if it is formed in the opposite direction.

2.2.3.1 Knots and Catenanes

Topologically speaking, a *knot* is a concept only meaningful for closed curves. If we imagine a piece of string, loop it around itself as we want, and then splice the ends together, we got a *knot*. In turn, the topological interlinking of two or more rings is called a *catenane*. For example, if we imagine two pieces of string and then splice the ends of those strings in a way that they stay linked, we got a *catenane*. Knots and catenanes (topologically also known as *links*) constitute a vast subject of study in mathematical topology. For a more detailed mathematical description of knots and catenanes, the reader is referred to [Goo05].

Thus, although being uncommon in nature, *knots* can occur in closed-circular DNA molecules. In fact, Liu and colleagues were the first to observe knots in a single-stranded DNA [LDW76]. Although knots can occur in single-stranded DNA in nature, they can also be produced artificially using synthetic oligonucleotides and DNA ligase [BS02]. A few years later, Liu and colleagues were also the first to observe knots in double-stranded DNA [LPCW81]. These earlier studies were important because their data, coupled with simulations of knotting probability, made possible to calculate the effective diameter of the DNA double helix [BM05]. As occurs to single-stranded DNA knots, double-stranded DNA knots can also be artificially produced using a variety of enzymes [BC81].

There are several types of *knots*. The most simple is the *trefoil knot*, so called because it looks like having three lobes when laid flat (see Figure 2.4). The *trefoil knot* can be made both in a right-handed sense (clockwise) or left-handed sense (counter-clockwise). Knowing that *nodes* are cross-over points of DNA helices that can be either positive or negative, by convention, a clockwise rotation defines a *negative node* (-1) and a counter-clockwise rotation defines a

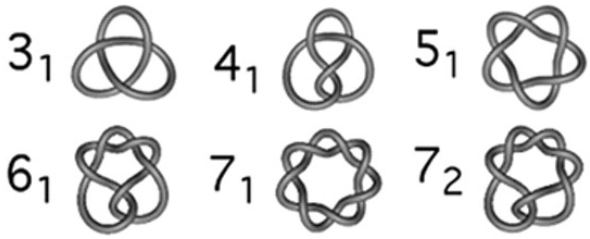


Figure 2.4: Examples of knots using the Alexander-Briggs notation.

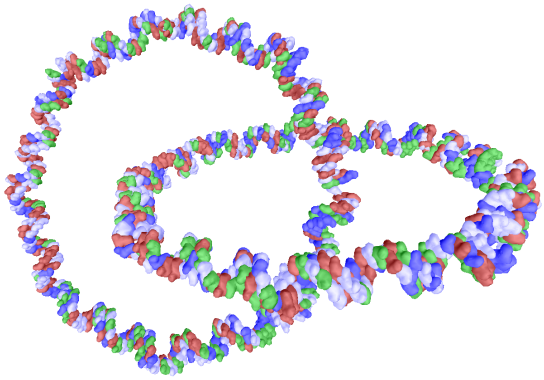


Figure 2.5: Simple catenane: two interlinked closed-circular DNA molecules.

positive node (+1). The sum of the nodes in a plane projection is called the *intrinsic linkage of the knot* and it is usually represented by Kn. In the case of the trefoil knot $Kn = +3$ or $Kn = -3$. Although there are various notation systems that can be used to classify knots [WMC87], the Alexander-Briggs notation is the most popular, being usually represented as n_i, where n represents the number of nodes in a plane projection and i distinguishes knots with the same number of nodes [Ale28]. Later, the *ideal knots* definition has been proposed as a development of the Alexander-Briggs classification [KBM+96]. It is also worthy to say that knots that are composed by two or more knots are usually called *composite knots*.

Unlike *knots*, closed-circular DNA *catenanes* are common in nature, occurring mostly during replication, as observed, for example, in bacterial plasmids [NSSM73, SST76]. Closed-circular DNA catenanes were first observed in human cells by Hudson and Vinograd [HV67] and by Clayton and Vinograd [CV67]. Like for knots, Wang and Schwartz were the first to artificially generate closed-circular DNA catenanes [WS67]. DNA catenanes may be generated using a variety of enzymes including topoisomerases, which can also be used for both unknotting and decatenation of closed-circular DNA molecules [TW80, KC80]. For further details on DNA decatenation and unknotting, the reader is referred to [WS10] and [LDCZ09].

The simplest type of catenane is a set of two interlinked closed-circular DNA molecules (see Figure 2.5). When laying in a plane projection, this type of catenane can have two positive or two negative nodes. The *intrinsic linkage* of a catenane, usually represented as Ca, is the sum of its nodes. The notation n_i^j, similar to the Alexander-Briggs notation [Ale28], is often used to classify catenanes. In this notation, n is the number of intermolecular nodes, i is the number of closed-circular DNA molecules, and j distinguishes catenanes with the same number of nodes. Like for knots, a definition for *ideal catenanes* has also been proposed by Laurie et al. [LKS+98].

As shown further ahead, knots are very important in closed-circular DNA simulation procedures. More specifically, one must be able to check for the existence of knots during closed-circular

DNA simulations. One of the methods that can be used to analyze knots is the Harris-Harvey algorithm [HH99]. Assuming that two knots are considered topological different if they have different Alexander polynomials [AB27], the Harris-Harvey algorithm calculates the Alexander polynomial $\Delta(t)$ of knots to check whether they are topologically different or not. Thus, knowing that the Alexander polynomial of the circle (or trivial knot) is $\Delta = 1$, the algorithm assumes that any knot whose Alexander polynomial is not identically equal to unity is knotted. In the cases where it might be needed, the algorithm also determines a lower bound on the minimum number of path crossings. The original implementation of the Harris-Harvey algorithm is available as a library for the C programming language at http://uracil.cmc.uab.edu/Publications.

In this thesis, we are specifically interested in the role of knots in simulations that involve only single closed-circular DNA molecules. Thus, we will not discuss any methods to check for the occurrence of catenanes.

2.3 Predictive Methods

Predicting the trajectory of the DNA molecule from its base-pair sequence is not trivial and has been a challenge and a subject of study for many years. Fortunately, it was found evidence that there exists some dependencies between the DNA base-pair sequence and its trajectory [YK09]. However, each isolated base-pair by itself, when placed in an arbitrary position, does not provide enough geometric information for the definition of the global DNA trajectory. On the other hand, when two or more base-pairs are assembled together, it can be seen that each base-pair locally influences the positioning of its neighbors, thus contributing to the definition of the global trajectory of the DNA. The smallest sequence of consecutive nucleotides is usually called a *dinucleotide* (or *dimer*), i.e., a sequence of two nucleotides. In meanwhile, the geometric properties of dinucleotides for small DNA molecules (e.g., tetramers, hexamers, octamers, decamers and dodecamers) were obtained in laboratory using, for example, X-ray crystallography or nuclear magnetic resonance (NMR).

Recall that, and taking into account that we have four DNA nucleotides (A, C, G and T), we theoretically end up having 16 distinct dinucleotides. But, in practice, there are only 10 distinct dinucleotides, namely AA (=TT), AC (=GT), AG (=CT), AT, CA (=TG), CG, GA (=TC), GC, GG (=CC) and TA, because of the two important constraints imposed on the base-pairs: A always matches T and C always matches G. The leading idea was to find the values of important geometric features of each one of those 10 dinucleotides in the solved (or determined) tetramers, hexamers, octamers, decamers and dodecamers to reliably build the 3D structures of an arbitrary DNA sequence, i.e., to predict the trajectory of the DNA axis. Following Hunter and Lu [HL97], the rationale behind this idea is that if a given dinucleotide has an approximately similar geometry in most of the solved DNA molecules, its geometry might be used in the assembling of such dinucleotide in any arbitrary DNA sequence. Then, the real challenge has become to find the most appropriate geometric parameters for better characterization of the geometry of dinucleotides. Historically, two methods were elaborated to determine these parameters: the *Cambridge meeting method* (CAM) and the *wedge angle method* (WAG).

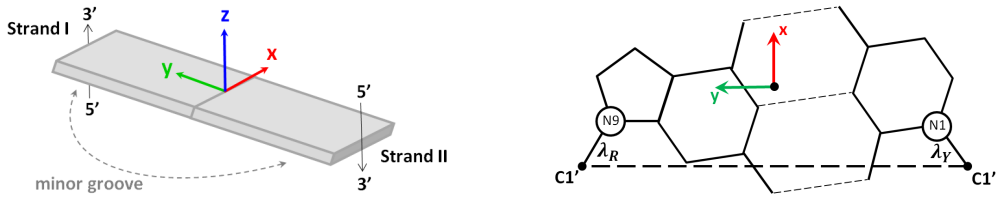


Figure 2.6: Base-pair coordinate systems: (a) original reference frame; (b) standard reference frame.

2.3.1 CAM Method

Possibly, the first attempt to establish standard parameters for DNA structure geometry occurred during the 1988 EMBO Workshop on DNA Curvature and Bending held at Churchill College, Cambridge, in September 1988 (cf. [Dic89] and also [VKD87, ST88, LS88, BB88]). As reported in [Dic89], the objective was to come up with a number of "recommendations for the definitions and nomenclature of nucleic acid structure parameters" in the hope that "the standards agreed will prove acceptable to others in the field, and ultimately will be approved by the IUPAC/IUB Commission on Chemical Nomenclature". This would be important to develop a generic library for DNA assembling in the future.

During and after the EMBO workshop, a number of decisions were taken in order to standardize the following parameters of DNA geometry:

1. Base-pair representation;
2. Base-pair coordinate system;
3. Base-pair parameters;
4. Dinucleotide steps.

2.3.1.1 Base-pair representation

It was then established that nucleotides would be represented as rectangular plates, so that each base-pair would be represented by two of these rectangular plates placed side by side, as shown in Figure 2.6(a).

2.3.1.2 Base-pair coordinate system

Standard reference axes were defined in a way that the x was aligned with the short axis of the base pair, y was aligned with the long axis of the base pair, and z was perpendicular to the base-pair plate, in a right-handed orthogonal reference frame.

Later, this local reference frame was changed because of the following two problems:

1. It may produce different base-pair stacking results when a distinct base-pair is chosen as reference base-pair;
2. Performing a roll rotation followed by a twist rotation may produce a different result if we exchange the order of rotations, i.e., the rotations are not commutative.

Table 2.1: Standard distances between base-pairs.

Base-pair	Hydrogen-bond distances (Å)					
	C1'···C1' (Å)	O2···H-N2	N3···H-N1	N4-H···O6	N3-H···N1	O4···H-N6
C-G	10.8	2.87	3.00	3.00	-	-
T-A	10.7	-	-	-	2.96	3.05

To solve these problems, and with the aim of drawing a map of conformational preferences of independent dinucleotide steps, El Hassan and Calladine [EHC95] added the mid-step reference frame concept for assessing the geometry of dinucleotide steps locally and independently. They called this scheme CEHS (Cambridge University Engineering Department Helix Computation Scheme).

But, as observed at some point by Lu and Olson [LO99], different DNA software packages — even if using the same scheme — were generating conflictual interpretations of the same structure, mainly because they were using different reference frames. In order to solve this problem, a group of researchers, including Olson, Dickerson, Lu, Tung and Berman, who created those programs, proposed a standard base reference frame [OBB+01], as shown in Figure 2.6(b).

In respect to the standard coordinate frame shown in Figure 2.6(b), Olson et al. [OBB+01] established that:

1. The x-axis is the perpendicular bisector of the C1'···C1' vector spanning the base-pair and points in the direction of the major groove;
2. The y-axis is parallel with the C1'···C1' vector;
3. The z-axis is defined by the right-handed rule, i.e., $z = x \times y$;
4. The location of the origin depends upon the C1'···C1' distance, i.e., the coordinates of the C1' atoms define the line in the base-pair plane where $y = 0$, and the rotation angles λ_R and λ_Y around a normal axis passing through the C1' atoms determine the two positions used to define the line $x = 0$.

Using this standard reference frame, Olson et al. [OBB+01] observed that local complementary base-pair and dinucleotide step parameters are nearly independent of analytical treatment. These authors also presented standard values for distances between the C1' atoms, and hydrogen-bond distances between base-pairs, as presented in Table 2.1.

Additionally, Olson and colleagues adopted the base coordinates proposed by Clowney et al. [CJS+96] (see Table 2.2) as the standard atomic coordinates for the four nucleotides, A, C, G and T. Note that C1' atoms were not included in Clowney et al.'s work, but they were added to the base atomic coordinates, in largely because of the unpublished results due to Berman et al. [BOB+92], who adopted base's atomic coordinates for their Nucleic Acid Database, as shown in Table 2.2.

2.3.1.3 Base-pair parameters

In addition to the reference frame, it was also defined five rotations and five translations for a generic base-pair to describe the parameters (i.e., movements) of the two nucleotides of a base-pair [Dic89], as shown in Figure 2.7. Such rotations include: tip (θ), inclination (η), opening

Table 2.2: Coordinates of non-hydrogen atoms in the standard reference frames of the four DNA bases.

Base	Atom	x (Å)	y (Å)	z (Å)
A (Adenine)	C1'	-2.479	5.346	0.000
	N9	-1.291	4.498	0.000
	C8	0.024	4.897	0.000
	N7	0.877	3.902	0.000
	C5	0.071	2.771	0.000
	C6	0.369	1.398	0.000
	N6	1.611	0.909	0.000
	N1	-0.668	0.532	0.000
	C2	-1.912	1.023	0.000
	N3	-2.320	2.290	0.000
	C4	-1.267	3.124	0.000
C (Cytosine)	C1'	-2.477	5.402	0.000
	N1	-1.285	4.542	0.000
	C2	-1.472	3.158	0.000
	O2	-2.628	2.709	0.000
	N3	-0.391	2.344	0.000
	C4	0.837	2.868	0.000
	N4	1.875	2.027	0.000
	C5	1.056	4.275	0.000
	C6	-0.023	5.068	0.000
G (Guanine)	C1'	-2.477	5.399	0.000
	N9	-1.289	4.551	0.000
	C8	0.023	4.962	0.000
	N7	0.870	3.969	0.000
	C5	0.071	2.833	0.000
	C6	0.424	1.460	0.000
	O6	1.554	0.955	0.000
	N1	-0.700	0.641	0.000
	C2	-1.999	1.087	0.000
	N2	-2.949	0.139	0.000
	N3	-2.342	2.364	0.000
	C4	-1.265	3.177	0.000
T (Thymine)	C1'	-2.481	5.354	0.000
	N1	-1.284	4.500	0.000
	C2	-1.462	3.135	0.000
	O2	-2.562	2.608	0.000
	N3	-0.298	2.407	0.000
	C4	0.994	2.897	0.000
	O4	1.944	2.119	0.000
	C5	1.106	4.338	0.000
	C5M	2.466	4.961	0.000
	C6	-0.024	5.057	0.000

(σ), propeller twist (ω), and buckle (κ). In respect to translations, they are: x-displacement (dx), y-displacement (dy), shear (Sx), stretch (Sy) and stagger (Sz). Recall that translations and rotations are rigid transformations, also known as metric transformations of Euclidean geometry.

The base-pair steps were further studied by El Hassan and Calladine in order to check whether some of them were not significant for the definition of the DNA conformation, and could be thus discarded [EHC97]. They found that shear, stagger and stretch steps show no or very little variation because they present typical values of 0 Å, 0 Å and 5.5 Å. On the contrary, propeller and buckle steps exhibited significant variations. Based on these results, *translational* base-pair parameters were considered as not significant, so that it was concluded that only *rotational* base-pair parameters are instrumental to the shape of DNA conformations.

2.3.1.4 Dinucleotide steps

As illustrated in Figure 2.8, it was recommended the following six geometric transformations for dinucleotides steps: 3 rotations and 3 translations, one per reference axis [Dic89]. The rotations around the x, y and z axes are called tilt (γ), roll (ϱ) and twist (Ω), respectively. The 3 translations in the x, y and z axes are called shift (Dx), slide (Dy) and rise (Dz), respectively.

The helical *twist* is one of the dinucleotide steps that contributes the most for the formation of the DNA helical shape, and one of the first to be geometrically studied (cf. Kabsch et al. [KST82]), what happened a few years before the EMBO meeting. Kabsch and colleagues used a collection of 33 short DNA sequences to determine averaged twist values for the ten dinucleotides listed in Table 2.3.

Some years after the EMBO meeting, Gorin et al. [GZW95] adopted Dickerson's definitions and nomenclature for DNA parameters in order to study the dependence of the twist, roll, slide, tilt and propeller angles in relation to the DNA sequence, well as the slide values of 38 previously solved B-DNA crystal structures obtained from the Nucleic Acid Database (NDB) [BOB+92]. Gorin and colleagues presented three important findings:

1. DNA twisting depends on DNA sequence and is directly involved in interactions with proteins;
2. The bending of DNA double helix also depends on the base sequence because of the correlation between twist and roll;
3. It is possible to predict DNA's bending and twisting changes after chemical changes.

The average values for twist, roll and slide presented by Gorin and colleagues are compared with other authors' results in Table 2.3. The values for tilt and propeller were not considered in Table 2.3 because they were considered not significant in posterior works [EHC97, GHP+03]. Interestingly, Olson et al. [OBB+01] observed and confirmed that the transformation from B-DNA to A-DNA tends to decrease twist, to increase roll and to reduce slide.

El Hassan and Calladine [EHC97] also studied the EMBO/CEHS geometric parameters of the dinucleotide steps [Dic89, EHC95], in particular in relation to a dataset of 60 DNA sequences (25 docecamers, 18 decamers, 16 octamers and 1 tetramer) previously solved by X-ray crystallography and digitally stored in the NDB database [BOB+92]. More specifically, El Hassan and Calladine studied the role of the backbone in determining the conformational preferences of the

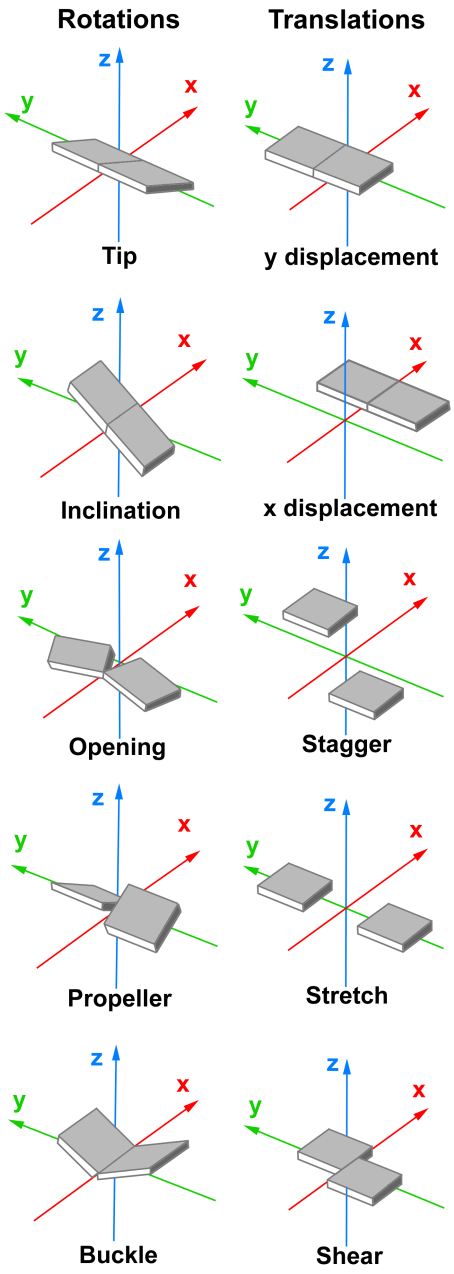


Figure 2.7: Cambridge meeting base-pair parameters.

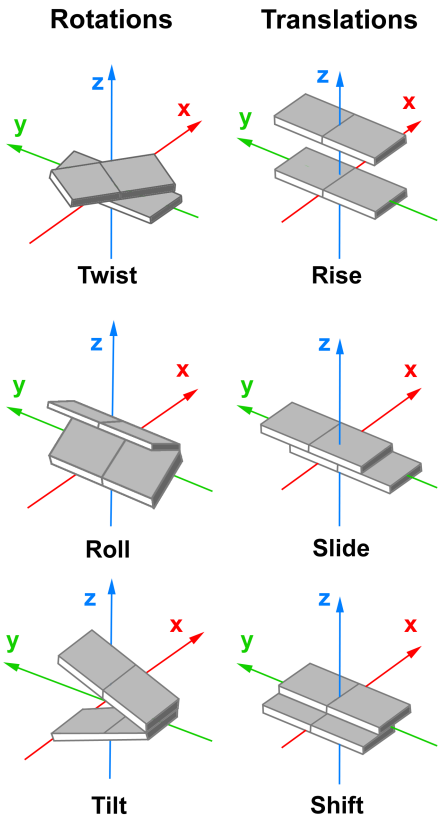


Figure 2.8: Cambridge meeting dinucleotide steps.

Table 2.3: Average values for DNA dinucleotides steps, in particular for *twist* (°), *roll* (°) and *slide* (Å), as presented by: (KST) Kabsch, Sander and Trifonov [KST82]; (GZO) Gorin, Zhirkin and Olson [GZW95]; (EHC) El Hassan and Calladine [EHC97]; and (GHP) Gardiner, Hunter, Packer, Palmer and Willett [GHP+03].

Dinucleotide	Twist (deg.)				Roll (deg.)				Slide (Å)			
Step	KST	GZO	EHC	GHP	KST	GZO	EHC	GHP	KST	GZO	EHC	GHP
AA	35.6	35.8	35.9	37.0	-	0.5	1.3	3.0	-	-0.03	-0.16	-0.20
AC	34.4	35.8	32.9	34.0	-	0.4	3.3	-1.0	-	-0.13	-0.89	-0.40
AG	27.7	30.5	37.8	36.0	-	2.9	4.4	0.0	-	0.47	-0.35	-0.10
AT	31.5	33.4	32.4	38.0	-	-0.6	-0.4	-6.0	-	-0.37	-0.44	-0.50
CA	34.5	36.9	37.4	31.0	-	1.1	1.2	9.0	-	1.46	1.18	-0.30
CG	29.8	31.1	35.1	34.0	-	6.6	0.0	7.0	-	0.63	0.00	-0.20
GA	36.9	39.3	37.8	38.0	-	-0.1	-0.4	5.0	-	-0.07	-0.35	-0.30
GC	40.0	38.3	37.4	36.0	-	-7.0	0.3	2.0	-	0.29	0.25	-0.60
GG	33.7	33.4	31.9	35.0	-	6.5	-1.1	2.0	-	0.60	-1.06	-0.20
TA	36.0	40.0	30.6	34.0	-	2.6	-0.8	9.0	-	0.74	-0.80	-0.30

dinucleotide steps. In fact, they studied A-DNA equally as B-DNA because B-DNA dinucleotide steps do not have always 0° *roll* and 36° helical *twist* (original way of classifying DNA), since it is sometimes difficult to distinguish B-DNA from A-DNA when some deformations occur.

Unlike Gorin et al. [GZW95], El Hassan and Calladine did not exclude dodecamer-end sequences from their study to compare their conformations with similar sequences that occur elsewhere in the DNA molecule. They classified the dinucleotide steps into two major categories: *rigid* steps for AA, AT and GA, and *loose* steps for CA, GG, GC, CG and TA; AC was classified as an intermediate step, i.e., not rigid nor loose. After excluding the unwanted steps, their database included a total of 400 dinucleotide steps: 64 occurrences of AA; 83 occurrences of CG; 56 occurrences of GG; 25 occurrences of GA; 21 occurrences of TA; 26 occurrences of CA; and 9 occurrences of AG. As a result of their study, El Hassan and Calladine classified twist, roll and slide as significant parameters, and tilt, shift and rise as nonsignificant parameters because they found negligible values of 0°, 0 Å and 3.4 Å for these latter parameters, respectively. The only exception was detected for some shift values greater than 0.5 Å that sometimes occur in GC and AC steps. Thus, they concluded that the conformation of a DNA molecule can be reasonably defined just using three parameters: *twist*, *roll* and *slide*. Table 2.3 shows the average values of these three dinucleotide steps as a result of research studies due to Kabsch et al. [KST82], Gorin et al. [GZW95], El Hassan and Calladine [EHC97], and Gardiner et al. [GHP+03].

In respect to Gardiner et al.'s work [GHP+03], the focus was on studying the DNA structural and flexibility parameters when DNA lies in its minimum energy conformation. For that purpose, they built up a database of the energy maps of all 136 unique tetramers, 2080 hexamers and 32,896 octamers to calculate the structural and flexibility parameters (or properties) of these sequences in order to study the overall behavior of the DNA structures resulting from the increase of their length. This work was essentially focused on conformations that tend to be those ones of the individual dinucleotide steps, classifying families of tetramers, hexamers and octamers by the type of the central dinucleotide step; for example, any sequence with a pattern like NAAN is an AA tetramer, NNAANN is an AA hexamer and NNNAANNN is an AA octamer, where N is any nucleotide, and analogously for the other 9 dinucleotide steps. With this approach, Gardiner and colleagues obtained interesting results about the overall behavior of the *slide* parameter:

1. As the DNA sequence length increases, the slide of the central step tends to zero.
2. 566 sequences presented features of A-DNA and B-DNA at the same time, specially in sequences in which the central step is GA, the step with more ambivalent conformational preferences;
3. All the octamers presented slide values lower than 0.8 Å, but the average slide was -0.3 Å, which is very close to the result obtained by El Hassan and Calladine [EHC97];
4. Most mixed DNA sequences tend to adopt a B-form conformation.

In relation to DNA *flexibility*, Gardiner and colleagues also claimed that:

1. The parameters that most contribute to DNA flexibility are twist and roll;
2. Flexibility clearly decreases from tetramer to hexamer to octamer;
3. Increasing twist and decreasing roll are the most flexible deformations; and,

4. Untwisting is the most difficult deformation for DNA.

It was also found that *tilt*, *rise* and *shift* show very little variation, what is in conformity with the results previously obtained by El Hassan and Calladine [EHC97]. Gardiner and colleagues also noted that stability of the elastic energy decreases in the order AA>AT>TA and GC>CG$\geq$GG, having also found that AG, GA, CA and AC have approximately identical average energies. For some particular steps, it was also found that there are correlations with the *twist* and *roll* parameters:

1. TA is the most flexible step with respect to increasing *roll* and decreasing *twist*;
2. AG, GA and GG are particularly inflexible with respect to decreasing *roll*;
3. CA is the most flexible step with respect to decreasing *roll*; and,
4. GG is the least flexible step with respect to decreasing *roll*.

Finally, Gardiner and colleagues found two extreme types of behavior for DNA *bending*:

1. Sequences with central step AA are straighter than any other sequence, reflecting preference for the B-type DNA;
2. Sequences with central step GG tend to be more bent than other sequences, reflecting preference for the A-type DNA.

Several authors studied the correlation of the backbones structure and the overall DNA structure [TS96, MJS96, PH98]. Having assumed that the sugar-phosphate backbone is driven by the structure of the nucleotides in DNA duplex, Tung and Soumpasis proposed a method to predict A-DNA and B-DNA using the three-dimensional coordinates of the phosphorus atoms at the DNA backbones [TS96]. The goal was to find the structure of a tetramer based on the positions of the three pairs of phosphorus atoms, one between two consecutive base-pairs. This method uses a Monte Carlo energy simulation procedure [MRR$^+$53]. The accuracy of Tung-Soumpasis method was tested by comparing the predicted base-and-sugar structures for 10 DNA sequences with the corresponding crystal structures obtained from the Protein Data Bank [BKW$^+$77].

Using a different approach, Packer and Hunter also studied the correlation between the conformational properties of the sugar-phosphate backbone and the base stacking interactions in dinucleotide steps [PH98]. These authors divided the dinucleotide steps parameters into three separate groups according to their dependence on the backbone or on the base stacking interactions:

1. The *roll*, *tilt* and *rise* steps do not depend on the backbone, depending exclusively on the base stacking interactions;
2. The *twist* step does not depend on base stacking interactions, being determined by the constraints of a relatively rigid fixed length backbone;
3. The *slide* and *shift* steps do not depend on the backbone neither on the base stacking interactions, being influenced by the neighboring steps.

It was also observed that the length of the backbone is a single parameter that might work as a descriptor for the conformation of the backbone and the way according which it is coupled to the

Table 2.4: Average values (and dispersion) of base-pairs parameters (BP), dinucleotide step parameters (DS) and local helical parameters (LH) observed in A-DNA and B-DNA crystal structures.

	Parameter	A-DNA	B-DNA
BP	Buckle (deg.)	-0.10 (7.80)	0.50 (6.70)
	Propeller (deg.)	-11.80 (4.10)	-11.40 (5.30)
	Opening (deg.)	0.60 (2.80)	0.60 (3.10)
	Shear (Å)	0.01 (0.23)	0.00 (0.21)
	Stretch (Å)	-0.18 (0.10)	-0.15 (0.12)
	Stagger (Å)	0.02 (0.25)	0.09 (0.19)
DS	Tilt (deg.)	0.10 (2.80)	-0.10 (2.50)
	Roll (deg.)	8.00 (3.90)	0.60 (5.20)
	Twist (deg.)	31.10 (3.70)	36.00 (6.80)
	Shift (Å)	0.00 (0.54)	-0.02 (0.45))
	Slide (Å)	-1.53 (0.34)	0.23 (0.81)
	Rise (Å)	3.32 (0.20)	3.32 (0.19)
LH	Inclination (deg.)	14.70 (7.30)	2.10 (9.20)
	Tip (deg.)	-0.10 (5.20)	0.00 (4.30)
	Helical twist (deg.)	32.50 (3.80)	36.50 (6.60)
	x-displacement (Å)	-4.17 (1.22)	0.05 (1.28)
	y-displacement (Å)	0.01 (0.89)	0.02 (0.87)
	Helical rise (Å)	2.83 (0.36)	3.29 (0.21)

base-pairs. Packer and Hunter used El Hassan and Calladine's scheme [EHC95] and dinucleotide steps geometry [EHC97]. Among others, Mazur et al. also presented a method to assemble DNA molecules based on an energy optimization-based construction of the backbones [MJS96].

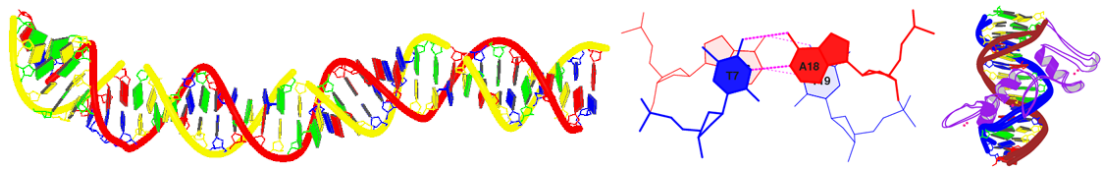
It is worthy noting that, based on the analysis within 3DNA [LO03], Olson et al. [OBB+01] obtained the average values and dispersion of base-pairs parameters, dinucleotide step parameters, and local helical parameters that were observed in A-DNA and B-DNA crystal structures. These values, here presented in Table 2.4, were calculated with respect to the standard reference frame given in Tables 2.1 and 2.2.

Before proceeding any further, let us remind that the final goal of studying all these DNA geometric parameters is to analyze, rebuild, assemble and visualize DNA molecules as 3D objects, what can be accomplished either from the atomic structure obtained by X-ray crystalography or from a given base-pair sequence.

2.3.2 CAM-based Software

Let us present a number of CAM-based software packages, that is, packages that follow Cambridge meeting's rules [Dic89] and Olson et al.'s standard reference frame [OBB+01] to assemble and analyze the conformation and curvature of a DNA molecule from its base-pair sequence, namely:

1. FREEHELIX, Dickerson [Dic98], 1998.
2. 3DNA, Lu and Olson [LO03], 2003
3. w3DNA, Zheng et al. [ZLO09], 2009.
4. Curves, Lavery and Sklenar [LS88], 1988.
5. Curves+, Lavery et al. [LMM+09], 2009.



(a) (b) (c)

Figure 2.9: (a) Example of w3DNA reconstruction of an arbitrary 40 base-pairs sequence (10 A-DNA, 10 B-DNA, 10 C-DNA and 10 B-DNA; (b) Example of a w3DNA visualization: stacking diagram for the base-pair number 6 (AT) in the 1BNA structure from the Protein Data Bank; (c) Another example of a w3DNA visualization: block representation of an ensemble structure, namely, the randomly choose 1TF3 structure from the Protein Data Bank.

2.3.2.1 FREEHELIX

As its name suggests, FREEHELIX (cf. [Dic98]) was free of any assumptions about a global helix axis, considering that the helix is a secondary phenomenon that results from the cumulative stacking of individual base-pairs. As Dickerson stated "One base pair says to its neighbors, *Let us stack*. It does not say, *Let us build a helix*". FREEHELIX could handle up to 50 base-pairs and 2000 non-hydrogen atoms in one continuous double-stranded nucleic acid, and it was able to present, in a tabular form, values for *roll*, *tilt*, *twist*, *slide*, *shift* and *rise*, relative to local axes defined between adjacent base-pairs. These parameters had been previously calculated by other programs [LS88, BPO93], but FREEHELIX had the special feature of using normal vectors attached to individual base pairs as a device to know how much the helix bends. The FREEHELIX project was later discontinued.

2.3.2.2 3DNA

One of the most popular software packages that adopted the Cambridge recommendations was 3DNA [LO03, LO08]. According to its developers, 3DNA was able to:

- Perform a complete conformational analysis of the DNA;
- Classify the double helical steps;
- Build DNA representations (standard stacking diagrams, sugar-phosphate backbone, schematic models and fiber models).

In relation to the conformational analysis, and since we use a PDB file with the atomic structure of a DNA molecule or segment, 3DNA is able to:

- Identify the base pairs;
- Calculate the six parameters for each base pair, namely: shear, stretch, stagger, buckle, propeller and opening;
- Treat non-Watson-Cricks base pairing motifs;
- Calculate the parameters for each dinucleotide step, namely: shift, slide, rise, tilt, roll, and twist.

On the other hand, and starting from a base-pair sequence instead, 3DNA is able to generate a sequence-dependent atomic structure of nucleic acids, with or without the sugar-phosphate backbone.

Also, 3DNA developers claimed that it was the first program to produce, in an automated way, the simplified representation of the DNA bases as rectangular blocks, assigning different styles to each of the six faces of the blocks. Based on the dimensions of a standard Watson-Crick base pair and on the average geometry of high resolution structures, 3DNA rectangular blocks have the following dimensions: lenght = 10 Å; width = 4.5 Å; and thickness = 0.5 Å. The blocks are color coded by residue type following the NDB convention (see Figure 2.9(a)): C - yellow; G - green; A - red; and T - blue (U - cyan). 3DNA also follows the NDB conventions for atoms naming and ordering. The DNA molecules generated with 3DNA have the following features: 11 bp/turn with a generic twist of 32.7° for A-DNA; 10 bp/turn with a generic twist of 36.0° for B-DNA. For a complete list of features included in 3DNA, the reader is referred to [LO03].

At this point, it is worthy noting that 3DNA can also export a PDB file that may work as input data for a third-party molecular visualization software such as, for example, Raster3D [MM94], PyMol [Sch10], BALLView [MHLK06, HDR+10], and UCSF Chimera [PGH+04]. The 3DNA suite was written in ANSI C programming language, with binding Perl scripts, and consists of over 30 executable programs running directly from the command line in any Unix/Linux operating system, or using Cygwin for the Microsoft Windows operating system.

2.3.2.3 w3DNA

The lack of both a friendly user interface and platform independence in relation to the operating system led to the development of w3DNA as a follow-up of 3DNA [ZLO09]. In fact, according to its developers, w3DNA is essentially a web-based interface to the 3DNA suite of programs. It consists of three major modules or components: analysis component, reconstruction component, and visualization component.

The w3DNA *analysis component* allows us to perform the analysis of a DNA molecule, being for that necessary to upload a PDB-formatted file (or simply a PDB ID or a NDB ID) via web browser. The w3DNA server then produces an output page containing five sections:

- Brief summary of the structure;
- Block representation (see Figure 2.9(a));
- Embedded 3D views using Jmol [Her06] and WebMol [Wal97];
- Output file with a complete listing of 3DNA-derived parameters;
- Parameter tables (sequence, local base step parameters, base pairing, local base-pair parameters and local base-pair step parameters)

In respect to the w3DNA *reconstruction component*, it allows for 3D assembling of arbitrary DNA sequences and arbitrary helical types, including:

- 55 models of DNA, RNA, and hybrid DNA/RNA helices;
- user-defined parameters for bases, base-pairs and base-pair steps;

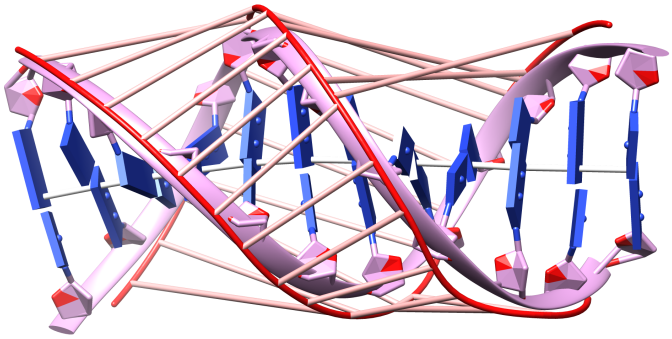


Figure 2.10: 1BNA structure from the Protein Data Bank combined with the helical axis (white tube in the center), backbones (red tubes on the outside) and groove geometries (pink tubes connecting the backbones) generated by Curves+. Chimera [PGH+04] was used for visualizing and exporting the image.

- curved DNA fragments of A-DNA, B-DNA and C-DNA; and
- models of DNA with proteins and other molecules.

The output of the reconstruction component is a PDB file with the atomic structure generated by w3DNA, although the web server also allows the preview of the structure using the block representation, as shown in Figure 2.9(a). For the arbitrary DNA depicted in Figure 2.9(a), we had to specify as input data the number of DNA segments, well as the base-pair sequence and the helical type for each segment. At the end, we can download the PDB-formatted file containing the atomic structure generated by w3DNA. Notice that sequences accepted by w3DNA's reconstruction component are limited to 1,000 base pairs.

Finally, w3DNA also includes a *visualization component* for rendering purposes. The w3DNA takes a PDB file or a PDB ID as input in order to produce high-quality drawings and scenes that can be rendered as raster-graphics images, including: (1) block/tube/backbone representation; (2) stacking diagram of base-pair steps (see Figure 2.9(b)); and (3) representations of structural ensembles (see Figure 2.9(c)).

2.3.2.4 Curves

Just before the 1988 Cambridge meeting, Lavery and Sklenar introduced *Curves*, a nucleic acid conformational analysis program [LS88, LS89]. Unlike Lu and Olson's 3DNA [LO03, LO08], who abandoned the idea of helical axis, the idea behind Curves was, indeed, to extend the notion of a helical axis to curved structures. According to the authors of Curves, the 3DNA approach is not well adapted to defining curvature and lacks the notion of base-pair positioning with respect to the axis that characterizes the differences between A-DNA and B-DNA. On the other hand, Curves had the advantage of simplifying the definition of curvature, maintaining the notions of macroscopic helical parameters.

2.3.2.5 Curves+

Curves+ is a follow-up of Curves developed by Lavery et al. [LMM+09], and was built up according to the Cambridge recommendations [Dic89] and Olson et al.'s standard reference frame [OBB+01]. Curves+ is freely available as a Fortran source code and, similar to w3DNA, it is also

available over the web. In fact, Lavery's group launched the *Curves+ web server* in 2011 as a wrap of the original Curves [BPZL11].

Unlike 3DNA and w3DNA, Curves+ does not reconstruct the atomic structure of a DNA molecule from its base-pair sequence. Instead, the user has to provide a PDB-formatted file with the description of atomic structure of such a DNA molecule that is being subject to analysis. Afterwards, Curves+ is able to generate numerical and graphical outputs on computer screen or, alternatively, into hard disk files, which in turn can be then analyzed using a third-party molecular visualization program. Nevertheless, Curves+ web server allows the user to visualize the results of the analysis using an embedded Jmol applet on-line. Note that Curves+ web-based interface generates three outputs:

1. The main result file containing base-pair parameters and dinucleotide step parameters (respecting the CAM nomenclature [Dic89]), well as backbone parameters and groove parameters;
2. A PDB file containing the helical axis structure;
3. A PDB file containing the backbone and groove geometry.

Figure 2.10 shows the helical axis (white), backbones (red), and groove geometry (light pink) that resulted from the Curves+ analysis (i.e., the input DNA atomic structure together with the two PDB files generated by Curves+) of the structure of the DNA molecule retrieved from the Protein Data Bank using the 1BNA identifier, which possesses 24 nucleotides, with nucleotides 1 to 12 in strand 1 and nucleotides 24 down to 13 in strand 2.

2.3.3 WAG Method

The wedge angle method (WAG), or simply wedge method is due to Trifonov and colleagues [KST82, BMHT91]. Instead of using values for each CAM step parameter, these researchers assumed that the axial path of the DNA can be described at each step by a translation and a rotation in terms of three independent angles:

- Rotation by half of the helical twist $\Omega/2$ about the axis z;
- Rotation by the wedge angle σ about the line in the xy plane perpendicular to direction of the DNA deflection, when the direction angle δ is measured from the new position of the axis x';
- Rotation by half of the helical twist $\Omega/2$ about the new position of the axis z'.

In addition, wedge roll (ρ) and wedge tilt (γ) can be expressed as the product of wedge angle value and cosine or minus sine of the direction angle, respectively, i.e., $\rho = \sigma \cos(\delta)$ and $\gamma = -\sigma \sin(\delta)$. The calculations are done using the matrix product $\mathbf{A}$ as follows [BMHT91]:

$$\mathbf{A}(\Omega, \sigma, \delta) = \prod_{j=1}^{n-1} \mathbf{T}(\Omega_j/2)\, \mathbf{M}(\sigma_j, \delta_j)\, \mathbf{T}(\Omega_j/2) \tag{2.1}$$

Table 2.5: Bolshoy and Trifonov wedge angles [BMHT91]. The values for dinucleotides CC, CT, GT, TC, TG and TT are not presented in the original article [BMHT91], so that these values were taken from [HFGG06].

Dinucleotide	Twist (Ω)	Wedge (σ)	Direction (δ)
AA	35.62	7.2	-154
AC	34.40	1.1	143
AG	27.70	8.4	2
AT	31.50	2.6	0
CA	34.50	3.5	-64
CC	33.67	2.1	-57
CG	29.80	6.7	0
CT	27.70	8.4	-2
GA	36.90	5.3	120
GC	40.00	5.0	180
GG	33.67	2.1	57
GT	34.40	1.1	-143
TA	36.00	0.9	0
TC	36.90	5.3	-120
TG	34.50	3.5	64
TT	35.62	7.2	154

In Eq. (2.1), $\mathbf{T}$ stands for the matrix of the twist rotation and $\mathbf{M}$ the matrix of rotation by the wedge angle, while the values Ω_j, σ_j and δ_j are the values defined in Table 2.5 for the j-th dinucleotide in the sequence of length n, being the product carried out over the entire sequence.

This wedge model assumes that the three-dimensional trajectory of DNA can be calculated as the vectorial sum of all dinucleotide wedge deflections along the length of the molecule. Trifonov and colleagues used 54 synthetic DNA fragments to calculate the wedge angles presented in Table 2.5; values for twist were taken from [KST82], whereas wedge and direction values were taken from [BMHT91]. Notice that, unlike in Table 2.3, Table 2.5 includes dinucleotides CC, CT, GT, TC, TG and TT, because they do not have the same values for direction as their homologous dinucleotides, i.e., they have the same values but with opposite signs.

2.3.4 WAG-based Software

In the literature, we find a number of codes that implement the WAG method, namely:

1. CURVATURE, Shpigelman et al. [STB93], 1993.
2. DNACURVE, Gohlke,
   `http://www.lfd.uci.edu/~gohlke/dnacurve`.
3. NAB, Macke and Case [MC98], 1998.
4. ADN Viewer, Gherbi and Hérisson [GH00, GH02, HFGG06], 2000.

2.3.4.1 CURVATURE

CURVATURE was one of the first software packages to implement the WAG model. It was developed by Shpigelman, Trifonov and Bolshoy [STB93], who claimed that, given any DNA sequence

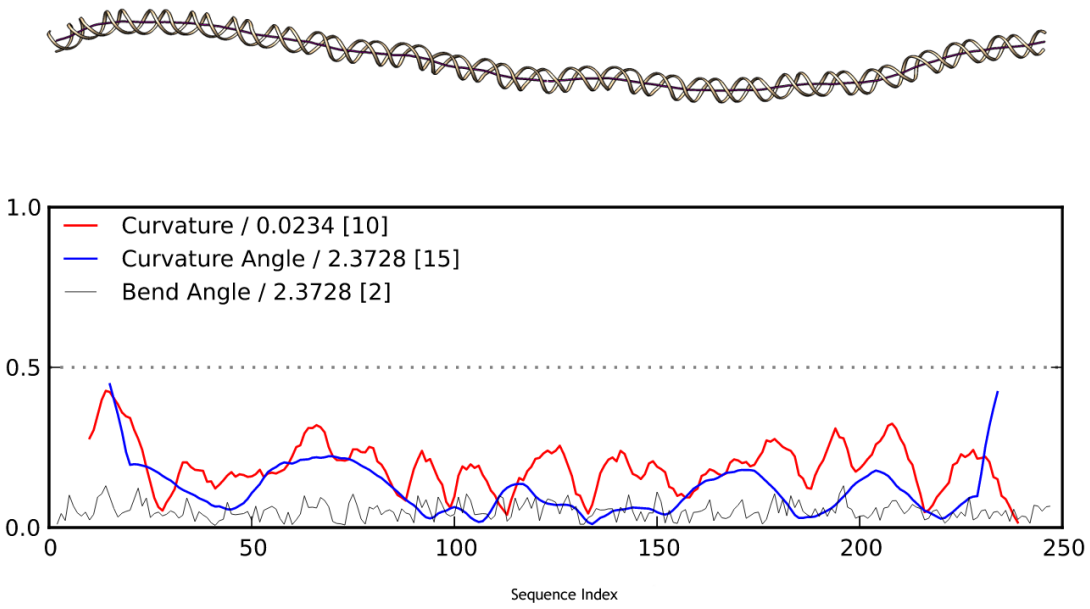


Figure 2.11: Gohlke's program results for a 250 base pairs DNA sequence using the WAG model: helix PDB file loaded and visualized with Chimera [PGH+04] (top); and the graphs of curvatures (bottom).

as input, CURVATURE was able to calculate the likely degree of curvature at each point along the molecule axis. They also claimed that their program could be used to investigate possible roles of curvature in the interactions of DNA with other molecules.

2.3.4.2 DNACURVE

CURVATURE is no longer available, but there is a web-based DNA Curvature Analysis program, named DNACURVE, by Christoph Gohlke (unpublished but available at `http://www.lfd.uci.edu/~gohlke/dnacurve`) that, among other models, implements the WAG model for a sequence with up to 256 base-pairs. Gohlke's program outputs are:

- A CSV file with all the calculated values;
- A PDB file with the helix coordinates, which can be loaded to a third-party visualization software such as Chimera [PGH+04];
- Plot of coordinates and curvatures.

Figure 2.11 shows an example of the output generated by Gohlke's program for a sequence of 250 base-pairs.

2.3.4.3 NAB

The Nucleic Acid Builder (NAB) was developed by Macke and Case [MC98] as a C-like programming language, compiled to C at an intermediate stage, in order to allow the construction of three-dimensional models of helical and non-helical nucleic acids having up to a few hundreds of base pairs. NAB allows the user to build models of molecules using three different atomistic methods:

- *Rigid body transformations*. These transformations allow for the incremental building of models of molecules by applying coordinate transformations to atoms in order to move them to their 'correct' positions and orientations after their instancing. The method is specially suited to helical nucleic acid molecules with their highly regular structures.
- *Distance geometry*. This method makes usage of interatomic distances as internal coordinates of the molecule. This means that the initial positions of atoms may be left unspecified. This also means that much structural information of the molecule is in the form of distances. Thus, distance geometry converts a molecule expressed in terms of a set of interatomic distances into a 3D structure of atoms and their bonds.
- *Molecular mechanics*. This method includes both energy minimization and molecular dynamics. Unlike the previous two methods, the molecular mechanics method is not suitable for creating initial structures of molecules, simply because it requires *a priori* good estimates for the initial positions of the atoms of a given molecule. Usually, it is used to adjust and refine an initial structure generated by either of the previous two methods, even when a structure comes with distorted geometry.

Currently, this software package is bundled as part of AmberTools v.13 [CCD+05]. For a complete overview of the Amber biomolecular simulation package, the reader is referred to [SFCW13].

2.3.4.4 ADN Viewer

Another WAG software package, called ADN-Viewer [GH00, GH02, HFGG06] (`http://nicolas.ferey.free.fr/dna-webviewer`), is due to Gherbi and Hérisson. They aimed to design a three-dimensional visualization tool for DNA sequences, making it clear that their objective was not to represent reality. Instead, their initial interests were in the management of complex scenes and in the interaction within a virtual reality environment [HGF+04].

However, because their approach was based on a three-dimensional biological model that predicts the trajectory of DNA molecules, i.e., the WAG model, they also claim that ADN-Viewer could be a powerful tool for the *in silico* analysis of three-dimensional DNA structures, providing functionalities in order to perform quantitative geometric measures as, for example, curvature, compactness and geometric distances.

ADN-Viewer's authors also claimed that their program could load and visualize multiple sequences of tens of million of base-pairs, not having the limitations of previous DNA visualization tools that were limited to small size molecules, being so possible the representation of complete chromosomes. Besides, according to Hérrison and Gherbi, at the time ADN-Viewer was released, the existing DNA visualization tools had been almost exclusively developed by biologists and were dedicated to particular biological problems. It is worthy noting that ADN-Viewer was also available within a virtual-reality environment which provided stereoscopic visualization and user-friendly interaction in a multimodal human-computer interface.

In regards to the implementation details of ADN-Viewer, its engine accepts as input DNA sequences. Then, using Shpigelman's conformational model [STB93] and the dinucleotides wedge angle values presented in Table 2.5, ADN-Viewer calculates the spatial coordinates of each nucleotide. These computations start by placing the first base pair at the origin of the scene, so that the position and orientation of each base pair is then calculated applying the matrix calculations in Eq. (2.2) to the position and orientation of its base pair predecessor:

$$\mathbf{M}_{xy} = \mathbf{T}(-\frac{h}{2})\mathbf{R}_z(\frac{\Omega}{2})\mathbf{Q}(\sigma, \delta - 90)\mathbf{R}_z(\frac{\Omega}{2})\mathbf{T}(-\frac{h}{2}) \tag{2.2}$$

where Ω, σ and δ take on values listed in Table 2.5; $\mathbf{T}$ is the translation matrix for moves along Z-axis; $h = 3.39$ Å is the value of the vertical translation (analogous to rise in the Cambridge convention [Dic89]); $\mathbf{R}_z$ is the rotation about the z-axis; and $\mathbf{Q}(\alpha, \beta) = \mathbf{R}_z(-\beta)\mathbf{R}_x(-\alpha)\mathbf{R}_z(\beta)$, where $\mathbf{R}_x$ stands for the rotation matrix about the x-axis.

Notice that this method (cf. Eq. (2.2)) of calculating the spatial position and orientation of each base pair is different from the one originally proposed by Bolshoy et al. [BMHT91] (cf. Eq. (2.1)). From here, it follows that the ADN-Viewer 3D engine made possible to calculate spatial coordinates at three different levels of detail: (1) coordinates of the atoms that make up each nucleotide; (2) coordinates of each nucleotide; and (3) coordinates of the plates (base pairs). According to ADN-Viewer's mentors, coordinates of the plates are the most appropriate to study the DNA trajectory.

To better adjust the visualization of DNA molecules, i.e., the greater the distance, the fewer details of the molecule are perceptible, ADN-Viewer visualization allows for three levels of detail:

- chromosomal level representation (Figure 2.12 (a));
- genic level representation (Figure 2.12 (b));
- atomic level representation (Figure 2.12 (c)).

At *chromosomal level*, the DNA trajectory is represented by a polyline that connects the center points of the base pairs. This is necessary to visualize structural information (e.g., compactness and relaxation) of large DNA molecules, in particular to guide the analysis over a specific region of a chromosome.

At *genic level*, the DNA representation is finer than at chromosomal level, since we are able to look at short fragments of DNA with up to a few hundreds of base pairs that make up genes. In this case, nucleotides are represented as colored spheres, one color per each type of nucleotide (A, C, G and T), so enabling an easier visual perception of some compositional properties of DNA, such as the zones of nucleotidic repetition.

Finally, the *atomic level* representation of DNA plays an important role when a fine biochemical information of the DNA is required. To represent DNA at atomic level, ADN-Viewer combines the predictive method by Trifonov et al. [KST82, BMHT91] to calculate the positions of the base pairs, together with the 3D coordinates of the four nucleotides obtained by observation methods such as crystallography or NMR (nuclear magnetic resonance).

Interestingly, ADN-Viewer was also used to address the problem of 3D pattern matching for DNA sequences [HPG07]. It is worthy noting that 3D sequence analysis does not replace 1D textual sequence analysis; it is rather a complementary approach when a global sequence analysis of DNA is needed. Based on the fact that two very different textual DNA sequences could have similar conformations [GH02, HGF+04], the ADN-Viewer includes an algorithm that compares the angles of the trajectories of DNA molecules in order to quantify the differences between the 3D sequences.

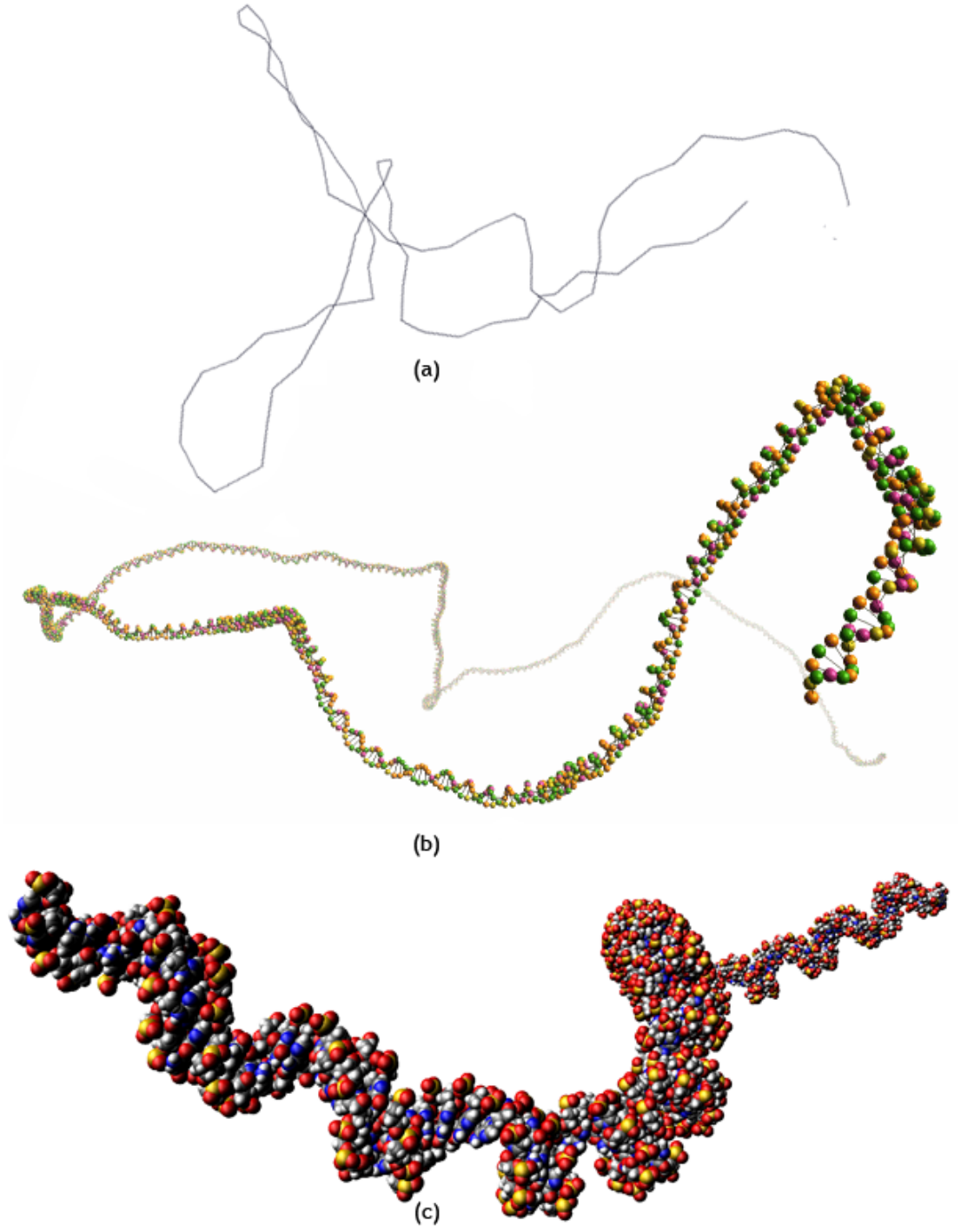


Figure 2.12: ADN-Viewer levels of detail: (a) *chromosomal level*; (b) *genic level*; and (c) *atomic level*.

2.4 Adaptive Methods

So far, we have only reviewed the conformational prediction methods, i.e., methods that predict the trajectory of DNA molecules based on its base pair sequences and on geometric rules for stacking of dinucleotides. However, in some scenarios, there is the need for assembling DNA sequences when the DNA axis is known *a priori*, as it is the case of DNA conformations generated by simulation methods like *Monte Carlo* methods. It happens that neither the CAM methods [Dic89] nor the WAG methods [BMHT91] are completely adequate for this challenge, in largely because they assume that the trajectory of the DNA axis in the process of dinucleotide stacking is determined by the sequence of nucleotides. Thus, unlike predictive methods, the adaptive methods assume that the geometry of DNA axis is known in advance.

In the literature, we find three different solutions that address the problem of assembling DNA molecules along arbitrary trajectories, (e.g., conformational trajectories previously calculated in simulation processes):

1. Macke and Case [MC98], 1998.
2. Raposo-Gomes method [RG12], 2012.
3. Hornus et al. [HLLF13], 2013.

2.4.1 Macke-Case Method

Macke-Case's method follows a strategy for wrapping DNA around a curve (i.e., the DNA axis) [MC98]. For that purpose, it is necessary to first find the locations of base-pairs on such a curve (i.e., the coordinate system origins of base-pairs); secondly, the base-pairs are placed at those locations along the curve; thirdly, the base-pairs are oriented in a manner that their helical axes are tangent to the curve, and finally rotated so that they end up having the correct helical twist. It is also assumed that the rise is constant and equals to 3.38 Å.

In order to compute the locations of base-pairs on a helix, let us spiral an inelastic wire (featuring the helical axis of the bent duplex) around a cylinder (featuring the protein core). This spiralling procedure can be described by four parameters, but only three of those are independent. The independent parameters are the number of base pairs n, the number of turns of DNA around the protein core t, the winding angle θ that controls how quickly the helix advances along the axis of the core. The fourth (dependent) parameter is the helix radius r, which is calculated from the following expression:

$$r = \frac{3.38(n-1)\cos(\theta)}{2\pi t} \tag{2.3}$$

Calculated the helix radius r, the method just needs to calculate the displacement along the axis (d) and the rotation around the same axis (ϕ) as follows:

$$d = 3.38\sin(\theta) \quad \text{and} \quad \phi = 2\pi\frac{t}{n-1} \tag{2.4}$$

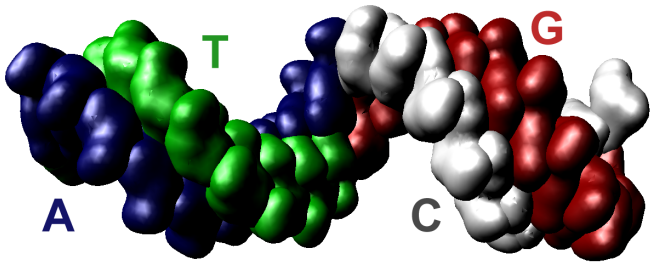


Figure 2.13: Raposo and Gomes DNA molecular surface building blocks: adenine (blue); cytosine (gray); guanine (red); and thymine (green).

in order to find the position of the base pair.

2.4.2 Raposo-Gomes Method

Raposo-Gomes' assembling DNA method works at the base-pair level of resolution [RG12]. For that purpose, it uses (triangulated) Gaussian molecular surfaces of the four nucleotides as building blocks to adapt DNA sequences to any arbitrary conformation. The Gaussian molecular surfaces are color-coded in order to easily identify the nucleotides (see Figure 2.13).

Unlike Macke-Case's method, which wraps the DNA around the axis curve, the Raposo-Gomes method makes usage of a piecewise linear approximation to the DNA whose straight line segments possess 3.3 Å of length each, i.e., the canonical value of rise in B-DNA. The base-pairs are stacked along this approximation of the DNA axis curve (see Figure 2.14(a)). The segment midpoint is the center of each base-pair. For the sake of convenience, each base-pair is positioned and aligned at the origin firstly, being then displaced to its final position in a plane perpendicular to the midpoint of its corresponding segment in the DNA axis.

More specifically, each base-pair is positioned in a way that phosphorus atoms of its nucleotides are offset by an angle of $\frac{2\pi}{3}$ radians around the DNA axis. Using the positions of phosphorus atoms as pivot points, the two nucleotides are then rotated in order to be aligned to each other (see Figure 2.15). The $\frac{2\pi}{3}$ radians rotation angle was adopted to obtain an approximation to the canonical dimensions of the major and minor grooves in B-DNA, assuming that B-DNA has approximately 10.5 base pairs per turn. At each iteration of the algorithm, the building blocks are first placed at the same initial position and then rotated $(i-1).(2\pi/10.5)$ radians around the z-axis (see Figure 2.15), where i stands for the i-th base pair in the sequence, i.e., the second base pair is rotated $(2\pi/10.5)$, the third base pair is rotated $2.(2\pi/10.5)$, and so on.

Note that Raposo-Gomes stacking algorithm allows for other types of building blocks other than Gaussian molecular surfaces. For instance, using VDW (van der Waals) surfaces as geometric representations of nucleotides, we end up obtaining a result that is visually equivalent to the atomic representation proposed by Gherbi and Hérrison in the ADN-Viewer software package, with the advantage that the stacking works along arbitrary axial trajectories, including closed-circular DNA, also called plasmid DNA. In fact, predictive assembling algorithms are not very-well adequate for plasmid DNA because it is very difficult, if not impossible, to predict that a specific base pair sequence has a closed-circular conformation. Besides, adaptive assembling algorithms have the advantage that they apply to DNA molecules that deform over time, as usual in simulation scenarios (e.g., Monte Carlo methods), as illustrated in Figure 2.14(b).

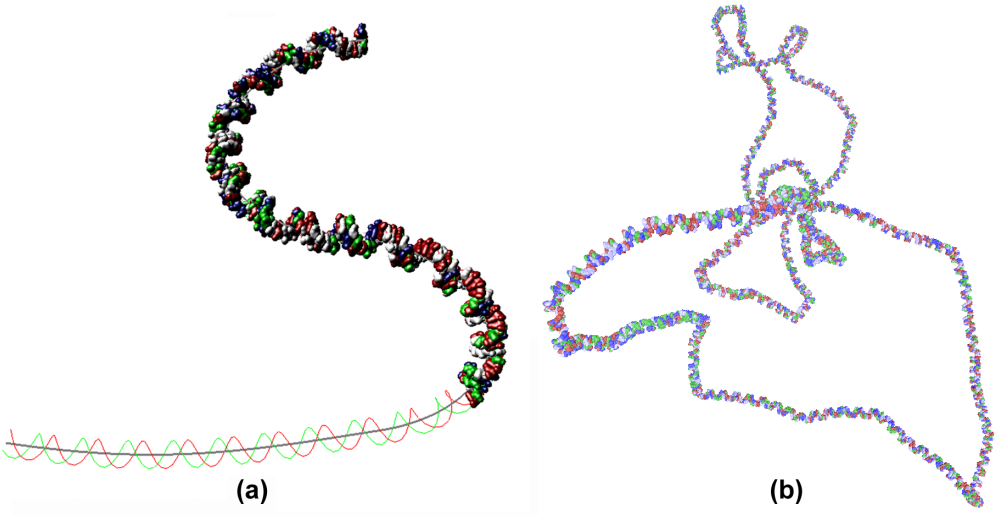


Figure 2.14: Raposo and Gomes assembling algorithm: (a) partial step in helix building for a DNA fragment; (b) closed-circular DNA (pUC19) during simulation process.

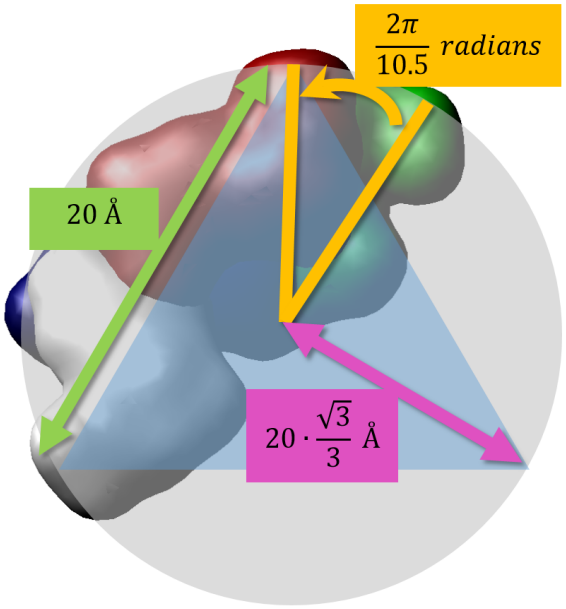


Figure 2.15: Raposo and Gomes building blocks relative positions: angle between consecutive base pairs (orange); distance between the two complementary building blocks (green); and distance from the backbones to the axis (pink).

2.4.3 Hornus et al.'s Method

Similarly to Raposo-Gomes method, Hornus et al.'s method makes usage of the same point sampling of the smooth curve that features the DNA axis [HLLF13]. But, this point sampling forms a linear approximation of the curve and that is used only for quickly visualizing the helical axis (or the DNA molecule when it is far from the viewpoint). In fact, Hornus et al.'s adaptive method makes usage of a uniform sampling of such a curve into curved segments having the same arc-length of about 3.4 Å, whose endpoints work as anchor points for DNA base pairs.

In order to coherently orient the base pairs along the curve, one uses a moving frame at every single anchor point (i.e., sampled point), being the local z-axis of the frame defined by the corresponding tangent vector. It is clear that the normal vector is computed at each sample point of the curve to obtain an orthonormal frame. This sequence of moving frames is recomputed every time the user modifies the DNA trajectory. The user is allowed to adjust the DNA helix by defining the rotation angles for specific base pairs. Note that the uniform sampling eliminates the slight discontinuities caused by adaptive sampling that outputs the piecewise linear approximation (i.e., polyline) of DNA axis.

2.4.4 Software for Adaptive Methods

In the literature, we find only a few codes that implement the DNA assembling method for arbitrary conformations, namely:

1. NAB [MC98], 1998.
2. isDNA [RG15], 2012.
3. GraphiteLifeExplorer [HLLF13], 2013.

2.4.4.1 NAB

The Nucleic Acid Builder (NAB) was presented in Section 2.3.4.3 as a WAG-based software. Interestingly, NAB was also the first software to support atomic assembly of DNA molecules along arbitrary trajectories using an adaptive approach based on the *rigid body transformations* package [MC98]. This means that NAB allows us to construct arbitrary DNA conformations using atoms as building blocks.

Recall that NAB was implemented as a C-like compiler with several new data types; more specifically, molecular types as `atom`, `residue` and `molecule`, which provide a three level molecular hierarchy, and `bound`, for use by the *distance geometry* package. Two purely geometric types, `point` (holding 3 floating point number) and `matrix` (holding a 4×4 transformation matrix), are also provided, as well as some special syntax for iterating (i.e., loop statements) over atoms of molecules.

2.4.4.2 isDNA

isDNA [RG15] is a C++/OpenGL library that implements the DNA stacking method introduced by Raposo and Gomes [RG12], and is freely available at `https://github.com/ISDNA`. The nu-

cleotide molecular surfaces used by isDNA were previously generated by the triangulation algorithm also proposed by Raposo and Gomes [RQG09]. Although isDNA did not aim to be a complete molecular visualization application, it includes a simplified GUI that allows the user to build, visualize and manipulate in 3D a few examples of DNA conformations built with this library. For example, Figures 2.14(a) and (b) were both generated using isDNA.

2.4.4.3 GraphiteLifeExplorer

GraphiteLifeExplorer incorporates a DNA assembling method due to Hornus et al.'s that was described above in Section 2.4.3. Recall that the ultimate goal of GraphiteLifeExplorer is to reconstruct a complete bacterial cell from its individual parts [HLLF13]. Being DNA one of the constituent parts of the cell, this tool enables the user to model DNA molecules of arbitrary length by modeling its helical axis as a quadratic or cubic Bézier curve in space. In addition, GraphiteLifeExplorer's graphical interface enables the user to directly modify the Bézier curve representing the DNA axis by adding, displacing, deleting and duplicating its control points to model open or closed DNA of any shape. The deformations made by the user are automatically propagated to the neighboring base pairs to get as close as possible to the canonical DNA helix.

When compared to previous approaches, only the Nucleic Acid Builder (NAB) [MC98] and the isDNA [RG12] methods are *also* able to build arbitrary DNA conformations. In the case of the isDNA method, it does not provide a mechanism of direct interaction of the user to change the trajectory of the molecules. In the case of NAB, it was able to generate a portion of DNA over a cubic Bézier curve but the control points must be specified numerically. The authors claim that, even being less accurate than, for example, w3DNA [ZLO09], GraphiteLifeExplorer is faster and gives the user more freedom for building arbitrary DNA conformations. Like the isDNA method [RG12], GraphiteLifeExplorer can also import DNA trajectories as sequences of 3D points turning these sequences into smooth interpolating curves, and like ADN-Viewer [GH00] it also provides three levels of detail: (1) DNA is displayed as a thick line; (2) DNA is displayed as a helicoidal double ribbon; and (3) DNA is displayed using the traditional atomic representation. Even being an interesting tool for studying DNA conformations, GraphiteLifeExplorer was not designed for detailed local structures studies. Finally, it is important to notice that, unlike the isDNA method [RG12], GraphiteLifeExplorer does not allow the user to input arbitrary sequences of nucleotides. Instead, it uses a repetition of the generic sequence ACTG. For further details or quality figures of GraphiteLifeExplorer, the reader is referred to [HLLF13].

2.5 Discussion

At the time of the writing this chapter we believed that it was important to clarify the differences between the two main paradigms of geometric DNA stacking: predictive methods and adaptive methods. The *predictive methods* follow either the Cambridge meeting approach [Dic89, VKD87, ST88, LS88, BB88] or the wedge angle approach [KST82, BMHT91]. Given a specific DNA base-pair sequence, these methods try to predict possible conformations of the DNA molecule. This prediction is based on statistical values for angles and displacements between DNA nucleotides. As such, these DNA assembling methods are particularly adequate for scenarios where the DNA base-pair sequence is known, but not its conformation.

Unlike predictive methods, *adaptive methods* were not thought of to predict conformations [MC98, RG12]. They use a pre-calculated DNA conformation (e.g., a conformation produced by the Monte Carlo simulation) along which they assemble base pairs according to a given sequence of nucleotides. Recall that a DNA conformation is abstracted by its DNA axis in 3D space. In other words, adaptive methods adapt a DNA base-pair sequence to any arbitrary conformation.

Nevertheless, it is necessary to be aware that both categories of methods have a geometric nature. Although the geometric parameters of DNA stacking have been obtained from chemistry laboratory experiments, they do not take into account the accentuated deformation issues and collisions between consecutive base pairs of DNA conformations.

In fact, it should be noted that while adaptive methods allow stacking of the same DNA sequence along different conformations, they do not allow their nucleotides to deform themselves when DNA changes from one conformation to another. This is so because the DNA building blocks that feature nucleotides are rigid. It is clear that this opens a window for future research in respect to the development of flexible and deformable geometric models for DNA stacking not only at the global level of the conformations or axes of DNA, but also at more localized levels of nucleotides and their atoms.

2.6 Concluding Remarks

In this chapter, we have reviewed most key geometric aspects and developments in 3D DNA stacking. Our main goal was to survey the most important DNA stacking methods (i.e., predictive and adaptive methods), as well as most relevant concepts needed to build and handle DNA molecules in 3D in terms of geometry.

DNA stacking is an important procedure to better understand the nucleic acid structure and function. Unfortunately, determining DNA structures in laboratory at high resolution (i.e., at atom level) is cumbersome, expensive, and likely intractable, in particular for huge complex systems. Hence, the need for developing computational tools for modeling and handling nucleic acid structures is nowadays more and more important, in particular those that address structural complexity and representational resolution. In this respect, DNA stacking can be considered as a coarse-graining approach (i.e., at nucleotide level) to handle DNA molecules, which allows us to mitigate somehow the complexity and resolution issues. This way, it makes it possible simulating the transitions from a DNA conformation into another in a realistic manner in real-time.

The first step towards the real-time simulation for the DNA conformational changes requires the design and implementation of a triangulation algorithm for Gaussian molecular surfaces, as needed not only to triangulate nucleotide surfaces, but also DNA molecules fully or not. As shown in the next chapter, the triangulated surfaces of nucleotides play an important role as independent building blocks in the molecular composition mechanism of DNA, also known as DNA stacking.

Related Publications

The work described in this chapter originated a survey on 3D stacking methods for DNA, which is provisionally accepted for publication as indicated below:

> Adriano N. Raposo and Abel J. P. Gomes: Computational 3D Stacking Methods for DNA: a Survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2014 (provisionally accepted for publication).

Chapter 3

Triangulation of Gaussian Molecular Surfaces

Several computational models and algorithms have been proposed for real-time rendering, realistic visualization and simulation of large biomolecules (e.g., proteins and nucleic acids). Most algorithms represent a biomolecule as a 3D *molecular surface* bounding most of the electron density field generated by the molecule. As usual in geometric modeling, the molecular surface is generally rendered using triangular meshes. The quality and resolution of these meshes determine how smooth the molecular surface looks. Recall that the mesh quality has to do with how much regular the triangles are, while the resolution refers to the number of triangles per area unit. At our best knowledge, this chapter concerns the first continuation-based triangulation algorithm for molecular surfaces formulated in terms of Gaussian functions. Furthermore, triangulations of molecular surfaces play an important role in the computation of mass properties (e.g., surface area and volume) of molecules, as required in molecular dynamics and Monte Carlo simulation methods.

3.1 Introduction

In biochemistry, one studies biomolecules such as, for example, proteins and DNA, as well as the interactions between them. These molecules are very important, for example, in drug discovery, where researchers try to find new molecules to inhibit the function of other biomolecules (usually proteins) that are responsible for many diseases. Thus, to better understand the function of large biomolecules, it is very important to study their volume and shape, because, as in many other fields, shape and function are closely related. For example, in *computer added drug discovery* (CADD), one uses graphic computational models in the search for new drugs [Wol06].

Several computational models and algorithms have been proposed for real-time rendering, realistic visualization and simulation of large biomolecules, even when they show highly dynamic behavior and deformability. Interestingly, most families of algorithms represent a biomolecule as a 3D *molecular surface* that bounds the electron density field around the molecule, despite they follow principles and utilize geometric tools that are quite distinct from each other.

In a way, independently of the mathematical formulation used to describe a molecular surface, its rendering on computer screen often requires its triangulation. In other words, given a molecular surface, we need to triangulate it in order to display it on screen. The quality of the resulting mesh is related to the regularity of its triangles, and determines the visual smoothness of the molecular surface, a very important feature for realistic molecular visualization.

In fact, we find a number of mathematical models for molecular surfaces in the literature. Molecular surfaces can be represented analytically as a patch complex of spheres (van der Walls surfaces and Lee-Richards surfaces) [LR71, KWB10] or as a patch complex of spheres and tori (Connolly surfaces) [Con83, RPK07] or as a patch complex of spheres and quadratic hyperboloids [Ede99] or as parametric B-spline or NURBs surfaces [BLMP97, BPS+03] or as level sets resulting

from summing up Gaussian atomic electron density functions (Gaussian surfaces) [Bli82, CCW06, DO93, GP95, WSS99] or as implicit algebraic splines or A-patch surfaces [ZXB07]. Of course, surface triangulations can be generated from those analytical forms of molecular surfaces. However, we do not find many triangulation algorithms for molecular surfaces in the literature, with the exception of those described in [Con85, VB93, AE96, CDES01, LB02, ZXB06, RPK09]. In contrast, we find many triangulation algorithms for surfaces in general. As Gomes et al. [GVJ+09] noted, these general triangulation algorithms divide into three families: *space partitioning*, *continuation*, and *mesh fitting*.

Our method adopts the continuation approach for molecules formulated in terms of Gaussian functions, and, at our best knowledge, it is the first of its kind to sample and triangulate the Gaussian molecular surfaces. Recall that triangulation of molecular surfaces is often required not only for the visualization of the molecules, but also for the computation of mass properties such as surface area and volume, as needed in molecular dynamics and Monte Carlo simulation methods. Since large molecules (e.g., proteins and nucleic acids) usually consist of thousands to hundreds of thousands of atoms or even millions of atoms, an efficient algorithm to triangulate a molecular surface is a critical issue [RPK09].

This chapter is organized as follows. Section 3.2 presents the work related with the triangulation of Gaussian molecular surfaces. Section 3.3 presents some important molecular surfaces background concepts. Section 3.4 presents an overview of the triangulation algorithm. Section 3.5 presents the three-dimensional Newton method used for surface sampling. Section 3.6 describes how the Newton method is used for finding a seed point of the surface. Section 3.7 shows how to predict surface points around a given point of the surface. Section 3.8 describes how to find surface points from predicted points. Section 3.9 presents the techniques used to triangulate the mesh of the molecular surface. Section 3.10 presents some results obtained using this new triangulation method. Finally, Section 3.11 concludes this chapter.

3.2 Related Work

Let us now review the three main categories of algorithms to triangulate and render general implicit surfaces (not necessarily Gaussian surfaces), namely: *space partitioning algorithms*; *mesh fitting algorithms*; and *continuation algorithms*.

3.2.1 Space Partitioning Algorithms

Space partitioning algorithms subdivide (either regularly or adaptively) the domain in $\mathbb{R}^3$ into a lattice of cells to find those that intersect the implicit surface [Blo88, HW90, LC87, MS93, SH97, VFG99]. Usually, cells are either cubes or tetrahedra. The sign of the surface-describing function at the cell vertices determines the configuration type (i.e., triangulation) inside the cell. Unlike cubes, tetrahedra generate topologically consistent triangular meshes (i.e. without ambiguities), yet with distorted triangles. These distorted triangles require some kind of post-processing procedure to enhance the quality of the resulting mesh. Even worse, the triangulation inside each cube may lead to ambiguous configurations because more than one mesh may be created for the same configuration type. Some disambiguation strategies have been proposed in the literature, including simplex decomposition, modified look-up table disambiguation, gra-

dient consistency heuristics and quadratic fit, tri-linear interpolation techniques, and recursive subdivision of space into smaller sub-cells [Blo88].

3.2.2 Mesh Fitting Algorithms

Unlike space partitioning techniques, triangulation fitting methods are not based on partitioning space into cells. Starting from a seed triangle mesh homeomorphic with a sphere that roughly approximates the implicit surface, this technique progressively adapts and deforms the mesh towards the implicit surface. The progressive deformations of the mesh leave its topology invariant, so the final triangulated surface remains homeomorphic with a sphere. As much as we know, this shrink-wrapping technique was introduced by van Overveld and Wyvill [vOW93] [vOW04] in the computer graphics field. Later on, Bottino et al. [BNvO96] extended this technique to meshes with a number of toroidal holes.

3.2.3 Continuation Algorithms

Also known as *tracking algorithms*, they iteratively create a simplicial approximation of the surface using a mesh growing scheme from a starting seed element that intersects (or belongs to) the surface [DLTW90, MM95]. This seed element depends on the type of continuation used in the triangulation: *predictor-corrector* (PC) or, alternatively, *piecewise-linear* (PL) [AG87, RB83, Rhe87, Bro98]. The PC continuation uses the tangent plane to the surface and Newton's corrector to converge successive estimate points toward the surface, whereas PL continuation uses a seed cell (e.g., a cube) that straddles the surface to compute surface points through geometric intersection of its edges with the surface, i.e. the mesh growth results from the growing of cells straddling the surface. Nevertheless, these algorithms may miss important shape details of the surface, including very small components of the surface or even isolated points, because cell size is constant. Another drawback of this technique comes from the need of having a seed point or cell for each surface component, which may be a rather difficult requirement to satisfy. Araújo and Jorge presented an adaptive polygonization algorithm for implicit surfaces that also uses a Newton corrector to overcome some limitations of traditional continuation algorithms [AJ05].

Our algorithm rightly belongs to the category of *continuation* algorithms. It tessellates the Gaussian molecular surface progressively from a seed point that works as the center of a first hexagon of triangles. The vertices of each triangle are determined by a surface sampling process based on a 3D Newton corrector. This algorithm is a follow-up of the algorithm presented in the author's Master's Thesis [Rap06]. Note that the problem of multiple components does not exist in molecular modelling because we always know that the surface is where the atoms are, even when the molecule has hetero-atoms.

3.3 Background

3.3.1 Gaussian Molecular Surface

Let us now present the basic concepts related to Gaussian molecular surfaces. Considering the real function $f(x, y, z)$, which defines a scalar field in $\mathbb{R}^3$, an *isosurface* is the set of points for which $f(x, y, z) = c$, where c is a real value. Sometimes, $f(x, y, z)$ is also called *implicit function*, while $f(x, y, z) = c$ is said to be an *implicit surface*.

Based upon the quantum mechanics principles, we consider that every atom has a surrounding region of influence called *density field*. Let $d_i(\mathbf{p})$ be the density field value of an atom i for every point $\mathbf{p} \in \mathbb{R}^3$. We can consider that the density field function for a molecule is the sum of density field functions of its atoms as follows [Bli82]:

$$D(\mathbf{p}) = \sum_{i=1}^{N} d_i(\mathbf{p}) \tag{3.1}$$

Let us consider the following density function for an atom centered at $\mathbf{c}_i$, being r_i its corresponding van der Waals radius and B the *blobbiness* parameter [ZXB06]:

$$d_i(\mathbf{p}) = e^{B\left(\frac{\|\mathbf{p}-\mathbf{c}_i\|^2}{r_i^2}-1\right)} \tag{3.2}$$

To ensure that $D(\mathbf{p})$ goes to zero, B must be negative. Thus, from (3.1) and (3.2), we have

$$D(\mathbf{p}) = \sum_{i=1}^{N} e^{B\left(\frac{\|\mathbf{p}-\mathbf{c}_i\|^2}{r_i^2}-1\right)} \tag{3.3}$$

Then, the implicit surface $f(\mathbf{p}) = 0$ that encloses the molecule can be defined as the set of points $\mathbf{p} \in \mathbb{R}^3$ whose density fields equals a threshold value T:

$$f(\mathbf{p}) = D(\mathbf{p}) - T \tag{3.4}$$

The value of the threshold is not important because there is a factor of T in the contribution of each atom. The same surface can be obtained for different values of T just by adjusting the blobbiness parameter B. Thus, in our method we considered $T = 1$. Since $\log 1 = 0$, this is a convenient value for T because the surface defined by a single atom will be a simple quadric [Bli82]. Hence,

$$f(\mathbf{p}) = D(\mathbf{p}) - 1 \tag{3.5}$$

3.3.2 Speeding up the Computation of Density Field

The kernel of a Gaussian function in unbounded, so that every single atom contributes to the value of $D(\mathbf{p})$ at every point $\mathbf{p} \in \mathbb{R}^3$. It is clear that this leads to more processing time in the computation of $f(\mathbf{p})$ (cf. Eq. (3.5)). A way of limiting the effect of the Gaussian associated to an atom in $\mathbb{R}^3$ is using influence boxes and spheres. This is particularly relevant for large molecules.

In large molecules, as it is the case of DNA molecules, we do not need to take into account all the atoms of the molecule to compute the density field value at an arbitrary point $\mathbf{p}$. In Eq. (3.1) we just need to consider atoms that are close enough to $\mathbf{p}$ because the contribution of distant atoms is negligible. This allows us to considerably reduce the final computation time of Eq. (3.5) for every single vertex of the triangulation and, consequently, of the mesh.

Considering that the entire molecule is contained in a bounding box, which can be divided into a finite set of equal cubic *influence boxes* $B_1, B_2, \ldots, B_N$ with a predefined edge length λ (see Section 3.10 and Table 3.1 for different experimental values of λ), we consider an *influence sphere* S_i centered at the center of the B_i and with radius λ, as illustrated in Figure 3.1. Note that each B_i is entirely contained in S_i because its size equals the radius of its associated influence sphere S_i, so that this sphere only intersects its neighbor influence spheres.

Each influence box contains a number of atoms that are not in any other influence box. Thus, to calculate the density field value at a point $\mathbf{p}$ inside an influence sphere S_i, we only consider the contribution of the atoms inside the influence box B_i. If $\mathbf{p}$ is in the intersection of two or more influence spheres, we calculate the density field value adding the contributions of all the atoms inside the correspondent influence spheres, i.e., if $\mathbf{p}$ is in the intersection of S_i and S_j we consider the atoms inside B_i and B_j. These intersections can occur between two to twenty seven spheres. This way, we take into account the atoms of a maximum of twenty seven influence boxes at a time.

Recall that the overlapping influence spheres guarantee the continuity of the surface. This influence box distribution mechanism may also be an advantage in the computation of deformable molecules that may require local deformations of the surface.

3.4 Algorithm Overview

The leading idea of our predictor-corrector continuation algorithm for triangulating surfaces is to incrementally perform three main stages in each iteration: prediction, correction, and triangulation. Given a vertex on the expanding mesh front, the prediction procedure calculates a number of points in a small circumference centered at such a vertex, with the circumference laying in the plane that is tangent to the surface at the same vertex (cf. steps 2-3 in Algorithm 1). Ideally, the number of these predicted points is six, so we can form a hexagon on the tangent plane, called Henderson's hexagon. Calculated the predicted points, we use the corrector to take them to the surface (cf. step 4 in Algorithm 1), which is done using the Newton corrector described in Section 3.5. Finally, we attach the new triangles to the mesh front, more specifically from the given vertex to the new calculated points belonging to the surface (cf. step 5 in Algorithm 1).

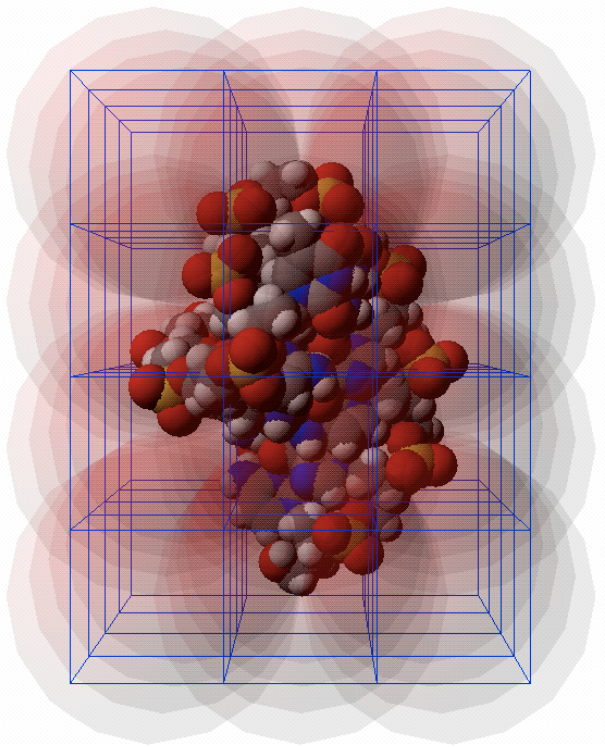


Figure 3.1: Molecule divided into influence boxes.

Algorithm 1 Triangulation of a Gaussian Molecular Surface

1: Calculate a seed point (first front vertex) on the surface
2: Determine the tangent plane at the front vertex
3: Calculate Henderson's hexagon partially or totally centered at the front vertex
4: Correct the required hexagon vertices towards the surface
5: Triangulate around the front vertex using the new sampled points obtained by correction
6: **if** $\exists$ front vertex **then**
7: go to step 2
8: **end if**

It is worthy noting that the seed vertex of the surface is also calculated using Newton's corrector (cf. step 1 in Algorithm 1). This works well because the corrector flows in the gradient field of the implicit function to a point at which it hits the molecular surface. Recall that the gradient vector is perpendicular to every single regular point of the surface. Taking into account that the molecular surface is a zero set, easily one realizes that the molecular surface works as an attractor for Newton's corrector.

3.5 Newton Corrector

Let us then revisit the three-dimensional Newton's method, whose iterative formula is as follows:

$$\mathbf{p}_{k+1} = \mathbf{p}_k - \frac{f(\mathbf{p}_k)}{\|\nabla f(\mathbf{p}_k)\|^2} \nabla f(\mathbf{p}_k). \tag{3.6}$$

where $\nabla f(\mathbf{p}_k)$ (the gradient of f at $\mathbf{p}_k \in \mathbb{R}^3$) is given by

$$\nabla f(p) = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right] \tag{3.7}$$

and

$$\frac{\partial f}{\partial v} = \sum_{i=1}^{N} \left[e^{B\left(\frac{\|p-p_i\|^2}{r_i^2}-1\right)} \cdot B \cdot \frac{r_i^2(2p_v - p_{iv})}{r_i^4} \right], \text{ with } v = x, y, z. \tag{3.8}$$

The iterative formula (3.6) is also called Newton's corrector, and produces successive estimates $\mathbf{p}_{k+1}$ that converges to a surface point $\mathbf{p}$ from a first guess $\mathbf{p_0}$. This iterative process stops when $||\mathbf{p}_{k+1} - \mathbf{p}_k|| \leq \varepsilon$ is sufficiently small, i.e. $\mathbf{p} \approx \mathbf{p}_{k+1}$.

3.6 Finding the Surface Seed Point

As usual in continuation algorithms, the first step of Algorithm 1 consists in finding the seed point of the surface, i.e., its first sampled point. It is clear that this requires to make a first guess for such a seed point. In true, the first estimate $\mathbf{p}_0$ for the seed point of the surface may be any point inside the bounding box that encloses the molecule, but, preferably, we use its centroid because it is more certain to hit the surface during the process of Newton's convergence (cf. Section 3.5) to a solution. If this process misses the surface, we try again and again from a point generated randomly inside the bounding box until the seed point is found. Recall that the triangulation of the mesh starts with a seed point p_0.

3.7 Predictor of Surface Points

Predicting the surface points around an already sampled point of the surface corresponds to the second and third steps of Algorithm 1. Such a prediction is carried out on the tangent plane to the molecular surface at every single sampled point or vertex $\mathbf{p}_0$ belonging to the boundary of the current growing mesh that approximates the surface. Note that $\mathbf{p}_0$ is the seed point when the growing mesh reduces to a single vertex.

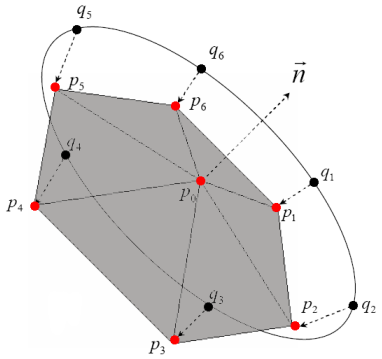


Figure 3.2: The starting Henderson's hexagon of the surface triangulation.

For the purpose of predicting surface points, we compute the orthonormal basis $(\overrightarrow{n}, \overrightarrow{t_1}, \overrightarrow{t_2})$ at $\mathbf{p}_0$, where $\overrightarrow{n}$ is the normal vector to the surface at $\mathbf{p}_0$, and $\overrightarrow{t_1}$ and $\overrightarrow{t_2}$ denote unit vectors that define the tangent plane α at $\mathbf{p}_0$. Considering a circle with radius δ on α centered in $\mathbf{p}_0$, we calculate the points $\mathbf{q}_1, \ldots, \mathbf{q}_6$ (cf. Figure 3.2) as follows:

$$\mathbf{q}_{i+1} = \mathbf{p}_0 + \delta \cos(i\pi/3)\overrightarrow{t_1} + \delta \sin(i\pi/3)\overrightarrow{t_2} \tag{3.9}$$

As illustrated in Figure 3.2, it is important to note that the points $\mathbf{q}_i$ are the vertices of a regular hexagon inscribed in a circle centered at $\mathbf{p}_0$, whose equilateral triangles are on the tangent plane to the surface at $\mathbf{p}_0$.

3.8 Surface Sampling

The six vertices $\mathbf{q}_1, \ldots, \mathbf{q}_6$ of Henderson's hexagon are not sampled points of the molecular surface (Figure 3.2). To compute their corresponding sampled points $\mathbf{p}_1, \ldots, \mathbf{p}_6$ of the molecular surface, we apply the Newton corrector to them, as we did to calculate the seed point p_0. This constitutes the fourth step of Algorithm 1.

The vertices $\mathbf{p}_1, \ldots, \mathbf{p}_6$ of the starting hexagon of the molecular surface and their connecting edges form the initial 1-dimensional boundary of the growing mesh that approximates the Gaussian molecular surface. Then, the mesh grows from the boundary of such starting hexagon. It is clear that from here on, it is not necessary to calculate all the hexagon vertices for each triangulation vertex belonging to the boundary of the growing mesh, as explained further ahead in the next section.

3.9 Surface Triangulation

Tessellating a molecular surface with nearly regular hexagons divided into six approximately identical triangular pieces originates a mesh with nearly regular triangles. This comes from the common idea that the quality of a mesh depends on the regularity of its triangles. Meshes built with approximately regular triangles are in general smoother.

3.9.1 External angle at a mesh boundary vertex

The incremental expansion of the mesh is determined by the *external angle* θ at each vertex v_i of the mesh boundary. The external angle associated with each vertex v_i is defined by the two boundary edges $\overline{v_{i-1}v_i}$ and $\overline{v_iv_{i+1}}$ incident on v_i (Figure 3.3). Let $\overrightarrow{u} = \overrightarrow{v_iv_{i-1}}$ and $\overrightarrow{v} = \overrightarrow{v_iv_{i+1}}$. By definition, $\angle(\overrightarrow{u}, \overrightarrow{v})$ is the minimum angle between $\overrightarrow{u}$ and $\overrightarrow{v}$. We need to be aware that, in this method, there are cases where the external angle θ is not equal to $\angle(\overrightarrow{u}, \overrightarrow{v})$; it is precisely $2\pi - \angle(\overrightarrow{u}, \overrightarrow{v})$. To correctly determine the value of θ, we have to check whether the normal vector $\overrightarrow{n}$ at v_i and $\overrightarrow{t} = \overrightarrow{u} \times \overrightarrow{v}$ have opposite signs or not. If $\overrightarrow{t}.\overrightarrow{n} > 0$, then the external angle $\theta = \angle(\overrightarrow{u}, \overrightarrow{v})$ (Figure 3.3 (a)); otherwise, $\theta = 2\pi - \angle(\overrightarrow{u}, \overrightarrow{v})$ (Figure 3.3 (b)). This will prevent

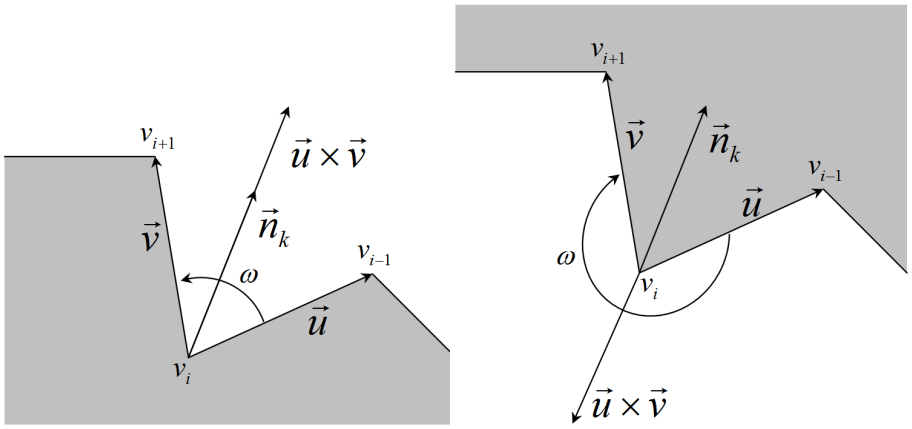


Figure 3.3: Two examples of external angles: (a) the first is less than 90°, while (b) the second is greater than 90°.

backwards triangulation, i.e., mesh overlapping around the vertex for which $\theta = 2\pi - \angle(\overrightarrow{u}, \overrightarrow{v})$.

3.9.2 Approximately uniform partition of the minimum external angle

Let $\Lambda_0 = \{v_1, v_2, \ldots, v_n\}$ be the sequence of vertices that constitutes the current mesh boundary. Also, let $\theta_1, \theta_2, \ldots, \theta_n$ be the external angles of the vertices $v_1, v_2, \ldots, v_n$, respectively. In each iteration, the mesh expansion is carried out around the vertex at which the external angle is minimum. The minimum external angle θ must be partitioned into a set of angles with approximately $\frac{\pi}{3}$ radians. This division of θ will also contribute to build a mesh with nearly regular hexagons and equilateral triangles (Figure 3.4).

To get equilateral triangles we need to divide the front angle θ in angles of exactly $\frac{\pi}{3}$ radians. This is the ideal case, but it rarely happens. In practice, we divide θ in a number of triangles with angles close to $\frac{\pi}{3}$ as much as possible. Thus, we consider a range of acceptable angles around the ideal angle $\frac{\pi}{3}$ that goes from θ_{inf} to θ_{sup}, subtracting and adding a pre-defined tolerance from and to $\frac{\pi}{3}$, respectively.

Let n_{inf} and n_{sup} be the number of triangles that result from dividing θ by θ_{inf} and θ_{sup}, re-

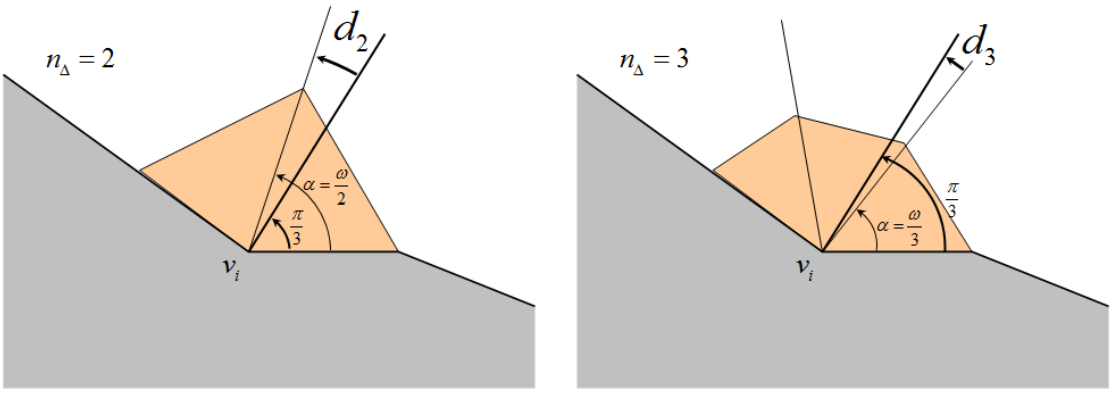


Figure 3.4: Approximately uniform angle partition.

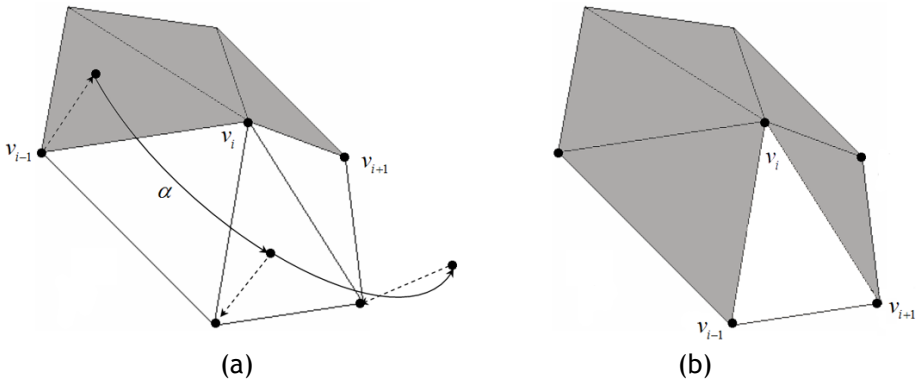


Figure 3.5: (a) Attaching two or more triangles. (b) Attaching only one triangle.

spectively. Then, we choose either n_{inf} or n_{sup} as the optimal number n_Δ of triangles, i.e., triangles whose angles are closest to $\frac{\pi}{3}$ (cf. Figure 3.4), in conformity with the following:

$$n_\Delta = \begin{cases} n_{inf} & \text{if } |\frac{\theta}{n_{inf}} - \frac{\pi}{3}| \leq |\frac{\theta}{n_{sup}} - \frac{\pi}{3}| \\ n_{sup} & \text{if } |\frac{\theta}{n_{inf}} - \frac{\pi}{3}| > |\frac{\theta}{n_{sup}} - \frac{\pi}{3}| \end{cases} \tag{3.10}$$

Two pathological situations can occur: (1) $n_\Delta = 0$; or (2) the distance $d(v_{i-1}, v_{i+1}) < \delta$. In both cases n_Δ will be set to 1, i.e. only one triangle will be attached to the mesh.

3.9.3 Mesh growth

Once calculated the number of triangles that fit θ around v_i, the current mesh is ready to grow. The molecular surface mesh grows by attaching of one or more triangles as described below:

(1) *Attaching two or more triangles*: This case is illustrated in Figure 3.5(a), and occurs when $n_\Delta \geq 2$. First, we calculate the angle $\alpha = \frac{\theta}{n_\Delta}$ of each slice triangle. Next, we compute the point that results from the orthogonal projection of v_{i-1} on a plane tangent to the surface at v_i. Then, we rotate the projected point by α radians about an axis perpendicular to the surface at the point v_i which is corrected to the molecular surface using the Newton Corrector. This procedure is repeated $n_\Delta - 2$ times.

(2) *Attaching one triangle*: The second case is illustrated in Figure 3.5(b), and occurs when $n_\Delta = 1$. In this situation, it is enough to connect v_{i-1} to v_{v+1} by a new edge to form the triangle defined by v_{i-1}, v_i and v_{v+1}.

It is clear that the mesh should not be allowed to grow indefinitely without any stopping condition; otherwise, the molecular surface would be re-triangulated repeatedly. One possible solution to this problem is described in the next section.

3.9.4 Mesh overlapping

As known, in continuation-based algorithms, it may occur a re-triangulation of the surface. To prevent the mesh from overlapping itself, we must take some precautions before any incre-

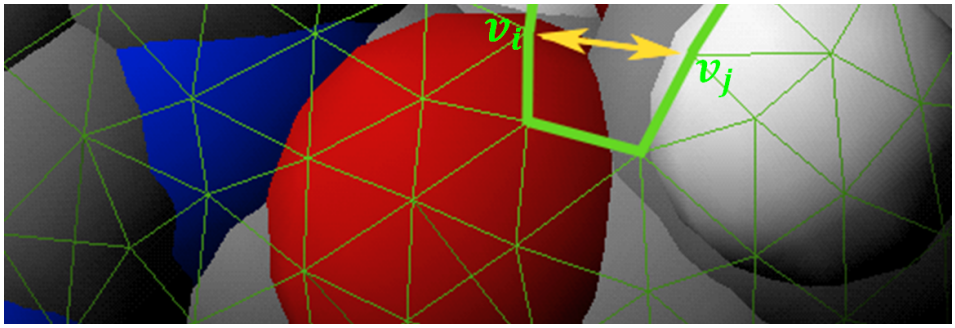


Figure 3.6: Two non-consecutive vertices belonging to the same mesh boundary (green highlighted) are near to each other.

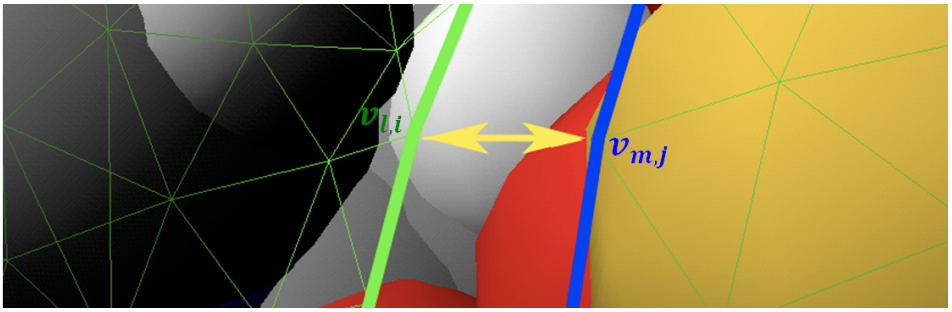


Figure 3.7: Two vertices belonging to distinct mesh boundaries (green and blue highlighted) are near to each other.

mental expansion of the mesh. This controlled behavior is based on a proximity criterion and originates two different cases as described below:

(1) *Near non-consecutive vertices of the same boundary*: This case is illustrated in Figure 3.6. Let Λ_l be a mesh boundary, and let v_i and v_j (with $i < j$) be two vertices of Λ_l with at least two vertices of Λ_l between v_i and v_j. If the Euclidean distance between v_i and v_j is less than a small δ, Λ_l splits into two new mesh boundaries, Λ_l itself and Λ_m. Thus, after splitting Λ_l, the new Λ_l is given by $\{v_1, \ldots, v_i, v_j, \ldots, v_N\}$ (where N is the number of vertices of the former Λ_l) and Λ_m consists of $\{v_i, \ldots, v_j\}$.

(2) *Near vertices belonging to distinct mesh boundaries*: This case is illustrated in Figure 3.7. Let Λ_l and Λ_m two expansion boundaries. If there is a vertex $v_{l,i} \in \Lambda_l$ and another vertex $v_{m,j} \in \Lambda_m$ such that the Euclidean distance between them is less than δ, the boundaries are going to be merged into each other. Thus, Λ_m is eliminated and its vertices are transferred into Λ_l, that is,

$$\Lambda_l = \{v_{l,1}, \ldots, v_{l,i}, v_{m,j}, \ldots, v_{m,N_m}, v_{m,1}, \ldots, v_{m,j}, v_{l,i}, \ldots, v_{l,N_l}\},$$

where N_l and N_m are, respectively, the numbers of vertices of the former Λ_l and Λ_m.

Henceforth, we can guarantee that the re-triangulation of the Gaussian molecular surface is avoided. However, putting a barrier to remeshing requires a neat control on the current mesh boundaries of the triangulation.

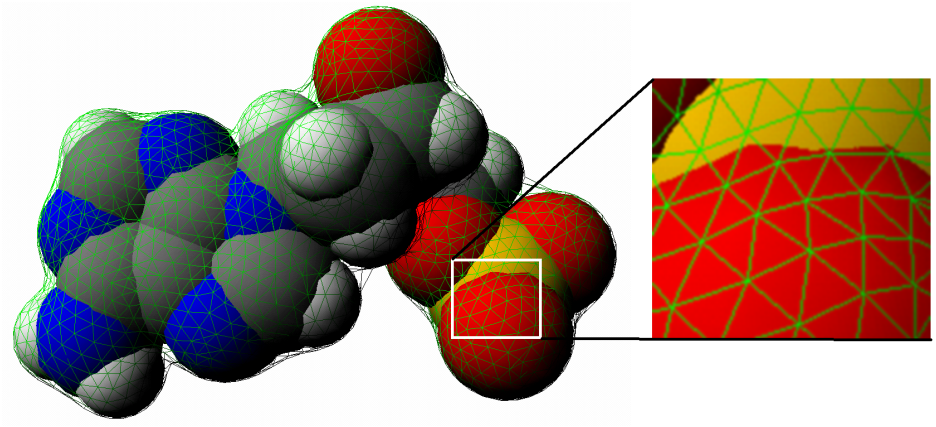


Figure 3.8: Adenine nucleotide (left) and a mesh close-up detail (right).

3.10 Results and Discussion

This section presents the most relevant results of the new triangulation algorithm. The algorithm has been coded in Java3D to running on the Mac OS X (10.4.11) operating system. All the tests were performed using a MacBook Pro with a 2.16 GHz Intel Core Duo processor, 2 GB of RAM and an ATI Radeon X1600 with 256 MB of memory.

Figure 3.8 illustrates the quality of the meshes generated by the new method, in particular for a small *adenine* nucleotide molecule, i.e., one of the four nucleotides of DNA. On the left hand side of Figure 3.8, in which the VDW model is represented by the colored spheres, it is possible to see that the mesh in green closely approximates molecule's surface. On the right hand side, where a close-up detail of the molecule is shown, we can observe the quality of the mesh as the triangles are very regular and almost equilateral. Thus, with this method there is no need to use any post-triangulation smoothing algorithm, because the visual quality of the molecular surface mesh is enough. The molecule in Figure 3.8 has 32 atoms and has been tessellated with 4970 triangles with $\delta = 0.4$ and $B = -4.0$, without using influence boxes (see Section 3.3.2).

It is clear that the algorithm was not only tested for nucleotides, but also for DNA molecules themselves. For this purpose, we have used DNA biomolecules retrieved from the Protein Data Bank (`http://http://www.rcsb.org/`) as PDB files. The resulting molecular surfaces for some of those DNA molecules, as well as their corresponding van der Waals representations, are depicted in Figure 3.9. All the triangulated surfaces have been generated with the same parameters: $\delta = 1.7$ and $B = -1.0$. The resulting molecular surfaces are very accurate and smooth, even for high values of δ (the δ parameter also defines the approximated size of the mesh triangles).

As indicated in Table 3.1, the algorithm was first tested without using influence boxes. Then, the same tests were performed using influence boxes with length λ varying in the range $[7$ Å$, 15$ Å$]$. In respect to the molecule in Figure 3.9(a), which has 411 atoms, its triangulation took 776 milliseconds for $\lambda = 7$ Å, while the computation time without the influence boxes was of 3,678 milliseconds with identical mesh quality, what represents a performance gain of 4.74x. For larger molecules as those shown in Figure 3.9(b) and (c), the gain in time was even higher: the *1kbm* molecule (664 atoms) took 1,195 milliseconds to be triangulated using influence boxes with $\lambda = 8$ Å, and 8,670 milliseconds without influences boxes; the *1pdt* molecule (2024 atoms) took 3,753 milliseconds to be triangulated using influence boxes with $\lambda = 9$ Å , and 16,298 milliseconds without influence boxes. Therefore, the speedups were 7.25x and 4.34x for the

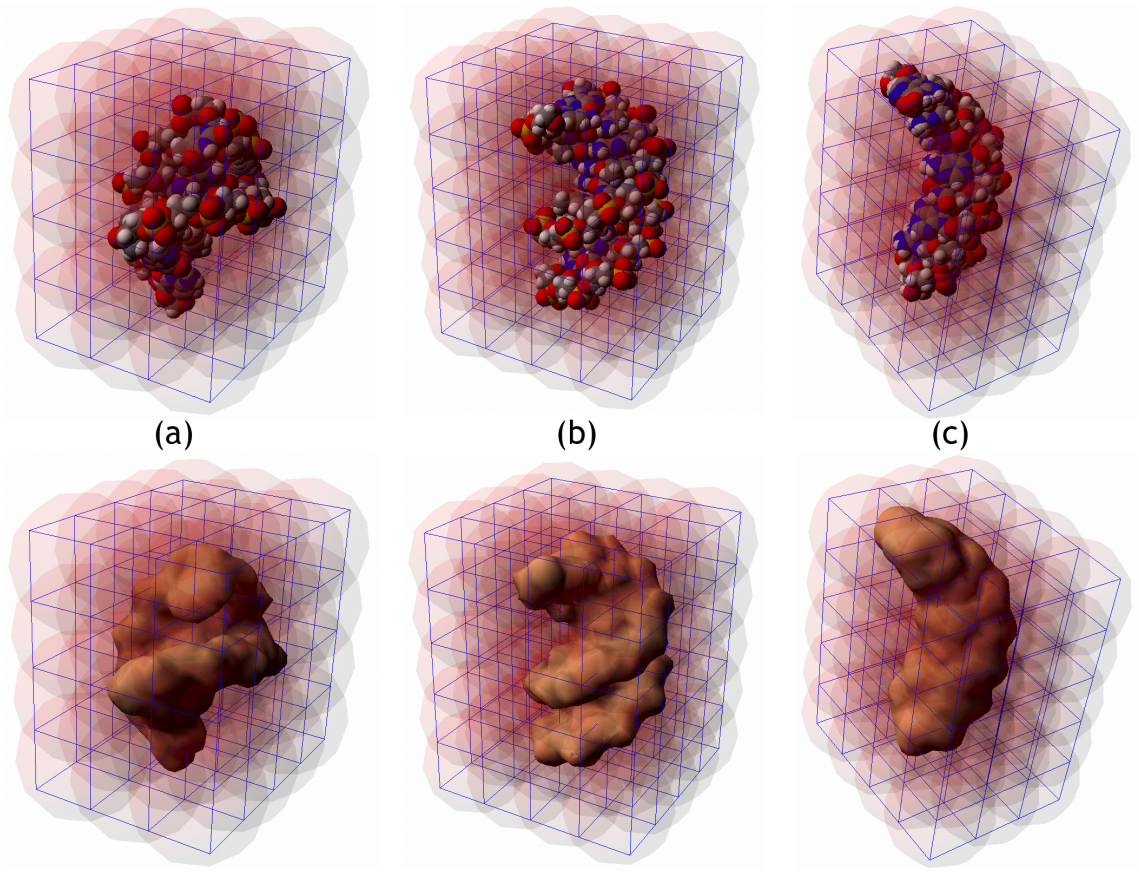


Figure 3.9: DNA molecules from the Protein Data Bank: (a) `1la8` with 411 atoms; (b) `1kbm` with 664 atoms; and (c) `1pdt` with 2024 atoms.

1kbm molecule and the *1pdt* molecule, respectively. The chart in Figure 3.10 illustrates how the computation time has been reduced as we decreased the size of the influence boxes. But, after a series of experiments, we concluded that the influence box optimal size is around $\lambda = 9$ Å, a value of tradeoff between triangulation speed and mesh quality (i.e. mesh smoothness); recall that the bigger atoms have a radius of about 2 Å.

If we compare molecules (a) and (b) in Table 3.1, for box size of 9 Å, it is interesting to verify that even with 1.6 times more atoms (411 in the first case and 664 in the second), the second molecule only takes 1.3 times more computing time than the first molecule (1195 ms for molecule (b) and 929 for molecule (a)). This happens because both molecules have a similar maximum and average number of atoms per box (Max. APB and Avg. APB). These are two important measures to evaluate the algorithm performance because, for approximately identical values of Max. APB and Avg. APB, the total number of atoms is not an issue. In the other hand, when these measures are higher, as happens for molecule (c), the performance may be slightly affected. Thus, even when two molecules approximately have the same number of atoms, it cannot be fair compare the algorithm computing times if Max. APB and Avg. APB are very different.

			Influence boxes size (λ)									
Figure	PDB ID		(*)	15 Å	14 Å	13 Å	12 Å	11 Å	10 Å	9 Å	8 Å	7 Å
(a)	1la8 (411 atoms)	Time (ms)	3678	1909	1813	1653	1462	1131	1122	975	929	**776**
		Triangles	1828	1799	1793	1735	1864	1852	1719	1752	1790	1759
		Max. APB	411	149	137	104	95	94	73	70	45	35
		Avg. APB	411	41	37	29	29	27	24	18	14	12
(b)	1kbm (664 atoms)	Time (ms)	8670	3237	2899	2869	2577	2465	1817	1641	**1195**	-
		Triangles	2808	2837	2779	2833	2724	2888	2751	2856	2869	-
		Max. APB	664	134	92	95	117	90	86	64	52	-
		Avg. APB	664	55	55	55	36	25	21	20	15	-
(c)	1pdt (2024 atoms)	Time (ms)	16298	7582	7415	6262	5279	5015	4479	**3753**	-	-
		Triangles	1737	1783	1904	1902	1783	1871	1738	1766	-	-
		Max. APB	2024	823	665	505	402	383	364	330	-	-
		Avg. APB	2024	289	289	184	168	144	126	106	-	-

(*) all atoms without box distribution
APB = atoms per box

Table 3.1: Molecules and computation details for molecular surfaces in Figure 3.9.

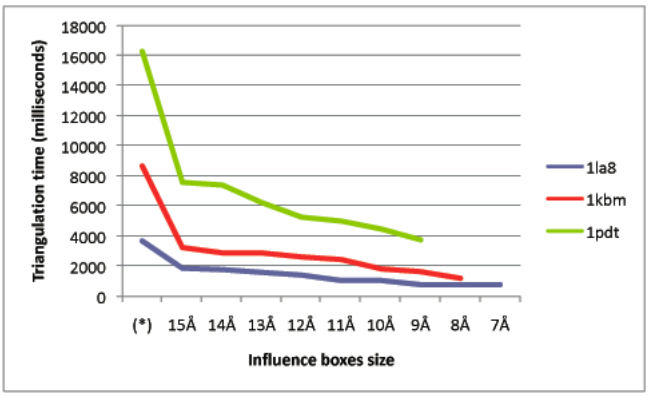


Figure 3.10: Computation times for molecules in Figure 3.9: (a) 1la8 in blue; (b) 1kbm in red; and (c) 1pdt in green. X-axis represents the size of the influence boxes (* - all atoms without influence box distribution) and Y-axis represents the triangulation time in milliseconds. See Table 3.1 for details.

3.11 Concluding Remarks

This chapter has presented a new predictor-corrector algorithm for the triangulation of molecular surfaces, in particular for Gaussian molecular surfaces. As a mesh continuation algorithm, it is computationally less expensive when compared to space-partitioning approaches (e.g., the marching cubes algorithm). Besides, because the surface sampling is performed using Newton's method, which makes use of an 1-point iterative function, so that it applies to both sign-invariant and sign-variant components of surfaces. Another advantage of this method is the smoothness of the meshes, because the tessellation of molecular surfaces tends to use nearly regular triangles. Finally, influence boxes were proposed as an efficient way to speed up the surface triangulation of both small and large molecules, as it is the case of nucleotides and DNA macromolecules, respectively.

As shown in the next chapter, this triangulation algorithm is particularly important for the DNA stacking algorithm, which is based upon the shape composition of the pre-triangulated surfaces of four types of nucleotides (and sugar-phosphates).

Related Publications

The triangulation algorithm for Gaussian molecular surfaces described in this chapter is a follow-up of the algorithm described in the following publication:

> Adriano N. Raposo and Abel J. P. Gomes: *Polygonization of multi-component non-manifold implicit surfaces through a symbolic-numerical continuation algorithm*. In Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and South-East Asia (GRAPHITE'06), Kuala Lumpur, Malaysia, November 29 - December 2, ACM Press, pp. 399-406, 2006. [RG06]

Let us also say that such triangulation algorithm for triangulating Gaussian molecular surfaces originated a paper was published in due course [RQG09], as indicated below:

> Adriano N. Raposo, João A. Queiroz, and Abel J. P. Gomes: *Triangulation of Molecular Surfaces Using an Isosurface Continuation Algorithm*. In Proceedings of the International Conference on Computational Science and Its Applications (ICCSA'09), Yongin, Korea, June 29 - July 2, IEEE Computer Society Press, pp. 145-153, 2009. [RQG09].

Chapter 4

DNA Stacking

This chapter proposes a new 3D space-filling model for sequences of DNA base pairs using nucleotides, instead of atoms, as building blocks of DNA molecules. This nucleotide-based model is more scalable than the current atomistic models, and has the advantage that easily adapts to any conformation of DNA. In fact, this nucleotide-based model allows for DNA stacking (or assembly) along arbitrary conformations, as required in Monte Carlo simulations of coiling/uncoiling of plasmid DNA. Furthermore, this model also allows the building of the molecular surface of the DNA, either partly or entirely, as needed for energy computations in molecular applications. Moreover, it allows us to grasp the DNA shape at different levels of shape composition: atom, nucleotide (or molecule), and macromolecule.

4.1 Introduction

Molecular shape determines how most biological molecules recognize and interact to one another. Also, molecular shape can be understood at different levels of abstraction and composition of shape. More specifically, our goal is to speed up the assembling and visualization of DNA molecules using molecular composition of nucleotides. Thus, unlike the traditional atomistic model for molecules, the primary building blocks of our DNA model are nucleotides, instead of atoms. In this chapter, we deal with only the predominant form of DNA found in cells, the B-form DNA, but, as seen in Chapter 2, there are other forms of DNA [BM05].

Basically, our model is structured into three levels: atomic level, molecular level, and macromolecular level. In other words, a macromolecule is a set of smaller molecules, and in turn a molecule is a set of atoms. Thus, we can say that DNA is a macromolecule composed by a number of nucleotides (i.e., molecules), each one of which holds a collection of overlapping atoms.

It is true that molecular composition and visualization based on the assembly mechanism of atoms is useful for small molecules, or even for mid range molecules as proteins, but not for macromolecules as DNA that are made up of hundreds of thousands or even millions of atoms. For example, one of the smallest DNA is that one of pUC19, which possesses about 180,000 atoms (for 2686 base pairs, one base per nucleotide), while the human DNA has about 204 billion atoms (or, equivalently, 3 billion base pairs).

Thus, in the case of DNA, we have a clear change of scale because we may have to deal with molecules having hundreds of thousands to billions of atoms. In fact, we need coarse-grain shape composition tools for DNA in order to enable:

- A faster visualization of DNA molecules at different levels of shape composition. For example, for a DNA molecule with some millions of atoms it suffices to use its helical axis for visualization purposes in large, but it may be necessary to look at a DNA fragment of

various nucleotides and their atoms locally. This also means that, in principle, we would be allowed to superimpose distinct geometric representations of molecules whenever necessary [GH00, GH02, HFGG06].

- The visualization, inspection, and testing of synthetic sequences (i.e., artificial sequences) of DNA that do not exist in nature, as required in 3D molecular nano-fabrication [GST+05] and synthetic biology [NTT07, PLWH11].

- The simulation of coiling/uncoiling behavior of plasmid DNA (e.g., pUC19) purification on computer to find the right values of temperature and salt that keep the DNA in the supercoiled state, instead of using the repetitive *in vitro* experiments in lab that take weeks in time. Note that the purification is the last step of plasmid DNA production, as needed in genetic therapy, and some genetic vaccines will be only effective if at least around 90% of produced plasmid DNA is encountered in a supercoiled conformation [STPQ05].

Nevertheless, the main focus of this chapter is on the 3D molecular assembling of DNA sequences, using nucleotides as building blocks, that adapts to different conformational changes over time as may happen, for example, in plasmid DNA purification. Therefore, the main contributions of this chapter are:

- A 3D geometric stacking method for DNA that uses pre-triangulated nucleotides, instead of pre-triangulated atoms, as a way of speeding up the assembling and visualization of DNA molecules.

- A molecular stacking method for DNA that adapts to distinct topological conformations of DNA. This is particularly important in the simulation of the coiling and uncoiling mechanisms of DNA, as needed in the DNA production for genetic therapy.

The remainder of this chapter is organized as follows. Section 4.2 presents the work related with DNA modeling. Section 4.3 introduces the DNA structure-related concepts and some of its most relevant representations. Section 4.4 briefly reviews the triangulation of Gaussian molecular surfaces, in particular those concerning nucleotides. Section 4.5 describes the DNA stacking algorithm for arbitrary conformations proposed in this thesis. Section 4.6 presents the most relevant results produced by the DNA stacking algorithm. Section 4.7 concludes the chapter providing some hints for future work.

4.2 Related Work

With the rising of computational technologies, including those engendered in databases, computer graphics, well as simulation methods, researchers and practitioners were able to use better tools for visualization and validation of DNA molecules [Car07]. As Carson noted [Car07], The Human Genome Project went three-dimensional in late 2000. But, some early attempts to computationally visualize DNA sequences as simple surfaces was presented by Carson himself a few years before [CY94]. Other DNA modeling methods have been proposed for comparative visualization of DNA sequences using the 2D projection of the DNA trajectories [VTKY07]. These methods are used in evolutionary biology to compare phylogeny of species.

Besides, over the years, static interpretation of nucleic acids was replaced by the idea that this type of macromolecules is very dynamic and permanently changing conformation. In order to cope with this non-static behavior, several approaches were developed to model DNA dynamics and mechanics [KM02]. Simulation methods, particularly *Monte-Carlo simulation* methods [CVBC05, WS02, KLT98, TM94], *molecular dynamics simulation*, and elasticity theories for both single-stranded (ssDNA) and double-stranded DNA (dsDNA) molecules [BMS06, ZZY03], have played an important role in advances in computational chemistry, computational biology, and bioinformatics.

Also, with the rebirth of the interest on gene therapy, several Monte-Carlo simulation methods have been proposed for certain DNA molecules [KP05, SBT02, TKKI98]. In these simulation approaches, DNA representation is often simplified using a curve to represent its "skeleton" or axis. This means that when a DNA molecule undergoes a conformational change its axis also changes concomitantly. As a result, the atoms must be repositioned and aligned along the axis of the new DNA conformation, yet following the original sequence of bases. Nevertheless, this assembly technique was designed for atoms, but not for nucleotides, of DNA. Apparently, all 3D DNA models found in the literature, including the most modern biological visualization systems as, for example, Chimera (`http://www.cgl.ucsf.edu/chimera`), make usage of an atom-based approach as a way of building up DNA molecules.

It is true that a similar 3D DNA model has been proposed in the literature in order to allow the DNA molecules could embody nonlinear conformations [LO03], but there is no evidence about its scalability, not to say arbitrary conformability, and nor even the ability of generating a visually realistic surface for DNA molecules. Remember that space-filling DNA surface models are required in molecular modeling and simulation because, as noted in Chapter 3, the surface plays an important role in the computing the energy of the molecule.

Finally, we believe that simulation methods, such as those of molecular dynamics, can be used together with molecular surface models to achieve better results in modeling, simulation, and visualization of molecular interactions, well as the underlying biochemical phenomena. In this respect, deformable surfaces will be surely of paramount importance in this field. Indeed, there is already some work done in dynamic maintenance of molecular surfaces in relation to conformational changes [EH05, KBE09].

4.3 DNA Structure

DNA is an helical double-stranded macro-biomolecule. Each DNA strand is a sequence of biomolecules, known as *nucleotides*, which are the DNA *building blocks* used in this chapter.

4.3.1 DNA Schematic Representation

A nucleotide consists of a nucleobase (i.e., A, C, G, or T), a five-carbon sugar (S), and one to three phosphate (P) groups, as illustrated in Fig. 4.1. It is clear that this base-sugar-phosphate structure is the same for all nucleotides; what differs is the type of base. For example, in Fig. 4.1, the blue area puts in evidence the adenine-sugar-phosphate structure of an adenine nucleotide. Besides, each base of the first strand connects to a single base of the second strand,

and vice-versa, what results in a one-to-one correspondence between every two bases, that is, we end up having a sequence of *base pairs* along the DNA.

Recall that the Watson-Crick base pairing rule states the following: (i) A matches T; (ii) C matches G [WC53]. In fact, the nucleotides of each base pair are connected (dashed lines in Fig. 4.1) to each other through their hydrogen bonds. Also, each nucleotide connects to the next nucleotide of the same strand by setting up a connection between its sugar (S) and the phosphate (P) of the next nucleotide. As shown further ahead, the sequence of DNA base pairs is instrumental to build up a 3D geometric representation for DNA molecules and their possible conformations.

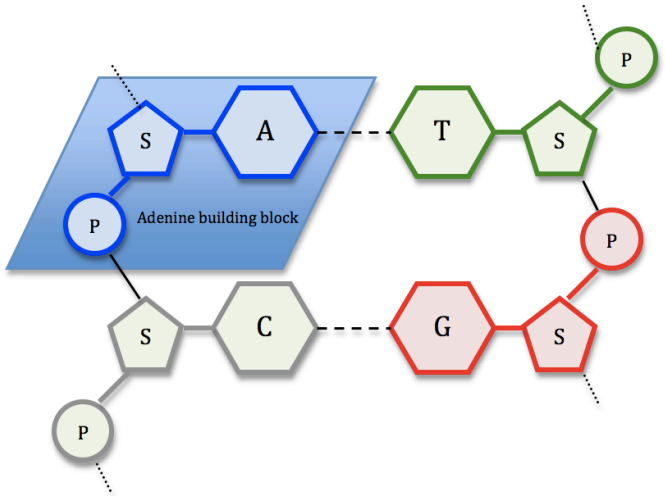


Figure 4.1: Two stacked DNA base pairs, C-G and A-T, where S stands for a five-carbon sugar and P a phosphate.

4.3.2 3D Representation of DNA

The chemical base-sugar-phosphate structure of each DNA nucleotide is well known (Fig. 4.2). On the top of Fig. 4.2 we have the traditional van der Waals (VDW) representations of the four DNA nucleotides, where atoms appear as balls with different colors that overlap geometrically. These representations include the following atoms: phosphorus (orange), oxygen (red), carbon (gray), nitrogen (blue), and hydrogen (white). We use the van der Waals radii to render atoms; 1.80 Å for phosphorus, 1.52 Å for oxygen, 1.70 Å for carbon, 1.55 Å for nitrogen, and 1.20 Å for hydrogen [Bon64].

As will be explained further ahead, the phosphorus atom is of major importance in the correct placement of each nucleotide through the procedure of assembling of DNA nucleotides. Recall that each DNA backbone consists of an alternate sequence of phosphate and sugar residues where each nucleotide is attached to the sugar of its predecessor by means of its own phosphate. At the bottom of Fig. 4.2, we see the 3D surface representation of each nucleotide. Note that a specific coloring scheme has been adopted for nucleotide surfaces to distinguish them in the visualization and inspection of DNA molecules. This coloring scheme was also designed to easily distinguish nucleotides drawn in gray-scale.

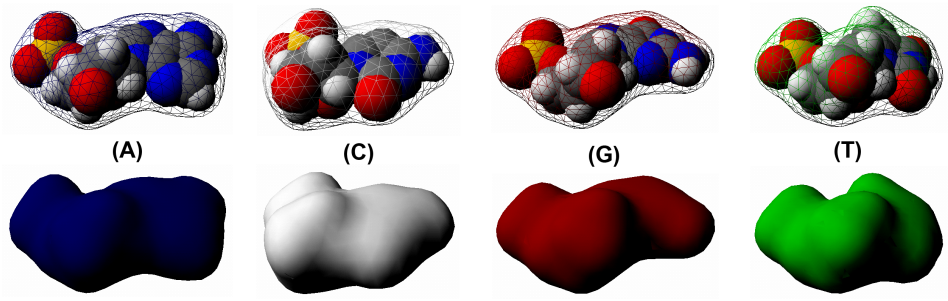


Figure 4.2: DNA nucleotides, A (adenine), C (cytosine), G (guanine), T (thymine): the standard VDW representations (top); the corresponding *non-standard* surface representations (bottom).

4.4 Triangulation of DNA Nucleotides

As presented in the previous chapter, computing the surface of a molecule is an important requirement in molecular modeling and dynamics simulation, particularly in DNA modeling. Unlike other approaches, the method presented in this chapter does not need to compute the isosurface of a DNA molecule as a whole. Instead, pre-triangulated Gaussian isosurfaces are used, one per nucleotide (see Fig. 4.2), as building blocks to generate a geometric representation of the DNA molecule from its sequence of base-pairs.

4.4.1 Gaussian Molecular Surface

Let us now briefly describe Blinn's mathematical formulation of Gaussian surfaces for nucleotides (and, in general, other molecules), well as a triangulation algorithm for such molecular surfaces described in Chapter 3. For this purpose, let us consider that the electron density field function D for a nucleotide (or even the DNA molecule itself), with n atoms, is the result from summing up n density field functions, each function per atom [Bli82]:

$$D(\mathbf{p}) = \sum_{i=1}^{n} d_i(\mathbf{p}) \tag{4.1}$$

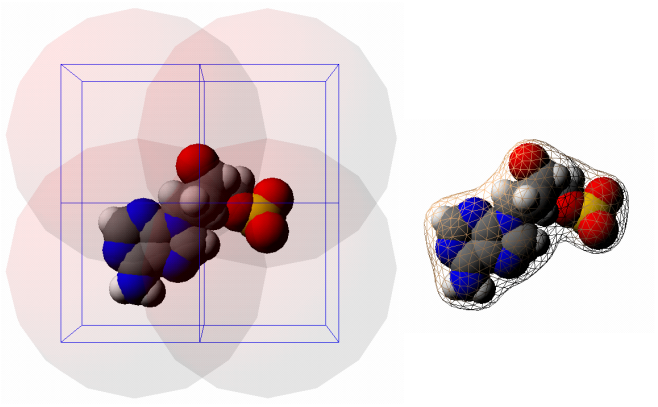
where d_i is the density function for the i-th atom, which is given by

$$d_i(\mathbf{p}) = e^{B\left(\frac{\|\mathbf{p}-\mathbf{c}_i\|^2}{r_i^2}-1\right)} \tag{4.2}$$

where $\mathbf{c}_i$ and r_i stand for the center and the van der Waals radius of the i-th atom, respectively, while B the *blobbiness* parameter [ZXB06]. Obviously, $d_i(\mathbf{p})$ only decays to zero if B were negative.

Then, replacing the expression of (4.2) into (4.1), we obtain

$$D(\mathbf{p}) = \sum_{i=1}^{n} e^{B\left(\frac{\|\mathbf{p}-\mathbf{c}_i\|^2}{r_i^2}-1\right)} \tag{4.3}$$



(a) (b)

Figure 4.3: Triangulation of the adenine building block: VDW representation divided into 4 influence boxes and 4 influence spheres (left); adenine isosurface mesh (right).

That is, the isosurface $f(\mathbf{p}) = 0$ of a DNA nucleotide can be defined as the set of points $\mathbf{p} \in \mathbb{R}^3$ where its density field equals a threshold value T:

$$f(\mathbf{p}) = D(\mathbf{p}) - T \tag{4.4}$$

We assume that T takes on the value 1 for our convenience [Bli82], because $\log 1 = 0$. Hence,

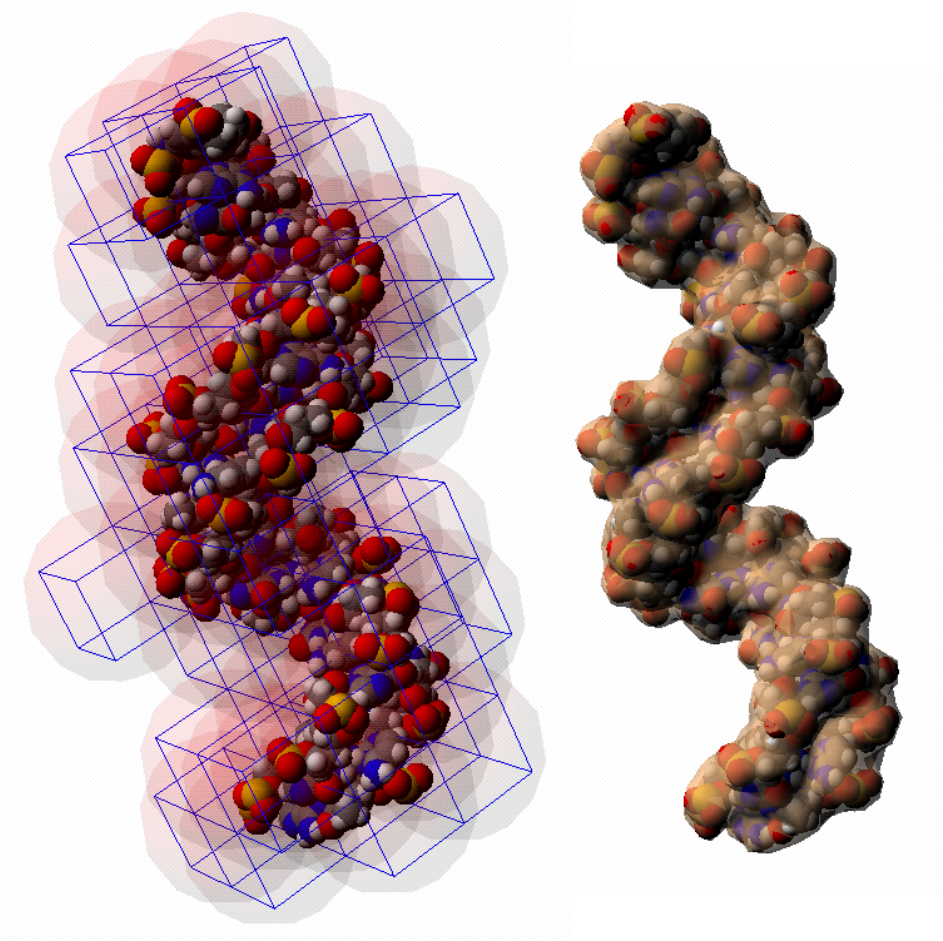
$$f(\mathbf{p}) = D(\mathbf{p}) - 1 \tag{4.5}$$

In short, the isosurface $f(\mathbf{p}) = 0$ of each type of DNA nucleotide works as a building block that can instantiated whenever needed in the DNA stacking procedure.

4.4.2 DNA Influence Boxes and Influence Spheres

In order to speed up the computation of the density field of each nucleotide, we consider that the atoms of each nucleotide are contained in a bounding box which is divided into a finite set of equal cubic *influence boxes* $B_1, B_2, \ldots, B_m$ (blue boxes in Fig. 4.3(a)) with a predefined size λ. Let us also consider an *influence sphere* S_i of radius λ (transparent pink spheres in Fig. 4.3(b)) centered at the centroid of the B_i. For example, the nucleotide molecule (adenine) depicted in Figure 4.3(a) consist of 32 atoms divided into 4 influence boxes (in blue) and influence spheres (in transparent pink), while Figure 4.3(b) shows the isosurface mesh wrapping its atoms.

Note that the radius of each sphere S_i is λ (which is also the length of the influence boxes), so that S_i is slightly bigger than B_i. This means that S_i only intersects its (possibly twenty six maximum) neighbor spheres. This guarantees the smooth blending of atomic functions locally and, consequently, the continuity and smoothness of the surface. That is, only the electron density field functions d_i of atoms inside S_i and its neighboring spheres contribute to the local value of f, hence resulting a speedup in the computation of the density field f. This is so because the Gaussian d_i is negligible beyond some distance in relation to its corresponding atom center.



(a) (b)

Figure 4.4: (a) DNA fragment of 20 base pairs: its influence boxes in blue and spheres in pink (left); (b) its transparent molecular surface (right).

It is clear that each influence box shall contain a subset of atoms. Thus, to calculate the density field value at a point $\mathbf{p}$ inside a influence sphere S_i, we have only to take into account the contributions of the atoms inside the influence box B_i. But, if $\mathbf{p}$ belongs to the intersection of two influence spheres S_i and S_j, all the atoms inside both B_i and B_j contribute to the overall density at $\mathbf{p}$. Intuitively, the overlapping influence spheres insures the continuity (and smoothness) of the nucleotide surface.

4.4.3 Molecular Surface Triangulation

Tessellating nucleotide isosurfaces with nearly regular hexagons divided into six approximately identical triangles gives rise to an almost regular mesh or triangulation. As known, meshes produced with approximately regular triangles are generally smoother [Far12].

Our triangulation algorithm belongs to the family of continuation algorithms [GVJ+09], and uses a Newton corrector to pull every new estimated vertex towards to a surface point of a specific nucleotide. Therefore, the surface of each nucleotide works as an attractor of points. For a more detailed description of the meshing algorithm, the reader is referred to Chapter 3.

It is worthy noting that this triangulation algorithm applies not only to nucleotides, but also to

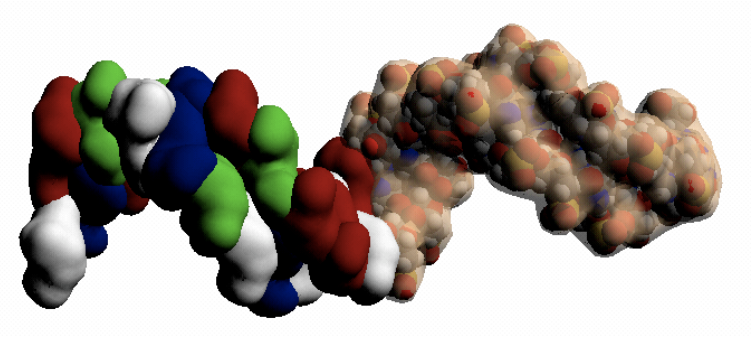


Figure 4.5: DNA fragment with two types of building blocks: Gaussian building blocks (left); and VDW building blocks with post-assembling triangulation (right).

other molecules (e.g., DNA molecules). For example, in Fig. 4.4 (b), we see the final result of the triangulation of a smooth molecular surface (with 4993 triangles) concerning a DNA fragment, having 20 bp (base pairs) and 1268 atoms, which took about 2.9 seconds in processing time. Using this approach and a semi-transparent surface, we are able of still visually identifying the atoms, even after the triangulation.

4.5 3D Stacking Algorithm for DNA

3D assembling of geometric nucleotides is at the heart of the algorithm described in this chapter, and requires the following input data:

- A geometric template (i.e., a triangulated Gaussian surface generated by the algorithm described in Chapter 3) for each one of four types of nucleotides. Each geometric template was generated from a PDB file at `http://www.nyu.edu/pages/mathmol/library/dna`. The PDB file format (`http://www.wwpdb.org/docs.html`) provides a description of the atomic coordinates and structure of molecules in general [Mey97].

- A base-pair sequence (e.g., ACCTGTTACT) of a single DNA strand retrieved from a GBK file. GBK format is a file format for encoding sequence data related to complete bacterial genomes in a computer file (`http://www.ncbi.nlm.nih.gov/genbank`) [BKML+11]. It is clear that the sequence of the complementary strand is provided by the Watson-Crick base pairing rule. The base-pair sequence dictates the assembling order of each pair of nucleotides along the DNA axis.

- An arbitrary DNA axis or, simply, a DNA axis acquired in lab experiments or predicted via simulations on computer.

The algorithm works equally well independently of the geometric template used for each nucleotide, or as a set of atom balls or as a van der Waals surface or as a Gaussian surface or as any other molecular surface found in the literature. We can even mix and superimpose distinct representations of nucleotides as illustrated in Fig. 4.5.

4.5.1 Building up an Arbitrary DNA Axis

The axis of a DNA molecule can be represented by a curve [Whi95]; for example, pUC19 is a plasmid DNA with a closed, curved axis and 2686 base pairs. For the sake of generality, arbitrary DNA conformations are used. This assumption has to do with the fact a given DNA molecule is not a static spring, but a dynamic molecule that may adopt a number of conformations; for example, plasmid DNA may appear in one of various conformations, including relaxed circular DNA (supercoils removed) and supercoiled conformations [SBT02]. It is clear that distinct DNA conformations have distinct axes. The idea is thus to allow for conformational changes of DNA over time as happens in real life [RZP09].

In order to operationalize the assembling of nucleotides, the DNA axis is approximated by a polyline whose segments have a length of $H = 3.3$ Å [BM05], which corresponds approximately to the axial distance between two consecutive nucleotides.

Algorithm 2 Building up a Segmented DNA Axis

1: $H = 3.3$
2: $\vec{v} \leftarrow (0, 0, H)$
3: $p \leftarrow (0, 0, -H/2)$
4: $q \leftarrow (0, 0, H/2)$
5: **for** $i \leftarrow 0$ to $N - 1$ **do**
6: $\quad segment_i \Leftarrow$ line segment from p to q
7: $\quad p \leftarrow q$
8: $\quad q \leftarrow q + \vec{v}$
9: **end for**
10: **for** $i \leftarrow 1$ to $N - 1$ **do**
11: $\quad$ **for** $j \leftarrow i$ to $N - 1$ **do**
12: $\quad\quad$ Rotate $segment_j$
13: $\quad$ **end for**
14: **end for**

Notice that in the construction of an arbitrary segmented DNA axis (Algorithm 2), we consider that each base pair is centered at the midpoint of each segment. That is, the number of segments equals the number N of base pairs. Thus, the first pair of bases at the origin is centered at the midpoint of the segment whose endpoints are $(0, 0, -H/2)$ and $(0, 0, +H/2)$ (steps 3 and 4 of Algorithm 2). Besides, each base pair is orthogonal to its segment, i.e., there are as many segments as pairs of building blocks (nucleotides).

We start by building a rectilinear axis divided into N segments (steps 5-9 of Algorithm 2). The second part of the algorithm fits the midpoints of these N segments to the DNA axis (steps 10-14 of Algorithm 2). This fitting procedure consists of a rotation of a segment in relation to the previous one, as a way of featuring changes in the curvature and torsion of the axis. Note that when we perform a rotation of a segment, the same rotation has to be applied to all subsequent segments.

4.5.2 Assembling of DNA Nucleotides

As said before, the assembling of DNA nucleotides is the core of the algorithm. Essentially, the algorithm iterates on the DNA base pairs A-T, T-A, C-G, and G-C in order to generate 3D geometric instances of both nucleotides of each base pair, which are subsequently assembled

on the top of the current stack of pairs of DNA nucleotides (see Fig. 4.6). In this DNA assembly procedure, we have to consider the following geometric parameters:

- *Distance between two consecutive base pairs*. This parameter measures the height H between two consecutive nucleotides of the same DNA strand. It is 3.3 Å long [BM05], and is measured between two consecutive phosphorus atoms (orange spheres in Fig. 4.2) of a DNA backbone.
- *Distance between nucleotides of a base pair*. This parameter measures the width W of the double stranded DNA, i.e., the distance that goes from one nucleotide to its mate in the same base pair. It is 20.0 Å long [BM05], and is measured between the phosphorus atoms placed side-by-side between the two DNA backbones.
- *Major and minor grooves*. As noted elsewhere, DNA backbones are closer together on one side of the helix than on the other side, i.e., the DNA helix has not an evenly coiling. As a consequence, the major groove is where the backbones are furthest away, while the minor groove is where they are closer to each other. As expected, the grooves are twisted around the biomolecule on opposite sides, i.e., they intercalate themselves along the DNA molecule. The size of the major groove is approximately 22 Å, while the minor groove is approximately 12 Å [WDT$^+$80].
- *Number of base pairs per helix turn*. Another important parameter we need to know about DNA helix is how many base pairs exist in a helix turn, i.e., the number of base pairs needed to complete a turn of the DNA helix. It is generally accepted that the DNA has 10.5 base pairs per turn [BM05].

The assembling of nucleotides can be thought of as the coiling of helicoidal DNA backbones around a curved pipe whose longitudinal axis is the DNA axis, as shown in Fig. 4.6. Note that one full turn of the DNA helix consists of 10.5 base pairs. The height of each full turn is designated by *pitch* of the DNA helix, which is $10.5 \times 3.3 = 34.65$ Å approximately.

Taking into account that one full helix turn has 2π radians, we just have to divide this value by 10.5 (i.e., the number of base pairs per turn) to calculate the rotation angle $\alpha = 2\pi/10.5$ of the next building block (or nucleotide) in relation to the current one. That is, there is an incremental rotation angle α between consecutive building blocks of the same strand (Fig. 4.7 (a)).

Each building block must be placed in a way that the position of the phosphorus atom is located on the circumference that bounds an imaginary disk of the curved pipe that is wrapped by the DNA molecule (see Fig. 4.7). Also, since the length of the major groove almost doubles the length of the minor groove, we can say that they are about $2/3$ and $1/3$ of the size of each turn of the DNA helix, respectively. This leads us to imaginatively inscribe an equilateral triangle on a disk (section of the curved pipe), and then put the building blocks of the same base pair on consecutive vertices of the triangle (see Fig. 4.7(b)).

However, these two building blocks are previously aligned to each other, so that the angle formed between their positions and the center of the disk is $2\pi/3$ radians (see steps 16 to 18 in Algorithm 3). Taking into consideration that the length of the each side of the equilateral triangle is 20 Å (i.e., the DNA width), then the radius of the circle circumscribing the triangle is $R = 20\frac{\sqrt{3}}{3}$ Å. Notice that if both building blocks of a base pair were positioned at opposite points of the circle, the major and minor grooves would be identical.

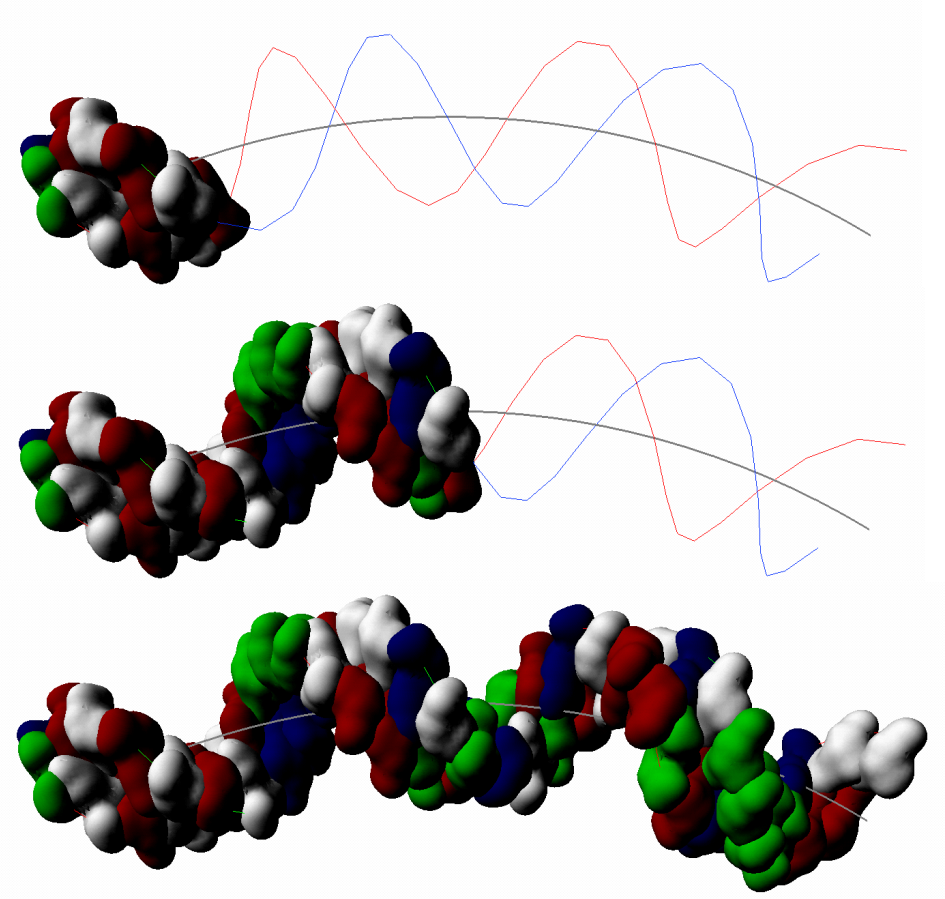
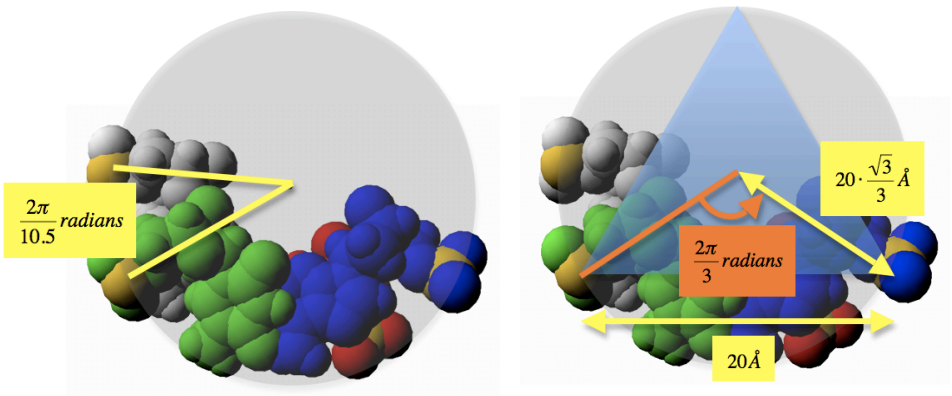


Figure 4.6: Assembling of DNA nucleotides: 5 base pairs (top), 15 base pairs (middle), and 30 base pairs (bottom), with the DNA axis in grey, and the two backbone paths in red and blue.



(a) (b)

Figure 4.7: (a) Upper view of the angle between two building blocks of the the same strand in consecutive base pairs; (b) upper view of the distances and angle between two building blocks of the same base pair.

We are now ready to start the assembly procedure of DNA nucleotides, placing the first pair of building blocks on plane $z = 0$ along the DNA axis (step 2 of Algorithm 3), which is preliminarily assumed to be the DNA axis in the geometric construction of the DNA molecule. This procedure is illustrated in Fig. 4.6, where three different assembly stages are shown for 30 base pairs DNA.

Algorithm 3 Building up the double stranded DNA

1: nucleotide templates, DNA sequence, DNA axis
2: $W \leftarrow 20; H \leftarrow 3.3; \alpha \leftarrow 2\pi/10.5$
3: $R \leftarrow \frac{\sqrt{3}}{3}W$
4: $\vec{u} \leftarrow (0, 0, 1)$
5: $o \leftarrow (0, 0, 0)$
6: **for** $i \leftarrow 0$ to N-1 **do**
7: Load nucleotide n_i
8: $b_i \leftarrow$ building block of n_i
9: $B_i \leftarrow$ building block of twin of n_i
10: $\vec{v_i} \leftarrow$ the vector of the segment i of the DNA axis
11: $m_i \leftarrow$ the midpoint of the segment i
12: Place b_i at $(-R, 0, 0)$
13: Place B_i at $(-R, 0, 0)$
14: Rotate b_i $(\alpha \times i)$ radians around $\vec{u}$ at o
15: Rotate B_i $(\alpha \times i)$ radians around $\vec{u}$ at o
16: Rotate b_i $(\pi/6)$ radians around $\vec{u}$ at o
17: Rotate B_i $(-2\pi/3)$ radians around $\vec{u}$ at o
18: Rotate B_i $(-\pi/6)$ radians around $\vec{u}$ at o
19: Rotate b_i $\angle(\vec{u}, \vec{v_i})$ radians around $(\vec{u} \times \vec{v_i})$ at o
20: Rotate B_i $\angle(\vec{u}, \vec{v_i})$ radians around $(\vec{u} \times \vec{v_i})$ at o
21: Translate b_i to position $b_i + m_i$
22: Translate B_i to position $B_i + m_i$
23: **end for**

The core of Algorithm 3 is the **for** loop (steps 6-23) that iterates on N base pairs of DNA, assembling a single base pair b_iB_i per iteration. Each iteration comprises three distinct stages:

1. *Generation of geometric instances for both nucleotides b_i and B_i.* Taking into consideration that there are only four possible base pairs, given a nucleobase n_i of a DNA strand, two geometric instances of nucleotides must be generated, the first for the building block b_i and the second for the mate building block B_i (steps 7-9).

2. *Positioning of the base pair b_iB_i on the plane $z = 0$.* This stage comprises steps 10 to 18. In the end of this stage, the base pair remains aligned with the plane $z = 0$.

3. *Alignment of the base pair b_iB_i with the plane perpendicular to segment i.* This stage involves the steps 19 and 20. Note that this alignment is also done about the origin o of the coordinate system.

4. *Translation of the base pair b_iB_i to the plane perpendicular to segment i.* Finally, because all geometric transformations are performed in relation to the origin o, b_i and B_i must be displaced to their correct positions in relation to the midpoint of the corresponding segment i of the DNA axis (steps 21-22).

At this point, it is worthy to clarify the third stage of this algorithm. The unit vector $\vec{u} = (0, 0, 1)$ is normal to the plane P_0, which is the plane $z = 0$, where the nucleotides are correctly placed on the imaginary disk referred above. Also, every segment i of the DNA axis has a normalized

vector $\vec{v_i} = (v_x, v_y, v_z)$ that is perpendicular to the plane P_i at the midpoint m_i of such a segment. Taking into consideration that $(\hat{x}, \hat{y}, \hat{z})$ is a positive oriented orthonormal basis, we know that

$$\vec{u} \times \vec{v_i} = \begin{vmatrix} \hat{x} & \hat{y} & \hat{z} \\ 0 & 0 & 1 \\ v_x & v_y & v_z \end{vmatrix} = (-v_y, v_x, 0) \tag{4.6}$$

is a vector belonging to the plane $z = 0$ at o that defines the intersection between P_0 and P_i. Of course, we know that

$$\angle(P_0, P_i) = \angle(\vec{u}, \vec{v_i}) \tag{4.7}$$

that is, we can use the vector given by the cross product $\vec{u} \times \vec{v_i}$ at o as the rotation axis, and $\angle(\vec{v}, \vec{v_i})$ as the rotation angle, to align the building blocks with the plane P_i (steps 19-20).

4.5.3 Interaction Between Nucleotides

Using Gaussian surfaces to represent atoms, nucleotides, and DNA molecules has the advantage that we can analytically determine not only their volumes, areas, and derivatives, but also their intersection volumes [GP95]. More importantly, the Gaussian representation allows us to calculate analytically all the volumes and intersection volumes by means of simple formulae that involve only exponentials. For example, the intersection volume of two or more atoms is approximated by the product of the atomic Gaussian distributions, but the same applies to intersection volumes of nucleotides. Besides, Gaussians may be used to blend the surfaces of two or more nucleotides into a single Gaussian surface. This avoids the difficulties of stitching algebraic patches of the Connolly surfaces used in molecular dynamics simulations [RPK07].

4.6 Experimental Results

The tests of the algorithm were performed in a Mac Book Pro laptop equipped with an Intel Core Duo processor at 2.16 GHz, 2 GB of RAM, and an ATI X1600 graphics card with 256 MB of memory. The DNA assembling algorithm was implemented using Java programming language (using Java3D for graphics) without using any parallel computing resources. It is also important to notice that all pictures in this chapter were generated by the software prototype, which is available to public at `https://github.com/ISDNA/isDNA` [RG15].

The tests were focused on the scalability of assembling and visualizing DNA molecules. We used the triangulation algorithm described in Chapter 3 (using the same blobbiness and threshold parameters), to compare the traditional approach of post-triangulation of the DNA molecular surface with the pre-triangulation approach of nucleotides for DNA fragments with 5, 10, 20, 40, 80 and 160 base pairs (bp), as shown in Table 4.1. The building blocks of post-triangulation approach are VDW representations of nucleotides (i.e., sets of overlapping atoms), while those of the pre-triangulation approach are triangulated Gaussian surfaces of nucleotides.

	#Base Pairs (bp)	5	10	20	40	80	160
	#Atoms	316	634	1,268	2,539	5,075	10,141
Post-Triangulation	Loading Time	10	16	26	49	52	83
	Assembly Time	49	42	51	48	67	56
	Meshing Time	583	1,271	2,777	6,061	13,228	28,869
	Total Time	642	1,329	2,853	6,158	13,347	29,008
Pre-Triangulation	Loading Time	258	266	322	437	661	1,073
	Assembly Time	86	140	222	384	727	1,458
	Meshing Time	0	0	0	0	0	0
	Total Time	344	406	544	821	1,388	2,531

Table 4.1: Computing times (milliseconds) comparison between the traditional molecular surface and the pre-triangulated isosurface building blocks approaches using the same triangulation algorithm and parameters for 5, 10, 20, 40, 80 and 160 base pairs.

From the results shown in Table 4.1, we conclude that using pre-triangulated nucleotides is far faster than triangulating the molecular surface after assembling the VDW nucleotides of the DNA molecule, in particular for large sequences of base pairs. For example, the molecular surface of the DNA sequence with 160 base pairs of VDW nucleotides took almost 30 seconds (29,008 ms) to triangulate and render, whereas the same DNA fragment was rendered in about 2.5 seconds (2,531 ms) using the pre-triangulated nucleotides. Obviously, there is an overhead in loading pre-triangulated nucleotides in memory because each pre-triangulated nucleotide has a significant number of vertices, edges and triangles.

Interestingly, there is an approximately linear ratio between the total rendering time and the number of base pairs in the post-triangulation method. In fact, by doubling the number of pairs, the total rendering time also doubles approximately. That is, it looks that the time complexity of the post-triangulation method is approximately linear. But, the average time per base pair tends to increase when the number of base pairs increases; it takes 128.4 ms/bp in rendering 5 bp, and 181.3 ms/bp to render 160 bp. This means that by increasing further the number of base pairs, we possibly end up having a super-linear or even quadratic behavior for the post-triangulation method in terms of time complexity.

On the contrary, the average rendering time per base pair of the pre-triangulated method decreases when the number of base pairs increases; the rendering of 5 bp takes 68.8 ms/bp, whereas the rendering of 160 bp takes 15.8 ms/bp. Thus, the time complexity of the pre-triangulated method is sub-linear. It is already possible to render huge DNA molecules in real time without using parallel computing facilities. For example, in Fig. 4.8, we present some arbitrary conformations for partial sequences of the pUC19. The entire pUC19 has actually 2686 bp.

4.7 Concluding Remarks

The traditional molecular models often treat DNA molecules as sets of bonded atoms, ignoring the fact that, in most of the genetic research work, a base sequence is much more important and gives much more structural information than the atomic structure of the molecule.

Thus, taking into account that DNA is primarily a sequence of pairs of nucleotides (i.e., a set of a nucleobase, a sugar and a phosphate), this chapter has introduced a novel 3D geometric

Figure 4.8: Arbitrary conformations of pUC19 partial sequences: (a) 160 bp; (b) 180 bp; (c) 400 bp; (d) 700 bp; (e) 1000 bp; and (f) 1300 bp.

assembling method for DNA that uses geometric instances, here called building blocks, of DNA nucleotides. This approach reduces the computational costs of assembling and visualization of very large DNA molecules because the building blocks are molecules (i.e., nucleotides), instead of atoms as usual in other atom-based methods.

As will be presented in the next chapter, this DNA assembly or stacking model can be applied in the study of the DNA topology and its conformations because Gaussian surfaces easily adapt to positional changes of atoms and nucleotides, as necessary in molecular modeling and simulation. More importantly, and based upon principles underlying to DNA assembling, it was possible design an algorithm to simulate conformational changes of DNA molecules, i.e., the transitions between DNA conformations, as happens in nature or in lab experiments.

Related Publications

The 3D DNA stacking algorithm described in this chapter originated a paper that was published in due course [RG12], as indicated below:

> Adriano N. Raposo and Abel J. P. Gomes: 3D molecular assembling of B-DNA sequences using nucleotides as building blocks. *Graphical Models* 74(4): 244-254 (2012). [RG12].

Chapter 5

Deformation of pDNA for Monte Carlo Simulations

Plasmid DNA molecules are closed circular molecules that are widely used in life sciences, particularly in gene therapy research. Monte Carlo methods have been used for several years to simulate the conformational behavior of DNA molecules. In each iteration these simulation methods randomly generate a new trial conformation, which is either accepted or rejected according to a criterion based on energy calculations and stochastic rules. These simulation trials are generated using a method based on crankshaft motion that, apart from some slight improvements, has remained the same for many years.

In this chapter, we present a new algorithm for the deformation of plasmid DNA molecules for Monte Carlo simulations. The move underlying our algorithm preserves the size and connectivity of straight-line segments of the plasmid DNA skeleton. We also present the results of three experiments comparing our deformation move with the standard and biased crankshaft moves in terms of acceptance ratio of the trials, energy and temperature evolution, and average displacement of the molecule. Our algorithm can also be used as a generic geometric algorithm for the deformation of regular polygons or polylines that preserves the connections and lengths of their segments.

Compared with both crankshaft moves, our move generates simulation trials with higher acceptance ratios and smoother deformations, making it suitable for real-time visualization of plasmid DNA coiling. For that purpose, we have adopted a DNA assembly algorithm that uses nucleotides as building blocks.

5.1 Introduction

Plasmid DNA (pDNA) is a family of DNA molecules widely used in life sciences, more specifically in gene therapy research. These molecules are produced inside host cells in a supercoiled conformation (i.e., their natural conformation), which is the desired conformation for therapy purposes. However, such molecules can lose their original conformation in the production and purification processes, assuming more relaxed or even linear conformations, owing to thermodynamic changes (e.g., temperature changes). One of the main challenges for researchers is to find optimal thermodynamic conditions for plasmid DNA therapeutic application without losing its supercoiled conformation or, at least, minimizing the occurrence of relaxed or open DNA molecules.

For many years, computational methods based on laboratory experimental data have been proposed to model and simulate the dynamic behavior and conformational changes in pDNA molecules under certain conditions. The Monte Carlo (MC) method has generally been accepted as a reliable tool for simulation purposes, and is seen as the standard. This iterative method tries to minimize the elastic energy of the molecule in each iteration step of the simulation process, testing the probability of acceptance of each new trial. The goal is to make the molecule

converge to an equilibrium state after performing as few iterations as possible, i.e., maximizing the acceptance ratio of the trials without compromising the effectiveness and reliability of the simulation.

To simplify the simulation process, each plasmid DNA molecule is reduced to a linear skeleton (i.e., polyline) with equal sized segments that represents the topological conformation of the molecule. Random deformations are then applied to this skeleton, generating new trial conformations, which are either accepted or rejected. Interestingly, the essence of the method used to randomly generate each new trial, referred to as the *standard* crankshaft move, has remained the same for many years, with its origins dating back to the early 1960s [VS62, HD75, Bin97], more specifically in the context of lattice polymer chains. This move was later adapted for simulation of flexible molecules like DNA using MC methods.

However, the standard crankshaft move has a very low acceptance ratio of trials, i.e., many trials are rejected. Moreover, it can present very unnatural behavior as it features very sudden motions along large portions of the molecule. To enhance the efficiency of MC moves, biasing was found to be a solution [KVA+91, VLK+92]. However, as Earl and Deem noted [ED08], biasing a deformation move implies that the probability of moving from one state to another is no longer symmetric; consequently, the acceptance rule used must be altered to maintain the detailed balance.

In this chapter, we present a new unbiased move for plasmid DNA, whose skeleton is a closed polyline. This move not only preserves the size of each segment and its connectivity, but is also very effective in maximizing the acceptance ratio of the trials and stabilizing the molecule, thereby allowing steady, gradual temperature changes during the simulation. Our method also generates natural and realistic animations that can be used in real-time simulation and visualization.

The remainder of this chapter is organized as follows. Section 5.2 presents the work related with simulation methods and generative methods of DNA conformations. Section 5.3 describes the methods behind the new move of DNA conformations. Section 5.4 presents the results obtained by the DNA deformation move introduced in this thesis. Finally, Section 5.5 concludes this chapter.

5.2 Related Work

5.2.1 Simulation Methods

In this section, we briefly review the simulation methods in computational biology and chemistry, as well as the generative methods for DNA conformations that form the core of MC methods.

5.2.1.1 Monte Carlo Simulations

The MC simulation method is one of the most important methods used in DNA simulations. This method, which was originally presented by Metropolis et al. [MRR+53], generates DNA conformations combining energy calculations, random conformational changes, and statistics.

Frank-Kamenetskii et al. [FKLV75], Vologodskii et al. [VALFK79] and Lebret [LB80] were the

first to use an MC method to present numerical results of the probability of the occurrence of knots on pDNA. Frank-Kamenetskii and Vologodskii also presented valuable information on DNA torsional rigidity [FKV81]. A few years later, Vologodskii et al. used MC simulations to study the conformational and thermodynamic properties of DNA molecules with physiological levels of supercoiling [VLK+92]. Vologodskii also included a chapter on "Monte Carlo Simulation of DNA Topological Properties" in the book "Topology in Molecular Biology" [Vol07], and with Rybenkov, they reviewed how conformational properties of DNA catenanes can be studied using MC simulations [VR09].

Gebe et al. [GACS95] presented an MC algorithm to simulate supercoiling free energies in unknotted and trefoil knotted inextensible circular chains with finite twisting and bending rigidity, while Marko et al. [VM97] made use of MC simulations to study the relationship between the amount of twisting in DNA molecules and its supercoiling.

Kundu et al. [KLT98] used an MC algorithm to explain denaturation characteristics in a supercoiled plasmid and calculate the probability of denaturation for each base pair at different supercoiling degrees.

In their work on the relationship between knots and supercoiling, Cozzareli et al. [PCV99] used an MC simulation procedure to generate an equilibrium set of conformations.

Burnier et al. used MC calculations to identify a mechanism by which topoisomerases can keep the knotting level low [BDS08].

MC simulations have also contributed to the understanding of the interplay between base-pair stacking interaction and permanent hydrogen-bond constraints in supercoiled DNA elasticity [YHZC00].

Based on the fact that atomic force microscopy has generated images of supercoiled DNA confined to a surface, which affects conformational properties such as twist and writhe, Fujimoto and Schurr modified an existing program, developed to perform MC simulations of supercoiled DNA in solution, flattening the DNA to simulate the effect of deposition on a surface [FS02]. Fujimoto and Schurr also presented a method to estimate torsional rigidities of weakly strained DNA [FBS06].

More recently, Olson et al., in their paper "How stiff is DNA?", used MC simulations to understand the behavior of a long, double-helical polymer in the tight confines of a cell and in the design of novel nanomaterials and molecular devices [ZCSO10].

5.2.1.2 Molecular Dynamics Simulations

Molecular dynamics (MD) is the second big family of simulation methods used on biomolecules. Historically, MD has not been so popular for DNA simulations as Monte Carlo methods. Although MD was first presented by Alder and Wainwright back in 1957 [AW57] (approximately 3 years after Metropolis presented his Monte Carlo paper [MRR+53]), it was not until 1983 that MD has been first applied to DNA studies by Karplus [TIBK83] and Levitt [Lev83].

However, only in the mid 1990s and in the first decade of 2000, molecular dynamic methods start to become more popular among the DNA research community. This historical delay might explain why Monte Carlo methods has been more used in DNA simulations.

According to Karplus [KM02], there are three types of applications of simulation in the macro-

molecular area:

- Simulation as means of sampling configuration space;
- Simulations to obtain a description of the system at equilibrium and the values of thermodynamic parameters;
- Simulation to examine the actual dynamics, representing the development of the system over time.

For the first two applications, we can use either MC or MD. However, according to Karplus, for the third application only MD can provide the necessary information. Besides, in contrast with the MC method, MD is a deterministic technique.

Thus, over the years, a large amount of DNA research has been done using MD simulations. The use of MD on DNA started with the study of the internal mobility of three double-stranded DNA hexamers presented by Karplus in his seminal paper entitled "Dynamics of DNA oligomers" [TIBK83].

At the same time, Levitt's published his seminal paper entitled "Computer Simulation of DNA Double-helix Dynamics", in which the author simulated the movement of atoms for DNA double helices with 12 and 24 base-pairs, providing information about the pathway of conformational changes [Lev83].

Almost two decades later, Gonzalez and Maddocks presented a method to extract a complete set of sequence-dependent energy parameters for a rigid base-pair model of DNA from MD simulations [GM01].

Langowski and colleagues also investigated the sequence-dependence deformability at base-pair level using MD simulation of two oligomers with 18 base-pairs [LSLC03].

The "Ascona B-DNA Consortium" carried out MD simulations on B-DNA oligomers containing all 136 unique tetranucleotide base sequences [BBB+04, DBC+05].

The sequence-dependent deformability was also studied by Fujii using MD to analyze the DNA conformations of all 136 possible tetrameric sequences sandwiched between CGCG sequences [FKT+07].

Kannan and Zacharias used MD simulations to study DNA double-strand dissociation and formation [KZ09].

MD was also applied to the study of DNA nanocircles. For example, Lavery and colleagues carried out MD simulations of 94 base-pair minicircles in explicit solvent with two different linking numbers, corresponding to a torsionally relaxed state and a positively supercoiled state to study how DNA behaves in string bending scenario [LLM06].

Harris and colleagues used atomistics MD simulations to investigate the effects of DNA length, sequence, salt concentration and superhelical density on the conformation of DNA nanocircles containing up to 178 base pairs [HLL08].

More recently, Orozco reviewed the most recent contributions in the study of nucleic acid flexibility using MD [ONP08]. Some advances have been presented by MacKerell and Nilsson in molecular dynamics simulations of protein-DNA complexes [MN08].

5.2.1.3 Brownian Dynamics Simulations

Originally, *Brownian dynamics* (BD) was proposed to simulated the dynamics of particles that undergo Brownian motion. Thus, based on earlier work on molecular BD simulation [DO71, EM78], Chirico and Langowski were the first authors to apply a BD algorithm to the calculation of hydrodynamic properties of simple bead-chain models of linear DNA [CL92]. A couple of years later, the same authors used BD to study the kinetics of supercoiled closed-circular DNA [CL94]. This model was later extended to include local curvature of the DNA helix axis, and to analyze the effect of a permanent bend on the structure and dynamics of a DNA superhelix [CL96].

Klenin and colleagues also presented some advances in DNA BD, developing a program [KML98] for the interpretation of solution structural and dynamic data of both linear and closed-circular DNA molecules, and for the prediction of the effect of local structural changes on the global conformation. In this approach, as in other BD simulation methods, the DNA is modeled by a chain of rigid segments interacting through harmonic spring potentials for bending, torsion, and stretching. Note that the electrostatics are handled using pre-calculated energy tables for the interactions between DNA segments as a function of relative orientation and distance. In meanwhile, Merlitz and colleagues applied BD to the study of the looping dynamics of linear DNA molecules and the effect of DNA Curvature [MRKL98]. Larson and Hu also used BD to perform simulations of a DNA molecule in an extensional flow field [LHSC99].

A few years later, Klenin and Langowski also used BD to model an irreversible biochemical intra-chain reaction of supercoiled DNA at the instant of collision between two reactive groups bound to distant DNA sites [KL01]. It this study they used a supercoiled DNA molecule of 850 nm length in dilute aqueous solution at an *NaCl* concentration of 0.1 M.

The solid phase amplification method (SPA) was presented a few years ago to amplify DNA. This method leads to the formation of small but dense DNA brushes, called DNA colonies. The replication of one of the DNA molecules in these colonies implies that molecule's free end must get out of the DNA colony to find a matching primer. Mercier and Slater used BD to model the basic kinetics of an SPA experiment [MS05].

Vologodskii tested the accuracy of a BD simulation of DNA bending motion using data from previous studies on the diffusion of knots along stretched DNA molecules [Vol06]. He simulated stretched DNA molecules with three specific types of knots and determined their diffusion coefficients. According to Vologodskii, comparison of the simulated and experimental results showed that BD simulation is capable of predicting the rates of large-scale DNA rearrangements.

More recently, Pei and colleagues simulated the diffusion of rods and wormlike chain models of duplex DNA in the size range of 100 to over 2000 base pairs in a gel modeled as a cubic lattice [PAHA09]. According to Pei and colleagues, their results showed fair agreement between modeling and experiment for duplex DNA in the size of several hundred to several thousand base pairs in an agarose gel of 2% or less. However, modeling overestimates the length dependence observed experimentally.

5.2.2 Generative Methods of DNA Conformations

It has generally been accepted that supercoiled, i.e., the self twisting of the double stranded molecule over itself, is the desired conformation for pDNA molecules [TSH88]. Thus, it is nec-

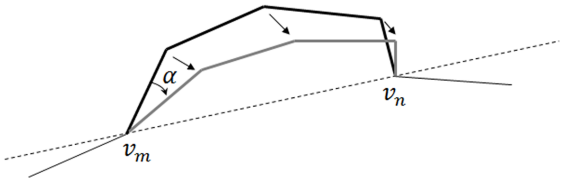


Figure 5.1: Crankshaft motion.

essary to measure the supercoiling of a given molecule. One of the most important quantitative measures of closed circular DNA supercoiling is the linking number (Lk) of the two DNA strands, which is an integer corresponding to the number of double-helical turns of the molecule.

There are several methods for calculating Lk, with one of the most widely used involving the computation of two very important geometrical properties of closed circular DNA molecules: twist (Tw) and writhe (Wr). Tw features the coiling of the two DNA strands around the axis of the helix, while Wr is a measure of the coiling of the helix axis in space [BM05]. Thus, the main result is:

$$Lk = Tw + Wr.$$

In our implementation, we used Klenin and Langowski's computation method (2a) to calculate Wr [KL00].

Owing to the nature of three-dimensional closed polylines, knots can occur in some pDNA conformations. This is not a desirable feature, i.e., each closed circular DNA molecule must remain unknotted during the simulation process, keeping its original topology even if supercoiling occurs. Knot detection methods must be used during simulation to reject possible knotted conformations. We adopted Harris-Harvey's knot detection algorithm, which uses the Alexander polynomial to detect the existence of knots [HH99]. This algorithm is based on the predicate that if two knots have different Alexander polynomials, then the knots are topologically distinct. Thus, because the Alexander polynomial of an unknotted closed circular DNA molecule is equal to one, all conformations for which this polynomial does not equal one must be rejected during the simulation.

Each trial conformation must be generated in such a way that the size and connectivity of each segment of the DNA chain do not change. A major deformation method used to displace vertices of the DNA chain was introduced by Klenin et al. [KVA+91, VLK+92]. This method, which is just a biased crankshaft move, starts by randomly choosing two vertices v_m and v_n. Then, all the vertices (and consequently all connecting segments) are rotated a randomly selected angle θ around the axis defined by the line connecting v_m and v_n, as shown in Fig. 5.1. Furthermore, following Klenin et al. [KVA+91], the value of θ is uniformly distributed over a certain interval, and must be continuously adjusted during the simulation to guarantee that about half the steps are accepted.

To increase the acceptance ratio of the simulation trials another type of motion has been proposed in the literature. This improvement performs a sub-chain translation, which is usually referred to as *reptation motion* [VLK+92], and is illustrated in Fig. 5.2. First, two vertices v_i and v_j are randomly chosen. Then, the sub-chain between v_i and v_j is translated by one segment length along the chain contour. The segment that was immediately after v_j is also translated

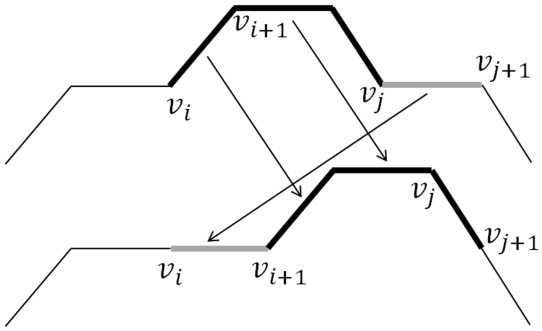


Figure 5.2: Reptation motion.

to fill the gap between v_i and v_{i+1}. This motion suggests movement analogous to a snake slithering and, hence, the name reptation motion. Other types of motion can be adopted if the Metropolis microscopic reversibility requirement is satisfied, i.e., if the probability of each trial conformation is the same as that of the reverse movement [MRR+53].

Visualization of DNA conformational changes over time is also important as part of the entire simulation of DNA behavior. This is usually performed only when the entire simulation procedure ends and is typically done by assembling the DNA atoms along the DNA axis. Interestingly, a more efficient DNA assembly algorithm was presented to allow the visualization of all the steps of the simulation procedure in real-time [RG12]. In this method, each DNA nucleotide is represented by a three-dimensional building block, allowing the assembly of the entire molecule faster, but in a realistic way. In geometric terms, each of the four building blocks featuring DNA nucleotides is a Gaussian isosurface, which was previously generated by an algorithm that triangulates molecular surfaces [RQG09].

5.3 Methods

The deformation algorithm presented in this chapter uses a linear skeleton (i.e., a polyline) with equal sized segments, henceforth called the DNA skeleton. Before introducing the core of the method itself, we explain how the DNA skeleton can be created for use by the deformation algorithm.

5.3.1 Initial Conformation of the DNA Skeleton

The DNA skeleton can assume any closed unknotted conformation. The simplest of these conformations is a completely relaxed circular conformation. Besides, the length of each segment of the DNA skeleton corresponds approximately to 30 base pairs of the double helix [Vol07].

That said, the first step of the algorithm is to determine the number of segments of the DNA skeleton ensuring around 30 base pairs per segment. Assuming that the DNA has a sequence of n base pairs, we want to find an integer s denoting the number of segments of the DNA skeleton. We define two integer parameters min and max, respectively, as the minimum and maximum numbers of base pairs that are admissible per segment, such that $min < 30 < max$. Then, for each integer value $i, min \leq i \leq max$, we calculate the corresponding $s_i = \mathrm{round}(n/i)$. Finally, we adopt $s = s_i$ as the number of segments of the DNA skeleton that minimizes $|n - (s_i \, . \, i)|$.

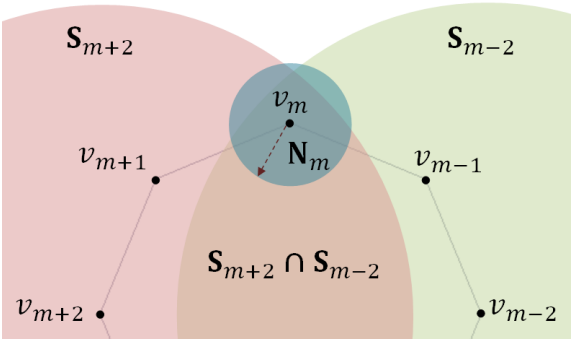


Figure 5.3: *Mobile vertex* $\mathbf{v}_m$ can be displaced randomly in the intersection of three spheres, $\mathbf{N}_m$, $\mathbf{S}_{m-2}$, and $\mathbf{S}_{m+2}$.

Once we have the number of segments s, we just have to build a regular polygon with s sides inscribed in a circle. From the number of base pairs, we can infer the approximate perimeter of the circle, as well as the corresponding radius R, from which we obtain the first vertex of the skeleton at $p_0 = (R, 0, 0)$. Then, we apply s successive rotations to p_0 about the origin to obtain all the vertices of the DNA skeleton of the initial relaxed conformation; the rotation angle is given by $\alpha = 2\pi/s$. Note that, although the initial conformation is circular, the methods for DNA assembly and deformation apply to any initial conformation.

5.3.2 Skeleton Deformation Algorithm

Assuming we have a three-dimensional closed polyline $\mathbf{P}_k$ representing the DNA skeleton, we need to deform this polyline to obtain a new polyline $\mathbf{P}_{k+1}$ with the same number of equal sized segments as $\mathbf{P}_k$, but without loss of its connectivity.

Let s be the number of segments of $\mathbf{P}_k$ and $\{\mathbf{v}_i\}$, $i = 0, ..., s-1$, its vertices. We choose a random vertex $\mathbf{v}_m$, $0 \leq m \leq s-1$ as our *mobile vertex*, i.e., the vertex with the most motion freedom in the current trial conformation. Any movement of the *mobile vertex* $\mathbf{v}_m$ implies movement of its closest neighbors $\mathbf{v}_{m-1}$ and $\mathbf{v}_{m+1}$, called *semi-mobile vertices* (Fig. 5.3). The remaining neighbors $\mathbf{v}_{m-2}$ and $\mathbf{v}_{m+2}$ are *fixed vertices* because they do not move in the deformation. Thus, in each deformation step, only three vertices will be displaced: $\mathbf{v}_m$, $\mathbf{v}_{m-1}$ and $\mathbf{v}_{m+1}$.

However, $\mathbf{v}_m$ cannot be freely displaced (Fig. 5.3). In the first instance, $\mathbf{v}_m$ moves within the sphere $\mathbf{N}_m$ centered at $\mathbf{v}_m$ with radius $r = 2\,\Delta$, where $\Delta = 3.3$ Å is the distance between two consecutive base pairs. More specifically, the new position of $\mathbf{v}_m$ is found randomly within the region resulting from the intersection of the three spheres, $\mathbf{N}_m$, $\mathbf{S}_{m-2}$ and $\mathbf{S}_{m+2}$. The latter two spheres with radius $2l$ are centered on the fixed vertices $\mathbf{v}_{m-2}$ and $\mathbf{v}_{m+2}$, respectively, where l is the length of each segment of the DNA skeleton. Note that the optimal value $2\,\Delta$ for r was found experimentally and based on the rate of successful trial conformations in the first attempt, about 30 percent, though this rate remained high for a value of r up to 3Δ, as illustrated in Fig. 5.4. The small radius r of sphere $\mathbf{N}_m$ ensures that the transition from $\mathbf{P}_k$ to $\mathbf{P}_{k+1}$ occurs without noticeable jumps.

As noted above, the new position of $\mathbf{v}_m$ was found randomly within the region $\mathbf{N}_m \cap \mathbf{S}_{m-2} \cap \mathbf{S}_{m+2}$ (Fig. 5.3), but no explanation of this random procedure was given. In fact, to calculate the new position of $\mathbf{v}_m$, we first convert its Cartesian coordinates (x, y, z) to spherical coordinates (d, θ, ϕ) relative to $\mathbf{v}_{m-2}$, where d is the distance between $\mathbf{v}_{m-2}$ and $\mathbf{v}_m$. Next, we randomly generate

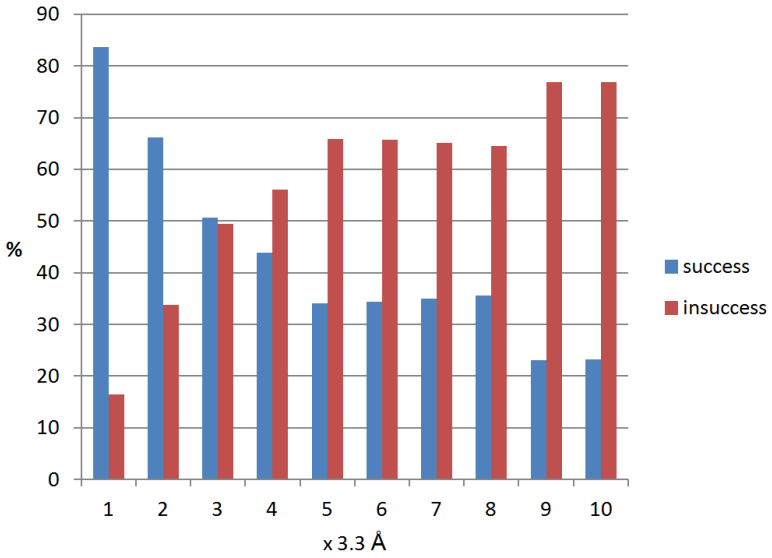


Figure 5.4: Success and insuccess ratios obtained experimentally for values of r from Δ to 10Δ.

a new position for $\mathbf{v}_m$ as $(d + \Delta d, \theta + \Delta\theta, \phi + \Delta\phi)$, where $\Delta d \in [-r, r]$ and $\Delta\theta, \Delta\phi \in [-\pi, \pi]$. It is clear that the displacement of the flanking vertices $\mathbf{v}_{m-1}$ and $\mathbf{v}_{m+1}$ depends on the previous movement of $\mathbf{v}_m$. Here we focus on the computation of the new position of $\mathbf{v}_{m-1}$ since the new position of $\mathbf{v}_{m+1}$ can be calculated similarly.

For this purpose, we also convert the Cartesian coordinates of $\mathbf{v}_{m-1}$ to spherical coordinates (l, α, β) relative to $\mathbf{v}_{m-2}$, where l is the radius of the three spheres $\mathbf{s}_m$, $\mathbf{s}_{m-1}$, and $\mathbf{s}_{m-2}$ centered on $\mathbf{v}_m$, $\mathbf{v}_{m-1}$, and $\mathbf{v}_{m-2}$, respectively. Moving $\mathbf{v}_{m-1}$ to a new position must be done in such a way that its distance l to $\mathbf{v}_{m-2}$ and $\mathbf{v}_m$ remains unchanged. In other words, the new $\mathbf{v}_{m-1}$ must lie on the circumference resulting from the intersection of the two surfaces bounding $\mathbf{s}_m$ and $\mathbf{s}_{m-2}$ (Fig. 5.5).

If $\Delta d = 0$, the new position of $\mathbf{v}_{m-1}$ relative to $\mathbf{v}_{m-2}$ is given by $(l, \alpha + \Delta\theta, \beta + \Delta\phi)$; otherwise, the new location of $\mathbf{v}_{m-1}$ is $(l, \alpha + \Delta\theta + \Delta\psi, \beta + \Delta\phi)$, where $\Delta\psi$ is the angle of the angular motion of $\mathbf{v}_{m-1}$ on $\mathbf{s}_{m-2}$ resulting from the translational displacement Δd of $\mathbf{v}_m$ along the line defined by $\mathbf{v}_m$ and $\mathbf{v}_{m-2}$ (Fig. 5.6). We compute $\Delta\psi$ by rearranging the equation that describes the

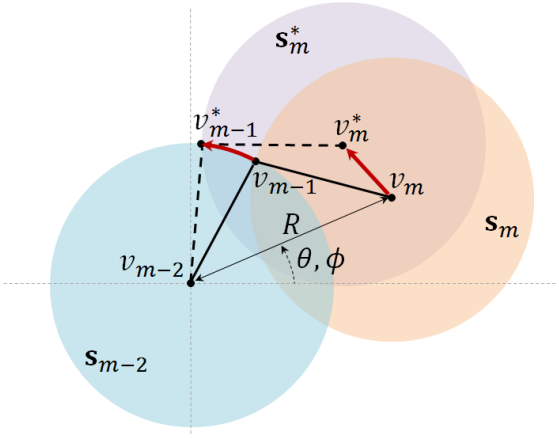


Figure 5.5: Displacement of vertices $\mathbf{v}_m$ and $\mathbf{v}_{m-1}$.

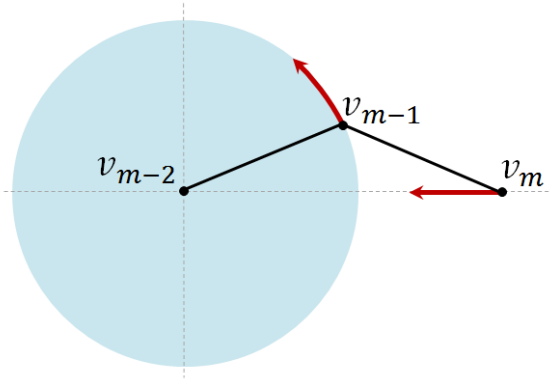


Figure 5.6: Translational piston move of vertex $\mathbf{v}_m$ translates into the rotational move of $\mathbf{v}_{m-1}$.

reciprocal motion of the piston with respect to the crank angle as follows (cf. [Hey88, p.44]):

$$\cos(\alpha + \Delta\theta + \Delta\psi) = \frac{d + \Delta d}{2\,l} \tag{5.1}$$

Note that applying the translational displacement Δd to $\mathbf{v}_m$ before the rotational motions $\Delta\theta$ and $\Delta\phi$ means that $\Delta\theta = 0$ in Eq. (5.1); otherwise, $\Delta\theta \neq 0$. In summary, moving $\mathbf{v}_m$ implies a translational and two rotational motions relative to $\mathbf{v}_{m-2}$ expressed in spherical coordinates. This causes $\mathbf{v}_{m-1}$ to rotate accordingly on the sphere centered at $\mathbf{v}_{m-2}$, with part of this rotational motion determined by the translational displacement Δd of $\mathbf{v}_m$.

These types of moves (i.e., translation and rotation) satisfy the principle of microscopic reversibility [Bol64], although this is not critical for our purposes because the simulation procedure is only used to locate energy minima. As noted by Mauri in [Mau13], for a conservative n-body system, as in the case of a DNA molecule, microreversibility stems from the invariance of the equations of motion with respect to time reversal, i.e., every microscopic motion reversing all particle velocities also results in a solution. This leads to the so-called principle of detailed balance [Weg01], which states that under stationary conditions (i.e., all probability distributions are invariant under time translation) each possible transition from one conformation to another balances itself with the reversed transition in time. In other words, the probability of obtaining trial conformation $\mathbf{P}_{k+1}$ if the current conformation is $\mathbf{P}_k$ must be equal to the probability of obtaining trial conformation $\mathbf{P}_k$ if the current conformation is $\mathbf{P}_{k+1}$ [VLK+92].

With this in mind, and having calculated the constrained position of $\mathbf{v}_{m-1}$ as a consequence of the move of $\mathbf{v}_m$, we need to determine its new position after rotating it randomly about the axis defined by $\mathbf{v}_{m-2}$ and $\mathbf{v}_m$. It is clear that the old and new locations of $\mathbf{v}_{m-1}$ lie on the circumference resulting from the intersection of spheres $\mathbf{s}_{m-2}$ and $\mathbf{s}_m$. Likewise, we find the new position of $\mathbf{v}_{m+1}$ after rotating it randomly about the axis defined by $\mathbf{v}_{m+2}$ and $\mathbf{v}_m$. Interestingly, these two rotations can be seen as two particular crankshaft rotations.

Finally, it is worth noting that the deformation algorithm described above can also be used in other biochemical systems such as internal coordinate models of cyclic peptides, as well as in some mechanical problems related to articulated arms and chain moves. In fact, this algorithm can be used to randomly deform any regular polygon (or polyline with equal sized segments) in two or three dimensions with guaranteed preservation of connectivity and the length of segments.

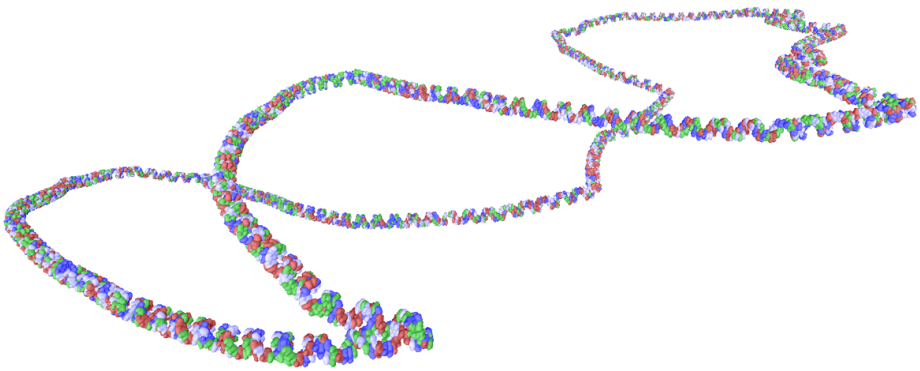


Figure 5.7: Experiment B: pUC19 after 350,000 MC steps with temperature varying between 350 K and 10 K.

5.3.3 DNA Assembly Algorithm

For a realistic visualization of closed-circular DNA simulations in real-time, what we do is to combine the new deformation algorithm described above with the DNA stacking algorithm detailed in Chapter 4.

It is important to note that this DNA stacking algorithm does not take into account the sharp kinks that may occur at the junctions of the conformation segments, as shown in Fig. 5.7. Nevertheless, a possible solution to this problem is the smoothing procedure proposed by Kummerle and Pomplun [KP05].

5.3.4 Monte Carlo Simulation

MC simulations are iterative methods based on the concept of *elastic energy* of closed circular DNA and on stochastic parameters and calculations aimed at converging to the energetic and thermodynamic equilibrium of the molecule. The main principle is to perform random DNA deformations and check whether the resulting new conformations should be accepted or rejected according to energy changes and acceptance probability. More specifically, a random deformation of the DNA is accepted if it reduces the elastic energy of the molecule or has some probability of occurring. In the experiments and results presented in this chapter, we used the same MC simulation method and parameters as those used in [KP05], where elastic energy E is calculated as

$$E = E_b + E_t. \tag{5.2}$$

Here E_b is the *bending energy* given by

$$E_b = k_B T \alpha \sum_{i=1}^{N} \Theta_i^2, \tag{5.3}$$

where k_B is the Boltzmann constant, T is the temperature, $\alpha = 2.403$ is the bending constant, and Θ_i is the angular displacement between the directions of segments i and $i+1$. *Torsional*

energy E_t is given by

$$E_t = (2\pi^2 C/L)(\Delta Lk - Wr)^2, \tag{5.4}$$

where C denotes a constant parameter known as the torsional rigidity, L is the total length of the chain, and Wr is the writhe of the skeleton. The linking number difference ΔLk in (5.4) is the difference between the linking number Lk of the DNA molecule and that, Lk_0, of its relaxed DNA conformation

$$\Delta Lk = Lk - Lk_0 = \sigma Lk_0, \tag{5.5}$$

where $-0.07 \leq \sigma \leq -0.05$ is the superhelix density parameter [Bau78].

For calculating writhe Wr, we used the method proposed in [KL00], more specifically, method 2b [LB80]. This method is based on the principle that writhe can be calculated as the difference between linking number Lk and twist Tw:

$$Wr = Lk - Tw. \tag{5.6}$$

This method for computing Wr uses an auxiliary chain close enough to the DNA skeleton, and with as many segments s_i as the DNA skeleton. Then, considering that r_i is the initial point of segment s_i, we can find the directional writhe as follows:

$$Wr_z = \sum_{i=2}^{N} \sum_{j<i} w_{ij} \tag{5.7}$$

where

$$w_{ij} = \begin{cases} sign((s_j \times s_i)(r_j - r_i)) & \text{if } \pi(s_i) \cap \pi(s_j) \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{5.8}$$

For the computation of (5.8), we must check whether segments s_i and s_j cross, i.e., whether their projections $\pi(s_i)$ and, $\pi(s_j)$ onto plane $z = 0$ intersect. We used LaMothe's algorithm to check whether the projections of these segments intersect [LaM02].

In turn, and following Klenin and Langowski [KL00], *twist* is given as

$$Tw = \frac{1}{2\pi} \sum_{i=1}^{N} [\cos^{-1}(\mathbf{a}_{i-1}.\mathbf{p}_i) - \cos^{-1}(\mathbf{p}_i.\mathbf{a}_i)] sign((\mathbf{p}_i)_z) \tag{5.9}$$

where $\mathbf{p}_i$ denotes the vector, normal to both s_{i-1} and s_i, $(\mathbf{p}_i)_z$ denotes the z-th component of vector $\mathbf{p}_i$ vector, and

$$\mathbf{a}_i = \frac{\mathbf{u} \times s_i}{|\mathbf{u} \times s_i|}, \tag{5.10}$$

where $\mathbf{u}$ is the unit vector in the z axis direction.

Then, using the results of (5.7) and (5.9), we get the final *writhe* number:

$$Wr = Wr_z - Tw. \tag{5.11}$$

Once we know how to perform the necessary energy calculations, we can apply the MC method to each iteration of the simulation to obtain a new DNA conformation from a random deformation of the DNA skeleton. Then, we calculate the energy E_{i+1} of the new candidate conformation, and compare it with the energy, E_i, of the previous conformation. The new conformation $i+1$ is accepted if $E_{i+1} < E_i$ or

$$\exp\left[(E_i - E_{i+1})/(k_B T_M))\right] > \rho \tag{5.12}$$

where T_M is the temperature of the experiment and ρ is a random value between 0 and 1 [KP05].

5.3.5 Knots Detection

It is important to note that knots can occur when random deformations are applied to DNA conformations. Because this is not desirable, i.e., DNA supercoiling must occur without generating knots, we must check for the existence of knots and reject the deformation if we find one or more knots. To optimize the performance of the method, this checking procedure is done before the MC acceptance test, avoiding unnecessary energy calculations.

For knot detection we used the method of Harris and Harvey [HH99]. In this method, based on the principle that two knots are topologically distinct if they have distinct Alexander polynomials, the DNA skeleton is converted to a knot data structure, and its Alexander polynomial is computed and compared with the Alexander polynomial of the circle, which is a trivial knot. If these two polynomials are different, the DNA skeleton contains at least one non-trivial knot.

5.4 Experiments and Results

To evaluate the effectiveness and performance of our deformation method when applied in MC simulations, we performed a set of experiments comparing our method with two types of DNA chain moves, namely, the *standard* crankshaft move and the *biased* crankshaft move.

The standard crankshaft move is a randomly chosen move. In fact, the ends, v_m and v_n, of each sub-chain are randomly chosen, as is the case with the rotating angle θ of the sub-chain around the line that passes through its ends (cf. Fig. 5.1). That is, the standard crankshaft move does not adjust the size of the sub-chain nor the rotation angle in any way. On the other hand, as in the deformation method introduced by Klenin et al. [KVA+91, VLK+92], the biased crankshaft move used here adjusts only the rotation angle. Recall that this type of biased moves is a way of enhancing the efficiency of MC moves, because it allows us to choose moves with a higher acceptance ratio [ED08].

Through these experiments we aim to demonstrate that our method generates a smoother and more controlled deformation, which leads to more consistent and even faster convergence to molecular energy equilibrium.

5.4.1 Experimental Setup

Three experiments (A, B, and C) were performed to compare the proposed method with two classic methods, namely, the standard crankshaft move and biased crankshaft move.

We used a setup based on Kummerle and Pomplun's work [KP05] for the pUC19 plasmid DNA molecule. All the three experiments were performed using the same closed circular DNA sequence with 2686 base pairs (pUC19 [YPVM85]) and exactly the same conditions and MC simulation parameters for the three Monte Carlo moves under comparison, that is: $k_B = 1.38^{-23}$; $\alpha = 2.403$; $C = 3 \times 10^{-}28$; and $\sigma = -0.04$. However, we performed experiment A at a constant temperature of 293 K, while experiments B and C were performed progressively reducing the temperature from 350 K to 10 K.

Finally, it should be noted that the efficiency of the compared methods has been measured in trial acceptance rate and not in computation time. However, it is worth mentioning that both the crankshfat and the new method have similar time performances, i.e., there is not a noticeable difference in iteration times. The average iteration time was 0.0107 seconds for the crankshaft and 0.013 seconds for the new method. Nevertheless, it is important to mention that all MC simulations were performed on an 64-bit Windows 7 laptop computer with an Intel i5 2.40GHz CPU, 4GB RAM and an Nvidia Geforce GT 520 MX 1GB graphics card.

5.4.2 Experiment A : pUC19 with Constant Temperature

In experiment A, we performed an MC simulation with 500,000 steps using the pUC19 closed circular DNA sequence at a constant temperature of 293 K as in [KP05]. This experiment was replicated using: (a) the standard crankshaft move, (b) the biased crankshaft move as described in [KVA^{+}91] and [VLK^{+}92], and (c) the proposed method. In the particular case of the biased crankshaft move, after an exhaustive optimization procedure with more than 100,000 steps, we came to the conclusion that the crankshaft rotation angle should initially be in the range $[-2.043, 2.043]$ (radians), as shown in Fig. 5.8. Furthermore, as expected, this angle decreases over time Fig. 5.9.

We used two measures to compare the efficiency of the three methods: (1) *elastic energy equilibrium* and (2) *acceptance ratio* of trials. The graphs of *elastic energy* for the three methods are shown in Fig. 5.10, where we can see that the average elastic energy for each method remains approximately the same over time.

Nevertheless, on average, the elastic energy of the proposed method is slightly higher than that of the standard crankshaft method, which in turn is higher than the energy associated with the biased crankshaft method.

On the other hand, the *acceptance ratio* of trials was evaluated for each slice of 10,000 steps from a total of 500,000 steps (see Fig. 5.11), according to the acceptance condition (5.12). The acceptance ratio was steadily higher for the proposed method, always remaining above 4,000 (and even reaching 5,000) accepted steps for each slice of 10,000 steps, i.e., an average acceptance ratio around 45%, and achieving higher ratios around 50% in the second half of the experiment. On the contrary, the acceptance ratio for the standard crankshaft move was always under 30%, and even lower in the first 10,000 steps of the experiment. With respect to the biased crankshaft move, the average acceptance ratio was slightly above 30%, but far below the results

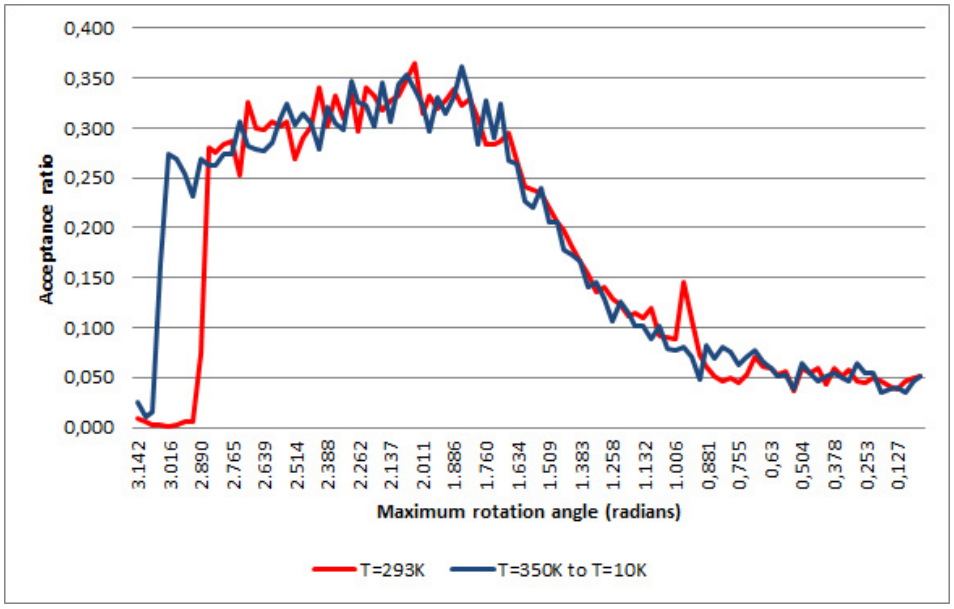


Figure 5.8: Optimization of initial angle range for 100,000 steps.

obtained using the proposed method. This indicates that our new method generates trials with much higher probabilities of being accepted under the MC simulation conditions at a constant temperature, i.e., it minimizes the number of trial rejections, and avoids useless computations.

During experiment A, we also noted that the crankshaft moves generated a few hundred conformations that were rejected owing to the existence of knots. On the contrary, our new method did not produce any knots at any time during the 500,000 simulation steps because the deformation is done smoothly and without conformational jumps. This concurs with the fact that, despite DNA molecules in living cells being long and compactly coiled, they rarely get knotted [BDS08], which suggests that supercoiling inhibits DNA knotting.

5.4.3 Experiment B : pUC19 with Variable Temperature

Experiment B also involved 500,000 MC steps. This experiment was also replicated for each of three methods analyzed in this chapter. In the particular case of the biased crankshaft move, once again, after an exhaustive optimization procedure with over 100,000 steps, we concluded that the initial crankshaft rotation angle should be in the range $[-1.854, 1.854]$ (radians) because, in experiment B, the temperature is not constant. Adjustments to this rotation angle range during the experiment are shown in Fig. 5.12.

More specifically, the temperature decreases with energy, i.e., if the average elastic energy of a 1,000-step interval is higher than that of the previous slice of 1,000 steps multiplied by a 0.9 factor, the temperature is also multiplied by a 0.9 factor. In fact, the temperature decreased progressively from 350 K to 10 K. As expected, the closer the method converges to the energy equilibrium, the greater is the decrease in temperature. As in experiment A, the acceptance ratio of the proposed deformation method was always higher than those of the classic deformation methods (see Fig. 5.13).

It was a somewhat surprising to observe how the energy decreased during the simulation. As shown in Fig. 5.14(top), when using the proposed method, the elastic energy of the molecule

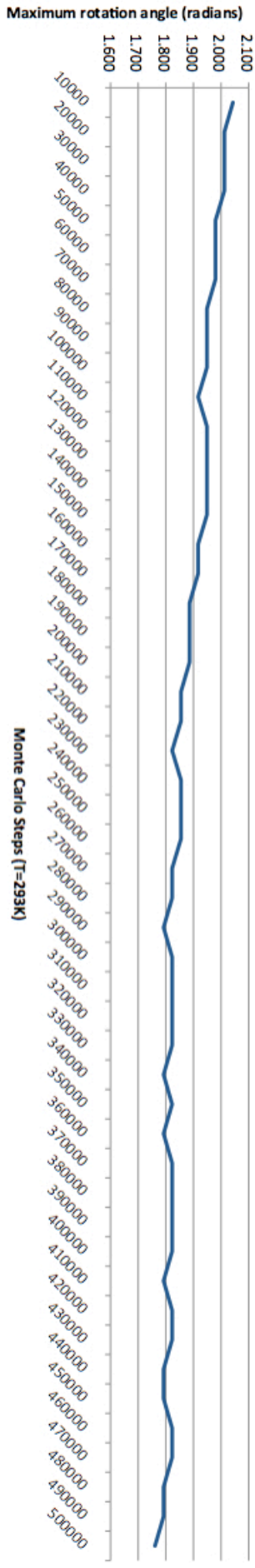


Figure 5.9: Experiment A: Crankshaft rotation angle for 500,000 steps at a constant temperature of 293K.

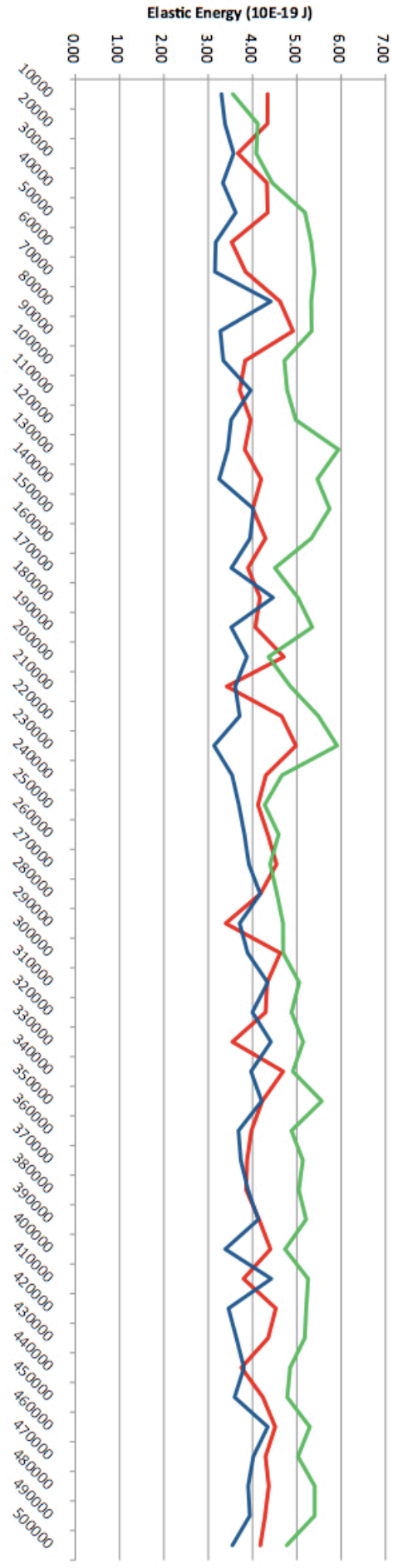


Figure 5.10: Experiment A: Elastic energy for 500,000 steps at a constant temperature of 293K.

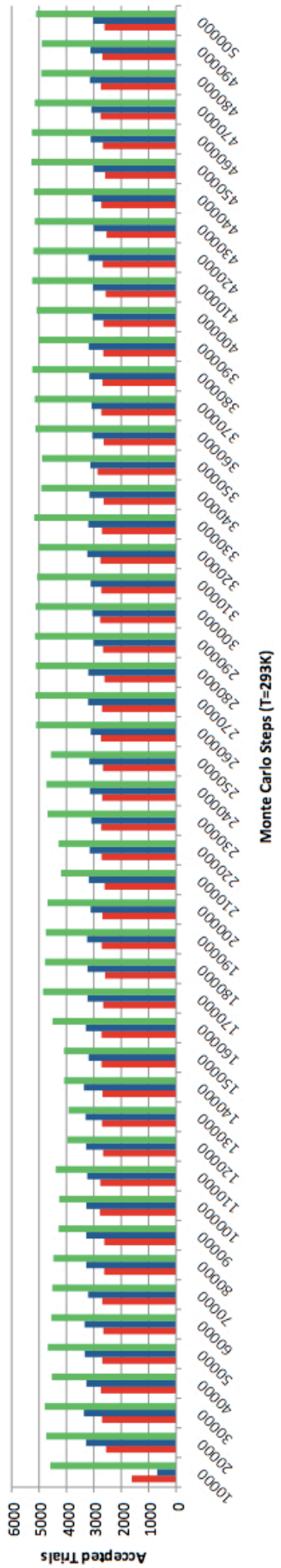


Figure 5.11: Experiment A: Acceptance trials for slices of 10,000 steps from a total of 500,000 steps at a constant temperature of 293 K.

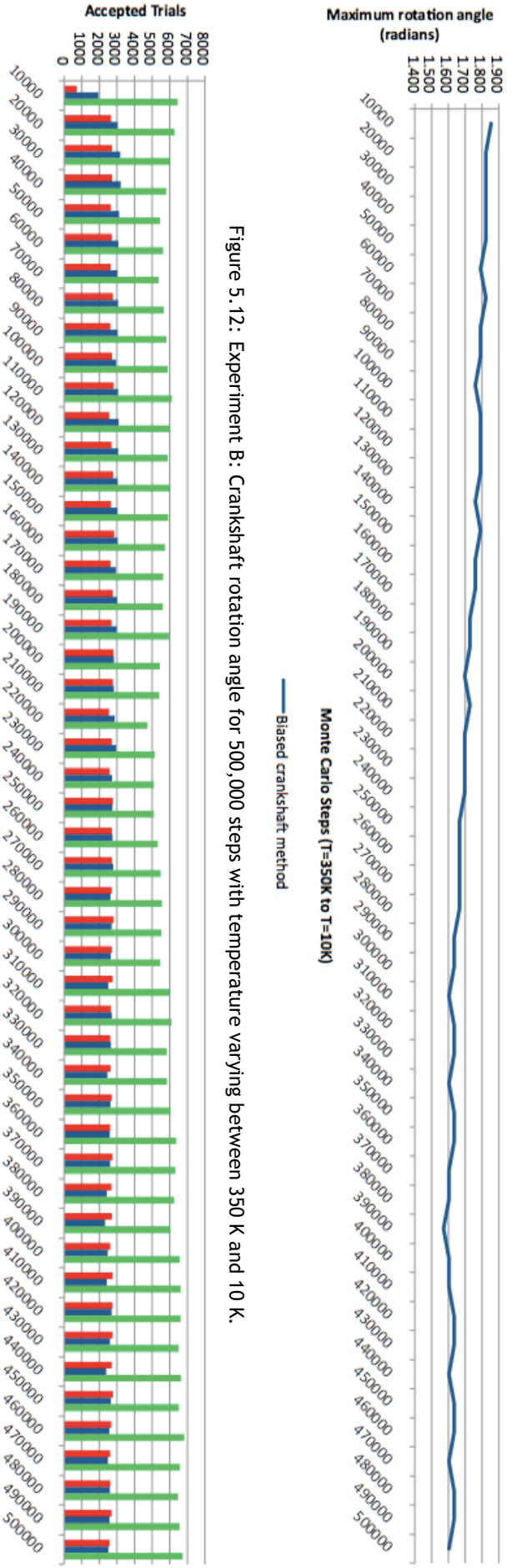


Figure 5.12: Experiment B: Crankshaft rotation angle for 500,000 steps with temperature varying between 350 K and 10 K.

Figure 5.13: Experiment B : Acceptance trials for slices of 10,000 steps from a total of 500,000 steps with temperature varying between 350 K and 10 K.

converged sooner and more consistently to equilibrium. As shown, we achieved energy equilibrium after approximately 80,000 MC steps, while the crankshaft moves only stabilized after 160,000 steps. Besides, the standard crankshaft move generated a number of very slight energy jumps, i.e., the energy did not decay as consistently as in the proposed method. However, the energy level at equilibrium was the same for all three methods, approximately 0.14×10^{-19}.

No less meaningful was the temperature decay during this experiment. As presented in Fig. 5.14 (bottom), when using the proposed method, the MC temperature decreased more rapidly and in a more consistent way, i.e., the graph for the new method is much smoother with the advantage of reaching the equilibrium temperature sooner. Fig. 5.7 shows the final result of experiment B.

In summary, we can say that the proposed deformation method requires fewer simulation steps to achieve energy equilibrium, largely owing to its high acceptance ratio.

5.4.4 Experiment C : Average Displacements

In experiment C, we set out to measure the amount of deformation of plasmid DNA caused by each type of MC move. This was accomplished by computing the average displacement of the DNA skeleton vertices for each accepted trial. In this experiment, we only considered the standard crankshaft move and the move proposed in this chapter. Taking into account vertices $v_i, v_{i+1}, ..., v_{i+n}$ that are displaced during a simulation trial, we determined the distances $d_i, ..., d_{i+n}$ between the new positions and the previous positions of these vertices, and straightforwardly computed the average displacement given by $(d_i + ... + d_{i+n})/(n + 1)$. Finally, we considered the accumulated displacement for a slice of steps as the sum of the average displacements of the accepted trials of that slice.

More specifically, we performed a 5,000-step simulation for each of the methods, namely, the standard crankshaft move and the proposed move. As shown in Fig. 5.15, the new move generates much smaller average displacements than the standard crankshaft move. Besides, from Fig. 5.16 we can see that the new move generates displacements right from the start of the simulation, whereas the standard crankshaft move starts to produce displacements later. This can be explained by the high acceptance ratio of the new method, as well as its more steady deformations.

Moreover, from Fig. 5.15, we conclude that smaller displacements in each trial do not mean there will be smaller accumulated displacements. The accumulated displacements of the new method form a logarithmic curve, while the curve of the standard crankshaft move is clearly exponential (cf. Fig. 5.16). This means that in the new method, the closer we get to the point of energy equilibrium, the shorter is the displacement toward a stable conformation, i.e., the average displacement for each accepted trial converges to zero as the conformation converges to the equilibrium. This is not the case in the standard crankshaft move, as illustrated by the accentuated variations in average displacement of the trials in Fig. 5.15.

5.4.5 Discussion

As mentioned above, we used the same molecule (i.e., pUC19), the same conditions/parameters, and the same MC simulation method in experiments A, B, and C. For comparison purposes, each experiment was performed using three different deformation methods: (a) the *standard*

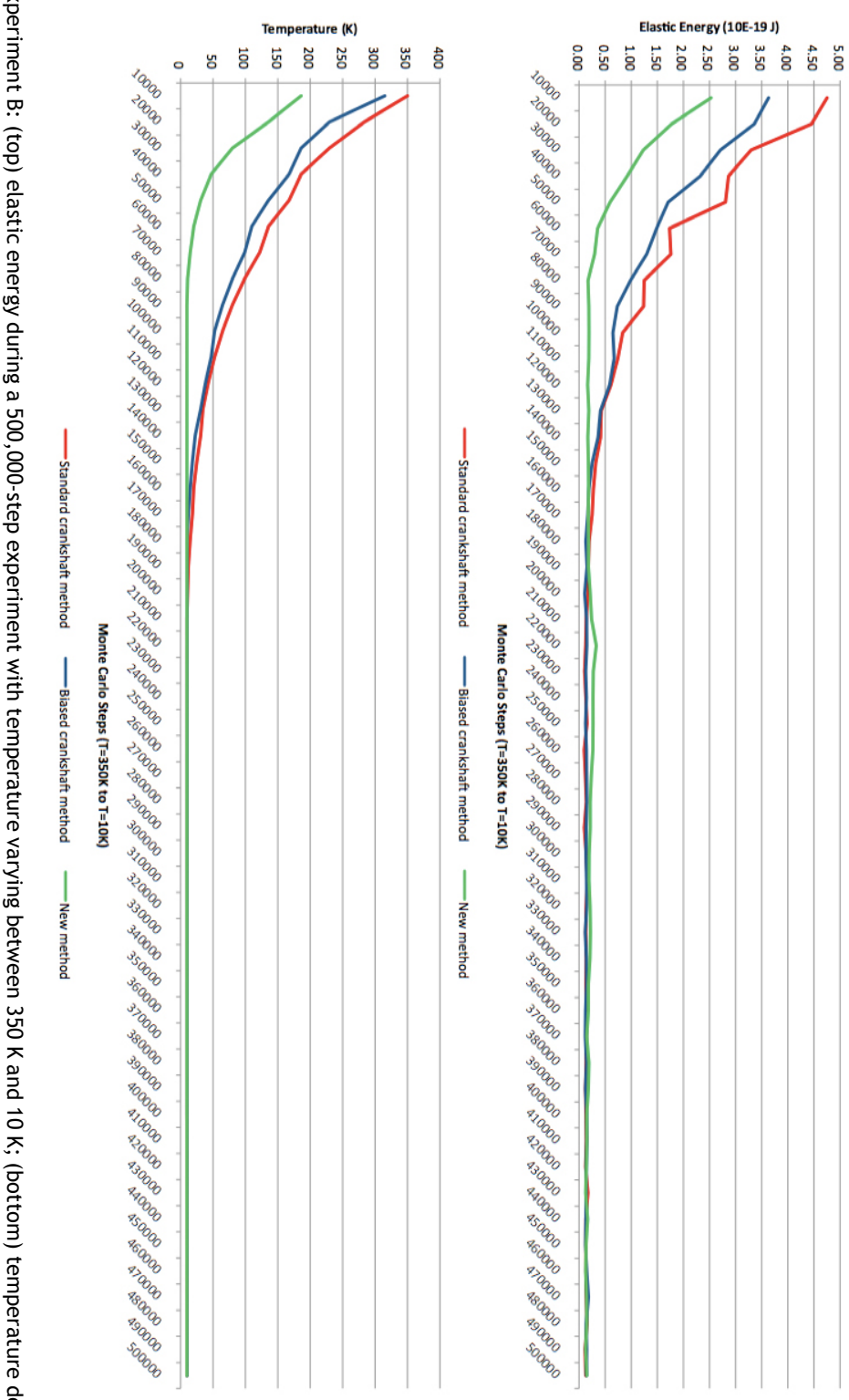


Figure 5.14: Experiment B: (top) elastic energy during a 500,000-step experiment with temperature varying between 350 K and 10 K; (bottom) temperature decaying during a 500,000-step experiment.

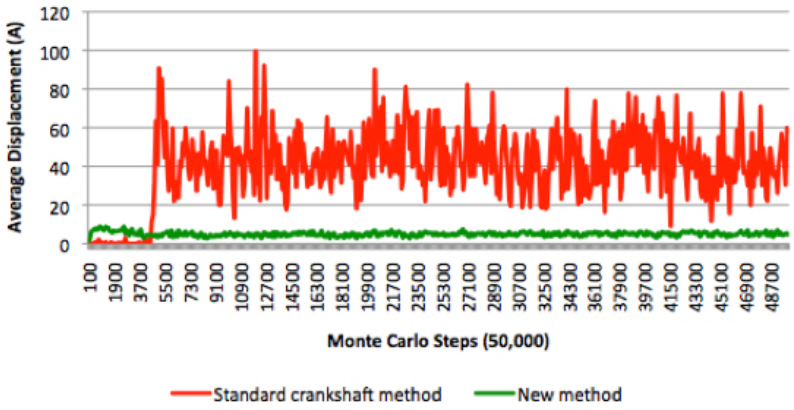


Figure 5.15: Experiment C: Average displacement per 100 steps for pUC19 50,000 steps.

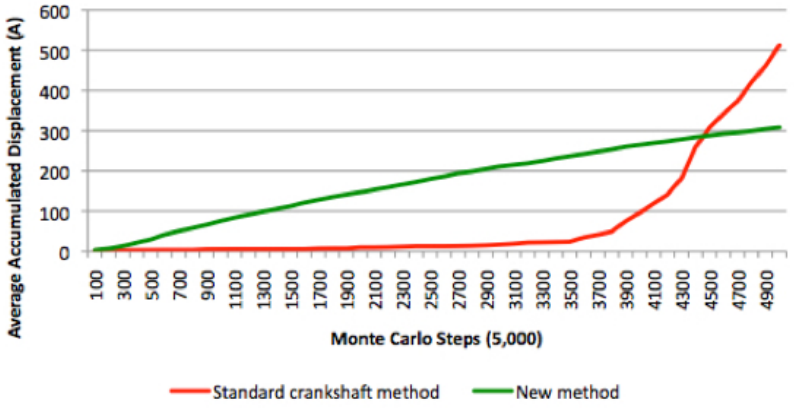


Figure 5.16: Experiment C: Average accumulated displacement for pUC19 5,000 steps.

crankshaft move, (b) the *biased* crankshaft move (i.e., with rotation angle optimization and adjustment), and (c) the proposed method.

As expected, the acceptance ratio of trials for the proposed method is higher than that for either of the crankshaft moves. The acceptance ratio of the new method is almost always greater than 40%, and even reaches more than 60% at certain times. Moreover, unlike the crankshaft moves, the acceptance ratio for the proposed method is very high from the very first steps of the simulation. More specifically, in a scenario with decreasing temperature, we obtained an acceptance ratio of more than 60% for the proposed method compared with 5% for the standard crankshaft move and 20% for the biased crankshaft move in the first 10,000 MC steps.

With respect to elastic energy, the experiments also show that as the temperature decreases the new move achieves better performance than either of the crankshaft moves. In fact, we noted that elastic energy tends to its equilibrium point not only in a smoother and more natural way, but also more quickly with fewer MC steps.

On the other hand, with regard to the average displacement of vertices in each trial, which provides the deformation measure of each tentative conformation, we noted that, as expected, the proposed move produces smaller deformations than either of the crankshaft moves. However, the accumulated displacement of the proposed move is actually greater than those of both crankshaft moves in the first 4,000 or so simulation steps (cf. Fig. 5.16). This high acceptance ratio in the initial simulation steps means that the proposed move generates a much more consistent deformation, the behavior of which obeys a logarithmic curve instead of the exponential curve that describes the accumulated deformation of each of the two crankshaft moves considered in this chapter.

5.5 Concluding Remarks

The crankshaft rotation method is the most common move found in the plasmid DNA simulation methods for generating new DNA conformations. Recall that this classic method first selects two random vertices of the DNA skeleton, and then all the segments between these two vertices are rotated around the axis defined by them. This move is not very effective because many trials are rejected by the MC method. In addition to its low acceptance ratio, this method can generate unnatural movements with large portions of the DNA molecule displaced at once, unless the relevant parameters are appropriately adjusted.

In this chapter, we introduced a new move for plasmid DNA through MC simulations. In each iteration of the simulation, only one vertex and its two closest flanking vertices are subjected to the motion procedure. Thus, for each new trial, a single vertex is randomly chosen and then randomly displaced to a point within a small neighborhood. To maintain connectivity of the DNA chain, as well as the size of its segments, the two flanking vertices are also displaced but in a less free way. Thus, only three vertices are displaced in each new trial.

Interestingly, considering that our algorithm generates small deformations in the transition from one DNA conformation to another, we can conclude that it can be applied not only in the simulation of DNA coiling, but also in real-time visualization. We have already done this by employing the DNA assembly algorithm that uses Gaussian surfaces as geometric representations of nucleotides.

In the future, we intend to incorporate a smoothing mechanism into our DNA algorithm like, for example, that presented in [KP05]. This will enable our algorithm to produce even more realistic simulations, eliminating the occurrence of slightly sharp corners like those shown in Fig. 5.7. We also intend generating deformations that depend on the DNA's stiffness, which varies according to the sequence of nucleotides. This will mean greater deformations on more flexible segments and smaller deformations on less flexible segments of DNA.

Finally, our ultimate goal is to be able to replicate *in silico*, and visualize what happens to plasmid DNA during the production and purification processes in laboratory experiments.

Related Publications

The work presented in this chapter gave origin to a paper that was published in due course [RG14], as indicated below:

> Adriano N. Raposo and Abel J. P. Gomes: Efficient deformation algorithm for plasmid DNA simulations. *BMC Bioinformatics*, **15**(301), 2014. [RG14].

Chapter 6

Conclusions

This thesis presents a few contributions to the advance of knowledge in the graphics-based molecular modeling field, in the sense that not only presents solutions for major problems that were unsolved, like plasmid DNA stacking over arbitrary conformations, but also presents new improvements to existing methods as, for example, the spatial partitioning of the atoms of a given molecule into influence boxes and influence spheres that speeds up and makes more efficient the triangulation of Gaussian molecular surfaces. This thesis also presents an alternative solution for an already solved problem, namely, the deformation of plasmid DNA for Monte Carlo simulations, but with better results than the traditional approach, i.e., with a much higher trial acceptance ratio and faster convergence to plasmid DNA elastic energy equilibrium state.

6.1 Framing of the Problem

Plasmid DNA is a special type of DNA molecules that is used in DNA vaccination and gene therapy. These DNA molecules are characterized by presenting a closed-circular topology and by occurring in nature in a supercoiled conformation, which is the desired conformation for its therapeutic applications.

Plasmid DNA is produced in laboratory by incorporating it into the cell of host microorganisms, such as *E.coli*, that will replicate the plasmid DNA at the same time they reproduce themselves and replicate their own DNA. The next step in the production of plasmid DNA, known as purification, consists of extracting the wanted plasmid DNA, and separating it from the DNA of the host and other possible contaminants. During the purification procedure, plasmid DNA is submitted to physical and chemical changes such as temperature variations. These changes may modify the conformation of naturally supercoiled plasmid DNA molecules, uncoiling them into a relaxed conformation, or even making them to assume open conformations, which is not desirable at all.

Researchers are constantly experimenting with new lab techniques in order to maximize the amount of supercoiled plasmid DNA after purification, i.e., they are always trying to find optimal conditions to prevent, as much as possible, plasmid DNA relaxation during purification processes. These lab experiments may take many hours or even several days, and every time an experiment fails, researchers lose time and money. The possibility of replicating *in silico* what happens to the conformation of these molecules during lab experiments, specially during purification procedures, has many advantages in terms of time and money savings.

It happens that the DNA reproduction *in silico* requires the satisfaction of a number of requirements, namely: multi-abstraction geometric models (from atom to macromolecule), DNA stacking at different levels of abstraction for real-time simulation and visualization, and microreversibility-based DNA deformation methods as a support for DNA coiling/uncoiling simulation. In a way, this is what we have achieved with this thesis work.

6.2 Contributions

This section synthesizes the most significant empirical findings to answer the research questions put forward in Chapter 1:

- Using an effective divide-and-conquer approach, and a general-purpose implicit surface triangulation algorithm applied to Gaussian molecular surfaces, it was made possible to triangulate the molecular surface of DNA molecules of any size, including plasmid DNA, within acceptable computation times. Dividing the atoms into influence boxes and influence spheres makes it possible to accurately triangulate the molecular surface of a certain region of the molecule despising the contribution of distant atoms. This is equivalent to use blending functions with a bounded kernel. Consequently, we end up having a significant speedup of the algorithm in relation to a solution that uses unbounded kernel blending functions (e.g., Gaussians) for electron density fields of atoms.

- Assuming that DNA can be divided into subsidiary molecules, called nucleotides, it is possible to use instances of the molecular surfaces of these molecules as building blocks to assemble DNA molecules over any arbitrary conformation, starting from its base-pairs sequence. Modeling the DNA's axis as a three-dimensional arbitrary skeleton, and assigning each pair of nucleotides to a segment of this skeleton, it becomes possible to place each pair of nucleotides building blocks around its corresponding DNA skeleton segment using empirically found parameters as distances, rotation angles, and alignment angles. Unlike conventional DNA assembling algorithms, which use statistical values for distances and angles between base pairs to predict the expected conformation of the overall DNA molecule, the new DNA stacking method applies to arbitrary conformations as those obtained from, for example, plasmid DNA simulation methods.

- By adopting the Monte Carlo simulation method for plasmid DNA molecules, it was made possible to replace the crankshaft move and its variants by a more efficient deformation algorithm. Using a polyline model of the plasmid DNA conformation, we can achieve overall large deformations, as needed for Monte Carlo simulations purposes, using more efficient smaller deformations in each iteration step. The efficiency of the deformation can be measured by the acceptance ratio of trials. In this respect, the new deformation method proposed in this thesis is clearly more efficient than the crankshaft move and its variants.

- It was made possible to visualize in real-time the evolution of plasmid DNA simulations in 3D using a commodity PC (personal computer), since we use a DNA stacking method as the one described in this thesis. In fact, the use of nucleotides' molecular surfaces as building blocks, instead of atoms, reduces the amount of data to be processed and visualized on screen. Besides, the use of a deformation algorithm with higher acceptance ratio of the conformation trials, and with more progressive deformations, generates a more smooth animation of the simulation without sudden large displacements of big portions of the molecule.

6.3 Research Limitations

This section presents some of the limitations that were found during this research. The goal of this section is to clarify whether the methods or techniques presented in this thesis are the best option, or not, to solve some specific problems that each researcher might have. This thesis has been focused essentially on geometric modeling, simulation, and visualization of plasmid DNA. Because of that, there could be other problems where the algorithms presented in this thesis might not be the optimal solution. Besides, pointing some limitations of this work might be useful for a broader discussion that could lead to improvements on the solutions presented. In this section, the research limitations are separated and organized by the core algorithms.

Some of the research limitations of the Gaussian molecular surface triangulation algorithm presented in Chapter 3 might be the following:

- The divide-and-conquer approach of the triangulation algorithm is not optimized for highly concentrated atomic structures. The fixed size of all influence boxes and influence spheres slows the algorithm for molecules with high atomic density, i.e., with a large amount of atoms concentrated in a relatively small region, as it is the case of some proteins. With the presented algorithm, it takes longer to triangulate the molecular surface inside those influence boxes containing more atoms.
- Even if the triangulation algorithm divides the atomic structure of a molecule into influence boxes and influence spheres, it worthy to say that it is not a space partitioning algorithm, being instead a continuation algorithm. Because of this, it assumes that the influence box contains only one piece of the molecular surface. Thus, the triangulation algorithm might miss molecule's inner voids or cavities when the surface of those voids or cavities is not connected to the remainder of the molecular surface.

In turn, some of the research limitations of the DNA assembling algorithm presented in Chapter 4 might be the following:

- Being an adaptive DNA assembling algorithm, in the sense that it is able to assemble a DNA base-pair sequence over any arbitrary conformation, means that it may be using some unconventional angles and distances between specific dinucleotides. This feature, even if it is essential for plasmid DNA simulation purposes where the assembling is oriented to a specific conformation, might not be appropriate for other DNA applications. However, the idea of using the molecular surfaces of the nucleotides as building blocks can be used together with conventional predictive DNA assembling algorithms.
- The DNA assembling algorithm presented in this thesis is only suitable for the B-DNA form, which is the most common form of plasmid DNA. This means that the algorithm, as it is, should not be used to assemble other forms of DNA molecules like, for example, A-DNA or Z-DNA. Similar algorithms can be designed for these other forms of DNA using the same building blocks by slightly adjusting the rotation angles, distances and other empiric parameters used by the assembling procedure.

Finally, some of the research limitations of the plasmid DNA deformation algorithm presented in Chapter 5 might be the following:

- After a few hundreds of simulation steps, the deformation algorithm might originate plasmid DNA conformations with sharp kinks, which are not natural, and most probably will not occur in laboratory experiments. The algorithm itself does not include a smoothing mechanism because the main goal was to compare it with conventional deformation techniques without changing the results, or affecting the acceptance of the trials and the molecule's elastic energy, with a smoothing procedure.

- In terms of experimental physico-chemical conditions, the Monte Carlo simulation method used in the experiments together with the new deformation algorithm only takes into account temperature changes. The simulations that were performed did not take into consideration changes in, for example, salt concentration as used in lab experiments of plasmid DNA purification.

- Unlike in plasmid DNA purification laboratory experiments, where researchers manipulate many molecules at once, all the simulation experiments that were performed in the context of this thesis used a single plasmid DNA molecule.

The author believes that some of the limitations referred in this section can be surpassed in the future, as addressed in the following section.

6.4 Future Work

In the course of this thesis, several questions have been arising, though not implemented due to time limitations or because they were considered to be slightly out of the core of the thesis. Nevertheless, some of these ideas are worthy to be included in this section, at least as topics for future research. It is author's intent to approach some of them as part of, for example, a post-doctoral research project as a followup of the present thesis.

Some of such major outlines for future work are the following:

- *To reformulate the Gaussian molecular surface triangulation algorithm with influence boxes and influence spheres of different sizes*. The idea is to create a load balancing mechanism according to the atomic density of the molecules. The solution may be creating smaller influence boxes in regions with high concentration of atoms.

- *To design a parallel implementation of the Gaussian molecular surface triangulation algorithm*. The idea is to use the division of the molecules into influence boxes and influence spheres, and assign a set of these boxes and spheres, for example, to each one of the cores of a GPU in order to speed up the triangulation algorithm even further.

- *To design a parallel implementation of the DNA assembling algorithm*. Here the idea is to divide the DNA skeleton into pieces, assigning then each one of these pieces to a core of the GPU. Each core of the GPU will be responsible by assembling of only the portion of the base-pair sequence that corresponds to its assigned piece of the DNA skeleton. This is equivalent to replace a DNA stacking by a general DNA assembly procedure in the sense that we no longer need to stack a base pair to stack the one that comes immediately afterwards.

- *To design an adaptive/predictive hybrid DNA assembly algorithm*. The idea is to use the Cambridge Meeting guidelines together with the DNA assembly algorithm presented in this thesis to create an algorithm that takes advantage of traditionally acceptable geometric parameters of DNA molecules, but that is also capable of some degree of conformational adaptability.
- *To design a kink-free smoothing technique for the deformation algorithm*. The idea is to apply a smoothing technique to the plasmid DNA conformations that had been obtained from simulations, in order to prevent the occurrence of sharp kinks.
- *To compare* in silico *results with those produced by real plasmid DNA lab experiments*. The idea is to validate the plasmid DNA geometric models and simulation results here presented using real lab experiments. Both *in silico* and laboratory experiments must be performed using exactly the same molecules and identical physical and chemical variables. The goal is to check whether the coiling/uncoiling *in silico* mimics the coiling/uncoiling *in vitro* or not.

6.5 Closure of the Research Work

At this point, it is worth recalling the thesis statement put forward in Chapter 1, which is as follows:

> *It is geometrically possible to replicate and render the DNA uncoiling/coiling processes in real-time using a commodity PC without parallel computing and graphics acceleration.*

This thesis statement is sustained on two principles: *shape composition* and *micro-reversibility*. The first is underlying the DNA assembly method that allows us to model, simulate, and visualize plasmid DNA in real-time without using parallel computing of any sort (cf. Chapter 4). The second is behind the DNA deformation method that is at the core of the DNA uncoiling/coiling processes (cf. Chapter 5). Geometrically speaking, the grand objective of this thesis has been achieved since a complete geometric model for plasmid DNA has been introduced, which satisfies those two principles, as needed for DNA simulation and visualization *in silico*.

Bibliography

[AB27] J. W. Alexander and G. B. Briggs. On types of knotted curves. *Annals of Mathematics*, 28:562-586, 1927. 18

[AE96] N. Akkiraju and H. Edelsbrunner. Triangulating the surface of a molecule. *Discrete Applied Mathematics*, 71(1-3):5-22, 1996. 46

[AG87] E. Allgower and S. Gnutzmann. An algorithm for piecewise linear approximation of implicitly defined two-dimensional surfaces. *SIAM Journal on Numerical Analysis*, 24(2):452-469, 1987. 47

[AJ05] B. R. Araújo and J. A. P. Jorge. Adaptive polygonization of implicit surfaces. *Computers and Graphics*, 29(5):686 - 696, 2005. 47

[Ale28] J. W. Alexander. Topological invariants of knots and links. *Transactions of the American Mathematical Society*, 30:275-306, 1928. 17

[AW57] B. J. Alder and T. E. Wainwright. Phase transition for a hard sphere system. *The Journal of Chemical Physics*, 27(5):1208-1209, 1957. 79

[Bau78] W. R. Bauer. Structure and reactions of closed duplex dna. *Annual Review of Biophysics and Bioengineering*, 7(1):287-313, 1978. PMID: 208457. 88

[BB88] D. Bhattacharya and M. Bansal. A general procedure for generation of curved DNA molecules. *Journal of Biomolecular Structure and Dynamics*, 6(1):93-104, Aug 1988. 19, 41

[BBB+04] D. L. Beveridge, G. Barreiro, K. S. Byun, D. A. Case, T. E. Cheatham, S. B. Dixit, E. Giudice, F. Lankas, R. Lavery, J. H. Maddocks, R. Osman, E. Seibert, H. Sklenar, G. Stoll, K. M. Thayer, P. Varnai, and M. A. Young. Molecular dynamics simulations of the 136 unique tetranucleotide sequences of DNA oligonucleotides. I. Research design and results on d(CpG) steps. *Biophysical Journal*, 87(6):3799-3813, Dec 2004. 80

[BC81] P. O. Brown and N. R. Cozzarelli. Catenation and knotting of duplex DNA by type 1 topoisomerases: a mechanistic parallel with type 2 topoisomerases. *Proceedings of the National Academy of Sciences*, 78(2):843-847, Feb 1981. 16

[BDS08] Y. Burnier, J. Dorier, and A. Stasiak. DNA supercoiling inhibits DNA knotting. *Nucleic Acids Research*, 36(15):4956-4963, Sep 2008. 79, 91

[Bin97] K. Binder. Applications of Monte Carlo methods to statistical physics. *Reports on Progress in Physics*, 60(5), 1997. 78

[BKML+11] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers. GenBank. *Nucleic Acids Research*, 39 (Suppl):D32-D37, January 2011. 68

[BKW+77] F. C. Bernstein, T. F. Koetzle, G. J. Williams, E. F. Meyer Jr, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The protein data bank: a computer-based archival file for macromolecular structures. *Journal of Molecular Biology*, 112(3):535-42, 1977. 26

[Bli82] J. F. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235-256, July 1982. 4, 46, 48, 65, 66

[BLMP97] C. Bajaj, H. Y. Lee, R. Merkert, and V. Pascucci. NURBS based B-rep models for macromolecules and their properties. In *SMA '97: Proceedings of the Fourth ACM Symposium on Solid Modeling and Applications*, pages 217-228, New York, NY, USA, 1997. ACM. 11, 45

[Blo88] J. Bloomenthal. Polygonization of Implicit Surfaces. *Computer Aided Geometric Design*, 4(5):341-355, 1988. 46, 47

[BM05] A. Bates and A. Maxwell. *DNA Topology*. Oxford University Press, 2nd edition, 2005. 14, 15, 16, 61, 69, 70, 82

[BMHT91] A. Bolshoy, P. McNamara, R. E. Harrington, and E. N. Trifonov. Curved dna without a-a: experimental estimation of all 16 dna wedge angles. *Proceedings of the National Academy of Sciences*, 88(6):2312-2316, 1991. xxix, 12, 31, 32, 35, 37, 41

[BMS06] A. Balaeff, L. Mahadevan, and K. Schulten. Modeling DNA loops using the theory of elasticity. *Physical Review E*, 73(3 Pt 1):1-24, March 2006. 63

[BNvO96] A. Bottino, W. Nuij, and K. van Overveld. How to shrinkwrap through a critical point: an algorithm for the adaptive triangulation of iso-surfaces with arbitrary topology. In *Proceedings of Implicit Surfaces 96 Conference*, pages 256-267, Eindhoven, September, 1996. 47

[BOB+92] H. M. Berman, W. K. Olson, D. L. Beveridge, J. Westbrook, A. Gelbin, T. Demeny, S. H. Hsieh, A. R. Srinivasan, and B. Schneider. The nucleic acid database. A comprehensive relational database of three-dimensional structures of nucleic acids. *Biophysical Journal*, 63(3):751-759, Sep 1992. 20, 22

[BoCGB06] G.M. Blackburn and Royal Society of Chemistry (Great Britain). *Nucleic Acids in Chemistry and Biology*. RSC Pub., 2006. 14

[Bol64] L. Boltzmann. *Lectures on gas theory*. University of California Press, Oakland, California, USA, 1964. 86

[Bon64] A. Bondi. Van der Waals Volumes and Radii. *The Journal of Physical Chemistry*, 68(3):441-451, 1964. 64

[BPO93] M. S. Babcock, E. P. Pednault, and W. K. Olson. Nucleic acid structure analysis: a users guide to a collection of new analysis programs. *Journal of Biomolecular Structure and Dynamics*, 11(3):597-628, Dec 1993. 28

[BPS+03] C. L. Bajaj, V. Pascucci, A. Shamir, R. J. Holt, and A. N. Netravali. Dynamic maintenance and visualization of molecular surfaces. *Discrete Applied Mathematics*, 127(1):23-51, 2003. 45

[BPZL11] C. Blanchet, M. Pasi, K. Zakrzewska, and R. Lavery. CURVES+ web server for analyzing and visualizing the helical, backbone and groove parameters of nucleic acid structures. *Nucleic Acids Research*, 39(Web Server issue):68-73, Jul 2011. 31

[Bro98] M. Brodzik. The computation of simplicial approximations of implicitly defined p-manifolds. *Computers and Mathematics with Applications*, 36(6):389-423, 1998. 47

[BS02] A. Bucka and A. Stasiak. Construction and electrophoretic migration of single-stranded DNA knots and catenanes. *Nucleic Acids Research*, 30(6):e24, Mar 2002. 16

[Car07] M. Carson. *Electron density fitting and structure validation*, pages 191-201. Oxford Scholarship Online Monographs, 2007. 62

[CCD+05] D. A. Case, T. E. Cheatham, T. Darden, H. Gohlke, R. Luo, K. M. Merz, A. Onufriev, C. Simmerling, B. Wang, and R. J. Woods. The Amber biomolecular simulation programs. *Journal of Computational Chemistry*, 26(16):1668-1688, Dec 2005. 34

[CCW06] T. Can, C. Chen, and Y. Wang. Efficient molecular surface generation using level-set methods. *Journal of Molecular Graphics and Modelling*, 25(4):442-454, 2006. 46

[CDES01] H. Cheng, T. K. Dey, H. Edelsbrunner, and J. Sullivan. Dynamic skin triangulation. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '01)*, pages 47-56. SIAM Press, 2001. 46

[CJS+96] L. Clowney, S. C. Jain, A. R. Srinivasan, J. Westbrook, W. K. Olson, and H. M. Berman. Geometric parameters in nucleic acids: Nitrogenous bases. *Journal of the American Chemical Society*, 118(3):509-518, 1996. 20

[CL92] G. Chirico and J. Langowski. Calculating hydrodynamic properties of dna through a second-order brownian dynamics algorithm. *Macromolecules*, 25(2):769-775, 1992. 81

[CL94] G. Chirico and J. Langowski. Kinetics of dna supercoiling studied by brownian dynamics simulation. *Biopolymers*, 34(3):415-433, 1994. 81

[CL96] G. Chirico and J. Langowski. Brownian dynamics simulations of supercoiled DNA with bent sequences. *Biophysical Journal*, 71(2):955-971, Aug 1996. 81

[Con83] M. L. Connolly. Analytical molecular surface calculation. *Journal of Applied Crystallography*, 16(5):548-558, 1983. 45

[Con85] M. L. Connolly. Molecular surface triangulation. *Journal of Applied Crystallography*, 18:499-503, 1985. 46

[Con96] M. L. Connolly. Molecular surfaces: A review, 1996. 11

[CV67] D. A. Clayton and J. Vinograd. Circular dimer and catenate forms of mitochondrial DNA in human leukaemic leucocytes. *Nature*, 216(5116):652-657, Nov 1967. 17

[CVBC05] G. Charvin, A. Vologodskii, D. Bensimon, and V. Croquette. Braiding DNA: experiments, simulations, and models. *Biophysical Journal*, 88(6):4124-4136, 2005. 63

[CY94] M. Carson and Z. Yang. DNurbs: DNA modeled with NURBS. *Journal of Molecular Graphics*, 12:178-184, 1994. 62

[DBC+05] S. B. Dixit, D. L. Beveridge, D. A. Case, T. E. Cheatham, E. Giudice, F. Lankas, R. Lavery, J. H. Maddocks, R. Osman, H. Sklenar, K. M. Thayer, and P. Varnai. Molecular dynamics simulations of the 136 unique tetranucleotide sequences of DNA oligonucleotides. II: sequence context effects on the dynamical structures of the 10 unique dinucleotide steps. *Biophysical Journal*, 89(6):3721-3740, Dec 2005. 80

[Dic89] R. E. Dickerson. Definitions and nomenclature of nucleic acid structure components. *Nucleic Acids Research*, 17(5):1797-1803, Mar 1989. 12, 19, 20, 22, 27, 30, 31, 35, 37, 41

[Dic98] R. E. Dickerson. DNA bending: the prevalence of kinkiness and the virtues of normality. *Nucleic Acids Research*, 26(8):1906-1926, Apr 1998. 27, 28

[DLTW90] D. Dobkin, S. Levy, W. Thurston, and A. Wilks. Contour tracing by piecewise linear approximations. *ACM Transactions on Graphics*, 9(4):389-423, 1990. 47

[DO71] J. M. Deutch and I. Oppenheim. Molecular theory of brownian motion for several particles. *The Journal of Chemical Physics*, 54(8):3547-3555, 1971. 81

[DO93] B. S. Duncan and A. J. Olson. Shape analysis of molecular surfaces. *Biopolymers*, 33(2):231-238, 1993. 46

[DV63] R. Dulbecco and M. Vogt. Evidence for a ring structure of polyoma virus DNA. *Proceedings of the National Academy of Sciences*, 50:236-243, Aug 1963. 15

[ED08] D. J. Earl and M. W. Deem. Monte Carlo Simulations. In Andreas Kukol, editor, *Molecular Modeling of Proteins*, volume 443 of *Methods in Molecular Biology*, chapter 2, pages 25-36. Humana Press, 2008. 78, 89

[Ede99] H. Edelsbrunner. Deformable smooth surface design. *Discrete & Computational Geometry*, 21(1):87-115, 1999. 45

[EH05] E. Eyal and D. Halperin. Dynamic maintenance of molecular surfaces under conformational changes. In *Proceedings of the 21st Annual Symposium on Computational Geometry (SCG '05)*, pages 45-54. ACM Press, 2005. 63

[EHC95] M. A. El Hassan and C. R. Calladine. The assessment of the geometry of dinucleotide steps in double-helical DNA; a new local calculation scheme. *Journal of Molecular Biology*, 251(5):648-664, Sep 1995. 20, 22, 27

[EHC97] M. A. El Hassan and C. R. Calladine. Conformational characteristics of dna: empirical classifications and a hypothesis for the conformational behaviour of dinucleotide steps. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 355(1722):43-100, 1997. xxix, 22, 24, 25, 26, 27

[EM78] D. L. Ermak and J. A. McCammon. Brownian dynamics with hydrodynamic interactions. *The Journal of Chemical Physics*, 69(4):1352-1360, 1978. 81

[Far12] G. Farin. Shape measures for triangles. *IEEE Transactions on Visualization and Computer Graphics*, 18:43-46, 2012. 67

[FBS06] B. S. Fujimoto, G. P. Brewood, and J. M. Schurr. Torsional rigidities of weakly strained DNAs. *Biophysical Journal*, 91(11):4166-4179, Dec 2006. 79

[FK97] M. D. Frank-Kamenetskii. Biophysics of the DNA molecule. *Physics Reports*, 288(1-6):13-60, 1997. 15

[FKLV75] M. D. Frank-Kamenetskii, A. V. Lukashin, and A. V. Vologodskii. Statistical mechanics and topology of polymer chains. *Nature*, 258(5534):398-402, Dec 1975. 79

[FKT+07] S. Fujii, H. Kono, S. Takenaka, N. Go, and A. Sarai. Sequence-dependent DNA deformability studied using molecular dynamics simulations. *Nucleic Acids Research*, 35(18):6063-6074, 2007. 80

[FKV81] M. D. Frank-Kamenetskii and A. V. Vologodskii. Topological aspects of the physics of polymers: The theory and its biophysical applications. *Soviet Physics Uspekhi*, 24(8):679, 1981. 79

[FS02] B. S. Fujimoto and J. M. Schurr. Monte Carlo simulations of supercoiled DNAs confined to a plane. *Biophysical Journal*, 82(2):944-962, Feb 2002. 79

[GACS95] J. A. Gebe, S. A. Allison, J. B. Clendenning, and J. M. Schurr. Monte Carlo simulations of supercoiling free energies for unknotted and trefoil knotted DNAs. *Biophysical Journal*, 68(2):619-633, Feb 1995. 79

[GH00] R. Gherbi and J. Herisson. ADN viewer a software framework toward 3D modeling and stereoscopic visualization of genome. *International Conference Graphicon*, 2000. 32, 34, 41, 62

[GH02] R. Gherbi and J. Herisson. Representation and processing of complex DNA spatial architecture and its annotated genomic content. *Pacific Symposium on Biocomputing*, pages 151-162, 2002. 32, 34, 35, 62

[GHP+03] E. J. Gardiner, C. A. Hunter, M. J. Packer, D. S. Palmer, and P. Willett. Sequence-dependent {DNA} structure: A database of octamer structural parameters. *Journal of Molecular Biology*, 332(5):1025-1035, 2003. xxix, 22, 24, 25

[GM01] O. Gonzalez and J. H. Maddocks. Extracting parameters for base-pair level models of DNA from molecular dynamics simulations. *Theoretical Chemistry Accounts*, 106(1-2):76-82, 2001. 80

[Goo05] S. E. Goodman. *Beginning Topolgy*. Thomson Brooks/Cole, 2005. 16

[GP95] J. Grant and B. Pickup. A Gaussian description of molecular shape. *The Journal of Physical Chemistry*, 99(11):3503-3510, 1995. 46, 73

[GST+05] R. Goodman, I. Schaap, C. Tardin, C. Erben, R. Berry, C. Schmidt, and A. Turberfield. Rapid chiral assembly of rigid DNA building blocks for molecular nanofabrication. *Science*, 310(9):1661-1665, 2005. 62

[GVJ+09] A.J.P. Gomes, I. Voiculescu, J. Jorge, B. Wyvill, and C. Galbraith. *Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms*. Springer-Verlag, London, 2009. 46, 67

[GZW95] A. A. Gorin, V. B. Zhurkin, and K. Wilma. B-DNA twisting correlates with base-pair morphology. *Journal of Molecular Biology*, 247(1):34-48, 1995. xxix, 22, 24, 25

[HD75] H. Hilhorst and J. Deutch. Analysis of Monte Carlo results on the kinetics of lattice polymer chains with excluded volume. *The Journal of Chemical Physics*, 63(12):5153-5161, 1975. 78

[HDR+10] A. Hildebrandt, A. K. Dehof, A. Rurainski, A. Bertsch, M. Schumann, N. C. Toussaint, A. Moll, D. Stockel, S. Nickels, S. C. Mueller, H. P. Lenhof, and O. Kohlbacher. BALL--biochemical algorithms library 1.3. *BMC Bioinformatics*, 11:531, 2010. 29

[Her06] A. Herraez. Biomolecules in the computer: Jmol to the rescue. *Biochem Mol Biol Educ*, 34(4):255--261, Jul 2006. 29

[Hey88] J. Heywood. *Internal Combustion Engine Fundamentals*. McGraw-Hill, Inc., New York, USA, 1988. 86

[HFGG06] J. Herisson, N. Ferey, P. E. Gros, and R. Gherbi. ADN-Viewer: a 3D approach for bioinformatic analyses of large DNA sequences. *Cellular and Molecular Biology*, 52(6):24-31, 2006. xxix, 32, 34, 62

[HGF+04] J. Hérisson, P. E. Gros, N. Férey, O. Magneau, and R. Gherbi. Dna in virtuo visualization and exploration of 3d genomic structures. In *Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, AFRIGRAPH '04, pages 35--40, New York, NY, USA, 2004. ACM. 34, 35

[HH99] B. A. Harris and S. C. Harvey. Program for analyzing knots represented by polygonal paths. *Journal of Computational Chemistry*, 20(8):813-818, 1999. 9, 18, 82, 89

[HL97] C. A. Hunter and X. J. Lu. Construction of double-helical DNA structures based on dinucleotide building blocks. *Journal of Biomolecular Structure and Dynamics*, 14(6):747-756, Jun 1997. 18

[HLL08] S. A. Harris, C. A. Laughton, and T. B. Liverpool. Mapping the phase diagram of the writhe of DNA nanocircles using atomistic molecular dynamics simulations. *Nucleic Acids Research*, 36(1):21-29, Jan 2008. 80

[HLLF13] S. Hornus, B. Levy, D. Lariviere, and E. Fourmentin. Easy DNA modeling and more with GraphiteLifeExplorer. *PLoS ONE*, 8(1):e53609, 2013. 12, 37, 40, 41

[Hoo63] K. Hoogsteen. The crystal and molecular structure of a hydrogen-bonded complex between 1-methylthymine and 9-methyladenine. *Acta Crystallographica*, 16(20):907-916, September 1963. 13

[HPG07] J. Herisson, G. Payen, and R. Gherbi. A 3D pattern matching algorithm for DNA sequences. *Bioinformatics*, 23(6):680-686, Mar 2007. 35

[HR96] A. Herbert and A. Rich. The biology of left-handed Z-DNA. *The Journal of Biological Chemistry*, 271(20):11595-11598, May 1996. 14, 15

[HV67] B. Hudson and J. Vinograd. Catenated circular DNA molecules in HeLa cell mitochondria. *Nature*, 216(5116):647-652, Nov 1967. 17

[HW90] M. Hall and J. Warren. Adaptive polygonization of implicitly defined surfaces. *IEEE Computer Graphics and Applications*, 10(6):33-42, 1990. 46

[JKSS96] A.D. Jenkins, P. Kratochvíl, R.F.T. Stepto, and U.W. Suter. Glossary of basic terms in polymer science. *Pure and Applied Chemistry*, 68(8):1591-1595, 1996. 13

[KBE09] M. Krone, K. Bidmon, and T. Ertl. Interactive visualization of molecular surface dynamics. *IEEE Transactions on Visualization & Computer Graphics*, 15(6):1391-1398, 2009. 63

[KBM+96] V. Katritch, J. Bednar, D. Michoud, R.G. Scharein, J. Dubochet, and A. Stasiak. Geometry and physics of knots. *Nature*, 384:142-145, 1996. 17

[KC80] K. N. Kreuzer and N. R. Cozzarelli. Formation and resolution of DNA catenanes by DNA gyrase. *Cell*, 20(1):245-254, May 1980. 17

[KL00] K. Klenin and J. Langowski. Computation of writhe in modeling of supercoiled DNA. *Biopolymers*, 54:307-317, Oct 2000. 82, 88

[KL01] K. V. Klenin and J. Langowski. Intrachain reactions of supercoiled DNA simulated by Brownian dynamics. *Biophysical Journal*, 81(4):1924-1929, Oct 2001. 81

[KLT98] S. Kundu, A. Lahiri, and A. R. Thakur. Denaturation of supercoiled DNA: a Monte Carlo study. *Biophysical Chemistry*, 75(3):177-186, Dec 1998. 63, 79

[KM02] M. Karplus and J. A. McCammon. Molecular dynamics simulations of biomolecules. *Nature Structural Biology*, 9(9):646-652, Sep 2002. 63, 80

[KML98] K. Klenin, H. Merlitz, and J. Langowski. A Brownian dynamics program for the simulation of linear and circular DNA and other wormlike chain polyelectrolytes. *Biophysical Journal*, 74(2 Pt 1):780-788, Feb 1998. 81

[KP05] E. A. Kummerle and E. Pomplun. A computer-generated supercoiled model of the pUC19 plasmid. *European Biophysics Journal*, 34(1):13-18, Feb 2005. 63, 87, 89, 90, 99

[KST82] W. Kabsch, C. Sander, and E. N. Trifonov. The ten helical twist angles of B-DNA. *Nucleic Acids Research*, 10(3):1097-1104, Feb 1982. xxix, 12, 22, 24, 25, 31, 32, 35, 41

[KVA+91] K. Klenin, A. Vologodskii, V. Anshelevich, A. Dykhne, and M. Frank-Kamenetskii. Computer simulation of DNA supercoiling. *Journal of Molecular Biology*, 63(3):413-419, 1991. 78, 82, 89, 90

[KWB10] D.S. Kim, C.I. Won, and J. Bhak. A proposal for the revision of molecular boundary typology. *Journal of Biomolecular Structure & Dynamics*, 28(2):277-288, 2010. 45

[KZ09] S. Kannan and M. Zacharias. Simulation of DNA double-strand dissociation and formation during replica-exchange molecular dynamics simulations. *Physical Chemistry Chemical Physics*, 11:10589-10595, 2009. 80

[LaM02] A. LaMothe. *Tricks of the Windows Game Programming Gurus*. Sams, Indianapolis, IN, USA, 2nd edition, 2002. 88

[LB80] M. Le Bret. Monte Carlo computation of the supercoiling energy, the sedimentation constant, and the radius of gyration of unknotted and knotted circular DNA. *Biopolymers*, 19(3):619-637, Mar 1980. 79, 88

[LB02] P. Laug and H. Borouchaki. Molecular surface modeling and meshing. *Engineering with Computers*, 18(3):199-210, 2002. 46

[LC87] W. Lorensen and H. Cline. Marching Cubes: A High Resolution 3-D Surface Construction Algorithms. *Computer Graphics*, 21(4):163-169, 1987. 46

[LDCZ09] Z. Liu, R. W. Deibler, H.S. Chan, and L. Zechiedrich. The why and how of DNA unlinking. *Nucleic Acids Research*, 37(3):661-671, 2009. 17

[LDW76] L. F. Liu, R. E. Depew, and J. C. Wang. Knotted single-stranded DNA rings: A novel topological isomer of circular single-stranded DNA formed by treatment with escherichia coli ω protein. *Journal of Molecular Biology*, 106(2):439-452, 1976. 16

[Lev83] M. Levitt. Computer simulation of DNA double-helix dynamics. *Cold Spring Harbor Symposia on Quantitative Biology*, 47:251-262, 1983. 79, 80

[LHSC99] R. G. Larson, Hua Hu, D. E. Smith, and S. Chu. Brownian dynamics simulations of a DNA molecule in an extensional flow field. *Journal of Rheology*, 43(2):267-304, 1999. 81

[LKS+98] B. Laurie, V. Katritch, J. Sogo, T. Koller, J. Dubochet, and A. Stasiak. Geometry and physics of catenanes applied to the study of DNA replication. *Biophysical Journal*, 74(6):2815-2822, Jun 1998. 17

[LLM06] F. Lankas, R. Lavery, and J. H. Maddocks. Kinking occurs during molecular dynamics simulations of small DNA minicircles. *Structure*, 14(10):1527-1534, Oct 2006. 80

[LMM+09] R. Lavery, M. Moakher, J. H. Maddocks, D. Petkeviciute, and K. Zakrzewska. Conformational analysis of nucleic acids revisited: Curves+. *Nucleic Acids Research*, 37(17):5917-5929, Sep 2009. 27, 30

[LO99] X. J. Lu and W. K. Olson. Resolving the discrepancies among nucleic acid conformational analyses. *Journal of Molecular Biology*, 285(4):1563-1575, Jan 1999. 20

[LO03] X. J. Lu and W. K. Olson. 3DNA: a software package for the analysis, rebuilding and visualization of three-dimensional nucleic acid structures. *Nucleic Acids Research*, 31(17):5108-5121, Sep 2003. 27, 28, 29, 30, 63

[LO08] X. J. Lu and W. K. Olson. 3DNA: a versatile, integrated software system for the analysis, rebuilding and visualization of three-dimensional nucleic-acid structures. *Nature Protocols*, 3(7):1213-1227, 2008. 28, 30

[LPCW81] L. F. Liu, L. Perkocha, R. Calendar, and J. C. Wang. Knotted DNA from bacteriophage capsids. *Proceedings of the National Academy of Sciences*, 78(9):5498-5502, Sep 1981. 16

[LR71] B. Lee and F.M. Richards. The interpretation of protein structures: estimation of static accessibility. *Journal of Molecular Biology*, 55(3):379-400, 1971. 45

[LS88] R. Lavery and H. Sklenar. The definition of generalized helicoidal parameters and of axis curvature for irregular nucleic acids. *Journal of Biomolecular Structure and Dynamics*, 6(1):63-91, Aug 1988. 19, 27, 28, 30, 41

[LS89] R. Lavery and H. Sklenar. Defining the structure of irregular nucleic acids: conventions and principles. *Journal of Biomolecular Structure and Dynamics*, 6(4):655-667, Feb 1989. 30

[LSLC03] F. Lankas, J. Sponer, J. Langowski, and T. E. Cheatham. DNA basepair step deformability inferred from molecular dynamics simulations. *Biophysical Journal*, 85(5):2872-2883, Nov 2003. 80

[Mau13] R. Mauri. Microscopic reversibility. In *Non-Equilibrium Thermodynamics in Multiphase Flows*, Soft and Biological Matter, pages 13-24. Springer, Dordrecht, Netherlands, 2013. 86

[MC98] T. J. Macke and D. A. Case. *Modeling Unusual Nucleic Acid Structures*, chapter 25, pages 379-393. American Chemical Society, 1998. 32, 33, 37, 40, 41, 42

[Mey97] E. F. Meyer. The first years of the Protein Data Bank. *Protein Science*, 6:1591-1597, July 1997. xxv, 14, 68

[MHLK06] A. Moll, A. Hildebrandt, H.P. Lenhof, and O. Kohlbacher. BALLView: a tool for research and education in molecular modeling. *Bioinformatics*, 22:365-366, Feb 2006. 29

[MJS96] J. Mazur, R. L. Jernigan, and A. Sarai. Constructing optimal backbone segments for joining fixed DNA base pairs. *Biophysical Journal*, 71(3):1493-1506, Sep 1996. 26, 27

[MM94] E. A. Merritt and M. E. Murphy. Raster3D Version 2.0. A program for photorealistic molecular graphics. *Acta Crystallographica D Biological Crystallography*, 50(Pt 6):869-873, Nov 1994. 29

[MM95] R. Melville and D. Mackey. New algorithm for two-dimensional numerical continuation. *Computers and Mathematics with Applications*, 30(1):31-46, 1995. 47

[MN08] A. D. Mackerell and L. Nilsson. Molecular dynamics simulations of nucleic acid-protein complexes. *Current Opinion in Structural Biology*, 18(2):194-199, Apr 2008. 80

[MRKL98] H. Merlitz, K. Rippe, K. V. Klenin, and J. Langowski. Looping dynamics of linear DNA molecules and the effect of DNA curvature: a study by Brownian dynamics simulation. *Biophysical Journal*, 74(2 Pt 1):773-779, Feb 1998. 81

[MRR+53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087-1092, 1953. 26, 78, 79, 83

[MS93] H. Muller and M. Stark. Adaptive generation of surfaces in volume data. *The Visual Computer*, 4(9):182-199, 1993. 46

[MS05] J. F. Mercier and G. W. Slater. Solid phase DNA amplification: a Brownian dynamics study of crowding effects. *Biophysical Journal*, 89(1):32-42, Jul 2005. 81

[NSSM73] R. P. Novick, K. Smith, R. J. Sheehy, and E. Murphy. A catenated intermediate in plasmid replication. *Biochemical and Biophysical Research Communications*, 54(4):1460-1469, Oct 1973. 17

[NTT07] S. Nagatoishi, Y. Tanaka, and K. Tsumoto. Circular dichroism spectra demonstrate formation of the thrombin-binding DNA aptamer G-quadruplex under stabilizing-cation-deficient conditions. *Biochemical and Biophysical Research Communications*, 352(3):812-817, 2007. 62

[OBB+01] W. K. Olson, M. Bansal, S. K. Burley, R. E. Dickerson, M. Gerstein, S. C. Harvey, U. Heinemann, X. J. Lu, S. Neidle, Z. Shakked, H. Sklenar, M. Suzuki, C. S. Tung, E. Westhof, C. Wolberger, and H. M. Berman. A standard reference frame for the description of nucleic acid base-pair geometry. *Journal of Molecular Biology*, 313(1):229-237, Oct 2001. 20, 22, 27, 30

[ONP08] M. Orozco, A. Noy, and A. Perez. Recent advances in the study of nucleic acid flexibility by molecular dynamics. *Current Opinion in Structural Biology*, 18(2):185-193, Apr 2008. 80

[PAHA09] H. Pei, S. Allison, B. M. H. Haynes, and D. Augustin. Brownian dynamics simulation of the diffusion of rods and wormlike chains in a gel modeled as a cubic lattice: Application to DNA. *The Journal of Physical Chemistry B*, 113(9):2564-2571, 2009. 81

[PCV99] A. A. Podtelezhnikov, N. R. Cozzarelli, and A. V. Vologodskii. Equilibrium distributions of topological states in circular DNA: interplay of supercoiling and knotting. *Proceedings of the National Academy of Sciences*, 96(23):12974-12979, Nov 1999. 79

[PGH+04] E.F. Pettersen, T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, and T.E. Ferrin. UCSF Chimera a visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25:1605-1612, Oct 2004. xxv, 29, 30, 33

[PH98] M. J. Packer and C. A. Hunter. Sequence-dependent DNA structure: the role of the sugar-phosphate backbone. *Journal of Molecular Biology*, 280(3):407-420, Jul 1998. 26

[PLWH11] C. Pagba, S. Lane, and S. Wachsmann-Hogiu. Conformational changes in quadruplex oligonucleotide structures probed by raman spectroscopy. *Biomedical Optics Express*, 2(2):207-217, 2011. 62

[PS84] C.O. Pabo and R.T. Sauer. Protein-DNA recognition. *Annual Review of Biochemistry*, 53:293-321, 1984. 14

[Rap06] A. N. Raposo. Poligonização de Superfícies Implícitas por Amostragem baseada num Corrector de Newton. Master's thesis, Universidade da Beira Interior, Covilhã, Portugal, 2006. 47

[RB83] W. Rheinboldt and J. Burkardt. A locally parameterized continuation process. *ACM Transactions on Mathematical Software*, 9:236-246, 1983. 47

[RG06] A. N. Raposo and A. J. P. Gomes. Polygonization of multi-component non-manifold implicit surfaces through a symbolic-numerical continuation algorithm. In *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, pages 399-406. ACM Press, 2006. xi, 59

[RG12] A. N. Raposo and A. J.P. Gomes. 3D molecular assembling of B-DNA sequences using nucleotides as building blocks. *Graphical Models*, 74(4):244-254, 2012. GMP2012. xii, xxv, 12, 16, 37, 38, 40, 41, 42, 76, 83

[RG14] A. N. Raposo and A. J. P. Gomes. Efficient deformation algorithm for plasmid DNA simulations. *BMC Bioinformatics*, 15(1):301, Sep 2014. 99

[RG15] A. N. Raposo and A. J. P. Gomes. isDNA: A Tool for Real-Time Visualization of Plasmid dna Monte-Carlo Simulations in 3D. *Lecture Notes in Bioinformatics*, 9044(Part II):566-577, 2015. xxv, 16, 40, 73

[Rhe87] W. Rheinboldt. On a moving frame algorithm and the triangulation of equilibirum manifolds. In T. Kuper, R. Seydel, and H. Troger, editors, *ISNM79: Bifurcation: Analysis, Algorithms, Applications*, pages 256-267, 1987. 47

[RPK07] J. Ryu, R. Park, and D.S. Kim. Molecular surfaces on proteins via beta shapes. *Computer-Aided Design*, 39(12):1042-1057, 2007. 45, 73

[RPK09] J. Ryu, R. Park, and D.S. Kim. Triangulation of molecular surfaces. *Computer-Aided Design*, 41(6):463-478, 2009. 46

[RPS+07] J. Ryu, R. Park, J. Seo, C. Kim, H. Lee, and D.S. Kim. Real-time triangulation of molecular surfaces. In *ICCSA (1)*, pages 55-67, Berlin / Heidelberg, 2007. Springer. 11

[RQG09] A. N. Raposo, J. A. Queiroz, and A. J. P. Gomes. Triangulation of molecular surfaces using an isosurface continuation algorithm. In *Proceedings of the 2009 International Conference on Computational Science and its Applications*, pages 145-153. IEEE Computer Society, 2009. xi, xii, 11, 41, 60, 83

[RZP09] G. L. Randall, L. Zechiedrich, and B. M. Pettitt. In the absence of writhe, DNA relieves torsional stress with localized, sequence-dependent structural failure to preserve b-form. *Nucleic Acids Research*, 37:5568-5577, 2009. 69

[SBT02] D. Serban, J. M. Benevides, and G. J. Thomas. DNA secondary structure and Raman markers of supercoiling in Escherichia coli plasmid pUC19. *Biochemistry*, 41(3):847-853, Jan 2002. 63, 69

[Sch10] Schrödinger, L.L.C. The PyMOL molecular graphics system, version 1.3r1. The PyMOL Molecular Graphics System, Version 1.3, Schrödinger, LLC., August 2010. 29

[SFCW13] R. Salomon-Ferrer, D. A. Case, and R. C. Walker. An overview of the AMBER biomolecular simulation package. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 3(2):198-210, 2013. 34

[SH97] B. Stander and J. Hart. Guaranteeing the topology of an implicit surface polygonization for interactive modeling. *Computer Graphics*, 31(3):279-286, August 1997. 46

[SST76] Y. Sakakibara, K. Suzuki, and J. I. Tomizawa. Formation of catenated molecules by replication of colicin E1 plasmid DNA in cell extracts. *Journal of Molecular Biology*, 108(3):569-582, Dec 1976. 17

[ST88] D. M. Soumpasis and C. S. Tung. A rigorous basepair oriented description of DNA structures. *Journal of Biomolecular Structure and Dynamics*, 6(3):397-420, Dec 1988. 19, 41

[STB93] E. S. Shpigelman, E. N. Trifonov, and A. Bolshoy. CURVATURE: software for the analysis of curved DNA. *Computer Applications in the Biosciences*, 9(4):435-440, Aug 1993. 32, 34

[STPQ05] F. Sousa, C. T. Tomaz, D. M. F. Prazeres, and J. A. Queiroz. Separation of supercoiled and open circular plasmid DNA isoforms by chromatography with a histidine-agarose support. *Analytical Biochemistry*, 343(1):183-185, Aug 2005. 62

[TIBK83] B. Tidor, K. K. Irikura, B. R. Brooks, and M. Karplus. Dynamics of DNA oligomers. *Journal of Biomolecular Structure and Dynamics*, 1(1):231-252, Oct 1983. 79, 80

[TKKI98] H. Tomita, M. Kai, T. Kusama, and A. Ito. Monte Carlo simulation of DNA strand-break induction in supercoiled plasmid pBR322 DNA from indirect effects. *Radiation and Environmental Biophysics*, 36:235-241, Feb 1998. 63

[TM94] Y. Timsit and D. Moras. DNA self-fitting: the double helix directs the geometry of its supramolecular assembly. *The EMBO Journal*, 13:2737-2746, Jun 1994. 63

[TS96] C. S. Tung and D. M. Soumpasis. Structural prediction of A- and B-DNA duplexes based on coordinates of the phosphorus atoms. *Biophysical Journal*, 70(2):917-923, Feb 1996. 26

[TSH88] W. Thumm, A. Seidl, and H. J. Hinz. Energy-structure correlations of plasmid DNA in different topological forms. *Nucleic Acids Research*, 16(24):11737-11757, Dec 1988. 81

[TW80] Y. Tse and J. C. Wang. E. coli and M. luteus DNA topoisomerase I can catalyze catenation of decatenation of double-stranded DNA rings. *Cell*, 22(1 Pt 1):269-276, Nov 1980. 17

[VALFK79] A. V. Vologodskii, V. V. Anshelevich, A. V. Lukashin, and M. D. Frank-Kamenetskii. Statistical mechanics of supercoils and the torsional stiffness of the DNA double helix. *Nature*, 280(5720):294-298, Jul 1979. 79

[VB93] A. Varshney and F. P. Brooks. Fast analytical computation of Richards's smooth molecular surface. In *Proceedings of the 4th Conference on Visualization (VIS '93)*, pages 300-307, 1993. 46

[VFG99] L. Velho, L. Figueiredo, and J. Gomes. A unified approach for hierarchical adaptive tesselation of surfaces. *ACM Transactions on Graphics*, 18(4):329-360, 1999. 46

[VKD87] E. Von Kitzing and S. Diekmann. Molecular mechanics calculations of dA12.dT12 and of the curved molecule d(GCTCGAAAAA)4.d(TTTTTCGAGC)4. *European Biophysics Journal*, 15(1):13-26, 1987. 19, 41

[VLK+92] A. V. Vologodskii, S. D. Levene, K. V. Klenin, M. Frank-Kamenetskii, and N. R. Cozzarelli. Conformational and thermodynamic properties of supercoiled DNA. *Journal of Molecular Biology*, 227(4):1224-1243, Oct 1992. 78, 79, 82, 86, 89, 90

[VLR+65] J. Vinograd, J. Lebowitz, R. Radloff, R. Watson, and P. Laipis. The twisted circular form of polyoma viral DNA. *Proceedings of the National Academy of Sciences*, 53(5):1104-1111, May 1965. 15, 16

[VM97] A. V. Vologodskii and J. F. Marko. Extension of torsionally stressed DNA by external force. *Biophysical Journal*, 73(1):123-132, Jul 1997. 79

[Vol06] A. Vologodskii. Brownian dynamics simulation of knot diffusion along a stretched DNA molecule. *Biophysical Journal*, 90(5):1594-1597, Mar 2006. 81

[Vol07] A. Vologodskii. Monte Carlo simulation of DNA topological properties. In MichaelIlych Monastyrsky, editor, *Topology in Molecular Biology*, Biological and Medical Physics, Biomedical Engineering, pages 23-41. Springer Berlin Heidelberg, 2007. 79, 83

[vOW93] K. van Overveld and B. Wyvill. Shrinkwrap: an adaptive algorithm for polygonizing an implicit surface. Technical Report 93/514/19, Department of Computer Science, The University of Calgary, Calgary, Canada, March 1993. 47

[vOW04] K. van Overveld and B. Wyvill. Shrinkwrap: An efficient adaptive algorithm for triangulating an iso-surface. *The Visual Computer*, 20(6):362-379, 2004. 47

[VR09] A. Vologodskii and V. V. Rybenkov. Simulation of DNA catenanes. *Physical Chemistry Chemical Physics*, 11:10543-10552, 2009. 79

[VS62] P. Verdier and W. Stockmayer. Monte Carlo calculations on the dynamics of polymers in dilute solution. *The Journal of Chemical Physics*, 36(1):227-235, 1962. 78

[VTKY07] M. Vlachos, B. Taneri, E. Keogh, and P. S. Yu. Visual exploration of genomic data. In *Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'07)*, pages 613-620. Springer-Verlag, 2007. 62

[Wal97] D. Walther. WebMol--a Java-based PDB viewer. *Trends Biochem. Sci.*, 22(7):274--275, Jul 1997. 29

[WC53] J. D. Watson and F. H. C. Crick. Molecular structure of nucleic acids. *Nature*, 171:737-738, April 1953. xi, 4, 12, 64

[WDT+80] R. Wing, H. Drew, T. Takano, C. Broka, S. Tanaka, K. Itakura, and R. E. Dickerson. Crystal structure analysis of a complete turn of B-DNA. *Nature*, 287:755-758, 1980. 70

[Weg01] R. Wegscheider. Über simultane Gleichgewichte und die Beziehungen zwischen Thermodynamik und Reactionskinetik homogener Systeme. *Monatshefte für Chemie / Chemical Monthly*, 32(8):849-906, 1901. 86

[Whi95] J. H. White. Winding the double helix: Using geometry, topology, and mechanics of DNA. In Eric S. Lander and Michael S. Waterman, editors, *Calculating the Secrets of Life: Contributions of the Mathematical Sciences to Molecular Biology*, pages 153-178. National Academy Press, Washington, D.C., USA, 1995. 69

[WMC87] J. H. White, K. C. Millett, and N. R. Cozzarelli. Description of the topological entanglement of DNA catenanes and knots by a powerful method involving strand passage and recombination. *Journal of Molecular Biology*, 197(3):585-603, Oct 1987. 17

[Wol06] Hans J. Wolters. Geometric modeling applications in rational drug design: a survey. *Computer Aided Geometric Design*, 23(6):482-494, 2006. 45

[WS67] J. C. Wang and H. Schwartz. Noncomplementarity in base sequences between the cohesive ends of coliphages 186 and λ and the formation of interlocked rings between the two DNA's. *Biopolymers*, 5(10):953-966, 1967. 17

[WS02] R. Watanabe and K. Saito. Monte Carlo simulation of strand-break induction on plasmid DNA in aqueous solution by monoenergetic electrons. *Radiation and Environmental Biophysics*, 41:207-215, 2002. 63

[WS10] G. Witz and A. Stasiak. DNA supercoiling and its role in DNA decatenation and unknotting. *Nucleic Acids Research*, 38(7):2119-2133, Apr 2010. 17

[WSS99] J. Weiser, P. S. Shenkin, and W. C. Still. Optimization of Gaussian surface calculations and extension to solvent-accessible surface areas. *Journal of Computational Chemistry*, 20(7):688-703, 1999. 46

[WV63] R. Weil and J. Vinograd. The cyclic helix and cyclic coil forms of polyoma viral DNA. *Proceedings of the National Academy of Sciences*, 50:730-738, Oct 1963. 15

[YHZC00] Z. Yang, Z. Haijun, and O. Y. Zhong-Can. Monte Carlo implementation of supercoiled double-stranded DNA. *Biophysical Journal*, 78(4):1979-1987, Apr 2000. 79

[YK09] Y. Yonetani and H. Kono. Sequence dependencies of DNA deformability and hydration in the minor groove. *Biophysical Journal*, 97(4):1138-1147, Aug 2009. 18

[YPVM85] C. Yanisch-Perron, J. Vieira, and J. Messing. Improved M13 phage cloning vectors and host strains: nucleotide sequences of the M13mp18 and pUC19 vectors. *Gene*, 33(1):103-119, 1985. 90

[ZCSO10] G. Zheng, L. Czapla, A. R. Srinivasan, and W. K. Olson. How stiff is DNA? *Physical Chemistry Chemical Physics*, 12(6):1399-1406, Feb 2010. 79

[ZLO09] G. Zheng, X. J. Lu, and W. K. Olson. Web 3DNA--a web server for the analysis, reconstruction, and visualization of three-dimensional nucleic-acid structures. *Nucleic Acids Research*, 37(Web Server issue):W240-246, Jul 2009. 27, 29, 41

[ZSC+01] G. Zuccheri, A. Scipioni, V. Cavaliere, G. Gargiulo, P. De Santis, and B. Samorì. Mapping the intrinsic curvature and flexibility along the DNA chain. *Proceedings of the National Academy of Sciences*, 98(6):3074-3079, 2001. 15

[ZXB06] Y. Zhang, G. Xu, and C. Bajaj. Quality meshing of implicit solvation models of biomolecular structures. *Computer Aided Geometric Design*, 23(6):510-530, 2006. 46, 48, 65

[ZXB07] W. Zhao, G. Xu, and C. Bajaj. An algebraic spline model of molecular surfaces. In *Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling (SPM'07)*, pages 297-302. ACM Press, 2007. 46

[ZZY03] H. Zhou, Y. Zhang, and Z.-C. O. Yang. The elastic theory of a single DNA molecule. *Journal of Physics*, 61(2):353-360, 2003. 63