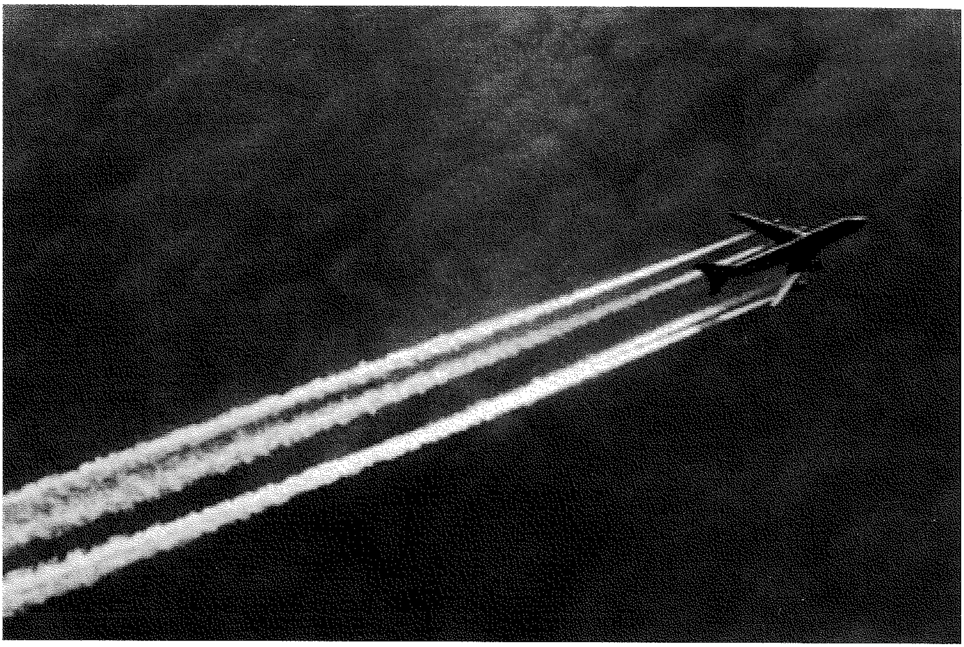
Snel en robuust

Afscheidsrede

Prof. Dr. Ir. P. Wesseling



6 juni 2007

TU Delft

Faculteit Electrotechniek, Wiskunde en Informatica

Technische Universiteit Delft

Snel en robuust

Afscheidsrede

Prof.dr.ir. P. Wesseling

Hoogleraar Numerieke Wiskunde

aan de Faculteit Electrotechniek, Wiskunde en Informatica

van de Technische Universiteit Delft

6 juni 2007

Mijnheer de Rector Magnificus, leden van het College van Bestuur, collegae hoogleraren en andere leden van de universitaire gemeenschap,

Zeer gewaardeerde toehoorders,

Dames en heren,

Op een warme augustusmiddag in 1959 arriveerde ik op het fraaie spoorwegstation alhier om in Delft vliegtuigbouwkunde te gaan studeren. Als ik toen had geweten dat ik later dertig jaar lang vrijwel dagelijks op dat station zou aankomen en vertrekken, dan zou ik dat een afschuwelijk vooruitzicht hebben gevonden. Maar het vooruitzicht van een hoogleraarschap in Delft zou mij daarentegen wel hebben toegelachen. Dat leek echter in de jaren zestig uitgesloten, want veel vacatures leken er niet te zijn. Veel hoogleraren waren nog jong. Zoals mijn gewaardeerde leermeester wijlen professor Steketee. Deze theoretisch aerodynamicus maakte mij enthousiast voor de wiskunde. De wiskundevakken, inclusief de numerieke wiskunde, die gedoceerd werd door mijn voorganger professor van Spiegel en zijn medewerkers, vond ik boeiend, maar omdat je bij de Onderafdeling Wiskunde niet kon vliegen, terwijl je destijds bij de Onderafdeling Vliegtuigbouwkunde als student je vliegbrevet kon halen, bleef ik bij Vliegtuigbouwkunde. Van Spiegel vertrok later naar het departement als Directeur-Generaal voor het Wetenschapsbeleid. Niet vermoedend dat ik hem vijftien jaar later zou gaan opvolgen, verzuimde ik zijn colleges bij te wonen, omdat ze samenvielen met de colleges wijsbegeerte van professor van Riessen. Het gevolg was een onvoldoende voor het studievak numerieke wiskunde. Maar ik heb niet verzuimd om de lezing van van Spiegel bij te wonen, die hij op 21 april jongstleden hield ter gelegenheid van het vijftigjarig bestaan van de opleiding Technische Wiskunde.

Steketee raadde mij aan om na het behalen van het ingenieursdiploma in de Verenigde Staten verder te gaan studeren. Daar was ik onmiddellijk enthousiast voor, en mijn vriendin Tineke gelukkig ook. In augustus 1964 trouwden wij en vertrokken naar het California Institute of Technology. Laatst vroeg een promovendus wie mijn promotor was, zodat we konden aanhaken bij de stamboom van het Mathematics Genealogy project [1], waarin wordt bijgehouden wie bij wie is gepromoveerd. En wat blijkt: via mijn promotor Paco Lagerstrom bij Caltech komen we in vier generaties bij David Hilbert terecht; wat wil je nog meer? En dan gaat het verder terug tot niemand minder dan Carl Friedrich Gauß. Voor de niet-wiskundigen: de laatste heeft als bijnaam de prins der wiskunde, en Hilbert was de grootste wiskundige van de twintigste eeuw. Erg exclusief is deze afstamming nu ook weer niet, want Gauß heeft ruim 36.000 afstammelingen, waarvan 33 via mij, inclusief vier die bij het CWI onder leiding van Piet Hemker werkzaam zijn geweest. Hieruit blijkt de ijver van de numerieke groep, maar dat niet alleen, want er zitten ook co-promoties met andere faculteiten bij, hetgeen illustreert dat er overal tegenwoordig wiskunde onder de motorkap zit, zodat men er nogal eens een numericus bij haalt.

Maar het wordt tijd om aandacht te schenken aan mijn eigenlijke onderwerp, dat ik heb aangeduid als 'Snel en robuust'. Toen in de jaren vijftig de motorisering van Nederland langzaam begon, schafte mijn vader een bromfiets aan van het merk Puch, en ik kocht een boekje getiteld 'Tuning for speed'. De aanwijzingen in dat boek volgend, bleek het niet moeilijk de snelheid van de Puch aanzienlijk op te voeren. Ook het geluidsniveau nam toe, maar daar stoorde men zich toen nog niet aan. 'Tuning for speed' bleek later een belangrijk thema in het werk van de numerieke groep; daar kom ik zo op terug. Mijn vader gaf mij de vrije hand, mits hijzelf de brommer op ieder gewenst tijdstip makkelijk kon starten. Dat nu valt onder de noemer van robuustheid. Als een optimale eigenschap van een apparaat of procédé alleen gerealiseerd kan worden onder bijzondere omstandigheden of door tussenkomst van een expert, dan heet die eigenschap niet robuust.

Snelheid en robuustheid dus. Wat heeft dat te maken met het vakgebied van de numerieke groep? Dat vakgebied is de numerieke wiskunde, oftewel de studie van rekenmethoden voor het oplossen van wiskundige problemen. Met de draai die wij er in Delft aan geven kan men beter spreken van wetenschappelijk rekenen of 'computational science'. Modellen van de werkelijkheid worden gesimuleerd, wereldmodellen zogezegd. De meesten zullen hierbij denken aan 'virtual reality' en computer games, en inderdaad heeft de wereld van de 'gaming' de numerieke wiskunde nodig, zoals onlangs werd uitgelegd in de TV serie 'Delft Blauw' met medewerking van Kees Vuik. Bij 'serious gaming' gaat het om opleiden en trainen door computersimulatie. Maar ook voor games ter vermaak is 'computational science' nodig. Zo zei een 'lead programmer' bij een 'games maker' in een interview in ons faculteitsblad Quadraad (Juni 2006), en ik citeer: *'Lineaire algebra bijvoorbeeld is iets dat overal gebruikt wordt binnen "games development", en gelukkig is daar op de TU Delft redelijk diep op ingegaan'.* Aldus een informatica alumnus over een onderwijsvak dat ons na aan het hart ligt. Tot zover over gaming als applicatie. In het vakgebied scientific computing wordt simulatie met computers vooral ingezet ten dienste van wetenschap en techniek. Onderwerpen die recent zijn behandeld in proefschriften in de numerieke groep [2-7] zijn de berekening van diverse fenomenen in de stromingsleer, van golven in de aardbodem, van de warmte-behandeling van legeringen ter verbetering van de mechanische eigenschappen, en simulatie van het schrijfproces op DVD's. Deze variatie van onderwerpen illustreert dat de wiskunde overal goed voor is. En ook, dat de leden van de numerieke groep flexibel zijn en gewaardeerde samenwerkingspartners voor materiedeskundigen uit andere disciplines.

Met de numeriek-wiskundige virtuele wereldmodellen kunnen nieuwe ontwerpen veel gemakkelijker gevarieerd en geoptimaliseerd worden dan met fysische modellen. Verder kunnen situaties bestudeerd worden die onmogelijk of slechts met onaanvaardbaar risico in de praktijk onderzocht kunnen worden. Zodoende mag men rustig zeggen dat er een nieuwe wetenschappelijke aanpak is ontstaan: naast theorie en experiment hebben we nu 'scientific computing' als derde pijler van wetenschap en techniek. Het enorme belang hiervan is vergelijkbaar met de uitvinding van nieuwe wetenschappelijke instrumenten, zoals de telescoop en de microscoop in de zeventiende eeuw.

Het simuleren van wereldmodellen met computers heeft zo'n grote vlucht heeft kunnen nemen doordat de processoren van computers voortdurend sneller en de geheugens groter en goedkoper worden. Dat gaat al een hele tijd zo en dat ervaringsfeit wordt de wet van Moore genoemd. De snelste computer gaat van 59,7 Gigaflops in 1993 naar 280,6 Teraflops in 2006 [8], bijna een verdubbeling per jaar. Flops betekent rekenoperaties per seconde (floating point operations per second); 1 Gigaflops is 1.000.000.000 (10^9) flops, en 1 Teraflops = 1000 Gigaflops. Nu is niet alles goud wat hier blinkt, want een supercomputer haalt vrijwel nooit zijn topsnelheid. Dat komt doordat men vele processoren parallel (simultaan) gebruikt om een grote rekensnelheid te halen. Echter, uitwisseling van tussenresultaten tussen processoren is relatief langzaam, zodat processoren regelmatig op elkaar moeten wachten als het algoritme veel uitwisseling vergt. Ondanks grote inspanningen gedurende de afgelopen vijftien jaar is dit nog steeds het grote probleem bij het gebruik van supercomputers. De vraag is of massaal parallellisme haalbaar is. De tijd zal het leren.

Het is denkbaar dat supercomputing in de toekomst vooral zal gebeuren door via het internet samenwerkende PC's, het zogenaamde gridcomputing. Dat gebeurt nu al voor toepassingen die zich daar goed voor lenen, omdat het werk makkelijk opgesplitst kan worden in onafhankelijke deeltaken. Dan kun je het werk verdelen over een groot aantal PC's die niet op elkaar hoeven te wachten. Een voorbeeld hiervan is het SETI@home project. SETI staat voor Search for Extra-Terrestrial Intelligence. Het gaat er om signalen uit de ruimte te herkennen als boodschappen van intelligent leven op andere planeten. Een intelligent signaal zoeken in de radiogolven die we uit de ruimte opvangen is nog moeilijker dan het zoeken van een naald in een hooiberg. Maar als je het vele werk verdeelt over heel veel computers heb je misschien een kans. In SETI@home doen ongeveer 1,3 miljoen PCs mee. Deze activiteit wordt beschouwd als serieuze wetenschap.

Ter afwisseling even een kleine uitweiding naar aanleiding van SETI. Hoe herken je intelligentie in een signaal? Dat weten we niet goed. Men hoopt maar dat er iets kunstmatigs te ontdekken valt aan de radiogolven die men opvangt. Voor sommigen hangt het er bovendien van af waar het signaal vandaan komt. Stel dat het volgende signaal ontvangen wordt:

GCAAACTTGTAGCACGTGGAGAATTCTGGGACATAGTTGCAATAA

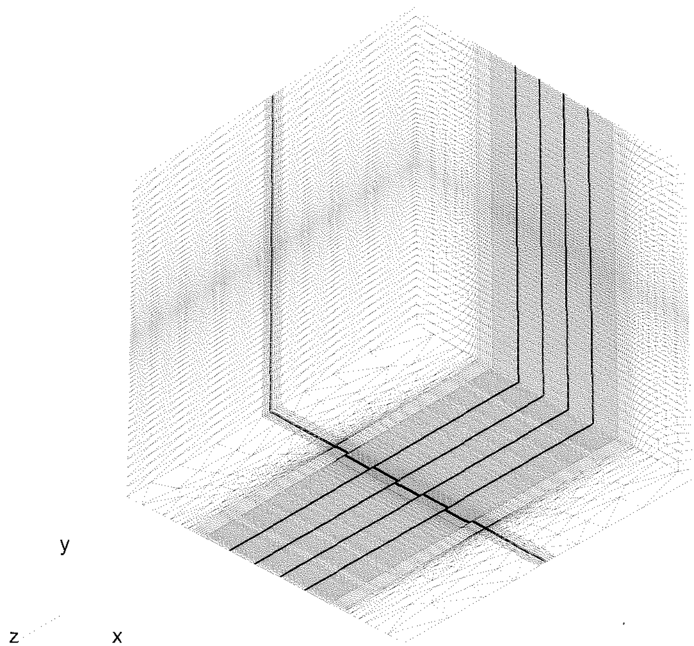
Dan zal men het ongetwijfeld tot de taak van de wetenschap rekenen om uit te dokteren of er intelligentie aan ten grondslag ligt of niet. Maar als je erbij zegt dat dit geen signaal uit de ruimte is maar de volgorde van de bases in een stukje DNA [9], dan zijn er velen die zonder meer verklaren dat hier geen intelligentie achter zit en dat deze structuur toevallig is ontstaan. Anderen daarentegen willen serieus proberen wetenschappelijk aan te tonen dat er intelligentie achter zit, intelligent design zogezegd. Van sommigen mag dat geen wetenschap heten; maar ik maak me sterk dat dezelfde personen er geen bezwaar tegen zouden hebben als het om een boodschap uit de ruimte ging. Het zoeken naar sporen van intelligentie in buitenaardse radiogolven en in DNA lijken mij beide acceptabele wetenschappelijke activiteiten, maar ik verwacht er niet veel van.

Men kan zich behoorlijk opwinden over intelligent design, getuige bijvoorbeeld de provocerende titel van Dawkins' laatste boek 'The God Delusion' [10]. Daartegenover staan boeken van zijn collega in Oxford, Alistair McGrath, zoals bijvoorbeeld 'The Dawkins Delusion?' [11]. 'Delusion' betekent waanidee. Het gaat dus van dik hout zaagt men planken daar in Oxford. Een subtielere aanpak vinden we bij de Amerikaanse filosoof Plantinga (zie [12]), die met sterke argumenten betoogt dat het naturalisme zoals Dawkins dat uitdraagt geen grond biedt voor zekerheid omtrent de betrouwbaarheid van ons kenvermogen. So what, denkt u misschien, maar voor het betoog dat Dawkins voert is deze innerlijke strijdigheid wel fataal. Voor interessante beschouwingen van eigen bodem, zie de boeken geredigeerd door Dekker, Meester en van Woudenberg [13,14]. Wetenschap en geloof zijn minder makkelijk te scheiden dan men vaak meent. De geschiedenis leert dat geloof het vermeerderen van kennis niet in de weg hoeft te staan, en zelfs stimulerend kan werken.

Terug naar mijn onderwerp: snel en robuust rekenen. Om met hoge snelheid te kunnen rekenen is snelle computerinfrastructuur niet genoeg; je moet de rekenmethode, het algoritme, op de geheugenstructuur en de netwerkstructuur aanpassen. En dan nog hangt het er van af of het wat oplevert. Het komt veel voor dat een algoritme dat met hoge snelheid draait op een multiprocessor systeem meer werk vergt dan een slecht parallelliserend algoritme, zodat men niets opschiet met het gebruiken van een supercomputer. Het probleem laat zich met het volgende voorbeeld verduidelijken. Stel je gaat regelmatig met de fiets van A naar B, langs een korte route waarin zich een fietsbrug bevindt die niet toegankelijk is voor auto's. Met de auto haal je een hogere snelheid, maar doordat je om moet rijden kan de reistijd toch langer zijn.

Wat te doen om toch vooruit te komen? Dat kan met behulp van de held van mijn verhaal: het snelle en robuuste algoritme. Een algoritme is een voorschrift om een rekenkundig probleem stap voor stap op te lossen met behulp van eenvoudige rekenkundige operaties (optellen, vermenigvuldigen en dergelijke). De numerieke wiskunde gaat over het ontwerpen van efficiënte algoritmen die resultaten opleveren die in de buurt komen van de oplossing van het wiskundige model van, kort door de bocht, een stukje werkelijkheid.

Hoe gaan we te werk bij het simuleren van een wereldmodel door middel van wetenschappelijk rekenen? Om dat in tien minuten te vertellen moet ik flink oversimplificeren. Laten we ons beperken tot toepassingen in de techniek, met als voorbeeld het simuleren wat er gebeurt als je met de computer op een DVD schrijft. Een stukje van de DVD en een deel van de omringende ruimte wordt opgedeeld in kleine stukjes, die we rekencellen noemen. Zoals in deze figuur, afkomstig uit het proefschrift van John Brusche [7].



Waar we zwart zien zijn de rekencellen erg klein, terwille van de nauwkeurigheid. In elke cel worden de fysische wetten toegepast voor de electro-magnetische veldsterkte in de cellen. Dat zijn de onbekenden die we willen bepalen. Hoe kleiner de cellen, hoe nauwkeuriger. Dus hoe meer onbekenden, hoe beter. Laten we het aantal onbekenden *N* noemen. En we hebben *N* vergelijkingen waaruit de onbekenden opgelost kunnen worden. Voor het specificeren van de vergelijkingen zijn typisch gN geheugenplaatsen nodig, met g een probleem-afhankelijk getal. Zoals gezegd, hoe groter N, hoe beter. Dus het computergeheugen is altijd kleiner dan men zich zou wensen. Zoals alles wat met chips te maken heeft verbetert de prijs/prestatie verhouding voortdurend: ruwweg gesproken krijg je elke 18 maanden twee keer zo veel geheugen voor je geld. Geheel volgens de eerder genoemde wet van Moore. Dus N kan steeds maar groter worden. Maar als N toeneemt, neemt de benodigde rekentijd ook toe. In welke mate, dat hangt af van de rekenmethode of algoritme dat je gebruikt. Laten we zeggen dat de rekentijd op onze computer cN^{α} is, met c een of ander getal en α een exponent. Dit eenvoudige model voor de rekentijd is redelijk realistisch. Hoe groter α, hoe slechter natuurlijk. De exponent α hangt af van het gehanteerde algoritme. Bijvoorbeeld, voor een bekend algoritme van stamvader Gauss geldt $\alpha=2$ voor een belangrijke klasse van problemen. Stel nu dat we een nieuwe computer aanschaffen die twee keer zo snel is en twee keer zoveel geheugen heeft als de vorige. We nemen onze kans waar en verdubbelen het aantal cellen. Daardoor neemt het rekenwerk toe, maar dat wordt gedeeltelijk gecompenseerd doordat de nieuwe computer sneller is. De nieuwe rekentijd wordt $c(2N)^{\alpha}/2$, oftewel $2^{\alpha-1}$ maal de oude rekentijd. Indien $\alpha=2$ wordt de rekentijd dus twee keer zo groot. Als α groter is dan 1, dan is dat niet best, want dan neemt de rekentijd met elke nieuwe generatie computers toe, of we moeten stoppen met het groter maken van N. Alleen maar investeren in de computers, de hardware, helpt dus niet. De α moet omlaag. En dat kan alleen door vooruitgang in de wiskunde, door verbetering van het algoritme, de software, zeg maar. Software wordt voortdurend veeleisender. Dat wordt wel de wet van Wirth genoemd: *software wordt sneller langzaam dan de computer sneller wordt*. Dat is de tegenhanger van de wet van Moore. En als α groter is dan 1 dan gaat Wirth het winnen van Moore, en lopen we tegen grenzen aan bij het wetenschappelijk rekenen. Gevraagd daarom: snellere rekenmethoden. En daarmee ben ik aangekomen bij wat ik bedoel met de titel van mijn voordracht: snel en robuust. Tuning for speed, van rekenvoorschriften, dat is mooi werk, met een sportief element van wedijver. Dat heeft mij nogal intensief bezig gehouden, samen met gelijk gezinden in de groep, en elders in de wereld. En dat heeft toekomst. Want, zoals ik juist heb betoogd, maken snellere en grotere computers verbetering van algoritmen niet overbodig, maar juist meer gewenst. Bovendien: voor het verbeteren van rekenmethoden en wiskundige modellen zijn de grenzen niet in zicht, terwijl het sneller maken van computers niet eindeloos kan doorgaan. De toekomst is aan het algoritme en de software!

De ideale waarde van α is dus 1. Dan blijft de rekentijd gelijk als we met een snellere computer een groter probleem oplossen. Sommige rekenmethoden halen $\alpha=1$, doch alleen onder bijzondere omstandigheden. Zulke methoden noemen we niet robuust. Maar omstreeks 1970 heeft zich een aanpak aangediend, die in principe robuust is en snel, want $\alpha = 1$ wordt gehaald. Dat is de zogenaamde multirooster methode. In Nederland zijn we hier in een vroeg stadium op ingesprongen, met name Piet Hemker bij het CWI in Amsterdam en ik in Twente en vervolgens in Delft, en jongere leden van de reeds vele jaren florerende nationale Werkgemeenschap Scientific Computing namen de fakkel over, zoals Kees Oosterlee in Delft en Bonn en Arnold Reusken in Aken. Dat heeft ons vruchtbare jaren opgeleverd met een aantal promoties bij Hemker en mij [15-18] en de nodige activiteiten, zoals twee maal de Europese Multirooster Conferentie in Nederland [19,20], de laatste met name aangedreven door Kees Oosterlee. En vanmorgen heeft collega Wittum uit Heidelberg ons weer met dit onderwerp weten te boeien. Multirooster is ongetwijfeld een der belangrijkste ontwikkelingen in de numerieke wiskunde in de tweede helft van de twintigste eeuw. Als gezegd, de multirooster methode geeft $\alpha=1$. Klaar dus, zult u zeggen. Ander werk gaan doen. Maar zo ligt het niet. Multirooster is meer een methodologie dan een klip-en-klaar rekenvoorschrift, en toepassing kan behoorlijk ingewikkeld zijn. Niettemin is multirooster min of meer geruisloos geïntegreerd en onder de motorkap verdwenen van belangrijke commerciële computerprogramma's voor het simuleren van technisch-fysische apparaten en verschijnselen, zoals STAR-CD en FEMLAB.

Een andere aanpak die heel belangrijk is om α klein te maken is die der Krylov deelruimte methoden. De α is 1,25 zeg maar (erg gesimplificeerd). Krylov methoden laten zich in een paar regels eenduidig formuleren, zijn erg makkelijk te programmeren en passen naadloos in de klassieke discipline van de numerieke lineaire algebra. Daardoor zijn ze erg populair, ook al is hun α iets groter dan voor multirooster. Nederlandse numerici spelen op dit gebied een belangrijke rol. Het meest geciteerde artikel in de hele wiskunde in de jaren negentig gaat over een Krylov methode, namelijk Bi-CGSTAB, en is van de hand van Henk van der Vorst in Utrecht [21]. De Delftse numeriekegroep blaast een flinke partij mee op het gebied van Krylov methoden. Daarover nu iets meer.

Toen ik in 1977 bij de numerieke groep in Delft aankwam als hoogleraar was daar Peter Sonneveld bezig met het ontwikkelen van een methode die hij de 'Induced Dimension Reduction' oftewel IDR methode noemde. Hoewel Sonneveld er een andere interpretatie aan geeft kun je dit als een Krylov methode beschouwen. Mijn eerste project in Delft was om samen met Sonneveld een vergelijking te maken tussen IDR en multigrid [22]. Uiteindelijk liet Sonneveld in 1989 een publikatie [23] het licht zien, en had zijn methode inmiddels omgedoopt tot CGS. Dat artikel trok wereldwijd veel aandacht. CGS was snel, maar niet robuust. CGS kon onverwacht uit de bocht vliegen. Van 1984 tot 1990 was Van der Vorst lid van onze numerieke groep als hoogleraar, of zoals hij het zelf noemde, als HIO: hoogleraar in opleiding. Van der Vorst had toen al furore gemaakt met een methode [24] voor het geval dat de matrix symmetrisch is, samen met Koos Meijerink die bij het Shell laboratorium in Rijswijk werkte. En Van der Vorst ontdekte hoe CGS robuust gemaakt kon worden. Dat leidde tot de eerder genoemde bekende Bi-CGSTAB methode [21]. Verrassend is dat na zijn pensionering Sonneveld na dertig jaar de IDR draad weer heeft opgepakt. Samen met Martin van Gijzen is IDR onlangs doorontwikkeld tot IDR(s) [25], dat de snelheid van Bi-CGSTAB ruimschoots overtreft, zonder aan robuustheid in te boeten. Dat gaat vast weer een invloedrijke publicatie opleveren.

Wat nut ons nu deze sportieve wedijver op het gebied van snelle en robuuste algoritmen? Laat ik volstaan met een voorbeeld. In 'De Telegraaf' van 18 maart 2006 stond een artikel getiteld 'Meer inzicht in olievoorraden dankzij Delftse wiskundige'. Om een lang verhaal kort te maken, er was succes geboekt met het sneller maken van een rekenmethode voor de zogenaamde Helmholtz vergelijking. En dat helpt bij het vinden en winnen van aardolie. Daar ging het proefschrift van Yogi Erlangga [5] over. Dat viel duidelijk onder het thema 'snel en robuust': het golfpatroon van seismische golven in oliehoudende lagen wordt veel sneller en onder grotere variaties van omstandigheden berekend dan met eerdere methoden. De ontwikkelde rekenmethode vindt nu ingang in de industrie. De Helmholtz vergelijking sneller oplossen - dat wordt door velen nagestreefd. Dat Yogi hier opeens furore mee maakte, dat kwam door gelijktijdige toepassing van preconditionering, multigrid en Krylov. In plaats van ze te laten wedijveren kun je deze methoden ook laten samenwerken.

'Scientific computing' is goed voor van alles en nog wat. De afgelopen decennia is de computational science tot wasdom gekomen en doorgedrongen in tal van technische disciplines, waarin het als gereedschap nu op veel plaatsen gebruikt wordt. De ECCOMAS associatie is op dit gebied een belangrijk forum in Europa. Met veler medewerking kon het de tweejaarlijkse conferentie van ECCOMAS [26] vorig jaar in Nederland gehouden worden.

Zij die intensief gebruik maken van 'scientific computing' bij de TU Delft, afkomstig uit vijf faculteiten, werken samen en houden voeling met elkaar in het Delft Center for Computational Science and Engineering [27], gestimuleerd door het College van Bestuur. Een goed initiatief, dat meerwaarde oplevert in de vorm van samenwerking en uitwisseling.

Ik vat samen. Numerieke algoritmen stellen ons in staat wereldmodellen te simuleren met computers, voor vermaak of training (gaming), of ten dienste van ontwerp en vervaardiging van industriële producten, of tal van andere zaken. Computersimulatie neemt een grote vlucht.

Proefmodellen worden overbodig, 'first time right' productie komt in zicht. Naarmate computers sneller worden, neemt verrassenderwijs het belang van snelheid en robuustheid van algoritmen toe. Want computergeheugens worden voortdurend groter, en daarmee de wereldmodellen, want hoe meer vrijheidsgraden, hoe meer realisme. Gebruikt men rekenmethoden uit de oude doos, dan neemt bij het vervangen van een computer het reken*werk* sneller toe dan de reken*snelheid*. De toekomst is aan nieuwe efficiënte algoritmen, draaiend op Gigaflop (10^9) desktops of Petaflop (10^{15}) supercomputers.

Dames en heren,

Aan het einde van mijn afscheidsrede gekomen dank ik u die hier gekomen bent voor uw hartverwarmende belangstelling bij de passage van deze mijlpaal in mijn professionele leven.

De bestuurders van deze universiteit dank ik voor het vertrouwen dat zij steeds in mij hebben gesteld.

De leden van de numerieke groep ben ik bijzonder erkentelijk voor hun inzet en loyaliteit, waardoor wij steeds in een goede werksfeer een gevarieerd en omvangrijk takenpakket konden uitvoeren. In het voorgaande heb ik mij beperkt tot het belichten van een aspect van ons onderzoek. Maar een aanzienlijk deel van jullie inspanningen betreft het onderwijs. Dat is belangrijk, want het opleiden en begeleiden van jonge mensen bij het ontplooien van hun talenten is niet alleen verantwoordelijk werk, maar is tevens de bepalende factor van het maatschappelijk nut van de universiteit. De gestage stroom afstudeerders in de numerieke richting toont aan hoe zeer jullie onderwijs door de studenten gewaardeerd wordt.

Vele anderen dank ik eveneens voor opbouwende en stimulerende collegiale omgang en samenwerking, bijvoorbeeld in het kader van het Delft Center for Computational Science and Engineering, de landelijke Werkgemeenschap Scientific Computing, de nationale onderzoekschool in de stromingsleer het J.M. Burgers Centrum, en de Kontaktgroep Numerieke Stromingsleer.

Eén persoon wil ik met name bedanken, namelijk mijn vrouw Tineke. Dat we deze dag gezond en wel mogen beleven danken we in de eerste plaats aan de zorg van hem die we kennen als onze hemelse Vader, die zijn trouwe zorg in voor- en tegenspoed over alle mensen wil uitstrekken, en die daarvoor naar mijn besef in mijn geval in de eerste plaats jou gebruikt, nu al bijna 43 jaar lang. Dankjewel!

Ik heb gezegd.

Referenties

[1] http://genealogy.math.ndsu.nodak.edu

[2] Duncan Roy van der Heul: *A Staggered Scheme for Nonconvex Hyperbolic Systems of Conservation Laws.* Proefschrift, Technische Universiteit Delft, 28 februari 2002.

[3] Dragan Vidović: *Superlinearly Convergent Unstructured Staggered Schemes for Compressible and Incompressible Flows.* Proefschrift, Technische Universiteit Delft, 14 februari 2005.

[4] Sander van der Pijl: *Computation of Bubbly Flows with a Mass-Conserving Level-Set Method.* Proefschrift, Technische Universiteit Delft, 22 november 2005.

[5] Yogi Ahmad Erlangga: *A Robust and Efficient Iterative Method for the Numerical Solution of the Helmholtz Equation.* Proefschrift, Technische Universiteit Delft, 22 december 2005.

[6] Etelvina Javierre Pérez: *Numerical Methods for Vector Stefan Models of Solid-State Alloys.* Proefschrift, Technische Universiteit Delft, 21 november 2006.

[7] John Brusche: *Mark Formation Model for Optical Rewritable Recording.* Proefschrift, Technische Universiteit Delft, 27 februari 2007.

[8] http://www.top500.org/lists/2006/11/performance_development

[9] http://www.dsi.univ-paris5.fr/genatlas/1/html/FPGT_1.html

[10] Richard Dawkins: *The God Delusion.* Bantam Press, London, 2006

[11] Alistair McGrath: *The Dawkins Delusion?* SPCK Publishing, London, 2007

[12] J.K. Beilby (ed.): *Naturalism defeated? Essays on Plantinga's Evolutionary Argument against Naturalism.* Cornell University Press, Ithaca, 2002

[13] C. Dekker, R. Meester, R. van Woudenberg (red.): *Schitterend ongeluk of sporen van ontwerp? - Over toeval en doelgerichthied in de evolutie.* Uitgeverij Ten Have, Kampen, 2005

[14] C. Dekker, R. Meester, R. van Woudenberg (red.): *En God beschikte een worm.* Uitgeverij Ten Have, Kampen, 2006

[15] H. Schippers: *Multiple Grid Methods for Equations of the Second Kind with Applications in Fluid Mechanics.* Proefschrift, Technische Universiteit Delft, 22 april 1982

[16] S.P. Spekreijse: *Multigrid Solution of the Steady Euler Equations.* Proefschrift, Technische Universiteit Delft, 5 november 1987

[17] B. Koren: *Multigrid Solution of the Steady Navier-Stokes Equations. Applications to Aerodynamics.* Proefschrift, Technische Universiteit Delft, 11 mei 1989.

[18] P.M. de Zeeuw: *Acceleration of Iterative Methods by Coarse Grid Corrections.* Proefschrift, Universiteit van Amsterdam, 22 januari 1997

[19] P.W. Hemker and P. Wesseling (eds.) : *Multigrid Methods IV. Proceedings of the Fourth European Multigrid Conference, Amsterdam, July 6-9, 1994.* Birkhäuser, Basel, 1994

[20] http://pcse.tudelft.nl/emg2005

[21] H.A. van der Vorst: Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM Journal on Scientific Computing* **13**: 631--644, 1992

[22] P.Wesseling and P.Sonneveld: Numerical experiments with a multiple grid and a preconditioned Lanczos type method. In: R.Rautmann (ed.): *Approximation methods for Navier-Stokes problems. Proceedings, Paderborn 1979.* Lecture Notes in Math. **771**, pp. 543-562. Springer-Verlag, Berlin etc. 1980

[23] P. Sonneveld: CGS, a fast Lanczos-type solver for nonsymmetric linear systems. S*IAM Journal on Scientific Computing* **10**:36-52, 1989

[24] J.A. Meijerink and H.A. van der Vorst: An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of Computation* **31**:148-162, 1977

[25] Peter Sonneveld and Martin B. van Gijzen: IDR(s): A family of simple and fast algorithms for solving large nonsymmetric linear systems. Delft University of Technology, Report 07-07. Submitted to *SIAM Journal on Scientific Computing*. Available at http://ta.twi.tudelft.nl/nw/users/gijzen/idrs_report.pdf

[26] http://www.eccomascfd2006.nl

[27] http://www.cse.tudelft.nl