In Focus **High-Performance Computing**

Excerpts from an interview on January 30, 2012

Patrick Regan

At one end of the Internet-video call, Hans Fischer Senior Fellow Markus Hegland (MH) and TUM doctoral candidate Christoph Kowitz (CK) were easing into a warm summer evening. Still shaking off the chill of a winter morning were Carl von Linde Junior Fellow Miriam Mehl (MM), postdoctoral researcher Dirk Pflüger (DP), doctoral candidate Valeriy Khakhutskyy (VK), and TUM Professor Hans-Joachim Bungartz (HB), host of the High-Performance Computing Focus Group (and co-coordinator of a new DFG Priority Program on "Software for Exascale Computing").

Meetings like this, linking TUM's Garching campus with Hegland's home institution, the Australian National University in Canberra, are nothing unusual for this tightly knit group of researchers. The only thing out of the ordinary was devoting a whole meeting to answering a reporter's questions, all the while doing a remarkable job of seeming unaware of the circling, snapping photographer. The aim of the meeting was – and the aim of this article is – to offer a special kind of insight into how the TUM-IAS functions by looking inside its basic working unit, the Focus Group.


skype
TOSHIBA

As in most of the TUM-IAS Focus Groups, basic scientific questions and application-oriented issues intertwine with and enhance each other throughout a program that is, by necessity and design, multidisciplinary. In this case, the research focuses on problems in computer science, mathematics, engineering, and physics that have been brought to the fore by advances in supercomputing technology; at the same time, applications ranging from aerospace engineering to plasma physics stand to benefit from the results. (PR)

PR: The concept of high performance gets redefined a lot more frequently for computers than, say, for automobiles. What does "high-performance computing" mean today, and looking ahead over the next decade or so? And why do the performance trends raise new research problems?

HB: It has always meant the top level of computing power that is available. Today a typical benchmark would be several petaFLOPS, meaning 10^{15} floating-point operations per second, and within ten years we definitely will have entered into the "exa" era, 10^{18} FLOPS. For decades, beginning in the 1960s, it was the machines themselves that dominated the field of high-performance computing. But over the past ten years, algorithms and software have been getting more of the focus.

People see that if you have a fast car, a Ferrari, you need someone who is really able to drive it. That's basically where we are today: People see that if the hardware moves on at such a speed, then there won't be that many groups that are really able to manage all the software issues.

There's another reason this is a turning point. For a long time, people in application areas could avoid the difficult transition to parallel computing – which requires parallel programming – by buying a bigger machine with a small number of faster processors. But today, at around 3 GHz, the processors are coming to a physical boundary, and going into the parallel is your only chance.

MM: That's something we all have to face, the problem of getting a complex algorithm, either composed of different models or involving multiple dimensions, to a multicore computer.

HB: I always give this analogy: If a field needs to be ploughed, typically a farmer would prefer four big oxen to one billion ants. But what we will have in the future is one billion ants, and the farmer has to think about how he can do his classical jobs, no longer with four strong oxen but with the ants. And that's the technological challenge we have now.

PR: Coming from a farming family, I can say that sounds like a discouraging prospect. But I get the point: A change as radical as that is unavoidable for high-performance computing. What are the implications?

MM: In the past if you did parallelization, 64 processors was already a lot. That's what you still hear in some fields. But now we have to face a hundred thousand processors, and we have to do completely different things. For example, you have to handle faults. You have hardware that doesn't work as you would expect it to. Think about having a hundred thousand cores, and you can calculate the probability that something doesn't work.

HB: It will get even harder to get pieces of software that run decently, to get data into and out of the computer, to get the data processed, and also to extract the knowledge out of the numbers. Imagine that you have just one set of data with 10^{18} bytes – someone has to tell us what this means, whether it is a weather forecast or some technical project. Parallelization, and in particular in a massively parallel way – not tens or hundreds, but hundreds of thousands of processors – is the biggest issue. However, this goes beyond a mere parallelization in the sense

Hans-Joachim Bungartz, Miriam Mehl

of taking existing codes and algorithms and turning them into something parallel by force. Many current algorithms have intrinsically sequential parts. Think of a coffee machine – the standard joke that coffee comes first and the cup drops out at the end is an example of a sequential part of an algorithm. One interpretation of that, called Amdahl's law, led to a very critical perspective on parallelism for decades. However, not all that is sequential in an algorithm is enforced by the underlying problem. Sometimes, we just have to think about a completely new algorithm design. To "think parallel" – designing algorithms and programs that are inherently parallel, and which could only be sequentialized by force – that's what is needed.

PR: This transition to larger-scale multicore processing is just part of a whole constellation of "multi" issues that come up in your group's research plans and publications: multiphysics – meaning the coupling of multiple physical models – multidimensional, multiscale, multilevel. What's the best way to sort them out for people, like me and most of our readers, who are outside this field?

HB: There is a way of classifying all these multis. Maybe this reduces the jungle a bit. There are multis that come from the problem, which is typically multiscale. A phenomenon like turbulence, for example, is multiscale. You have very tiny vortices, but these tiny vortices define the macroscopic picture. Multidimensional is also coming from the problem, because you have the dimensionality in the problem. Then you have a second group of multis that deal with the algorithms, how you want to work on these problems. Multilevel for example is a classical approach to tackling multiscale; it's the algorithmic weapon that lets you represent a multiscale phenomenon in a very efficient way.

And then you have a third group, not the algorithmic but the technological weapons, such as multicore. Multicore has nothing to do with the problem, has not that much to do with the algorithms; that's just the machine part. If you think in terms of the three boxes associated with problems, algorithms, and hardware, then maybe there's less chance for confusion about all these multis and the interplay between them.

MM: They are all related with each other, and that means you cannot just focus on one topic. You need them all to be high-performing in the end. Why do we do multiphysics? Because one model is not accurate enough. Typically you neglect something. If you simulate an airplane only simulating the flow, you don't include the flexing, up-and-down movements of the wings and their interaction with the flow again. So to be more accurate you need multiphysics, and then you have to be accurate on each field. And for that you need the multicore. And if you want to optimize in addition you have multiple parameters, and then you are in the multidimensional field.

DP: To tackle today's challenges, you can't as a scientist just choose a problem, go back to your room, and come back a few years later with a solution. That just doesn't work any more. The problems are too complex. So you need to bring different ideas together, and you need a group that tackles a variety of sub-problems. That's one of the big advantages of the TUM-IAS Focus Group.

Patrick Regan

Valeriy Khakhutskyy and Dirk Pflüger

PR: That puts the spotlight on the other big "multi" in the picture, multidisciplinary. I'm interested in understanding how the group divides the problems and melds the diverse activities into an integrated program bigger than the sum of the parts.

HB: There's a strong bias toward the multidimensional right now, with Markus, Christoph, Dirk, and Valeriy working mainly on that. Miriam is concentrating on multiphysics. We're hoping to be joined in 2012 by someone who will focus more on multicore; for now I'm wearing the multicore hat, which is natural since I serve on the directorate and steering committe of the Leibniz Supercomputing Center. I'm eager to spend more time thinking about parallel concepts, working on parallel solutions for important problems where the current state of algorithmics faces huge roadblocks. But to be honest, finding time for that will not be easy for me, since I'm also serving as the "glue" for our Focus Group, helping to keep the three threads together.

As Miriam said, these things are interwoven, much more than you'd suspect from the way they have been addressed in the past. Typically the groups are quite separated, especially the multidimensional, which lies more in classical computer science or applied mathematics and not that much in physical modeling. But here we are bringing in a classical multiphysics problem from plasma physics and trying to combine it with a multidimensional approach and then to bring it into a multicore machine.

That's a unique chance we have here with this group, but it calls for an orchestrator, so that's my main role.

PR: Sometimes it helps to have a musician in the group, any kind of group.

HB: It's true, over the weekend I was playing Beethoven and Bruckner. Yes, I like the orchestrator metaphor better than glue.

PR: One common theme seems to be that models and predictive analyses are inherently limited for the same reason that they work, because you've abstracted something out of reality. Could you tell me more about the various ways you're working to close the distance between simulation and the real world, or between prediction and the way things happen?

MH: Let's consider the multidimensional aspects. In the early days of computation, the first flow simulations for example were in two dimensions. That's as you say an abstraction from the real world, and the real world is three-dimensional. So it was a big improvement of course when we got simulations that could really deal with three-dimensional effects; that's especially important in fluid flow. But now we're looking at breaking the barrier and adding even extra dimensions. Why do we need extra dimensions? Well, let's say with fluid flow, in addition to our three-dimensional spatial variables, we can have different velocities, for example in gases, in the same places. So we have an additional three dimensions, and that gives us six. Now why don't we have that in traditional fluid dynamics? That's because a distribution is assumed at every point. We assume a so-called Boltzmann law or a Gaussian law of the velocities. But for realistic simulations, in many areas we need at least six dimensions.

VK: My research will be on data mining in high dimensions, where we have problems with ten, twenty, or a hundred dimensions. We're looking at actual data, any measured data, either from simulations or from real measurements, and here "dimensions" means the number of parameters that we can simultaneously deal with. For data mining this means

Christoph Kowitz and Markus Hegland

having lots of data, looking at the relationships between the data, and trying to induce the rules and make predictions. The challenge is how to master these problems using methods from computational science. And there are a lot of applications.

For example, one of the next projects we'll be working on has to do with time series predictions in all kinds of applications, like financial series or physics. Everywhere we have data depending on time, measures are taken every moment of time, and in order to get the prediction correct, you have to consider everything that happened in the past. That's where high dimensions can appear.

MH: Talking about predictions, that is a very important problem too. Starting in the 1990s, for example, we worked together with insurance companies and predicted risk. Insurance companies need to be able to predict the risk of having an accident, and banks need to predict the risk of default. These predictions are based on features, lots of different features that will allow you to predict the risk that some event will happen. Similarly, companies like Google and Amazon try to predict what an individual might be interested in, and agencies involved with security or immigration control have questions that are basically not so different.

The main tools used for prediction are mathematical functions. You can have functions of many variables, and of course each variable can take on many different values. If you have a new customer, typically this new customer will have features that are different from the ones you've seen in other customers. How do you interpolate between all these other customers? So you have a function with 20 variables, and you need to interpolate what you've seen before. Or you need to do a regression. That's a classic problem, which Valeriy and Dirk have also worked on. And of course my main interest is in computation.

How can we do this fast? And there's this curse of dimensionality. So one way is to calculate the values of your functions for all possible customers. And again you have this curse of dimensionality. So if you have your first feature, and you can take ten different values, and the second one can take ten different values, you have a hundred combinations. It's a combinatorial problem really. And if you have a third feature, you know, you have already a list with a thousand values. If you have ten features, you have ten to the ten different values. That's a lot of values. And it's infeasible to even store these things. That's one reason we use so-called sparse grids. But another interesting aspect is that a colleague of mine, Jochen Garcke, found that some of the traditional combinatorial techniques we used were unstable.

PR: Meaning what?

MH: Unstable means basically that small perturbations or small errors can have a big effect. That's something that you know from extrapolation. These techniques are related to extrapolation. Weather forecasting, for example, is extrapolation. It's very difficult. You can get it wrong. The further you want to extrapolate, the worse it is. So you tabulate on a regular grid, and you want to extrapolate what is in between the values on the grid. And you have exactly the same effects as if you would like to predict what happens to the stock market in the future. It can go wrong. Extrapolation is, we call it, inherently unstable. But we have an answer. We have a cure for this instability. We have a stable approach based on relatively classical numerical techniques, an idea in numerical analysis; you can show that you can solve these problems, partial differential equations, by relating them to minimization problems. Minimization problems are inherently stable, whereas extrapolation problems are inherently unstable, and we found a way to cure this instability by reframing it as an optimization problem. I'm working also with Christoph on applying this concept, which originally came from our work in data mining, in more general contexts in physics.

PR: I'd like to hear more about this "curse" of dimensionality, which comes up regularly in group members' papers. What exactly does it mean?

CK: It may be easiest to explain in connection with an application. I'm collaborating with scientists at the Max Planck Institute for Plasma Physics in Garching. The larger context is research toward a future energy source from nuclear fusion rather than fission. The basic idea is to magnetically confine isotopes of hydrogen in a really hot vessel so that nuclei fuse together and produce energy.

The problem there is they want to simulate the behavior of a hot fusion plasma in this vessel, a so-called tokamak. The simulations are done to be able to confine the plasma effectively, to prevent the particles from going out of the interesting zone, the hot zone in the plasma, so that they are fusing. You have a spectrum of applicable models. You can have a single-particle description, where you just take every plasma particle for itself, to look how it's moving through the magnetic field. At the other end of the spectrum you can have a fluid-like description of the plasma, where you say, this is a continuum, or more or less a gas, which is acting a bit stranger than a regular gas. And somewhere in between these extremes are the gyrokinetic simulations that the IPP physicists are doing. They don't have single particles – the particles are not resolved themselves – what you have is more like a statistical description of these particles; but it's still more complicated and more detailed than a gas-like description or a fluid description. So, somewhat "in between."

What the physicists are especially interested in is simulating the turbulence, with high resolution in space and time, to be able to understand how it works in detail. And there we're not just using a three-dimensional fluid simulation, but a five-dimensional simulation, a so-called gyrokinetic simulation. The problem is now if you want to simulate something in five dimensions, you have to resolve it in this space. And just to get a moderately high resolution in this space, you already need vast amounts of data points. So there are huge amounts of data you have to handle and you have to do computations with. Here we've really come to the curse of dimensionality.

Just imagine a really small one-dimensional space. If you want to have a cube in one-dimensional space, it's basically just two points, left and right. If you extend that cube to two dimensions you get a square, and that's already four points. If you extend it to three dimensions, you already get eight points, which is a three-dimensional cube. Now imagine just adding two more dimensions. You're suddenly at 32 points for that cube, and this is just the smallest cell that you can resolve – but for this plasma physics code you need vast amounts of these unit-squares or unit-cubes. For current simulations, which run on supercomputers for days or even weeks, they have up to one billion data points, which is 200 gigabytes of data they do computations with.

For each of them you have to do some computations, and then you move on to the next time step. Then you do all the calculations again, and then again you move a tiny bit forward in time. But only that way are you able

to resolve the turbulence, to understand what's really going on in that tokamak. If you don't understand that, then it will be harder to confine that 200 million degree plasma.

PR: And this is without even considering any multiphysics aspects of the fusion plasma simulation.

CK: That's right. I'm focusing at this point on ways to handle the multidimensionality of the gyrokinetic turbulence simulation using the IPP code called GENE.

PR: Meanwhile, Miriam, you are leading the multiphysics effort. I guess the most familiar example of multiphysics would be the coupling of atmosphere and ocean models to produce more reliable weather reports or climate simulations. Here you're focusing mainly on coupling flow models for fluids such as air, water, or circulating blood with models of flexible solid structures. What are the difficulties, and how are you addressing them?

MM: Imagine two people meet at a conference and decide to couple one code that solves for fluid flows with another that does structural mechanics computations. You first have to tackle this technically, how to make these two codes talk to each other, and then you probably notice that you get something that is not reasonable at all. You may get for example large oscillations, wings breaking away from the plane, and that's just the numerical instability. So then you have to add something there, and once you have done all this, you probably want to concentrate on efficiency, how to make the solving fast, not only stable, and of course also accurate. And the next step will be to bring this onto a large machine, so there of course you have additional difficulties. We have several codes, they should run in parallel, preferably in a load-balanced way, so you should prevent a situation where one part is ready very fast and the other one runs over a long time and you have idle processors. It's a challenge to run just one code efficiently on a multicore machine.

One of the things we use to address multiphysics problems is a coupling environment called preCICE, which was originally developed in our group under a grant from the German Research Foundation. That was originally developed for fluid-structure interactions. The idea was to have something that works in a way that computer scientists always like things to work, in a very modular way. The idea is you have two solvers, fluid solver and structure solver, and you just plug them together with this glue software, and then everything works. And once you decide you want to exchange your flow solver – because you want to simulate a different application with different needs, and you've found a specific solver that works for that – you exchange it and all the other parts remain unchanged.

That's something you don't find in currently available commercial tools. There is commercial multiphysics software, but it's not modular in this sense. There also is coupling software, but it doesn't have the full functionality. You don't just need a tool that performs data communication between two codes. There's also numerics in it: So, how do you iterate, how many

times do you call the flow solver, the structure solver, in one time step, and in which order, and what data do they exchange? There is a lot that should be in that central coupling unit, because if it's in one of the solvers, you have to redo everything every time you exchange one of the parts. That was the original idea, and it has more and more become a generic tool for coupling different physics. That's the direction we are going, to make it more generic, not only for fluid and structure, but also to make it work for other kinds of multiphysics applications. And on the solver side, we've found we can greatly enhance performance by implementing certain kinds of data structures, particularly tree-structured adaptive computational grids. We have a home-grown flow solver called Peano that brings both hardware efficiency and numerical efficiency to simulations of fluid-structure interactions.

PR: A concept that comes up a lot in your papers, in all kinds of different contexts, is "sparse grids." What does it mean, and how does it help?

DP: If you are dealing with high-dimensional problems, you can't just use your brute force approaches, the classical ones. You have to try to represent the important structures with as little effort as possible, and you have to try to adapt to certain properties of the problem at hand. With sparse grid techniques you focus on the most important things first, and then try to express the less important parts, and if you can start to neglect some of those parts, you can tackle really high-dimensional problems. And that's where my focus is, trying to express, with as little work as possible algorithmically, computationally, the high-dimensional problems.

We're also building a software toolbox called SG++, for dealing with these high-dimensional problems using spatially adaptive sparse grids. We're trying to include the basic ingredients for different types of high-dimensional problems, from plasma physics to data mining to other problems, option pricing in finance for instance, bringing those things together to provide the means of tackling that whole range of problems.

HB: If you have one dimension, you have n points. If you have d dimensions, then you have n to the d points. So n square, n cube. So to achieve a specific accuracy for your problem, you have to invest n to the d points. If d increases, this is the curse of dimensionality. So the question now is: Is there any strategy you could develop where you can do only with n – that is, where you always invest, say, 117 grid points, and it doesn't matter if you are in a one-dimensional or a 15-dimensional domain. And the answer is funnily yes, and this is Monte Carlo; there you don't work with grid points, but you work with degrees of freedom. So you need 117 shots, and you can get similar quality, and the problem is not interested in the dimension of your domain. But the drawback is that you get a very lousy approximation quality.

So that's the reason why so many people work with Monte Carlo. It works, it is easy. But as soon as they have something else, they turn to something else, because something else is basically always better. And now the question was, is there anything else? This was the idea of sparse grids, and the result is now – I simplify strongly but – the result is that you get an algorithm that lets you work with a number of grid points that only moderately depends on the dimensionality and still obtain the same quality that you get with the n to the d approach. So that means you get a product of the same quality but for a much smaller price. It is attractive for three dimensions – you can just increase your resolution and make things finer – but it is essential for the higher-dimensional case, because now you can do something numerically in situations where you could not do anything numerically apart from Monte Carlo before. That's a very rough idea of it.

PR: Research involving sparse grids has a history in Munich, doesn't it – including your own contributions?

HB: It has a history in Munich, and it has a history beyond Munich. The new era of sparse grids started in Munich around 1990 with the work of Christoph Zenger and his PhD students at that time. The older part of the story is like always. I always say you have to do this "cherchez le Russe" – you have to find

the Russian or the Chinese guy who did it before. In the case of sparse grids, we haven't discovered the Chinese guy yet, but there are definitely Russian guys in the 1950s, in approximation theory. But no one did anything practical with it, or even thought of solving partial differential equations. The equations are something that definitely came with this reinvention, but the mathematics behind it goes way back, to Archimedes. So now I have to draw a picture. The task of Archimedes was to get the area of the parabola, but he didn't have calculus, he didn't know about an integral. But he could make points here, and then trapezoids, and calculate the area of each trapezoid and sum them up. As the story goes, Archimedes' wife came along and said that's not accurate enough, so you have to introduce additional ones. And the problematic thing is then you have to forget everything you've calculated so far; you have to do it completely from scratch for every change in the grid points, which is extremely expensive if you don't have a pocket calculator.

Good mathematicians are always lazy. So he thought about something to improve that, and this is where this picture comes in, where he invented this so-called hierarchical basis. He said let's start with a big triangle, and let's add smaller and smaller triangles. And you see with this first big triangle, it already is quite a decent approximation. And if you add the next two, then you've already improved the precision. This is how Archimedes in the third century before Christ was basically defining all the ingredients you need for the sparse grids. Not in higher dimensionality in this case – that came later – but the main idea.

So I would therefore say that the first hour as far as we know so far is actually Archimedes sitting on the beach and thinking about integrals.

At this point, laughter was heard from the Australian end of the videoconference.

MM: And now they're sitting on the beach.

HB: They are trying to mimic this, sitting on the beach and thinking and getting great ideas like Archimedes.

PR: That brings me to a simple, practical question. You've explained a lot about what makes the group a group, such as the interconnectedness of the problems and the active role of Hans as orchestrator. But I'm also interested in how the advantages of this geographically challenging collaboration outweigh the potential disadvantages.

HB: We met Markus before, at conferences, but thanks to the TUM-IAS we have now a framework to collaborate in a deeper way – together with time and free space to think about things – and I think that's a very important contribution you cannot overestimate in that context.

DP: We are spending a great effort to bridge the gap between Australia and Germany. Currently Christoph is over there, and I was in Australia the last three months of 2011. At the beginning of June, Markus will be here. We have video group meetings on a regular basis.

MH: We talk to each other of course when there are pressing scientific questions that we need to discuss. But in addition to that, our colleagues from Munich join by video in an extended research group meeting I have every Wednesday, typically a group of ten or so including five or six doctoral candidates.

CK: When I'm here at ANU, I can work intensively with Markus, and that's definitely one advantage. But it's not only Markus. We are now working together with Mike Osborne and other scientists here. I'm not a mathematician, so for me it's always

interesting to hear what the mathematical PhD candidates or professors are thinking. It's interesting to get in this different environment where people see my problem in a completely different way, or bring up problems where I don't see any problems at all.

DP: When I was there, we even started new projects. We were able to look at new scientific problems, which is not that easy to do via electronic communications alone.

PR: Can you give an example?

DP: There is one subtask I ran into using sparse grids for classification. We found a solution that works, but we didn't understand how it works or why it works from a theoretical point of view. Now, bringing together our hands-on approach from computer science and Markus's knowledge from mathematics, we've been able to study that and gain a deeper understanding. This is something that needs plenty of discussions, that needs intensive cooperation, and that just doesn't work electronically, in my experience.

MH: And of course this close connection to Munich, to Garching in particular, has advantages for me and my research group at ANU.

HB: I think if you look on a European scale, from a computational point of view, there is no better place to do this. Facing each other across Boltzmannstrasse in Garching are, on one side, our computer science and mathematics departments and the Leibniz Supercomputing Center, and on the other side, the Max Planck Institute for Plasma Physics (among others!) and their computing center. Complementary expertise in theory, practice, and applications situated right here, and now we have crossed the street. But beyond that, there's another special advantage over other places that have big machines, challenging applications, and active research – that is the ease with which we can bring in the young scientists, for example through TUM's International Graduate School of Science and Engineering and our computational methods study programs.

Going back to the "multi" of multidisciplinarity: It is invoked today actually in an inflationary way, but I think it's really true for this group, and it can be an especially mind-opening experience for young scientists. In your everyday work you can be trapped in your discipline. That starts with teaching, with the colleagues you typically meet, and with your way of thinking. But here we can really bring together mathematics with computer science and with the applications in physics and engineering, with a great scope for different ideas and perspectives. It can be difficult to get your mind open in this way if you are 50 or above – I can say this because I am 49 – and I think it's important that we really put this into researchers' lives very early. Then you see the mixture as a natural thing, and monodisciplinary research as more or less the exception. That's where we should be heading, and I think this is a splendid opportunity where we can actually live it. ■